

---

This is an electronic reprint of the original article.  
This reprint may differ from the original in pagination and typographic detail.

Peltokorpi, Jaakko; Niemi, Esko; Tokola, Henri

## Comparing Worker Coordination Policies in Parallel Station Systems with Different Worker Skill Level Distributions

*Published in:*

25th International Conference on Flexible Automation & Intelligent Manufacturing (FAIM 2015), 23-26 June 2015, Wolverhampton, UK

Published: 01/01/2015

*Document Version*

Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

*Please cite the original version:*

Peltokorpi, J., Niemi, E., & Tokola, H. (2015). Comparing Worker Coordination Policies in Parallel Station Systems with Different Worker Skill Level Distributions. In *25th International Conference on Flexible Automation & Intelligent Manufacturing (FAIM 2015), 23-26 June 2015, Wolverhampton, UK*

---

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

# Comparing worker coordination policies in parallel station systems with different worker skill level distributions

Jaakko Peltokorpi<sup>\*</sup>, Esko Niemi, and Henri Tokola

Department of Engineering Design and Production  
Aalto University School of Engineering  
Espoo, 02150, Finland

## ABSTRACT

*This paper investigates the performance of a parallel assembly system with workers that have different levels of skills. The paper compares worker coordination policies that involve different amounts of helping (that aim to avoid idleness of the workforce) such as a floater helping, pair partners helping each other and everyone helping everyone. In addition, the principles of who is to be helped and to whom to assign a job are studied as random and optimal cases. The system is studied by using a continuous-time Markov model. The results show that the skill level distribution of the workers has the greatest impact with the floater policy. Helping the least skilled ones and assigning jobs to the most skilled ones prior to others improves the performance of the system in relation to random priorities. An optimal assignment of jobs is especially beneficial at low system utilisation rates and with the policies that most restrict the amount of helping. In general, the impacts of skill level distribution and whether optimal or random priorities are used are less significant than the amount of helping.*

## 1. INTRODUCTION

This paper studies parallel manual assembly which receives jobs in independent, variable arrivals and with highly varying work contents. In such manual assembly, the varying workload can be handled and the performance of the system improved by assigning jobs and coordinating workers in a reasonable way. This is done most effectively by using clear policies that also consider the unequal skills of the workers.

According to [1], a *worker coordination policy* “allocates workers to tasks (or tasks to workers) over time”. Such policies have previously been examined in numerous studies and typically from the perspective of cross-training (such as in [2], [3] and [4]). Some studies (e.g. [4], [5] and [6]) assume that the uncertainties related to job arrivals and processing follow stochastic, Markov processes. The vast majority of studies deal with serial line production. Parallel systems that consider worker and task allocation priorities based on the skill levels of workers have been studied less frequently. Generic models for such types of systems were provided by Gurumurthi and Benjaafar [7], who recommend further studies on “the impact of demand and service variability on different system configurations and different control policies”.

On the basis of the research need and differently from the preliminary study of the present authors in [8], this paper aims to compare the performances of different worker coordination policies and priorities with different skill level distributions of workers. The present study assumes a parallel system of four stations and workers. Each station has, in principle, a dedicated worker but in a case in which workers would otherwise be idle they are allowed to help the others. The amount of helping allowed depends on the worker coordination policy, starting from *no helping* and ending with *complete helping*, in which everyone can help everyone. In addition, the policies involve the principles of who is to be helped and to whom to assign a job in different situations. These are examined as random and optimal cases. All the above is studied by using continuous-time Markov models. The performances of different experiments that comprise given workforce skill level distributions, worker coordination policies and system utilisation rates are calculated as the average job cycle time. On the basis of the results, the impacts of these variables on system performance are compared and analysed.

The remainder of this paper is organised as follows. In Section 2, a brief review of the relevant literature is provided. In Section 3, the system and worker coordination policies under study are described, the principles for

---

<sup>\*</sup> Corresponding author: Tel.: (+358) 50 566 2382; E-mail: jaakko.peltokorpi@aalto.fi

studying the system as Markov models are introduced and the results from the different experiments are presented. In Section 4, the impacts of different variables in the experiments are discussed. Finally, in Section 5, the study is concluded and aspects for further research are given.

