Computers & Industrial Engineering 118 (2018) 366–382

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A multi-skilled worker assignment problem in *seru* production systems considering the worker heterogeneity

Jie Lian[a], ChenGuang Liu[b,*], WenJuan Li[a], Yong Yin[c]

[a] School of Economics and Management, Xi'an University of Technology, Xi'an 710048, China
[b] School of Management, Northwestern Polytechnical University, Xi'an 710072, China
[c] Graduate School of Business, Doshisha University, Kyoto 602-8580, Japan

ARTICLE INFO

Keywords:
Worker-task assignment
Workload balance
Interpersonal equity
Genetic algorithm
Assembly cells

ABSTRACT

The flexibility of production systems is becoming more and more important for manufacturers facing various changes. As an effective way to increase the flexibility in a short time with a small investment, multi-skilled workers have received more attention in recent years. In this paper, a multi-skilled worker assignment problem is solved in the context of *seru* production systems, in which differences in workers' skill sets and proficiency levels are taken into consideration. Worker grouping, cell loading, and task assignment are solved concurrently in the problem. A mathematical model with objectives of improving the inter-*seru* and inter-worker workload balance is proposed to solve the problem. In order to verify the proposed model, a numerical example is presented and solved by a sum weighted method. Due to the NP-hard nature of the model, a meta-heuristic algorithm based on NSGA-II is developed. The algorithm is tested by several numerical examples and the impact of differences in workers' competency on workload balance is analyzed based on the computational results.

1. Introduction

The flexibility of production systems is becoming increasingly important to manufacturers confronting demands characterized by varied product assortment and fluctuant volumes. Production systems in an Industrial Revolution 4.0 environment must have a high degree of flexibility and agility to deal with product changes (Niakan, Baboli, Moyaux, & Botta-Genoulaz, 2016). *Seru* production is a kind of production system that can meet such requirements. As the most applied production mode in the Japanese electronics industry, *seru* production employs a new type of work-cell-based manufacturing system in which the majority of tasks are manual ones assisted by hand tools or light-weight, small, inexpensive equipment (Liu, Lian, Yin, & Li, 2010). *Seru* production originated in dismantling of a lengthy multi-product conveyor assembly line at Sony in Japan for the purpose of improving flexibility (Liu et al., 2010). Also, it keeps the high efficiency of the conveyor assembly line. Detailed introductions to *seru* production can be found in Yin, Kaku, and Stecke (2008), Liu et al. (2010), Stecke, Yin, Kaku, and Murase (2012), Liu, Stecke, Lian, and Yin (2014). Although '*seru*' is the Japanese pronunciation for cell, *seru* production systems differ from the cellular manufacturing systems which have been investigated for more than 30 years in many aspects. Comparisons between these two types of work-cell-based manufacturing systems have been investigated in Isa and Tsuru (2002), Sakazume (2005), Miyake (2006), Liu et al. (2010), Stecke et al. (2012), Yin, Stecke, Swink, and Kaku (2017).

Multi-skilled workers and reconfigurable *serus* which are composed of movable workstations and light-weight equipment enable *seru* production systems to make quick responses to unexpected changes. In highly uncertain and competitive environments, multi-skilled workers may act as a potential capacity buffer against varied demands (Nembhard & Bentefouet, 2014). The benefits that manufacturers can gain from multi-skilled workers mainly depend on how to effectively assign workers to appropriate tasks that they are qualified to perform (Gomar, Haas, & Morton, 2002). Generally, the competency of multi-skilled worker differs from one another in the skill set and proficiency level (Bokhorst, Slomp, & Gaalman, 2004). Differences in the proficiency level further affect the processing time and output rate, even the quality level (Mital et al., 1999; Norman, Tharmmaphornphilas, Needy, Bidanda, & Warner, 2002). However, the vast majority of previous research on the multi-skilled worker assignment ignores the heterogeneity inherent in multi-skilled workers. Moreover, the workload balance among multi-skilled workers has received little attention. For multi-skilled workers who may take charge of more than one task simultaneously, the workload balance is desirable from a social, psychological, and organizational viewpoint by avoiding overuse of

* Corresponding author.
E-mail address: liuc@nwpu.edu.cn (C. Liu).

https://doi.org/10.1016/j.cie.2018.02.035
Received 21 July 2016; Received in revised form 22 January 2018; Accepted 20 February 2018
Available online 22 February 2018

sophisticated workers and enhancing feelings of interpersonal justice and equity (Cesaní & Steudel, 2005; Kim & Nembhard, 2013). It's worth noting that the interpersonal justice and equity is crucial in improving social impacts of manufacturing enterprise and achieving sustainability (Zhang & Haapala, 2015).

In this paper, a multi-skilled worker assignment problem is investigated in the context of *seru* production system, in which the inter-*seru* and inter-worker workload balances are represented as the objectives, and differences in workers' skill sets and proficiency levels are involved. Worker grouping, cell loading, and task assignment problems are solved concurrently.

The remainder of this paper is organized as follows. Section 2 reviews the related research on multi-skilled worker assignment problems. A bi-objective mathematical model is proposed in Section 3. Section 4 presents the proposed algorithm based on NSGA-II. Numerical examples and computational results by the developed algorithm are shown in Section 5. Based on the computational results, the impact of differences in workers' competency on the workload balance is analyzed. The final section draws the conclusions and provides some suggestions for further research.

2. Literature review

Most previous research related to the multi-skilled worker allocation focused on scheduling problems in the service industry and project management, where the aim is to construct a work time table which shows when tasks are to be operated or to generate shift schedules for workers. Since the main point of this paper is to study how to assign workers to *serus* and tasks appropriately, this literature review mainly explores the research devoted to multi-skilled worker assignment problems in the manufacturing industry.

2.1. Multi-skilled worker assignment problems

A considerable amount of research has been conducted on multi-skilled worker assignment problems. The majority of existing research works investigated multi-skilled worker assignment problems by combining it with cell loading, product sequencing, cell formation, and other problems. These problems were usually solved in a sequential manner or concurrently.

Several studies considered the training problem in the context of multi-skilled worker assignment. McDonald, Ellis, Van Aken, and Patrick Koelling (2009) presented a worker-task assignment model to simultaneously determine the assignment of workers to tasks and the training necessity for workers who cannot meet the skill requirement of tasks. Liu et al. (2013) investigated the worker training and assignment problem when a conveyor assembly line is reconfigured into several *serus*. They proposed a three-stage heuristic algorithm to obtain worker-to-*seru* assignment plan and task-to-worker training plan. Olivella, Corominas, and Pastor (2013) formulated a model to assign a set of tasks to a set of workers by considering the performance of cross-trained workers. To solve the dynamic cell formation, operator assignment, and inter/intra cell layouts with machine duplication simultaneously, Mehdizadeh and Rahimi (2016) presented an integrated mathematical model in which a worker can be trained to operate a specific machine in a production period with a training cost. However, the reduced availability and worker absence during training period are not incorporated in most studies (De Bruecker, Van den Bergh, Beliën, & Demeulemeester, 2015). For instance, McDonald et al. (2009) assumed that the task must be performed by a worker from other areas or a temporary worker with the required skill depth level while the assigned worker is receiving training, and Mehdizadeh and Rahimi (2016) assumed that the training is performed between periods and it takes no time.

In labor intensive cells, where most of operations require continuous attendance and involvement of operators, the worker performance directly affects the output of cells (Süer & Dagli, 2005). As a result, the assignment of multi-skilled workers is usually involved with the cell loading and product sequencing. Süer and Dagli (2005) solved the product sequencing problem and the cell loading problem in labor-intensive cells with the consideration of intra-cell manpower transfers. Later, this research was extended by Süer, Cosner, and Patten (2009) to a new case in which a three-phase method including manpower allocation, cell loading, and product sequencing is proposed. Süer, Kamat, Mese, and Huang (2013) developed a two-step method to determine the manpower allocation plan and the cell loading plan for labor intensive cells. Egilmez, Erenay, and Süer (2014) addressed a stochastic skill-based manpower allocation and cell loading problem in a labor intensive jewelry manufacturing system, and the manpower levels, production rates, and manpower configuration are optimized hierarchically by using a four-step method. Due to the high complexity in solving worker assignment, cell loading, and product sequencing concurrently, the above research works solved these problems or sequentially in spite of the strong inter-relationship between them.

In cellular manufacturing systems, the human-related issue is usually neglected in the existing models for cell formation problems (Rafiei & Ghodsi, 2013; Zohrevand, Rafiei, & Zohrevand, 2016). Recently, some researchers started to exploit the integration of worker assignment problem and cell formation problem as a way to increase the productivity of cellular manufacturing systems. The representative mathematical paper which was devoted to human-related aspects of a dynamic cell formation problem is due to Aryanezhad, Deljoo, and Mirzapour Al-e-hashem (2009). They proposed a new mathematical model which solves the dynamic cell formation and the worker assignment problem concurrently. A new mathematical model capturing the capability of workers in doing different jobs is presented by Mahdavi, Aalaei, Paydar, and Solimanpur (2012) for solving the cell formation problem based on a three-dimensional machine-part-worker incidence matrix. Moreover, the new concept of exceptional elements is discussed to show the interpretation of inter-cell movements of both workers and parts for processing on corresponding machines. Aalaei, Paydar, and Saidi-Mehrabad (2014) developed an integer mathematical programming model to design the cellular manufacturing systems with worker flexibility by the method of data envelopment analysis. Bagheri and Bashiri (2014) investigated a comprehensive problem incorporating with the cell formation problem, the inter-cell layout problem, and the operator assignment problem in a dynamic cellular manufacturing environment. Liu, Wang, Leung, and Li (2016b) considered a joint decision model of both multi-skilled workers and multi-functional machines grouping and task scheduling under a dual-resource constrained setting. Since automated machines used in cellular manufacturing systems do not require much involvement of workers, more attention has been paid to the influence of worker assignment on inter-cell movements rather than on the output of cells.

Although the multi-skilled worker assignment problem has been integrated into various problems in numerous studies, the majority of them aim at improving the productivity or the flexibility of manufacturing systems. Table 1 summarizes objective functions used in recent studies which have dealt with multi-skilled worker assignment problems in manufacturing environment. According to this table, we can see that most of these studies concentrate solely on optimizing productivity, cost-related and time-related criteria. Nevertheless, more attention should be paid to the interpersonal justice and equity considering the marked differences in competency among multi-skilled workers.

2.2. Heterogeneity of multi-skilled workers

The heterogeneity inherent in individual capabilities would increase the problem complexity if human-related issues are integrated into problems, especially for cross-trained workers who acquire multiple skills (Othman, Bhuiyan, & Gouw, 2012). The heterogeneity of

Table 1
Summary of objective functions used in recent studies on multi-skilled worker assignment problems.

Work	OR	MC	IM	IC	RC	PC	PPC	UC	OC	LSC	MBC	BC	FC	LC	TC	HC	SC	IT	TN	WT	WN	UT	WB	FT	LO	FA
Jaber and Neumann (2010)																		*								*
Mahdavi et al. (2010)		*	*	*	*							*				*	*									
Sirovetnukul and Chutima (2010)																				*	*		*			
Ghotboddini et al. (2011)		*	*		*	*			*	*								*								
Mahdavi et al. (2011)			*	*								*														
Mahdavi et al. (2012)			*																							
Nikoofarid and Aalaei (2012)				*								*														
Othman et al. (2012)									*						*	*									*	
Aalaei and Shavazipour (2013)			*									*														
Azizi and Liang (2013)													*	*	*											
Rafiei and Ghodsi (2013)		*	*		*	*			*	*												*				
Süer et al. (2013)	*																		*							
Bagheri and Bashiri (2014)			*		*										*	*	*									
Egilmez et al. (2014)	*																				*					
Nembhard and Bentefouet (2014)	*																									
Givi et al. (2015)				*								*														
Hewitt et al. (2015)	*																									
Liu et al. (2016a)				*								*														
Liu et al. (2016b)		*	*														*									
Mehdizadeh et al. (2016)		*	*		*		*	*							*	*	*	*								
Mehdizadeh and Rahimi (2016)		*	*			*			*						*	*	*									
Sakhaii et al. (2016)			*	*	*						*	*			*	*										
Ferjani et al. (2017)																								*		

Note: OR: output rate; MC: machine fixed/variable cost; IM: inter/intra-cell movement (cost); IC: inventory holding cost; RC: machine relocation cost; PC: machine purchasing/selling cost; PPC: production planning cost; UC: set up cost; OC: overtime cost; LSC: labor shifting cost; MBC: machine breakdown cost; BC: backorder cost; FC: flexibility cost; LC: productivity loss cost; TC: training cost; HC: hiring/firing cost; SC: salary cost; IT: idle time; TN: tardiness; WT: walking time; WN: the number of workers; UT: worker/machine utilization; WB: inter-worker workload balance; FT: flow time; LO: the top performers laid off; FA: fatigue.

competency for multi-skilled workers is mainly embodied in two aspects: skill set and proficiency level. The skill set is a set of skills that a worker is qualified to perform, and the proficiency level measures to what extent the worker is qualified.

