e-ISSN:2525-3654

JPBReview

International Journal of Professional Business Review

Responsible Editor: Maria Dolores Sánchez-Fernández, Ph.D.
Associate Editor: Manuel Portugal Ferreira, Ph.D.
Evaluation Process: Double Blind Review pelo SEER/OJS

UTILIZING MODEL KNOWLEDGE FOR DESIGN DEVELOPED GENETIC ALGORITHM TO SOLVING $FJS|r_j|F_{Max}$ PROBLEM

UTILIZANDO O MODELO DE CONHECIMENTO PARA PROJETAR O DESENVOLVIMENTO DO ALGORITMO GENÉTICO PARA SOLUCIONAR O PROBLEMA $FJS|r_j|F_{Max}$

Hamidreza Salmani Mojaveri [1]

[1]Department of technology management; Islamic Azad University – Islamic Republic of Iran
E-mail: mazisalmani@yahoo.com

ABSTRACT

One of the discussed topics in scheduling problems is Dynamic Flexible Job Shop with Parallel Machines (FDJSPM). Surveys show that this problem because of its concave and nonlinear nature usually has several local optimums. Some of the scheduling problems researchers think that genetic algorithms (GA) are appropriate approach to solve optimization problems of this kind. But researches show that one of the disadvantages of classical genetic algorithms is premature convergence and the probability of trap into the local optimum. Considering these facts, in present research, represented a developed genetic algorithm that its controlling parameters change during algorithm implementation and optimization process. This approach decreases the probability of premature convergence and trap into the local optimum. The several experiments were done show that the priority of proposed procedure of solving in field of the quality of obtained solution and convergence speed toward other present procedure.

Keywords: Optimization, Parallel Machines, Genetic Algorithm, Dynamic and Adjustment of controlling parameters

RESUMO

Um dos tópicos discutidos na programação de problemas é a Flexibilização Dinâmica da Produção com Máquinas Paralelas (FDPMP). Pesquisas mostram que este problema, por conta de sua natureza côncava e não-linear, usualmente possui vários locais ideais. Alguns dos pesquisadores de programação de problemas pensam que a Genética dos Algoritmos (AG) são abordagens apropriadas para resolver os problemas de otimização desse tipo. Mas pesquisadores mostram que uma das desvantagens do Algoritmo Genético clássico é a convergência prematura e a probabilidade de armadilha dentro do local ideal. Considerando estes fatos, a presente pesquisa, representa um algoritmo genético desenvolvido em que seus parâmetros de controle mudem durante a implementação e otimização do processo. Esta abordagem reduz a probabilidade de convergência prematura e de armadilhas dentro de um local ideal. A maior parte dos experimentos realizados, mostram que a prioridade do procedimento proposto de solucionar no campo da qualidade da obtenção de solução e aceleração de convergência em direção da presença de outro procedimento.

Palavras-chave: Otimização, Máquinas Paralelas, Algoritmo Genético, Dinâmica, Ajuste dos Parâmetros de Controle

Received on January 30, 2018
Approved on April 22, 2018

How to Cite (APA)

Mojaveri, H., S., (2018). Utilizing model knowledge for design developed genetic algorithm to solving FJS r_j F_{max} problem. *International Journal of Professional Business Review*, 3 (2), 172–186. http://dx.doi.org/10.26668/businessreview/2018.v3i2.49

INTRODUCTION

In this paper, we have studied more efficient procedure to solving scheduling problems of Flexible Dynamic Job-Shop with Parallel Machines (FDJSPM). FDJSPM problem is an optimization problem in discontinuous space. This problem because of concave and nonlinear nature usually has several local optimums [1]. Regarding reviewing literature, we show that to solving scheduling problem mainly used over innovative procedure. From among presented procedure, genetic algorithm has been better performance toward other procedure. Some researchers of scheduling problems believe that this algorithm is appropriate approach to solving optimization problems of this kind [2]. Surveys show that one of the disadvantages of classic genetic algorithms is property of procedure convergence [3]. Genetic algorithm after some repetition converges towards local optimum solution or close solution to optimum.

But there are somewhat considerable variations in how we can utilize the operators, type of operators, selection mechanism, and way of making primitive population and how parents can select for doing operators that can open researcher's hand completely for defining more effective genetic algorithms. With considering of these facts that FDJSPM problem usually has several local optimum and also definition of genetic algorithm in premature convergence and trap into the local optimum. In present research, we presented a developed genetic algorithm that its controlling parameters dynamically change during algorithm implementation and optimization process. This approach leads to probability decrease of premature convergence and trap into the local optimization. Several experiments show that priority of proposed procedure in field of quality of solutions and convergence speed. In continuation, we first investigate literature in field of using genetic algorithm to solving research problem and section 3 deals with definition of FDJSPM problem and representation of mathematics model. Section 4 proposed solving procedure based on genetic algorithm, operators and way of dynamic adjustment of its controlling parameters are analyzed. Section 5 represents the performance of test proposed algorithm performance with two present genetic algorithms in literature.

