

INFLUENCE OF EA CONTROL PARAMETERS TO OPTIMIZATION PROCESS OF FJSSP PROBLEM

Koblasa, F.*; Kralikova, R.** & Votrubic, R.*

*Technical University in Liberec, Faculty of Mechanical Engineering, Department of Manufacturing Systems and Automation, Studentská 2, 461 17 Liberec 1, Czech Republic

** Technical University in Košice, Faculty of Mechanical Engineering, Department of Process and Environmental Engineering, Park Komenského 5, 042 00 Košice, Slovak Republic

E-Mail: frantisek.koblasa@tul.cz, ruzena.kralikova@tuke.sk, radek.votrubic@tul.cz

Abstract

The ability of Evolution Algorithms (EA) to find an optimal solution is usually given by various algorithm operators. Population size and a maximal number of generations are usually set base on available timespan. Setting selection and elimination methods together with crossover probability are usually based on intuition and sometimes are problem specific. That is the reason presented research is focusing on the approach of how to set elimination methods and crossover probability by the statistical approach to eliminate the necessity of experience and intuition. This article describes the scheduling model together with the used EA to solve the Flexible Job Shop Scheduling Problem (FJSSP). Statistical process control methods are applied as there is a designed experiment to find out the statistical significance of each parameter during solving one of the FJSSP hardest problems. Crossover and elimination statistical importance are analysed and suitable levels of them are suggested. The statistical approach as a possible methodology to set the mentioned parameters is then discussed.

(Received in April 2020, accepted in August 2020. This paper was with the authors 2 weeks for 1 revision.)

Key Words: Evolution Algorithms, Flexible Job Shop Scheduling Problem, Parameter Control, Statistical Process Control

1. INTRODUCTION

The implementation of advanced scheduling and optimization techniques are besides academic outcomes rare. Computer simulation software was one of few able to practically solve complex scheduling models [1]. Usual computer simulation software has the main advantage in the ability to simulate wide types of manufacturing systems as job shops (flexible, hybrid). The major disadvantage of industry-oriented simulation is the inability to generate active schedules as they use mostly only non-delay schedules [2] by selecting optimal operation from the set of available operations that are ready to be processed. Nowadays, with outgoing 4.0 industrial revolution, scheduling models of complex manufacturing problems as FJSSP are implemented to Manufacturing Execution Systems (MES) [3]. That opens an opportunity to implement metaheuristic optimization approaches as well as the opportunity to find optimal schedules.

There are enormous techniques how to construct and optimize schedule beginning with constructive algorithms which use dispatching rules [4, 5], over searching strategies as tabu search [6] to hybrid algorithms combining evolution principles and searching strategies [7-9]. Popular natural cooperation based behaviour algorithms take the optimization principles from observing an intelligent group of ants (ACO) [10], bees [11] (ABC), bats (BA) [12], grey cuckoos (CS) [13] and even improvising musicians (HS) [14] or intelligent water drops (IWD) [15] to solve combinatorial problems. Agent systems become frequently used as a requirement for autonomous decision making arise [16, 17].

Genetic algorithm has proven to be one of the most effective and popular evolutionary techniques for solving scheduling problems as FJSSP [18]. According to the reviews on heuristic optimization techniques [19, 20], more than 20 % of the research studies focussed on

scheduling problems have been conducted using EA related algorithms. That is the highest percentage of any artificial intelligence-based technique.

The major disadvantage of optimization algorithms and particularly of EA from the practical point of view is the necessity to set its operators to find the best possible and feasible solution in a reasonable time. This requires knowledge about mechanisms and interactions between inner mechanisms. There are in general two main approaches to make optimization algorithm for user seemingly parameterless [21, 22]. Parameter tuning [23] has a goal to find the best possible setting which does not change during optimization while parameter control [24] starts with initial parameters setting and changes during optimization. That can be done by on the fly changing parameters of the whole optimization process by adaptive control or parameters evolve as a part of the individual solution (chromosome) by self-adaptive technique.

Adaptive and self-adaptive techniques have a great advantage in possible changing parameters according to the explorative or exploitative phase of the search [24, 25], however, it is not desirable due synergy and influence of parameters to adapt every evolution operator. This leads to a problem of sequentially selecting and testing parameters which can depend on coding [26] and are not generally known for being critical for both optimization time and quality of the solution. That includes most of the parameters [27] except population size and number of generations which are generally suitable for adaptive control [24].

The main idea is to identify and set parameters which suites for tuning and control of particular EA to make the optimization method accessible for common user.

The general approach of searching suitable parameters is taken [23], however with additional steps to find influence and synergy of selected parameters rather than just using a statistical approach to find if newly developed approach generates significantly better solution than others.

This research aims to analyse the behaviour of the presented EA, analyse influence and interaction of selected parameters to identify the way of improvements and to discuss the possible general approach to tune parameters base on statistical approach control in scheduling environment.

To address this problem this paper is using and describing the Flexible Job Shop Scheduling Problem (FJSSP) in the following chapter. The third chapter is describing the modified Random Key-based Evolution Algorithm (RKEA) with parameter setting in crossover and developed parent selection operators. This algorithm is then used as an example of setting parameters base on a statistical approach to find the best objective function of makespan of one of the hardest FJSSP models given by Brandimarte [28]. The RKEA algorithm with selected setting is compared with recently published metaheuristic algorithms as well as with simple evolution algorithm to discuss its limitations and possible improvements base on the statistical approach.