## 2. LITERATURE REVIEW

Many previous studies examined worker coordination policies for allocating workers to tasks or tasks to workers. One of the earliest related policies is chaining, which was originally introduced in [9]. In chaining, flexibility in terms of connecting workers to tasks can be increased in small increments. In a study on flexibility between parallel sections of an automotive assembly line [3] a closed, complete chain configuration was found to result in the best performance. Further, the study found that the policy that shared the flexibility between two adjacent sections performed well. Another study [2] also found the benefits from chaining that shifts the capacity indirectly to the stations in a serial system.

Gurumurthi and Benjaafar [7] studied flexible queuing systems with multiple categories of customers and multiple parallel servers. The study found that with identical arrivals between different types of customers and identical service rates (servers), an initial fully chained configuration achieves most of the benefits of total flexibility. On the contrary, with non-identical arrival rates and unequal servers, flexibility should be associated with either the fastest servers or with customers of the largest demand. The study also found that the impacts of control policies that involve priority principles for server and customer selection were less significant than that of flexibility.

According to [10], a work group is at its most effective when the best workers are kept fully utilised while the worst workers are only used if a job would otherwise be idle. The study in [11] highlighted the importance of allocating flexible, cross-trained workers in such a way that their special expertise is taken into consideration. Andradottir et al. [6] studied a serial queuing system and concluded that, when the processing times depend on both the worker and the task, workers should be assigned to the tasks in such a way that the total average throughput of the system is maximised. According to [5], when workers are not fully skilled in a serial assembly system, a method in which flexibility (in terms of task assignment and worker movement) can be shared between two adjacent stations performs well when compared to traditional and bucket brigade methods.

Unlike any of the relevant studies above, the present study compares the performances of practical worker coordination policies in parallel systems with different skill level distributions in the workforce. This study also contributes to the call of Gurumurthi and Benjaafar [7], who presented a general framework for further studying various configurations related to control policies and different arrival and processing rates.

## 3. EXPERIMENTS

This section studies different worker coordination policies in a parallel stations system by continuing the previous research [8] of the present authors. First, the parallel assembly system is described. Then the worker coordination policies under study are presented and the principles for studying them as Markov processes are described. Finally, the results from different experiments are presented.

### 3.1. SYSTEM DESCRIPTION

The system consists of four stations and their dedicated workers (Figure 1). Each worker has their own skill level and a corresponding average processing rate. The average processing rates studied for four different skill level distributions of the workers are presented in Table 1.

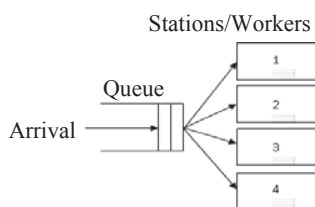


Figure 1: Parallel assembly system

Table 1: Average processing rates with different skill level distributions of workers

	<b>Identical skills (Identical)</b>	<b>Uniformly distributed skills (Uni)</b>	<b>Two skilled and two unskilled (2+2)</b>	<b>One skilled and three less skilled (1+3)</b>
$\mu_1$	1	1.2	1.2	1.2
$\mu_2$	1	1.067	1.2	0.933
$\mu_3$	1	0.933	0.8	0.933
$\mu_4$	1	0.8	0.8	0.933

For the present assembly system, the following principles hold:

- (1) the jobs have Poisson-distributed arrivals and they are centrally assigned, one at a time, to a station from a single queue in the event that there is a need for jobs. The worker to whom a job is assigned also completes that job;
- (2) when a worker completes a job and the queue is empty, that worker can help others according to the worker coordination policy that is used. The principles used in helping are presented in Table 2. When helping others, each worker can only help a worker who is not yet being helped. When helping is not possible, a worker will be idle;
- (3) if a new job arrives while all the stations are busy, the job is added to the queue. Otherwise it is primarily allocated to an idle worker, optimally or randomly, as explained later. When several workers are idle, helping, determined by the worker coordination policy being used and whether an optimal or random case is applied, takes place as soon as a new job arrives.

