XVII International Scientific Conference on Industrial Systems (IS'17)
Novi Sad, Serbia, October 4. – 6. 2017.
University of Novi Sad, Faculty of Technical Sciences,
Department for Industrial Engineering and Management
Available online at http://www.iim.ftn.uns.ac.rs/is17

Use of Heuristic Kalman Algorithm for JSSP

Ojstersek Robert
(Researcher, Faculty of Mechanical Engineering, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia, robert.ojstersek@um.si)

Zhang Hankun
(Researcher, School of Economics & Management, Beijing Jiaotong University, Beijing 100044, P.R. China, zhanghankun@bjtu.edu.cn)

Palcic Iztok
(Assoc. Professor, Faculty of Mechanical Engineering, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia, iztok.palcic@um.si)

Buchmeister Borut
(Full Professor, Faculty of Mechanical Engineering, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia, borut.buchmeister@um.si)

Abstract

This paper presents an effective use of well know prediction method of Heuristic Kalman Algorithm (HKA) for solving Job Shop Scheduling Problem (JSSP). Basic method of HKA is implemented on searching near optimal solution for JSSP. First, a mathematical model of the HKA was developed and tested with test data on 4-bencharks to test minimum makespan. A mathematical model of HKA is written in MATLAB software environment, which is responsible for prediction calculation of near optimum solution for JSSP. Secondly, results compressing and evaluation between our HKA and particle swarm optimization (PSO), the multi-phase particle swarm optimization (MPPSO) and the bare-bones particle swarm optimization (BBPSO) was made. Promising results shows that HKA can be used for solving different scheduling problems, like minimum needed number of workers (MNW), open vehicle routing problem and automated guided vehicle scheduling problems. Results show that the new method of implementing HKA for JSSP can predict near optimal solutions especially for low-dimensional cases in which our algorithm gives us the best results.

Key words: *Heuristic Kalman Algorithm, Job Shop Scheduling Problem, Optimization*

1. INTRODUCTION

The Kalman Algorithm (KA), as a well-known method, is usually implemented for problems related to prediction, like route planning, guidance, control of trajectory optimization and navigation. KA is also known as linear quadratic estimation algorithm, which makes measurements observed over time. The algorithm can predict unknown variables on premeasured simple's data. The basic method of KA works in two steps; in the first step of prediction, the algorithm estimates current state variables and update variables data using the weighted average. In the second step, algorithm present input variables and previously calculated weighted average state. Newly calculated variables represent the next step of using predicted results for execution. Execution gives more weight to higher certainty results, which run in real time, using only present input measurements, calculated average state and uncertainty matrix. Algorithm results are always corrupted with some random noise and numerical data error, especially when we use basic method of KA. That's why in current days, we can find some extensions and generalizations of KA method, such as the unscented KA (uses unscented transform of matrix function), extended KA (nonlinear version of KA) and HKA (used for differentiable systems). In this paper HKA is presented, this method shows the most positive results, which in our case is used and implemented for solving JSSP.

In this research work, we present the basic implementation of HKA for solving JSSP. In our research, the main objectives are:

- Modeling and programing the HKA suitable for solving JSSP using MATLAB software,
- Testing newly made HKA on 4-benchmarks,
- Comparing results with four existing population-based stochastic methods.

Results will represent by numerical experiments of our algorithm on four test benchmarks. Results compressing will show advantages and disadvantages according to the existing population-based stochastic methods, particle swarm optimization (PSO), the multi-phase particle swarm optimization (MPPSO) and the bare-bones particle swarm optimization (BBPSO). The contribution consists a newly created HKA for solving

JSSP. Experimental results are compared to the already existing methods for getting better knowledge about HKA, needed for further research of solving scheduling problems, such as the traveling salesman problem (TSP) and the vehicle routing problem (VRP).

