A Novel Variant of QGA with VNS for Flowshop Scheduling Problem

Muhammad Shahid Latif and Hong Zhou
Beihang University of Aeronautics and Astronautics, Beijing, China
School of Economics and Management
Email: shahid_phy@yahoo.com; h_zhou@buaa.edu.cn

Amir Ali
Beihang University of Aeronautics and Astronautics, Beijing, China
School of Computer Science and Engineering
Email: amir_12877@yahoo.com

Abstract—**In this paper, scheduling problem of flowshop with the criterion of minimizing the total flow time has been considered. An effective hybrid Quantum Genetic Algorithm and Variable Neighborhood Search (QGA-VNS or QGA$_{VNS}$) has been proposed as solution of Flow Shop Scheduling Problem (FSSP). First, the QGA is considered for global search in optimal solution and then VNS has been integrated for enhancing the local search capability. An adaptive two-point crossover and quantum interference operator (QIC) has been used in quantum chromosomes, which is based on the probability learning and quality of solution at each iteration. Further, a Longest Common Sequence (LCS) method has been adopted to construct the neighborhood solutions for intensifying local search with VNS. The neighborhood solutions will be based on the common sequence similar to the longest common sequence in global solution in each iteration, represented as LCSg. After selection of individual, VNS will be applied further exploring the local search space based on LCS neighborhood solutions. Results and comparisons with different algorithms based on the famous benchmarks demonstrates the effectiveness of proposed QGA-VNS.**

Index Terms—**Flowshop scheduling, QGA, VNS, LCS, Quantum Population**

I. INTRODUCTION

In recent years, the hybridization of different evolutionary algorithms has been successfully implemented for scheduling problems and has proven the effectiveness of this approach. In literature, the flowshop scheduling problem (FSSP) has been solved with different evolutionary algorithms hybridized with each other. In this paper, the hybridization of QGA with VNS has been proposed. To the best of author's knowledge, no such attempt has been reported or published before, which considers hybridization of QGA and VNS for flowshop scheduling (FSSP) with total flow-time minimization criterion. In recent decade, hybridization approach has emerged as a prominent and effective tool for taking the advantages evolutionary algorithms to produced promising optimal solutions.

A. Quantum Genetic Algorithm

The standard QGA was implemented for scheduling problem initially by [1]. Similarly, QGA was implemented successfully for first time for Flowshop scheduling problem (FSSP) by [2]. In literature, the hybridization of QGA has been reported many times for FSSP. It is obvious; merging the advantages of different evolutionary algorithms can be effective, efficient and produces better results. For FSSP many hybrid heuristics have been designed which show excellent robustness in terms of quality solutions, performance, and superior algorithms in their standalone structures. In last few years, for FSSP many algorithms have been integrated with QGA. The hybrid Genetic Algorithm (HGA)[3], Hybrid Particle swarm optimization (HSPO)[4], Hybrid Simulated Annealing[5], Hybrid Immune Algorithms[6], Quantum Differential Evolution Algorithms[7], and many other heuristics can be searched in literature which has been integrated with QGA for other scheduling problem like RQEA[8].

The quantum evolutionary algorithm (QEA or QGA) is based on the concept and principals of quantum mechanics like quantum chromosomes and parallelism of sates or superposition of quantum states. The significance of QGA is keeping the balance between exploring and exploitation search space for optimal solution. The design of QGA is based on the Q-bit, a smallest unit which contains the information, a rotation gate for variation and updating the quantum population and encoding through random keys[2], for sequencing and order problems and an observation process which is used to convert Q-bits in binary "1" or "0". Mathematically the QGA can be described as follows:

The state of a quantum bit (Q-bit) can be represented as, $|\Psi\rangle = \alpha|0\rangle+\beta|1\rangle$. Where α and β are complex numbers that represents the probability amplitudes of the corresponding states. Therefore, α^2 and β^2 denote the probabilities that the Q-bit will be found in the "0" state

Manuscript received January 10, 2014; revised March 25, 2014; accepted April 9, 2014.

doi:10.4304/jcp.9.9.2191-2197

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 can be 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,..., 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 probability of quantum states can be represented like $-\frac{1}{2\sqrt{2}}|00 + \frac{\sqrt{3}}{2\sqrt{2}}|01 + \frac{1}{2\sqrt{2}}|10 - \frac{\sqrt{3}}{2\sqrt{2}}|01$. 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.

