Journal of AL-Qadisiyah for computer science and mathematics Vol.11 No.2 Year 2019

ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

Solving Job Scheduling Problem Using Fireworks Algorithm

Jamal N. Hasoon[1]
Department of Computer Science
University of Technology
jamal.hasoon@uomustansiriyah.edu.iq

Rehab Hassan[2]
Department of Computer Science
University of Technology
110019@uotechnology.edu.iq

Recived : 23\1\2019 **Revised : 27 \ 1\ 2019** **Accepted : 29\1\2019**

Available online : 17 /4/2019

Abstract:

Scheduling is critical part in most creation frameworks and information processing as sequencing of tasks or jobs framework executed on a grouping of processors. One of the NP-hard problem is "***Job Shop Scheduling Problem***". In this work, a method of optimization proposed called "**Fireworks *Algorithm***". The solutions divided into fireworks and each one applied sparks to find the best solution. For some selected spark applied Gaussian mutation to find enhanced solution and find optimum solution. FWA tested on dataset to improve performance and it do well with respect to some other algorithm like Meerkat Clan Algorithm (MCA), Camel Herds Algorithm) CHA) , and Cukoo Search Algorithm (CSA).

Keywords: component; Metaheuristic, Firework Algorithm, Flexible Job-shop Scheduling, Make-span Time

1. Introduction

There several NP-hard problem in artificial intelligent space, one of these problems is "***Job-Shop Scheduling Problem***" (JSSP) that extended to "***Flexible Job-Shop Scheduling Problem***" (FJSP) that allows its tasks to execute on multiple machine (many jobs on several machines). A sequence of tasks for each job should execute in a specific order [10, 16]. There are different works related to this problem to find the optimum solution, classical method perform well in low number of machines and be complicated when number of machines increased. In recently years, there are some of metaheuristic approaches presenting to solve FJSP. These algorithms are as follow: Genetic Algorithms (GA) [8], Tabu Search [4], Particle Swarm Optimization (PSO) [6], Cuckoo Search Algorithm (CSA) [13], Camel Herds Algorithm (CHA) [3], Meerkat Clan Algorithm (MCA) [2], and other approaches including Fuzzy Logic [9] all these ideas gained a lot of attention to use artificial intelligent algorithm to solve one of NP- hard problem.

Metaheuristic algorithms produced solutions in suitable time in general, when there is no classical algorithms [15]. To solve FJSP some constraints should aware such as in a job no task start until previous task completed, each task run on one machine at a time without preemption [11].

The problem of FJSP is that task run in variable time in on different machine and increase complexity [12].

Assignment and scheduling task and jobs are two sub-problem in FJSP and there are several objectives such as minimization make-span time, and minimization total idle time [1, 17]. These objectives most widely studied and there many dataset used for this problem such as Hurink-Data deal with this problem and widely used [7].

2. Fireworks Algorithm

Explosion of fireworks in night simulate in algorithm called FWA that produced sparks around each elimination [14] as shown in figure 1. It considered as Swarm intelligence algorithm, local search presented by each firework that spark around some positions in range called amplitude. The new populations cooperated to find global search. FWA contains some characteristics that verify it from other algorithms such as simplicity, locality, distribute parallelism, diversity and extendibility [15].

The basic idea of FWA works as follow: first is initializing randomly N fireworks with their fitness value (quality), each one evaluated to find the number of sparks and the range of explosion (explosion amplitude). The new generation obtained by explosion the previous fireworks and each one considered as local search. The amplitude balance the global and local space and ensure diversity by generate large population in small range and small population with large range when low fitness value this technique give chance to escape from local minima. The mutation in FWA called Gaussian mutation or "Gaussian Spark" and the selection strategy may use subset of whole population in each iteration to balance the global search.