The differences among workers were neglected in many research articles related to multi-skilled worker assignment problems, such as works by Süer et al. (2013), Rafiei and Ghodsi (2013), Egilmez et al. (2014), Zohrevand et al. (2016). In this case, the number of workers assigned to each cell, task or workstation is determined, while the correspondence that which worker is assigned to a specific task is not considered. Such problems are easier to handle since the complexity are considerably reduced (Attia, Edi, & Duquenne, 2012). These studies shows that workers who are capable of processing all tasks and all workers can take the same time to perform a task. However, the measure of completely cross-trained workers is usually not taken in practice. It has been well documented that workers should not necessarily be completely cross-trained for all skills in view of the high training costs (Gel, Hopp, & Van Oyen, 2007; Hopp & Van Oyen, 2004; McDonald et al., 2009; Slomp & Molleman, 2002). Furthermore, the insufficiency and frequent rotating of overtrained workers brings adverse effects on the productivity and quality, especially for complex products (Inman, Jordan, & Blumenfeld, 2004; Jordan, Inman, & Blumenfeld, 2004; Kaku, Murase, & Yin, 2008; Nembhard & Bentefouet, 2014).

In Firat, Hurkens, and Laugier (2014), skills of technicians have a hierarchical structure. That means a technician is also qualified at skills which have lower levels if he/she is proficient in a high level skill. Mehdizadeh, Daei Niaki, and Rahimi (2016) assumed that each worker may operate one machine or more and several workers are available for serving a specific machine. In the work by Sakhaii, Tavakkoli-Moghaddam, Bagheri, and Vatani (2016), workers are able to operate multiple machines at the beginning and they can be trained to operate other machines as well. Ferjani, Ammar, Pierreval, and Elkosantini (2017) considered workers with different flexibility levels. The flexibility level indicates the number of machines that the worker can operate. In above studies, workers differ from one another in the skill set, but their skills are defined as binary parameters. Therefore, a worker can perform a skill as a specialist, or he can't perform it at all. In reality, multi-skilled workers also have different proficiency levels as a result of differences in the training time and the cognitive ability among workers.

Süer and Tummaluri (2008) grouped proficiency levels into nine categories where level 5 represents the average skill, level 1 represents the best, and level 9 is the worst. Based on proficiency levels, operation times of workers are computed by using a standard deviation value. Bootaki, Mahdavi, and Paydar (2014) considered the impact of worker skills on the part quality. They classified the process quality of a part into five levels and the integer values of 1–5 represent very bad, bad, medium, well, and very well, respectively. These two studies characterized proficiency levels as discrete integers. In the work of Costa, Cappadonna, and Fichera (2014), three levels of expertise are considered for a worker: senior, normal, and junior. For each level of expertise, multiple proficiency levels can be defined. Attia, Duquenne, and Le-Lann (2014) expressed the performance of an actor by the efficiency which is the ratio of the total work needed for an actor to the standard workload. In the above two works, the proficiency levels are measured by numbers of the interval [0,1], and a worker with the proficiency level equals to 1 is considered as an expert.

2.3. Sharing policies between workers and tasks

The sharing policies between workers and tasks used in multi-skilled worker assignment problems can be classified into three categories: no sharing (one worker can be assigned to only one machine and vice versa), operator sharing (one worker can be assigned to more than one task), and workload sharing (more than one worker can be assigned to one task).

Corominas, Olivella, and Pastor (2010) and Olivella et al. (2013) didn't take sharing into account in assigning tasks to workers, and they even assumed that changes in assignments during a given period are not possible. In the work by Nembhard and Bentefouet (2014), both the number of tasks occupied by a worker and the number of workers assigned to a task are restricted to exactly one. Liu et al. (2016b) assumed that machines and workers are of equal size and one machine and one worker are grouped as a pair and assigned to a workstation. Valeva, Hewitt, Thomas, and Brown (2017) proposed a two-stage stochastic program to assign workers to tasks in which the sharing between workers and tasks is not allowed. However, transferring workers among different tasks may yield several benefits to manufacturing factories and workers such as the improved flexibility and job satisfaction, and the decreased fatigue, injuries, and boredom (Azizi, Zolfaghari, & Liang, 2010).

Assigning one worker to multiple tasks is common in the U-shaped cell/line where tasks are divided into several parts and each part is processed by one worker. Sirovetnukul and Chutima (2010) investigated the worker allocation problem in U-shaped assembly lines considering the walking time of workers who work at different locations. Azadeh, Nazari-Shirkouhi, Hatami-Shirkouhi, and Ansarinejad (2011a), Azadeh, Sheikhalishahi, and Koushan (2013) studied the worker assignment problem in U-shaped manned cells in which the rapid rebalancing is permitted by the walking multi-skilled workers. However, only a single U-shaped cell/line is considered in these studies. The operator sharing policy also can be found in several studies related to dual resource constrained (DRC) systems where the number of workers is less than that of machines or workstations, such as Bootaki et al. (2014), Mehdizadeh et al. (2016), and Mehdizadeh and Rahimi (2016). All of them assumed that workers can operate various machines but each machine can be served only by one worker.

In the manufacturing environment with a high staffing level, workloads can be shared among different workers. All of the afore mentioned studies on worker assignment problems in labor intensive cells, including Süer and Dagli (2005), Süer et al. (2009, 2013), Egilmez et al. (2014), considered the workload sharing policy. In labor intensive cells, the number of workers is much larger than that of machines/tasks, and the output of cells is directly determined by the number of workers assigned to each task. Liu, Wang, and Leung (2016a) incorporated the workload sharing into the worker assignment and production planning in a dynamic cellular manufacturing system. They assumed that at least one worker is assigned to each workstation but workers can only be reassigned to different cells and workstations at the end of each period.

2.4. Solution techniques

Worker assignment problems are combinatorial optimization problems which are generally NP-hard (Ammar, Pierreval, & Elkosantini, 2013). It is difficult to find optimal solutions in the polynomial time in view of the complexity of the problems. As a result, meta-heuristic algorithms have been used in many cases to find acceptable solutions in a reasonable computation time for realistic problems (Costa et al., 2014; De Bruecker et al., 2015). The genetic algorithm seems to be the most used among the meta-heuristic algorithms to address worker assignment problems (Ammar et al., 2013).

Azizi et al. (2010) applied a SAMED algorithm composed of a simulated annealing module, three types of memory, a genetic algorithm component, and a blockage removal feature to a job rotation model with the objective of minimizing the total delay caused by the lack of skill and/or motivation. Azadeh, Nokhandan, Asadzadeh, and Fathi (2011b) used the computer simulation and genetic algorithm to find the optimal operator allocation in a U-shaped cellular manufacturing system. Attia et al. (2014) modeled the problem of multi-period multi-skilled worker allocation with the objective to minimize the related costs and proposed a genetic algorithm as the solution method. Costa et al. (2014) studied a flow-shop group scheduling problem integrated with the multi-skilled worker assignment problem with an objective to minimize the makespan. They developed three meta-heuristics based on a genetic algorithm framework and carried out an extensive comparison among them.

Many studies applied the non-dominated sorting genetic algorithm II (NSGA-II) as a solution approach to solve problems with multiple objective functions. Rafiei and Ghodsi (2013) discussed a dynamic cell formation problem with objectives of minimizing costs and the labor utilization. A hybrid algorithm based on ant colony optimization (ACO) and genetic algorithm is developed to solve the problem. Bootaki et al. (2014) solved a bi-objective mathematical model for grouping machines, parts, and workers into cells by a hybrid GA-augmented ε-constraint method. Niakan et al. (2016) proposed a new hybrid meta-heuristic composed of NSGA-II and Multi-Objective Simulated Annealing (MOSA) to solve the dynamic cell formation problem and the worker assignment problem concurrently with objectives of minimizing production and labor costs and the total production waste. To test the performance of the proposed algorithm, two conventional evolutionary algorithms (NSGA-II and MOSA) were compared based on different metrics. Zohrevand et al. (2016) addressed a dynamic cell formation problem aiming at minimizing related costs and maximizing the worker utilization by using a hybrid tabu search-genetic algorithm (TS-GA).

3. Problem formulation

The problem to be solved in this paper concentrates on grouping multi-skilled workers into *serus*, assigning batches to *serus*, and assigning workers to tasks concurrently in a *seru* production system with the consideration of the worker heterogeneity. Workers differ from one another in skill sets and proficiency levels. Skill sets mainly affect the number of possible worker-task combinations, while proficiency levels determine which worker can be chosen from several available workers to process a task based on their influence on the processing time. Overlap among workers' skill sets is allowed, thus more than one worker is available for some tasks. Each worker may take charge of more than one task of a batch and visit each of the corresponding workstations once a cycle. The whole production process of a product must be fulfilled within a single *seru* without inter-*seru* product movements and worker transfers. But workers can be assigned to different tasks when a new batch is started.

The processing time of a task varies with workers. It can be represented as the product of the standard processing time and the proficiency level of a worker who performs the task. In the proposed problem, proficiency levels are not limited to [0, 1] in order to better quantify productivity losses and gains resulted from unskilled workers and skilled workers. The standard processing time means the time required by a worker with proficiency level 1 to perform a specific task. A skilled worker with the proficiency level lower than 1 is able to perform the task faster than the standard processing time, whereas an unskilled worker whose proficiency level is greater than 1 needs more time than the standard processing time.

3.1. Assumptions

We make some assumptions for the research problem as follows.

- The demand of each product type is known and deterministic.
- The available time of each worker is known.
- The upper bound of the cell size is known and deterministic.
- The standard processing time for all tasks of a product type are known and deterministic.
- Each batch should be processed in a single *seru*. Lot splitting is not considered.
- Workers can only be assigned to tasks at which they are competent. The worker training is not considered.
- The proficiency levels of workers are static. Learning and forgetting are not considered.
- A worker can be assigned to only one *seru*. The worker transfer between *serus* is not considered.
- For each task, only one worker can be assigned to it.
- A worker can be assigned to more than one task based on his/her ability.

- For each batch processed in a *seru*, each worker assigned to the *seru* should be assigned to at least one task.
- Workers' walking distance and time between different workstations are negligible in view of the compact layout of workstations in *serus*.
- The required cost and time for changing from one task to another are zero.

3.2. Notation

Indices:

p	the index for product types ($p = 1,...,P$);
b	the index for batches ($b = 1,...,B$);
c	the index for *serus* ($c = 1,...,C$);
w	the index for workers ($w = 1,...,W$);
s	the index for tasks/skills ($s = 1,...,S$);

Input parameters:

P	the number of product types;
B	the number of batches;
C	the number of *serus*;
W	the number of workers;
S	the number of tasks/skills;
V_b	the volume of batch b;
t_p^s	the standard processing time for task s of product type p;
f_w^s	the proficiency level of worker w on task s;
N	the upper bound of the number of workers assigned to each *seru*;
M	the upper bound of the number of tasks assigned to each worker;
G	the available time of each worker;
x_b^p	1, if batch b belongs to product type p; 0, otherwise;
y_p^s	1, if task s is required by product type p; 0, otherwise;
e_w^s	1, if worker w is competent at skill s; 0, otherwise;

•*Decision variables*

l_b^c	1, if batch b is assigned to *seru* c; 0, otherwise;
a_w^c	1, if worker w is assigned to *seru* c; 0, otherwise;
z_w^{bs}	1, if worker w performs task s of batch b; 0, otherwise.