Literature review

The Scheduling of Flexible Dynamic Job-Shop is most important subjects of production manufacturing and is part of most difficult synthetic optimization. Overall, researches have studied about scheduling in dynamic environments divided into two main classifications. First classification is based on line theory and second classification is based on Rolling Time Horizon technique [4, 5]. Regarding the above complexity problem of analytic methods solve most of problems just with single machine. To solve problems with more than single machine mainly has used Meta heuristic procedures. From among presented procedures, genetic algorithm has better performance toward other procedures and some researchers think genetic algorithm is appropriate approach to solving optimization problems of this kind [2]. In continuation, we deal with interviewing of genetic algorithm literature to solving research problems.

In 1997, Gen and Cheng showed that in genetic algorithm the large size of primitive population, the number of generation and crossover rate can ended up extension search space as a result of more quick convergence of algorithm [6]. They also addressed that because of mutation operator is a marginal operator, so it is better we prevent from making an approach only accidental with reduction. In 1999, Brandimart solved the problem of flexible process program and multiple objectives with an exact optimization algorithm [7]. He suggested that two procedures for it, one of them based on Dual theory and the other based on genetic algorithm. In this year, Ghedjati offered a synthetic approach of Meta heuristic procedures based on genetic algorithm to solving FJS problem [8].

He advised about impact strategy of limitations and unjustified chromosomes to solve problems that have no cave solution space and said it is better we use a procedure based on rejection unjustified chromosomes. In 2002, Kacem and et al. was investigated FJS problem in multiple objectives for the first time [9]. Their presented procedure is compound of fuzzy logic and evolutionary algorithm that use fuzzy logic for searching in targets space to access Prato optimum solutions. At this time, Lee and et al. presented a genetic algorithm to solve similar problem with FJS in supply chain [10].

In 2004, Tay and Wibow used a specific representation for them proposed genetic algorithm to solving flexible job-shop scheduling [1]. They introduced scheduling problems of this kind due to their concave and nonlinear nature to part of hard problems from the one way and the other way a part of hard problems that are usually different local optimum. In this year, Gen and Cheng in collaboration with Kacem and et al. presented a genetic algorithm to solving them problem [11]. In this research, first sub problem of assigning solve through SPT prioritize rule and then based on that select an appropriate representation of chromosome is most important stage in obtain good quality solution and a genetic algorithm with representation of chromosome present based on operation-basis to solving problem.

The results of calculations show the efficiency of algorithm impact to large problems. At this year, Kurz and Askin to solving FSPM problem present over innovative RKGA problem [12,13]. They used heuristic approach for solve their problem and analyzed their problem to two sub problem operation sequence of assigning and determining. In 2007, Tay and Ho used genetic programming to solving the problem of their flexible job shop scheduling [2]. In this year, Ho et al. for solving flexible job-shop with suppose of secondary jobs rotation, over dependence evolutionary algorithms to secondary synthetic mechanisms and they also introduced accidental selection as most limitation of this procedure [14]. In order to overcome to this limitation and get efficient solutions for FJSP problems they try to use cooperation between learning and evolutionary.

Thus, they suggested their proposed genetic algorithm under the heading LEGA. They considered the compound of three macula algorithm, learning pattern and production of population simultaneously. At this approach for macula population production has used simple distribution rule and for macula learning pattern has used *K-nearest* procedure. Coline in use of genetic operators, advised that use of two or some section point of the operators have priority towards other operators [15]. Dagli and Sittisathanchai introduced one of the main defects of classic genetic algorithms is premature convergence and traps them into the local optimum [16]. In 2007, Gao and et al. use the compound of genetic algorithm with innovative procedure transfer of channel in order to solve flexible job-shop scheduling [17]. They ask for help from Gen and et al. procedure for representation their chromosome of genetic algorithm. That of course, they used two vectors to assigning machine and sequence of operation instead of a vector of a gene procedure.

They also ask for help capability of nation wide searching of genetic algorithm and also capability of local searching innovative algorithm of transfer of bottleneck to solving their problem. In 2009, Nahavandi and Abbasian presented a simple genetic algorithm with two dimensioned chromosomes to solving FDJSP problem and they also showed their superiority of procedure towards similar procedure in literature [18, 19].