3. Explosion Operator

For each firework in fireworks need, explosion operator that selects a number of sparks in explosion operator is found using equation 1 as follow:

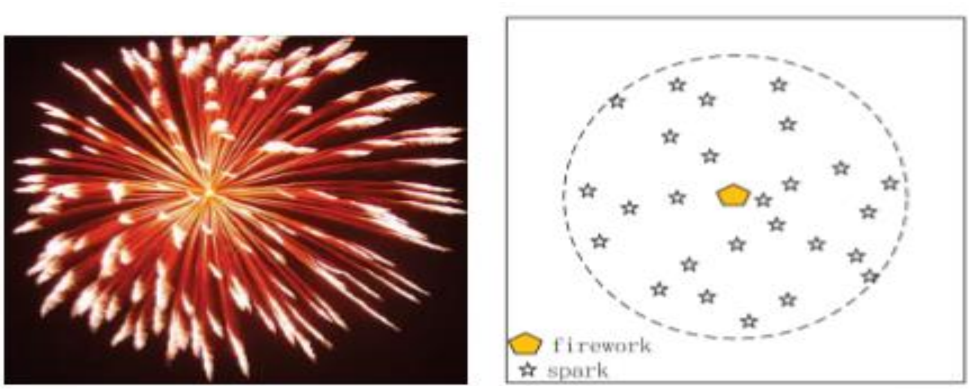

Figure 1: fireworks search algorithn

$$S_i = m \times \frac{y_{max} - f(x_i) + \varepsilon}{\sum_{i=1}^{N}(y_{max} - f(x_i)) + \varepsilon} \dots 1$$

Where S_i is number of spark in each firework, m is number selected as total number of individual in firework, N is total number of spark, y_{max} is the worst fitness value in population. $f(x_i)$ Is the fitness value of xi, and ε is small value for avoiding division by zero. The amplitude of explosion is found using equation 2 as follow:

$$A_i = \hat{A} \times \frac{f(x_i) - y_{min} + \varepsilon}{\sum_{i=1}^{N}(f(x_i) - y_{min}) + \varepsilon} \dots 2$$

Where A_i *is a* range (amplitude) of each one is, $\hat{A}$ is a fixed value specified (may be sum of all amplitude), and ε is used to same reason before.

These parameters used find distance on each firework using equation 3 as follow:

$$x_i^k = x_i^k + U(-A_i, A_i) \dots 3$$

Where $U(-A_i, A_i)$ is random number in uniform order in the amplitude Ai interval.

The total steps for generation spark shown in algorithm 1.

4. Mutation Operator

The selected individual or the current individual are used to applied mutation operation on it as x_i^k, where i represent from interval (1 to N) and k represent dimension of current state. The sparks of Gaussian explosion are calculated using equation 4 as follow:

$$x_i^k = x_i^k \times RndGauss(1,1) \dots 4$$

Where $RndGauss$ is random number in Gaussian distribution. Gaussian mutation in FWA explain in algorithm 2.

5. Mapping Rule

The mapping rule is a process that keeps all individuals in population in the accepted range. Any individual result from FWA operation return inside range space if it lies out boundaries by applying modular arithmetic operation. The mapping rule utilizes a modular operation and is stated as follows:

$$x_i^k = X_{LBd,k} + x_i^k ArthMod(X_{UBd,k} - X_{LBd,k}) ..5$$

Algorithm 3: Fireworks Algorithm

Step 1: Select Randomly N location for fireworks
Step 2: While Terminate cond is false **Do**
Step 3: Select N represent a fireworks at N locations:
Step 4: **for** all x_i in fireworks **Do**
Step 5: Calculate *Si* **(**the number of sparks)
Step 6: Calculate *Ai* **(**the Amplitude of sparks**)**
Step 7: **End for**
Step 8: Ms=NoOfSparkofGaussianMutation
Step 9: **for** k =1→ Ms **do**
Step 10: select x_i randomly
Step 11: Generate Si (spark for each one)
Step 12: **next**
Step 13: select Best Si (best sparks using selection strategy)
Step 14: end while

Algorithm 1: Generate Spark

Step 1: Generate Initial Population
Step 2: Calculate *F(x)* that represent Fitness Value for each one (firework)
Step 3: Calculate *Si* that represent the number of sparks
Step 4: Calculate *Ai* that represent the amplitude of sparks
Step 5: zr=random (1, dim) // random selection
Step 6: for m=1 to dim do
Step 7: **if** m ∈ z **then**
Step 8: $x_i^k = x_i^k + U(-A_i, A_i)$
Step 9: **End if**
Step 10: End for

Where x_i^k denotes sparks positions that lie out of bounds, while X_{UBound} and X_{LBound} are the maximum and minimum boundaries of a spark position. “ArthMod” represents modular arithmetic.