3.3. Mathematical model

With the above notations, we define $\overline{C} = \frac{1}{C}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s l_b^c a_w^c z_w^{bs} t_p^s f_w^s V_b$, and $\overline{W} = \frac{1}{W}\sum_{w=1}^{W}\sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s l_b^c a_w^c z_w^{bs} t_p^s f_w^s V_b$. The research problem can be modeled as below.

Minimize

$$WB1 = \frac{1}{C}\sum_{c=1}^{C}\left(\sum_{w=1}^{W}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s l_b^c a_w^c z_w^{bs} t_p^s f_w^s V_b - \overline{C}\right)^2, \tag{1}$$

$$WB2 = \frac{1}{W}\sum_{w=1}^{W}\left(\sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s l_b^c a_w^c z_w^{bs} t_p^s f_w^s V_b - \overline{W}\right)^2, \tag{2}$$

subject to

$$\sum_{c=1}^{C} l_b^c = 1 \quad \forall\, b, \tag{3}$$

$$\sum_{c=1}^{C} a_w^c = 1 \quad \forall\, w, \tag{4}$$

$$z_w^{bs} \leqslant e_w^s \quad \forall\, w,b,s, \tag{5}$$

$$\sum_{w=1}^{W} z_w^{bs} = \sum_{p=1}^{P} x_b^p y_p^s \quad \forall\, b,s, \tag{6}$$

$$\sum_{w=1}^{W} a_w^c \leqslant N \quad \forall\, c, \tag{7}$$

$$\sum_{s=1}^{S} z_w^{bs} \leqslant M \quad \forall\, b,w, \tag{8}$$

$$\sum_{w=1}^{W} a_w^c e_w^s \geqslant \sum_{p=1}^{P} x_b^p y_p^s \quad \forall\, c,s,b, \tag{9}$$

$$z_w^{bs} \leqslant \sum_{c=1}^{C} l_b^c a_w^c \quad \forall\, w,b,s, \tag{10}$$

$$\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s z_w^{bs} \geqslant \sum_{c=1}^{C} l_b^c a_w^c \quad \forall\, b,w, \tag{11}$$

$$\sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s l_b^c a_w^c z_w^{bs} t_p^s f_w^s V_b \leqslant G \quad \forall\, w, \tag{12}$$

$$l_b^c, a_w^c, z_w^{bs} \in \{0,1\} \quad \forall\, c,b,w,s. \tag{13}$$

Objective function (1) minimizes the deviations from the average workload of *serus*. Considering the operation sequence of tasks and various plans for dividing tasks, we assume that only one workpiece can be processed in a *seru* at any time, that is a workpiece can start to be processed only when the previous one is finished. Therefore, the flow time of a batch is the sum of processing times of each workpiece. Objective function (2) minimizes the deviations from the average workload of workers.

Constraints (3) and (4) ensure that each batch and each worker can only be assigned to one *seru*. Constraint (5) guarantees that workers can only be assigned to tasks at which they are competent. Constraint (6) ensures that each task of a batch can only be performed by one worker. Constraints (7) and (8) specify the upper bounds of the number of workers assigned to each *seru* and the number of tasks for a given batch assigned to each worker. For each task of a specific batch assigned to a *seru*, constraint (9) guarantees that at least one worker assigned to the *seru* is competent at it. Constraint (10) guarantees that workers can only be assigned to tasks which belong to batches assigned to the same *seru* as them. Constraint (9), together with constraint (10), can avoid inter-*seru* product movement and worker transfer. Constraint (11) guarantees that each worker assigned to a *seru* should be assigned to at least one task for each batch processed in the same *seru*. Constraint (12) assures that available times of workers are not exceeded.

3.4. Linearization

The mathematical model proposed in this paper is a non-linear model. In the following, the proposed model is reformulated as a pure 0–1 linear programming model by introducing some new variables with auxiliary constraints according to the linearization techniques from Bagheri and Bashiri (2014).

Step 1

Consider the pure quadratic 0–1 term $Z = X_1 \times X_2 \times ... \times X_n$ where $X_i (i = 1,...,n)$ is a binary variable. It is obvious that Z is 1 if and only if all the variables are 1 and otherwise it must be 0. Therefore, some new auxiliary constraints can be introduced.

$$Z \leqslant X_i \quad \forall\ i = 1,...,n,$$

$$Z \geqslant \sum_{i=1}^{n} X_i-(n-1).$$

We define the following new binary variables:

$$D_{c,b,w} = l_b^c a_w^c \quad \forall\ c,b,w;$$

$$H_{c,b,w,s} = D_{c,b,w} z_w^{bs} \quad \forall\ c,b,w,s.$$

By considering these equations, the following constraints should be added to the proposed model:

$$D_{c,b,w} \leqslant l_b^c \quad \forall\ c,b,w; \tag{14}$$

$$D_{c,b,w} \leqslant a_w^c \quad \forall\ c,b,w; \tag{15}$$

$$D_{c,b,w} \geqslant l_b^c + a_w^c-1 \quad \forall\ c,b,w; \tag{16}$$

$$H_{c,b,w,s} \leqslant D_{c,b,w} \quad \forall\ c,b,w,s; \tag{17}$$

$$H_{c,b,w,s} \leqslant z_w^{bs} \quad \forall\ c,b,w,s; \tag{18}$$

$$H_{c,b,w,s} \geqslant D_{c,b,w} + z_w^{bs}-1 \quad \forall\ c,b,w,s; \tag{19}$$

Step 2

According to Bhaskar and Srinivasan (1997), minimizing the sum of squares of the deviations from the average workload can be approximated to minimizing the difference between the maximum and minimum workloads. Therefore, the objective functions of the proposed model can be presented as follows:

$$\text{Minimize} \quad WB1 = \frac{1}{C}\left[\max_c\left(\sum_{w=1}^{W}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b\right) -\min_c\left(\sum_{w=1}^{W}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b\right)\right]$$

$$\text{Minimize} \quad WB2 = \frac{1}{W}\left[\max_w\left(\sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b\right) -\min_w\left(\sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b\right)\right]$$

The objective functions can be linearized by replacing an additional variable and two auxiliary constraints as follows:

$$\begin{array}{lll} \min\ T & & \min\ Z \\ St: & \rightarrow & St: \\ T = \max(X,0) & & Z \geqslant X \\ & & Z \geqslant 0 \end{array}$$

We define the following new variables:

$$I = \max_c\left(\sum_{w=1}^{W}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b\right);$$

$$J = \min_c\left(\sum_{w=1}^{W}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b\right);$$

$$K = \max_w\left(\sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b\right);$$

$$U = \min_w\left(\sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b\right).$$

By considering these equations, the following auxiliary constraints should be added to the proposed model:

$$I \geqslant \sum_{w=1}^{W}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b \quad \forall\ c; \tag{20}$$

$$J \leqslant \sum_{w=1}^{W}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b \quad \forall\ c; \tag{21}$$

$$K \geqslant \sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b \quad \forall\ w; \tag{22}$$

$$U \leqslant \sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b \quad \forall\ w. \tag{23}$$

Thus, the objective functions of the final version of the linear 0–1 programming model can be presented as follows.

Minimize

$$WB1 = \frac{1}{C}(I-J), \tag{24}$$

$$WB2 = \frac{1}{W}(K-U), \tag{25}$$

Next we introduce the constraints for this reformulated model as follows. The first class of constraints are set constraints (3)–(9) and constraints (14)–(23) shown in above. The second class of constraints are the following set constraints which are new to replace of some original set constraints. Replace set constraint (10) by

$$z_w^{bs} \leqslant \sum_{c=1}^{C} D_{c,b,w} \quad \forall\ w,b,s, \tag{26}$$

Replace set constraint (11) by

$$\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s z_w^{bs} \geqslant \sum_{c=1}^{C} D_{c,b,w} \quad \forall\ b,w, \tag{27}$$

Replace set constraint (12) by

$$\sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{s=1}^{S} x_b^p y_p^s e_w^s H_{c,b,w,s} t_p^s f_w^s V_b \leqslant G \quad \forall\ w, \tag{28}$$

Replace set constraint (13) by

$$l_b^c, a_w^c, z_w^{bs}, D_{c,b,w}, H_{c,b,w,s} \in \{0,1\}, \quad I,J,K,U \geqslant 0. \tag{29}$$

The number of variables and constraints in the linearized model are presented in Tables 2 and 3, respectively.

4. An algorithm based on NSGA-II

Worker assignment problems in manufacturing systems are complex and they are often NP-hard, even for some problems containing only a single objective function and homogeneous skills (Ammar et al., 2013; De Bruecker et al., 2015). The proposed problem considering workers' differences in skill sets and proficiency levels is NP-hard too. Considering the complexity of the proposed mathematical model, we develop an algorithm based on NSGA-II in this section. NSGA-II, proposed

Table 2
The number of variables in the linearized model.

Variable	Count	Variable	Count
l_b^c	$B \times C$	$D_{c,b,w}$	$C \times B \times W$
a_w^c	$W \times C$	$H_{c,b,w,s}$	$C \times B \times W \times S$
z_w^{bs}	$W \times B \times S$	I,J,K,U	4
$Sum = (B \times C) + (W \times C) + (W \times B \times S) + (C \times B \times W) + (C \times B \times W \times S) + 4$			

Table 3
The number of constraints in the linearized model.

Constraint	Count	Constraint	Count	Constraint	Count
(3)	B	(14)	$C \times B \times W$	(21)	C
(4)	W	(15)	$C \times B \times W$	(22)	W
(5)	$W \times B \times S$	(16)	$C \times B \times W$	(23)	W
(6)	$B \times S$	(17)	$C \times B \times W \times S$	(26)	$W \times B \times S$
(7)	C	(18)	$C \times B \times W \times S$	(27)	$B \times W$
(8)	$B \times W$	(19)	$C \times B \times W \times S$	(28)	W
(9)	$C \times B \times S$	(20)	C	(29)	$(B \times C) + (W \times C) + (W \times B \times S) + (C \times B \times W) + (C \times B \times W \times S) + 4$
$Sum = 4(C \times B \times W \times S) + 4(C \times B \times W) + 3(W \times B \times S) + (C \times B \times S) + 2(B \times W) + (B \times S) + (B \times C) + (W \times C) + B + 4W + 3C + 4$					

by Deb, Pratap, Agarwal, and Meyarivan (2002), is a Pareto-based multi-objective meta-heuristic algorithm. It has been used extensively to solve multi-objective optimization problems for better spread of solutions and better convergences used to search a near-optimal Pareto frontier.

4.1. Encoding scheme

In order to represent the batch-*seru*, the worker-*seru*, and the worker-task assignment results in one chromosome, a double-layered hierarchical chromosome structure is developed. The chromosome has $B + W$ genes in the first layer where B is the number of batches and W is the number of workers. For the second layer, the length of the chromosome is the sum of tasks required by each batch. Let the number of tasks of batch b be n_b, then the length of the chromosome is $\sum_{b=1}^{B} n_b$ in the second layer. The allele of genes in the first layer is the index of *seru* to which a batch or a worker has been assigned, whereas that in the second layer is the index of worker who has been assigned to a task. The worker-task assignment results for each batch are determined based on the batch-*seru* and the worker-*seru* assignment results provided by the first layer.

For illustration, consider a data set with 7 batches and 10 workers to be assigned to 3 *serus*. The numbers of tasks of these batches are 3, 4, 5, 3, 3, 5, and 4, respectively. Fig. 1 shows the chromosome for a solution. There are 17 genes in the first layer and 27 genes in the second layer. Genes in the first layer mean that *seru 1* contains worker {2, 5, 6, 9} and processes batch {2, 3, 7}, *seru 2* contains worker {1, 3, 8} and processes batch {1, 5}, and *seru 3* contains workers {4, 7, 10} and processes batch {4, 6}. Genes in the second layer represent that task 1 of batch 1 is processed by worker 8, and task 2 of batch 1 is processed by worker 1, and task 3 of batch 1 is processed by worker 3, and task 1 of batch 2 is processed by worker 6, and so on.