B. Variable Neighborhood Search (VNS)

The second major content of this paper is comprised on the most recent heuristics named Variable Neighborhood Structure (VNS)[9]. Due to inherited generic simplicity and efficiency, often VNS has the most targeted choice when hybridization is required. On the other hand, the VNS exhibits the capability of finding good solutions when used with other heuristics. Recently new approaches have been developed where VNS has exhibited its capability either standalone or hybridized with other evolutionary approaches. In recent decade, VNS has been hybridizes with many heuristics for solving FSSP problem. These approaches proved to be extremely effective and produced state of art results. Most recent and significant contribution has made by [10, 11]. In mentioned approaches, most researchers has combined the two well-known neighborhoods structures named as job insert and job interchange. Both of these neighborhood structures are combined in the similar order by [12-14]. As far as the concern of flowshops literature, the authors of [15], have tested different ways to combine these two neighborhoods. They proposed a hybrid heuristic with particle swarm and VND for investigation of no-wait flowshop problem. The experiments executed by [15], investigated the hybridization of a PSO with two above mentioned VNS approaches, one is initiated with job insert and the other is started with job interchange. They have generated all neighborhood solutions before the decision which solutions to be migrated. The result has shown that job interchange structure as the initial neighborhood proved to be more effective and yield best performance on tested instances of the no-wait flowshop. Most recently [11], has examined the six distinct combinations of most used neighborhood structures for FSSP. In [11], they have provided comparison of their novel VNS4 with other approaches known as AGA and V4AGA, both are mentioned above. The VNS is known for its simplicity and effectiveness. The standard VNS [9], gets an initial solution which belongs to global search space. After that, VNS mostly uses two procedures named as shake; which diversifies the solutions by changing to different local neighborhoods. The second named as local search; which explores for an improved solution within local neighborhood. For more detailed study of VNS the reader may refer to [9].

The paper has been divided into five sections, focusing on different aspects. In Section II, basic formulation for Flowshop has been described. Section III describes the proposed hybrid algorithms QGA-VNS or QGA$_{VNS}$. Different strategies are discussed which are implemented. Section IV presents the computational experimentation and comparison of results with standalone algorithm like GA, EDA, QGA, EDA, and hybrid format with VNS like PSO$_{VNS}$, NEG$_{VNS}$. Further, these results are tested and compared with Carlier, Reeves and Taillard [16-18] benchmark for different instances. Finally, in Section V the conclusions and directions for future research are discussed

II. Mathematical Formulation of Flowshop Scheduling Problem (FSSP)

The Flow-shop scheduling has a well-known history and is renowned for its NP-hard nature. In manufacturing industry, 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 sequence. 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.

$$C(\pi_1,1) = p(\pi_1,1) \tag{1}$$

$$C(\pi_i,1) = C(\pi_{i-1},1) + p(\pi_i,1) \square for\, i = 2,\ldots..,n \tag{2}$$

$$C(\pi_1,j) = C(\pi_1,j-1) + p(\pi_1,j) \square for\, j = \quad \ldots \quad m \tag{3}$$

$$C(\pi_i,j) = \max\{C(\pi_{i-1},j), C(\pi_1,j-1)\} + p(\pi_i,j) \tag{4}$$

$$for\, i = 2,\ldots..,n; \square j = 2,\ldots..,m$$

$$C\square\_\,max(\pi) = C(\pi \quad n \;\, m) \tag{5}$$

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

III. Proposed Hybrid QGA-VNS or QGA$_{VNS}$

Both QGA and VNS have been widely used in hybrid format for mining good optimal solutions at very low computational cost in terms of time and memory of machine. QGA known for finding optimal solutions in a very short time and with considerably small size of population. While VNS is known as one of the best method for local search, hence, frequently hybridize with many heuristics.

