Desheng LI[1], Na DENG[2]

Anhui Science and Technology University (1), Hubei University of Technology (2)

An electoral quantum-behaved PSO with Lévy flights for permutation flow shop scheduling problem

***Abstract**. Permutation flow shop scheduling problem (PFSSP), a NP-hard combinatorial optimization problem, has strong engineering background of finding the optimal processing sequence and time of jobs on machines under the constraints of resources. Recently, several approaches based on Particle Swarm Optimization (PSO) have been developed to solve the PFSSP, and the experimental results show that they are efficient. To solve this issue, a novel variant of quantum-behaved particle swarm optimization algorithm for permutation flow shop scheduling is proposed in this paper. This algorithm is a combination of quantum-behaved PSO, electoral mechanism, and a disturbance generated by Lévy flights. Inspired by the election behavior in society, an electoral and cooperative mechanism is imported to get the elite particles from the primitive sub-swarms respectively. Moreover, the character unequal hop length of Lévy flights provides a method to escape the local optima efficiently. The numerical results on the Taillard's benchmark also show it outperforms other related algorithms.*

Streszczenie:** Problem szeregowania zmiany przepływów magazynowych (PFSSP) jest silnie nie–wielomianowym (NP) problemem optymalizacji kombinatorycznej. Ma ważny inżynierski aspekt w wyznaczaniu optymalnej kolejności procesu i czasu pracy maszyn, wymuszonej zmianą zasobów. Ostatnio, do rozwiązania PFSSP, zastosowano szereg przybliżeń opartych o algorytm optymalizacji rojem cząstek (PSO) a wyniki praktyczne pokazują, że są to rozwiązania efektywne. W prezentowanym opracowaniu, do szeregowania przepływów magazynowych, zaproponowano nowy wariant algorytmu optymalizacji rojem cząstek z zachowaniem kwantowym (QPSO). Algorytm jest kombinacją QPSO, mechanizmu wyborczego i zakłóceń generowanych rozkładem lotów Levy'ego. Do wyłonienia cząstek elitarnych z prymitywnego pod-roju wykorzystano, inspirowany zachowaniami wyborczymi w społeczeństwie, mechanizm wyborczy i współpracy. Ponadto, unikalny charakter długości skoków lotów Levy'ego pozwala skutecznie uniknąć optimów lokalnych. Wyniki numeryczne, przeprowadzone na danych testowych Taillard'a, także wskazują na przewagę nad innymi porównywalnymi algorytmami.* ***Algorytm wyborczy PSO z kwantowym zachowaniem z lotem Levy'ego do problemu szeregowania zmian przepływów magazynowych

Keywords: Particle Swarm Optimization; Quantum-Behaved; Lévy flights; Permutation Flow Shop Scheduling Problem.
Słowa kluczowe: Optymalizacja rojem cząstek, Zachowanie kwantowe, Loty Levy'ego, Problem zmian przepływów magazynowych

1. Introduction

NP-complete combinatorial optimization problems occur in many real-world applications, which are believed that there exists no polynomial-time algorithm for probing best solutions. The flow shop scheduling problem (FSSP) is a typical one of these kinds of combinatorial optimization problem with a strong engineering background of finding the optimal processing sequence and time of jobs on machines under the constraints of resources. Among them, Permutation FSSP (PFSSP), with an additional constraint that all jobs must enter the machines in the same sequence order, is also to find a job permutation that minimizes a specific performance criterion (such as makespan or total flowtime).

Recently, several approaches based on PSO algorithm have been also developed to solve the PFSSP, and the experimental results show that they are more efficacious than the algorithms based on GA and constructive heuristics, such as Gupta [1], Koulamas [2], and NEH [3]. However, these algorithms are still suffered from the problem of premature convergence and easily trapped into local optimum. Moreover, along with the increase of the dimension of particle, a problem called the curse of dimensionality that refers to various phenomena that arise when analyzing and organizing high-dimensional spaces would hinder the fast running of PSO program. In order to refrain from these shortcomings, some variations of PSO are proposed using techniques such as Variable Neighborhood Search (VNS) [4], Cooperative Evolution (CE) [5] and so forth. In literature [4], a hybrid PSO algorithm was proposed based on VNS, which adopted a random key rule to construct position of particle mapping to job scheduling. B. Yu et al. have developed an improved cooperative Particle Swarm Optimization, ICPSO, to solve PFSSP, which use both greed and random approaches with a policy of synthetically learning method in their research paper [5].