4.2. Initialization of the population

The initial population is generated only once with the desired population size either randomly or by using a given heuristic at the beginning of the genetic algorithm. A special procedure is introduced in this section to generate a random initial population. Chromosomes created in accordance with the procedure can satisfy constraints (3)–(5) and constraint (10) of the proposed mathematical model and ensure each *seru* has at least one batch and one worker. Assuming that B batches and W workers to be assigned to C *serus*, each chromosome of the initial population is created according to the following procedure.

The first layer of the chromosome is generated firstly. For each *seru* c ($c = 1,2,\ldots,C$), randomly select a batch and a worker, then assign them to the *seru*. For the remaining *(B-C)* unassigned batches, randomly select a batch and a *seru*, then assign the selected batch to the selected *seru* until all batches are assigned. The remaining *(W-C)* unassigned workers are treated in the same way. Next the second layer of the chromosome is generated. For task s of batch b which is assigned to *seru* c, randomly select a worker who is qualified to perform the task from *seru* c, then assign the selected worker to the task. If there is no such a worker, then set the allele of the corresponding gene to 0. The procedure is iterated until the given population size is achieved.

4.3. Feasibility correction

Illegal chromosomes may take a relatively large portion, especially for highly constrained problems, in the randomly generated initial population and the new population which has been modified by genetic operators (Gen & Cheng, 1999). For chromosomes violate constraints (6)–(9) and constraint (11), a feasibility correction procedure is proposed to correct them by adjusting the allele of genes. For chromosomes violate constraint (12), a penalty function is employed to handle them considering the difficulty in adjustment and the influence on objective function values.

The feasibility correction procedure adjusts illegal chromosomes in accordance with the following rules. For the task to which no worker is assigned, reassign the corresponding batch to other *serus*, or transfer a worker who is able to perform it from other *serus* to the corresponding *seru* if the corresponding *seru* has only one batch. For the *seru* where the number of workers exceeds the upper bound, transfer the redundant worker(s) to other *serus*. For the worker who takes charge of more tasks than the upper bound for the certain batch, reassign the redundant task(s) to other qualified workers who take charge of less tasks of the same batch in the same *seru*, or transfer a worker who is able to perform the redundant task(s) from other *serus* to the corresponding *seru* if the number of tasks assigned to other workers in the same *seru* all exceeds the upper bound. For the worker who is assigned to no task of a given batch, reassign a task from other workers who are assigned to more tasks to the worker.

4.4. Fitness evaluations

Fitness functions used in the present algorithm are made up of

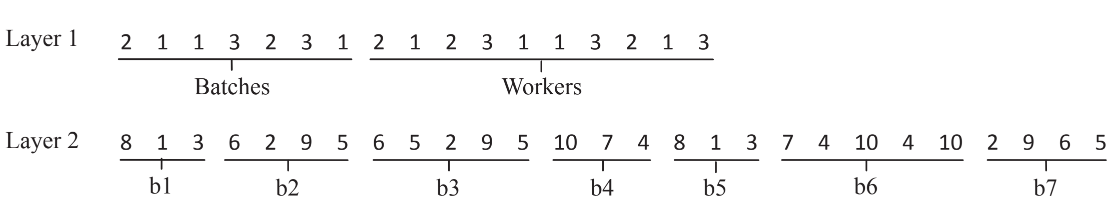

Fig. 1. An example of chromosome representation.

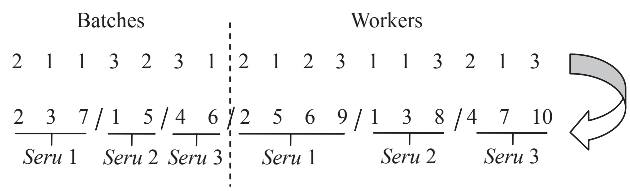

Fig. 2. A chromosome representation with group structure.

P 1 3 1 / 2 4 6 / 5 7 ¦ 6 7 / 1 4 5 8 9 / 2 3 10

P 2 2 3 7 / 1 5 / 4 6 ¦ 2 5 6 9 / 1 3 8 / 4 7 10

C 1 3 1 / 1 5 / 5 7 ¦ 6 7 / 1 3 8 / 2 3 10

C 2 2 3 7 / 2 4 6 / 4 6 ¦ 2 5 6 9 / 1 4 5 8 9 / 4 7 10

C 1` 3 / 1 5 / 7 ¦ 6 7 / 1 3 8 / 2 10

C 2` 3 7 / 2 4 6 / ¦ 2 6 / 1 4 5 8 9 / 7 10

C 1* 3 2 6 / 1 5 / 7 4 ¦ 6 7 4 5 9 / 1 3 8 / 2 10

C 2* 3 7 / 2 4 6 / 1 5 ¦ 2 6 3 / 1 4 5 8 9 / 7 10

Fig. 3. A numerical illustration of the crossover operator.

objective functions of the proposed mathematical model and a penalty function. The penalty function is able to transform a constrained problem into an unconstrained one by incorporating constraints into objective functions. The insertion of penalty value reduces the possibility of illegal chromosomes being selected to produce offspring. In this paper, the penalty function proposed by Gen and Cheng (1999) is adapted.

For a minimization problem, the better a chromosome is, the lower its fitness value must be. Therefore, the penalty function $p(i)$ constructed in the multiplication form should satisfy

$$\begin{cases} p(i) = 1, & \text{if chromosome } i \text{ is legal;} \\ p(i) > 1, & \text{otherwise.} \end{cases} \tag{30}$$

Given a chromosome i in the current population P(t), the adapted penalty function is constructed as follows.

$$p(i) = 1 + \left(\frac{\Delta b(i)}{\Delta b^{max}}\right)^k, \tag{31}$$

$$\Delta b(i) = max\{0, g(i) - G\}, \tag{32}$$

$$\Delta b^{max} = max\{\varepsilon, \Delta b(i) | i \in P(t)\}, \tag{33}$$

where $\Delta b(i)$ is the value of violation for Eq. (12), Δb^{max} is the maximal violation among chromosomes in the current population, k is a user-

C 1 2 1 1 3 2 1 3 2 3 2 1 1 1 1 2 1 3

C 1* 2 1 1 3 2 1 3 2 3 2 1 1 3 1 2 1 3

Fig. 4. A numerical illustration of the mutation operator.

Table 4
The competency of workers.

Skill	1	2	3	4
Worker 1	1.04	–	0.99	1.07
Worker 2	0.90	0.93	1.06	1.09
Worker 3	–	0.98	1.00	0.93
Worker 4	0.93	1.06	–	0.91
Worker 5	0.96	1.05	0.92	–

Table 5
The processing time of each task for all product types.

Task	1	2	3	4
Product type 1	3.51	2.51	3.04	–
Product type 2	3.49	–	3.13	2.29
Product type 3	–	1.57	–	3.05

Table 6
The product type and demand volume for each batch.

Batch	1	2	3	4	5
Product type	2	1	3	1	3
Demand volume	24	30	36	29	36

specified parameter which is used to adjust the severity of penalty function, G is the available time of each worker, and ε is a small positive number avoiding zero-division. Then, the fitness functions for each chromosome can be stated as

$$\begin{cases} eval^1(i) = WB1(i)p(i); \\ eval^2(i) = WB2(i)p(i). \end{cases} \tag{34}$$

4.5. Selection strategy

The selection strategy employed in the present algorithm is the "crowding" tournament selection. Before selection, chromosomes in the population should be divided into several non-dominated fronts and the crowding distance for chromosomes in the same front should be calculated according to the method given by Deb et al. (2002). For any two chromosomes, the one with lower rank is selected. If two chromosomes belong to the same front, the one located in a "less crowded" region, namely with the greater crowding distance, is selected to maintain the diversity of chromosomes in a population.

4.6. Crossover

The crossover operation is only applied to the first layer of chromosomes by reason of the allele of genes in the second layer is determined based on the information provided by the first layer and has no impact on the grouping results. In order to keep the integrity of group structure and minimize the number of illegal chromosomes,

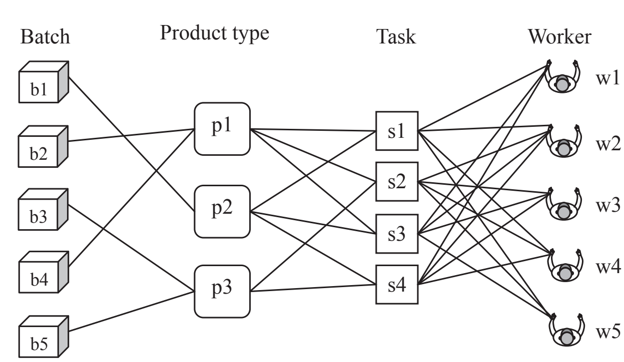

Fig. 5. Links between workers and tasks.

Table 7
Objective functions and components for the illustrative example.

Total	*WB*1	*WB*2	*I*	*J*	*K*	*U*
32.9839	49.9981	15.9697	411.4654	261.4710	261.4710	181.6224

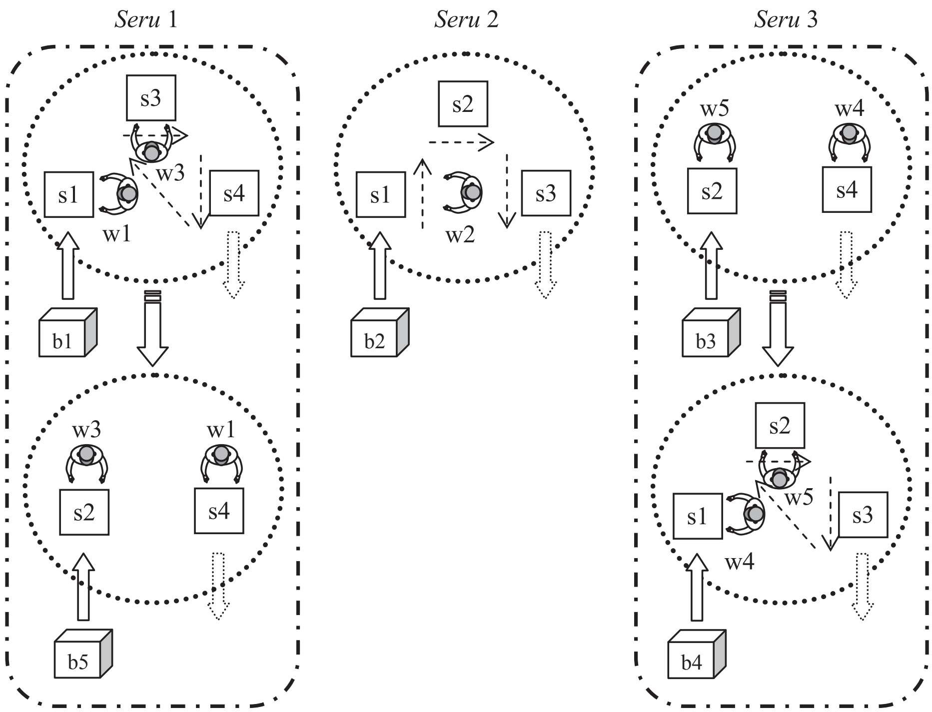

Fig. 6. Assignment details for the illustrative example.

Table 8
Patterns of data generation.

Parameter	Generation pattern
Number of product types	3 ($p1 \sim p3$)
Number of batches	10 ($b1 \sim b10$)
Number of *serus*	3 ($c1 \sim c3$)
Number of workers	10 ($w1 \sim w10$)
Number of tasks/skills	8 ($s1 \sim s8$)
Volume of batches	$\sim U(20,40)$
Standard processing time	$\sim U(1,4)$
Size of skill sets under OW	8
Size of skill sets under EWS & EWSP	6
Proficiency level of workers under OW & EWS	1
Proficiency level of workers under EWSP	$\sim U(0.9,1.1)$
Upper bound for workers assigned to each *seru*	4
Upper bound for tasks assigned to each worker	5
Available time of each worker	2400

chromosomes in the format shown in Fig. 1 should be transformed into the format shown in Fig. 2 before crossover. As we can see from Fig. 2, genes in the first layer are rearranged by the order of *serus*, and the allele of genes stands for the index of batches and workers that have been assigned to *serus*.