The Poisson-distributed arrivals of jobs describe the situation in which work orders are received from multiple independent customers. The jobs are promised a relatively quick delivery by a FIFO (first-in, first-out) principle, depending on the current workload in the system. The assembly is implemented by a make-to-order principle and the work contents of independent jobs are highly variable. This customisation is modelled by varying processing rates that are assumed to follow a Poisson distribution (many small jobs and few large jobs) with a mean processing rate of  $\mu = 1$ . The skill level of the worker finally determines the actual average processing rate of that worker.

The relevance of considering different skill level distributions is confirmed e.g. in [12], which states that especially within a parallel stations design, worker differences, in terms of the time taken to perform the required tasks, are significant. The study revealed an experience from the car industry, where, at one assembly station, in a group of six workers, the most skilled worker performed the task twice as fast on average as the least skilled one. Similar differences were observed earlier in [13]. In the present study, among the four parallel stations, the assumption is that the most skilled worker processes at a rate that is at most one and a half times as fast in relation to the least skilled one. In the experiments in the present study, four different skill level distributions with a mean worker processing rate of one are examined. Another option would be to study skill level distributions with a mean processing time of one. The skill levels presented in Table 1 are assumed to be static.

### 3.2. WORKER COORDINATION POLICIES

A worker coordination policy determines the ways in which the workers help each other. The principles used in helping within each policy are presented in Table 2. These principles first determine who is allowed to help others. In addition, the priority of who is to be helped is examined in two cases: optimal (OPT), which is based on the skill levels of the workers and, random (RAND), which means that the worker who is to be helped is selected arbitrarily.

Table 2: Principles of worker coordination policies

Worker coordination policy	Who is allowed to help others	Who is to be helped in random (RAND) and optimal (OPT) cases
<i>no helping</i> (NH)	none	none (RAND and OPT)
<i>floater</i> (FL)	only the floater	a randomly selected worker (RAND) the worker with the lowest skill level (OPT)
<i>pairs</i> (P)	everyone	a fixed pair partner that is selected randomly (RAND) a fixed pair partner that is selected optimally (OPT) by minimising the differences between the total skills of the pairs
<i>complete helping</i> (CH)	everyone	a randomly selected worker (RAND) the worker with the lowest skill level (OPT); in the case of pair completes a job the most skilled moves to help the lowest skilled

Optimal helping that enables the average job cycle time to be minimised means that the worker with the lowest level of skill should be helped first (as in *floater* and *complete helping*). This is due to the fact that it takes the longest time to complete a job for the worker with the lowest processing rate. The same principle also holds for *pairs*, for which the selection of pair partners (who is to be helped) is optimal when the most skilled one forms a pair

with the least skilled one, in which case the differences in skills between the pairs are minimised. This principle in pairing is also applied in *complete helping* (OPT) when the workers in one pair complete their job and move to help those with work left.

The optimal and random cases within each worker coordination policy also cover the priorities in assigning jobs. In both cases a job is primarily assigned to an idle worker if at least one such exists. As expected, the random priority assigns a job to an arbitrarily selected idle worker. The optimal priority, instead, assigns a job to the worker with the greatest skill level. This is in line with e.g. [10] as reviewed above. Within both the random and optimal cases, an assignment for a busy worker refers to the situations in which a worker (the one who helps) from a busy pair takes care of a job that has arrived. In practice, this forces the worker to whom the job was originally assigned to carry out the completion of that job, which is clearly reasonable. However, here a worker from a busy pair to whom a job is assigned will be selected arbitrarily, as the pair partner to whom the job was originally assigned is not known. This is a consequence of the memoryless Markov property, according to which the previous occurrences in processes do not matter.

When a new job arrives, in addition to the priority in assignment of that job, helping (the principles presented in Table 2) takes place, whenever it is possible. Within the *floaters* policy, in fact, the floater (who is always the most skilled one) helps others up to the moment when the floater himself is the last one able to take care of a job that has arrived. Within *pairs*, an idle worker immediately starts to help his fixed pair partner and within *complete helping*, the idle worker who starts helping is determined by the optimal and random priorities.

