



ISSN: 0067-2904

Solving Flexible Job Shop Scheduling Problem Using Meerkat Clan Algorithm

Ahmed T. Sadiq Al-Obaidi¹, Hasanen S. Abdullah¹, Zied O. Ahmed^{2*}

¹ Department of Computer Science, Technology University, Baghdad, Iraq.

² Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq

Abstract

Meerkat Clan Algorithm (MCA) that is a swarm intelligence algorithm resulting from watchful observation of the Meerkat (*Suricata suricatta*) in the Kalahari Desert in southern Africa. Meerkat has some behaviour. Sentry, foraging, and baby-sitter are the behaviour used to build this algorithm through dividing the solution sets into two sets, all the operations are performed on the foraging set. The sentry presents the best solution. The Flexible Job Shop Scheduling Problem (FJSSP) is vital in the two fields of generation administration and combinatorial advancement. In any case, it is very hard to accomplish an ideal answer for this problem with customary streamlining approaches attributable to the high computational unpredictability. Most scheduling problem are mind boggling combinatorial problem and exceptionally hard to settle. The experimental result that compare with Cuckoo Search algorithm, Artificial Fish Search Algorithm, and Camel Herd Algorithm show that the MCA can find optimal solution because it provides a good strategy.

Keywords: Meerkat Clan Algorithm, Flexible job shop scheduling, Swarm Intelligence, Cuckoo Search Algorithm, Artificial Fish Search Algorithm, Camel Herd Algorithm.

حل مشكلة جدولة المهام المرنة باستخدام خوارزمية عشيرة السرقاط

أحمد طارق صادق¹، حسنين سمير عبد الله¹، زيد عثمان أحمد^{2*}

¹ قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق.

² قسم علوم الحاسوب، كلية العلوم، الجامعة المستنصرية، بغداد، العراق.

الخلاصة

خوارزمية عشيرة السرقاط (MCA) هي إحدى خوارزميات الأسرب الذكية الناتجة عن المراقبة الساهرة لحيوان السرقاط في صحراء كالاهايري في جنوب أفريقي. حيوان السرقاط لديه بعض السلوكيات. كالمراقبة والبحث عن الطعام وحضانة الأطفال. تم استخدام هذه السلوكيات لبناء الخوارزمية من خلال تقسيم مجموعة الحل إلى مجموعتين، يتم تنفيذ جميع العمليات على مجموعة البحث عن الطعام. اما الحارس فهو يمثل أفضل حل. مشكلة جدولة ورشة عمل مرنة تعد أمراً حيوياً في مجالين من إدارة الجيل والتقدم التجميعي. وعلى أية حال، فإنه من الصعب جدا تحقيق إجابة مثالية عن هذه المشكلة مع نهج تبسيط العادة تعزى إلى عدم القدرة على التنبؤ الحسابية عالية. معظم مشكلة جدولة هي مشكلة محيرة العقل وصعبة بشكل استثنائي لتسوية .

*Email: zied_othman@uomustansiriyah.edu.iq

والنتيجة التجريبية التي تقارن مع خوارزمية البحث الوقواق، خوارزمية البحث عن الأسماك وخوارزمية قطع الجمال تبين أن خوارزمية السرقات يمكن أن تجد الحل الأمثل لأنه يوفر استراتيجية جيدة.

Introduction

The inspiration of Swarm Intelligence is the aggregate insight of gatherings of straightforward specialists, for example, creepy crawlies, angles, winged creatures, microscopic organisms, worms, and different creatures in light of their conduct, all things considered. As basic as these creatures seem to be, they can show remarkable knowledge at whatever point they work all in all as a gathering. These algorithms track the aggregate conduct of creatures that display decentralized, self-organized out examples in their scrounging obligations. Cases of these calculations are the Bee Colony Optimization, Firefly Algorithm, Particle Swarm Optimization, Ant Colony Optimization, Artificial Bee Colony, Bacteria Foraging Algorithm, et cetera [1].