The *seru*-swap crossover strategy is adopted. At first, generate an integer random number d between 1 and C (C is the number of *serus*). Then, exchange genes belong to *seru* d in both batch and worker sections to generate two new chromosomes. Thereafter, transform new chromosomes into the format shown in Fig. 1 and determine the allele of genes in the second layer of two new chromosomes based on the allele of genes in the first layer. Fig. 3 illustrates how the crossover operator works. To show the group structure distinctly, genes representing batch-*seru* assignment results and worker-*seru* assignment results are divided by a dotted line, and genes belong to different *serus* are divided by a slash. P1 and P2 are parent chromosomes. Assuming the crossover point is 2, and we get C1 and C2 after exchanging genes belong to *seru* 2.

The crossover operation may cause the occurrence of missing alleles and redundant alleles which occur twice. For example, C1 misses batch 2, 4, 6 and worker 4, 5, 9, but has double batch 1, 5 and double worker 3. To correct these illegal chromosomes, we need to eliminate redundant alleles which are directly copied from parent chromosomes firstly. We get $C1'$ and $C2'$ after eliminating the redundant alleles in C1 and C2. After that, if there exists a *seru* with no batch or no worker, assign a missing batch or worker to it or transfer a batch or worker from other *serus* to it. And then, assign the remaining missing batches to *serus* with less batches and more workers, and assign the remaining missing workers to *serus* with less workers and more batches. Finally, we get $C1^{*}$ and $C2^{*}$.

4.7. Mutation

The mutation operation is only applied to the first layer of chromosomes also. For a selected gene g in the first layer, replace the allele of it with an integer random number between 1 and C, where C is the number of *serus*. Thereafter, the allele of genes in the second layer of the altered chromosome should be updated according to the alteration. Finally, apply the feasibility correction procedure to the new chromosomes. The example given in Fig. 4 illustrates how the mutation operator works. In chromosome *C1*, the allele of gene 13 is replaced with

Table 9
The parameter values of the genetic algorithm.

Parameter	Population size	Max generation	Crossover rate	Mutation rate	Penalty coefficient
Notation	N^{*}	*MAXGEN*	p_c	p_m	k
Value	60	90	0.9	0.025	2

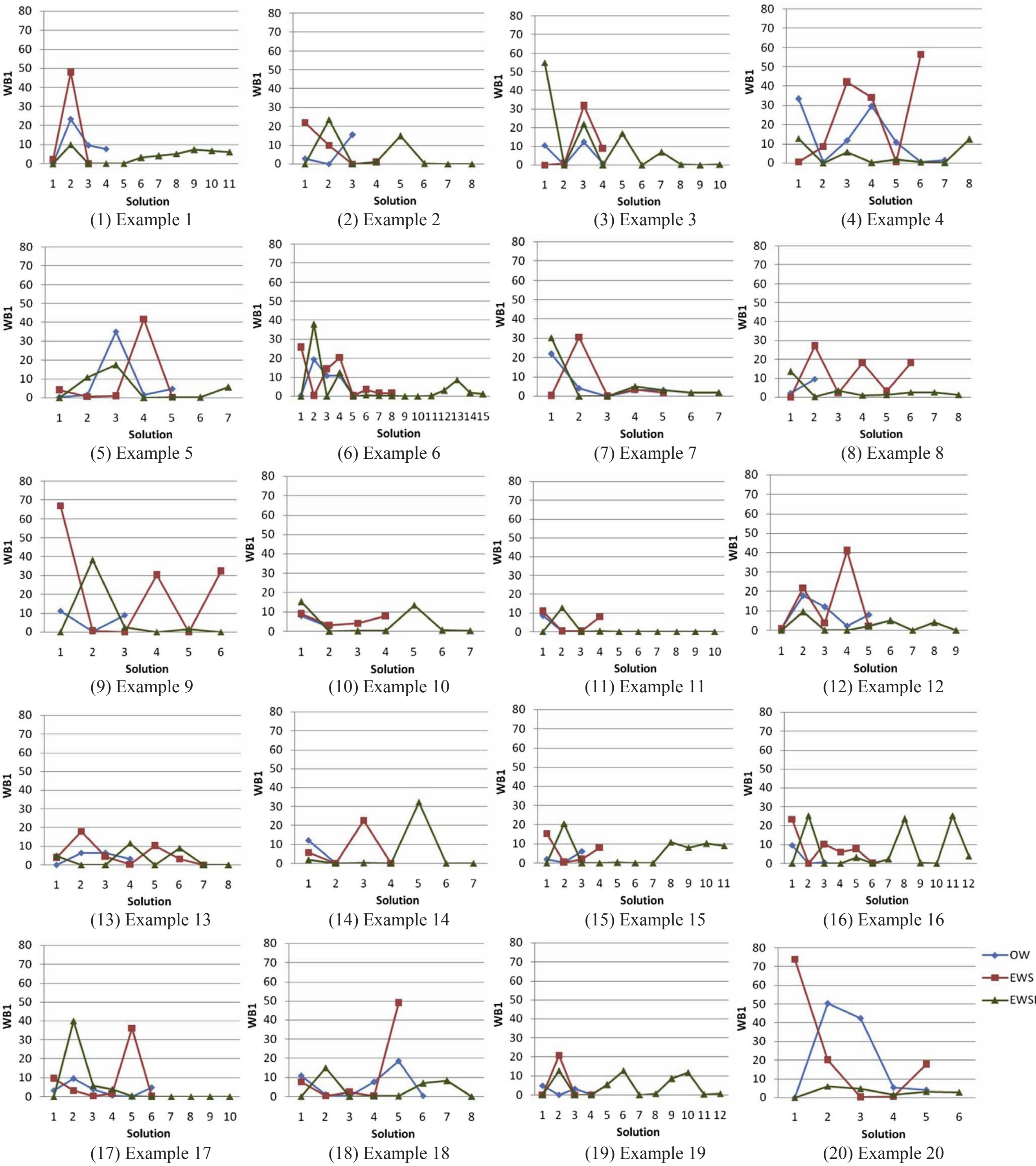

Fig. 7. The comparison of the first objective function's value of non-dominated solutions under different conditions.

3. That means worker 6 assigned to *seru* 1 originally is reassigned to *seru* 3.

4.8. Replacement

Chromosomes of the next generation are selected following the "elitist strategy". Firstly, combine the old generation with new chromosomes generated by the crossover and mutation operators, and form a population R_t of size $2*N^*$ (N^* is the population size), and apply the non-dominated sorting to R_t. Then, select the best N^* chromosomes to form the next generation. Since all previous and current population members are included in R_t, elitism is ensured (Deb et al., 2002).

5. Computational analysis

In order to evaluate the applicability and effectiveness of the proposed model and algorithm, several numerical examples with randomly generated data are provided. A small-size illustrative example is solved by the branch-and-bound (B&B) method using Lingo 11 software firstly and 100 medium-size problem instances are solved by the proposed

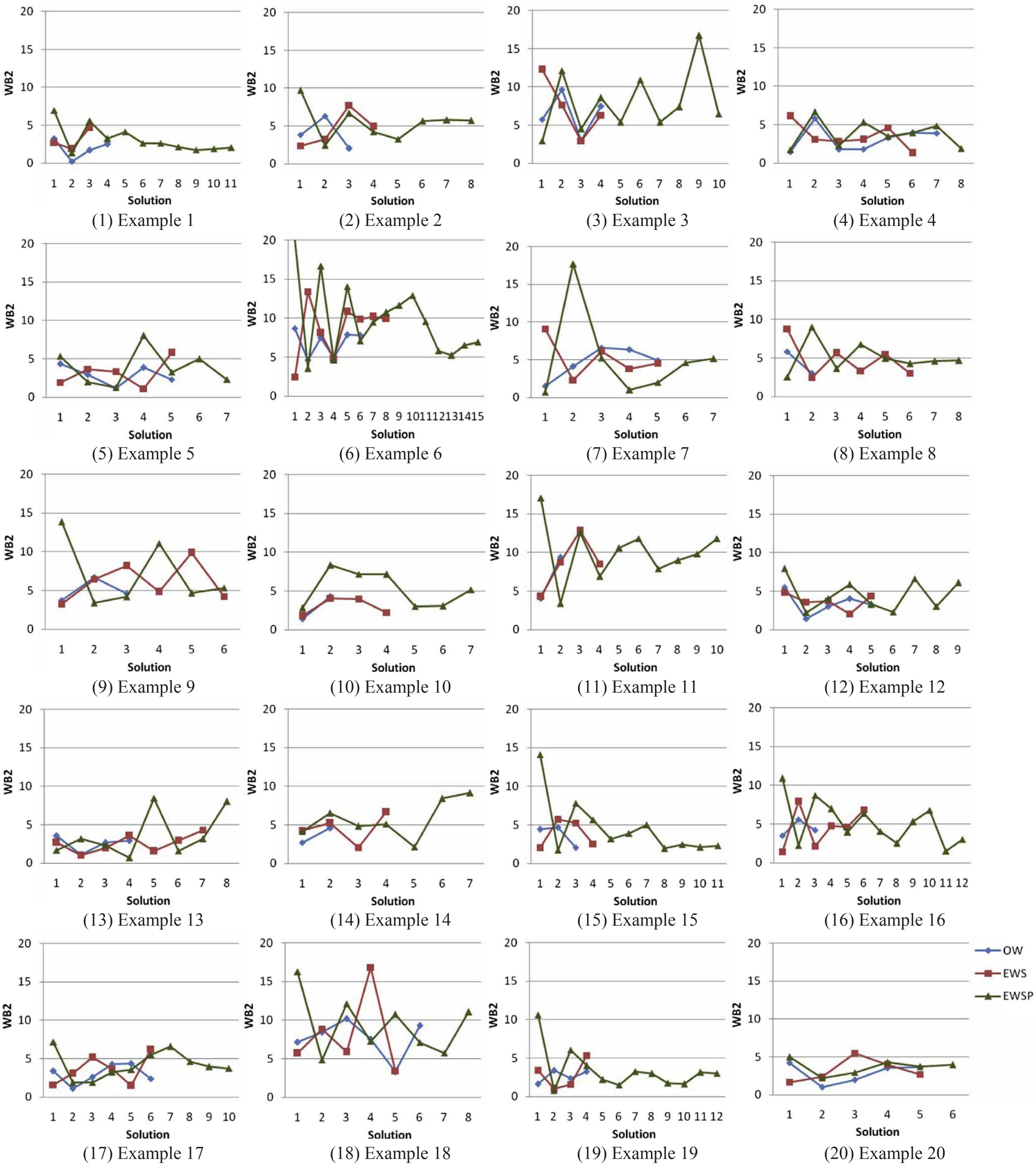

Fig. 8. The comparison of the second objective function's value of non-dominated solutions under different conditions.

NSGA-II algorithm which is coded using MATLAB 2012b software on a personal computer with Intel Core 2 Quad CPU 2.66 GHz and 2 GB RAM. On the basis of the computational results of the proposed NSGA-II algorithm, the impact of workers' differences in competency on the workload balance is analyzed.

