

Use of a Simulation Environment and Metaheuristic Algorithm for Human Resource Management in a Cyber-Physical System



Hankun Zhang, Borut Buchmeister, Shifeng Liu and Robert Ojstersek

Abstract At the time of Industry 4.0 and the emergence of collaborative workplaces based on the cooperation of robots (machines) and humans, the number of human workplaces in the Industry 4.0 production system is crucial. In this chapter, we present the use of the evolutionary computation methods that use the input data of a real production system and transfer it through the five-stage Cyber-Physical System architecture into the simulation environment in order to determine the optimal number of workers. By using these methods, we confirm the hypothesis of the importance of correctly determining the number of workers in the manufacturing process in Industry 4.0. Number of workers' determination has a key influence on the product flow time, machine utilization and cost-effectiveness of a production system. Research results show the importance and effectiveness of combining evolutionary computation methods and simulation modelling for the purpose of implementing the advanced approaches of Industry 4.0. The demonstrated approach of combining evolutionary computing, simulation environments and methods of Industry 4.0 can be used from mass customization to mass production systems for the purpose of single-criteria or multi-criteria optimization.

Keywords Simulation modelling · Evolutionary computation · Cyber-physical system · Heuristic Kalman Algorithm · Human resource management

H. Zhang · S. Liu

School of Economics and Management, Beijing Jiaotong University, Beijing 100044, People's Republic of China

e-mail: zhanghankun@bjtu.edu.cn

S. Liu

e-mail: shfliu@bjtu.edu.cn

B. Buchmeister · R. Ojstersek (✉)

Faculty of Mechanical Engineering, University of Maribor, Smetanova 17, Maribor 2000, Slovenia

e-mail: robert.ojstersek@um.si

B. Buchmeister

e-mail: borut.buchmeister@um.si

© Springer Nature Switzerland AG 2019

M. M. Gunal (ed.) *Simulation for Industry 4.0*, Springer Series in Advanced Manufacturing, https://doi.org/10.1007/978-3-030-04137-3_14

1 Introduction

The globalised world of Industry 4.0 is focused on mass customization [31], dynamic response to product demand, and real-time optimization of the manufacturing environment. Human Resource Management (HRM) is a very important aspect [3, 13]. HRM methods are, in Industry 4.0, considered as one of the primary sources for appropriate work skills, capabilities, and behaviours to achieve production system goals. In these cases, they have a significant value in the manufacturing environment. Research work done in past was focused mainly on development of human expertise in-depth knowledge. Now scientists are developing methods of Artificial Intelligence (AI) and Evolutionary Computation (EC) to solve different problems. In our case, we have introduced HRM as the main objective to optimise with AI for Job Shop Scheduling Problems (JSSP) [32, 45]. Our research work proposed a new evolutionary algorithm combined with discrete system simulation to optimise the number of workers on a factory production line. In this chapter, we present the use of simulation environment in production systems, supported by the concept of Industry 4.0, and a metaheuristic algorithm for the Minimum Number of Workers' (MNW) determination. The fundamental research work was first presented by Zhang et al. [45]. The MNW determination problem is a complex problem, due mainly to the following features:

- Possible variation of the workstation capacity during the time period.
- Legal constraints on the capacity and its evolution.
- Different skills must be considered for an operator, especially in production systems involving Industry 4.0.
- Individual company expectations.

Research work presented in this chapter is limited to an HRM problem just for MNW determination in a production system supported by Industry 4.0. The main research contribution is a newly proposed metaheuristic algorithm and its simulation testing on a real-world production system to achieved appropriate MNW determination. We proposed an improved estimation method of the Heuristic Kalman Algorithm (HKA) [24] for the purpose of HRM optimization. The research problem is based on the need to increase the productivity of the existing production system. The problem can be essentially solved by employing new workers in the existing production system or by automating the existing production system by applying the concept of Industry 4.0. In the second case, the productivity of the production system increases, and the number of workers can be reduced or unchanged. In the following research work presented in this chapter, we want to present how important it is to correctly determine the number of workers (MNW), when the concept of Industry 4.0 is introduced into exiting production system. Use of the main Industry 4.0 architecture model (CPS model) leads to an increase in productivity with a uniform workers' workload and the economic viability of workers for the enterprise. The presented research work is based on applying the 5C CPS architectural model, metaheuristic algorithm and simulation environments in order to determine the optimal number of workers in the

production process supported by the concept of Industry 4.0. All proposed methods are based on the Cyber-physical system [26], which is 5C (connection, conversion, cyber, cognition, and configuration) level cloud architecture (first connection, second conversion, third cyber, fourth cognition, and fifth configuration) based technologies. It refers to a modern manufacturing system that offers an information-transparent environment to facilitate asset management, HRM, provide flexibility, and maintain productivity.

The chapter is structured as follows: in Sect. 2, the literature review is presented in four research areas (simulation modelling, 5C CPS architecture in Industry 4.0, HRM, and Heuristic Kalman Algorithm). This section is followed by Sect. 3, in which the presentation of the 5C CPS architectural model, which is generally presented as one of the constituent concepts of Industry 4.0. In the continuation of this section, the model apply a real-world production system model, which serves as a reference model throughout the whole chapter. Section 4 presents the method of evolutionary computation called HKA, which was developed during the first development phase of our project: Implementation of HKA for the purpose of production systems' single-objective optimization. In Sect. 5, we present an Improved HKA (IHKA) method, and its results are tested and displayed on benchmark test data, followed by the implementation of the IHKA for the purpose of MNW determination. At the beginning of this section, we give some general basic knowledge and mathematical modelling in the field of HMR and MNW determination. The following is a description of the solution coding and the experimental part carried out in a real-world production system. Section 6 represents a simulation modelling of the production system in which we want to optimize a single-objective MNW parameter. The whole section is based on a real-world example, the implementation of IHKA and simulation modelling on the 5C CPS architectural model. Section 7 presents conclusions and further research work.

2 Literature Review

In a time of rapid development of companies that meet in the global market with the introduction of the Industry 4.0 concept based on mass personalization of customised products, simulation methods are very important. The introduction of simulation methods for the purpose of production systems' modelling and analysing was first presented by Emery [10], Askin and Standridge [4], who defined the basic simulation methods. These methods were improved and represented on application cases by Law and Kelton [19]. In order to optimise production, researchers use a wide range of software environments to analyse and optimise production processes [17, 29]. Due to the wide range of different simulation methods, their advantages and disadvantages, it is essential that the correct choice of simulation methods be made with respect to the optimization problem's characteristics [7]. The simulation methods are divided into two groups: Continuous simulation, in which the simulation tracks the system dynamics continuously over time. On the other hand, we have event-based simula-

tion, also called an activity-based simulation, in which time is broken up into small slices and the system state is updated according to the set of activities happening in the time slice. Because discrete-event simulations do not have to simulate every time slice, they can, typically, run much faster than the corresponding continuous simulation [11]. Practical examples of discrete systems' simulation are presented in solving scheduling problems using linear programming [14], layout and material flow optimization in a digital factory [8], and on production optimization [25]. The mentioned authors present various simulation methods and approaches, and, in doing so, they discuss a problems that arise with the application of simulation methods. In the already existing and newly proposed production processes, the use of simulation methods is particularly important in the implementation of the Industry 4.0 concept [5]. In designing production processes, authors suggest the introduction of AI and EC [43, 44], which imply the concepts of the Internet of Things (IoT), Cloud Computing, Cyber-Physical Systems (CPS) and Big Data in Industry 4.0. In most cases, the authors use the CPS model as a reference architectural model for dynamically variable production processes [36]. Several dimensional approaches are proposed for the design of advanced mechatronic systems in production processes [30], which differ essentially from Product Service Systems (PSS), especially in the processing efficiency of a Big Data calculation [22]. The established architecture, the CPS model, in relation to the implementation of Artificial Intelligence and simulation methods at all five architectural stages, is presented below with reference to the cited research work [20]. When implementing the concept of Industry 4.0 and its associated CPS architecture, the Human Resource Management (HRM) aspect is particularly important. The regularity of labour load planning affects the flexibility, productivity and efficiency of the production process significantly [12]. The HRM area has been well researched in the past production systems [13]. In the current time of Industry 4.0, based on the CPS architectural model, the appropriate treatment of HRM is more and more important [3, 34]. When discussing HRM in Industry 4.0, we also talk about collaborative workspaces that are occurring increasingly in production systems. They will have a significant impact on the setting up and organising of jobs in the future [47]. Recently, in this field of research, we can find the proposals of new methods related to a holistic HRM based on the support of robotised and automated production processes [15]. Researchers use modern approaches of AI to determine the Near Optimum (NO) solution [45] when introducing new methods proposed for the purpose of HRM optimization. The importance of HRM in the production planning and scheduling [32] and the introduction of Artificial Intelligence methods [1, 9, 35] present new research challenges for the future. Recently, research results [18, 38] demonstrated the benefits of heuristic and metaheuristic methods for the purpose of optimising production processes. Due to the complexity of the optimization problems, the solutions mentioned refer to NP-hard or strongly NP-hard problems. In solving strongly NP-hard problems, researchers use either hybrid Artificial Intelligence methods based on combining the positive properties of individual evolutionary computing methods, or solving multi-objective optimization problems [23, 46]. Particularly deeply explored is the field of Planning and Scheduling, from service activities [42] to production systems [37]. The authors implement