2. LITERATURE REVIEW

Toscano et al. [1] present a new optimization method of HKM, as an alternative approach for solving continuous, non-convex optimization problems. The algorithm has measurement process designed to give an estimated optimum. Measurements and data collection of an existing already done experiment and data of an experiment in real time is a basic principle of KA used for prediction. Application example was introduced by Toscano et al. [2] implement HKA for the design of a robust flux estimator and the optimal design of no-chip spiral inductors. Pakrashi et al. [3] use HKA for partition data clustering. They present new improved approach of HKM-K, which has benefits of fast convergence to the global optimum. Maranakis et al. [4] proposed the new hybrid method of Particle Swarm Optimization (PSO) for solving Open Vehicle Routing Problem which is suitable for solving the large-scale problem within short computational time. All basic methods for solving JSSP are represented in a book written by Pinedo [5], which in all cases represents the foundation for further research work to implementing HKA for JSSP. Newly created algorithms are tested on benchmarks examples Perez [6], on which the results of new algorithm can be tested and compared to the existing one. Our HKA for solving JSSP is tested on 4 benchmarks, J3M4 [5], J6M6 [6], J10M10 [6] and J20M5 [6]. Eberhart et al. [7] presented the use of benchmarks testing between PSO and both artificial life and evolutionary computation. Shi et al. [8] present how the implementation of the new parameter (inertia weight) influences on PSO algorithm. Zhang et al. [9] in paper proposes a new bare-bones multi-objective PSO algorithm to solve the dispatch problems, the algorithm is capable of generating an approximation of the Pareto frontier. PSO algorithms are also used for geophysical inverse problems, where Poormirzaee et al. [10] shows that PSO is a suitable method for investing microtremor waves. Hybrid Genetic Algorithm (GA) and PSO algorithm for multi-objective Automated Guided Vehicle (AGV) scheduling is presented by Mousavi [11], the algorithm is capable of scheduling AGV for transporting materials within a manufacturing facility or a warehouse. Proposed method obtain less mean computational time as benchmark method. The multiphase PSO algorithm, which is a basic method in mentioned hybrid algorithms, is presented in the Ph.D. thesis of Al-Kazemi [12]. Zhang et al. [13] also presented multi-phase PSO algorithm in case for solving break-even distance of railway transportation. In another case, Tang et al. [14] use the same method for solving the bulk cargo port scheduling problem. As mentioned before [9], also Kennedy [15] present bare-bones PSO basic method and compared to the other stochastic population-based methods. In a lot of papers rescuers test their proposed algorithm on mathematical models and simulation real world applications, as Zapciu et al. [16] proposed algorithm for production systems flow modeling using decomposition method and required buffers size, they test the mathematical model of Markov chains with the discrete system simulation model.

3. HEURISTIC KALMAN ALGORITHM

The Heuristic Kalman Algorithm (HKA), as a Kalman Filtering based heuristic approach, has been proposed for solving continuous and non-convex optimization problems, which is only needed to set a small number of parameters by the user (only three) [1,2]. Although it belongs to the so-called "population based stochastic optimization techniques", the HKA search heuristic is entirely different from others, which explicitly considers the optimization problem as a measurement process designed to give an estimate of the optimum [1,2]. Through the measurement process, a specific procedure based on the Kalman estimator was developed to improve the quality of the estimate obtained [1,2]. The HKA is first tested by several unconstrained and constrained non-convex test problems and has a comparative advantage in terms of computational time and success rate [1]. Then the HKA has been applied in the design of a robust flux estimator of an induction machine and the optimal design of on-chip spiral inductors [2]. Pakrashi and Chaudhuri first employ the HKA into partitional data clustering and combines the benefits of global exploration of HKA and the fast convergence of K-Means method to obtain good clustering in a reasonable amount of time [3]. The practical implementation of the HKA requires properly initializing the Gaussian distribution, selecting the user-defined parameters and introducing a stopping rule [1]. During the optimization process of the HKA, the solution is first generated according to the Gaussian distribution parametrized by a given mean vector with a given variance–covariance matrix, then a measurement procedure is followed, and finally, an optimal estimator of the parameters of the random generator is introduced. The general procedure of HKA shown in Algorithm 1 [1,2].