In proposed QGAVNS algorithm, QGA is used for global search for optimal solution and then VNS is incorporated for exploring the local search space. The VNS is applied on the solutions provided by QGA after each iteration. Before implementation of VNS, the solutions has been built with application of Longest Common Sequence (LCS). In the proposed algorithm, the LCS production structure used by [19], will be incorporated. The difference between implemented LCS production, as shown below, is LCS productions scheme is not initialized with *NULL* value as done in [19].

In this study, the LCS will take initial value from the permutation sequence of last iteration which represents the best permutation (π) for optimal solution in current generation. This will represent like LCSg, which means this is base LCS for current generation and other will be carved out from rest of current population. Subsequently, , this LCSg is kept constant and similar LCS is found throughout current population by production method similar to the one in [19].

```
PRODUCTION $\pi_k, \pi_k^l$
$\pi_o \leftarrow$ Null. /*offspring initialization*/
For $j \leftarrow 1$ to n
    Generate a uniform distributed random number $\mu\left(\mu \in [0,1]\right)$.
    If ( $\pi_{k,[j]} \in \pi_k^L$ AND $\mu \leq \lambda$ )
        Insert $\pi_{k,[j]}$ to the position j of $\pi_o$ .
        $\Omega_u \leftarrow \Omega_u - \left\{\pi_{k,[j]}\right\}$.
For each unassigned position j in $\pi_o$ .
    Select job $I$ from $\Omega_u$ with probability $\frac{M_{[i][j]}}{\sum_{i \in \Omega_u} M_{[i][j]}}$.
    Insert job i to the position j of $\pi_o$ .
    $\Omega_u \leftarrow \Omega_u - \{i\}$.
Return $\pi_o$ .
End PRODUCTION
```

The propped algorithm is novel in the sense that no efforts so far has been reported where these two algorithm are used with each other in hybrid format. Further we have include three very effective techniques to make this hybrid method more efficient and effective named as Longest Common Sequence(part of conclusion) (LCS), Quantum Interference and Multi point Quantum Cross Over [19-21]. The first two has been proved very effective and used by many researchers for digging out good blocks of genes that mostly constitute the optimal solutions, while the later ones are kept optional. Although in QGA nothing exists like crossover or mutation and it is significant part of standard GA. But cross-over and mutation in QGA has been used by few researchers for solving the numerical optimization problems[20]. It has been proved to be effective, so this has been chosen to be implemented in proposed algorithms for the first time.

A. *Longest Common Sequence (LCS)*

If we consider a sequence like π which is subsequence of π_a and also subsequence of π_b, then π is said to be common sequence between π_a and π_b. According to the properties of common sequence as proved by [22] has dependent upon three different sub-problem the *first,* building the LCS between $\pi_A^{p-1}\pi_A$ and $\pi_B^{q-1}\pi_B$, *second,* building the LCS between π_A^{p-1} and π_b, the *third* building the LCS between π_a and π_B^{q-1}. Which is shown the overleaped sub problem characteristics and so dynamic programming (DP) can be used. The author in [22] has obtained the matrix of common sequences by following procedure shown below.

