

# Analysis of Bowl Effects on Stochastic Assembly Line

Mehmet Pınarbaşı, Hacı Mehmet Alağaç, Mustafa Yüzükırmızı, Bilal Toklu

Industrial Engineering Department, Hitit University, Çorum, 19030, Turkey.  
 Industrial Engineering Department, Kirikkale University, Kirikkale, 71451 Turkey  
 Department of Business Administration, Melikşah University, Kayseri, 38280, Turkey  
 Industrial Engineering Department, Gazi University, Ankara, 06570, Turkey  
 Phone: +90 (364) 227-4533; Fax: +90 (364) 227-4535, [mehmetpinarbas@hitit.edu.tr](mailto:mehmetpinarbas@hitit.edu.tr), [hmalagas@kku.edu.tr](mailto:hmalagas@kku.edu.tr),  
[myuzukirmizi@meliksah.edu.tr](mailto:myuzukirmizi@meliksah.edu.tr), [btoklu@gazi.edu.tr](mailto:btoklu@gazi.edu.tr)

**Abstract**— In this study, variation of the task times, and precedence relation effects are investigated to evaluate the line performance. A new solution procedure based on queueing networks and constraint programming is proposed to model and solve the Assembly Line Balancing Problem (ALBP). Station utilization, total average number of jobs and smoothness index are used as performance measures. Bowl effect, inverted bowl effect and variability imbalance which are seen in balanced lines are examined by using proposed procedure. Also effects of the variability on the line performance are reviewed. Literature data sets are utilized to assess the effectiveness of the procedure.

**Index Terms**— Assembly line balancing, bowl effect, constraint programming, queueing networks

## I. INTRODUCTION

Processing sub-parts to produce a final product on a serial line is defined as an assembly line (AL). Assembly lines are preferred flow line production systems in high quality and large volume productions. Assembly line balancing problem (ALBP) is to determine the appropriate task assignment while satisfying some restrictions with an objective.

### A. Bowl Phenomenon for ALBP

The output rate of an assembly line is related to distribution of task times. If the operation times are assumed to be stochastic, then the bowl phenomenon arises in unpaced lines. Bowl phenomenon states that the stations in the middle should have less utilization than start or end of the line for optimal or better outputs Scholl [1]. A typical bowl phenomenon chart is depicted in Figure 1 (Scholl [1]).

Buffer size between the stations influences existence of the bowl phenomenon, since output rate of the line can be changed when the buffer sizes in the center of the line are increased compared to start or end of the line. So determination of output rate is a crucial step to construct the bowl-shaped work load distributions. The output rates may be estimated by using the simulation or queueing theory for unpaced assembly lines (Scholl [1]).

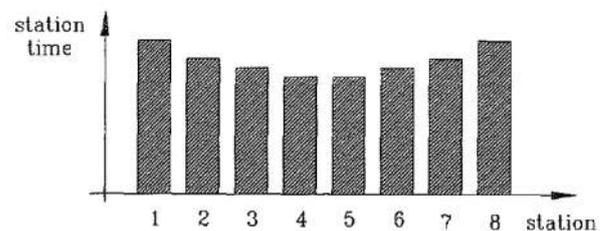


Fig. 1. Bowl Phenomenon

Increment of the output rate can be described with bowl phenomenon by increasing the task times or by changing the system variability in imbalanced assembly line Rao [2]. The coefficient of variation (CV) is used to determine the system variability.

There are some studies on the bowl phenomenon in the literature. Rao [2] analyses the serial production system with deterministic and exponential time in terms of the bowl phenomenon. He shows that if the differences in the coefficients of variation of production stages are generally less than 0.5, bowl phenomenon is constructed. When the differences exceed 0.5, variability imbalance is constructed.

Hillier and So [3] studied the effect of the coefficient of variation of operation times on the optimal allocation of storage space in production line systems. Task times are assumed as Coxian distribution. Providing that the overall optimal solution follows the bowl phenomenon, the optimal buffer allocation of buffer storage space fits an inverted bowl.

Das et al. [4] recently presented a study to evaluate the bowl phenomenon by using the simulation for assembly line. In this study, we investigate the bowl phenomenon for the assembly line which has normal and exponential task times with variance among 1 and 16. Bowl effect rises on the optimal solution with the normally distribution task times and also inverted bowl effect rises with the exponential task times.