5.1. An illustrative example

This example concerns on a *seru* production system in which 5 multi-skilled workers have to be assigned to 3 *serus* for responding to the demand for 3 product types. There are totally 4 tasks and each task corresponds to one skill. The lower and upper bounds of the proficiency level are 0.9 and 1.1, respectively. Table 4 demonstrates the competency of workers, where the number denotes the worker's proficiency level with respect to the corresponding skill. For example, worker 1 is competent at skill 1, skill 3, and skill 4, and the proficiency levels are 1.04, 0.99, and 1.07, respectively. That means worker 1 can perform task 3 faster than the standard processing time while he/she needs more

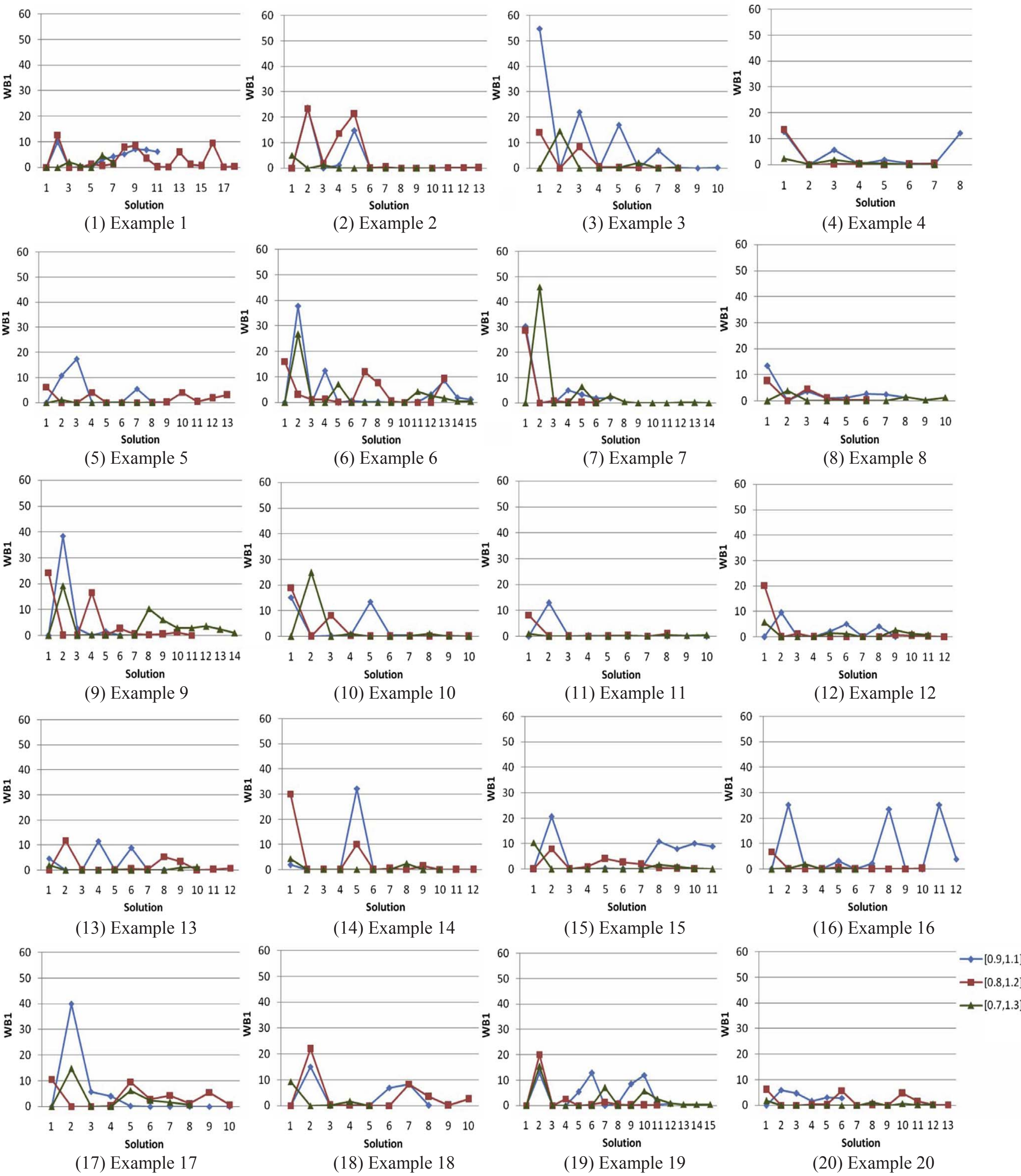

Fig. 9. The comparison of the first objective function's value of non-dominated solutions of EWSP with different ranges of proficiency level.

time when performing task 1 and task 4. The data sets related to standard processing times and demands are shown in Tables 5 and 6, respectively. The upper bound of the number of workers assigned to each *seru* and tasks assigned to each worker are 3. Moreover, the available time of each worker is assumed to be 2400. Fig. 5 shows the links between workers and tasks. We can see that there are numerous possible combinations of batches and workers.

The proposed multi-objective model is modified into a single-objective one by the weighted sum method. In this case, the weight of the objective functions is assumed to be equal. After the proposed model was linearized for the illustrative example, it contains 509 variables and 1490 linear constraints, and the software used about 3742 cuts during 1,037,429 iterations to find the global optimal solution in 184s. Table 7 shows the values of objective functions and its components. The solution of this example including batch-*seru*, worker-*seru*, and worker-task assignment results are illustrated in Fig. 6. As shown in

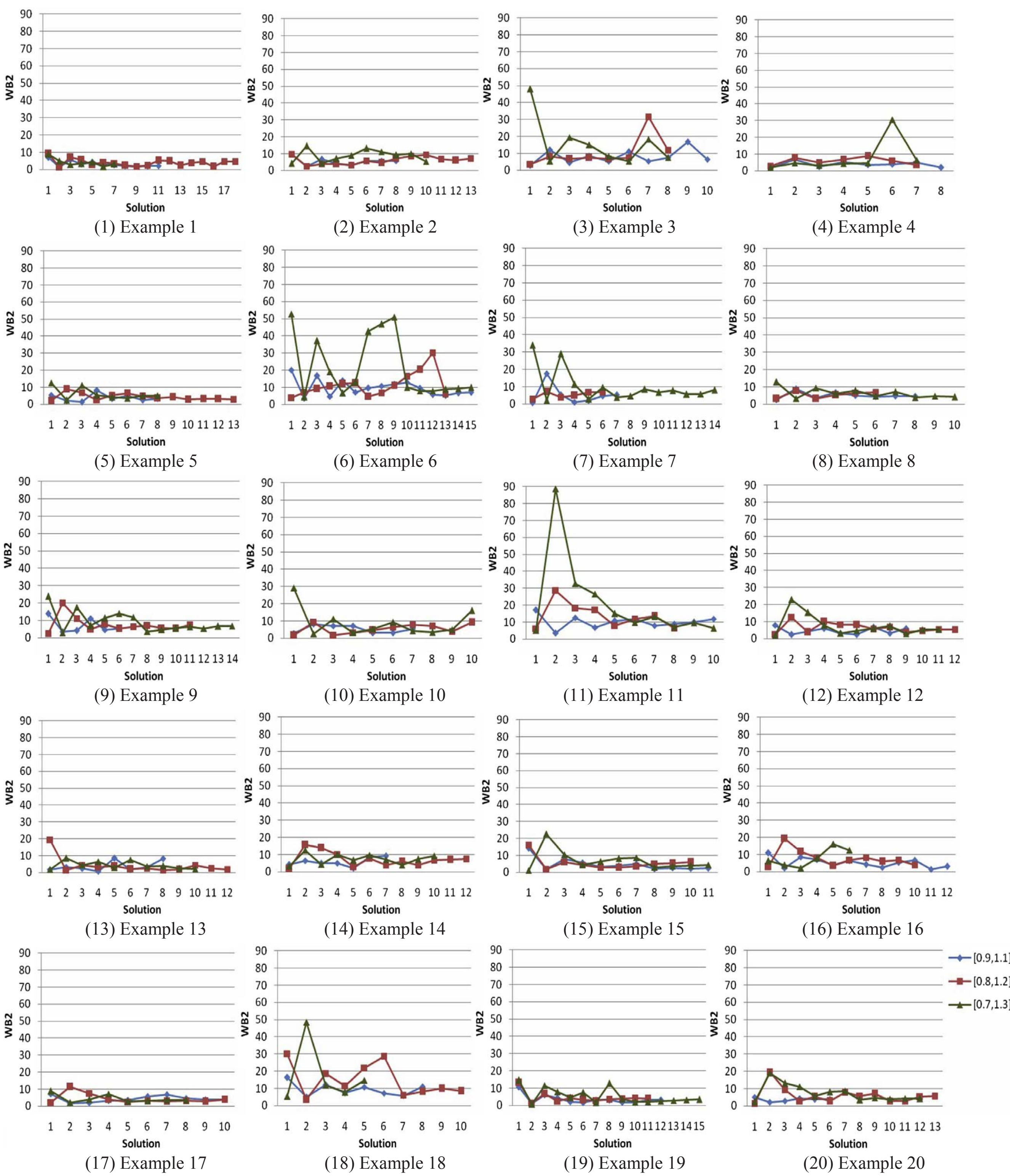

Fig. 10. The comparison of the second objective function's value of non-dominated solutions of EWSP with different ranges of proficiency level.

Fig. 6, batches 1 and 5 and workers 1 and 3 are assigned to *seru* 1, batch 2 and worker 2 are assigned to *seru* 2, and batches 3 and 4 and workers 4 and 5 are assigned to *seru* 3. In *seru* 1 and *seru* 3, tasks of each batch are assigned to different workers, while in *seru* 2, batch 2 is processed by worker 2 from start to finish. From Fig. 6, we can see that there is no inter-*seru* product movement, and no inter-*seru* worker transfer, and no idle worker in the result.

5.2. Experimental results and analysis

In this section, the proposed algorithm is tested by using randomly generated numerical examples and the results are compared to investigate the impact of differences in workers' competency on the workload balance. 20 specific examples are randomly generated based on the pattern given in Table 8. For each example, three conditions with different skill sets and/or proficiency levels are considered. In condition

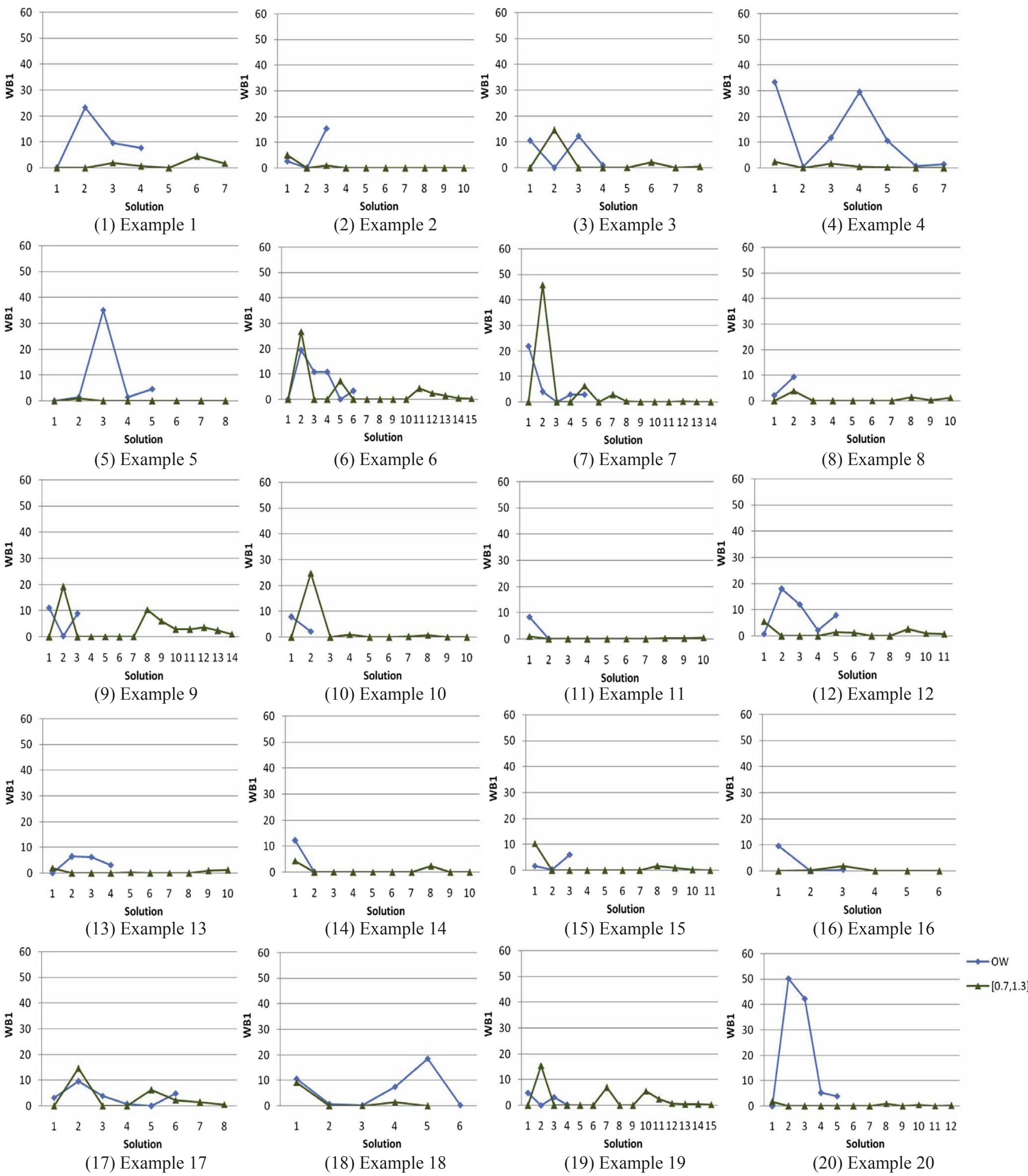

Fig. 11. The comparison of the first objective function's value of non-dominated solutions between OW and EWSP with proficiency level ranging from 0.7 to 1.3.

