Modified Heuristics for Scheduling in Flow Shop to Minimize Makespan

Tariq S. Abdul-Razaq **Hanan A. Chachan**

dr.tariqsalih@yahoo.com hanan_altaai@yahoo.com

Faez H. Ali

faezhasan@yahoo,com

Al-Mustansiriyah University

College of Sciences - Department of Mathematics

Baghdad – Iraq

Abstract: *The NP-completeness of flow shops scheduling problems has been discussed for many years. Hence many heuristics have been proposed to obtain solutions of good quality with a small computational effort. The CDS (Campbell et al) and NEH (Nawaz, Enscore and Ham) heuristics are efficient among meta-heuristics such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA).*

This paper discusses some methods and suggests new developing to the methods of the scheduling in flow shop to minimize makespan problems. Our main object in this paper, from one side, is to improve efficient heuristics which will be better than the *existing heuristics given in the literature and yield solutions within a short time like Simple Heuristic Methods (SHM) and the First Heuristic Decreasing Arrange (DR). From other side, we apply two local search methods like GA and PSO algorithms on flow shop problems.*

Experimental analysis has been given of the performance of the proposed heuristics and local search methods with the relative efficient existing heuristics.

Keywords: ***Flow Shops Scheduling, Particle Swarm Optimization (PSO), Genetic Algorithms (GA).***

1. Introduction

Today's rapidly developing world has increased the need for timely and cost effective production. Many real live scheduling problems are more complicated than those of single machine scheduling, where jobs consisting of more than one operation need to be processed on more than one machine.

The **Flow Shop Scheduling Problem (FSSP)** can be defined as sequencing a set of n jobs on a set of m machines. All jobs visit the machines in the same order, i.e. first machine 1, then machine 2 and so on until machines m, the input data to the problem consists of matrix of processing times. Therefore Pj,k denotes the processing time of job j (j=1,2,…,n) on machine k (k=1,2,…,m). The (FSSP) has received considerable attention from researcher since it was introduced more than five decades ago in the well known paper of Johson (1954). The makespan equal to the time at which the last job in the processing sequence is finished at machine m. Makespan minimization for m-machine flow shop has been addressed in the literature by many researches e.g. Nawaz, Enscore and Ham 1983

[2], Framinan et al 2003 [3], and more recently Ruiz and Stutzle 2007 [4]. The previous papers constitute just a small extract of the existing literature, A review of different heuristics to minimize makespan for flowshops can be found in Ruiz and Maroto 2005[5]. Other more general reviews can be found in Framinan, Gupta and Leisten 2004 [6], and in Hejaz and Saghafian 2005 [7].

The **CDS** (Campbell et al) and **NEH** (Nawaz et al) methods are computationally simple Meta-heuristics based on **Simulated Annealing** (SA) and **Genetic algorithm** (GA) requires seed solutions to start with. The CDS and NEH heuristic yield solutions within a short time. In view of this, we undertake the development of heuristics that are better than CDS and NEH. Section (3) describes the proposed heuristic methods and section (3) present two local search methods **Particle Swarm Optimization** (PSO) and GA's (Genetic algorithm's). Section (5) presents the details of the performance analysis of proposed heuristics algorithms and local search methods.

2. CDS and NEH Algorithms [13]

Campbell et al, (1970) proposed am algorithm for makespan problem. Using two main principles this procedure makes good solution:

1. Its uses Johnson's rules in a heuristic way and,
2. Its generally creates several scheduling the best one of which should be chosen.

The CDS algorithm creates m-1 artificial two-machines problems and then solved them by implementing Johnson's two machines algorithm, then the best m-1 obtained solution becomes the best solution.

Nawas et al, (1983) proposed their so called new heuristic on assumption that a job with more total processing time on all machines be given higher priority than jobs with less total processing time on all machines:

1. They proposed initially to arrange jobs in descending order of total processing time.

2. Then NEH heuristic uses the idea of partial sequencing based on this mentioned order.
3. A sequence is constructed by introducing one job at time from that unscheduled order into the partial sequence.
4. If there were k jobs in the previous partial sequence at each job introduction the best partial sequence among k+1 possible partial sequences is chosen.