This paper is organized as follows: After this introduction, a literature review has been given in Section 2. The mathematical model of type-F problem is presented in Section 3. The proposed solution procedure, constraint programming model and queueing network model (Diffusion approximation) are given in Section 4 respectively. The computational results with the proposed model to a set of test problems are presented

in Section 5. And also bowl effect is examined in Section 5. Conclusions are given in the last section.

## II. LITERATURE REVIEW

AL is classified by line layout in three settings: traditional/serial line, u-type line and parallel line with single, multi or mix model. ALBP is first modeled mathematically by Salvesson [5]. Following this study, several studies considering different line layouts have been investigated in the literature. In these studies heuristic or meta-heuristic procedures have been proposed as well as exact solution algorithms. To give a brief on ALBP studies, literature reviews are examined in respect of these authors: Scholl [1], Erel and Sarin [6], Scholl and Becker [7], Becker and Scholl [8].

ALBP is classified in two types regarding the task times: deterministic and stochastic. Most assembly line studies assume deterministic or constant task time due to the skilled and motivated workers during the period of continuous process. But in real life the task times may vary. Due to machine breakdowns, motivation loss, lack of training, unqualified operators, complex tasks etc., the task times are stochastic. This problem is called as Stochastic ALBP (SALBP) where each task has a probability function with a mean and variance. This problem is also NP-hard combinatorial problem alike the deterministic case Scholl [1].

There are three main classes in ALBP in terms of the presence of objective function: Type 1, Type 2 and Type-F. Type 1 aims to minimize the number of stations for a given number of cycle time. Type 2 seeks to minimize the cycle time for a given number of stations. When there is no objective function, the ALBP problem reduces to a problem which consists of finding out whether or not a task assignment exists for a given number of stations and cycle time. In the literature, type 1 problems are studied more than type 2 and type-F.

This research studies serial line, stochastic ALBP with a new objective function of maximizing the utilization of the stations.

The stochastic ALBP was first addressed by Moodie and Young [9]. Kottas and Lau 1981 [10] presented a heuristic procedure for paced assembly line with stochastic task times. Their objectives were minimizing the sum of incompleteness and labor cost. Silverman and Carter [11] analyzed the effect of stochastic task times on total operating costs of assembly line under the assumption that the line is stopped. Sarin et al. [12] developed a methodology to solve the stochastic assembly line balancing problem for the objective of total labor cost and incompleteness cost. Their methodology was based on DP and branch-and-bound procedure. Also several studies on other type assembly line balancing problems have been reported in the literature: for mixed model (Xiaobo et al. [13], Sun and Li [14]), for u-type problem (Ağpak and Gökçen [15], Özcan et al. [16]), for parallel line (Cakir et al. [17], McMullen and Tarasewich [18]) and for two-sided problem (Özcan [19]).

ALB problems are usually modeled as an integer programming and several solution methods are proposed accordingly as exact, heuristics and metaheuristics. Carraway [20] presented two dynamic programming algorithms for the

problem of minimizing the number of workstations with normally distributed task times. A simulated annealing technique was presented by Suresh and Sahu [21]. Liu et al. [22] proposed a heuristic algorithm to solve the type-2 stochastic assembly line balancing problem.

This study develops Constraint Programming (CP) methodology as a solution approach. CP is an alternative programming technique which utilizes logical expressions using computer programming Apt [23] and combines effectiveness of finding a solution with linear programming especially, logical expressions which can be hard to define in linear programming can be defined easily in constraint programming.

CP has been used to solve several NP-Hard combinatorial problems. Especially, scheduling applications with various production systems can be found in the literature. Khayat et al. [24], Hladik et al. [25], Zeballos [26], Trojct et al. [27], Li and Womer [28], Terekhov et al. [29] are presented studies in recent years.

Another unique contribution of this study is modeling assembly line as a queueing network to incorporate stochastic task times. Queueing models are used to analyze the performance of several manufacturing systems. In queueing applications, assembly line is generally dealt with a system which has exponential service times and Poisson arrival times e.g. Latouche and Neuts [30], Lipper and Sengupta [31]. Less applications of assembly line in queueing theory is related to deterministic times. Duenyas and Hopp [32] considered the assembly system feeding through more than one serial line with deterministic task time. Azaron et al. [33] developed a method which optimizes the due date in the multi stage assembly systems using decomposition method. Manitz [34] studies the production process belonging to fractional assembly lines. For these lines, he shows that arriving products following the previous station has unsynchronized stochastic task times and queueing can affect these lines in terms of the buffer capacity. And Lazaro and Perez [35] has modeled the assembly line as a closed loop network which consist of the machines that have the blocking and idle time probability with buffers on the conveyor. Manufacturing applications (e.g. flexible manufacturing systems, transfer and production lines, assembly/disassembly systems, flow lines etc.) of queueing networks can be investigated in detail from these literature review studies: Jackman and Johnson [36], Papadopoulos and Heavy [37] and Govil and Fu [38].