At this year, Amiri and et al. used a compound plan for stimulating behavior of chromosomes and adjustment their genetic operators [20]. They performance each treatment five times and considered their average as solutions and with SAS software determined the relationship between depended varieties and solutions and then with use of simultaneous gradient procedure and regarding limitations of their problem determined the number of their proposed genetic algorithm operator. At this year, Morino and et al. suggested that a binary representation of GEP chromosomes for justified responses [21]. They showed that this kind of representation improved GEP scaling considerably. In 2010, Verama and et al. suggested a procedure called technique of data calculation in their genetic algorithm and also they told this technique in scaling estimation the distribution of genetic algorithms have basic role [22].

Overall, with comprehensive study about history of research in field of the procedure of solution for FDJSPM and specially the use of Genetic Algorithm procedure to solve the problems of this kind were done, and it sis indentified that first the scheduling problems of this kind due to their concave and nonlinear nature usually have different local optimum and second one of the main defects of classic genetic algorithms are premature convergence and trap into the local optimum [1, 3].

There for, after investigation of subject literature and identify present gaps in proposed genetic algorithm for problem solution research and in order to overcome limitation of premature convergence and trap into the local optimum used of two strategies. First strategy try to increase the diversity of search algorithm, that is used a classic mutation (Mutation1) and innovative mutation (Mutation2). In next strategy regarding that over optimization process, role and rate importance of each genetic operator, we tried to search a proposed genetic algorithm intelligently. Therefore in proposed genetic algorithms probability of each operators during optimization process transfer dynamically and based on the number of elite chromosomes to next generation population directly. In continuation, we deal with the investigation mathematic model of problem with minimization function Fmax (as scheduling target function).

The presentation of mathematics model

In this section, first variation and sets is defined and then mathematics model for FDJSPM problem will be presented.

Definition of variation and sets

Variation and sets is defined below:

The number of jobs	:	n
The number of operation for ith job	:	n_i

Arrival time for i[th] job	:	r_i
The number of work station	:	m
$k=1,\dots,m$ Operation index	:	k
k[th] Process stage	:	M_k
r[th] parallel machine of k[th] stage	:	$M_{k,r}$
i[th] operation of j[th] job	:	$o_{i,j}$
The stage that $o_{i,j}$ will be process	:	$St_{i,j}$
The number of parallel machines in k[th] stage	:	l_k
Max (l_k)	:	Lmax
$pm=1,\dots,l_k$ Parallel machines index in per stage	:	pm
Speed for $M_{k,r}$ machine	:	$S_{k,r}$
Processing time in unique speed machine	:	$P_{i,j}$
Maximum flow time	:	F_{Max}
Completion time for $o_{i,j}$	:	$c_{i,j}$
Access time for $M_{k,r}$ machine for processing	:	$avail_{k,r}$
The number of operation for i[Th] job	:	$Ft_{k,pm}^{(i,j)}$
If k[th] machines is alternative machines for $o_{i,j}$ is equal 1 and otherwise is equal 0	:	$a_k^{(i,j)}=\begin{cases}1\\0\end{cases}$
If $o_{i,j}$ process under $M_{k,pm}$ is equal 1 and otherwise is equal 0	:	$X_{k,pm}^{(i,j)}=\begin{cases}1\\0\end{cases}$
If $o_{a,b}$ under $M_{k,pm}$ process faster than $o_{i,j}$ is equal 1 and otherwise is equal 0	:	$R_{k,pm}^{(i,j)(p,q)}=\begin{cases}1\\0\end{cases}$

The presentation of mathematics model

Mathematics model for FDJSPM problem is presented below:

(1) $\forall j: 1\le i\le n; \forall i: 1\le j\le (n_i-1);$ $\forall k,k': 1\le k,k'\le m;$ $\forall pm,pm': 1\le pm,pm'\le l_{k'};$

$$Ft_{k,pm}^{(i,j+1)}-Ft_{k',pm'}^{(i,j)}+L\times(1-a_k^{(i,j+1)}\times X_{k,pm}^{(i,j+1)})\ge P_{i,j+1,k}/S_{k,pm}$$

(2) $\forall i,q: i=1,\dots,n-1\ \&\ q=i+1,\dots,n$ $\forall i: 1\le j\le n_j; \forall p: 1\le p\le n_q;$

$$Ft_{k,pm}^{(i,j)}-Ft_{k,pm}^{(p,q)}\times X_{k,pm}^{(i,j)}+L\times R_{k,pm}^{(i,j)(p,q)}\ge X_{k,pm}^{(i,j)}\times P_{i,j,k}/S_{k,pm}$$

(3) $\forall k: 1\le k\le m;$ $\forall pm: 1\le pm\le l_k;$

$$Ft_{k,pm}^{(p,q)}-Ft_{k,pm}^{(i,j)}\times X_{k,pm}^{(p,q)}+L\times(1-R_{k,pm}^{(i,j)(p,q)})\ge X_{k,pm}^{(p,q)}\times P_{p,q,k}/S_{k,pm}$$