Meerkats, otherwise called suricates, are little (<1 kg) carnivores having a place with the mongoose family, which contains 37 species in 18 genera and two subfamilies. Meerkats are committing agreeable raisers, living in gatherings of up to 50 people. Inside the mongoose family, meerkats are a standout amongst the most exceptionally gregarious species, with other all around contemplated social mongooses including the united mongoose and diminutive person mongoose. Meerkats are betray adjusted, with their dissemination limited to the semi-bone-dry areas of south-western Africa (counting southern Angola, Namibia, Botswana and South Africa). They are not undermined and their preservation status is classified as 'slightest worry' by the International Union for Conservation of Nature [2].

Scheduling for the adaptable activity shop is essential in the two fields of creation administration and combinatorial improvement. Be that as it may, it is very hard to accomplish an ideal answer to this issue in medium and genuine size issue with conventional improvement approaches inferable from the high computational unpredictability [3].

This work improves the way to find optimal solution for FJSSP, throw using method inspired by behaviour of Meerkat's in the wild. The Meerkat Clan Algorithm proposed in [4], this method produces grouping of jobs on number of machines that limit the estimation of makespan function. Whatever is left of this paper is sorted out as takes after. Number of related works are presented in Section2. Section3 shows lifestyle for Meerkat behaviour. The proposed algorithm shows in Section4. Section5 contains applying the problem and show the experimental results. Section6 presents the conclusion.

Related Work

This paper presents an approach that Inspired from Meerkat lifestyle, this section presents the asymptotic studies.

In [5] (2010), they displayed another way to deal with tackle FJSSP and contrast it and another inquiry calculation. They propose a variation of the Climbing Discrepancy Search approach for taking care of this problem, and furthermore introduce different neighbourhood structures identified with task and sequencing problem.

In [6] (2011), the work shows a half and half Pareto-based discrete artificial bee colony algorithm for comprehending the multi-objective FJSSP. In the half breed calculation, every arrangement relates to a nourishment source, which makes out of two parts, i.e., the directing segment and the planning segment.

In [7] (2016), this paper exhibited enhanced Cuckoo search algorithm to determine FJSSP and contrasted it and the first CS algorithm connected on HUdataset, the improvement was done through two methodologies initially in light of current based neighbour age, the second in view of a frequented Levy flight.

In [8] (2017), this paper proposed an improved AFSA algorithm for solving the Flexible Job Shop Scheduling Problem (FJSSP). The improvement is based on Variable Neighbourhood Descent (VND) strategy which is performed on AFSA by different neighbourhood structures to improve the performance of the original AFSA. The improved algorithm called (AFSA-VND) has been tested on some FJSSP benchmark instances for performance examination.

In [9] (2017), this paper presents a new swarm intelligence Camel Herd Algorithm. FJSSP with the aim to limit makspan that using the proposed CHA to solve it. The proposed algorithm gives a decent

assorted variety arrangement through the camel system in view of neighbour which rely upon the pioneer of crowd with moistness proportion in the forsake.

Meerkat Behaviour

Meerkats live in gatherings, with each gathering comprising of a predominant match and around 20 (however up to 50) male and female subordinate partners, who are for the most part posterity from the dominants' past rearing endeavours or foreigner guys. Meerkats are described by expansive predominant one-sided skews in survival and generation, and extensive subordinate one-sided skews in commitments to helpful care [8].

A. Sentry behaviour

Meerkats show charitable conduct inside their provinces; at least one meerkats will stand sentry (lookout) while different individuals are scrounging or playing with a specific end goal to caution them of moving toward threats. At the point when a predator is recognized, the meerkat executing as sentry will give a notice bark, and every single other individual from the pack will run and cover up in one of the many jolt gaps the meerkats will have spread over their domain. The sentry meerkat will be the first to return from the tunnel and look for predators, continually woofing to keep the others underground. On the off chance that there is no danger, the sentry meerkat will quit yelping and the others will be protected to develop. Meerkats will likewise watch youthful that might be in the gathering. Females that have never delivered posterity of their own will frequently lactate to sustain the alpha match's young while the predominant female is away with whatever is left of the gathering. They will likewise shield the youthful from any danger, frequently jeopardizing their own particular lives to do as such. On notice of a peril, the sitter will either take the youthful underground to security and be set up to safeguard them if the threat can take after, or gather all youthful together and lie over them if withdrawing underground isn't conceivable [2].