Nevertheless, because the PSO could not promise the convergence of optimization, so it not always reaches the global optima. Thus, some techniques are imported to improve its performance, such as Quantum-Behaved Particle Swarm Optimization (QPSO) by Sun et al.[6]. The iterative equation of QPSO is very different from that of PSO. Besides, unlike PSO, QPSO needs no velocity vectors for particles, and also has fewer parameters to adjust, making it easier to implement.

In this paper, we will introduce our PSO algorithms based on QPSO and CPSO, which not only uses an electoral mechanism, but also employs approach to avoid premature convergence, i.e., a Lévy flights disturbance.

2. Mathematical formulation of PFSSP

The PFSSP with makespan minimization is usually denoted as $F_m \mid prmu \mid C_{max}$, where m is the number of machines, $prmu$ represents only permutation schedules are permitted, and C_{max} denotes the optimization criterion. Compared to the original FSSP, it has smaller search space $n!$ than the $(n!)^m$ of sequencing jobs in FSSP.

Firstly, we suppose several notations: A finite set J of n jobs $\{J_i\}_{i=1}^n$ to be processed; A finite set M of m machines $\{M_k\}_{k=1}^m$ can perform operations; Each job J_i consists of m operations $(O_{i,1}, O_{i,2}, ..., O_{i,m})$; The parameter t_{ik} denotes the processing time of job J_i on machine M_k; $C_{J_i k}$ denote the completion time of job J_i on machine M_k; $\pi = \{\pi_1, \pi_2, ..., \pi_n\}$ denote a permutation of jobs.

Secondly, we also comply with the following hypotheses: All jobs J_i must be processed on every machine in the same sequence, given by the indexing of the machines; $O_{i,k}$ has to be processed on machine M_k for an uninterrupted and fixed processing time period, while no operation can be preempted; Each machine can process only one job and each job can be processed by only one machine at a time (capacity constraints). Then the completion time could be calculated as following equations (Eq.(1)-Eq.(5)):

(1) $$C_{\pi_1,1} = t_{\pi_1,1}$$

(2) $$C_{\pi_j,1} = C_{\pi_{j-1},1} + t_{\pi_j,1}, j \in 2,...,n$$

(3) $$C_{\pi_1,k} = C_{\pi_1,k-1} + t_{\pi_1,k}, k \in 2,...,m$$

(4) $$C_{\pi_j,k} = max\{C_{\pi_{j-1},k}, C_{\pi_j,k-1}\} + t_{\pi_j,k}, j \in 2,...,n, \quad k \in 2,...,m$$

(5) $$C_{max}\{\pi\} = C_{\pi_n,m}$$

In our study, the goal is to find the permutation of jobs that minimizes the makespan as in Eq.(6). So, the PFSSP with the makespan criterion is to find a permutation π^*. In the set of all permutations Π such that

(6) $$C_{max}\{\pi^*\} \le C_{\pi_n,m}, \forall \pi \in \Pi$$

3. The proposed algorithm: EQPSO-LF

3.1 EQPSO: a QPSO with electoral mechanism

The electoral mechanism is on the basis of the multi-swarm and cooperative variants of PSO, Cooperative Particle Swarm Optimization (CPSO) proposed by Van den Bergh F. in [7], in which the high-dimension search space can be decompose into small scale ones similar to the idea of RELAX/CLEAN algorithm. However, its difference to it is that due to the imported information exchange mechanism among particles, the more accurate estimates did not need reduplicative iterations any more. Compared to basic single swarm PSO, both robustness and precision are improved and guarantied. The key idea of CPSO is to divide all the n-dimension vectors into k sub-swarms. So the front n/k swarms are $\lceil n/k \rceil$-dimensional, and the $k-(n/k)$ swarms behind have $\lfloor n/k \rfloor$-dimensional vectors. In each pass of iteration, the solution is updated based on k sub-swarms rather than the original one. When the particles in one sub-swarm complete a search along some component, their latest best position will be combined with other sub-swarms to generate a whole solution. The function b performs exactly this: it takes the best particle from each of the other sub-swarms, concatenates them, splicing in the current particle from the current sub-swarm j in the appropriate position.