6. Selection Method

The selection method may need to find the distance between individuals, Euclidean distance

Algorithm 2: Gaussian Mutation

Step 1: Calculate the fitness value f (xi) for each firework
Step 2: Calculate the coefficient g =N (1,1)
Step 3: z=random (1, dimension) //choose randomly from dimension
Step 4: for k=1 to dimension do
Step 5: **if** k ∈ z **then**
Step 6: $x_i^k = x_i^k \times RndGauss(1,1)$
Step 7: **End if**
Step 8: End for

(equation 6) is a measurement used to find the nearest one.

$$R(x_i) = \sum_{i=1}^{K} d(x_i, x_j) = \sum_{i=1}^{K} \|x_i - x_j\| \ldots 6$$

Where d represent distance (Euclidean distance for example) between any two individuals x_i and x_j. This distance may be combined sparks results by explosion operator or mutation operator.

Roulette Wheel one of methods may be used in FWA to select new generation, depend on $P(x_i)$, that calculate in equation 7 as follow:

$$P(x_i) = \frac{R(x_i)}{\sum_{j \in k} R(x_j)} \ldots 7$$

High distances have more chances to be in new generation and increased the diversity of the population.

7. Proposed method

The proposed method used FWA algorithm for solve FJSSP using sequence of steps. The represent by selecting tasks through assigning a resource for each one, to set up its start time, and so on until scheduling all the tasks. FWA used random operation to distribute jobs on the machine as population as initial then divide them into groups. Find number of sparks in each one. Apply explosion on each spark to find all sparks. Apply mutation on the random spark selected from total spark and replace if the result butter.

The job distributed on machine denoted by job-op on selected machine (1-1, 1-2, 1-3, 2-1, 2-2, 2-3, 3-1, 3-2, 3-3) these jobs and its operation distribute on machine randomly to initialize populations (initial populations) for each population find make-span time (expected finish time) that used as fitness function of algorithm. The goal of this algorithm is to find butter distribution of jobs and operation on available machine (resources) to execute in minimum time as possible. For applying fireworks algorithm need to select number of fireworks as specifying set of groups for example {100, 200, 400, 250 50…etc.}.

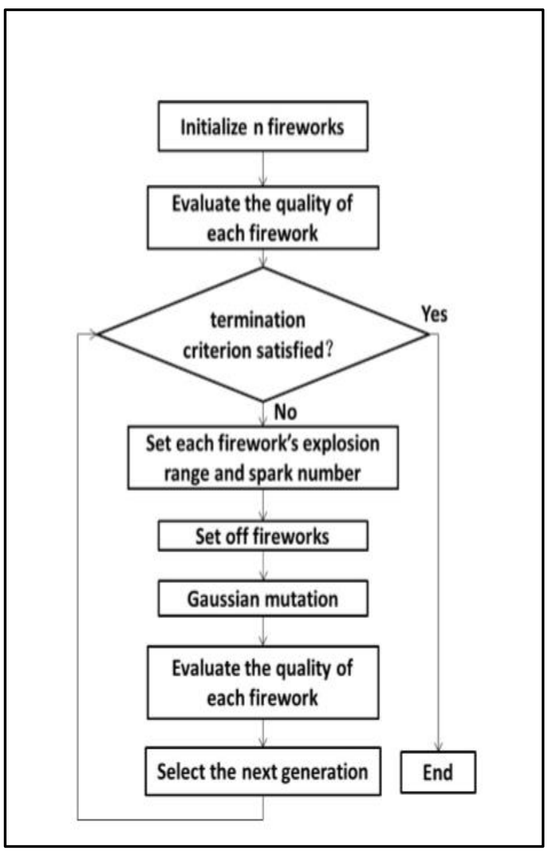

Figure 2: main steps of firework algorithm

Journal of AL-Qadisiyah for computer science and mathematics Vol.11 No.2 Year 2019

ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

	Job1		
	Mc1	**Mc2**	**Mc3**
Op1	26	77	41
Op2	84	49	33
Op3	64	87	95
	Job2		
	Mc1	**Mc2**	**Mc3**
Op1	50	53	70
Op2	86	87	33
Op3	43	106	63
	Job3		
	Mc1	**Mc2**	**Mc3**
Op1	29	X	25
Op2	122	121	31
Op3	122	106	X