(4) $\forall j: 1\le i\le n;$ $\forall i: 1\le j\le n_j;$

$$\sum_{k=1}^{m}\sum_{pm=1}^{l_k}a_k^{(i,j)}\times X_{k,pm}^{(i,j)}=1$$

(5) $$\forall j:1\le i\le n;\forall i:1\le j\le n_j;\quad \forall k:1\le k\le m;\quad \forall pm:1\le pm\le l_k; \qquad X_{k,pm}^{(i,j)}\le a_k^{(i,j)}$$

(6) $$\forall j:1\le i\le n;\forall i:1\le j\le n_j;\quad \forall k:1\le k\le m;\quad \forall pm:1\le pm\le l_k; \qquad Ft_{k,pm}^{(i,j)}\le L\times X_{k,pm}^{(i,j)}$$

(7) $$\forall j:1\le i\le n; \qquad C_i \ge \sum_{k=1}^{m}\sum_{pm=1}^{l_k} Ft_{k,pm}^{(i,n_j)}$$

(8) $$\forall j:1\le i\le n;\quad \forall k:1\le k\le m;\quad \forall pm:1\le pm\le l_k; \qquad Ft_{k,pm}^{(i,1)}\ge X_{k,pm}^{(i,1)}\times P_{i,1,k}/S_{k,pm}+r_i$$

In next section we introduced the structure of proposed genetic algorithm and after adjustment of parameters and its efficiency to solving problem based on target function of minimization maximum time in jobs rotation through numerical experiments that will be shown.

PROPOSED GENETIC ALGORITHM

Designing genetic algorithm to solving FDJSPM problem

A series of steps in proposed genetic algorithm are:

```
Procedure: GA2
Begin
    t 0;
    initialize P(t);
            evaluate P(t);
            while (not termination condition) do
                    recombine P(t) to yield C(t);
                    evaluate C(t);
                    select P(t+1) from P(t) and C(t);
                    t t+1;
              end
            end.
```

Exhibition of chromosomes (problem coding)

The first stage in solving optimization problems with genetic algorithm is representation of problem solutions as a chromosome [23]. In designing of chromosomes usually considering criteria such as the minimum need to space and time are important in complexity calculation of problem and avoidance of making unjustified chromosomes [24]. FDJSPM problem analyzed to assigning sub problem and determining sequence of operation.

Proposed genetic algorithm has designed in such away that it can be integrated and synchronic solved both of above-mentioned sub problem. Therefore, it used two dimensional chromosomes. At this representation, the length of chromosome is the number of all operation of present works for scheduled and its width is 3. Thus, each problem solution is shown in a two dimensional array way. This procedure is similar to procedure (Lee and et al. (2002)) that of course due to present parallel machines in each station [10]. The string of assigning considered in two separate strings way and the string of assigning job station and also the string of assigning machines.

In proposed genetic algorithm, first string of chromosome show that job station. Second string shows that the number of machine and third string of this representation also show that assignment priority to each operation. Each element of third string is number between one and the number of all operation. Therefore, regarding this representation procedure, the problem solution will be justified. So, at this designing we will never face to unjustified chromosome.

Primitive population

Accidental production of primitive population causes to maintenance diversity of chromosomes in production and the probability of decrease premature convergence and trap into the local optimum [10]. So, we in proposed genetic algorithm in order to avoidance of premature convergence don't have use any innovative procedure to produce primitive population.

Crossover operator

At proposed genetic algorithm utilized two crossover operators place of axis and RMX operator in compound way and Pc rate. Crossover operator of place of axis be implemented the string of assigning and crossover operator RMX on the string of operation sequence.

Mutation operator

Overall, mutation operator don't have programmed with making accidental changes and prepare searching probability of more vast sections of solution space, and prevent of premature convergence algorithm. That is, in proposed algorithm after crossover operator implement two parents string and then mutation operator implement to this two string separately and while probability of test has been successes the above-mentioned chromosome is mutated. In proposed genetic algorithm because of specific structure of problem and proposed chromosome put innovative mutation operator based on linear reversal operator and innovative rule in such a way

That this operator works with two different rates and it is used presented innovative rule of mutation operator for assigning of sub problem with mutation rate (Pm2) and for sub problem of determination of operation sequence used reversal operator with mutation rate (Pm2).

Selection approach

In proposed genetic algorithm is used developed sampling space. Sampling mechanism also for making next compound generations will be compounding of selection of elite procedure and also roller.

Priority function

In studied problem, the aim of minimization is maximum flow time of jobs rotation of works, so it will be assigned to each selected solution with Fmax and low number of priority. Here, priority function for each chromosome is defined $F(i)=1/Fmax(i)$, where $F(i)$, the number priority of chromosome is i th.

Criterion of stop

The algorithm is stopped after reach to max-gen reputation.