2. FJSSP DEFINITION AND MODEL

FJSSP problem is known NP-hard problem which consists of two decision problems of assigning operations to machines (assignment) and to determine the processing order of jobs on machines (sequencing). Each job $J = \{J_1, \dots, J_n\}$ consists of a sequence of operations $O = \{O_{i1}, \dots, O_{in}\}$ where they are allowed to be processed on any among a set of feasible machines M_{ik} . All n jobs and m machines are in theoretical problems available at time 0, and a machine can only execute one operation at a given time. Pre-emption is not allowed.

Fig. 1 shows the flow of jobs on machines $M_{n,p,i}$, where n indicates the number of job, $p(q)$ workshop and i machine on given workshop where the job is processed.

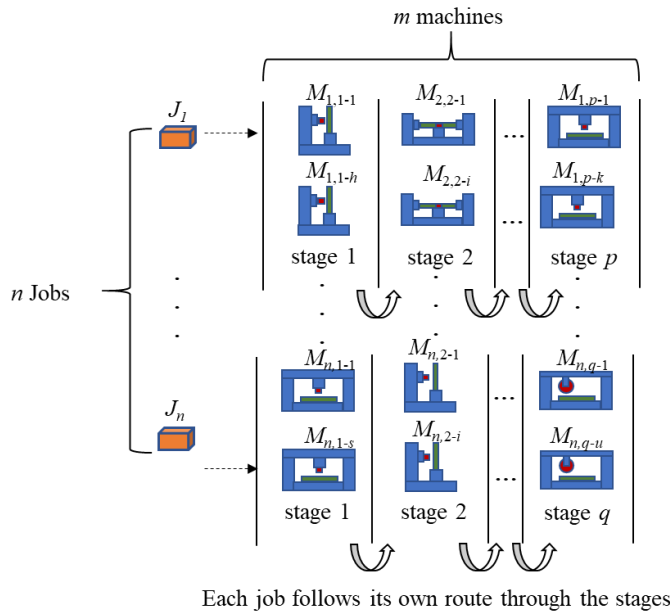


Figure 1: Model of FJSSP problem.

Despite FJSSP being mostly referenced as a multiobjective problem [20, 29] for this research single most used objective of minimizing makespan $f(x) = C_{max}$, Eq. (1) [20, 30] was considered to analyse the impact of presented EA operators without the interference of other objectives. Eqs. (2) and (3) are describing the operating sequence constraint. The constraint Eq. (4) guarantees machine allocation so each operation can be only process on one machine from machine set at one time. The constraints Eqs. (5) and (6) are nonnegative or 0 – 1 binary variable which are restrictions on decision variables, respectively.

$$C_{max} = \max\{c_{in_i}\} \quad (1)$$

$$\text{s.t. } c_{ik} - c_{i(k-1)} \geq t_{ikj}x_{ikj}, k = 2, \dots, n_i, \forall i, j \quad (2)$$

$$[(c_{hg} - c_{ik} - t_{hgj})x_{hgj}] \geq 0 \vee [(c_{ik} - c_{hg} - t_{ikj})x_{ijk} \geq 0] \forall i, j, g, h \quad (3)$$

$$\sum_{x_{ikj} \in A_{ik}} x_{ikj} = 1 \forall i, k, j \quad (4)$$

$$c_{ik} \geq 0, \forall i, k \quad (5)$$

$$x_{ikj} \in \{0,1\}, \forall i, k, j \quad (6)$$

Job indexes are noted as i and h ; $i, h = 1, 2, \dots, n$; j is machine index, $j = 1, 2, \dots, m$; k, g are operation indexes, $k, g = 1, 2, \dots, n_i$ (where n_i is a total number of operations of job i). Processing time of k^{th} operation of job i is t_{ikj} which leads to ending time c_{ik} of operation O_{ik} . Logic variable x_{ikj} stand for machine j is selected for O_{ik} .

3. RANDOM KEY BASED EVOLUTIONARY ALGORITHM

Used random key evolution algorithm (RKEA) is a general search metaheuristic proposed by Bean [31] and heavily improved by [32] for solving combinatorial optimization problems involving sequencing. The chromosomes of an RKEA are represented as vectors of randomly generated real numbers in the interval $[0, 1]$. Decoder than transforms solution vectors in a solution of the combinatorial optimization problem for which an objective value or fitness has to be computed. The proposed algorithm has several differences from the classical RKEA approach to chromosome manipulation (Fig. 2).

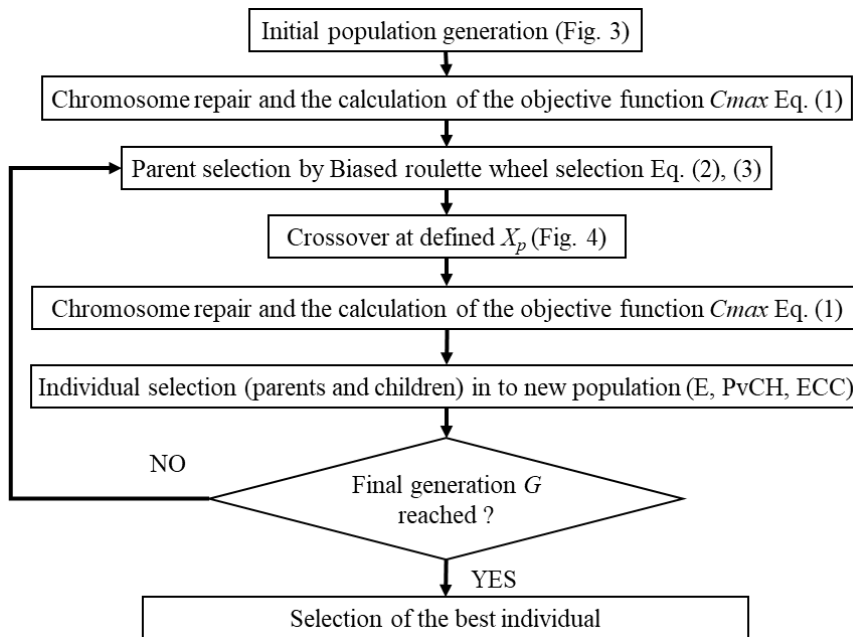


Figure 2: Random key representation of FJSSP problem individual.