To the best of the authors' knowledge, a method based on the queueing networks and constraint programming has not been developed by the researchers for ALBP. In this study, a novel solution procedure is proposed for stochastic ALBP. Firstly, a constraint programming model is utilized for obtaining the feasible task assignments. Then, an open queueing network algorithm of Diffusion Approximation combined with constraint programming model is used to measure the performance of the line. Also bowl phenomenon is investigated on the stochastic unpaced assembly line with general service time distributions.

### III. MATHEMATICAL MODEL OF ALBP

In this section, we give the underlying assumptions and the mathematical model of the Type-F ALBP. This problem minimizes the cycle time for a given number of stations. In practical, Type-F problems are more suitable in terms of the rebalancing the line. In this context, basic assumptions of related ALBP are given as follows:

- (i) Task times are probability distributions in which the parameters are known.
- (ii) The precedence relationships among the tasks are known.
- (iii) A task must be assigned to only one station.
- (iv) Tasks are indivisible.
- (v) A single model assembly line is considered.
- (vi) Parallel tasks and parallel stations are not allowed.
- (vii) An unpaced serial line is considered.
- (viii) A worker is assigned at each station with similar specifications.

Accordingly, we adopt the mathematical model of deterministic type-F problem which was developed by Patterson and Albracht [39].

Subject to

$$\sum_{k=1}^{\|W\|} x_{ik} = 1 \quad \forall i \in T \quad (1)$$

$$\sum_{k=1}^{\|W\|} k \cdot x_{ik} \leq \sum_{k=1}^{\|W\|} k \cdot x_{jk} \quad \text{for all } (i, j) \in Pr \quad (2)$$

$$c \geq \sum_{i=1}^{\|T\|} t_i \cdot x_{ik} \quad \forall k \in W \quad (3)$$

$$x_{ik} \in \{0, 1\} \quad \forall (i, k) \in (T, W) \quad (4)$$

In this model, Constraint (1) ensures that each task must be assigned to exactly one station. Precedence relations among the tasks are satisfied through Constraint (2). Constraint (3) is the cycle time constraint and it ensures that any station time must not be exceeded the cycle time. Constraint (4) is variable definition and it is satisfied the binary variable conditions.

Traditional ALBP has two main objective functions: minimization of the cycle time and minimization of number of the stations. But several objectives in terms of the different characteristics of the assembly line are considered to balance the line as mentioned above. In this study we consider the objective function as equalization of the station utilization. In this way, the workload will be distributed among the station as uniformly as possible and the line will be balanced. To equalize the station utilization, we use to smoothness index given in Equation (5) Scholl [1]. Task assignment which has the minimum smoothness index value is an optimum assignment in efficient assembly line.

$$SI = \sqrt{\sum_{k=1}^{\|W\|} (\rho_{\max} - \rho_k)^2} \quad (5)$$

Also total average number of jobs in the station as an objective is considered to balance the line by reducing the number of the incompleted tasks. These objective functions values are computed using queueing networks approach.

Earliest and latest station bound are satisfied for ALBP as constraints in the CP model. In this study, these bound proposed from Klein and Scholl [40] are considered in Equation (6) and Equation (7).  $E_i$  is the least station number to be assigned for a task  $i$ .  $L_i$  is the biggest station number to be assigned for a task  $i$ .  $P_i^*$  is a set of the predecessors of task  $i$  and  $F_i^*$  is a set of the followers of the task  $i$ .

$$E_i \leq \left\lceil \frac{\left( t_i + \sum_{h \in P_i^*} t_h \right)}{ct} \right\rceil \quad \forall i \in T \quad (6)$$

$$L_i \leq \|W\| + 1 - \left\lfloor \frac{\left( t_i + \sum_{h \in F_i^*} t_h \right)}{ct} \right\rfloor \quad \forall i \in T \quad (7)$$

### IV. SOLUTION PROCEDURE

#### A. Constraint Programming

Constraint Programming (CP) is an alternative programming technique, generated by combining effectiveness in achieving the optimal solution of linear programming and easy definition property of logical expressions of computer programming methods. Main concept of CP is “constraints”. Domain is a set of values that can be assigned to variables. Each constraint is defined as relation of between some variables related to their domains.