3. Simple Heuristic Methods (SHM)

In this section, the m-machines (FSSP) F_m // C_{max} (where F_m is the flow shop with m machines and C_{max} is the maximum completion time for scheduling) are considered, where F_m is the m machines flow shop problem with object is to find the order of the jobs on each machine to minimize the maximum completion time C_{max}. Two heuristics are described below.

3.1 The First Heuristic Decreasing Arrange (DR)

<u>Step (1)</u>: The jobs are arranged in non increasing order of the total processing time.

<u>Step (2)</u>: Evaluate C_{max} with respect to $F_m//C_{max}$. Stop.

3.2 Algorithm (SHM)

<u>Step (1)</u>: Create m-1 artificial two-machine problems and then solves them by implementing Johnson's two machines algorithm. Where the processing time for the artificial two machines sub problem for i^{th} job on j^{th} machine at the stage k calculated as:

$$P'_{i1} = \sum_{j=1}^{k} P_{ij} \quad and \quad P'_{i2} = \sum_{j=m-k+1}^{m} P_{ij}$$

<u>Step (2)</u>: Used Johnson's two machines algorithm, then the best m-1 obtained solution becomes the best solution. For the main m-machines makespan problem, let π_s be the best sequence.

<u>Step (3)</u>: Set x=1, $\pi_b=\pi_s$ and M=Makespan given by π_s.

<u>Step (4)</u>: y=1

<u>Step (5)</u>: Generate a sequence π_y which differs from π_b only by having moved job $\pi_b(x)$ to the y^{th} position, and calculate the makespan of the sequence π_y, denote the makespan by C_{max}^{y} and proceed to step (6).

Step (6): Set y=y+1, return to step (5) if y≤n, Otherwise pressed to Step (7).

$$C_{max}^{i} = \min\{C_{max}^{y} \mid y = 1,2,\ldots,n\}, \text{ if } C_{max}^{i} \le C_{max}^{b}$$

Step(7): Determine the sequence π_i such that:

Step (8) : Set x=x+1, return to step(4), if x≤n; otherwise processed to step (9).

Step (9): π_b is the heuristic sequence, and M given the Makespan of π_b. stop.

The following local search methods are used to solve the FSSP.

4. Partial Swarm Optimization (PSO) [8],[9],[10],[11]

One of the important new learning methods is particle swarm optimization (PSO), which is simple in concept, has few parameters to adjust and easy to implement PSO has found applications in a lot of areas.

4.1 Fitness criterion:

One of these stopping criterions is the fitness function value. The fitness value is related by the kind of the objective function, the PSO can be applied to minimize or maximize this function, and in this paper we focused in minimizing the objective function in order to improve the results.

4.2 PSO Algorithm

The PSO algorithm depends in its implementation, in the following two relations [8]:

$$v_{id}^{k} = \omega^{k-1} * v_{id}^{k-1} + c_1 * r_1 * (p_{id} - x_{id}) + c_2 * r_2 * (p_{gd} - x_{id}) \ldots(3.a)$$

$$x_{id}^{k} = x_{id}^{k-1} + v_{id}^{k} \qquad \ldots(3.b)$$

Where c_1 and c_2 are positive constants, r_1 and r_2 are random function in the range [0,1], $X_i=(x_{i1},x_{i2},\ldots,x_{id})$ represents the i^{th} particle, $P_i=(p_{i1},p_{i2},\ldots,p_{id})$ represents the best previous position (the position giving the best fitness value) of the i^{th} particular, the symbol g represents the index of the best particle among all the particles in the population $V_i=(v_{i1},v_{i2},\ldots,v_{id})$ represents the rate of the position change(Velocity) for particle i [8].

The original procedure for implementing PSO is as follow:-

1) Initialize a population of particles with random positions and velocities on dimensions in the problem space.
2) PSO operations include:
   a) For each particle, evaluate the desired optimization fitness function in d variables.
   b) Compare particle's fitness evaluation with its best position p_{best}. If current value is better than p_{best}, then set p_{best} equal to the current value, and P_i equals to the current location X_i.
   c) Identify the particle in the neighborhood with the best success sofar, and assign it index to the variable g.
   d) Change the velocity and position of the particle according to equation (3a) and (3b).
3) Go to step (2) until a criterion, end.