Figure 3: Job scheduling example

Each group will consider as firework and apply some operation as local search to find the best make-span time. Select elements of each group using one of selection strategy, find minimum and maximum fitness, and apply equation 1 to find number of spark and equation 2 to find amplitude range. For each spark, Gaussian mutation is applied to check the fitness value of the result. The mutation in FJSP used a couple of sparks then select a random value of sequence in each one and swap them and again find fitness value of each one if is butter is taken otherwise keep as them before. Figure 4. Explain mutation operation.

Figure 4: FJSP execution

8. Experimental Results

To improve the results of implementation algorithm an instances called Hurink et al Dataset, [14, 15] that have three different instance sets “edata”, “rdata”, and “vdata” used in experiments. The method applied with multiple runs and compare with other method to find the performance of algorithm. The lower bounds of selected dataset explain in table 1 that be the goal of scheduling method for evaluation. In proposed work there, some previous work used same these datasets. Total experimental results illustrate in table 1 of the proposed method and some related work used the same dataset. To evaluate the best solution of algorithm Mean Relative Error (MRE) used as explain in equation 8 as follows:

$$MRE = \frac{(Cmin - LB)}{LB} \quad ...8$$

Where ***Cmin*** is the best solution found, ***LB*** is lower bound of optimum solution

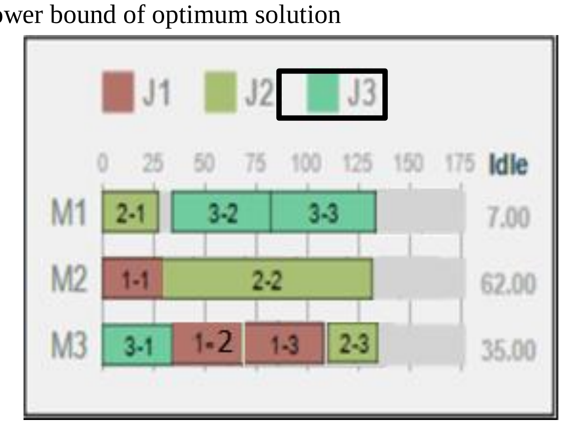

Table 1: Mean Relative Error of proposed

E data					
Inst.	**LB**	**MCA**	**CHA**	**CS**	**FWA**
mt06	55	0.001	0.001	0.018	0.001
mt10	871	0.022	0.094	0.367	0.036
la1	609	0.219	0.458	0.197	0.259
la2	655	0.095	0.257	0.195	0.067
la3	550	0.221	0.331	0.213	0.296
la4	568	0.192	0.461	0.248	0.204
la5	503	0.232	0.354	0.203	0.059
la6	855	0.304	0.546	0.142	0.106
la7	762	0.329	0.647	0.260	0.253
la8	845	0.252	0.488	0.185	0.058
R data					
Inst.	**LB**	**MCA**	**CHA**	**CS**	**FWA**
mt06	47	0.078	0.001	0.170	0.042
mt10	679	0.211	0.194	0.571	0.144
la1	570	0.278	0.167	0.268	0.070
la2	529	0.313	0.197	0.285	0.092
la3	477	0.326	0.237	0.302	0.071
la4	502	0.303	0.241	0.287	0.105
la5	457	0.315	0.243	0.263	0.227
la6	799	0.362	0.307	0.219	0.047
la7	749	0.333	0.342	0.224	0.038
la8	765	0.364	0.387	0.226	0.073
V data					
Inst.	**LB**	**MCA**	**CHA**	**CS**	**FWA**
mt06	47	0.041	0.001	0.170	0.002
mt10	655	0.233	0.092	0.526	0.192
la1	570	0.301	0.177	0.277	0.142
la2	529	0.313	0.059	0.276	0.068
la3	477	0.324	0.203	0.315	0.895
la4	502	0.289	0.277	0.299	0.081
la5	457	0.292	0.162	0.285	0.078
la6	799	0.341	0.304	0.228	0.027
la7	749	0.337	0.292	0.256	0.054
la8	765	0.352	0.187	0.244	0.107

Many tests applied on FWA that used to solve FJSP for evaluating the result by changing some algorithms parameters; the population size taken are 100, 200, 500, and 1000, the maximum iteration is 1000. Figure 5 shows the difference in performance between the proposed algorithm (FWA) and some other algorithm such as Meerkat Clan Algorithm (MCA), Camel Herds Algorithm (CA), and Cukoo Search Algorithm (CS). FWA methods are in some dataset better than other one and achieve in less number of iteration.