To formulate a problem as a constraint programming model:

- Define the variables and constraints and domains associated with variables
- Select a language to be identified the constraints Apt [23].

CP is classified the two main type in terms of the existence of the objective function. If there is no objective function, CP model is named Constraint Satisfaction Problem. If there is an objective which is a function such as minimizing or maximizing, this problem is Constraint Optimization Problem.

CP uses a systematic procedure to find a feasible solution which consists of two main steps.

1. **Search Tree:** In the search tree, a decision variable is declared as a node and a possible assignment shows a branch related to variables. The search starts with an empty assignment and proceeds until there are no variables that can be assigned a value. If the search could not be reached a possible solution, backtracking mechanism is executed to try some other branches. We can use four strategies according to problem size and problem type as follows in the used software:

- (i). Depth first strategy

- (ii). Multi-point strategy
- (iii). Restart strategy
- (iv). Automatic

In this study, automatic search strategy is preferred as default setting.

2. **Propagation and domain reduction:** The propagation is used to filter the variable domains by eliminating the inconsistent values. As soon as a variables' domain is changed, constraints related to that variable are propagated. Domain reduction is a process which removes the non-assigned variable values that do not satisfy the constraints. So we obtain a consistent assignment at every iteration of the backtracking.

### B. Constraint Programming Model for ALBP

In this study, we use CP to assign the task to the stations for assembly line balancing problems. All feasible task assignments are found via CP under precedence constraints, bound of earliest and latest station number of tasks. So proposed CP model for ALBP is as follows:

**Decision variable:**

$$StationNumber[Task_i] \quad \forall i \in T \quad (8)$$

Decision variable is stated the assigned number of station for task  $i$  and is an integer value between  $1, \dots, \|W\|$ .

**Precedence relations:** Precedence between task  $i$  and  $h$  is indicated as follows.

$$StationNumber[Task_i] \leq StationNumber[Task_h] \quad (9)$$

for all  $(i, h) \in Pr$

**Occurrence and station restrictions:** Each task must be assigned to a station.

$$count(StationNumber[Task_i; \forall i \in T]; j) \geq 1 \quad \forall j \in W \quad (10)$$

"count" as a function is a special construct in IBM ILOG CP Software which is performed the proposed constraint programming model. Using this form, at least one of the  $StationNumber_i$  variables has taken the station number  $j$ .

**Earliest and latest stations restrictions:** The variable of  $StationNumber_i$  must be greater or equal to and less or equal to:

$$StationNumber_i \geq E_i \quad \forall i \in T \quad (11)$$

$$StationNumber_i \leq L_i \quad \forall i \in T \quad (12)$$

### C. Queuing Network Model and Performance Evaluation

Queuing network can be classified as open and closed networks, topologically. When the jobs can enter into any node from outside and leave the network from any node, this queuing network is defined as **open**. But if jobs cannot enter into and leave the network, this system is said to be **closed**. We depicted an open queuing network in Fig. 2. The arrows

indicate the flow of workpieces, the rectangles represent the buffers and circles indicate the service stations. A workpiece is launched down the material handling system (e.g. pallets, conveyors etc.). Workpieces are processed into the stations respectively. They leave the system after processing at the last station. If the buffer capacity is not infinite and the buffer is full, the workpieces wait for a while and the current station is blocked.



Fig. 2. An open queuing network

A workstation in the assembly line can be considered as a service station in the queuing network. Every node in queuing model represents stations which the set of tasks are assigned to. The stochastic assembly line has a general station time consisting of a mean tasks time and variances inherently. When considered from this aspect, we could model an assembly line as a queuing network. The task assignment into the nodes and the conceptual framework of such assembly line as queuing network can be presented in Fig. 3.