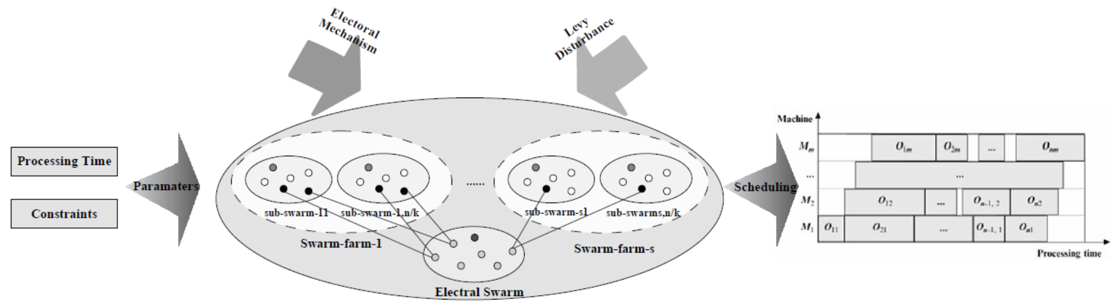


Fig.1. Electoral mechanism of ECPSO;

The principle of electoral mechanism is depicted in Fig.1, in which it can clearly seen that three parts: the local best position, the global best position in sub-swarm, and that of electoral swarm both take participate in the evaluation of fitness function with its own position. Note that the members of electoral swarm are voted from the primitive sub-swarms with dynamic population during different generation of iteration. The function b shown in Eq.(7) performs exactly this: it takes the best particle from each of the other sub-swarms, concatenates them, splicing in the current particle from the current sub-swarm j in the appropriate position. According to this function, the composition of P_{id}^{best}, P_{gd}^{best} and $\breve{P}_{gd}^{best}$ can be calculated based on Eq.(8)-Eq.(9). Particles in each sub-swarm update their latest best positions according to Eq.(8), while the latest global best positions of each sub-swarm are renovated by Eq.(9), where *Si* denotes the *i*-th sub-swarm.

(7) $$b(u,Z) = (S_1.P_{gid}^{best},...,S_{u-1}.P_{gid}^{best},Z,S_{u+1}.P_{gid}^{best},...,S_k.P_{gid}^{best}), \quad 1 \le u \le k$$

(8) $$b(u,S_u.P_{id}^{best}) = argmin\ fitness(b(u,S_u.P_{id}^{best}),b(u,S_u.P_{id})), \quad 1 \le u \le k$$

(9) $$b(u,S_u.P_{gd}^{best}) = argmin\ fitness(b(u,S_u.P_{id})), \quad 1 \le id \le s, 1 \le u \le k$$

In our algorithm, an electoral swarm is generated by the voting of primitive sub-swarms and also participates in evolution of swarm, whose candidate particles come from primitive sub-swarms with variable votes. In reverse, the number of selected particles could also impact the voting of the primitive sub-swarms, such as the total number of candidates and quota of selected ones. The selected candidates could share their components with best segments of position, which are then being composed into a new particle position to participate in the combining of positions. Like the treatment in our previous work [8], a new component of particle's position is also imported, i.e., P_{ed}^{best}, denoting the electoral best position composed by the dimensions of elected candidates. Recall that the local attractor in original quantum-behaved PSO can be written $\breve{P}_{id} = (c_1 r_1 P_{id} + c_2 r_2 P_{gd}) / (c_1 r_1 + c_2 r_2)$. After employing an electoral best position, it could be augmented into

(10) $$\breve{P}_{id} = (c_1 r_1 P_{id}^{best} + c_2 r_2 \breve{P}_{gd}^{best} + c_3 r_3 P_{ed}^{best}) / (c_1 r_1 + c_2 r_2 + c_3 r_3) .$$

So the new local attract position could be written as follows:

(11) $$\breve{P}_{id}' = \varphi \times P_{id}^{best} + \psi \times \breve{P}_{gd}^{best} + (1-\varphi-\psi) \times P_{ed}^{best} .$$

where $\varphi = c_1 r_1 / (c_1 r_1 + c_2 r_2 + c_3 r_3), \psi = c_2 r_2 / (c_1 r_1 + c_2 r_2 + c_3 r_3)$.