Absence of mutation operator to maintain diversity in the population is substituted by chromosome repair mechanism, which is used every time the new solution is evaluated. That includes both initial population generation as well as individuals created by crossover.

In used RKEA the population of random key vectors is evolved over a number of iterations (generations) G . The initial population is made up of vectors (individuals – see Fig. 3) of random keys (vector A – Operation identification) representing the sequence of operations and vector of integers representing operation machine assignment (vector B – Machine identification).

During evaluating it can frequently happen that operation sequencing (chromosome A) or job assigning (chromosome B) defined by active schedule (or non-delay) scheme generation does not match a defined solution by both chromosomes. The repaired solution is then made by randomly selecting one of the available operations in conflict set generated by schedule scheme putting this operation at the correct place in chromosome shifting the rest of the chromosome vector as it is to maintain originally proposed solution as much as possible. B chromosome is then in the case of repaired operation changed according to machine assigned by schedule generation.

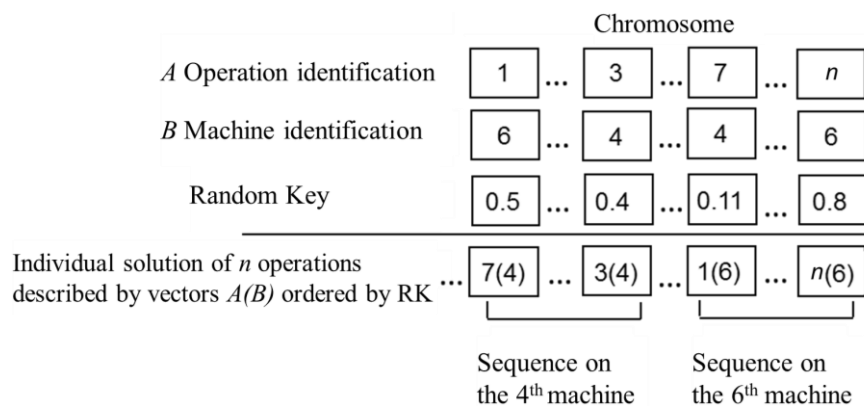


Figure 3: Random key representation of FJSSP problem individual.

After generating the initial population and calculating the objective function (minimizing makespan) algorithm select individuals to create a new solution. There was used modified

biased roulette wheel principle with probability P Eq. (7) given by the quality of objective function $f(x)$ of the individual x recalculated in the fitness function $O(x)$ as it is necessary for minimization problem while using roulette wheel.

$$P(x_i) = \frac{O_{bw}(x_i)}{\sum_{j=1}^N O_{bw}(x_j)} \text{ where } O_{bw}(x_i) = \frac{f(x_i)}{f_{\min}} \tag{7}$$

This principle is known for causing stagnation in population e.g. fast convergence to a local optimum. That is why Eq. (7) is used only to select the first parent with the ascending probability with better solutions. The second parent is selected with increasing probability with the worst value of fitness and also objective function Eq. (8).

$$P(x_i) = \frac{O_{wb}(x_i)}{\sum_{j=1}^N O_{wb}(x_j)} \text{ where } O_{wb}(x_i) = \frac{f(x_i)}{f_{\max}} \tag{8}$$

Selected individuals are then used to reproduce. That is done by proportional uniform crossover where crossover probability Xp defines the chance that parents will exchange genes to construct new offspring (child). Fig. 4 shows two exchanges in gene at crossover probability $Xp = 0.2$.

		Chromosome																			
Parent 1 Random key		0,55	0,34	0,86	0,27	...	0,44	0,65	0,91	0,12	0,23	0,7	0,35	0,64	...	0,8	0,37	0,1	0,58		
Machine		1	5	4	8	...	2	8	6	4	3	8	5	9	...	1	4	5	4		
Parent 2 Random key		0,6	0,3	0,81	0,45	...	0,53	0,95	0,7	0,61	0,55	0,34	0,35	0,27	...	0,44	0,37	0,91	0,12		
Machine		1	5	5	8	...	2	4	6	4	0,23	0,7	0,86	0,64	...	0,8	0,65	0,1	0,58		
Cross. rate prob. check		3	8	4	9	...	1	8	5	4	Child 1 Random key		0,55	0,34	0,35	0,27	...	0,44	0,37	0,91	0,12
		1	5	5	8	...	2	4	6	4	Machine		1	5	5	8	...	2	4	6	4
		3	8	4	9	...	1	8	5	4	Child 2 Random key		0,23	0,7	0,86	0,64	...	0,8	0,65	0,1	0,58
		3	8	4	9	...	1	8	5	4	Machine		3	8	4	9	...	1	8	5	4

Figure 4: Random key representation of FJSSP problem.

After generating new individuals (equal to population size), the objective function (makespan) is calculated to evaluate the optimization process. The last step is to select surviving solutions to next generations.