**THEOREM:** If a set of task assigned into a station are normally distributed with a certain averages and variances, total time of that station can be determined by using the aggregation property of normal distributions. The station time  $t(S_k)$ :  $N(\mu_k^s, \sigma_k^s)$  Equation (15) is obtained in such a way that as it is given in Equation (13) for average of station time and Equation (14) for variance of station time.

$$\mu_k^s = \sum_{j \in S_k} \mu_j^t \quad k = 1, \dots, \|W\| \quad (13)$$

$$\sigma_k^s = \sqrt{\sum_{j \in S_k} (\sigma_j^t)^2} \quad k = 1, \dots, \|W\| \quad (14)$$

$$t(S_k): N(\mu_k^s, \sigma_k^s) \quad k = 1, \dots, \|W\| \quad (15)$$

Generally, CV is used to characterize the relation between mean and variance. CV is the normalized measure of dispersion for a probability distribution. It can be concisely defined as the ratio of standard deviation ( $\sigma$ ) to the mean ( $\mu$ ) as shown in Equation (16). CV is used to convert the deterministic task times to the stochastic task times in the proposed procedure as the input parameters.

$$CV = \frac{\sigma}{\mu} \quad (16)$$

Accordingly, an assembly line can be depicted as an open queuing network with arrival rate  $\lambda$  and service (station) rate  $\mu_i$  ( $i = 1, 2, \dots, \|W\|$ ). There is no blocking in the system and buffer size between each station is infinite. As soon as a job is finished at any station within the current station time, it is released it onto the conveyor belt. This situation can be described as unpaced line in assembly line balancing literature. Arrival rate in queuing networks  $\lambda$  can be defined as the system workload as mentioned by the production rate in the assembly line.

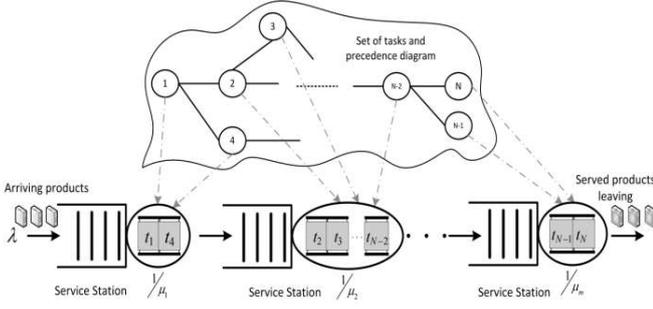


Fig. 3. Framework of assembly line queueing model

#### D. Diffusion Approximation

Considering that an assembly line where task times have an average and a variance, we can model this line with general time queueing network (with Kendall notations as GI/G/1). GI/G/1 queueing network consists of general arrival and service rate for each node in the network (i.e. for each station in the line). So to evaluate the line performance we use Diffusion Approximation as a GI/G/1 model approximation technique. Although deriving the diffusion approximations are very complex, application of the diffusion approximation on any problem is very simple (Bolch et al., [41]).

The overall approximation method of Diffusion approximation algorithm can be summarized as follows:

Step 1. Compute the utilizations for each nodes

$$\rho_k = \frac{\lambda \cdot e_k}{\mu_k} \quad \forall k \in W \quad (17)$$

Step 2. Compute the coefficient of variation of interarrival times for each nodes.

$$c_A^2 = 1 + \sum_{j=0}^{|W|} (c_{Bj}^2 - 1) p_{jk}^2 \cdot e_j \cdot e_k^{-1} \quad \forall k \in W \quad (18)$$

Step 3. Update the utilizations

$$\hat{\rho}_k = \exp\left(-\frac{2(1-\rho_k)}{c_{Ak}^2 \cdot \rho_k \cdot c_{Bk}^2}\right) \quad \forall k \in W \quad (19)$$

Step 4. Compute the mean number of jobs for each nodes.

$$\bar{K}_k = \frac{\rho_k}{1-\hat{\rho}_k} \quad \forall k \in W \quad (20)$$

Traffic intensity ( $ti$ ) is a general result and an important relation to prevent the blocking the line in GI/G/1 queues. When  $ti > 1$ , the mean number of arrivals into the line exceeds the maximum mean service rate of the line. In this situation, the line is not steady-state conditions any more.  $ti < 1$  must be strictly fulfilled to reach the steady-state conditions.

$$ti = \frac{\lambda}{\mu_k} \quad \forall k \in W \quad (21)$$

#### E. Proposed Solution Procedure

A two-step algorithm is proposed to balance and evaluate the stochastic assembly line by using CP and queueing network. In the first step, feasible task assignment combinations are generated using Constraint Programming model. Precedence relations, assignment restrictions and task times are inputs of this step which are controlled with station restrictions. Station times and task assignments are output of this step. In the next step, the feasible combinations are evaluated using queueing network approach. The best combination is determined under control the pre-defined algorithm parameters. The Integration Definition functional modeling is shown in Fig. 4 with inputs, outputs, controls and mechanism of the procedure.