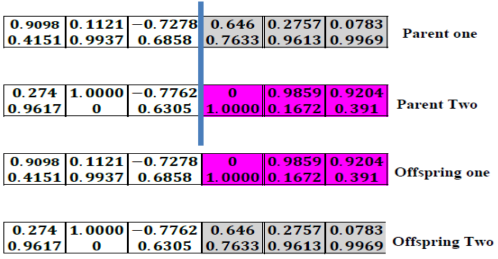
```
LENGTH_LCS( $\pi_a , \pi_b$ )
$p \leftarrow |\pi_a|, q \leftarrow |\pi_b|$ .
$C_{[i][0]} \leftarrow 0(\mathrm{i} = 1, 2, ..., \mathrm{p}), C_{[0][j]} \leftarrow 0(j = 0, 1, ..., q)$ .
For $\mathrm{i} \leftarrow 1$ to p
    For $\mathrm{j} \leftarrow 1$ to q
        If $\left(\pi_{A[i]} = \pi_{B[j]}\right)$, then $C_{[i][j]} \leftarrow C_{[i-1][j-1]} + 1, b_{[i][j]} \leftarrow$
        Else if $C_{[i-1][j]} \geq C_{[i][j-1]}$, then $C_{[i-1][j]} \leftarrow C_{[i-1][j]} + 1, b_{[i][j]} \leftarrow$ "↑".
        Else $C_{[i][j]} \leftarrow C_{[i][j-1]}, b_{[i][j]} \leftarrow$ " ← " .
Return c and b.
END LENGTH_LCS
```

In LCS, *p* and *q* are the two points, which correspond to the length of LCS. In the proposed algorithm VNS will be applied at p-1 and q+1, individuals. If no improvement can be found then in the next step VNS will be implemented p+1 and q-1 on individuals which are not include in LCS.

The idea of generating LCS has been implemented in a different way. In proposed $\mathrm{QGA_{VNS}}$, the LCS has initiated with reference to the permutation (π) sequences of jobs which provided optimal solutions in current generation. For example if π is permutation of which corresponds to optimal solution in current generation. Then this permutation will be considered a base or LCSg. Then similar common sequence will be found throughout current population by using the production method. The only common sequences considered which will contain the similar building as LCSg or permutation of job sequence for optimal solution in current generation. This procedure will be repeated after each iteration and LCS will be obtained. The LCS obtained after production method will be further undergo local search method. For the improvement of performance of proposed approach $\mathrm{QGA_{VNS}}$, a Variable Neighborhood Search (VNS) has been employed The VNS will be acted on the individual which are before the start of LCS or at the end of LCS. If no improvement has been found in certain iteration then VNS will act at different position within the length of LCS. Two effective and efficient neighborhood local search methods have been implemented named insert_local_search and swap_local_search. These are similar as introduced by [10], are adopted in QGA$_{VNS}$. The VNS will be make iterations on LCSg or current sequence π_c. These iterations will continue on current sequence until no further improvement is done through continuous 50 iterations. If the obtained solution is better than current sequence which contain LSCg, then this will replace the older solution and it will be included in a global search of QGA before next iteration.

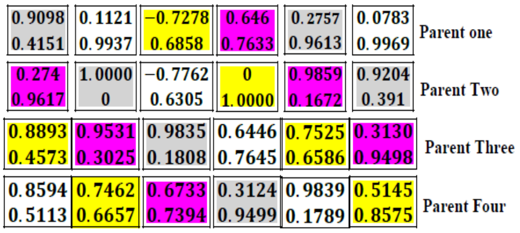
B. Quantum Interference and simple Cross Over

Although there does not exist crossover and mutation operator in standard QGA. But few researcher have applied the idea of simple crossover on quantum bits (Q-bits) and quantum interference operator[20]. In simple crossover, the constituents of Q-bits α and β in each chromosomes are chosen randomly and changes position of Q-bits α and β vice versa. In fact, there is only one difference in classical cross-over and quantum crossover. The later is applied on the probabilities of Q-bits within single quantum chromosomes. The illustrations shown in fig 1-3. These both figs. has been referred from[20].