This research is considering the following strategies where Elite is considered as classic elimination and Parent versus child and Elite modification with clone control is developed by authors:

- Elitist (E) – both populations (parents and children) are sort by objective function (as the goal is to minimize makespan – ascending) and better half survives to next generation.
- Parent versus child (PvCH) – Children are challenging their parents. If succeed they replace their parents, else they do not continue to the next generation. The tournament is set between male and female individuals where identification is given with respect of crossover probability. That means chromosomes with higher probability to be similar challenges each other (parent 1 vs child 1 and parent 2 vs child 2, see Fig. 4).
- Elitist with clone control (ECC) – same as E, however selection checks for the similarity of individual and if there are identical individuals (clones) in the terms of the value of objective function they are substituted by not successful individuals. This mechanism designed to keep the population more diverse so premature convergence will be prevented.

4. EXPERIMENT DESIGN

The goal of the designed experiment is to find the influence of crossover probability and selection strategy to optimization time and value of the objective function (total competition time – makespan).

To test mentioned influence RKEA (Fig. 2) parameters were set as follows:

- Population size (number of individual solutions in population) $N = 700$,
- Number of generations (iterations of the algorithm) $G = 200$,
- Crossover rate (coefficient) $Xp = 10\%$, 20% , 30% , 40% , (i.e. 0.1, 0.2, 0.3, 0.4),
- Elimination strategy = E, PvCH, ECC.

That means a total of 12 combinations (Crossover rate and selection strategy) with 10 repetitions were tested (120 experiments). This experiment was done on well known (and one of the hardest) FJSSP problems – MK10. MK10 is selected based on the difficulty to reach its optimum and its size among Brandimarte sets [28] with the assumption that if the found setting will be suitable for the difficult problem it will be also for less difficult one. MK10 represent model where is necessary to schedule 20 jobs with the maximal number of 15 operations that can be assigned to maximal 5 different (2.6 in average) but interchangeable machines with different operation processing time. The total pool includes 15 machines. The optimal value of makespan objective function is 165.

5. STATISTICAL ANALYSIS AND PARAMETER SETTING

The first set of tests focused on population convergence to one solution with the goal to find out how crossover probability influences the objective function of makespan as well as the convergence of population to one solution (local extreme). For that reason, 4 rates of crossover probability were tested on simple Elitist elimination strategy, which does not contain any mechanism to avoid premature convergence. Graph (see Fig. 5) presents a moment in which population of each of 10 repetitions converges to one local extreme.

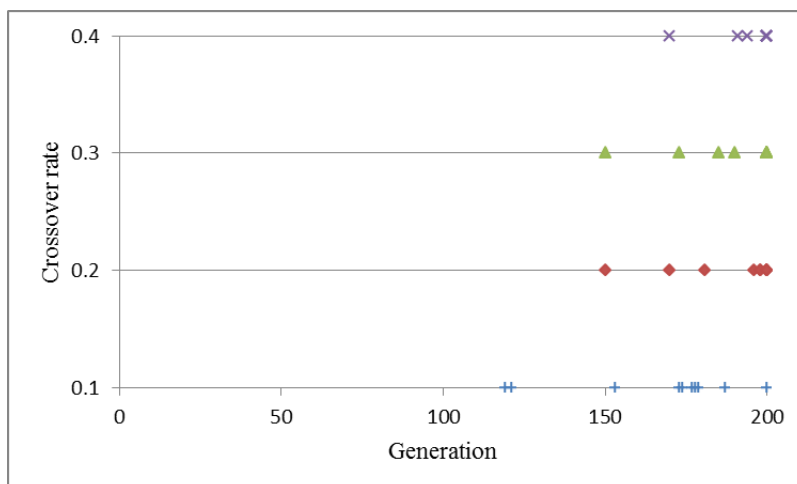


Figure 5: Population convergence to local extreme.

Used elitist strategy, in which the number of new individuals is equal to the population size and better half from new individuals and old generation survives, shows its problematic performance as 9 tests of $Xp = 0.1$ crossover probability (i.e. 10 %) end prematurely because of total convergence to the local extreme. With a higher crossover probability number of prematurely converged tests have a decreasing tendency ($Xp = 0.2 - 6$ individuals, $0.3 - 4$, $0.4 - 3$ converges). On the other hand, $Xp = 0.1$ (see Table I) has best results in all minimal,

maximal and average values of $f(x)$. Optimization time is heavily dependent on chromosome randomness in EA where the difference between optimization timespans is 10 % at maximum. It is probably caused by the repair mechanism of the constructive algorithm as duration of repair is irregular – influenced by the distance between the created solution and nearest feasible.

Table I: Elitist strategy performance.

Crossover c .	0.4	0.3	0.2	0.1
Min. $f(x)$	290	301	297	287
Max. $f(x)$	308	307	304	298
Average $f(x)$	303.4	304.2	301.8	294.6
Avg. timespan [hh:mm:ss]	1:42:56	1:55:34	1:54:56	1:51:00

Parent versus child strategy does not provide better results. However, its optimization time is 10 % lower in comparison with the elitist strategy, see Table II. This is probably caused by itself principle of the method as parents are replaced only when children have better value of objective function. That makes the number of necessary operations of sorting algorithm smaller in comparison to elitist strategy. However objectively good solutions which are better than other solutions cannot succeed in the competition if they have very strong parents.

Table II: PvCH strategy performance.

Crossover c .	0.4	0.3	0.2	0.1
Min. $f(x)$	300	300	289	290
Max. $f(x)$	317	315	309	306
Average $f(x)$	310	306.6	300.7	298.2
Avg. timespan [hh:mm:ss]	1:33:15	1:39:55	1:42:51	1:42:56

Elitist strategy with clone control has the best results among all strategies as it is preventing look-alike solutions to succeed in elimination on one hand, but on the other hand, it is not preventing good solutions as in PvCH to survive into the new generation. Its optimization time is in between previous methods (see Table III) and it heavily depends on the number of clones created in the reproduction phase.