Algorithm 1 Pseudo-code of HKA
Step 0 Initialization. Set the number of population size N, the number of top individuals under consideration N_ξ the slowdown coefficient α and the maximum number of iterations $MaxIter$. Initialize the current iteration $ite = 0$, the mean m and the variance–covariance vector S: $m = \left[\frac{\overline{x}_1+\underline{x}_1}{2}, \cdots, \frac{\overline{x}_d+\underline{x}_d}{2}\right]^T, S = \left[\left(\frac{\overline{x}_1-\underline{x}_1}{6}\right)^2, \cdots, \left(\frac{\overline{x}_d-\underline{x}_d}{6}\right)^2\right]^T$ where $\overline{x}_i$ (respectively, $\underline{x}_i$) is the i^{th} upper bound (respectively, lower bound) of the problem and d is the dimension of the problem. Step 1 iteration. for $ite = 1: MaxIter$ Step 2.1 Random generator. Generate a population x with N individuals by Gaussian distribution: $x = mvnrnd(m, diag(S), N)$ where $mvnrnd(.)$ is a function that generates random vectors from the multivariate normal distribution and $diag(.)$ is a function that generates diagonal matrices or diagonals of a matrix.

Step 2.2 Measurement process. Calculate the individual fitness f in x, choose the top N_ξ individuals according to f, compute the measurement ξ and the variance vector V:

$$\xi = \frac{1}{N_\xi}\sum_{j=1}^{N_\xi} x_j,\ V = \frac{1}{N_\xi}\left[\sum_{j=1}^{N_\xi}\left(x_{j,1}-\xi_1\right)^2, \cdots \sum_{j=1}^{N_\xi}\left(x_{j,d}-\xi_d\right)^2\right]^T$$

Step 2.3 Optimal estimation. Compute the posterior estimation the mean m_pe and the variance–covariance matrix S_pe:

$$L = S./(S+V)\ ,\quad W = (S - L.*S)^{0.5}\ ,\quad m\_pe = m + L.*(\xi - m)\ ,\quad \tau = min\left(1, mean\left(\sqrt{V}\right)^2\right)\ ,\quad a = \alpha\tau/(\tau + max(W)),$$

$$S\_pe = \left(S^{0.5} + a\left(W - S^{0.5}\right)\right)^2$$

where a is the slowdown factor, $mean(.)$ is a function that calculates the average or mean value and the symbol ./ (respectively, .*) stands for a componentwise divide (respectively, product).

Step 2.4 Initialization of the next step:

$m = m\_pe$, $S = S\_pe$

end

4. NUMERICAL EXPERIMENTS

4.1 **Encoding**

The HKA was originally proposed for continuous optimization problems, but the JSSP is a well-known combinatorial optimization problem, so this paper employs the relative position indexing[4]. In order to use a population-based continuous optimization algorithm, each element of the solution is converted to a floating point that randomly initializes in the open interval $(0,1)$ (see Fig 1 S0.). The optimized solution is transformed into the discrete domain by using the relevant position index (see Fig 1 S1.). Then the processing order sequence of the job can be got by the processing time table (see Fig 1 S2.). In Fig 1 S2, the operation processing order is as follows: (2, 2), (1, 1), (2, 1), (1, 2), (1, 3), (2, 3), which (., .) means job number and machine number.

no	jobs	machine sequence	processing time
1	1	1	10
2	1	2	8
3	1	3	4
4	2	2	8
5	2	1	3
6	2	3	6

S_0 {0.8147, 0.9058, 0.1270, 0.9134, 0.6324, 0.0975}

$\downarrow$

S_1 | 6 | 3 | 5 | 1 | 2 | 4 |

$\downarrow$

S_2 | 2 | 1 | 2 | 1 | 1 | 2 |

Figure 1. Example of the solution encoding.

4.2 **Experiment**

This paper chooses J3M4 [5], J6M6 [6], J10M10 [6] and J20M5[6] as the benchmark instances of the JSSP. Their optimal solutions of the benchmark instances have makespan of 28, 55, 930 and 1165 respectively.