Adjustment of parameters dynamically

One of the classic genetic algorithms is property of premature convergence. In genetic algorithm, two classic genetic operators compete in problem of convergence way [3]. While use of mutation operator make diversity in population, crossover operator was forced population become convergent [11]. Considering this fact, in adjustment of parameters of genetic algorithm are always try to find optimum adjustment for the probability application of mutation operators and crossover (with use of procedures such as simulation techniques). Also, determination and use of stable numbers in order to the probability of doing mutation operators and crossover in genetic algorithm implementation may make difficult in final generation of algorithm and caused to premature convergence of algorithm.

In other to improvement genetic algorithm and avoidance of premature convergence, we can use " changing of exchange rate and mutation during implementation of a genetic algorithm "at this procedure after production of each new generation with use of statistical data , we can estimate priority of space of members of population towards priority is its best member. While large number of population members has priorities next to priority of best member. At these conditions, premature convergence also may be happened. There for it seems that probity decrease of crossover operator and increase of mutation can emit algorithm from the local optimum. In retune, if the large number of chromosomes quality is far quality of best person in improvement of solutions. Due to few numbers of chromosomes have good quality (high priority) and present crossover rate don't have caused good solution effectively. To resolve this problem we can decrease mutation rate or increase crossover rate.

Thus v, based on priority space of strings the rate of crossover operator and mutation was adjusted during problem solution. In first proposed algorithm during use of selection of elite technique , if the number of best chromosome of ordered population are similar ,there is the probability of trap into the local optimum. Thus, it is possible that any progression about similarities of present shambles of gene convergence. So, for avoidance of premature (because of increasing crossover rating) and over diversity (because of increasing mutation rate) crossover rates and mutation are changed dynamically. At this position, designing of GA was tried that the number of mutation are twice and the number of crossover become half only once. If this condition is broken, the number of mutation and crossover changed the same primitive numbers.

Designing of numerical experiments

Comparison procedures

At this stage, we considered RKGA algorithm - that Kurz and Askin (2004) presented for FSPM problem- and Classic Genetic Algorithm (CGA) - that presented by Nahavandi and Abbasian (2009) to solving FDJSPM problem- the basis of investigation efficiency of proposed algorithm. Nahavandi and Abbasian put their proposed classic genetic algorithm in comparison with present most similar procedure in scheduling literature for FDJSPM problem (RKGA algorithm which was presented for FSPM problem by Kurz and Askin (2004)) [18, 19]. In this research, we tried to compare our efficiency of proposed algorithm with both above procedures. RKGA algorithm to soling FSPM problem are designed and presented regarding flexibility of parallel machines with similar speeds and also flexibility of operation between work stations. That is, solved problem by this algorithm has most similarity with FDJSPM problem and easily and without lose generalization algorithm, is useful for this problem.

Nahavandi and Abbasian also in their comparison have considered JSPM problem which is specific state of FDJSPM problem. We also have considered in comparison specific state of FDJSPM problem (JSPM problem). Nahavandi and Abbasian to solving their problems use hierarchical approach, they divided their problems to two sub problem (Assignment and Sequence sub problem) and then divided their Assignment sub problem to two sub problem of assigning station and machine.

Nahavandi and Abbasian (2009) and Kurz and Askin (2004) used RKGA approach for scheduling first works in their research and for assigning works to machine in next stages based on SPTCH (rotation SPT) and Jonsen rule, in fact, at this algorithm first operation of works schedule by RKGA and for scheduling next operations of works used above-mentioned heuristics [13, 18, 19]. We used the concept of equal for FDJSPM problem, it means we schedule first operation of works by RKGA and next operation with use of above-mentioned heuristics [18].

Production of accidental problems

In order to production of accidental problems, first we identify six parameters according to 1 table. So that, we used Kurz and Askin (2004) for first five parameters and Nahavandi and Abbasian (2009) for processing speed of similar machines with considering U[1,3] distribution for them. Overall, all scenario of this level will be tested. The factor of machine distribution as variable form need at least at one stage exist some different machines towards other stages and also most number of machines should fewer than the number of works and at least at one stage the number of parallel machines should be bigger than 1.

States	*Values*				:	*Parameter*
3	*30-100*		*6*		:	*Number of Jobs*
---	*Variable*	*Constant*	*Variable*	*Constant*	:	*Distribution of Machines*
4	*U[1,4] - U[1, 10]*	*2-10*	*U[1,4] - U[1, 6]*	*2-6*	:	*Number of Machines*
3	*2 - 4 - 8*				:	*Number of Operations*
2	*U[50,70] - U[20, 100]*				:	*Processing Times*
1	*U[1,3]*				:	*Speed of Machines*
72					:	***Number of Scenarios***

Table 1
The level of factors for GA implementation

There are 72 experimental scenarios here that for each of them produced 10 data collection.