0.9098 0.4151	0.1121 0.9937	−0.7278 0.6858	0.646 0.7633	0.2757 0.9613	0.0783 0.9969	Parent one
0.274 0.9617	1.0000 0	−0.7762 0.6305	0 1.0000	0.9859 0.1672	0.9204 0.391	Parent Two
0.9098 0.4151	0.1121 0.9937	−0.7278 0.6858	0 1.0000	0.9859 0.1672	0.9204 0.391	Offspring one
0.274 0.9617	1.0000 0	−0.7762 0.6305	0.646 0.7633	0.2757 0.9613	0.0783 0.9969	Offspring Two

Figure 1. Quantum Cross-over



0.9098 0.4151	0.1121 0.9937	−0.7278 0.6858	0.646 0.7633	0.2757 0.9613	0.0783 0.9969	Parent one
0.274 0.9617	1.0000 0	−0.7762 0.6305	0 1.0000	0.9859 0.1672	0.9204 0.391	Parent Two
0.8893 0.4573	0.9531 0.3025	0.9835 0.1808	0.6446 0.7645	0.7525 0.6586	0.3130 0.9498	Parent Three
0.8594 0.5113	0.7462 0.6657	0.6733 0.7394	0.3124 0.9499	0.9839 0.1789	0.5145 0.8575	Parent Four

Figure 2. QIC operator

In quantum interference cross over operator (QIC) Fig 4, the idea of inherited parallelism of QGA has been implemented on Q-bits. In this way, the QGA performance can be enhanced and more diversity in populations can be achieved, which further helps to find better optimal solution in shorter span of time.

The authors of [20], has claimed the better results than the results of simple QGA.

C. Structure of Proposed QGA-VNS (QGA$_{VNS}$)

Following, we will describe the procedures of hybrid QGA-VNS for solving the FSSP.

Step 1: Let t=0, Q-bits as an initial population are generated with equal probability. $P_Q(t)$ $\{p_1(t), p_2(t), p_3(t),..., p_N(t)\}$ where $p_i(t)$ denotes the i_{th} individual.

Step 2: Convert $P_Q(t)$ to a job permutation $P_p(t)$ and evaluate population $P_Q(t)$, and record the best solution b.

Step 3: If the stopping condition is satisfied, the best solution is exported. Otherwise, go on to the following steps.

Step 4: $P_Q(t)$ through rotation operation to generate $P_{Q'}(t)$ and $P_Q(t_{.}\,1)$ $_{.}P_{Q'}(t)$. $P_p(t)$ regarded as an initial population $P_{p'}(t)$. Evaluate population $P_{p'}(t)$ and update $P_p(t)$.

Step 5: Apply crossover on Q-bits and apply QIC, on $P_p(t)$ to generate $P_p'(t)$ .And convert $P_{Q'}(t)$ to $P_p(t)$ in to binary.

Step 6: Apply Production method on the converted $P_p(t)$ in to binary for generating nearly similar LCS as in best solution.

Step 7: Apply VNS as local search (Insert and swapping) on the produced LCS from the quantum population and evaluate the local solutions by VNS.

Step 8: Solutions are compared with current generation's best b with VNS, if better than replace it in QGA and adjust quantum bit positions and then perform updating operation through rotation gate.

Step 9: Let t=t+1 and go back to Step 2.

IV. COMPUTATIONAL RESULTS, COMPARISONS

Extensive experimentation was carried to demonstrate the capability and effectiveness of proposed QGAVNS algorithms. The algorithm was coded in MATLAB and executed on Pentium Dual core 1.6GHz PC. The size of quantum population was kept as 40. The length of Q-bit set to J*10 i.e. each 10 q-bits corresponds to a job. The crossover probability set to be maximum initially at 0.6 and further it will relate with ARPD. It will keep on reducing as ARPD value is reducing because ARPD equation (6) is representing the average deviation of results form a global best in in each generation.

$$ARPD = \sum_{i=1}^{R}\left(\frac{(X_i - X_{best})\cdot 100}{X_{best}}\right) / R \text{ ………... (6)}$$

where X$_i$ is solution produced by current algorithm and Xbest representing best solution to which comparison is being done. Where "R" is the number of replications. In our experiment we kept R=10, and termination criterion is set to (n*m/2)*30ms maximum computational time.