Due to the employment of this component, the particles in each sub-swarm therefore update their global best position by Eq.(12), which is the result associated with minimal fitness value of their local best positions and global best positions of electoral swarm.

(12) $$b(u,S_u.\breve{P}_{gd}^{best}) = argmin\ fitness(b(u,S_u.P_{id}^{best}),b(u,S_u.P_{ed}^{best})), \quad 1 \le id \le s, 1 \le u \le k$$

3.2 Lévy flights disturbance in EQPSO

The technique of random disturbance is often imported to improve the performance of PSO or QPSO. When QPSO was proposed, the Gaussian and Cauchy probability distribution disturbance have been used to avoid premature convergence. In [9], the random sequences in QPSO were generated using the absolute value of the Gaussian probability distribution with zero mean and unit variance. Based on the characteristic of QPSO, the variables of the global best and mean best positions are mutated with Cauchy distribution, and an adaptive QPSO version was proposed in [10].

In this paper, another random method, Lévy flights, is employed to do this work. Lévy flights, named after the French mathematician Paul Pierre Lévy, are Markov processes. After a large number of steps, the distance from the origin of the random walk tends to a stable distribution. In general, Lévy flights are a kind of random walk whose step lengths meet a heavy-tailed Lévy alpha-stable distribution, often in terms of a power-law formula:

(13) $$L(s) \sim |s|^{-1-\beta}.$$

where $0<\beta\leq 2$ is an index. A typical version of Lévy distribution can be defined as [11].

(14) $$L(s,\gamma,\mu)=\begin{cases}\sqrt{\dfrac{\gamma}{2\pi}}exp[-\dfrac{\gamma}{2(s-\mu)}]\dfrac{1}{(s-\mu)^{3/2}}, & 0<\mu<s<\infty;\\ 0, & s\leq 0.\end{cases}$$

As the change of β, this can evolve into one of Lévy distribution, normal distribution and Cauchy distribution.

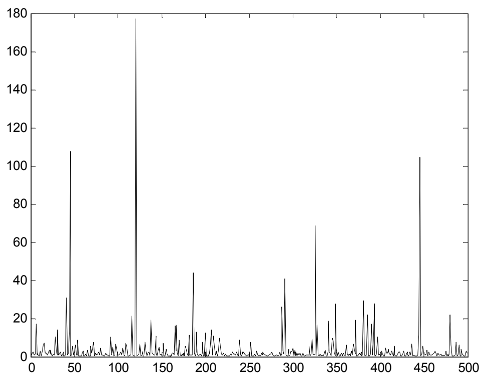


Fig.2. Step lengths of 500 random walks in Lévy flights;

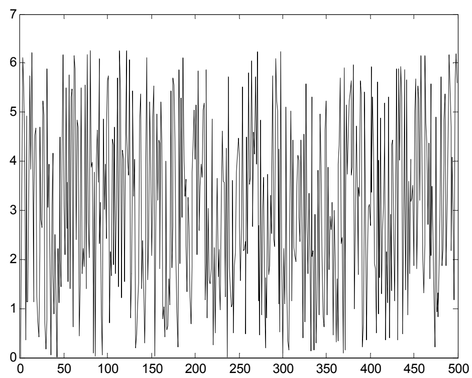


Fig.3. Angle values of 500 random turns in Lévy flights;

Taking the 2D-Lévy flights for instance, the steps following a Lévy distribution as in Fig.2, while the directions of its movements meet an uniform distribution as in Fig.3. As shown in Fig.4, an instance of the trajectory of 500 steps of random walks obeying a Lévy distribution. Note that the Lévy flights are often efficient in exploring unknown and large-scale search space than Brownian walks. One reason for this argument is that the variance of Lévy flights $\delta^2(t)\sim t^{3-\beta}, 1\leq\beta\leq 2$ increases faster than that of Brownian random walks, i.e., $\delta^2(t)\sim t$. Also, compared to Gaussian distribution, Lévy distribution is advantageous since the probability of returning to a previously visited site is smaller than for a Gaussian distribution, irrespective of the value of μ chosen.