Artificial Intelligence algorithms to benchmark examples [28], as well as to real-world examples [33]. The high efficiency of modern Artificial Intelligence methods are reflected in the implementation of the Kalman Filter approach for the purpose of multiprocessor evolutionary computation and obtaining estimated solutions [21]. The advantage of using Heuristic Kalman Algorithm is based on ease of use and real-world application implementation [27, 39]. The Heuristic Kalman algorithm can act as an estimator of single-objective problems, as well as a multi-objective problem estimator [16]. The authors [26] first used it for the purpose of production systems single-objective optimization. Based on the obtained NO results, researchers were expanded, and improved their algorithm in the further development phase for optimising multi-objective real-world problems [24].

2.1 5C CPS Architecture

A general model of 5C CPS architecture is presented in Fig. 1. It represents the five-level architecture, which is defined as a modern mechanism for monitoring and controlling production systems [20]. It offers an information environment for optimising and designing the following parameters: Human Resources Management, flexibility and productivity sustainability. In the following section, we present an application example of using an evolutionary algorithm and simulation environment, which refers to all five levels of the architectural model.

The first level of the architectural model is a smart communication level that relates to Plug & Play applications and open-source communication protocols to sensor networks. The second level of data-to-information conversion level allows intelligent evolutionary algorithms' optimization of the expected objective to obtain

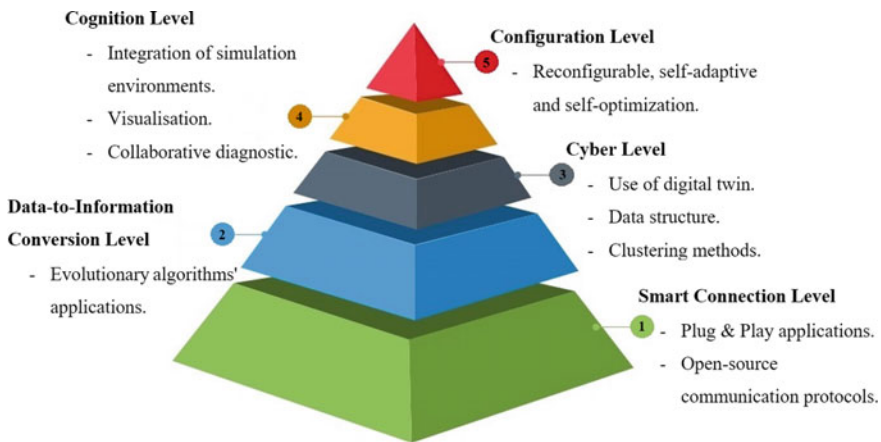


Fig. 1 The 5C CPS architecture

optimal solutions. The third level, or the cyber level, allows the use of a digital twin between the basic components in completed blocks, the data-time structure, and the use of the clustering in data structures. The fourth level, or cognition level, enables the integration of simulation environments with real-world data sets, visualization for the user, and machine interface in collaborative diagnostic and decision models. The highest, fifth level, or so-called configuration level, refers to reconfigurable, self-adaptive and self-optimization of the proposed evolutionary algorithms and simulation models. This architectural model shares similarity to the Internet of Things, but this model represents a higher-level integration degree of physical and digital elements.

In the next section, we present a real-world example of 5C CPS architecture integrated into a production system in which, at the connection level, we proposed smart communication and data transfer for the real-world simulation model, which is implemented in fourth level of cognition. The second level of conversion integrates the IHKA for MNW estimation and optimization. The Cyber level includes improved solution clustering of algorithm solutions. As said before, the cognition level integrates the simulation model with the mathematical model of IHKA. In the final, fifth configuration level, we proposed self-adjusted solutions for the purpose of near optimal HRM configuration.

2.2 *Applied 5C CPS Architecture*

The following is an introduction of the implementation of the CPS model in the case of advanced single-objective production systems' optimization methods. In the left column of Table 1, we can see five levels of the above-described 5C CPS architectural model and, presented in the right-hand column, is the architectural definition of the real production system.

1. *Smart communication level*: The input data of the real production system are captured through the analytical tools and advanced sensors systems presented in Table 3. The main constraint is that the production system must support Industry 4.0 methodology regarding individual machine sensors and data connectivity. It should be noted that the quality of the captured and processed input data has a significant impact on all the following architectural levels.
2. *Data to information conversion level*: In our case, we use the Heuristic Kalman Algorithm (HKA) [26] to evaluate the optimal solutions of the single-objective (MNW determination) production system optimization. Using the estimation method of the HKA, we can predict the optimal or almost optimal (near-optimum, NO) solutions of several production system criteria (flow rates, utilization of workplaces and machines, number of finished products in the simulation period, and MNW problem).
3. *Cyber level*: The characteristic of HKA is that, when assessing the NO solution, a negative noise (error) occurs, which contributes to the relative error of the

Table 1 Applied 5C CPS model

Architecture level	Production system implementation
Smart connection level	Input data of a real-world production system, suitable for the simulation model (all machines in production systems support Industry 4.0 methodology)
Data-to-information conversion level	Use of input data collected from a production system needed for mathematical and simulation modelling (evaluation algorithm computing)
Cyber level	Use of single-objective methods to determine optimal solutions (production system HRM and MNW determination)
Cognition level	Building a simulation model with the goal of single or multi-objective production system optimization
Configuration level	Use of self-adaptive methods with the goal of determining optimal solutions in a real-time environment

estimated value [45]. To this end, we use the method of clustering that allows us to use only the best solutions. At cyber architecture level, we proposed the Improved Heuristic Kalman Algorithm (IHKA) to obtain the best solutions.

4. *Cognitive level*: Based on all previously collected production system data and built mathematical models, the construction of the simulation model follows. The simulation model captures all production system real-world characteristics, followed by the optimal solution decision. Depending on the complexity of the production system and the built-in simulation model, we can choose to implement simulation scenarios [25], which allow us detailed simulation modelling according to the previously predicted production system characteristics.
5. *Configuration level*: The obtained solutions from the simulation experiments and mathematical model calculation depend on the function of the time variable, which, in general, means that the mentioned solutions change according to the time. To this end, we propose the introduction of self-adaptive decision-making methods for determining NO IHKA solutions.

3 Heuristic Kalman Algorithm

The Heuristic Kalman Algorithm (HKA), as a Kalman filtering based heuristic approach, only requires the user to set three parameters [38, 39]. The search heuristic of the HKA is entirely different from other population-based stochastic optimization algorithms, in that it considers the optimization problem explicitly as a measurement process designed to give an estimate of the optimum. During the measurement process, HKA develops a specific procedure based on the Kalman estimator to improve the quality of the estimate obtained. HKA needs initialising the Gaussian distribu-

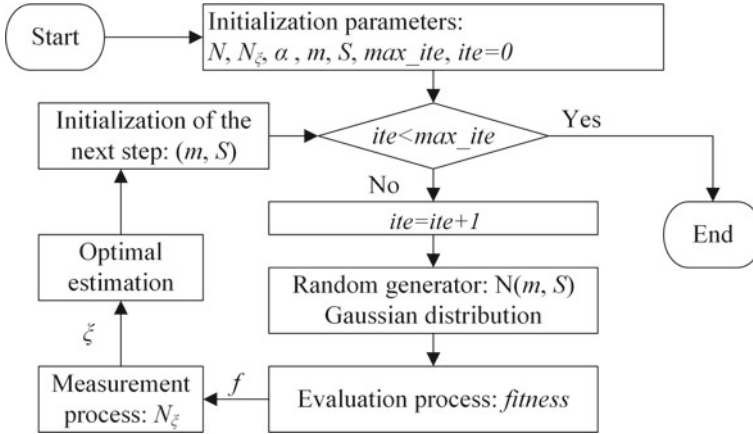


Fig. 2 The flowchart of the HKA

tion, selecting the user-defined parameters, and introducing the stopping rule for practical implementation [18]. During the HKA optimization process, first, the solutions are generated by the Gaussian distribution that is parametrised by a given mean vector with a given variance–covariance matrix, followed by the measurement procedure, and, finally, the optimum estimator of the parameters is introduced for the next generation. Figure 2 shows the flowchart of HKA [38, 39].