For comparison, the 4-benchmark instances were also solved by PSO, the MPPSO and BBPSO. The PSO is proposed by Eberhart and Kennedy [7], and improved by Shi and Eberhart (the PSO of this paper represents the improved version) [8]. The PSO is successfully used to solve many practical problems such as the open vehicle routing problem, the geophysical inverse problems and the automated guided vehicle scheduling problems [9–11]. The MPPSO is proposed by Al-Kazemi and applied to the problems such as the break-even distance of railway freight transportation and the bulk cargo port scheduling problem [12–14]. The BBPSO is proposed by Kennedy and applied to solve the environmental/economic dispatch problems [9,15].

Algorithms were implemented in MATLAB language. All experiments are simulated in MATLAB version R2016b. The algorithms are independently run 30 times for each instance.

4.3 **Result**

According to [1,3] and experiment, the parameter for the HKA is set as $N = 100$, $N_\xi = 10$, $\alpha = 0.3$ and $Maxlter = 500$. The parameter for the PSO is set as $N = 50$, $c_1 = 2.8$, $c_2 = 1.3$, $w = 0.729$ and $Maxlter = 1000$ [10]. The parameter for the MPPSO is set as $N = 50$, $ph = 2$, $pcf = 5$, $g = 2$, $sllu = [1, min(10, d)]$, $VC = 10$ and $Maxlter = 500$ [12]. The parameter for the BBPSO is set as $N = 50$ and $Maxlter = 1000$.

Fig 2 shows the four algorithms solution convergence for the benchmark instances. For the J3M4 (see Fig 2 a-d.), all four algorithms can get the optimal solution at the time of initialization. For the J6M6 (see Fig 2 e-h.), all four algorithms tend to be optimal solutions. In addition, the MPPSO and the BBPSO can converge to the optimal fitness. For both the J10M10 (see Fig 2 i-l.) and the J20M5 (see Fig 2 m-p.), all 4 algorithms can tend to be optimal solutions but none of them can converge to optimal fitness.

The statistical analysis of the four algorithms for the 4-benchmark instances is shown in Fig 3. As shown in Fig 3 a, all four algorithms can always obtain the optimal value of the J3M4. The performance of the MPPSO and the BBPSO is more stable than others for the J6M6 (see Fig 3 b.). For both the J10M10 (see Fig 3 c.) and the J20M5 (see Fig 3 d.), the MPPSO performs best, followed by the HKA, then the BBPSO, and the last one is the PSO.

The computational statistics of the fitness of the four algorithms for the 4-benchmark instances shown in Table 1 and 2. The success rate of all algorithms for the J3M4 is 100 %. For the J6M6, the success rate of the MPPSO and the BBPSO is still 100 %, but the HKA and the PSO are only 43 % and 17 %, respectively. Not all algorithms can find the optimal makespan, for both the J10M10 and the J20M5.

All 4 algorithms perform well in low-dimensional of the 4 benchmark instances. However, with the dimension increases, the performance of all algorithms decreases. The robustness of the MPPSO is the best, which can perform better in all 4-benchmark instances than others. The BBPSO has a good performance on low-dimensional benchmark instances, but its performance has decreased significantly with the dimension of the benchmark instances increases. The HKA has a relatively slow descent performance with the dimension of the benchmark instances increases. The PSO shows the worst performance, its performance significantly decreased with the dimension of the benchmark instances increased.