### 3.3. MARKOV MODELS AND EXPERIMENTS

The parallel assembly systems with different workforce skill level distributions and worker coordination policies presented above are studied by using continuous-time Markov models. The systems can be modelled similarly to a basic M/M/s multi-server queuing model as presented in [14]. In such a system jobs arrive with the Poisson-distributed arrival rate  $\lambda$ . There are  $s$  numbers of servers, or workers, who process the jobs according to the FIFO principle. The mean processing rate for the workers is  $\mu$ . In the present study, at each point in time, the number of jobs in the system, together with the workers who are working as singles and as pairs, determine the state of the process. All the states in the system, from zero to infinite, together construct a continuous-time Markov chain. A transition can occur between two adjacent states, depending on the state and whether a job is completed or has arrived, as well as whether the optimal or random priority within a given worker coordination policy is used. As representative examples of that, Figures 2-5 present how the Markov state diagrams are constructed with optimal priorities for different worker coordination policies with skills uniformly distributed among the workers. In various states, the number of jobs in the system  $n$  and the workers (numbered from 1...4) who are working as singles (S) and as pairs (P) are shown. In addition, the possible numbers of jobs in a queue (Q) are shown. The state diagrams for the other skill level distributions with optimal and random priorities were constructed as well, but because of space limitations they are omitted from this paper.

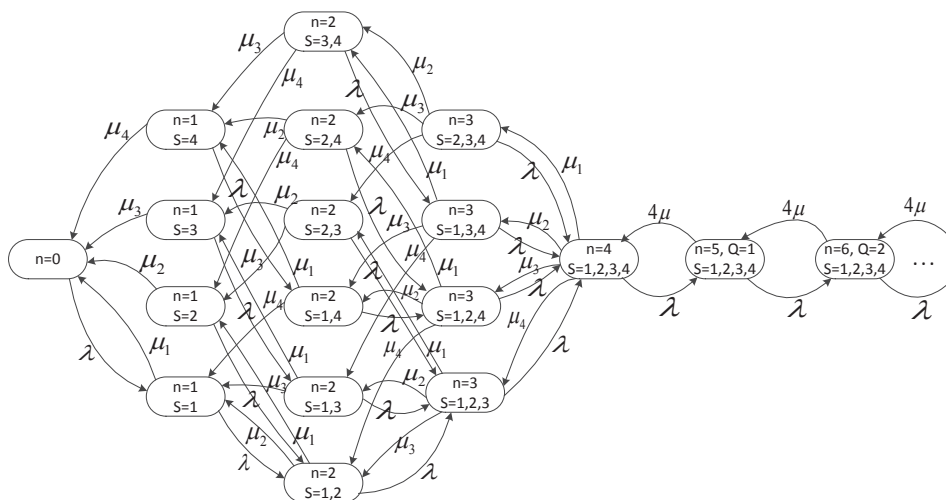


Figure 2: Markov state diagram for *no helping* (OPT) with skills uniformly distributed among the workers.