Designing the procedure of doing experiments

In continuation, in order to comparison, proposed algorithm, CGA algorithm (Nahavandi and Abbasian) and RKGA algorithm (Kurz and Askin) are coded in C language and with a computer (CPU 3GHz, 1GB of RAM) Pentium IV in Borland C++ 5.02 environment implement. Each of the three algorithms implemented with the same 720 produced data collection.

Adjustment of parameters

The different levels of parameters have influence on the quality of obtained solutions of genetic algorithm and different sets of parameters presented different solutions of algorithm. We utilized only the adjustment of parameters approach for adjustment 4 parameters (μ'), (p_c), (p_{m1}), (p_{m2}). Considered different levels for parameters are follows:

%5, %10, %15, %20, %25 & %30	=	(μ')
0.3, 0.4, 0.5, 0.6, 0.7, 0.8 & 0.9	=	(p_c)
0.005, 0.01, 0.015, 0.02, 0.025 & 0.03	=	(p_{m1}) & (p_{m2})
100	=	(pop_size)
125	=	$(\max\_gen)$

Gráfico Nº 24
Solsticios, equinoccios incaicos para indicar el calendario agrícola. Tomado de: https://www.google.es/

The result for adjustment of parameters based numerical experiments consequently are: %20, 0.7, 0.02 & 0.025 for 4 above parameters.

The result of experiments

At this stage, efficiency of proposed genetic algorithm (GA2) with Classic Genetic Algorithm (CGA) -proposed by Nahavandi and Abbasian (2009) - and RKGA algorithm -proposed by Kurz and Askin (2004) - to solving 216 different scenario of problem and each of them 10 times compared with according to table1. To comparison of this there procedures from a main indicator of "The average of target function" indicator, "The average of solving time each of Scenario" and also "Degree and time number of improvement in target function" are used in different scales of tables (2), (3) are follows:

1) In other to 10 implementation of each of 72 data collection "The average of target function" seem to best criteria for estimation algorithms. As we considered in Tables (2), (3), GA2 algorithms based on this indicator in different scales of problem (small, medium & large scale) on Average of improvement %1.99, %1.77 & %2.42, respectively, and generally an average of improvement %2.06 have more performance towards RKGA. Also, GA2 algorithm based on this indicator in different scales of problem an average of improvement %0.82, %0.43 & %0.95, respectively and generally and average of %0.73 have better performance towards CGA.

2) GA2 algorithms based on best result "lowest Fmax" indicator for 10 implementation of each of 72 Scenario of problem an average of improvement %1.11, %1.31 & %2.02, respectively and generally on average of improvement %1.48 have noticeable priority toward RKGA. Also, GA2 algorithm based on this indicator in different scales of problem an average of improvement %0.53, %0.29 & %0.84, respectively and generally and average of %0.56 have better performance towards CGA.

Problem Scales		Small				Medium				Large				
Number of Jobs		6				30				100				
Number of Operations		2	4	8	Average	2	4	8	Average	2	4	8	Average	
	Lowest *Fmax*	40.36	73.93	172.6	95.62	91.85	122.6	204	139.47	283.9	307.1	362.7	317.89	184.33
	Average *Fmax*	121.09	184.65	302.83	202.86	527.43	646.83	845.23	673.16	1743.80	1905.88	2612.90	2087.53	987.85

	The average of time solution (S)	0.025	0.028	0.035	0.029	0.201	0.248	0.233	0.227	1.602	1.551	1.745	1.633	0.63
	Lowest *Fmax*	39.34	74.12	170.73	94.73	92.86	118.29	200.82	137.32	279.21	306.02	347.98	311.07	181.04
	Average *Fmax*	118.22	182.33	295.74	198.76	524.52	636.02	819.11	659.88	1736.59	1834.89	2531.11	2034.20	964.28
	The average of time solution (S)	0.031	0.034	0.054	0.040	0.502	1.151	4.213	1.955	1.117	7.124	35.781	14.67	5.56
	Lowest *Fmax*	2.53%	-0.26%	1.06%	1.11%	-1.10%	3.50%	1.54%	1.31%	1.64%	0.36%	4.06%	2.02%	1.48%
	Average *Fmax*	2.37%	1.26%	2.34%	1.99%	0.55%	1.67%	3.09%	1.77%	0.41%	3.72%	3.13%	2.42%	2.06%
Improvement of solution Time (S) toward RKGA		-0.24	-0.24	-0.21	-0.54	-0.3	-1.50	-3.64	-17.08	-7.4	0.30	-3.59	-19.50	-7.6
	Lowest *Fmax* *	58	29	63	50.0	23	66	51	46.7	62	41	53	52.0	49.6
	Average *Fmax* **	15	7	18	13.3	10	19	13	14.0	18	15	16	16.3	14.6

* Lowest number F_{max}, from among 72 implementation of scenarios that based on the number of jab factors are separated and chosen and it isn't including equal RKGA

** Lowest number F_{max} from among 21 implementation of scenario that based on the number of job factors are separated and chosen and it is not including equal RKGA.