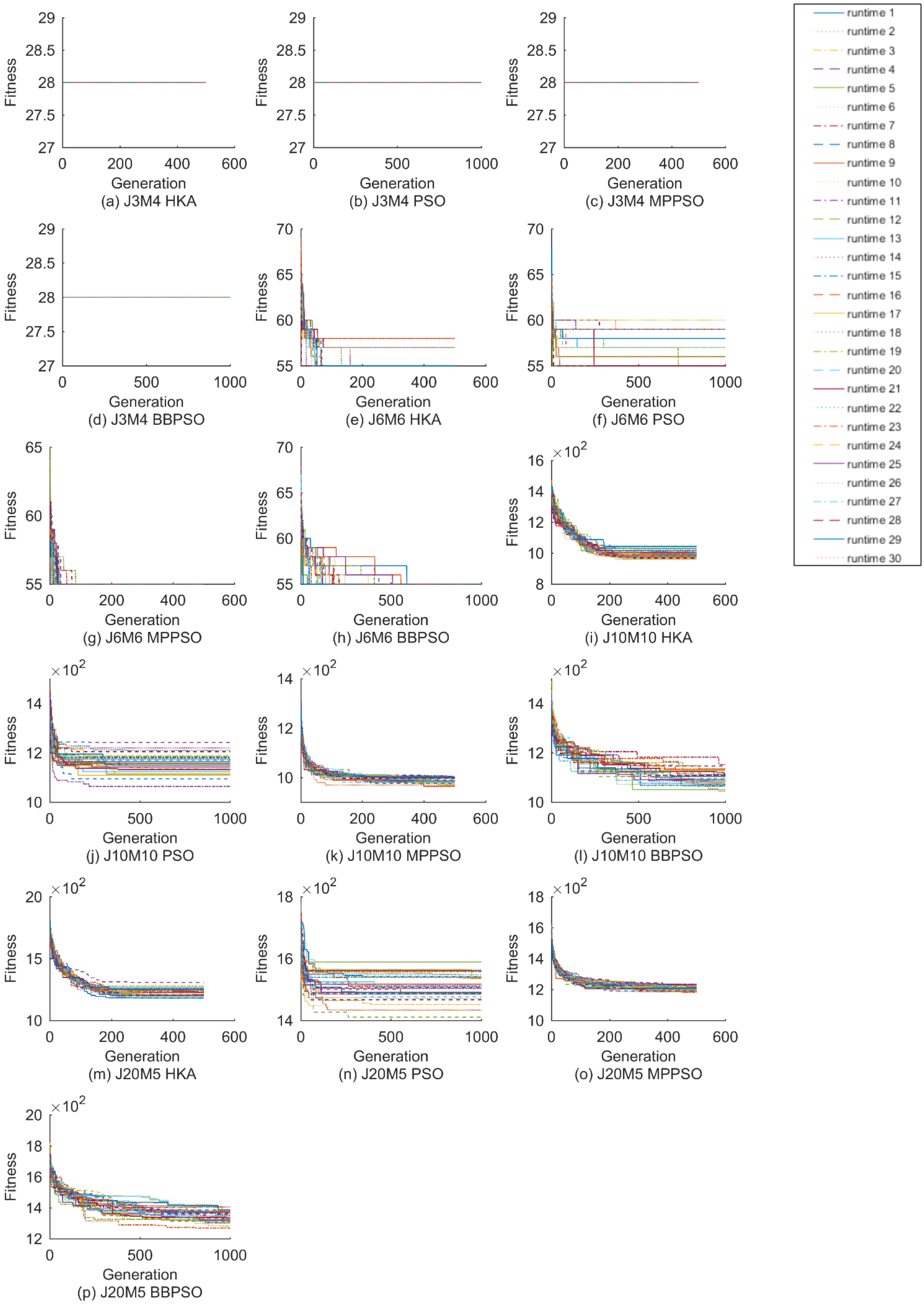

Figure 2. The four algorithms convergences of the best solutions for the 4-benchmark instances.