I, workers are homogenous. They have the same skill set and can perform skills at the same proficiency level. In condition II, workers are heterogenous and have different skill sets, but proficiency levels of them remain equal. In condition III, workers differ from one another in both skill sets and proficiency levels. The three conditions will hereinafter be represented by OW, EWS, and EWSP, respectively. Table 9 shows the value of parameters used in the genetic algorithm. Each example was performed 20 runs to ensure obtaining a good result.

Figs. 7 and 8 show the comparison of non-dominated solutions of numerical examples under three conditions. From Fig. 7, we can see that both homogenous workers and workers with different skill sets and proficiency levels perform better from the point of the inter-*seru* workload balance than workers with different skill sets. When the skill set of workers are different, the number of skills that each worker is able to perform is less than that of homogenous workers. Therefore, it is hard to find an assignment plan which is excellent from the point of the inter-*seru* workload balance under the influence of a big drop in the number of possible combinations of batches and workers. While for

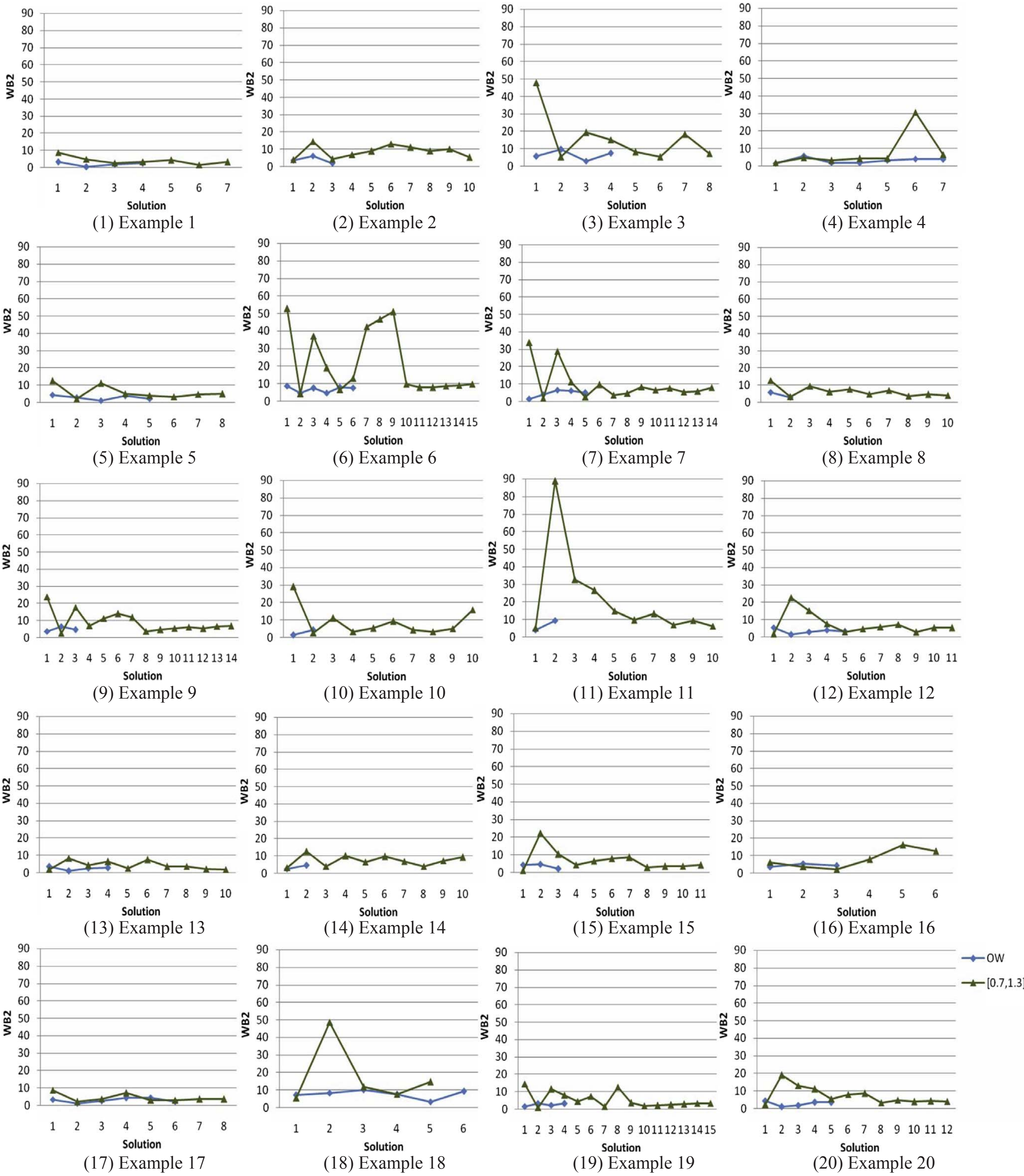

Fig. 12. The comparison of the second objective function's value of non-dominated solutions between OW and EWSP with proficiency level ranging from 0.7 to 1.3.

workers with different skill sets and proficiency levels, the processing time of a task varies with the proficiency level of worker assigned to the task. In this case, the workload of *serus* can be easily balanced by adjusting batches and workers assigned to *serus* and worker assigned to each task. As shown in Fig. 8, the workload balance among workers gets worse as the diversity of workers' competency increases. As stated by Jordan and Graves (1995), homogenous workers can form a closed complete skill chain which facilitates the transfer of workload from a worker to any other one. As a result, homogenous workers perform better from the point of the inter-worker workload balance.

Although both homogenous workers and workers with different skill sets and proficiency levels perform well from the point of the inter-*seru* workload balance, it is not clear which one is better. In order to clarify the impact of proficiency level on the inter-*seru* workload balance, we test the 20 sets of examples under condition EWSP with the proficiency level in the range of [0.8, 1.2] and [0.7, 1.3], respectively. The comparison of non-dominated solutions of EWSP with different ranges of proficiency level can be found in Fig. 9 and 10. As we can see from those two figures, the level of inter-*seru* workload balance gets higher as the range of proficiency level becomes wider, while the inter-worker workload balance is exactly the opposite.

Figs. 11 and 12 show the comparison of non-dominated solutions between OW and EWSP with proficiency level ranging from 0.7 to 1.3. There is no doubt that homogenous workers perform well from the point of the inter-worker workload balance. However, it is noteworthy that workers under condition EWSP with proficiency level ranging from 0.7 to 1.3 even perform better in inter-*seru* workload balance than homogenous workers with total flexibility.

In summary, homogenous workers can bring the high level of inter-worker workload balance whereas heterogenous workers with diversified competency perform well in balancing inter-*seru* workload. Under the condition that workers assigned to a *seru* are treated as a team, diversifying the competency of workers is helpful in enhancing interpersonal justice and equity. For cases with special requirements on the inter-worker workload balance, a small number of homogenous workers are recommended though numerous researches stated that workers should not necessarily be completely cross-trained in all skills.

6. Conclusions

In this paper, a multi-skilled worker assignment problem considering differences in workers' skill sets and proficiency levels in a *seru* production environment is investigated. A bi-objective mathematical model is formulated, in which the objectives are to improve the inter-*seru* workload balance and the inter-worker workload balance. The proposed nonlinear model is linearized by employing a number of new variables with auxiliary constraints and then validated by a numerical example. Due to the complexity of the problem, an algorithm based on NSGA-II is presented. Finally, the performance of the proposed algorithm is tested by numerical examples and the impact of differences in workers' competency on the workload balance is analyzed based on the computational results.

Some guidelines for future researches can be outlined as follows. First, it is necessary to explore a more accurate way to calculate the flow time of a batch by incorporating WIP into the problem. In the present paper, we assume that a workpiece can start to be processed only when the previous one is finished in view of the complicated task division plans. Although the computational difficulty is decreased, there exists a large amount of idle time for workers. Second, the effects of learning and forgetting should be investigated for a multi-period multi-skilled worker assignment problem with varying skill sets and proficiency levels.

Acknowledgements

The authors would like to thank two anonymous reviewers for their helpful comments and constructive suggestions. This research is supported by the National Natural Science Foundation of China (Grant No. 71671139, 71371153, 71171161) and National Natural Science Foundation of Shaanxi Province, China (Grant No. 2016JM7005).

References

Aalaei, A., & Shavazipour, B. (2013). The Tchebycheff norm for ranking DMUs in cellular manufacturing systems with assignment worker. *International Journal of Applied Operational Research, 3*(3), 41–57.

Aalaei, A., Paydar, M. M., & Saidi-Mehrabad, M. (2014). Data envelopment analysis in cellular manufacturing systems considering worker assignment. *International Journal of Services and Operations Management, 18*(3), 258–280.

Ammar, A., Pierreval, H., & Elkosantini, S. (2013). Workers assignment problems in manufacturing systems: A literature analysis. In: *5th International conference on industrial engineering and systems management. 28–30 October. Rabat, Morocco* (p. 7).

Aryanezhad, M. B., Deljoo, V., & Mirzapour Al-e-hashem, S. M. J. (2009). Dynamic cell formation and the worker assignment problem: A new model. *International Journal of Advanced Manufacturing Technology, 41*, 329–342.

Attia, E., Edi, K. H., & Duquenne, P. (2012). Flexible resources allocation techniques: Characteristics and modelling. *International Journal of Operational Research, 14*, 221–254.

Attia, E., Duquenne, P., & Le-Lann, J. (2014). Considering skills evolutions in multi-skilled workforce allocation with flexible working hours. *International Journal of Production Research, 52*(15), 4548–4573.

Azadeh, A., Nazari-Shirkouhi, S., Hatami-Shirkouhi, L., & Ansarinejad, A. (2011a). A unique fuzzy multi-criteria decision making: Computer simulation approach for productive operators' assignment in cellular manufacturing systems with uncertainty and vagueness. *The International Journal of Advanced Manufacturing Technology, 56*(1–4), 329–343.

Azadeh, A., Nokhandan, B. P., Asadzadeh, S. M., & Fathi, E. (2011b). Optimal allocation of operators in a cellular manufacturing system by an integrated computer simulation-genetic algorithm approach. *International Journal of Operational Research, 10*(3), 333–360.

Azadeh, A., Sheikhalishahi, M., & Koushan, M. (2013). An integrated fuzzy DEA-Fuzzy simulation approach for optimization of operator allocation with learning effects in multi products CMS. *Applied Mathematical Modelling, 37*, 9922–9933.

Azizi, N., Zolfaghari, S., & Liang, M. (2010). Modeling job rotation in manufacturing systems: The study of employee's boredom and skill variations. *International Journal of Production Economics, 123*, 69–85.

Azizi, N., & Liang, M. (2013). An integrated approach to worker assignment, workforce flexibility acquisition, and task rotation. *Journal of the Operational Research Society, 64*(2), 260–275.