4 Improved Heuristic Kalman Algorithm

Experiments show that HKA is so convergent that it is easy to fall into the local minimum. This chapter proposes a new improved estimation method of the Heuristic Kalman Algorithm, the IHKA. In the IHKA, a mutation operation is introduced after the solutions are generated by the Gaussian distribution, then a function is introduced that handles the boundary constraint, and, finally, a random number is introduced in the updating formula of the standard deviation vector of the Gaussian generator. The general pseudo-code of the IHKA is shown in Algorithm 1 [24].

Algorithm 1 The general pseudo-code of the IHKA.

Step 0 Initialization. Set the size of the population Ns , the number of dimensions of the actual problem Nd , the number of top individuals under consideration $N\xi$, the slowdown coefficient α , the mutation parameter β the maximum block size of the random $Bsize$ and the maximum number of iterations max_ite . Initialise the mean vector m and the variance-covariance vector S .

$$m = \left[\frac{lu(1,1)+lu(2,1)}{2}, \dots, \frac{lu(1,j)+lu(2,j)}{2}, \dots, \frac{lu(1,Nd)+lu(2,Nd)}{2} \right]^T$$

$$S = \left[\left(\frac{lu(2,1)-lu(1,1)}{6} \right)^2, \dots, \left(\frac{lu(2,j)-lu(1,j)}{6} \right), \dots, \left(\frac{lu(2,Nd)-lu(1,Nd)}{6} \right)^2 \right]^T$$

Where $lu(1, j)$ (respectively, $lu(2, j)$) is the j^{th} lower bound (respectively, upper bound) of the problem.

Step 1 Iteration.

For $ite = 1: max_ite$

Step 1.1 Random generator. Generate a population x with Ns individuals by a Gaussian distribution parametrised by m and S :

$$x = mvnrnd(mvnrnd(m, diag(S)), Ns)$$

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 1.2 For each individual in the population

for $i = 1: Ns$

Step 1.2.1 Mutation operator by Algorithm 2.

$$x(i, :) = mutate(x(i, :), Nd, \beta, Bsize, ite, max_ite)$$

Step 1.2.2 Handling the constraints of the problem (see Equation (1)):

$$x(i, :) = handleCons(x(i, :), Gbx, Nd, lu)$$

where Gbx is the global best position

Step 1.2.3 Evaluate fitness. Calculate the individual fitness $f(i)$ in $x(i, :)$.

end

Step 1.3 Update the global best position.

Step 1.4 Choose process. Choose the top $N\xi$ individuals according to f .

Step 1.5 Measurement process. Compute the measurement ζ and the variance vector V .

$$\xi = \frac{1}{N_\xi} \sum_{k=1}^{N_\xi} x_k \quad , \quad V = \frac{1}{N_\xi} \left[\sum_{k=1}^{N_\xi} (x_{k,1} - \xi_1)^2, \dots, \sum_{k=1}^{N_\xi} (x_{k,j} - \xi_j)^2, \dots, \sum_{k=1}^{N_\xi} (x_{k,Nd} - \xi_{Nd})^2 \right]^T$$

Step 1.6 Optimal estimation. Compute the posterior estimation the mean vector m_{pe} and the variance-covariance vector S_{pe} .

$$L = S / (S + V), \quad W = (S - L .* S)^{0.5}, \quad m_{pe} = m + L .* (\xi - m), \quad \tau = \min(1, \text{mean}(\sqrt{V})^2)$$

$$a = a\tau / (\tau + \max(W)), \quad S_{pe} = (S^{0.5} + ar_z .* (W - S^{0.5}))^2$$

where a is the slowdown factor, r_z is a random number vector generated by Logistic chaotic map, $\text{mean}(\cdot)$ is a function that calculates the average or mean value and the symbol $/$ (respectively, $.*$) stands for a component-wise divide (respectively, product).

Step 1.7 Initialise the next step.

$$m = m_{pe}, \quad S = S_{pe}$$

end

Mutation operator:

In order to improve the performance of the HKA in combinatorial optimization problems, which is likely to fall into a local optimum for the fast convergence speed, a mutation operation is introduced after the population is generated. In IHKA, a mutation parameter is set to control the decreasing speed of the mutation probability [46]. As the number of iterations increases, the probability of mutation operation decreases, that is, the effect of the mutation operator decreases [46]. In the mutation function *mutate*, inspired by Zhang et al. [45], this paper introduces four mutation operators with random size; insert operator, random size move backward operator, random size swap operator, and 2-opt operator. It should be noted that the solution is based on the sort to decode, so the mutation operation is also based on the sorted individual. When mutating, a mutation operator is selected from them randomly, then two positions are selected from the current mutated individual. After the mutation block size is determined, the mutation operation is finally executed. Algorithm 2 shows the general pseudo-code of the mutation function *mutate*.

Algorithm 2 The general pseudo-code of the *mutate*.

Input: The individual before mutation xi , the number of dimensions of the actual problem Nd , the mutation parameter β , the maximum block size of the random operator $Bsize$, the current iteration ite and the maximum number of iterations max_ite .

Output: the new individual after mutation xi .

Step 1 Calculate the current mutation probability.

$$e = \exp(-\beta * ite / max_ite)$$

Step 2 Determine whether to mutate.

if $rand < e$

Step 2.1 Select a neighbourhood structure randomly.

$$ri = randi(4)$$

Step 2.2 Select two different positions randomly, and the first selected position needs to be smaller than the second one:

$$Step\ 2.3\ SI = sort(randperm(Nd, 2)).$$

Step 2.4 Determine the random operator block size.

$$rs = randi(\min([SI(2) - SI(1), Nd - SI(2) + 1, Bsize]))$$

Step 2.5 Sort the individual.

$$[xiS, xiI] = sort(xi)$$

Step 2.6 Mutation operator.

switch ri

case 1

Step 2.6.1 Random insert operation.

$$xi(xiI(SI(2):SI(2) + rs - 1)) = xiS(SI(1):SI(1) + rs - 1)$$

$$xi(xiI(SI(1):SI(2) - 1)) = xiS(SI(1) + rs:SI(2) + rs - 1)$$

case 2

Step 2.6.2 Random move backward operation.

$$xi(xiI(SI(1):SI(1) + rs - 1)) = xiS(SI(2):SI(2) + rs - 1)$$

$$xi(xiI(SI(1) + rs:SI(2) + rs - 1)) = xiS(SI(1):SI(2) - 1)$$

case 3

Step 2.6.3 Random swap operation.

$$xi(xiI(SI(1):SI(1) + rs - 1)) = xiS(SI(2):SI(2) + rs - 1)$$

$$xi(xiI(SI(2):SI(2) + rs - 1)) = xiS(SI(1):SI(1) + rs - 1)$$

```

otherwise
  Step 2.6.4 The 2-opt.
   $xi(xil(SI(2): -1: SI(1))) = xi(xil(SI(1): SI(2)))$ 
end
end

```

Handling of the constraints:

In IHKA, a handling constraints function is introduced to handle the boundary constraints and increases the abundance of the population. If the value of a dimension exceeds the constraint boundary, it is replaced by random generation with a fifty percent probability. There is 25% probability to assign to the corresponding dimension value of the global optimal solution. Otherwise, it is replaced by the minimum or maximum boundary value corresponding to less than the minimum, or greater than the maximum, boundary value, respectively.

$$x_{i,j} = \begin{cases} lu_{1,j} + (lu_{2,j} - lu_{1,j})rand, & r < 0.5 \\ Gbx_j, & 0.5 \leq r < 0.75 \\ lu_{1,j}, & r \geq 0.75 \wedge x_{i,j} < lu_{1,j} \\ lu_{2,j}, & r \geq 0.75 \wedge x_{i,j} > lu_{2,j} \end{cases} \quad (1)$$

where $x_{i,j}$ represents the j th dimension of the i th individual in the population x exceeds the value range, $rand$ is a function that generates a uniformly distributed random number in the interval $(0, 1)$, and r is a random number generated by $rand$.