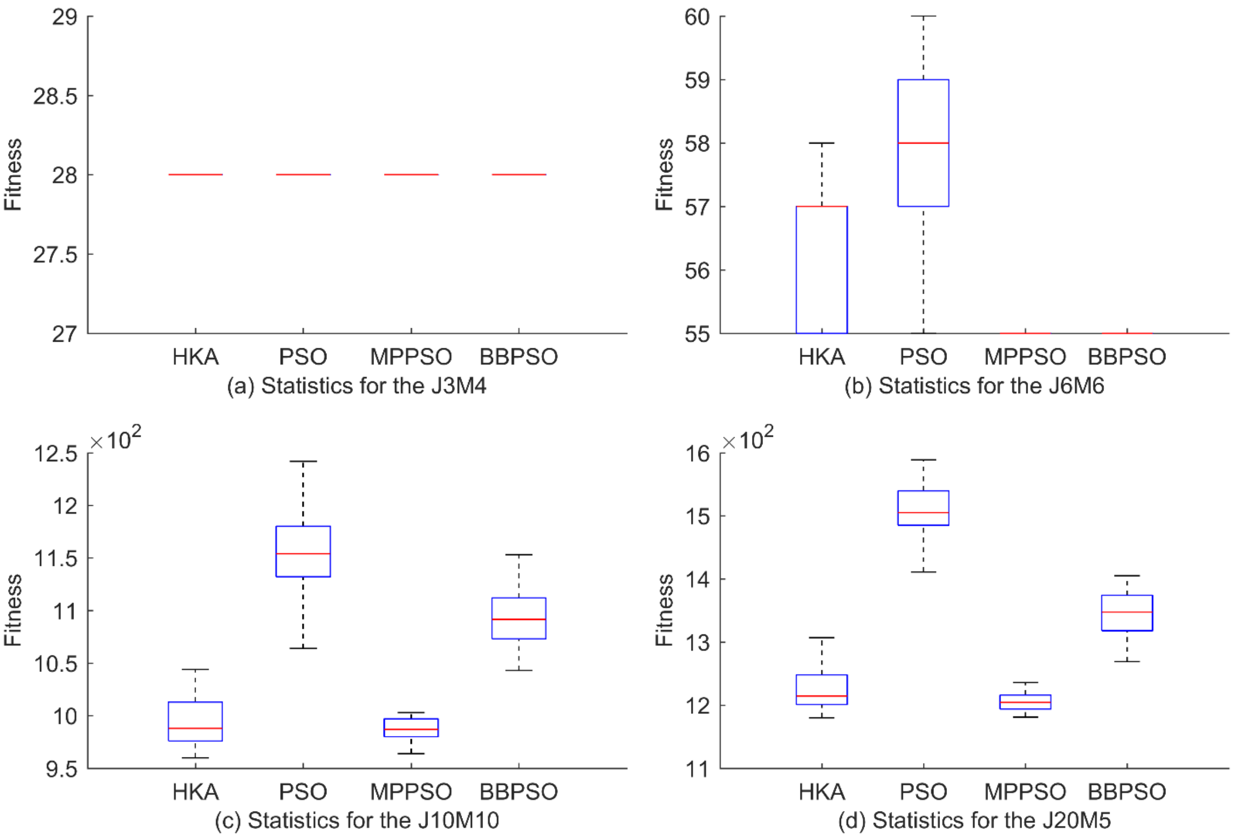

Figure 3. Statistical analysis of the four algorithms for the 4-benchmark instances.

Table 1. Computational statistics of the four algorithms on the fitness for the J3M4 and J6M6.

name	J3M4					J6M6				
	Min	Max	Mean	Standard deviation	Success rate (%)	Min	Max	Mean	Standard deviation	Success rate (%)
HKA	28	28	28	0	100	55	58	56	1	43
PSO	28	28	28	0	100	55	60	58	2	17
MPPSO	28	28	28	0	100	55	55	55	0	100
BBPSO	28	28	28	0	100	55	55	55	0	100

Table 2. Computational statistics of the 4 algorithms on the fitness for the J10M10 and J20M5.

name	J10M10					J20M5				
	Min	Max	Mean	Standard deviation	Success rate (%)	Min	Max	Mean	Standard deviation	Success rate (%)
HKA	960	1044	995	25	0	1180	1307	1224	31	0
PSO	1064	1242	1155	38	0	1411	1589	1505	42	0
MPPSO	964	1003	986	11	0	1181	1236	1206	15	0
BBPSO	1043	1153	1095	28	0	1269	1405	1344	35	0

5. CONCLUSION

In this paper, the HKA is used to solve the JSSP. Four benchmark instances of the JSSP are selected to test the algorithm. And 3 population-based continuous optimization algorithms were selected as a comparison. the HKA performs well in low-dimensional problems but decreases with the problem dimension increases. And its performance is always no better than the MPPSO in all 4 benchmark instances. However, in the high-dimensional problem, the performance of the HKA is better than both the PSO and the BBPSO. Computational results and comparisons showed that the HKA can be used to solve the JSSP, especially in low-dimensional.