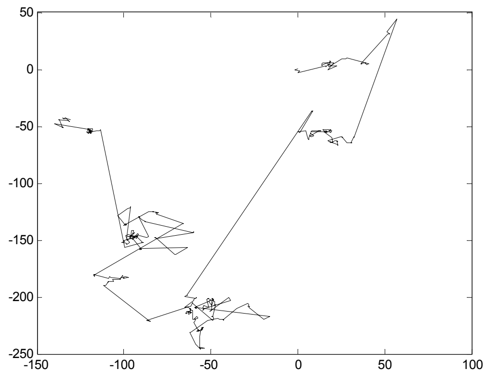


Fig.4. 2D Lévy flights in 500 steps;

From the update strategy of EQPSO-LF, we can draw a conclusion that all particles in EQPSO-LF will converge to a common point, leaving the diversity of the population extremely low and particles stagnated without further search before the iterations is over. To overcome the problem, we exert a disturbance generated by Lévy flights on the mean best position, global best position and electoral best position when the swarm is evolving as shown in the following Eq.(15)-Eq.(17). To the local attractor, the hop steps in Lévy flights promise the random travelsal in the search space. However, to the global and electoral best location, they only need a slightly disturbance, i.e., the angles meet an uniform distribution, to exploit the particles nearby.

(15) $$C_d' = C_d + step.$$

(16) $$\breve{P}_{gd}^{best'} = \breve{P}_{gd}^{best} + \varepsilon_1 \times angle.$$

(17) $$\breve{P}_{ed}^{best'} = \breve{P}_{ed}^{best} + \varepsilon_2 \times angle.$$

where ε_1 , ε_2 is a pre-specified parameter, $step$ is a number in a sequence by Lévy flights, $angle$ is the angles of directions in Lévy flights.

(18) $$P_{id} = \breve{P}_{id}' \pm \beta\times|C_d' - P_{id}|\times ln(1/u).$$

(19) $$\breve{P}_{id}' = \varphi\times P_{id}^{best} + \psi\times\breve{P}_{gd}^{best} + (1-\varphi-\psi)\times P_{ed}^{best'}.$$

3.3 Main steps

After discussing each component, we can now present the proposed EQPSO-LF algorithm in the following steps:

Algorithm 1: EQPSO-LF
Initiation;
Label 1: Generation primitive sub-swarms;
Foreach sub-swarm-i **In** sub-swarms **Do**
    Calculate the fitness value of makespan;
    If (run==firsttime)
    Then Update the personal and global optimal position as in QPSO;
    Else Update the personal and global optimal position with Eq.(11);
    Calculate the best particles;
    Compute the quota and votes of sub-swarm-i;
End Foreach
Generate the electoral swarm according to quota and votes;
Calculate the fitness value of makespan;
Foreach dimension-i **In** D **Do**

Update the personal and global optimal position by Eq.(18)-Eq.(19);
Update the particles based on quantum behavior with the Lévy disturbance by Eq.(15)-Eq.(16);
Calculate the best component of dimension-i of the electoral swarm;
End Foreach
Calculate the electoral best position according to Eq.(17);
Test whether satisfy the condition of termination;
If (Meet terminal condition) **Then** ends
Else repeat from **Label 1**;
End If
End.

4. Computational Results

4.1 Analysis of the diversities

Both the electoral mechanism and the disturbance generated by Lévy flights can diversify the population as an overall result achieved by diversification of the local attractor points. Herein we use a diversity measure to analyze the diversity changes of local attractor points in QPSO and EQPSO-LF shown in Eq.(20). In addition, the diversity measure for D-dimensional numerical problems is the "distanceto-averagepoint" measure defined as [13].

$$diversity(p) = \frac{1}{M} \times \sum_{i=1}^{M} \sqrt{\sum_{j=1}^{D} (p_{ij} - \overline{p_j})^2} \quad (20)$$

where M is the population size, D is the dimensionality of the problem, P_{ij} is the jth value of the ith individual, and P_j is the jth value of the average point. This diversity measure is dependent on swarm size, the dimensionality of the problem as well as the search range in each dimension. Low population diversity indicates that the swarm has clustered in a small region. Conversely, high population diversity indicates that the swarm has scattered in a wide region. Low population diversity is always taken the blame for the local convergence. However, high diversity may cause the algorithm not to converge. Thereupon the diversity should be considered together with the problem and the search process of the algorithm. The comparison of diversities between QPSO and EQPSO-LF is shown in Fig.5. From it, we can see the diversity in the latter is enhanced remarkablely than the original QPSO.