B. Foraging Behaviour

Foraging behaviour is average of social mongooses, in which creatures spread out and rummage independently while keeping up visual and vocal contact. A pack scrounges deliberately and completely inside its home range, taking an alternate defeat every day and more often than not permitting no less than seven days for a zone to recharge its sustenance supply between visits. Disguised prey are situated by little and uncovered with the forefeet. Grown-up promptly share sustenance with adolescents in the pack. In hostage creatures, three inborn reactions to dynamic prey have been noted [10]:

- A tendency to chase any small fleeing object.
- To bite at the most actively moving part.
- To eat mammalian prey starting to the head

C. Baby-sitter Behaviour

Meerkats take an interest in various helpful exercises. The key commitments to agreeable care are looking baby-sitting, assistants stay at the tunnel with pups 25 while whatever is left of the gathering is away scavenging, and pup nourishing where aides give an extent of their sustenance things to pups while searching. Both baby-baby-sitters are joined by calculable enthusiastic expenses to the partner: sitters do without bolstering for up to 24 hours, which prompts critical weight reduction, and pup-feeders forfeit their own particular scavenge things for provisioning them to pups [10].

Meerkat Clan Algorithm (MCA)

Meerkats are social creatures that live in states of 5 – 30 people. Being amiable animals, they share both latrine and parental care duties. Each crowd has a command alpha male and overwhelm alpha female. Each crowd has its own domain which they here and there move if nourishment is rare or when constrained out by a more grounded horde. The important parameters that derived from Meerkat behaviour and employed in MCA is clan size n , foraging group size m , care group size c , worst foraging & care ratio Fr & Cr , and number of neighbours K .

As shown in Figure-1, in the first the algorithm initializes the parameters that used in algorithm, n represents the number of solution in clan. m , c is the size of foraging and care group respectively. Also, the algorithm identifies the worst ratio of foraging and care. Finally initialize the number of neighbours K .

The algorithm starts with generating a group of solution randomly, it is called clan with size n . The clan generated is evaluated through calculate the fitness function. The best solution from the clan

generated is chosen and call it Sentry. The rest of the clan is divided in to two groups: foraging group with size m (where $m < n$), and care group with size c ($n-m-1$).

Meerkat Clan Algorithm

Parameter

n	clan size 30 to 50
m	foraging size where $m < n$
c	care size $n-m-1$
Fr	worst foraging rate
Cr	worst care rate
k	neighbour solution

Begin

Generate random clan of solutions $\text{clan}(n)$

Compute fitness for clan solutions

Sentry = best solution of clan

Divide the clan into two groups (foraging & care)

While not termination condition Do

For $i=1$ to m

Call neighbor_generat (k , Sentry, foraging(i), best_one)

foraging(i)= best one from k neighbor

end for

Swap the worst for Fr solution in foraging group with best ones' solution in care group;

Drop the worst Cr solution from care group and generate ones' solution randomly;

Select the best one of foraging call it best_forg

If best_forg \leq Sentry then

Sentry \leftrightarrow best_forg

end if

end while

End

Figure 1- Meerkat Clan Algorithm.

Now for each solution in foraging group sent to the *neighbor_generate* sub algorithm that shown in Figure-2 and return the best solution from the *neighbor generated*. In *neighbor_generate* receive the K , *Sentry* and *foraging_i*. *neighbor_generate* generate K neighbours from *foraging_i* and calculate the fitness function for its. If the all neighbours generated are worst from *foraging_i* then generate the K neighbour form *Sentry*. The best solution is selected from K neighbours and return it to the main algorithm. If the *foraging_i* that sent to the *neighbor_generate* is worst of best neighbour, then replace it.

neighbor_generated

Input: K , Sentry, foraging _{i}

Output: best_one

Begin

Generate k neighbour from foraging;

Compute fitness of k

If there is no one best than foraging _{i} then

Generate k neighbour from Sentry

end if

end

Figure 2- neighbour generated subroutine.