Like the other evolutionary algorithms, a PSO algorithm is population based on search algorithm with random initialization and there is an interaction among population members. Unlike the other evolutionary algorithms, in PSO, each particle flies through the solution space and has the ability to remember its previous best position scurvies from generation to another. The flow chart of PSO algorithm is shown in figure [1].

4.3 The Parameters of PSO [8]

A number of factors will affect the performance of the PSO. These factors are called PSO parameters, these parameters are:

1). Number of particles in the swarm affects the run-time significantly, thus a balance between variety (more particles) and speed (less particles) must be sought.
2). Maximum Velocity (V_{max}) parameter, this parameter limits the maximum jump that particle can make in one step.
3). The role of the inertia weight ω, in equation (3a) is considered critical for the PSO's convergence behavior. The inertia weight

is employed to control the impact of the previous history of velocities on the current one.

4). The parameter c_1 and c_2 in equation (3a) is not critical for PSO convergence. However, proper fine-tuning may result is faster convergence and alleviations of local minima c_1 than a social parameter c_2 but with $c_1+c_2=4$

5). The parameters r_1 and r_2 are used to maintain the diversity of the population, and they are uniformly distributed in the rang [0,1].

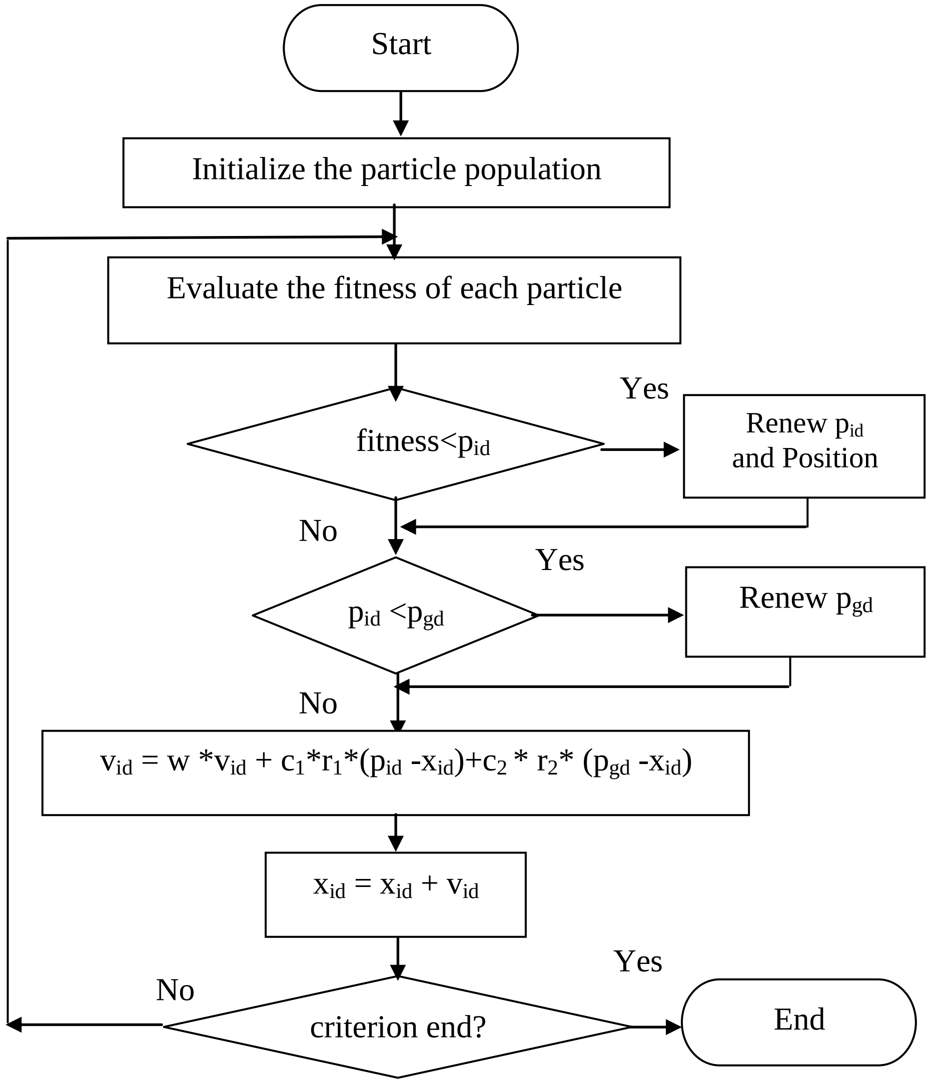

Figure (1) Flowchart of PSO Algorithm [8].

5. Genetic Algorithm (GA)

Genetic algorithms (GA's) are search algorithms based on the mechanics of natural selection and natural genetics. GA is an iterative procedure which maintains a constant size population of candidate solution. During each iteration step (Generation) the structures in the current population are evaluated, and one the basic

of those evaluations, a new population of candidate solution formed.
An abstract view of the GA is [12]:

Generation=0,
Initialize G (p) ;{G=Generation=P=population}.
Evaluate G (p);
While (GA) has not converged or terminated)
    Generation=Generation+1;
    Select G (p) from G(p-1);
    Crossover G (p);
    Evaluate G (p);
    End (while)
Terminate the GA.

6. Implementation of Evolving Methods forFm//Cmax

The NP-completeness of flow shops scheduling problems. So there is a need for local search methods to treat a large size instances problem.Effectively, evolving methods like PSO and GA have demonstrated their ability to solve this type of NP-hard flow shop scheduling problem to find near optimal solution. The applications of these two local search methods will be given.