We have used different algorithms for comparison with our proposed algorithm. In this study NEH[23], GA[24], HQGA[2] have been compared against the benchmark problem of Car., Rec. Taillard [16-18]. Further, few hybrid algorithms have also been compared like, EDA, PSO$_{VNS}$, NEG$_{VNS}$[25, 26] for the Taillard[16] instances of medium sized i.e. n=50 and m=20. In comparison with different algorithm for benchmarking instances, proposed algorithm shows excellent performance in terms of solution quality and computational time for medium size problems. For the instances where n,m>50, the algorithm consums more computational resources and time. Therefore propped algorithm exhibits excellent performance for mid-size problems.

TABLE 1
AVE. MAKESPAN FOR DIFFERENT TAILLARD INSTANCES

Sr.No.	Prob. Size(n,m)	GA	EDA	PSO$_{VNS}$	NEG$_{VNS}$	QGA$_{VNS}$
1	20,5	8.83	0.23	0.03	0	0
2	20,10	7.79	0.13	0.02	0.01	0.03
3	20,20	5.54	0.18	0.05	0.02	0.02
4	50,5	16.91	2.22	0	0	0.01
5	50,10	16.88	3.06	0.57	0.82	0.80
6	50,20	14.15	3.18	1.36	1.08	1.43

TABLE 2
AVE. COMPUTATIONAL TIMES FOR DIFFERENT TAILLARD INSTANCES

Sr.No.	CPU Times (Min.)					
	Prob. Size	GA	EDA	PSO$_{VNS}$	HQGA	QGA$_{VNS}$
1	20,5	0.02	0.23	0.17	0.04	0.19
2	20,10	0.05	0.13	0.34	0.06	0.30
3	20,20	0.12	0.18	0.85	0.10	0.73
4	50,5	0.06	2.22	33.50	28.50	42.50
5	50,10	0.07	3.06	62.10	55.80	96.20
6	50,20	0.13	3.18	168.08	132.05	192.05

TABLE 3
COMPARISON RESULT FOR CAR. AND REC.

P	*J, M*	*C**	NEH	GA	VNS	HQGA	QGA$_{VNS}$
Car1	11,5	7038	0	0	0	0	0
Car2	13,4	7166	2.93	0	0	0	0
Car3	12,5	7312	1.19	0	0	0	0
Car4	14,4	8003	0	0	0	0	0
Car5	10,6	7720	1.49	0	0	0	0
Car6	8,9	8505	3.15	0	0	0	0
Car7	7,7	6590	0	0	0	0	0
Car8	8,8	8366	2.37	0	0	0	0
Rec01	20,5	1247	4.49	0.15	0	0	0
Rec03	20,5	1109	2.07	0.01	0	0	0
Rec05	20,5	1242	3.14	0.24	0.24	0.24	0.21
Rec07	20,10	1566	3.83	0.4	0	0	0
Rec09	20,10	1537	2.99	0.44	0	0	0
Rec11	20,10	1431	8.32	0.45	0	0	0
Rec13	20,15	1930	3.73	1.39	0.14	0.16	0.11
Rec15	20,15	1950	3.23	0.98	0	0.05	0.03
Rec17	20,15	1902	6.15	2.78	0	0.63	0
Rec19	30,10	2093	4.4	1.7	0.28	0.2	0.18
Rec21	30,10	2017	5.65	1.63	0.15	1.44	0.1
Rec23	30,10	2011	7.16	1.53	0.35	0.5	0.26
Rec25	30,15	2513	5.21	3.43	0.12	0.77	0.09
Rec27	30,15	2373	5.27	2.32	0.25	0.97	0.2
Rec29	30,15	2287	4.55	3.09	0	0.35	0.14
Rec31	50,10	3045	4.2	2.78	0.26	1.05	0.28
Rec33	50,10	3114	4.08	0.95	0	0.83	0.09
Rec35	50,10	3277	1.1	0.09	0	0	0.06
Rec37	75,20	4951	5.58	8.35	1.72	2.52	1.98
Rec39	75,20	5087	4.34	6.16	0.85	1.65	1.81
Rec41	75,20		6.69	7.66	1.19	3.13	1.37