The algorithm isolates the worst solution in foraging group according to the worst foraging ratio and replaces it with the best solutions in care group. Drop the worst solution in care group according to the worst care ratio and swap it with a new random solution generated.

The best solution in foraging group is selected and compared with Sentry, if it is best then replacing the Sentry with the best solution in the foraging group. These steps are repeated until termination condition. In the end Sentry is the best solution.

Example for FJSSP

A case of FJSSP is shown in Table1; the information is given as far as the grouping of operations that process on accessible machines. Table1 shows three jobs and three machines. For instance, the job 1 has operation 1, which is indicated as $O_{1,1}$, which can be handled by machine number 1 with preparing time 10 and furthermore by machine number 2 with preparing time 10. In addition, the principal operation of the job 2, $O_{2,1}$ can be prepared by $M1$ and $M3$ with handling time 20 to each.

Table 1- (2×3) FJSSP example

Jobs	Operation sequence	Operations	Machine alternative	Processing time
J1	O1,1, O1,2	O1,1	M1	10
			M2	10
		O1,2	M2	12
			M3	12
J2	O2,1, O2,2	O2,1	M1	20
			M3	20
		O2,2	M1	25
			M2	25
J3	O3,1, O3,2	O3,1	M2	15
			M3	15
		O3,2	M1	9
			M2	9

MCA starts with initializing parameters value. For this example, suppose that $n= 6$, $m= 3$ ($m<n$), $c= 2$ ($c=n-m-1$), $Fr= 0.3$, $Cr= 0.25$, and $k= 2$. So now MCA generates random clan solutions by distributing the jobs on the machines randomly, after computation makespan functions (fitness function) MCA selects the best solution as Sentry and divides the rest of clan into two groups: FG and CG. Figure-3 shows the clan solutions and the FG, CG, and Sentry.