*** Negative number of Table shows that priority of answer from RKGA toward proposed solution procedure (GA2).

Table 2
The result of experiments for GA2 and RKGA (Kurz and Askin, 2004)

3) Also proposed procedure regarding "Improvement of target function" indicator un-different dimension of an average impartment 50, 46.7, 52, times respectively, generally on average of improvement 49.6 time are more efficient than RKGA, and also have 42, 38.7, 46.3 times improvement respectively generally on average of 42.3 times improvement towards CGA have priority.

4) Considering "The average of solution time" in small r proposed procedure in small dimensions don't have much different with RKGA. But in average and big on average of 7.4, 7.6 second are increased respectively and generally on average

of 5.1 second is increased has undesired per furnace towards RKGA. But as you see in table (3), proposed genetic algorithm based on this indicator in small average and big dimensions an average of 0.4, 0.4, 0.7 second is decreased and generally on average of 0.5 second is increased has undesired per furnace towards GA classic (CGA).

Problem Scales		Small				Medium				Large				
Number of Jobs		6				30				100				
Number of Operations		2	4	8	Average	2	4	8	Average	2	4	8	Average	
	Lowest *Fmax*	39.74	74.95	169.84	94.85	93.35	118.01	202.02	137.79	278.12	307.98	356.29	314.13	182.26
	Average *Fmax*	119.66	184.71	295.67	200.01	527.69	635.21	825.87	662.92	1731.32	1870.89	2562.59	2054.93	972.62
	The average of time solution (S)	0.035	0.049	0.146	0.077	0.735	2.152	7.982	3.623	2.653	18.745	158.237	59.88	21.19
	Lowest *Fmax*	39.34	74.12	170.73	94.73	92.86	118.29	200.82	137.32	279.21	306.02	347.98	311.07	181.04
	Average *Fmax*	118.22	182.33	295.74	198.76	524.52	636.02	819.11	659.88	1736.59	1834.89	2531.11	2034.20	964.28
	The average of time solution (S)	0.031	0.034	0.054	0.040	0.502	1.151	4.213	1.955	1.117	7.124	35.781	14.67	5.56
	Lowest *Fmax*	1.01%	1.11%	-0.51%	0.53%	0.52%	-0.24%	0.59%	0.29%	-0.39%	0.64%	2.33%	0.86%	0.56%
	Average *Fmax*	1.21%	1.29%	-0.02%	0.82%	0.60%	-0.13%	0.82%	0.43%	-0.30%	1.92%	1.23%	0.95%	0.73%
Improvement of solution Time (S) toward CGA			0.11	0.31	0.63	0.4	0.32	0.47	0.47	0.4	0.58	0.62	0.77	0.7
	Lowest *Fmax**	45	51	30	42.0	42	28	46	38.7	33	44	62	46.3	42.3

	Average *Fmax***	21	25	5	17.0	12	6	16	11.3	4	15	13	10.7	13.0

* Lowest number F_{max}, from among 72 implementation of scenarios that based on the number of jab factors are separated and chosen and it isn't including equal CGA

** Lowest number F_{max} from among 21 implementation of scenario that based on the number of job factors are separated and chosen and it is not including equal CGA.

*** Negative number of Table shows that priority of answer from CGA toward proposed solution procedure (GA2).

Table 3
The result of experiments for GA2 and CGA (Abbasian and Nahavandi, 2009)

CONCLUSIONS

In real scheduling problems exist of flexibility are effective technique for important of system performance. In this paper, implementation of more suitable solution for scheduling problem of flexible job-shop scheduling with parallel machines in dynamic environment were studied. Considering problem parameters and model of mathematic analysis though common procedures are very difficult and or non scientific by capability of genetic algorithm that had specific application in problem solution related schedule and scheduling were used. In conclusion, a genetic was proposed that its structure based on features of FDJSPM problem were designed. Considering one of the disadvantages of classic genetic algorithm is property of premature convergence. Techniques that by means of algorithm can use model knowledge in searching process, and intelligently prevent from trap into local optimum, and lead to increasing of its effectiveness in searching solution spatial and found optimum solutions. So, controlling parameters of proposed genetic algorithm dynamically changed during optimization process. The performance of developed writing of proposed algorithm were compared two present procedure in literature, that is showed that %0.73, %2.06 improvement in indicator "obtained beast solution" towards CGA, RKGA respectively and also %0.56 and %1.48 improvement in indicator "the average of obtained solution" toward CGA, RKGA respectively. Also proposed algorithm in the "average of implementation time" indicator is 0.5 (s) improvements toward CGA. The result of this development can use as verification no-free lunch theory "Use of model knowledge can help searching process and better solution for problem". Including future ways of research, increasing effectiveness of innovative algorithm in order to use model knowledge and found the more optimum solutions and also avoidance of trap into local optimum by the way decreasing of time implementation. On the other hands, we can consider other presumptions in modeling and solving scheduling problem and the problem that more close to them in order to we can use more real problem and out coming of the problem solution.