Bagheri, M., & Bashiri, M. (2014). A new mathematical model towards the integration of cell formation with operator assignment and inter-cell layout problems in a dynamic environment. *Applied Mathematical Modelling, 38*, 1237–1254.

Bhaskar, K., & Srinivasan, G. (1997). Static and dynamic operator allocation problems in cellular manufacturing systems. *International Journal of Production Research, 35*(12), 3467–3482.

Bokhorst, J. A. C., Slomp, J., & Gaalman, G. J. C. (2004). On the who-rule in dual resource constrained (DRC) manufacturing systems. *International Journal of Production Research, 42*(23), 5049–5074.

Bootaki, B., Mahdavi, I., & Paydar, M. M. (2014). A hybrid GA-AUGMECON method to solve a cubic cell formation problem considering different worker skills. *Computers & Industrial Engineering, 75*, 31–40.

Cesaní, V. I., & Steudel, H. J. (2005). A study of labor assignment flexibility in cellular manufacturing systems. *Computers & Industrial Engineering, 48*, 571–591.

Corominas, A., Olivella, J., & Pastor, R. (2010). A model for the assignment of a set of tasks when work performance depends on experience of all tasks involved. *International Journal of Production Economics, 126*, 335–340.

Costa, A., Cappadonna, F. A., & Fichera, S. (2014). Joint optimization of a flow-shop group scheduling with sequence dependent set-up times and skilled workforce assignment. *International Journal of Production Research, 52*(9), 2696–2728.

De Bruecker, P., Van den Bergh, J., Beliën, J., & Demeulemeester, E. (2015). Workforce planning incorporating skills: State of the art. *European Journal of Operational Research, 243*(1), 1–16.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Egilmez, G., Erenay, B., & Süer, G. A. (2014). Stochastic skill-based manpower allocation in a cellular manufacturing system. *Journal of Manufacturing Systems, 33*, 578–588.

Ferjani, A., Ammar, A., Pierreval, H., & Elkosantini, S. (2017). A simulation-optimization based heuristic for the online assignment of multi-skilled workers subjected to fatigue in manufacturing systems. *Computers & Industrial Engineering, 112*, 663–674.

Firat, M., Hurkens, C. A. J., & Laugier, A. (2014). Stable multi-skill workforce assignments. *Annals of Operations Research, 213*(1), 95–114.

Gel, E. S., Hopp, W. J., & Van Oyen, M. P. (2007). Hierarchical cross-training in work-in-process-constrained systems. *IIE Transactions, 39*(2), 125–143.

Gen, M., & Cheng, R. (1999). *Genetic Algorithms and Engineering Optimization.* Wiley-Interscience.

Ghotboddini, M. M., Rabbani, M., & Rahimian, H. (2011). A comprehensive dynamic cell formation design: Benders' decomposition approach. *Expert Systems with Applications, 38*, 2478–2488.

Givi, Z. S., Jaber, M. Y., & Neumann, W. P. (2015). Production planning in DRC systems considering worker performance. *Computers & Industrial Engineering, 87*, 317–327.

Gomar, J. E., Haas, C. T., & Morton, D. P. (2002). Assignment and allocation optimization of partially multiskilled workforce. *Journal of Construction Engineering and Management, 128*(2), 103–109.

Hewitt, M., Chacosky, A., Grasman, S. E., & Thomas, B. W. (2015). Integer programming techniques for solving non-linear workforce planning models with learning. *European Journal of Operational Research, 242*, 942–950.

Hopp, W. J., & Van Oyen, M. P. (2004). Agile workforce evaluation: A framework for cross-training and coordination. *IIE Transactions, 36*, 919–940.

Inman, R. R., Jordan, W. C., & Blumenfeld, D. E. (2004). Chained cross-training of assembly line workers. *International Journal of Production Research, 10*(42), 1899–1910.

Isa, K., & Tsuru, T. (2002). Cell production and workplace innovation in Japan: Toward a new model for Japanese manufacturing. *Industrial Relations: A Journal of Economy and Society, 41*(4), 548–578.

Jaber, M. Y., & Neumann, W. P. (2010). Modelling worker fatigue and recovery in dual-resource constrained systems. *Computers & Industrial Engineering, 59*, 75–84.

Jordan, W. C., & Graves, S. C. (1995). Principles on the benefits of manufacturing process flexibility. *Management Science, 41*(4), 577–594.

Jordan, W. C., Inman, R. R., & Blumenfeld, D. E. (2004). Chained cross-training of workers for robust performance. *IIE Transactions, 36*(10), 953–967.

Kaku, I., Murase, Y., & Yin, Y. (2008). A study on human-task-related performances in converting conveyor assembly line to cellular manufacturing. *European Journal of Operational Research, 2*(1), 17–34.

Kim, S., & Nembhard, D. A. (2013). Rule mining for scheduling cross training with a heterogeneous workforce. *International Journal of Production Research, 51*(8), 2281–2300.

Liu, C. G., Lian, J., Yin, Y., & Li, W. J. (2010). *Seru seisan* – An innovation of the production management mode in Japan. *Asian Journal of Technology Innovation, 18*(2), 89–113.

Liu, C. G., Yang, N., Li, W. J., Lian, J., Evans, S., & Yin, Y. (2013). Training and assignemtn of multi-skilled workers for implementing *seru* production systems. *International Journal of Advanced Manufacturing Technology, 69*, 937–959.

Liu, C. G., Stecke, K. E., Lian, J., & Yin, Y. (2014). An implementation framework for seru production. *International Transactions in Operational Research, 21*(1), 1–19.

Liu, C., Wang, J., & Leung, J. Y. T. (2016a). Worker assignment and production planning with learning and forgetting in manufacturing cells by hybrid bacteria foraging algorithm. *Computers & Industrial Engineering, 96*, 162–179.

Liu, C., Wang, J., Leung, J. Y. T., & Li, K. (2016b). Solving cell formation and task scheduling in cellular manufacturing system by discrete bacteria foraging algorithm. *International Journal of Production Research, 54*(3), 923–944.

Mahdavi, I., Aalaei, A., Paydar, M. M., & Solimanpur, M. (2010). Designing a mathematical model for dynamic cellular manufacturing systems considering production planning and worker assignment. *Computers and Mathematics with Applications, 60*, 1014–1025.

Mahdavi, I., Aalaei, A., Paydar, M. M., & Solimanpur, M. (2011). Multi-objective cell formation and production planning in dynamic virtual cellular manufacturing systems. *International Journal of Production Research, 49*(21), 6517–6537.

Mahdavi, I., Aalaei, A., Paydar, M. M., & Solimanpur, M. (2012). A new mathematical model for integrating all incidence matrices in multi-dimensional cellular manufacturing system. *Journal of Manufacturing Systems, 31*, 214–223.

McDonald, T., Ellis, K. P., Van Aken, E. M., & Patrick Koelling, C. (2009). Development and application of a worker assignment model to evaluate a lean manufacturing cell. *International Journal of Production Research, 47*(9), 2427–2447.

Mehdizadeh, E., & Rahimi, V. (2016). An integrated mathematical model for solving dynamic cell formation problem considering operator assignment and inter/intra cell layouts. *Applied Soft Computing, 42*, 325–341.

Mehdizadeh, E., Daei Niaki, S. V., & Rahimi, V. (2016). A vibration damping optimization algorithm for solving a new multi-objective dynamic cell formation problem with workers training. *Computers & Industrial Engineering, 101*, 35–52.

Mital, A., Pennathur, A., Huston, R. L., Thompson, D., Pittman, M., Markle, G., Kaber, D. B., Crumpton, L., Bishu, R. R., Rajurkar, K. P., Rajan, V., Fernandez, J. E., McMulkin, M., Deivanayagam, S., Ray, P. S., Sule, D., et al. (1999). The need for worker training in advanced manufacturing technology(AMT) environments: A white paper. *International Journal of Industrial Ergonomics, 24*(2), 173–184.

Miyake, D. I. (2006). The shift from belt conveyor line to work-cell based assembly systems to cope with increasing demand variation in Japanese industries. *International Journal of Automotive Technology and Management, 6*(4), 419–439.

Nembhard, D. A., & Bentefouet, F. (2014). Selection policies for a multifunctional workforce. *International Journal of Production Research, 52*(16), 4785–4802.

Niakan, F., Baboli, A., Moyaux, T., & Botta-Genoulaz, V. (2016). A bi-objective model in sustainable dynamic cell formation problem with skill-based worker assignment. *Journal of Manufacturing Systems, 38*, 46–62.

Nikoofarid, E., & Aalaei, A. (2012). Production planning and worker assignment in a dynamic virtual cellular manufacturing system. *International Journal of Management Science and Engineering Management, 7*(2), 89–95.

Norman, B. A., Tharmmaphornphilas, W., Needy, K. L., Bidanda, B., & Warner, R. C. (2002). Worker assignment in cellular manufacturing considering technical and human skills. *International Journal of Production Research, 40*(6), 1479–1492.

Olivella, J., Corominas, A., & Pastor, R. (2013). Task assignment considering cross-training goals and due dates. *International Journal of Production Research, 51*(3), 952–962.

Othman, M., Bhuiyan, N., & Gouw, G. J. (2012). Integrating workers' differences into workforce planning. *Computers & Industrial Engineering, 63*, 1096–1106.

Rafiei, H., & Ghodsi, R. (2013). A bi-objective mathematical model toward dynamic cell formation considering labor utilization. *Applied Mathematical Modelling, 37*, 2308–2316.

Sakazume, Y. (2005). Is Japanese cell manufacturing a new system? A comparative study between Japanese cell manufacturing and cellular manufacturing. *Journal of Japan Industrial Management Association, 55*(6), 341–349.

Sakhaii, M., Tavakkoli-Moghaddam, R., Bagheri, M., & Vatani, B. (2016). A robust optimization approach for an integrated dynamic cellular manufacturing system and production planning with unreliable machines. *Applied Mathematical Modelling, 40*, 169–191.

Sirovetnukul, R., & Chutima, P. (2010). The impact of walking time on U-shaped assembly line worker allocation problems. *Engineering Journal, 14*(2), 53–78.

Slomp, J., & Molleman, E. (2002). Cross-training policies and team performance. *International Journal of Production Research, 40*(5), 1193–1219.

Stecke, K. E., Yin, Y., Kaku, I., & Murase, Y. (2012). *Seru*: The organizational extension of JIT for a super-talent factory. *International Journal of Strategic Decision Sciences, 3*(1), 106–119.

Süer, G. A., & Dagli, C. (2005). Intra-cell manpower transfers and cell loading in labor-intensive manufacturing cells. *Computers & Industrial Engineering, 48*, 643–655.

Süer, G. A., & Tummaluri, R. R. (2008). Multi-period operator assignment considering skills, learning and forgetting in labour-intensive cells. *International Journal of Production Research, 46*(2), 469–493.

Süer, G. A., Cosner, J., & Patten, A. (2009). Models for cell loading and product sequencing in labor-intensive cells. *Computers & Industrial Engineering, 56*, 97–105.

Süer, G. A., Kamat, K., Mese, E., & Huang, J. (2013). Minimizing total tardiness subject to manpower restriction in labor-intensive manufacturing cells. *Mathematical and Computer Modelling, 57*, 741–753.

Valeva, S., Hewitt, M., Thomas, B. W., & Brown, K. G. (2017). Balancing flexibility and inventory in workforce planning with learning. *International Journal of Production Economics, 183*, 194–207.

Yin, Y., Kaku, I., & Stecke, K. E. (2008). The evolution of *seru* production systems throughout Canon. *Operations Management Education Review, 2*, 27–40.

Yin, Y., Stecke, K. E., Swink, M., & Kaku, I. (2017). Lessons from *seru* production on manufacturing competitively in a high cost environment. *Journal of Operations Management, 49–51*, 67–76.

Zhang, H., & Haapala, K. R. (2015). Integrating sustainable manufacturing assessment into decision making for a production work cell. *Journal of Cleaner Production, 105*, 52–63.

Zohrevand, A. M., Rafiei, H., & Zohrevand, A. H. (2016). Multi-objective dynamic cell formation problem: A stochastic programming approach. *Computers & Industrial Engineering, 98*, 323–332.