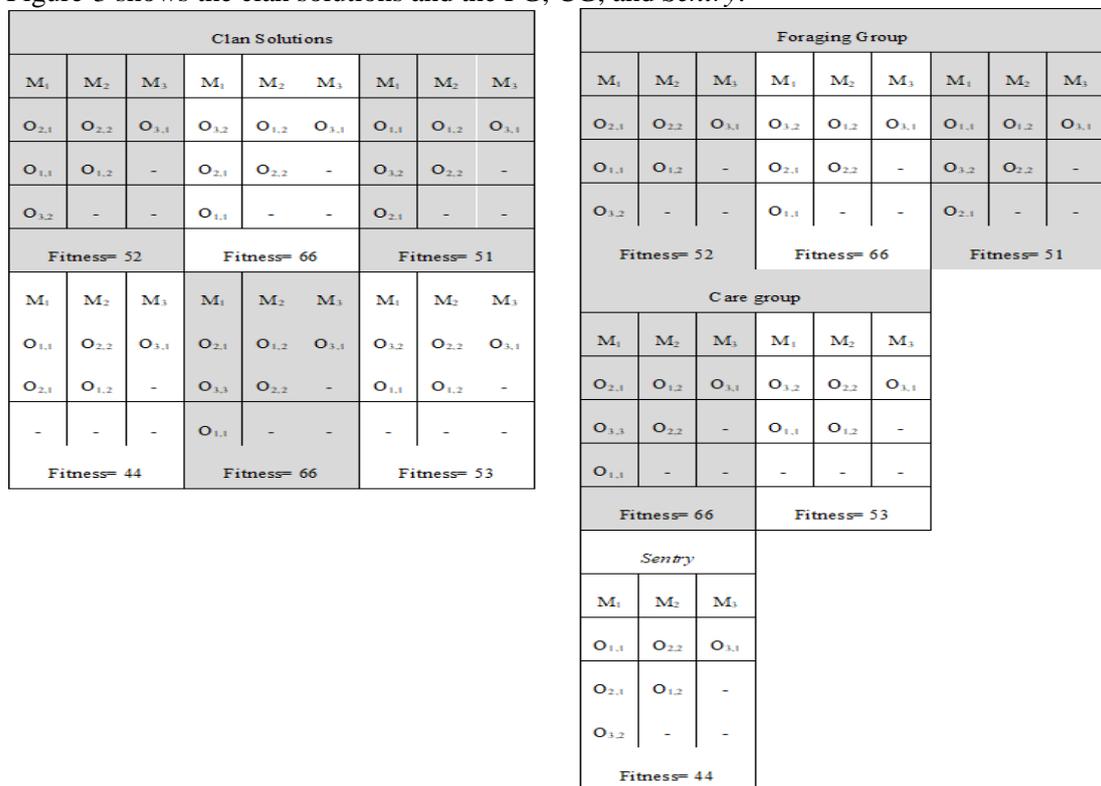


Figure 3- Clan solutions

The solutions for FG are sent to *neighbor_generate* function to generate neighbours and compute the makespan function for it. Figure-4 shows the first solution of FG and its neighbours generating.

M ₁	M ₂	M ₃			
O _{2,1}	O _{2,2}	O _{3,1}			
O _{1,1}	O _{1,2}	-			
O _{3,2}	-	-			
Fitness= 52					
Neighbors Generated					
M ₁	M ₂	M ₃	M ₁	M ₂	M ₃
O _{1,1}	O _{2,2}	O _{3,1}	O _{2,1}	O _{2,2}	O _{3,1}
O _{2,1}	O _{1,2}	-	O _{1,1}	O _{1,2}	-
O _{3,2}	-	-	O _{3,2}	-	-
Fitness= 44			Fitness= 52		

Figure 4- Neighbours generate for first FG solution

As seen in Figure-4 one of neighbour's solution is better than FG solution, so change it. After generating neighbours for all FG solutions, swap the 3% of worst FG solutions with 3% of best CG solutions. After number of generations the final result of MCA is found in *Sentry* shown in Figure-5.

<i>Sentry</i>		
M ₁	M ₂	M ₃
O _{2,1}	O _{1,1}	O _{3,1}
O _{3,2}	O _{2,2}	O _{1,2}
-	-	-
Fitness= 30		

Figure 5- Sentry solution

FJSSP Using MCA

The problem of FJSSP is to assign each operation to a machine and to order the operations on the machines, such that the maximal completion time (makespan) of all operations is minimized. MCA uses equation 1 as a fitness function to find the optimal solution.

$$\text{Minimize } C_{\max} = \text{Max} (C_j, j=1, 2, \dots, n) \tag{1}$$

MCA passes through three main phases as follows:

1. Distribute jobs on the machine randomly for each solution in the clan.
2. Chose the best solution as a Sentry.
3. Divide the rest of clan into two groups (foraging & care).
4. Find neighbours for each solution in foraging group.
5. Swap the worst solution with best ones in care group.
6. Generate new random solutions and swapped with the worst ones in care group.

The proposed algorithm was coded in MATLAB R2013b and applied on Intel Core i7 2.70 GHz personal computer with 4GB RAM. Its tested with different samples of the FJSS datasets (HUdata) that have a group of (129) of [11] problems. These problems are three by [12] and 40 problems by [13] (1a01 to 1a40). [11] produced three groups: E data, R data and V data. The first group flexibility (1.15), while the second group average flexibility 2, and m/2 for third group (scope between 2.50 and 7.50), in which the machines number are equal.

This section measures the performance of MCA through comparing the experimental result of these algorithms with original Cuckoo search algorithm [8], original Artificial Fish Swarm Algorithm (AFSA) [9], and Camel Herd Algorithm (CHA) [10] when applied on the same data.