The future research is how to enhance the optimization ability of the HKA, especially in high dimensional. In addition, we can try to apply the HKA to other combinatorial optimization problems such as the traveling salesman problem (TSP) and the vehicle routing problem (VRP).

6. ACKNOWLEDGEMENT

The authors acknowledge the financial support from the Slovenian Research Agency (ARRS).

7. REFERENCES

[1] Toscano, R. and Lyonnet, P. (2010), "*A new heuristic approach for non-convex optimization problems*", Information Sciences, Vol. 180, No. 10, pp. 1955–1966.

[2] Toscano, R. and Lyonnet, P. (2012), "*A Kalman Optimization Approach for Solving Some Industrial Electronics Problems*", IEEE Transactions on Industrial Electronics, Vol. 59, No. 11, pp. 4456–4464.

[3] Pakrashi, A. and Chaudhuri, B. B. (2016), "*A Kalman filtering induced heuristic optimization based partitional data clustering*", Information Sciences, Vol. 369, pp. 704–717.

[4] Marinakis, Y. and Marinaki, M. (2012), "*A Hybrid Particle Swarm Optimization Algorithm for the Open Vehicle Routing Problem*", Dorigo, M., Birattari, M., Blum, C., Christensen, A. L., Engelbrecht, A. P., Groß, R. and Stützle, T. (Ed.), Swarm Intelligence: 8th International Conference, ANTS 2012, Brussels, Belgium, September 12-14, 2012. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 180–187.

[5] Pinedo, M. L. (2009), "*Machine Scheduling and Job Shop Scheduling*", Planning and Scheduling in Manufacturing and Services, Springer New York, New York, pp. 83–115.

[6] Pérez, E. (2011), "*JOB SHOP SCHEDULIND PROBLEM: BENCHMARKING INSTANCES*", available at: https://www.eii.uva.es/elena/JSSP/InstancesJSSP.htm (accessed: 7 June 2011).

[7] Eberhart, R. and Kennedy, J. (1995), "*A new optimizer using particle swarm theory*", MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, Nagoya, pp. 39–43.

[8] Shi, Y. and Eberhart, R. (1998), "*A modified particle swarm optimizer*", 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), IEEE, Anchorage, pp. 69–73.

[9] Zhang, Y., Gong, D.-W. and Ding, Z. (2012), "*A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch*", Information Sciences, Vol. 192, pp. 213–227.

[10] Poormirzaee, R. (2016), "*S-wave velocity profiling from refraction microtremor Rayleigh wave dispersion curves via PSO inversion algorithm*", Arabian Journal of Geosciences, Vol. 9, No. 16, pp. 673.

[11] Mousavi, M., Yap, H. J. and Musa, S. N. (2017), "*A Fuzzy Hybrid GA-PSO Algorithm for Multi-Objective AGV Scheduling in FMS*", International Journal of Simulation Modelling, Vol. 16, No. 1, pp. 58–71.

[12] Al-Kazemi, B. S. N. (2002), "*Multiphase particle swarm optimization*", Ph.D. dissertation, Syracuse University, New York.

[13] Zhang, H., Liu, S. and Gong, D. (2016), "*Multi-phase Particle Swarm Optimization on the Break-even Distance of Railway Freight Transportation*", Journal of the China Railway Society, Vol. 38, No. 3, pp. 18--26.

[14] Tang, M., Gong, D., Liu, S. and Zhang, H. (2016), "*Applying multi-phase particle swarm optimization to solve bulk cargo port scheduling problem*", Advances in Production Engineering {&} Management, Vol. 11, No. 4, pp. 299–310.

[15] Kennedy, J. (2003), "*Bare bones particle swarms*", Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706), IEEE, Indianapolis, pp. 80–87.

[16] Zapciu, M., Cavalieri, S. and others (2017), "*Production Systems Flow Modelling Using Decomposition Method and Required Buffers.*", International Journal of Simulation Modelling (IJSIMM), Vol. 16, No. 2.