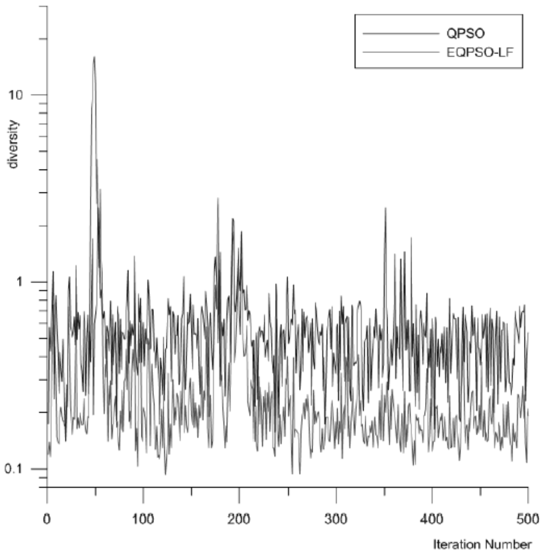
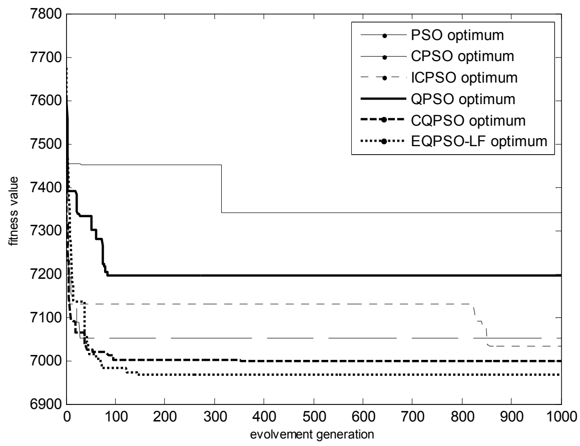


Fig.5. Diversities of 500 iterations with QPSO and EQPSO-LF;

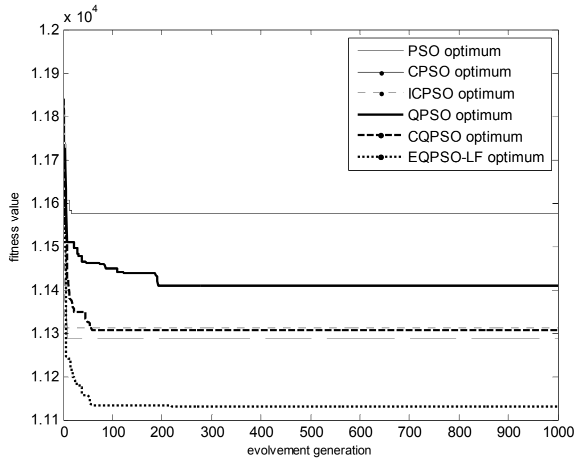
4.2 Results on Taillard benchmark

In this section, an analysis of the results acquired on the Taillard [15] benchmark suite are provided (the detailed results acquired for each of the 18 typical Taillard instances run are presented in Table 1, of which the solution quality is mainly measured in terms of the relative increase in makespan with respect to the best known solutions. Each instance was run 10 consecutive times and both deviations of the best and average makespan achieved are calculated (Min, Max and Avg), respectively. From Table 1, we can clearly get that the proposed EQPSO-LF algorithm performed greatly better than the plain PSO/QPSO algorithm. Also, compared to the basic Cooperative PSO (CPSO), Cooperative QPSO (CQPSO), the convergence property has been enhanced by the proposed techniques in the paper. Moreover, another interesting result is that although the EQPSO-LF has improved the quality of solution to a certain extent, but the ability to probe the best solution(denoted as BKS-Best Known Solution) is the same with CPSO and ICPSO.