6.1 Use of GA in Flow Shop Fm//Cmax Problem

1). Genetic operators

- **Selection operator**
  This method uses the roulette wheel selection method. The string with low fitness has a higher probability of contributing one or more offspring to the next generation.
- **Crossover Operator**
  The strength of genetic algorithms arises from the structured information exchange of crossover combinations of highly fit individuals so what we need is a crossover like operator that would exploit important similarities between chromosomes. For that purpose the crossover used in this algorithm is the order crossover (OX) [12], given two parents, builds off spring

by choosing a subsequence of a tour one parent and preserving the relative order of jobs from the other parent.

- **Mutation Operator**

  After the new generation has been determined the chromosomes are subjected to a low rate mutation process. Two mutation operators, are used introduce genetic diversity into the evolving population of permutation.

Note that, the genetic parameters which are used for flow shop problem, from our experience the following parameters are preferred to be used population size pop-size=20 probability of crossover (P_c=0.7), probability of mutation (P_m=0.1), and some hundreds of number of generations.

6.2 Use of PSO in flow Shop Fm//Cmax Problem

For this flow shop scheduling problem from our experience the following parameters are preferred to be used:- Number of particles (N-par=20,30), Maximum velocity (V_{max}=Number of Jobs (J)), Minimum velocity (V_{min}=1), Inertia weight ($\omega \in [0.4,0.9]$). First acceleration parameter ($c_1 \in [0.5,2]$), second acceleration parameter ($c_2=c_1$).Diversity of the population Maintenance (random $r_1,r_2 \in [0,1]$) and some hundreds of generations.

7. Performance Analysis of the Heuristics

We have chosen to compare the performance of the proposed algorithm given above SHM with NEH, CDS, HA and (Des.sort). The proposed heuristic was coded in (CPU, 2.53 GHZ, and 224MB computer).

For small sized problems, the number of job has been varied from 4 to 8, for large sized problems, the number of jobs has been varied from 10 to 500. In all the cases the number of machines has been varied from 3 to 25. The makespans given by the proposed heuristic and by NEH, CDS, HA, and Des.sort have been compared and the results have been tabled.