Table-2 shows the comparison between the results of proposed algorithm and the other algorithm results. First field shows the instances dataset name, the second field shows the lower bound for the instances, third field shows the result of the first proposed algorithm MCA. The field that shows the result of the proposed algorithms contains the best results obtained from the series of runs with different parameters values. The last three fields show the results of CHA and original algorithm of CS, AFSA, respectively. These difference between the proposed algorithms, CHA, CS, and AFSA are clearly shown in Figure-6.

Table 2- Result of testing FJSSP instances using the proposed algorithm & CHA & original CS & AFSA

Instances	LB	MCA	CHA	CS	AFSA
edata_mt06	55	55	55	56	55
edata_mt10	871	852	953	1191	985
edata_la1	609	780	888	729	620
edata_la2	655	724	823	783	691
edata_la3	550	706	732	667	578
edata_la4	568	703	830	709	606
edata_la5	503	655	681	605	518
edata_la6	855	1229	1322	976	860
edata_la7	762	1136	1255	960	811
edata_la8	845	1129	1257	1001	869
rdata_mt06	47	51	47	55	48
rdata_mt10	679	861	811	1067	848
rdata_la1	570	789	665	723	605
rdata_la2	529	770	633	680	563
rdata_la3	477	708	590	621	509
rdata_la4	502	720	623	646	540
rdata_la5	457	667	568	577	482
rdata_la6	799	1252	1044	974	827
rdata_la7	749	1123	1005	917	787
rdata_la8	765	1203	1061	938	797
vdata_mt06	47	49	47	55	47
vdata_mt10	655	854	715	1000	773
vdata_la1	570	815	671	728	599
vdata_la2	529	770	560	675	579
vdata_la3	477	706	574	627	508
vdata_la4	502	706	641	652	536
vdata_la5	457	645	531	587	484
vdata_la6	799	1212	1042	981	829
vdata_la7	749	1130	968	941	781
vdata_la8	765	1181	908	952	785

As presented in table 1, for Edata MCA is better than CS in 4 out of 10, and MCA is better than CS in 2 out of 10 for Rdata and Vdata, while the other results are converged. As AFSA is better than MCA for most instances. As shown the instances that are always better when applied MCA is mt06 and mt10 for all sets, from this comparison, it was concluded that MCA approaches to CS results and MCA is better when instances have less number of jobs, these comparisons illustrate in Figure-3.