Random coefficient:

In order to improve the convergence performance of the HKA, a random number is introduced in the updating formula of the Standard Deviation vector of the Gaussian generator. Inspired by literature [37], the random number is generated by the Logistic chaotic map.

4.1 IHKA Test

The Travelling Salesman Problem (TSP), as one of the most famous combinatorial optimization problems, is selected as the benchmark problem to test the performance of IHKA. We select wi29 [41], dj38 [41], eil51 [40] and eil76 [40] as the benchmark instances. Their optimal tours have lengths 27601.17, 6659.43, 429.98 and 545.39, respectively. In this chapter, algorithms were implemented in MATLAB and simulated in version R2017b. For each instance, algorithms are run independently 30 times. Figure 3 shows the IHKA and HKA convergence for the 4 TSP benchmark instances. The computational statistics of the IHKA and HKA for the fitness of the 4-benchmark instances are shown in Table 2 and in Fig. 3. As can be seen from,

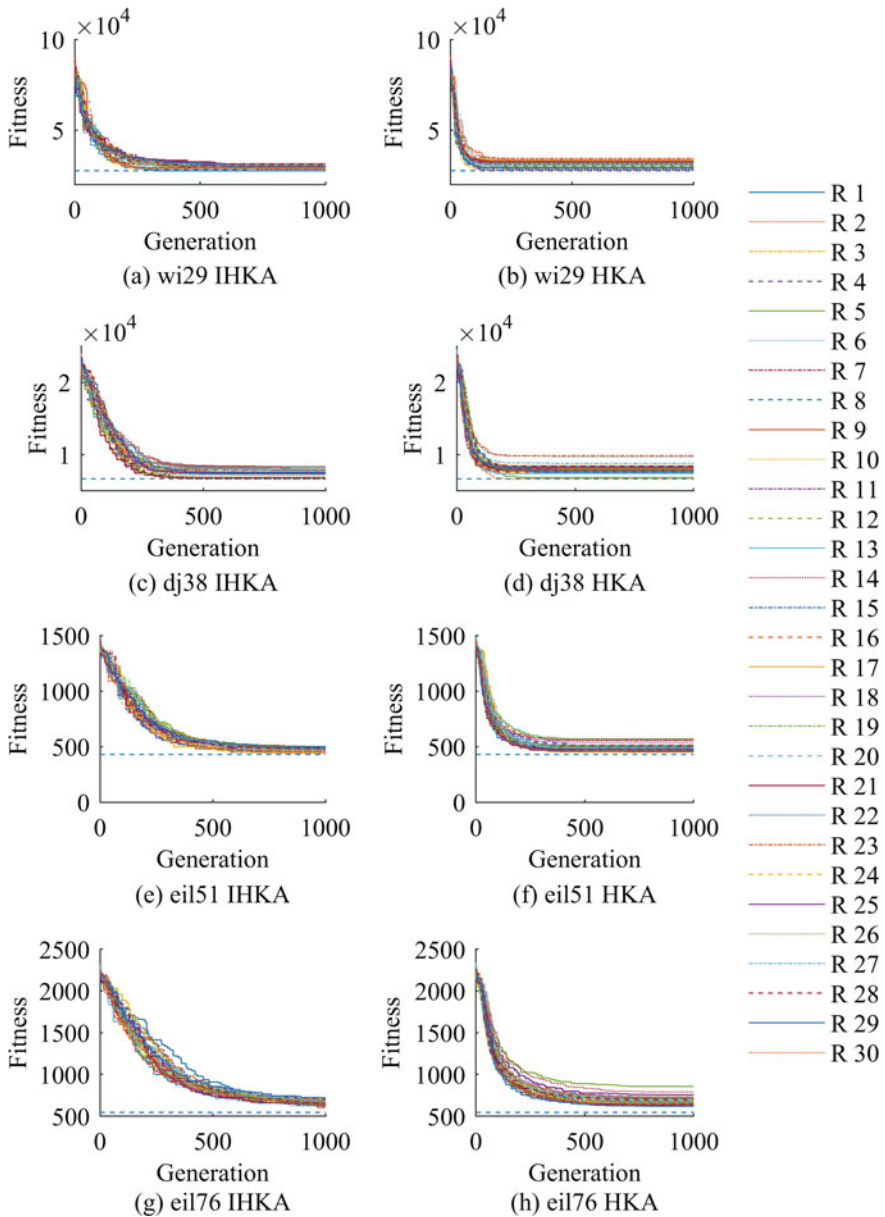


Fig. 3 The IHKA and HKA convergences of the best solutions for the benchmark instances (the horizontal dotted line is the length of the optimal tours for each instance and the “R” in the legend is an abbreviation for runtime)

Table 2 Computational statistics of the IHKA and HKA on the fitness for the benchmark instances

TSP	Algorithm	Min	Max	Mean	Standard deviation
wi29	IHKA	27,601.20	31,406.20	29,815.70	1056.77
	HKA	27,601.20	34,307.20	31,277.90	1743.08
dj38	IHKA	6659.43	8350.67	7494.76	464.09
	HKA	6659.43	9842.75	8047.69	648.52
eil51	IHKA	442.25	502.73	477.91	18.25
	HKA	451.98	570.28	496.61	33.66
eil76	IHKA	602.39	721.30	663.59	28.91
	HKA	619.35	856.57	691.71	51.00

both IHKA and HKA can tend to converge to the global optimal in the 4 benchmark instances. In the same benchmark instances, the convergence speed of IHKA is obviously lower than HKA, which reduces the possibility of falling into a local minimum. As the dimension of the problem increases, their speed of convergence decreases significantly. Table 2 shows that the robustness of IHKA is significantly better than HKA in all four selected benchmark instances. For the 30 independent runs, the mean and standard error of the IHKA is smaller than that of the HKA in all selected benchmark instances.

5 IHKA Applied in MNW

Human Resource Management, especially MNW determination, is a critical task in Industry 4.0 production systems. We must allocate workers appropriately due to two resources:

- *Technical resources*: Are smart manufacturing equipment supported by the Industry 4.0 production line where the workload per operation is calculated based on production systems' planning and scheduling tasks. In this case, we must assign the human resources carefully with regard to the machine specific constraints and demands (utilization, workflow, control time, maintenance time, etc.).
- *Human resources*: Are critical where they are assigned simultaneously to a job considering the same level technical resources. Specific qualifications, skills, capabilities, behaviour, attitude and technical knowledge are required from the workers.

Regarding the above described technical and human resources, HRM in Industry 4.0 is based on [20]:

- *Staffing*: The right candidate for every job must be selected using extensive recruitment and selection methods, where the potential of the candidate is very important.

- *Training*: Manufacturing or service companies in Industry 4.0 must design their own training programmes to enhance the innovative capability and knowledge of employees.
- *Compensation*: The contribution of the employees to the company should be a ratio between performance, working achievements and reward. With an appropriate ratio, we can enhance innovation and the learning curve of the company.
- *Job design*: In a mass personalization production, system, the job design must be flexible regarding tasks and responsibilities of the employees. High flexibility of all employees can help the company to adjust quickly to the customers' demands.

5.1 MNW Mathematical Model

The following is a mathematical model of HRM for determining MNW. The mathematical model was proposed by Becker and Scholl [6], and modified by Alghazi [2]. The presented mathematical model is adapted according to the 5C CPS architectural model and real-world production system characteristics.

Notation:

- c cycle time
 t time
 j potential worker
 i machine station
 N number of task indexed $h, l = 1, 2, 3, \dots, N$
 F_h the set of feasible stations that task h is assignable to
 F_l the set of feasible stations that task l is assignable to
 O_l the set of immediate predecessors of task l
 s_h starting time of task h
 s_l starting time of task l
 t_h^f lateness for task h
 t_h^s earliness for task h .