Table III: Elitist strategy with clone control.

Crossover c .	0.4	0.3	0.2	0.1
Min. $f(x)$	277	264	270	238
Max. $f(x)$	293	280	287	268
Average $f(x)$	286.2	271.2	278.9	255.3
Avg. timespan [hh:mm:ss]	1:42:47	1:47:55	1:47:09	1:50:41

Analysing given results, it is obvious that setting of $Xp = 0.1$ together with elimination strategy ECC (see Fig. 6) gives us the best results. The question is the main influence of both parameters, their interaction (if existent) and if better results from the best setting are statistically significant (note – all following figures has 10 – for 10 % crossover probability respectfully $Xp = 0.1$ etc.).

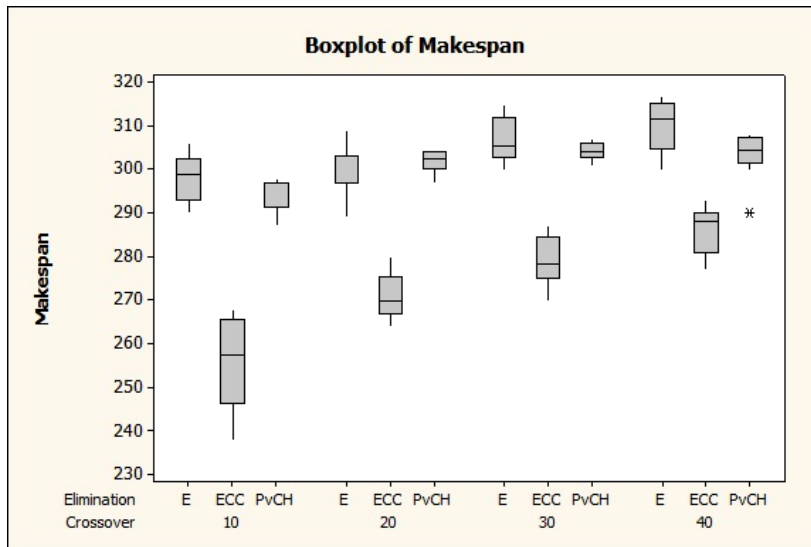


Figure 6: Major parameters boxplot of crossover probability and elimination strategy.

The first step before using statistical evaluation of results, it is necessary [33] to check if evaluate parameter outcome (objective function) has the normal distribution. There was used Minitab to perform all tests and level of importance in all tests was set to $\alpha = 5\%$ as it is common.

Checking normality of the data it was found that 7 out of 10 groups of experiments fall into interval given by normal distribution (see Fig. 7).

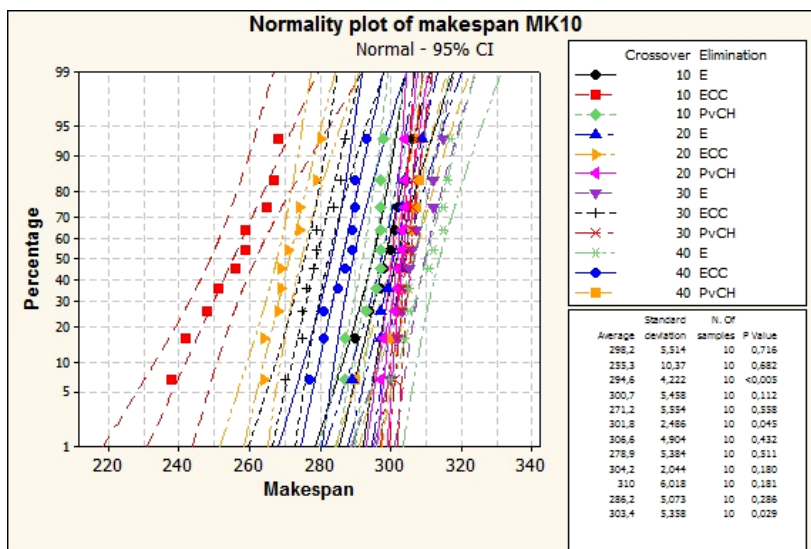


Figure 7: Normality test of MK10.

That means the whole experiment had not the normal distribution of objective function, thus it is not possible to use regular statistical methods (Anova etc.) to evaluate results properties (trends, influence, synergy etc.). Therefore, there was used Kruskal-Wallis test which is using medians to validate basic hypothesis H_0 of equality of solutions:

H_0 : Values of the objective function are same; no parameter has a statistical influence.

Both parameters obtain $p = 0$, which means that we cannot disprove that both the crossover and elimination strategy has a statistical influence on objective results. Test also has shown that ECC with $Xp = 0.1$ is the dominant solution and has a statistically significant influence on objective function (H_1 – objective functions are not the same) with overall $p < 0.05$. Fig. 8 shoves both main effects of parameters and their interaction.