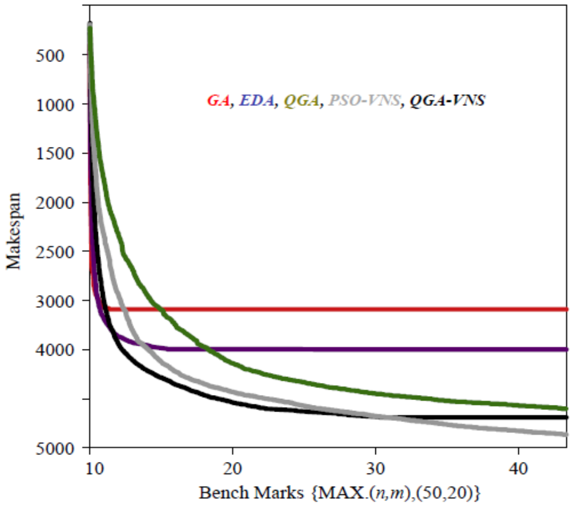


Figure 3 Convergence of different Algorithms for various prob. size (*n,m)*

In table. 2 comparison results are furnished for various instances and compared with other algorithms. These results demonstrate the effectiveness and capability of our proposed QGA$_{VNS}$. For the medium size problems, proposed algorithm has produced very good results and outperform standalone algorithm like GA, EDA HQGA. Further, this combination produced good results for medium size problems, which is better than the hybrid algorithms like HQGA, PSO$_{VNS}$. In table.3, average computational times are compared with other algorithm mentioned above. In terms of computational time, it is again quite efficient then standalone algorithms and performs better than the mentioned hybrid algorithms. In Fig.3, the overall ave. performances against makespan of different algorithms are shown. It is observed that the proposed algorithm outperforms the standalone algorithms like NEH, GA, EDA. While in case of hybrid format, it has shown excellent capability to perform better than HQGA, PSO$_{VNS}$ and NEG$_{VNS}$ for medium size problems.

V. Conclusion

In this research article, the authors propose a novel hybrid QGA$_{VNS}$ for solving permutation flowshop scheduling problem. They have used a standard QGA with quite recent VNS strategies like Jarboui [10]. Further studies will consider some improved QGA with most recent state of art new and developed strategies and structure of VNS. Few changes have been adopted to build up population based on the LCSg (LCS of best solution in current generation). All the solutions in current generation, which are similar to LCSg have been brought forward, therefore VNS can be applied as local search. Jarboui VNS structure has been adopted for local search. Before application of local search, different strategies are applied on quantum population to produce more diversity. Furthermore, it has undergone for the implementation of VNS. It has been compared with independent GA, NEH, EDA and standard QGA. The computational results in this study have shown the capability and effectiveness in terms of good solutions for the problem sizes up to 50*50. Therefore, in future it will be implemented on the problem size in which both n,m > 50. In future, proposed algorithm will be refined further and compared with state of art hybrid algorithms. The strategy which has been proposed in most recent VNS algorithms named as PHEDA[19] which outperform many state of art VNS structure will be considered for implementation for future studies.

Acknowledgment

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

References

[1] K.-H. Han, and J.-H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," pp. 1354-1360.

[2] 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.

[3] D. Z. Zheng, and L. Wang, "An effective hybrid heuristic for flow shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 21, no. 1, pp. 38-44, 2003/01/01, 2003.

[4] S. Chandrasekaran *et al.*, "A Hybrid Discrete Particle Swarm Optimization Algorithm to Solve Flow Shop Scheduling Problems." pp. 1-6.

[5] A. C. Nearchou, "A novel metaheuristic approach for the flow shop scheduling problem," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 3, pp. 289-300, 4//, 2004.

[6] R. Tavakkoli-Moghaddam, A. Rahimi-Vahed, and A. H. Mirzaei, "A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives:

Weighted mean completion time and weighted mean tardiness," *Information Sciences*, vol. 177, no. 22, pp. 5072-5090, 11/15/, 2007.