Variables:

$$x_{ijh} = \begin{cases} 1, & \text{if task } h \text{ is assigned to station } i \text{ and worker } j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$x_{ijl} = \begin{cases} 1, & \text{if task } l \text{ is assigned to station } i \text{ and worker } j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$y_{ij} = \begin{cases} 1, & \text{if potential worker } j \text{ at machine station } i \text{ is assigned} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$v_{hl} = \begin{cases} 1, & \text{if task } h \text{ is executed before task } l \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$v_{lh} = \begin{cases} 1, & \text{if task } l \text{ is executed before task } h \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

The main objective to minimise is $MNW = \sum_i \sum_j y_{ij}$, Eqs. (7–12):

- Just one task can be assigned to an individual worker.

$$\sum_{i \in FS_h} \sum_j x_{ijh} = 1, \quad \forall h \quad (7)$$

- Cycle time assigned to a worker.

$$\sum_h x_{ijh} \leq cy_{ij}, \quad \forall (i \in F_h, j) \quad (8)$$

- Individual task station time, each task assigned to a worker should be scheduled between the workers’ machine centre star and finish times.

$$\begin{cases} s_h \geq \sum_{i \in F_h} \sum_j S_i x_{ijh} \\ s_h \leq \sum_{i \in FS_h} \sum_j (S_i + c) x_{ijh} \end{cases}, \quad \forall h \quad (9)$$

- Individual task can only start when the existing task is finished.

$$s_h \leq s_l, \quad \forall h, l \in O_l \quad (10)$$

- Task assigned to worker must be executed before the next task can start.

$$\begin{aligned} v_{hl} + v_{lh} &\geq x_{ijh} + x_{ijl} - 1, \quad \forall j, h \neq l \wedge i \in F_h \cap F_l \\ s_h + t_h &\leq s_l + (1 - v_{hl}) (t_h^f - t_h^s), \quad \forall h \neq l \end{aligned} \quad (11)$$

- Generalised MNW calculation regarding the upper equations.

$$MNW = \frac{\sum t_h}{c} \quad (12)$$

The described mathematical model was implemented in the next section, where we represent the solution of using evolutionary computing, IHKA for solving the HRM problem with MNW determination in a real-world Industry 4.0 production system.

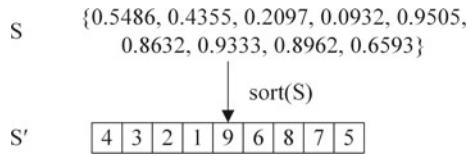


Fig. 4 The example of the solution coding

5.2 Solution Coding

In this chapter, the machines for the workers to check are represented by the index of the order in the production lines (see the ‘No.’ column in Table 3). The number of dimensions is set to $Nd = machinesNum + workersNum - 1$ to encode the workers into the coding, where *machinesNum* and *workersNum* are the number of machines and the number of workers in the problem, respectively [45]. It means that if the number of workers is more than 1, one or more decision $workersNum - 1$ variables are used as separators, which are greater than *machinesNum* [45]. The value range of the *j*th dimension of an individual is an open interval (0, 1). In order to solve the combinatorial optimization problem by IHKA, this chapter introduces the relative position indexing [23] to transform the optimised solution into the discrete domain. We sort the original solution to get the machines for each worker to check and the order in which they check the machines. In Fig. 4, we sort the S to get S', the decision variable 9 is serviced as a separator, the S' can be decoded as worker 1 checks the machines, and its sequence is (4, 3, 2, 1), and worker 2 checks the machines, and its sequence is (6, 8, 7, 5).

Table 3 The data of the production lines in the manufacturing enterprise E

No.	Name	Position (x, y)	Checking time (min)	No.	Name	Position (x, y)	Checking time (min)
1	A	(0, 1.5)	15	9	A'	(3, 28)	15
2	B	(0, 4.8)	17	10	B'	(3, 26)	17
3	C	(0, 9.1)	17	11	C'	(3, 22)	17
4	D	(0, 13.5)	24	12	D'	(3, 16.5)	24
5	E	(0, 16.5)	55	13	E'	(3, 13.5)	55
6	F	(0, 22)	28	14	F'	(3, 9.1)	28
7	G	(0, 26)	17	15	G'	(3, 4.8)	17
8	H	(0, 28)	17	16	H'	(3, 1.5)	17

5.3 Computational Experiment

A manufacturing enterprise E in Slovenia has two automated production lines with the same machines, but the order of the machines on these two production lines is opposite. During the operation of these two production lines, workers are required to check the machines on them. Table 3 shows the data of these two production lines in the manufacturing enterprise E [45]. According to the Zhang et al. [45], the minimum checking time for a worker and two workers is 380.75 and 190.36 min, respectively.

5.4 Experimental Results

For comparison, three well-known meta-heuristic algorithms, the Particle Swarm Optimization (PSO) proposed by Eberhart and Kennedy [9] and improved by Shi and Eberhart [35], the Multi-Phase Particle Swarm Optimization (MPPSO) proposed by Al-Kazemi [1], and the Bare Bones Particle Swarm Optimization (BBPSO) proposed by Kennedy [18], were selected to assess the performance of IHKA.

For all algorithms in this chapter, the size of the population $N_s = 100$, and the maximum number of iterations is set to $max_ite = 1000$. According to the literatures [28, 38] and experiments, the parameter for the IHKA is set as $N_\xi = 10$, $\alpha = 0.9$, $\beta = 5$ and $Bsize = \lceil \frac{Nd}{5} + 0.5 \rceil$. The parameter for the PSO is set as $c_1 = 2.8$, $c_2 = 1.3$ and $w = 0.729$ [33]. The parameter for the MPPSO is set as $ph = 2$, $pcf = 5$, $g = 2$, $sllu = [1, \min(10, Nd)]$ and $VC = 10$ [1].

Figure 5 and 6 shows the convergence of 5 algorithms for the MNW with 1 and 2 workers, respectively. The statistical analysis of the 5 algorithms for the MNW with 1 and 2 workers are shown in (a) and (b) of Fig. 7, respectively. The computational statistics of the 5 algorithms on the fitness for the MNW with 1 and 2 workers are shown in Tables 4 and 5, respectively. The proposed IHKA performed well in the MNW with 1 worker. In the MNW with 1 worker, the success rate of the improved algorithm to find the optimal value is 100%. However, the success rate of the improved algorithm to find the optimal value is very low in the MNW with 2 workers. However, the improved HKA performs better than the original HKA in both the MNW with 1 and 2 workers.

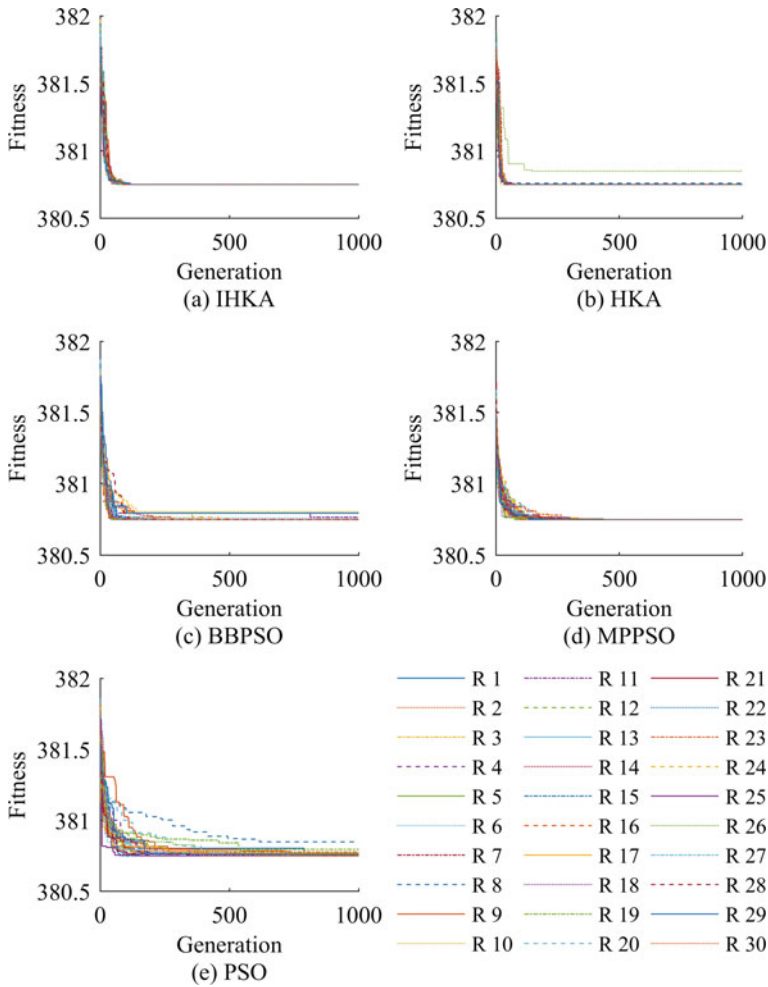


Fig. 5 The convergence rates of the five algorithms convergences for the MNW with 1 worker (the “R” in the legend is an abbreviation for runtime)