Fig.6 illustrate the typical convergences of PSO, CPSO, ICPSO, QPSO, CQPSO and EQPSO-LF in Taillard's benchmark suite, in which Fig.6 (a)-(b) illustrate the TA090 and TA100 instances respectively. From these figures, it can be seen that the varying curves of objective values using the family of Cooperative PSO/QPSO descend much faster than using plain PSO. In addition, the fitness values descent to lower level by using EQPSO-LF than Cooperative PSO/QPSO due to the different algorithmic mechanism. The results of the experiments indicated that the proposed EQPSO-LF can lead to more efficiency and stability than plain PSO/QPSO, and Cooperative PSO/ QPSO.



(a)



(b)

Fig.6. Evolution curves of the optimal solutions for PSO, CPSO, ICPSO, QPSO, CQPSO and EQPSO-LF algorithm. (a) TA090, (b) TA100;

5. Conclusions

In this paper, a variant of QPSO, EQPSO-LF, is presented for minimizing the makespan in permutation flowshop scheduling problem. Inspired by the election behavior in society, an electoral and cooperative mechanism is imported to get the elite particles from the primitive sub-swarms respectively. Using this mechanism, the global optima are easier to find and diversity is also increased. Moreover, to help escape from local optima, a disturbance generated by Lévy flights is embedded as a hybrid strategy. Computational results and comparisons on Taillard benchmark also show it outperforms other related algorithms. Our future work is to generalize the application of the EQPSO-LF algorithm to solve other combinatorial problems.

Table.1. Results for PFSSP regarding the quality of solutions found in benchmark

Problem	BKS	PSO			CPSO			QPSO			CQPSO			EQPSO-LF		
		Min	Max	Ave	Min	Max	Ave	Min	Max	Ave	Min	Max	Ave	Min	Max	Ave
TA030	2178	2199	2240	2206.6	2178	2190	2182.4	2190	2240	2208	2178	2240	2198.2	2220	2240	2233.4
TA031	2724	2724	2748	2733.6	2724	2751	2733.1	2752	2827	2776	2724	2752	2735.4	2724	2748	2730
TA035	2863	2864	2887	2868.6	2863	2864	2863.8	2922	2966	2939.2	2863	2922	2878	2863	2922	2901.2
TA040	2782	2783	2817	2789.7	2782	2814	2787.7	2800	2907	2838.4	2800	2814	2808.8	2783	2817	2804.8
TA041	2991	3092	3209	3144.2	3061	3141	3105.3	3261	3431	3361	3092	3135	3129.2	3126	3261	3214
TA045	2976	3092	3209	3144.2	3061	3141	3105.3	3306	3417	3350	3126	3160	3145	3094	3160	3123.2
TA050	3065	3166	3257	3211.7	3132	3212	3162.1	3389	3500	3430.8	3132	3257	3198.4	3187	3389	3256.3
TA051	3850	4016	4125	4059.1	3971	4105	4033.7	4310	4446	4372.2	4125	4312	4278.1	4093	4125	4104.8
TA055	3610	3847	3948	3892.3	3749	3867	3801.3	4156	4284	4201	3847	3874	3861.2	3874	4156	4031
TA060	3756	3914	4082	3976.9	3909	4002	3942.9	4261	4412	4314	3909	4082	4011.4	3915	4079	4021
TA061	5493	5495	5533	5507.4	5493	5495	5494.2	5564	5663	5606.2	5505	5550	5527.8	5514	5519	5517.2
TA065	5250	5255	5312	5226.4	5252	5255	5254.1	5352	5461	5406.2	5267	5314	5294	5266	5352	5298.4
TA070	5322	5342	5376	5356	5328	5342	5334.8	5486	5562	5522.2	5348	5392	5375	5341	5348	5345.8
TA071	5770	5900	6037	5962.1	5831	5918	5869	6117	6325	6266.8	6025	6162	6105	6025	6162	6102
TA075	5467	5641	5778	5712.4	5516	5718	5601.6	5959	6198	6087.6	5768	5954	5855.2	5546	5679	5603.2
TA080	5845	5903	6069	6001.1	5903	5947	5910.2	6192	6302	6260.4	6054	6163	6116.6	5903	5918	5908.4
TA081	6202	6640	6821	6721.6	6505	6700	6560.3	6946	7175	7079.2	6826	7048	6934.4	6502	6541	6534
TA085	6314	6705	6848	6798.3	6578	6708	6625.5	7102	7198	7141.8	7001	7064	7029.2	6590	6695	6637.2
TA090	6434	6783	6974	6872.1	6645	6760	6706.7	7271	7358	7332.4	7004	7201	7079.2	6580	6783	6669.2
TA091	10862	11448	11584	11515	11293	11544	11478	11380	11564	11468	11211	11293	11262	10986	11544	11238.1
TA095	10524	11386	11447	11416.3	11092	11195	11132	11244	11409	11340.9	11084	11246	11157.6	10645	11244	11014.8
TA100	10675	11436	11555	11478.4	11207	11241	11225	11292	11514	11402.1	11196	11250	11219.2	10807	11292	11124
TA101	11195	12720	12838	12780.2	12427	12561	12509.2	12557	12826	12676.4	12356	12549	12483.4	12356	12549	12407.4
TA105	11259	12839	12975	12909	12473	12598	12541	12643	12871	12745.8	12417	12629	12524	11624	11685	11647.2