The corresponding proposed solution algorithm is explained in details with the following steps (where **CurSI**=smoothness index of current assignment, **BestSI**= smoothness index of best assignment, **CurAssign**=current task assignment by generating CP, **BestAssign**=best task assignment)

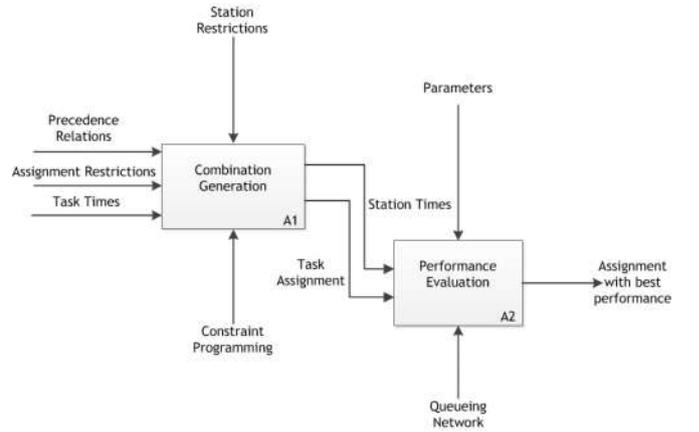


Fig. 4. Proposed solution procedure

- Step 0. **CurSI**=BigNumber, **BestSI**=BigNumber, **CurAssign**= $\emptyset$ , **BestAssign**= $\emptyset$
- Step 1. Generate a feasible task assignment and stations times satisfying all ALBP constraints by using CP model.
- Step 2. Run the diffusion approximation based on the current task assignment.
- Step 3. Calculate performance measurements related to current assignment (*SI, mean  $\rho$ , total K*).
- Step 4. If **CurSI** < **BestSI** then **BestSI** = **CurSI** and **BestAssign** = **CurAssign**, else go to Step 5.
- Step 5. If there is no feasible task assignment obtained from CP model then go to Step 6, else go to Step 1.
- Step 6. Terminate the algorithm and report the assignment with best performance.

#### V. COMPUTATIONAL RESULT

We obtain feasible task assignment combinations using the CP model. Then performance evaluation is carried out by means of the diffusion approximation based on the assembly line queueing model. The task assignment that has minimum

smoothness index value is optimal assignment. Performance measurements are station utilizations and total average number of jobs in the line.

Proposed procedure is coded by using IBM ILOG CP Optimizer software package and all experiments are performed in personal computer in Intel i5-2410M 2.30 GHz processor and 4 GB RAM. Effect of the variation of the task times and on the assembly line performance and balance is studied.

#### A. Effects of the variation of the task times

A task time depends on some system parameters such as complexity of task, experience and motivation of operator and machine breakdowns etc. In this case, variation of task time can be different for each task in the line.

Three test problems in the literature such as Jaeschke, Jaskson and Mitchell are tested with proposed procedure. Data sets are tested with different variation of task time in this stage. The task times of the original problems are used as the means of the tasks and the task variance are randomly generated as in Carraway [20] and Özcan [19]. In these studies, low task variances are generated between zero and  $(t_i/4)^2$  and high task variances are generated between zero and  $(t_i/2)^2$ . In this study, these variance levels are converted the coefficient of variation using Equation (16) and we used the CV as [0, 0.25] for low task CV, [0, 0.5] for medium task CV and [0, 1] for high task CV. Each test problem is tested five times using proposed procedure. Mean performance measurements of these five run are reported. Computational results are given in Table 1 - Table 3.

As it can be seen at Table 1 – Table 3, total average number of jobs in the line changes with different task CV levels. Considering the Mitchell problem, when the  $\lambda$  is assumed to 22, the numbers of jobs are obtained as 9.418 for low task CV, 10.252 for medium task CV and 20.422 for high task CV. This situation is same for the other test problems.

Computational results show that system workload ( $\lambda$ ) has a direct effect on the utilization of stations. We can easily observe this conclusion by analyzing Jaeschke sample problem. In this problem, mean line utilizations are obtained as 0.728 when  $\lambda$  is 7, 0.637 when  $\lambda$  is 8 and 0.567 when  $\lambda$  is 9. When the coefficient of variation of tasks are changed to different levels, mean utilization of the line remains constant. This situation arise from determining the utilization step in diffusion approximation where is obtained by using the Equation (17).