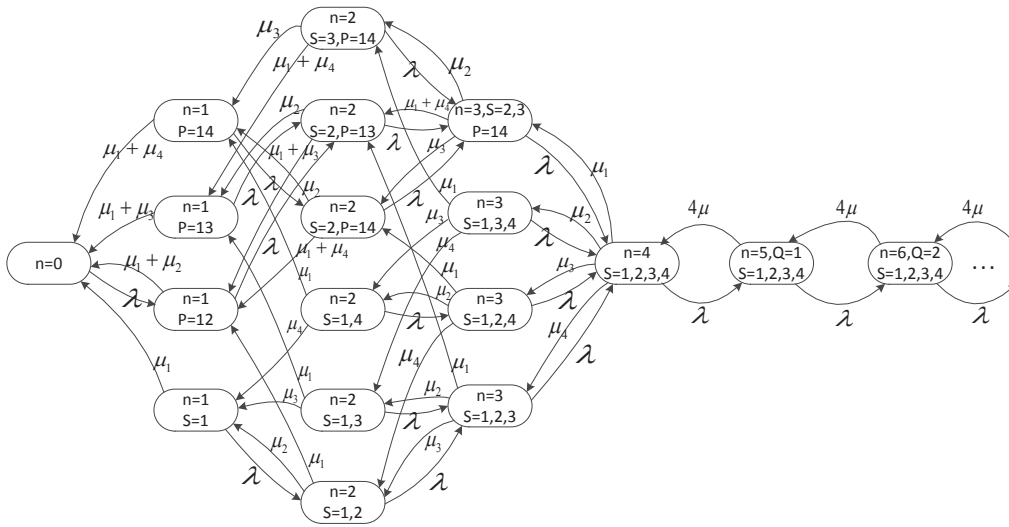


Figure 3: Markov state diagram for *floater* (OPT) with skills uniformly distributed among the workers.

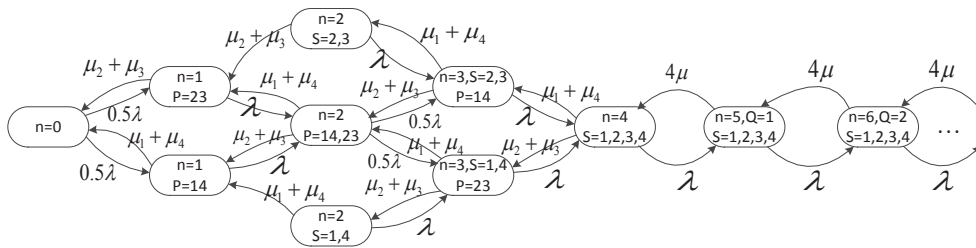


Figure 4: Markov state diagram for *pairs* (OPT) with skills uniformly distributed among the workers.

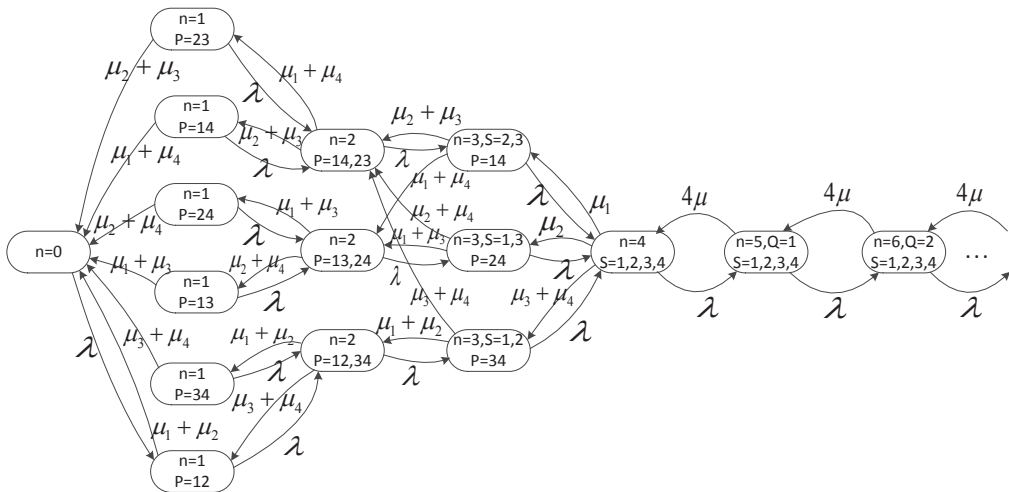


Figure 5: Markov state diagram for *complete helping* (OPT) with skills uniformly distributed among the workers.

Different experiments were simulated in Markov models by using the Monte Carlo method. The number of state transitions in each simulation run was 10 million. The simulator calculated the times the system was in each state, after which the steady-state probabilities were solved. The steady-state probabilities in each state were multiplied by

the number of jobs in the corresponding states to get the average number of products in the system (WIP = work-in-process). According to Little's law [15], the amount of WIP was divided by the arrival rate of jobs,  $\lambda$ , to obtain the average cycle time (CT) for one job.