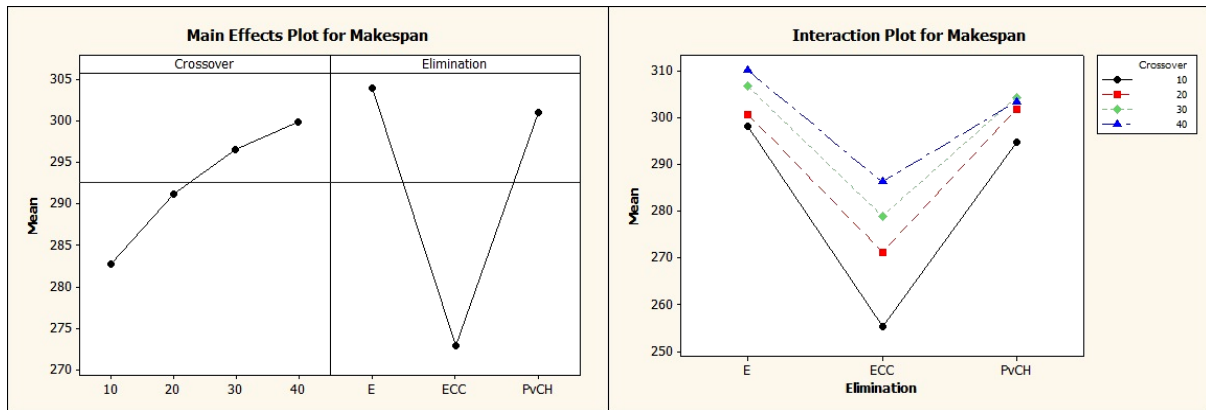


Figure 8: Main effects of parameters and their interactions.

These graphs show the dominance of ECC with $Xp = 0.1$ again. It can be stated, that there is small interaction between tested operators as all ECC solutions are better than others, same as in the case of $Xp = 0.1$ and there is the significant distance in trend between groups of 10-30 % and 40 % of crossover probability. However, interaction is not strong in comparison to others as all $Xp = 0.1$ or ECC results are better (in terms of the objective function) in the whole experiment, not just in their combination.

6. OPTIMIZATION RESULTS

The parameter setting of designed EA (BEACC) which was found in the previous chapter is tested together with recently published and previously mentioned algorithms. Along with multiagent approaches with hybrid genetic algorithm using tabu search (GATS+HM) [16] and with immune algorithm (NIMASS) [17], other EA as harmony based search (EO-DHS) [14] and improved Jaya (IJA) [8] were used. Simple evolution algorithm (SEA) based on Fig. 2 is also used. SEA is based on original research [32] using $Xp = 0.5$ and simple elitist strategy (E). Both BEACC and SEA used the same size of population $N = 2(\text{size of the problem}) + 100$ as in chapter 4 setting. The number of generations is the same for all cases $G = 200$. Table IV presents best results of makespan from 10 repetitions of SEA and BEACC while rest of the results are taken from comparison presented by Caldeira and Gnanavelbabu (2019) [8]. A different comparison was considered (min, max, average) however, each original paper is not using other joint metrics than the best result.

Table IV: Comparison of metaheuristics based on makespan.

Problem	Size	LB	SEA	BEACC	GATS+HM	NIMASS	EO-DHS	IJA
MK01	10×6	36	42	40	40	40	40	40
MK02	10×6	24	28	28	27	28	26	27
MK03	15×8	204	204	204	204	204	204	204
MK04	15×8	48	67	62	64	65	60	60
MK05	15×4	168	180	174	173	177	172	172
MK06	10×15	33	78	69	65	67	60	57
MK07	20×5	133	160	161	144	144	139	139
MK08	20×10	523	523	523	523	523	523	523
MK09	20×10	299	340	326	311	312	307	307
MK10	20×15	165	280	238	222	229	207	197

Results show (best results in bold) that clone control elimination with proper setting of crossover probability improves SEA. However, BEACC found same or worse values of

makespan than other methods. The usual reason is that the population converges to the local extreme. That was eliminated by clone control mechanism as none of 10 repetitions of BEACC converged in comparison with SEA which most repetitions converge. BEACC results are caused by poor individual recombination as presented uniform crossover manipulates with both parts of chromosome (A – sequencing, B – assigning).

Results also show that selection of MK10 was reasonable as it belongs to the hardest set among Brandimarte problems (MK04, MK06 and MK10) with the largest problem size.

7. CONCLUSION

Statistical process control approach presented in this paper allows common user to set most efficiently operators of crossover and elimination without the necessity to have experience of evolutionary computing optimization. This will not only provide the user with the proper setting of EA as it is common in known parameter tuning but also helps to understand the importance of setting up them correctly to find the good value of the objective function.

An important step of testing metric data (in this case objective function of makespan) for the statistical distribution is pointed out as most of the research in the area is neglecting this step by simply applying ANOVA without knowledge if the metric has a normal distribution. It is shown that performed testing of MK10 has not (as expected in the case of scheduling problem), so other statistical method had to be used.

Presented approach shoves not only frequent use of statistics to prove that the developed algorithm is statistically better than others, but also to identify which operator influences objective function.

MK10 was selected as the representative problem of Brandimarte set as it belongs to difficult ones as MK04 and MK06. It is also suitable from the size of the problem point of view as it is biggest from the mentioned set. On one hand, using (and not only) MK04-07 and MK09 during testing to find importance would be beneficial but out of the scope of this paper. On the other hand, using problems like MK03 and MK08 would be deforming results as even SEA was able to find the optimum. The significance would be falsely denied.

During testing was found out that crossover probability and elimination strategies have a statistically significant influence on the objective function in the case of tested algorithm settings and have little to no influence on each other. Elimination strategy is better than developed PvCH strategy while both are outperformed by ECC. Influence on computational time is limited in the case of MK10 so it is expected it would be same in the case of the smaller problems.

Comparison of developed EA with other algorithms shoves that while ECC prevents the population from converging to one solution it is not able to find in general better results in the given time (population size and defined generations). It can be assumed that better recombination (crossover, mutation) especially hybridization by the searching method [8, 16] can improve objective function as well as adaptive or self-adaptive control of population size and the number of generations.