A remarkable result obtained from computational results is that optimum task assignment combination has been different when the variations of tasks are changed. Considering Jackson with four stations in Table 3, for  $\lambda=14$  optimum task assignment to the stations is (1,2,3,5)[13]; (4,6,7)[12]; (8,9)[11]; (10,11)[9] for low variation; (1,2,5,6)[11]; (4,8)[13]; (3,7,10)[12]; (9,11)[9] for medium variation; (1,3,5)[11]; (2,4,7)[12]; (6,8,10)[13]; (9,11)[9] for high variation (values in box bracket imply the mean station times).

TABLE I.

#task	#station	CV	$\lambda$	SI	mean $\rho$	total K	CPU time
9	3	Low	14	0.258	0.881	-	0.140
			15	0.240	0.822	2.962	0.130
			16	0.225	0.771	2.593	0.220
		Medium	14	0.258	0.881	-	0.140
			15	0.240	0.822	3.035	0.740
			16	0.225	0.771	2.747	0.180
	High	14	0.258	0.881	-	0.200	
		15	0.240	0.822	4.771	0.140	
		16	0.225	0.771	2.981	0.190	
	4	Low	10	0.224	0.925	-	0.240
			11	0.203	0.841	3.961	0.240
			12	0.186	0.771	3.383	0.340
		Medium	10	0.224	0.925	-	0.250
			11	0.203	0.841	4.069	0.250
			12	0.186	0.771	3.411	0.290
		High	10	0.224	0.925	-	0.300
			11	0.203	0.841	9.706	0.290
			12	0.186	0.771	3.579	0.260
7		Low	7	0.728	0.755	-	0.900
			8	0.637	0.661	4.722	0.840
			9	0.567	0.587	4.138	0.770
	Medium	7	0.728	0.755	-	0.930	
		8	0.637	0.661	4.854	1.130	
		9	0.567	0.587	4.231	0.900	
	High	7	0.728	0.755	-	1.090	
		8	0.637	0.661	7.630	0.890	
		9	0.567	0.587	5.777	1.040	

TABLE II.

#task	#station	CV	$\lambda$	SI	mean $\rho$	total K	CPU time	
11	3	Low	16	0.000	0.938	5.903	11.240	
			17	0.000	0.882	3.907	12.140	
			18	0.000	0.833	3.182	12.280	
		Medium	16	0.000	0.938	6.233	10.840	
			17	0.000	0.882	3.960	12.570	
			18	0.000	0.833	3.206	10.830	
		High	16	0.000	0.938	10.673	10.850	
			17	0.000	0.882	5.472	11.120	
			18	0.000	0.833	3.618	9.680	
		4	Low	13	0.353	0.865	-	19.110
				14	0.327	0.804	5.827	25.040
				15	0.306	0.750	4.028	19.330
	Medium		13	0.353	0.865	-	19.580	
			14	0.327	0.804	6.125	19.760	
			15	0.306	0.750	4.041	19.390	
	High		13	0.353	0.865	-	18.540	
			14	0.327	0.804	8.181	19.550	
			15	0.306	0.750	4.698	19.210	
	5	Low	11	0.464	0.818	-	21.660	
			12	0.425	0.750	4.057	20.140	
			13	0.392	0.692	3.603	20.900	
		Medium	11	0.464	0.818	-	26.460	
			12	0.425	0.750	4.338	20.860	
			13	0.392	0.692	3.769	25.56	
High	11	0.464	0.818	-	26.53			
	12	0.425	0.750	6.474	22.33			
	13	0.392	0.692	4.989	20.02			

TABLE III.