In the experiments, the arrival rate of jobs is studied at  $\lambda = 2, 2.5$  and  $3$ . As the mean processing rate  $\mu = 1$ , the arrival rates generate the average system utilisation rates  $u = 0.5, 0.625$  and  $0.75$  (since  $u = \lambda/N\mu$ , and  $N = 4$ ). The actual processing rates for workers within different skill level distributions were presented in Table 1.

### 3.4. RESULTS

Table 3 presents the average job cycle times (CT) in the different experiments.

Table 3: Cycle times in the different experiments

System $u$	Identical skills (Identical)			Uniformly distributed skills (Uni)			Two skilled and two unskilled (2+2)			One skilled and three less skilled (1+3)		
	0.5	0.625	0.75	0.5	0.625	0.75	0.5	0.625	0.75	0.5	0.625	0.75
<b>NH-RAND</b>	1.088	1.214	1.511	1.096	1.222	1.517	1.104	1.229	1.520	1.092	1.218	1.514
<b>NH-OPT</b>	1.088	1.214	1.511	1.044	1.184	1.488	1.038	1.183	1.493	1.063	1.195	1.497
<b>FL-RAND</b>	0.795	0.971	1.311	0.761	0.936	1.281	0.762	0.938	1.276	0.762	0.937	1.282
<b>FL-OPT</b>	0.795	0.971	1.311	0.742	0.921	1.264	0.734	0.919	1.262	0.762	0.937	1.282
<b>P-RAND</b>	0.704	0.867	1.195	0.708	0.870	1.198	0.709	0.872	1.200	0.706	0.868	1.197
<b>P-OPT</b>	0.704	0.867	1.195	0.704	0.867	1.195	0.704	0.867	1.195	0.694	0.858	1.188
<b>CH-RAND</b>	0.666	0.822	1.145	0.669	0.821	1.144	0.670	0.824	1.145	0.667	0.821	1.143
<b>CH-OPT</b>	0.666	0.822	1.145	0.655	0.811	1.136	0.650	0.808	1.136	0.660	0.815	1.139

Figure 6 presents how different worker coordination policies (with random and optimal priorities) result in the average CT. In these experiments, the skill levels of workers are uniformly distributed and the system utilisation rate is studied at  $u = 0.5, 0.625$  and  $0.75$ .

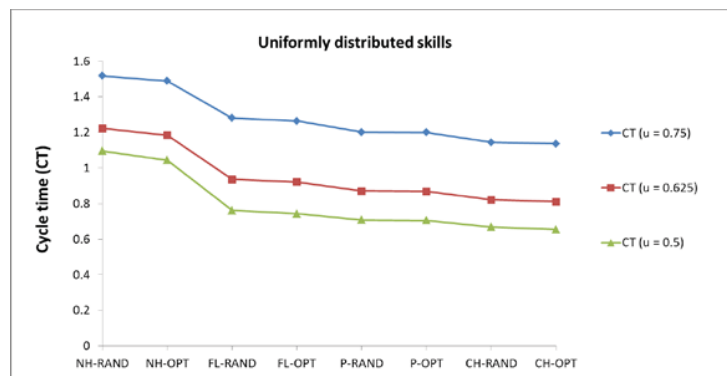


Figure 6 : Cycle time (CT) with different worker coordination policies. The skill levels of workers are uniformly distributed and the system utilisation rate is studied at  $u = 0.5, 0.625$  and  $0.75$ .

## 4. DISCUSSION

This section discusses the impacts of worker skill level distribution and optimal and random priorities, as well as the amount of helping (worker coordination policy itself), on system performance.