[7] 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.

[8] T.-C. Lu, and J.-C. Juang, "A region-based quantum evolutionary algorithm (RQEA) for global numerical optimization," *Journal of Computational and Applied Mathematics*, vol. 239, no. 0, pp. 1-11, 2/1/, 2013.

[9] P. Hansen, and N. Mladenović, "Variable neighborhood search: Principles and applications," *European Journal of Operational Research*, vol. 130, no. 3, pp. 449-467, 2001.

[10] 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.

[11] W. E. Costa, M. C. Goldbarg, and E. G. Goldbarg, "New VNS heuristic for total flowtime flowshop scheduling problem," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8149-8161, 7//, 2012.

[12] Y. Zhang, X. Li, and Q. Wang, "Hybrid genetic algorithm for permutation flowshop scheduling problems with total flowtime minimization," *European Journal of Operational Research*, vol. 196, no. 3, pp. 869-876, 2009.

[13] M. F. Tasgetiren et al., "A discrete artificial bee colony algorithm for the permutation flow shop scheduling problem with total flowtime criterion." pp. 1-8.

[14] X. Xu, Z. Xu, and X. Gu, "An asynchronous genetic local search algorithm for the permutation flowshop scheduling problem with total flowtime minimization," *Expert systems with Applications*, vol. 38, no. 7, pp. 7970-7979, 2011.

[15] Q.-K. Pan, M. Fatih Tasgetiren, and Y.-C. Liang, "A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem," *Computers & Operations Research*, vol. 35, no. 9, pp. 2807-2839, 9//, 2008.

[16] E. Taillard, "Benchmarks for basic scheduling problems," european journal of operational research, vol. 64, no. 2, pp. 278-285, 1993.

[17] C. R. Reeves, "A genetic algorithm for flowshop sequencing," *Computers & Operations Research*, vol. 22, no. 1, pp. 5-13, 1//, 1995.

[18] J. Carlier, "Ordonnancements a contraintes disjonctives.," *Rech Opér*, vol. 12, no. 4, pp. 333-350, 6//, 1978.

[19] Y. Zhang, and X. Li, "Estimation of distribution algorithm for permutation flow shops with total flowtime minimization," *Computers & Industrial Engineering*, vol. 60, no. 4, pp. 706-718, 5//, 2011.

[20] A. M. Mohammed *et al.*, "Quantum crossover based quantum genetic algorithm for solving non-linear programming," pp. BIO-145-BIO-153.

[21] D. Hongwei, and L. Cunhua, "Improved Quantum Interference Crossover-Based Genetic Algorithm and its Application," pp. 35-38.

[22] T. H. Cormen *et al.*, "Introduction to algorithms, 2006," vol. 2nd ed.

[23] M. Nawaz, E. E. Enscore 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.

[24] 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.

[25] G. I. Zobolas, C. D. Tarantilis, and G. Ioannou, "Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm," *Computers & Operations Research*, vol. 36, no. 4, pp. 1249-1267, 4//, 2009.

[26] M. F. Tasgetiren *et al.,* "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1930-1947, 3/16/, 2007.

Muhammad Shahid Latif is research scholar in the School of Economics & Management. Shahid has obtained his master degree in applied physics from Quaid-e-Azam University, Islamabad. He has also completed M.Sc. in engineering management from university of engineering and technology, Taxila, Pakistan. Currently he is doing his PhD, and his major area of interest is production scheduling with metaheuristic algorithm.

Hong Zhou is a professor in Beihang University, Beijing, China. His research interests include production planning and scheduling, Project management. His research interest includes the application of metaheuristics algorithms to the combinatorila optimization problems.

Amir Ali is a master student in School of Computer Science and Engineering. He had obtain his bechalor in Computer Engineering from NED university Karachi, Pakistan. His research interest includes cyber Seceuirty, Cloud Computing and implementation of heuristic algorithms in controls and image processing.