Table (1) applying (Des.sort, HA, NEH, CDS and SHM) Heuristics (No. of Ex.=20, No. of machines=20, No. of jobs=10)

Ex	Des.sort	HA	NEH	CDS	SHM
1	207	193	207	188	181
2	228	202	219	196	193
3	210	192	210	196	192
4	203	184	202	184	176
5	206	184	199	186	176
6	202	193	200	188	186
7	195	182	195	180	176
8	215	187	201	195	191
9	229	204	222	202	195
10	216	205	210	200	194
11	229	197	225	203	195
12	226	191	220	192	190
13	223	202	220	201	194
14	191	177	187	173	166
15	218	197	211	203	197
16	231	198	216	204	192
17	195	183	198	189	184
18	204	191	200	194	187
19	210	188	207	187	180
20	220	198	214	195	193
No. of best	**0**	**2**	**0**	**0**	**18**

Table (2) applying (Des.sort,HA,NEH,CDS and SHM) Heuristics (No. of Ex.=25, No. of machines=25, No. of jobs=25)

Ex	Des.sort	HA	NEH	CDS	SHM
1	361	316	360	329	310
2	359	322	359	341	320
3	369	332	366	339	319
4	374	330	369	350	328
5	354	320	347	341	320
6	374	330	370	340	320
7	370	322	366	333	315
8	358	320	348	334	319
9	382	340	381	343	331
10	365	337	362	343	326
11	359	323	350	337	325
12	368	316	368	320	306
13	371	329	372	323	318
14	369	322	367	343	318
15	383	330	373	329	317
16	368	327	357	333	319
17	359	324	349	336	323
18	361	323	352	335	318
19	367	328	358	323	318
20	358	319	352	324	303
21	372	337	366	344	321
22	366	326	366	337	318
23	392	335	385	340	326
24	339	310	339	328	306
25	369	331	366	345	331
No. of best	**0**	**1**	**0**	**0**	**24**

Table (3) applying (Des.sort, HA, NEH, CDS and SHM) Heuristics (No. of Ex=25, No. of machines=20, No. of jobs=20)

Ex	Des.sort	HA	NEH	CDS	SHM
1	286	260	284	262	252
2	302	259	299	269	252
3	280	263	280	274	262
4	293	253	289	263	250
5	287	249	273	257	242
6	302	260	292	274	257
7	302	274	291	290	270
8	280	264	277	277	258
9	294	254	292	258	248
10	285	259	283	275	257
11	311	256	304	259	246
12	285	265	281	269	263
13	281	255	272	264	257
14	279	256	279	267	255
15	300	265	295	268	260
16	287	258	257	260	249
17	308	266	306	273	257
18	289	260	284	269	261
19	316	270	311	276	263
20	279	261	278	262	255
21	278	263	278	276	260
23	291	249	283	256	240
24	286	269	281	267	255
25	300	260	295	263	259
26	293	253	284	267	243
No. of. best	**0**	**2**	**0**	**0**	**23**

Note that SHM gives best results from other methods (Des.sort, HA, NEH and CDS), for all above tables.

7.1 Experiment Results of Apply PSO and GA for FSSP

For this problem, a simulation has been constructed in order to apply the PSO and GA. When using the parameters of PSO and GA mentioned above the value of flow shop scheduling prob. Time and number of iterations for the best value of (FSP) results are showed in table (4) and table (5) which are obtained when applying PSO and GA methods respectively for number of Jobs=10, 20,25 with 1000 generations for 10 experiment for each number of Jobs.

Using the following abbreviations

1. J=number of jobs.
2. Values of FSSP.
   - Ex: Experiment number i.
   - Max: maximum value of FSSP of experiment i.
   - OPT: Optimal value of FSSP of experiment i(using complete search).
   - BV: Best value of FSSP of experiment i.
   - ABV: Average of Best value of FSSP for all experiments
3. Values of time
   - CT: completion time of experiment i.
   - BT: Best time of best value of FSSP of experiment i.
   - ABT:Average of best time of FSSP of all experiments.
4. NI: number of iteration of best value of FSSP of experiments.

The average values of FSSP, time and number of iterations for best value of FSSP results are showed in table(4) and table(5) which are obtained when applying PSO and GA methods respectively from chosen number of machine=3,6,10,14 number of job=200,300,500 for 10 experiments for each number of jobs.