The MPPSO performs the best in both the MNW with 1 and 2 workers, while the BBPSO performs the second. In the MNW with 1 worker, the IHKA and MPPSO perform the same, they are the best among the 5 algorithms. The performance of the PSO is the worst among the 5 algorithms for the MNW with 1 worker. However, both improved HKA and original HKA perform worst among the 5 algorithms in the MNW with 2 workers. The MPPSO and PSO are performing better among the 5 algorithms in the MNW with 2 workers. Therefore, the improved HKA improves the performance of the original HKA, but still needs further improvement to increase its performance.

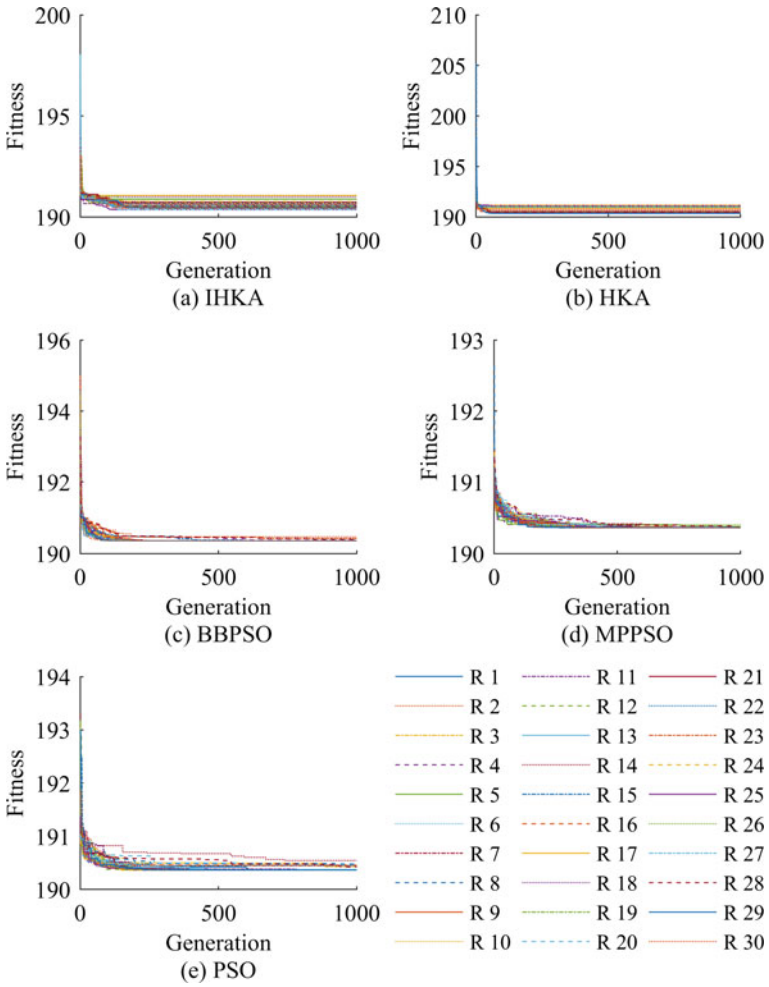


Fig. 6 The convergence rates of the five algorithms convergences for the MNW with 2 workers (the “R” in the legend is an abbreviation for runtime)

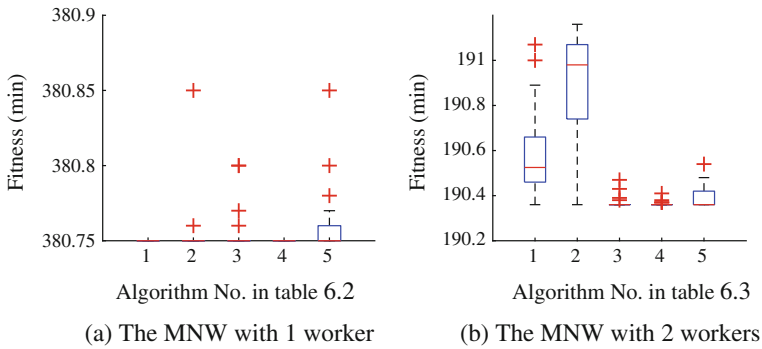


Fig. 7 The statistical analysis of the five algorithms for the MNW with 1 and 2 workers

Table 4 Computational statistics of the five algorithms on the fitness for the MNW with 1 worker

No.	Name	Min	Max	Mean	Standard deviation	Success rate (%)
1	IHKA	380.75	380.75	380.75	0	100
2	HKA	380.75	380.85	380.75	0.02	93.33
3	BBPSO	380.75	380.80	380.76	0.01	86.67
4	MPPSO	380.75	380.75	380.75	0	100
5	PSO	380.75	380.85	380.76	0.02	50

Table 5 Computational statistics of the five algorithms on the fitness for the MNW with 2 workers

No.	Name	Min	Max	Mean	Standard deviation	Success rate (%)
1	IHKA	190.36	191.07	190.59	0.18	6.67
2	HKA	190.36	191.16	190.88	0.24	3.33
3	BBPSO	190.36	190.47	190.37	0.02	86.67
4	MPPSO	190.36	190.41	190.37	0.01	83.33
5	PSO	190.36	190.54	190.39	0.05	63.33

6 Simulation Modelling

Simulation modelling in 5C CPS architecture is the process of creating a smart digital model of the physical model to estimate its performance and behaviour in a real-world production system. A smart digital simulation model can estimate and analyse a wide range of production system parameters by applying a software environment. In our case, we use Simio, simulation and scheduling software for the purpose of production system optimization and MNW determination. Then we propose data exchange and results' calculation between the mentioned optimization methods of IHKA for the purpose of MNW determination. We have implemented the real-world simulation model, which integrate analysis and design solution for production system, created in the simulation environment Simio [17]. The simulation model shown in Fig. 8

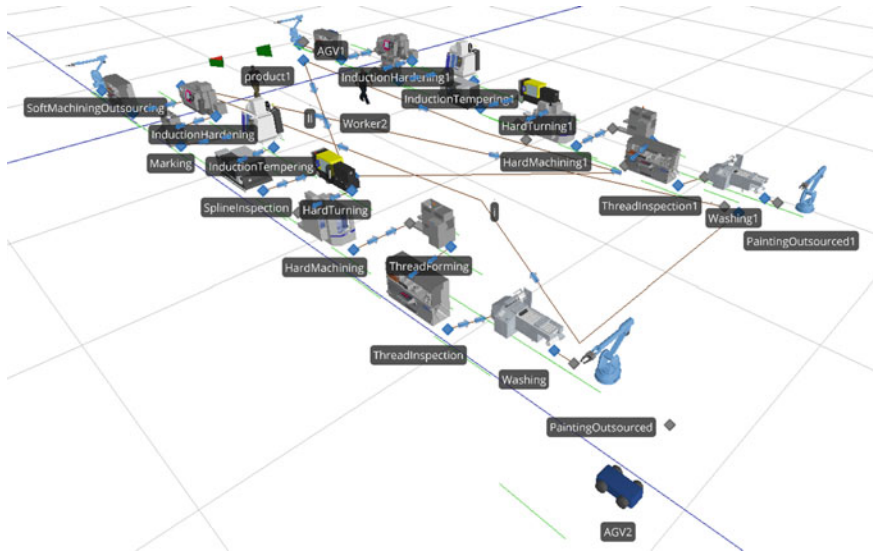


Fig. 8 The production system model in Simio

presents a real-world model of factory line E. The simulation model consists of all the necessary real-world data from a real production line: The number of the machine centre (machine centre parameters: Utilization, Overall Equipment Efficiency (*OEE*) and machine piece's capacity), distances, times and number of workers, which was calculated by IHKA.