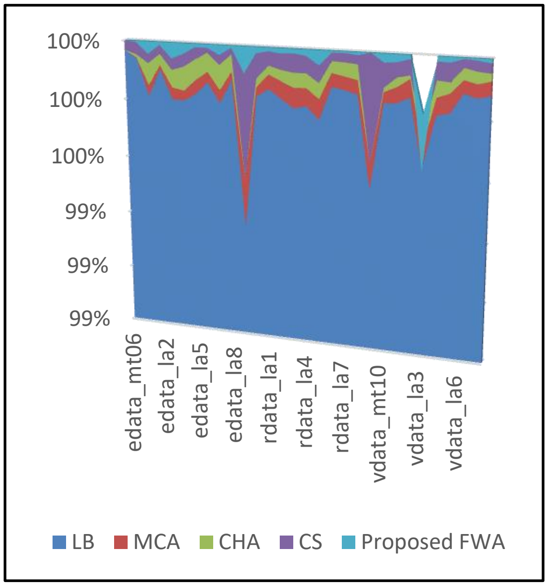

Figure 5: Mean Relative Error (MRE) of proposed method with related work

Conclusion

In proposed FWA as optimization method for FJSP for assigning all possible resources and project management to show the abilities of FWA in optimization. It used for different combinations of tasks length and assignation variability. FWA implemented depending on the behavior fireworks explosion and the amplitude of each spark. The performance of the algorithm measured based on a make-span function as fitness function. FWA present a better diversity solution through solving problem and high parallelism ability. The other characteristics are simplicity in mutation operator and in selection operator. FWA perform well and reduced make-span value when compare the results with some other work on the same dataset. To enhance convergence rate and the accuracy of FWA optimization problem may use different type of mutation method and may use dynamic evaluation on the parameters used in FWA iteration.

References

[1] Abir Ben Hmida, Mohamed Haouari, Marie-Jos_e Huguet, and Pierre Lopez, "***Discrepancy Search for the Flexible Job Shop Scheduling Problem***"; HAL Id: hal-00461981 Submitted on 8 Mar, 2010.

[2] Ahmed T. Sadiq, Hasanen S. Abdullah, and Zied O. Ahmed, "***Camel Herds Algorithm: a New Swarm Intelligent Algorithm to Solve Optimization Problems***", International Journal on Perceptive and Cognitive Computing, Vol. 3, no. 1, 2017.

[3] Ahmed T. Sadiq, Hasanen S. Abdullah, and Zied O. Ahmed, "***Solving Flexible Job Shop Scheduling Problem Using Meerkat Clan Algorithm***", Iraqi Journal of Science, Vol. 59, 2018.

[4] A. Gonzalez-Sieira, A. Bugarin, M. Mucientes, J. Moran, "***A Tabu Search Optimization Module for Scheduling***", Design and Integration in the Open Source Tool LibrePlan for Project Management 2011.

[5] Behnke, D., Geiger, M. J. ,"***Test instances for the flexible job shop scheduling problem with work centers***", Research Report RR-12-01-01, Helmut-Schmidt Universit¨ at, Hamburg, Germany, 2012.

[6] Hesam Izakian, Behrouz Tork Ladani, Kamran Zamanifar, and Ajith Abraham," ***A Novel Particle Swarm Optimization Approach for Grid Job Scheduling***", International Journal of Advanced Manufacturing Technology, Volume 60, Issue 1, pp. 303-315, Springer-Verlag, 2012.

[7] Ke Ding, Shaoqiu Zheng, and Ying Tan, "***A GPU-based Parallel Fireworks Algorithm for Optimization***", GECCO'13, July 6-10, 2013.