Table (4) Applying GA Method on FSP for chosen M=3, 6,10,14, and jobs=200,300,500

M	Job	AUB	MBV	ABV	MBT	ABT	MNI	AVNI
3	200	1259	1080	1130	0	1	1	40
	300	1875	1672	1714	0	2	3	37
	500	2992	2774	2829	0	6	3	42
6	200	1349	1174	1207	0	1	3	45
	300	1948	1753	1780	0	3	9	51
	500	3085	2868	2925	3589	354	2	44
10	200	1433	1254	1283	0	1	13	55
	300	2035	1841	1866	0	3	1	52
	500	3218	3000	3032	0	5	1	36
14	200	1512	1327	1360	0	1	3	47
	300	2127	1930	1947	0	2	4	49
	500	2651	3809	2498	0	2	1	18

AUB=AV.of UB.
MBV=Best value in all example.
ABV=AV.of the Best Value.
ABT=AV. of the best time.
AVNI=AV. of the best iterations.
MNI=Best iteration of best value in all example.
NI=iteration.

Table (5) Applying PSO Method on FSSP for chosen M=3, 6, 10, 14, and Jobs=200,300,500

M	J	AUB	MBV	ABV	MBT	ABT	MNI	AVNI
3	200	1259	1069	1120	0	9	15	445
	300	1875	1663	1704	8	20	180	497
	500	2992	2768	2818	18	60	165	509
6	200	1349	1162	1187	0	8	6	404
	300	1943	1724	1757	5	23	125	540
	500	3035	2833	2898	17	73	148	648
10	200	1433	1235	1262	0	9	258	449
	300	2035	1812	1835	1	19	2	453
	500	3218	2954	3002	9	62	8	542
14	200	1512	1315	1340	1	8	56	394
	300	2127	1910	1922	2	19	39	458
	500	3402	3217	3027	10	72	64	560

AUB=AV. of UB.
MBV=Best value in all example.
ABV=AV.of the Best Value.
ABT=AV.of the best time.
AVNI=AV. of the best iterations.
MNI=Best iteration of best value in all example.
NI=iteration.

Table (6) Comparative results for PSO, GA and UB Methods on FSSP for m=3,6,10, and Jobs=200,300,500, No. of Ex.=10.

Ex	No. of machines	UB	GA	PSO
1	3	1128	1080	1068
2		1155	1137	1134
3		1173	1150	1146
4		1175	1151	1143
5		1147	1107	1098
6		1148	1122	1107
7		1153	1139	1131
8		1170	1154	1140
9		1198	1138	1123
10		1139	1118	1106
1	6	1805	1755	1724
2		1835	1754	1726
3		1833	1798	1786
4		1873	1801	1774
5		1867	1783	1762
6		1865	1791	1764
7		1859	1753	1744
8		1861	1817	1794
9		1838	1781	1747
10		1796	1766	1748
1	10	3137	3000	2954
2		3176	3027	3011
3		3097	3035	3016
4		3103	3025	2974
5		3128	3034	3009
6		3130	3011	2986
7		3144	3042	3011
8		3110	3055	3027
9		3064	3057	3031
10		3090	3033	3000

8. Conclusions

1. This paper proposes a new heuristic algorithm (SHM) to address the problem of scheduling in the flow shop.
2. The proposed heuristic has been evaluated and compared with Des.sort, HA, NEH and CDS using a large number of randomly generated problems with different number of jobs and number of machines.
3. The proposed heuristic has been found to yield solutions better than that by the existing heuristic for a vast majority of the problem with respect to makespan minimization. The computation times of Des.sort, HA, NEH, CDS and SHM for different number of jobs and machines been evaluated.
4. From table (6), we notice that the achievement of PSO in FSSP is better than GA.

References

[1]. Johnson, S.M., "***Optimal Two and Three Stage Production Schedules with Setup Time Included***", Naval Research Logistic Quarterly 1(1), 61-68,1954.