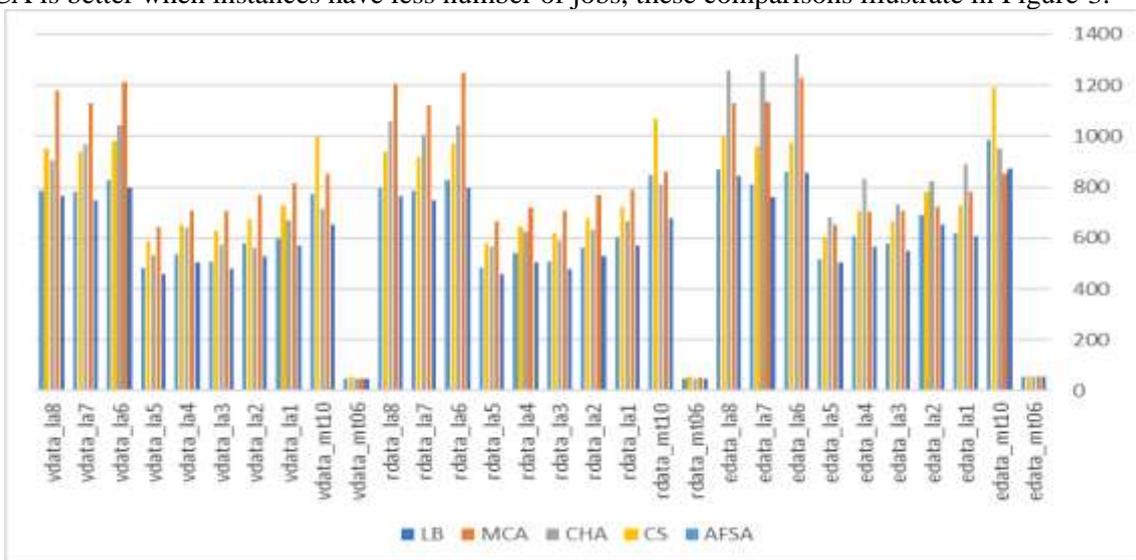


Figure 6- comparison between MCA, CHA, CS, and AFSA

Conclusion

MCA is implemented depending on the behaviour of Meerkats in wild. A Meerkat have many behaviours. The proposed algorithm uses the three main behaviour: sentry, foraging, and baby-sitter. The MCA shows amazing performance through dividing the solution set into two sets (foraging & care). Most of the operations are performed on foraging set and the worst solutions are replaced with the best ones in care solution. The worst solution in care set is dropped and added another solution that is created randomly. The performance of the algorithm was measured based on a makespan function. Also, this algorithm is employed to solve the FJSSP. The FJSSP with the aim to limit a makspan function that using MCA to solve it. The MCA algorithm present a better diversity solution through the Meerkat algorithm upon on neighbours which based on the sentry behaviour with foraging and baby-sitter groups.

References

1. Dhurandher, S.K., Misra, S., Pruthi, P., Singhal, S., Aggarwal, S. and Woungang, I. **2011**. "Using Bee Algorithm for Peer-to-Peer File Searching in Mobile Ad Hoc Networks", *Journal of Network and Computer Applications*, **34**(5): 1498–1508.
2. Sinéad English, **2009**. "Individual Variation in Cooperative Behavior in Meerkats"; University of Cambridge in application for the degree of Doctor of Philosophy.
3. 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.
4. Ahmed T. Sadiq Al-Obaidi, Hasanen S. Abdullah, and Zied O. Ahmed; " Meerkat Clan Algorithm: a New Swarm Intelligence Algorithm"; *Indonesian Journal of Electrical Engineering and Computer Science*; 2018.
5. Abir Ben Hmida, Mohamed Haouari, Marie-Jos_e Huguet, Pierre Lopez. **2010**. "Discrepancy Search for the Flexible Job Shop Scheduling Problem"; HAL Id: hal-00461981 Submitted on 8 Mar.
6. Jun-Qing Li, Quan-Ke Pan, Kai-Zhou Gao, **2011**. " Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems"; *The International Journal of Advanced Manufacturing Technology*, August , **55**(9): 1159–1169.
7. Dr. Ahmed T. Sadiq Al-Obaidi and Samer Alaa Hussein, **2016**. "Two Improved Cuckoo Search Algorithm to solve Flexible Job Shop Scheduling Problem" *International Journal on Perceptive and Cognitive Computing*; **2**(2).
8. Dr. Ahmed T. Sadiq Al-Obaidi and Samer Alaa Hussein, **2017**. "An Improved Artificial Fish Swarm Algorithm to Solve Flexible Job Shop Scheduling"; *New Trends in Information & Communications Technology Applications (NTICT)*.
9. Johanna F. Nielsen, **2012**. "The Evolutionary Genetics of Meerkats (*Suricata suricatta*)"; The University of Edinburgh for the degree of Doctor of Philosophy.
10. Ahmed T. Sadiq Al-Obaidi, Hasanen S. Abdullah, Zied O. Ahmed, **2017**. "Camel Herds Algorithm: a New Swarm Intelligent Algorithm to Solve Optimization Problems"; *International Journal on Perceptive and Cognitive Computing (IJPCC)*, **3**(1).
11. Moira J. van Staaden; **1994**. "Mammalian Species "The American Society of Mammalogists: No. 483, pp. 1-8.
12. Hurink E., Jurisch B. and Thole M. **1994**. "Tabu Search for the Job Shop Scheduling Problem with Multi-purpose Machines"; *Operations Research-Spektrum*, **15**: 205–215.
13. Fisher H. and Thompson G.L. **1993**. "*Probabilistic learning combinations of local job shop scheduling rules*"; In Industrial Scheduling, J.F. Muth and G.L. Thompson (Eds), Englewood Cliffs, NJ: Prentice-Hall. pp: 225–251.