#### 4.1. IMPACTS OF DISTRIBUTION OF WORKER SKILL LEVELS

As the results in Table 3 show, regardless of the skill level distribution, the order of the performances (in CTs) of the worker coordination policies remains the same. In fact, the skill level distribution generally has a relatively small impact on CT within the policies under study. To mention the most significant impacts for various policies, when  $u = 0.75$ , FL-OPT 2+2 results in a 3.7% lower CT than FL-OPT *Identical*, FL-RAND 2+2 results in a 2.7% lower CT than FL-RAND *Identical*, and NH-OPT 2+2 results in a 1.2% lower CT than NH-OPT *Identical*. When  $u = 0.625$ , FL-OPT 2+2 results in a 5.4% lower CT compared to FL-OPT *Identical*, FL-RAND *Uni* results in a 3.6% lower CT compared to FL-RAND *Identical* and NH-OPT 2+2 results in a 2.6% lower CT than NH-OPT *Identical*. When  $u = 0.5$ , FL-OPT 2+2 results in a 7.6% lower CT than FL-OPT *Identical*, NH-OPT 2+2 results in a 4.6% lower CT than NH-OPT *Identical* and FL-RAND 1+3 results in a 4% lower CT than FL-RAND *Identical*.

The above comparisons show that the worker skill level distribution is at its most significant in FL-OPT, followed by FL-RAND and NH-OPT. In all those comparisons, the worst CT was obtained from identical skills. In practice, this can especially be seen within *floater* as the benefit from a floater is clearly at its smallest when all the workers are identical. When the greatest benefit from the *floater* policy is considered, it is obtained with 2+2 distribution, and, perhaps unexpectedly, not with 1+3 skill level distribution, in which the skill level of the floater is most separated from those of the others. This is certainly due to the benefit gained from another skilled worker within the skill-based assignment of jobs. The impact of the skill level distribution is only up to 2.4% within *complete helping* and up to 1.4% within *pairs*. The minor impact within *pairs* is due to the equal total skills between optimal pairs (except with 1+3 distribution), compared to which randomly selected pairs do not make a significant difference. In general, the results show how an equal and increased amount of helping among the workers diminishes the benefit gained from the skill levels of the workers.

As shown above, a lower system utilisation rate causes greater differences in CTs between different skill level distributions. This is due to the fact that fewer job arrivals increase the frequency of skill level-based (OPT) assignments of jobs.

#### 4.2. IMPACTS OF OPTIMAL AND RANDOM PRIORITIES

The significance of whether the RAND or OPT priority is used varies slightly, depending on the skill level distribution of the workforce and worker coordination policy. As expected, these priorities do not matter for all the policies with identical skills or for *floater* when the floater is skilled and the others less skilled (1+3) (Table 3).

In general, the greatest relative reductions in CT with OPT priority compared to RAND appear within the 2+2 skill level distribution followed by uniform distribution. More specifically, for 2+2 distribution, with *no helping* the reduction is from 1.8% (with  $u = 0.75$ ) up to 6% (with  $u = 0.5$ ). The corresponding reduction with *floater* is 1.1-3.7% and with *complete helping* 0.8-3%. With *pairs*, the greatest benefit from the OPT priority compared to RAND is obtained within the 1+3 skill level distribution (however, only from 0.8 up to 1.7%) as the unequal total skills between the pairs enable a benefit to be derived from allocating jobs optimally. In general, the above results show that assigning jobs optimally is especially significant when helping others is not allowed or is at least severely limited.

#### 4.3. IMPACTS OF WORKER COORDINATION POLICIES

As the results in Table 3 show, the worker coordination policy (who is allowed to help whom) itself has the greatest impact on CT. In all the experiments that were compared, *floater* is 13-31% better than *no helping*, *pairs* 4.1-11.3% better than *floater* and *complete helping* 4.3-7.7% better than *pairs*. These results highlight the significance of an incrementally increased flexibility for system performance. Especially with a small number of parallel stations ( $N = 4$ ), the greatest improvement is obtained by adding a floater to the system.

### 5. CONCLUSIONS

This paper studied worker coordination policies in an assembly system with four parallel stations and workers among whom there were various skill level distributions. These policies aim to avoid the idleness of workers by allowing them to help each other. The policies involve different restrictions in principle on who is allowed to help others, as well as random and optimal cases in who is to be helped and to whom to assign a job. The systems were studied as continuous-time Markov models.