Together with the above mentioned, further research will aim to not only improve the existing algorithm but also to test the mentioned approach on multiobjective functions and different problems connected with combinatorial optimization in production planning scheduling to make EA optimization techniques more accessible for real-world applications.

ACKNOWLEDGEMENT

This work was supported by the Student Grant Competition of the Technical University of Liberec under the project “Optimization of manufacturing systems, 3D technologies and automation” No. SGS-2019-5011 and as part of project solved at the Technical University of Košice "(KEGA

032TUKE-4/2018) Intensification of education and information processes in engineering for quality of the environment“.

REFERENCES

- [1] Ottjes, J. A.; Veeke, H. P. P. (2000). Production scheduling of complex jobs with simulation, *Proceedings of the 1999 Business and Industry Simulation Symposium*, 6 pages
- [2] Du Plessis, C. J. (2017). *A framework for Implementing Industrie 4.0 in Learning Factories*, PhD Thesis, Stellenbosch University, Stellenbosch
- [3] Zhou, L.; Chen, Z.; Chen, S. (2018). An effective detailed operation scheduling in MES based on hybrid genetic algorithm, *Journal of Intelligent Manufacturing*, Vol. 29, No. 1, 135-153, doi:[10.1007/s10845-015-1097-6](https://doi.org/10.1007/s10845-015-1097-6)
- [4] Gao, K.; Cao, Z.; Zhang, L.; Chen, Z.; Han, Y.; Pan, Q. (2019). A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems, *IEEE/CAA Journal of Automatica Sinica*, Vol. 6, No. 4, 904-916. doi:[10.1109/JAS.2019.1911540](https://doi.org/10.1109/JAS.2019.1911540)
- [5] Hozdić, E.; Kozjek, D.; Butala, P. (2020). A cyber-physical approach to the management and control of manufacturing systems, *Strojnicki vestnik – Journal of Mechanical Engineering*, Vol. 66, No. 1, 61-70, doi:[10.5545/sv-jme.2019.6156](https://doi.org/10.5545/sv-jme.2019.6156)
- [6] Romero, M. A. F.; García, E. A. R.; Ponsich, A.; Gutiérrez, R. A. M. (2018). A heuristic algorithm based on tabu search for the solution of flexible job shop scheduling problems with lot streaming, *Proceedings of the 2018 Genetic and Evolutionary Computation Conference*, 285-292, doi:[10.1145/3205455.3205534](https://doi.org/10.1145/3205455.3205534)
- [7] Yang, X. P.; Gao, X. L. (2018). Optimization of dynamic and multi-objective flexible job-shop scheduling based on parallel hybrid algorithm, *International Journal of Simulation Modelling*, Vol. 17, No. 4, 724-733, doi:[10.2507/IJSIMM17\(4\)CO19](https://doi.org/10.2507/IJSIMM17(4)CO19)
- [8] Caldeira, R. H.; Gnanavelbabu, A. (2019). Solving the flexible job shop scheduling problem using an improved Jaya algorithm, *Computers & Industrial Engineering*, Vol. 137, Paper 106064, 16 pages, doi:[10.1016/j.cie.2019.106064](https://doi.org/10.1016/j.cie.2019.106064)
- [9] Wang, L.; Cai, J.; Li, M.; Liu, Z. (2017). Flexible job shop scheduling problem using an improved ant colony optimization, *Scientific Programming*, Vol. 2017, Paper 9016303, 11 pages, doi:[10.1155/2017/9016303](https://doi.org/10.1155/2017/9016303)
- [10] Wang, Y.; Yang, O.; Wang, S. N. (2019). A solution to single-machine inverse job-shop scheduling problem, *International Journal of Simulation Modelling*, Vol. 18, No. 2, 335-343, doi:[10.2507/IJSIMM18\(2\)CO7](https://doi.org/10.2507/IJSIMM18(2)CO7)
- [11] Gong, G.; Chiong, R.; Deng, Q.; Gong, X. (2019). A hybrid artificial bee colony algorithm for flexible job shop scheduling with worker flexibility, *International Journal of Production Research*, Vol. 58, No. 14, 4406-4420, doi:[10.1080/00207543.2019.1653504](https://doi.org/10.1080/00207543.2019.1653504)
- [12] Yang, F.; Ye, C. M.; Shi, M. H. (2018). A hybrid grey cuckoo search algorithm for job-shop scheduling problems under fuzzy conditions, *Advances in Production Engineering & Management*, Vol. 13, No. 3, 254-266, doi:[10.14743/apem2018.3.288](https://doi.org/10.14743/apem2018.3.288)
- [13] Xu, H.; Bao, Z. R.; Zhang, T. (2017). Solving dual flexible job-shop scheduling problem using a bat algorithm, *Advances in Production Engineering & Management*, Vol. 12, No. 1, 5-16, doi:[10.14743/apem2017.1.235](https://doi.org/10.14743/apem2017.1.235)
- [14] Gaham, M.; Bouzouia, B.; Achour, N. (2018). An effective operations permutation based discrete harmony search approach for the flexible job shop scheduling problem with makespan criterion, *Applied Intelligence*, Vol. 48, No. 6, 1423-1441, doi:[10.1007/s10489-017-0993-1](https://doi.org/10.1007/s10489-017-0993-1)
- [15] Crawford, B.; Soto, R.; Astorga, G.; Castro, C.; Paredes, F.; Misra, S.; Rubio, J.-M. (2018). Solving the software project scheduling problem using intelligent water drops, *Technical Gazette*, Vol. 25, No. 2, 350-357, doi:[10.17559/TV-20160629224348](https://doi.org/10.17559/TV-20160629224348)
- [16] Nouri, H. E.; Driss, O. B.; Ghedira, K. (2018). Solving the flexible job shop problem by hybrid metaheuristics-based multiagent model, *Journal of Industrial Engineering International*, Vol. 14, No. 1, 1-14, doi:[10.1007/s40092-017-0204-z](https://doi.org/10.1007/s40092-017-0204-z)
- [17] Xiong, W.; Fu, D. (2018). A new immune multi-agent system for the flexible job shop scheduling problem, *Journal of Intelligent Manufacturing*, Vol. 29, No. 4, 857-873, doi:[10.1007/s10845-015-1137-2](https://doi.org/10.1007/s10845-015-1137-2)