[2]. Nawaz M., Enscore, Jr., E.E. and Ham, I.,"***A Heuristic Algorithm for the m Machine n Job Flow Shop Sequencing Problem***" Omega-international Journal of Management Science 11(1), 91-95, 1983.

[3]. Framinan J.M. et al., "***Different Initial Sequence for The Heuristic of Nawaz Enscore and Ham To Minimize Makespan, Idletime or Flow Time in The Static Permutation Flow Shop Sequencing Problem***", International Journal of Production Research 11(1), 121-148, 2003.

[4]. Ruiz R. and Stutzle T., "***A Simple and Effective Iterates Greedy Algorithm for the Permutation Flow Shop Scheduling Problem***", European Journal of Operational and Research 177(3), 2033-2049, 2007.

[5]. Ruiz R. and Marato, "***A Comprehensive Review and Evaluation of Permutation Flow Shop Heuristics***", European Journal of operational Research, 162(2), 479-494, 2005.

[6]. Framinan J.M., et al., "***A Review and Classification of Heuristics for Permutation Flow Shop Scheduling with Makespan Objective***" Journal of the Operational Research Society 55(12), 1243-1255.2004.

[7]. Hejazi S.R. and Saghafian S., "***Flow Shop Scheduling Problems with Makespan Criterion A Review***", International Journal of Production Research 43(14), 2895-2929, 2005.

[8]. Kennedy J. and Eberhart R. C., "***Particle Swarm Optimization***", Proceeding of IEEE International Conference of NN, piseataway, 942-1948,1995.

[9]. Parsopoulos, K. E. and Vrahatic, M.N., "***Recent Approaches to Global Optimization Problems through Particle Swarm Optimization***", kuwer Academic Publishers, Netherlands Nortural computing, 1,235-306, 2008.

[10]. Riberio, P.F. and kyle W.S., "***A Hybrid Particle Swarm and Neural Network Approach for Reactive Power Control***", Member, 2003.

[11]. Shiy, "***Particle Swarm Optimization***", Electronic Data System Inc. Kokomo, IN 46902,USA Feature Artide IEEE. Neural Network, Society, February, 2004.

[12]. Whitley D., "***An Executable Model of A Simple Genetic Algorithm***", Ind Whilley (Ed.) Foundations of Genetic Algorithms II, PP.45-62, 2007.

[13]. Roopa, H. et al, "Heuristic for scheduling in flow shops to Minimize Makespan", Recent Developments in Operation Research, Narosa, Publishing House, New Delhi, India, 2000.

تطوير طرائق تقريبية لمسألة الجدولة الانسيابية لتصغير اكبر وقت إتمام

طارق صالح عبد الرزاق، حنان علي جيجان، فائز حسن علي

الجامعة المستنصرية / كلية العلوم/ قسم الرياضيات

المستخلص

سبق وان تم دراسة التعقيدات الحسابية لمسائل الجدولة الانسيابية لعدة سنوات. ولذلك اقترحت عدة طرائق تقريبية للحصول على حلول ذات جودة عالية وبأقل جهد حسابي. الطريقتان التقريبيتان CDS و NEH هي طريقتان كفؤتان من بين الطرائق التقريبية المطورة مثل طريقة أمثلية السرب الجزيئي (PSO) والخوارزمية الجينية (GA).

هدف البحث هو مناقشة بعض الطرائق، واقتراح تطويرات لهذه الطرائق المختصة بتصغير اكبر وقت إتمام لمسائل الجدولة الانسيابية.

الهدف الأساسي في هذا البحث، من جانب، فقد تم تطوير طرائق تقريبية كفوءة والتي ستكون أفضل من الطرائق التقريبية الموجودة حاليا والتي تعطي حلولا وبأقل وقت مثل طريقة الطريقة التقريبية المبسطة (SHM) وطريقة الترتيب التنازلي التقريبية (DR). ومن جانب آخر، فقد طبقنا طريقتين من الطرق البحث المحلية مثل طريقة (PSO) و (GA).

وقد تم تقديم التجارب التحليلية الكفؤة للطرق التقريبية المقترحة وطرق البحث المحلية بالمقارنة مع الطرائق الكفؤة الموجودة.