The experiment was carried out in a manufacturing enterprise based in Slovenia, in the European Union. The production system has two automated, Industry 4.0 supported production lines, which still needs workers to check the machines in the automated production lines while they are running. In this case, the number of workers must be determined. Therefore, at the pre-determined number of workers, it is possible to optimise the number of machines for each worker to check and their checking sequence of the machines. Once the optimal solution is obtained at the pre-determined number of the workers, the maximum time required among all workers (the time of the critical worker) to complete the check of the machines, denoted as *best_fitness*, can be determined, and the comparison can be made with the working time in one shift. If $best_fitness \leq shift_time$, the pre-determined number of workers is sufficient. Otherwise, it is not enough. In order to determine the optimal number of workers, we can reduce or increase the number of workers, then the new *best_fitness* *i* is obtained for the next comparison.

The real-world simulation, the model consists of ten machine centres that perform the operations listed in Table 6. We can also see the individual operation duration, Overall Equipment Effectiveness (*OEE*) and individual machine piece's capacity. Operations 0 and 11 are underlined due to outsourcing; their parameters are not

Table 6 Real-world production system parameters

Number of operations	Operation	Duration (s)	OEE	Capacity (parts)
0	<i>Soft machining outsourced</i>	/	/	/
1	Spline and thread rolling	15	85%	1630
2	Induction hardening	17	85%	1630
3	Marking	15	85%	1630
4	Induction tempering	17	85%	1630
5	Spline inspection	17	85%	1630
6	Hard turning	24	85%	1020
7	Combine hard machining	55	85%	890
8	Thread forming	28	85%	870
9	Thread inspection	17	85%	870
10	Washing	17	85%	1440
11	<i>Painting outsourced</i>	/	/	/

available, but the contract with an external supplier guarantees that they are always available.

Performing the processed operations is carried out in the above sequence, the transport of the product between machine centres is carried out using a conveyor belt. Its speed is 0.1 m/s. Two workers, who are responsible for the smooth operation of the machine centres, operate the production line and the work pieces' quality control at control points. An Automated Guided Vehicle (AGV) ensures the access to semi-finished products and the removal of finished products automatically. The AGV speed is 1.11 m/s, the speed of the AGVs is limited electronically. Semi-finished products that arrive at the processing line are already pre-treated at the external supplier, also the finished products require external corrosion protection. The entire production system is fully automated (production and logistics), just the operation of Quality Control (which is not performed on all products, just on randomly picked products) is made by workers, Fig. 9. In the further step, we propose machine vision operation at Quality Control for more robust, consistent and reliable Quality Control operation.

In the results received by IHKA, the E production line only needs one worker to complete all the machines' check in one shift. However, in the simulation of the Simio software, the *shift_time* is a constant value for one eight hour working shift, calculated in Eq. (13). *Shift_time* exclude three brakes, one 30 min lunch break and two shorter 15 and 10 min rest brakes. The OEE of the machines is 85%, that is, the effective working time of the machines in one shift is set as:

$$\begin{aligned}
 \text{shift_time} &= 8 * 60 - 30 - (15 + 10) \\
 &= 425 \text{ min}
 \end{aligned}
 \tag{13}$$

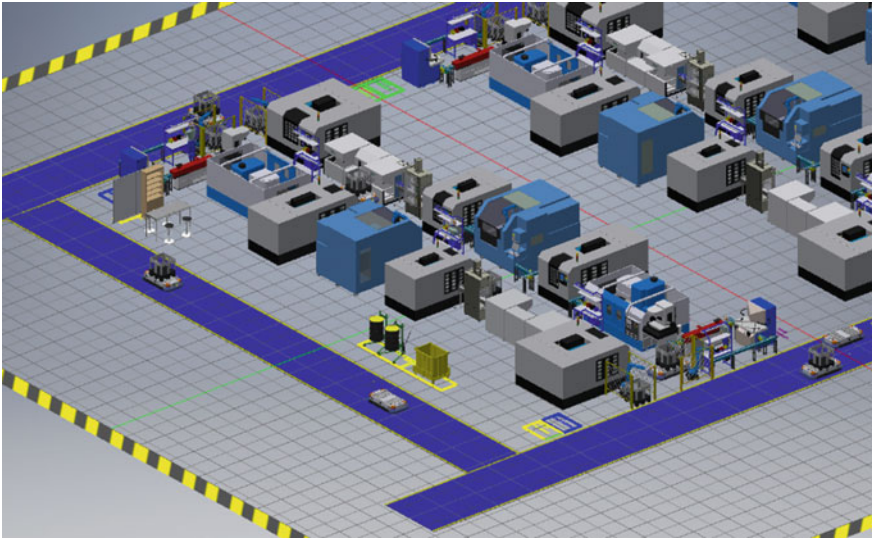


Fig. 9 The three dimensional view of production system model

$$\begin{aligned}
 machine_time &= shift_time \cdot OEE \\
 &= shift_time \cdot 85\% \\
 &= 361.25 \text{ min}
 \end{aligned}
 \tag{14}$$

The effective working time of the machines is 361.25 min in one shift. In this case, one worker is not enough to complete all the machine checking in one shift. The real-world simulation model takes into account also the tools' changing time, maintenance time and randomly occurring emergency situations that happen during the production lines' operation. Results from the simulation model recommend the presence of two workers to check the machines and perform Quality Control in one shift. The simulation results confirm the IHKA calculated optimal times for performing the control check of the production system in amount of 380.75 min in the case of one worker and 190.36 min in the case of controlling two workers. In this case, when *machine_time* is 361.25, one worker is not capable of quality performing controlling tasks in the production system, that is why we need two workers as is it calculated by the IHKA.

7 Conclusions

In this chapter, we have presented the Industry 4.0 5C CPS architectural model, which was applied successfully to the five-level architecture implemented with simulation modelling and Evolutionary Computation. We demonstrated a methodology

of 5C CPS architecture and practical approach for the transfer of theoretical knowledge to the real-world production system. Then, we presented the HKA evolutionary method for the purpose of determining single-criteria optimization and extended it to the IHKA. The IHKA is proposed and applied successfully to solve the MNW problem. Based on the original HKA, a mutation operation is introduced after the solutions are generated by the Gaussian distribution, a function that handles the boundary constraint is introduced, and, finally, a random number is introduced in the updating formula of the standard deviation vector of the Gaussian generator, to improve the performance of the algorithm. In the mutation operation, four operators with random size insert operator, random size move backward operator, random size swap operator, and 2-opt operators were introduced. A random number generated by the Logistic chaotic map is introduced in the updating formula of the standard deviation vector of the Gaussian generator. The discrete continuous mapping encoding system, based on the relative position indexing, is introduced for the MNW. The IHKA is tested on 4 selected TSP benchmark instances. From the 4 selected TSP benchmark instances, it can see that the improved HKA improves the performance of the original HKA. In solving the MNW problem; three algorithms were selected for comparison, a quantitative analysis method based on statistical analysis, and a qualitative method based on convergence figures, were used to clarify the performance of the IHKA. Although the improved HKA performs better than the original HKA in both MNW with 1 and 2 workers, especially in the MNW with 1 worker, but performs poorly in the MNW with 2 workers, it still needs further improvement to increase its performance. The optimization results of the IHKA algorithm were transferred to the simulation environment, where the correctness was simulated of the obtained NO solutions. The results confirm the correspondence between the proposed methods of Evolutionary Computing (IHKA) and simulation modelling (Simio). In this case, we have combined advanced knowledge of simulation modelling and evolution computing for the purpose of single-objective MNW optimization in Industry 4.0.

Further research work will be based on the implementation of collaborative workplaces in Industry 4.0 manufacturing systems. Here, the main question arises related to the impact of productivity, efficiency and, at the ultimate stage, workers social inclusion in collaborative workplaces. Their determination, eligibility and productivity could be determined with simulation modelling. In this case, simulation modelling will be very important in the phase of inclusion of collaborative workplaces in real-world production systems. However, in decision-making methods, we should not forget the need for the integration of evolutionary computational algorithms for the purpose of determining NO solutions regarding collaborative work places. Our further research work will be based on the optimal number determination and the setting of collaborative workplaces in the production systems using EC methods and simulation environments. For the laboratory simulation testing, we will introduce methods of virtual and augmented reality, which will combine EC algorithms and simulation models for collaborative workplaces optimization.