[8] Lee Yih Rou and Hishammuddin Asmuni," ***A Study ofCooperative Co-evolutionary Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem***", World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:4, No:12, 2010

[9] Ling Wang, Gang Zhou, Ye Xu, Shengyao Wang and Min Liu, "***An effective artificial bee colony algorithm for the flexible job-shop scheduling problem***", The International Journal of Advanced Manufacturing Technology, Volume 60, Issue 1, pp. 303-315, Springer-Verlag, 2012.

[10] M. L. Pinedo, "***Scheduling: Theory, Algorithms, and Systems***", 4th ed., Springer-Verlag, 2012.

[11] M. Mastrolilli, "Flexible Job Shop Problem", http://www.idsia.ch/monaldo/fjsp.html.

[12] Parviz Fattahi and Mohammad Saidi Mehrabad, Fariborz Jolai; "***Mathematical Modeling and Heuristic Approaches to Flexible Job Shop Scheduling Problems***", *Journal of Intelligent Manufacturing*, **18** (3): 331-342, Springer US; 2007.

[13] Prof. Dr.Ahmed T. Sadiq and Samer Alaa Hussein, "***Two Improved Cuckoo Search Algorithm to solve flexible Job shop Scheduling Problem***", International Journal on Perceptive and Cognitive Computing (IJPCC) Vol 2, Issue 2, 2016.

[14] Shaoqiu Zheng, Andreas Janecek, Junzhi Li, and Ying Tan, ***"Dynamic Search in Fireworks Algorithm"*** the National Natural Science Foundation of China under grants number 61375119, 61170057 and 60875080, 2014.

[15] Yao Wang, LinBo Zhu, Jiwen Wang and Jianfeng Qiu, "***An Improved Social Spider Algorithm for the Flexible Job-Shop Scheduling Problem***", International Conference on Estimation, Detection and Information Fusion (ICEDIF), 2015.

[16] Ying Tan and Yuanchun Zhu,"***Fireworks Algorithm for Optimization***", publication at: https://www.researchgate.net/publication/220704568, 2010.

[17] Xi-Guang Li, Shou-Fei Han, and Chang-Qing Gong,"***Analysis and Improvement of Fireworks Algorithm***", Academic Editors: Sergio Rajsbaum and George Karakostas, 2017.

Journal of AL-Qadisiyah for computer science and mathematics Vol.11 No.2 Year 2019

ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

حل مشكلة جدولة المهام باستخدام خوارزمية الالعاب النارية

جمال ناصر حسون [1]
الجامعة التكنولوجية
قسم علوم الحاسوب

د. رحاب فليح حسن [2]
الجامعة التكنولوجية
قسم علوم الحاسوب

المستخلص

الجدولة من المواضيع المهمة في مختلف مجالات العمل وتعتبر جدولة المهام في مجال معالجة البيانات وتوزيع المهام على اكثر من جزء في المنظومة وتعتبر من المشاكل الكبيرة والغير محددة (NP-Hard Problem) في هذا البحث تم اقتراح طريقة لحل هذه المشكلة تسمى خوارزمية الالعاب النارية وهي من الخوارزمية التي تبحث عن الحلول المثلى. الطريقة تعتمد على ايجاد مجموعة حلول عشوائية والخوارزمية تحاول تطبيق سلوك الالعاب النارية عن طريق البحث بواسطة تقسيم الحلول إلى ألعاب نارية وتطبيق الشرارات للعثور على أفضل حل بالنسبة لبعض الشرارة المختارة التي تم تطبيقها ، ودائما محاولة الحصول على حل محسن وإيجاد الحل الأمثل. تم اختبار FWA على مجموعة البيانات ومقارنتها مع بعض الخوارزميات لتي تستخدم قاعدة بيانات معدة لهذا الغرض، وكانت النتائج جيدة بالمقارنة مع الخوارزميات المختارة مثل خوارزمية عشيرة السرقاط (Meerkat Clan Algorithm (MCA وخوارزمية قطيع الابل Camel Herds Algorithm CHA وخوارزمية طائر الكوكو (Cukoo Search Algorithm (CSA.

الكلمات المفتاحية: التنقيب العالي، خوارزمية الالعاب النارية، جدولة المهام المرنة، وقت اكمال الصنع