REFERENCES

Tay J.C. and Wibowo D., "An Effective Chromosome Representation for Evolving Flexible Job-Shop Scheduling", Genetic and Evolutionary Computation Conference, (2004).

Tay J.C. and Ho N.B., "Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems", Computer & Industrial engineering, (2007), Available from <www.elsevier.com>.

Kim Y.K., Park K. and Ko J., "A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling", Computers & Operations Research, 30, (2004), 1151–1171.

Bruker, P., Jurisch, B. and Sievers, B., "Discrete Applied Mathematics", 49, (1994), 107-127.

Carlier, J., and Pinson, E., Management Science, 35, (1989), 164-176.

Gen, M. and Cheng, R., "Genetic Algorithms and Engineering Design", John Wiley & Sons, (1997).

Brandimarte P. "Theory and Methodology, Exploiting process plan flexibility in production scheduling: A multi-objective approach". European Journal of Operational Research, 114, (1999), 59-71.

Ghedjati, F., "Genetic algorithms for the job-shop scheduling problem with unrelated parallel constraints: heuristic mixing method machines and precedence", Computer & Industrial Engineering, 37, (1999), 39-42.

Kacem, I., Hammadi, S. and Borne, P., "Approach by localization and multi objective evolutionary optimization for Flexible Job-Shop scheduling problems". IEEE Transaction on Systems, Man and Cybernetics, 32(1), (2002), 1–13.

Lee, Y.H., Jeong, C.S. and Moon, C., "Advanced planning and scheduling with outsourcing in manufacturing supply chain", Computer & Industrial Engineering, 43, (2002), 351-374.

Gen M. and Cheng R. "Genetic Algorithm in Search, Optimization and Machine Learning", Addition-Wesley, Reading, MA, (2004).

Kurz, M.E. and Askin, R.G., "Scheduling flexible flow-lines with sequence-dependent setup times", European Journal of Operational Research, 159, (2004), 66-82.

Kurz, M.E. and Askin, R.G., "Comparing scheduling rules for flexible flow-lines", Int. J. Production Economics, 85, (2003), 371-388.

Ho, N.B., Tay, J.C. and Lai, E. "An effective architecture for learning and evolving Flexible Job Shop schedules". European Journal of Operational Research, 179, (2007), 316–333.

Reeves Coline R., "Modern Heuristic Techniques for Combinatorial Problems", John Wiley & Sons, (1993).

Dagli, C.H. and Sittisathanchai, S., "Genetic neuro-schedular: a new approach for job shop scheduling", International Journal of Production Economics, 41, (1995), 135-145.

Gao J., Gen M. and Sun L., "Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm", Jornal Intelligent Manufacturing, 17, (2006), 493–507.

Abbasian, M. and Nahavandi, N., "Minimization Flow Time in a Flexible Dynamic Job Shop with Parallel Machines", Tehran, Tarbiat Modares University, Engineering

Department of Industrial Engineering, Master of Science Thesis, (2008).

Abbasian, M. and Nahavandi, N., "Minimization Flow Time in a Flexible Dynamic Job Shop with Parallel Machines", Tehran, 6th International Industrial Engineering Conference, (2009).

Amiri, M., Jamshidi, S.F., and Sadeghiani J.S. “A Genetic Algorithm Approach for Statistical Multi-Response Models Optimization: A Case Study”, Journal of Science & Technology, 49, pp. 131-137, (2009), Available from <www.google.com>.

Moreno-Torres, J.G., Llora, X., and Goldberg D.E. “Binary Representation in Gene Expression Programming: Towards a Better Scalability”, IlliGAL Report No. 2009003, (2009), Available from <www.google.com>.

Verma, A., Llora, X., Venkataraman, S., Goldberg, D.E. and Campbell R.H. “Scaling eCGA Model Building via Data-Intensive Computing”, IlliGAL Report No. 2010001, (2010).

Jansen K. "APPROXIMATION ALGORITHMS FOR FLEXIBLE JOB SHOP PROBLEMS", International Journal of Foundations of Computer Science, (2001).

Low, C., “Simulated annealing heuristic for flow shop scheduling problem with unrelated parallel machines”, Computer & operation Research, 32, (2005), 2013-2025.