Acknowledgments

This work was supported by the Talent Introduction Special Fund of Anhui Science and Technology University under Grant No. ZRC2011304.

REFERENCES

[1] Gupta J.N.: A functional heuristic algorithm for the flowshop scheduling problem, *Operational Research Quarterly,* 22(1971), 39-47

[2] Koulamas C.: A new constructive heuristic for the flowshop scheduling problem, *European Journal of Operational Research*, 105(1998), 66-71

[3] Nawaz M., Enscore E., Ham I.: A Heuristic Algorithm for the m-Machine, n-Job flow-Shop Sequencing Problem, *Omega*, 11(1983), 91-95

[4] Tasgetiren M.F., Sevkli M., Liang Y.C., Gencyilmaz G.: Particle Swarm Optimization Algorithm for Permutation Flowshop Sequencing Problem, *LNCS*, 3172(2004), 366-385

[5] Yu B., Jiao B., Gu X.: An Improved Cooperative Particle Swarm Optimization and Its Application to Flow Shop Scheduling Problem, *Journal of East China University of Science and Technology (Natural Science Edition)*, 35(2009), 468-474

[6] Sun J., Feng B., Xu W.B.: Particle Swarm Optimization with Particles Having Quantum Behavior, *Proc. Congress on Evolutionary Computation*, 325(2004)

[7] van den Bergh F., Engelbrecht A.P.: A cooperative approach to particle swarm optimization, *IEEE Transactions on In Evolutionary Computation*, 8(2004), 1-15

[8] Li D., He Q.: A version of cooperative multi-swarm PSO using electoral mechanism to solve hybrid flow shop scheduling problem , *Przeglad Elektrotechniczny*, 88(2012), 22-26

[9] Coelho L.D.S: Novel Gaussian quantum-behaved particle swarm optimiser applied to electromagnetic design, *Science, Measurement & Technology*, IET 1(2007), 290-294

[10] Liu J., Sun J., Xu W.B.: Design IIR Digital Filters Using Quantum-Behaved Particle Swarm Optimization, *Advances in Natural Computation*, *LNCS,* 4222(2006), 637-640

[11] Yang X.S.: Engineering Optimization: An Introduction with Metaheuristic Applications, *John Wiley and Sons*, (2010)

[12] .Zhang J., Zhang C., Liang S.: A diversity-guided particle swarm optimizer-the ARPSO, *Expert Systems with Applications*, 37(2010), 5827-5834

[13] .Riget J., Vesterstroem J.: A diversity-guided particle swarm optimizer-the ARPSO, *Master's thesis*, Department of Computer Science, University of Aarhus, (2002)

[14] .Tasgetiren M.F., Liang Y.C.: A binary particle swarm optimization algorithm for lot sizing problem, *Journal of Economic and Social Research*, 5(2003), 1-20

[15] .Taillard E.: Benchmarks for basic scheduling problems, *European Journal of Operational Research*, 64(1993), 278-285

[16] .Lian Z., Gu X., Jiao B. : A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan, *Chaos, Solitons & Fractals*, 35(2006), 851-861

***Authors**: dr inż, Li Desheng, Anhui Science and Technology University, School of Science, ul. Donghua Road 9, 233100, Fengyang, Anhui, China, E-mail: ldsyy2006@126.com; dr inż, Deng Na, Hubei University of Technology, School of computer, ul. Li Jia Dun 1, Hong Shan District, 430068 Wuhan, China, email:* iamdengna@google.com.