The performances of different worker coordination policies in various experiments were compared as average cycle times of jobs. The results showed that:

- (1) the skill level distribution of the workforce plays the greatest role within the *float* policy, while a greater amount of helping (i.e. helping in *pairs* or *complete helping*) reduces the significance of the skill levels of the workers;
- (2) helping the least skilled ones and assigning jobs to the most skilled ones prior to others improves the performance of the system in relation to random priorities. Assigning jobs optimally matters especially at low system utilisation rates and when helping others is not allowed or is at least severely limited;
- (3) the amount of helping (who is allowed to help whom) itself has the greatest impact on performance.

For further research, it would be fruitful to incorporate different types of distributions related to the arrival and processing of jobs that reflect various situations in real production systems in the models. Another practical option would be to investigate dynamically changing worker coordination policies that consider the current level of the workload.

## REFERENCES

- [1] W. J. Hopp and M. P. Van Oyen: "Agile Workforce Evaluation: A Framework for Cross-training and Coordination", *IEEE Transactions*, Vol. 36, No. 10, pp. 919-940, 2004.
- [2] W. J. Hopp, E. Tekin and M. P. Van Oyen: "Benefits of Skill Chaining in Serial Production Lines with Cross-Trained Workers", *Management Science*, Vol. 50, No. 1, pp. 83-98, 2004.
- [3] R. R. Inman, W. C. Jordan and D. E. Blumenfeld: "Chained Cross-training of Assembly Line Workers", *International Journal of Production Research*, Vol. 42, No. 10, pp. 1899-1910, 2004.
- [4] L. Sennott, M. Van Oyen and S. Iravani: "Optimal Dynamic Assignment of a Flexible Worker on an Open Production Line with Specialists", *European Journal of Operational Research*, Vol. 170, No. 2, pp. 541-566, 2006.
- [5] A. Montano, J. R. Villalobos, M. A. Gutierrez and L. Mar: "Performance of Serial Assembly Line Designs under Unequal Operator Speeds and Learning", *International Journal of Production Research*, Vol. 45, No. 22, pp. 5355-5381, 2007.
- [6] S. Andradottir, H. Ayhan and D. G. Down: "Server Assignment Policies for Maximizing the Steady-state Throughput of Finite Queueing Systems", *Management Science*, Vol. 47, No. 10, pp. 1421-1439, 2001.
- [7] S. Gurumurthi and S. Benjaafar: "Modeling and Analysis of Flexible Queueing Systems", *Naval Research Logistics (NRL)*, Vol. 51, No. 5, pp. 755-782, 2004.
- [8] J. Peltokorpi, H. Tokola and E. Niemi: "Worker Coordination Policies in Parallel Station Systems: Performance Models for a Set of Jobs and for Continuous Arrival of Jobs", *International Journal of Production Research*, Vol. 53, No. 6, pp. 1625-1641, 2015.
- [9] W. C. Jordan, W. and S. C. Graves: "Principles on the Benefits of Manufacturing Process Flexibility", *Management Science*, Vol. 41, No. 4, pp. 577-594, 1995.
- [10] J. A. Buzacott: "Modelling Teams and Workgroups in Manufacturing", *Annals of Operations Research*, Vol. 126, No. 1-4, pp. 215-230, 2004.
- [11] E. S. Gel, W. J. Hopp and M. P. Van Oyen: "Hierarchical Cross-training in Work-in-Process-Constrained Systems", *IEEE Transactions*, Vol. 39, No. 2, pp. 125-143, 2007.
- [12] J. A. Buzacott: "Then and Now - 50 Years of Production Research", *International Journal of Production Research*, Vol. 51, No. 23-24, 2013.
- [13] J. E. Hunter, F. L. Schmidt and M. K. Judiesch: "Individual Differences in Output Variability as a Function of Job Complexity", *Journal of Applied Psychology*; Vol. 75, No. 1, 1990.
- [14] F. S. Hillier and G. J. Lieberman: "Introduction to Operations Research", 6th ed., McGraw-Hill, 1995.
- [15] J. D. Little: "A Proof for the Queueing Formula:  $L = \lambda W$ ", *Operations Research*, Vol. 9, No. 3, pp. 383-387, 1961.