Acknowledgements The authors gratefully acknowledge the support of the Slovenian Research Agency (ARRS), research core founding No. P2-0190 and the China National Science Foundation (71132008, 71390334).

References

1. Al-Kazemi BSN (2002) Multiphase particle swarm optimization. Syracuse University, New York, USA
2. Alghazi AA (2017) Balancing and sequencing of mixed model assembly lines. Clemson University, South Carolina, USA
3. Armstrong M (2006) A handbook of human resource management practice. Cambridge University Press, India
4. Askin RG, Standridge CR (1993) Modeling and analysis of manufacturing systems. Wiley, New York, USA
5. Bartodziej CJ (2016) The concept industry 4.0: an empirical analysis of technologies and applications in production logistics. Springer, Berlin
6. Becker C, Scholl A (2009) Balancing assembly lines with variable parallel workplaces: problem definition and effective solution procedure. *Eur J Oper Res* 199(2):359–374
7. Borshchev A (2013) The big book of simulation modeling: multimethod modeling with AnyLogic 6. AnyLogic North America, Chicago, USA
8. Centobelli P, Cerchione R, Murino T, Gallo M (2016) Layout and material flow optimization in digital factory. *Int J Simul Modell* 15(2):223–235
9. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95: Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan. pp 39–43
10. Emery JC (1969) Job shop scheduling by means of simulation and an optimum-seeking search. In: Winter simulation conference: proceedings of the third conference on applications of simulation, Los Angeles, USA, pp 363–372
11. Fishman GS (2013) Discrete-event simulation: modeling, programming, and analysis. Springer Science & Business Media
12. Franchini L, Caillaud E, Nguyen P, Lacoste G (2001) Workload control of human resources to improve production management. *Int J Prod Res* 39(7):1385–1403
13. Geyer M, Linner S (2005) Human aspects in manufacturing process management. In: Zulch G, Jagdev HS, Stock P (eds) Integrating human aspects in production management. Springer, Berlin, pp 101–109
14. Granja C, Almada-Lobo B, Janela F (2014) An optimization based on simulation approach to the patient admission scheduling problem using a linear programming algorithm. *J Biomed Inform* 52:427–437
15. Hecklau F, Galeitzke M, Flachs S, Kohl H (2016) Holistic approach for human resource management in Industry 4.0. *Procedia CIRP* 54:1–6
16. Hultmann Ayala HV, Santos Coelho L, Reynoso-Meza G (2017) Heuristic Kalman Algorithm for multiobjective optimization. *IFAC-PapersOnLine*, 50(1), 4460–4465. Available from <https://www.sciencedirect.com/science/article/pii/S2405896317307188>. Accessed 25 Sept 2018
17. Joines JA, Roberts SD (2013) Simulation modeling with SIMIO: a workbook. Pennsylvania, USA, Simio LLC Sewickley
18. Kennedy J (2003) Bare bones particle swarms. In: Proceedings of the 2003 IEEE swarm intelligence symposium, SIS'03, Indianapolis, USA, IEEE, pp 80–87
19. Law AM, Kelton WD (2007) Simulation modeling and analysis. USA, McGraw-Hill, New York

20. Lee J, Bagheri B, Kao HA (2015) A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manuf Lett* 3:18–23
21. Lin Z, Wang C (2013) Scheduling parallel Kalman filters for multiple processes. *Automatica* 49(1):9–16
22. Marilungo E, Papetti A, Germani M, Peruzzini M (2017) From PSS to CPS design: a real industrial use case toward Industry 4.0. *Procedia CIRP* 64:357–362
23. Marinakis Y, Marinaki M (2012) A hybrid particle swarm optimization algorithm for the open vehicle routing problem. In: Birattari DM, Blum MC (eds) *Swarm intelligence: 8th international conference, ANTS 2012, Brussels, Belgium, 12–14 Sept 2012, Proceedings*. Berlin, Springer Berlin Heidelberg, pp 180–187
24. Ojstersek R, Zhang H, Shifeng L, Buchmeister B (2018) Improved Heuristic Kalman Algorithm for solving multi-objective flexible job shop scheduling problem. *Procedia Manuf* 17:895–902
25. Ojstersek R, Buchmeister B (2017) Use of simulation software environments for the purpose of production optimization. In: Katalinic B (ed) *DAAAM interantional proceedings*. Zadar, DAAAM International, pp 750–758
26. Ojstersek R, Zhang H, Palcic I, Buchmeister B (2017) Use of Heuristic Kalman algorithm for JSSP. In: Andraš A (ed) *XVII International scientific conference on industrial systems*. Novi Sad, Faculty of Technical Sciences, Department for Industrial Engineering and Management, pp 72–77
27. Pakrashi A, Chaudhuri BB (2016) A Kalman filtering induced heuristic optimization based partitional data clustering. *Inf Sci* 369:704–717
28. Palacios JJ, Puente J, Vela CR, González-Rodríguez I (2016) Benchmarks for fuzzy job shop problems. *Inf Sci* 329:736–752
29. Pegden CD (2008) Introduction to SIMIO. In: *Winter simulation conference*. Miami, USA, pp 229–235
30. Penas O, Plateaux R, Patalano S, Hammadi M (2017) Multi-scale approach from mechatronic to cyber-physical systems for the design of manufacturing systems. *Comput Ind* 86:52–69
31. Pine BJ (1993) *Mass customization: the new frontier in business competition*. Harvard Business School Press, Boston, USA
32. Pinedo ML (2012) *Scheduling: theory, algorithms, and systems*. Springer, Boston, Massachusetts, USA
33. Poormirzaee R (2016) S-wave velocity profiling from refraction microtremor Rayleigh wave dispersion curves via PSO inversion algorithm. *Arab J Geosci* 9(16):661–673
34. Shamim S, Cang S, Yu H, Li Y (2016) Management approaches for Industry 4.0: a human resource management perspective. In: *2016 Congress on evolutionary computation (CEC)*. Vancouver, Canada, IEEE, pp 5309–5316
35. Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. In: *Proceedings of the 1999 congress on evolutionary computation, CEC 99*. Washington, USA, pp 1945–1950
36. Shin HJ, Cho KW, Oh CH (2018) SVM-based dynamic reconfiguration CPS for manufacturing system in Industry 4.0. *Wirel Commun Mob Comput* 2018:1–13
37. Singh MR, Mahapatra SS (2016) A quantum behaved particle swarm optimization for flexible job shop scheduling. *Comput Ind Eng* 93:36–44
38. Toscano R, Lyonnet P (2010) A new heuristic approach for non-convex optimization problems. *Inf Sci* 180(10):1955–1966
39. Toscano R, Lyonnet P (2012) A Kalman optimization approach for solving some industrial electronics problems. *IEEE Trans Ind Electron* 59(11):4456–4464
40. University of Heidelberg (2018) TSPLIB95. Available at <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/>. Accessed 25 June 2018
41. University of Waterloo (2017) National traveling salesman problems. Available at <http://www.math.uwaterloo.ca/tsp/world/countries.html>. Accessed 10 March 2017
42. Xiang W, Yin J, Lim G (2015) An ant colony optimization approach for solving an operating room surgery scheduling problem. *Comput Ind Eng* 85:335–345
43. Yang W, Takakuwa S (2017) Simulation-based dynamic shop floor scheduling for a flexible manufacturing system in the industry 4.0 environment. In: *Winter simulation conference*. Las Vegas, USA, pp 3908–3916

44. Yao X, Zhou J, Zhang J, Boer CR (2017) From intelligent manufacturing to smart manufacturing for Industry 4.0 driven by next generation artificial intelligence and further on. In: 5th International conference on enterprise systems. Beijing, China, pp 311–318
45. Zhang H, Liu S, Moraca S, Ojstersek R (2017) An effective use of hybrid metaheuristics algorithm for job shop scheduling problem. *Int J Simul Modell* 16(4):644–657
46. Zhang Y, Gong DW, Ding Z (2012) A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch. *Inf Sci* 192:213–227
47. Zhou Z, Li M, Shen L (2006) Manufacturing resource management for collaborative process planning. In: Wang K, Kovacs GL, Wozny M, Fang M (eds) *Knowledge enterprise: intelligent strategies in product design, manufacturing, and management*. Springer, Shanghai, pp 926–931