- [18] Ojstersek, R.; Brezocnik, M.; Buchmeister, B. (2020). Multi-objective optimization of production scheduling with evolutionary computation: a review, *International Journal of Industrial Engineering Computations*, Vol. 11, No. 3, 359-376, doi:[10.5267/j.ijiec.2020.1.003](https://doi.org/10.5267/j.ijiec.2020.1.003)
- [19] Çalış, B.; Bulkan, S. (2015). A research survey: review of AI solution strategies of job shop scheduling problem, *Journal of Intelligent Manufacturing*, Vol. 26, No. 5, 961-973, doi:[10.1007/s10845-013-0837-8](https://doi.org/10.1007/s10845-013-0837-8)
- [20] Amjad, M. K.; Butt, S. I.; Kousar, R.; Ahmad, R.; Agha, M. H.; Faping, Z.; Anjum, N.; Asgher, U. (2018). Recent research trends in genetic algorithm based flexible job shop scheduling problems, *Mathematical Problems in Engineering*, Vol. 2018, Paper 9270802, 32 pages, doi:[10.1155/2018/9270802](https://doi.org/10.1155/2018/9270802)
- [21] Harik, G. R.; Lobo, F. G. (1999). A parameter-less genetic algorithm, *GECCO 99: Proceedings of the Genetic and Evolutionary Computation Conference*, 258-267
- [22] Lobo, F. G.; Goldberg, D. E. (2004). The parameter-less genetic algorithm in practice, *Information Sciences*, Vol. 167, No. 1-4, 217-232, doi:[10.1016/j.ins.2003.03.029](https://doi.org/10.1016/j.ins.2003.03.029)
- [23] Eiben, A. E.; Smit, S. K. (2011). Parameter tuning for configuring and analyzing evolutionary algorithms, *Swarm and Evolutionary Computation*, Vol. 1, No. 1, 19-31, doi:[10.1016/j.swevo.2011.02.001](https://doi.org/10.1016/j.swevo.2011.02.001)
- [24] Aleti, A.; Moser, I. (2016). A systematic literature review of adaptive parameter control methods for evolutionary algorithms, *ACM Computing Surveys*, Vol. 49, No. 3, Paper 56, 35 pages, doi:[10.1145/2996355](https://doi.org/10.1145/2996355)
- [25] Huang, C.; Li, Y.; Yao, X. (2019). A survey of automatic parameter tuning methods for metaheuristics, *IEEE Transactions on Evolutionary Computation*, Vol. 24, No. 2, 201-216, doi:[10.1109/TEVC.2019.2921598](https://doi.org/10.1109/TEVC.2019.2921598)
- [26] Raska, P.; Ulrych, Z. (2015). Comparison of optimisation methods tested on testing functions and discrete event simulation models, *International Journal of Simulation and Process Modelling*, Vol. 10, No. 3, 279-293, doi:[10.1504/IJSPM.2015.071380](https://doi.org/10.1504/IJSPM.2015.071380)
- [27] Weise, T.; Wu, Y.; Liu, W.; Chiong, R. (2019). Implementation issues in optimization algorithms: do they matter?, *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 31, No. 4, 533-554, doi:[10.1080/0952813X.2019.1574908](https://doi.org/10.1080/0952813X.2019.1574908)
- [28] Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search, *Annals of Operations Research*, Vol. 41, No. 3, 157-183, doi:[10.1007/BF02023073](https://doi.org/10.1007/BF02023073)
- [29] Ojstersek, R.; Acko, B.; Buchmeister, B. (2020). Simulation study of a flexible manufacturing system regarding sustainability, *International Journal of Simulation Modelling*, Vol. 19, No. 1, 65-76, doi:[10.2507/IJSIMM19-1-502](https://doi.org/10.2507/IJSIMM19-1-502)
- [30] Chaudhry, I. A.; Khan, A. A. (2016). A research survey: review of flexible job shop scheduling techniques, *International Transactions in Operational Research*, Vol. 23, No. 3, 551-591, doi:[10.1111/itor.12199](https://doi.org/10.1111/itor.12199)
- [31] Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization, *ORSA Journal on Computing*, Vol. 6, No. 2, 154-160, doi:[10.1287/ijoc.6.2.154](https://doi.org/10.1287/ijoc.6.2.154)
- [32] Gonçalves, J. F.; Resende, M. G. C. (2018). Random-key genetic algorithms, Marti, R.; Pardalos, P. M.; Resende, M. G. (Eds.), *Handbook of Heuristics*, Springer, Cham, 703-715, doi:[10.1007/978-3-319-07124-4_30](https://doi.org/10.1007/978-3-319-07124-4_30)
- [33] George, E. P.; Hunter, J. S.; Hunter, W. G.; Bins, R.; Kirlin IV, K.; Carroll, D. (2005). *Statistics for Experimenters: Design, Innovation, and Discovery*, 2nd edition, Wiley, New York