COMPUTATIONAL RESULTS OF MITCHELL PROBLEM ( $cv_A^2 = 0,5$ )

#task	#station	CV	$\lambda$	SI	mean $\rho$	total K	CPU time
21	3	Low	36	0.000	0.972	11.073	19.030
			37	0.000	0.946	6.558	16.760
			38	0.000	0.921	5.028	20.310
		Medium	36	0.000	0.972	11.769	20.000
			37	0.000	0.946	6.742	19.100
			38	0.000	0.921	5.086	19.070
	High	36	0.000	0.972	14.840	18.380	
		37	0.000	0.946	8.882	18.210	
		38	0.000	0.921	6.034	25.290	
	5	Low	22	0.000	0.955	9.418	100.600
			23	0.000	0.913	6.589	105.490
			24	0.000	0.875	5.524	92.760
		Medium	22	0.000	0.955	10.252	106.660
			23	0.000	0.913	6.988	109.940
24			0.000	0.875	5.574	108.040	
High		22	0.000	0.955	20.422	80.290	
	23	0.000	0.913	8.544	112.580		
	24	0.000	0.875	6.606	108.190		
8	Low	14	*	*	*	*	

\*We do not reach the optimal solution

### B. Discussion of bowl effects

Fig. 2 shows bowl phenomenon in terms of the station utilizations for Jaeschke and Jackson problems. For Jackson problem inverted bowl is found out with 4 stations and high CV level. But, variability imbalance arises for Jaeschke problem 7 stations and medium CV level. Similar situation has been reported by Rao [2]. Rao [2] says that load from the more variable stages should be transferred to the less variable ones for variability imbalance to balance the line. And also Fig. 3 shows that variability imbalance for these problems respectively in terms of the total average number of jobs in the line. Mitchell problem shows almost same bowl effect of stations for both performance measurements.

Variations of between stations are related to station task time distributions and coefficient of variations of task times. We can say that while a job proceeds from a station with high CV level to a station with low CV level, line can be more variable and bowl effect can be return to inverted bowl or variability imbalance effects. And also if a job proceeds from a station with high station time to a station with low station time, same situation can be occurred for optimum balanced line.

These situations are defined the possible bowl effect shapes for balanced stochastic assembly line. The results show that bowl phenomenon affects shapes in terms of the station utilizations and total average number of jobs. So, balancing the line with these performance measurements is meaningful and realistic for assembly line balancing problems.

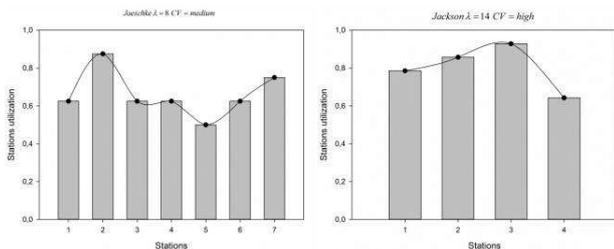


Fig. 2. Bowl effect on station utilizations

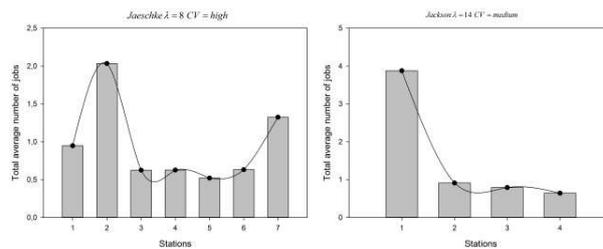


Fig. 3. Bowl effect on total average number of jobs

## VI. CONCLUSION

In this paper, stochastic single model assembly line balancing problem is studied by using constraint programming and queueing network approach. The resulting bowl phenomenon is analyzed in terms of the performance measurements of stations utilizations and total average number of jobs in the line. Inverted bowl effect and variability imbalance is shaped for test problems. Also an interesting result is that the optimum task assignment has been changed for optimum line balance while task time variability is changed. Proposed procedure is effective for the problem. For further researches, other types of assembly line balancing problems should be interest.

## VII. REFERENCES

- [1] A. Scholl, *Balancing and Sequencing Assembly Lines*. 2<sup>nd</sup> Edition, Physica, Heidelberg, 1999.
- [2] P.N. Rao, "A generalization of the 'bowl phenomenon' in series production systems", *International Journal of Production Research.*, vol. 14(4), pp. 437-443, 1976.
- [3] F.S. Hillier and C.K. So, "The Effect of the Coefficient of Variation of Operation Times on the Allocation of Storage Space in Production Line Systems", *IIE Transactions*, vol. 23(2), pp. 198-206, 1991.
- [4] B. Das, A. Garcia-Diaz, A.C. MacDonald, K.K. Ghoshal, "A computer simulation approach to evaluating bowl versus inverted bowl assembly line arrangement with variable operation times", *The International Journal of Advanced Manufacturing Technology*, vol. 51, pp. 15-24, 2010.
- [5] M.E. Salveson, "The assembly line balancing problem", *Journal of Industrial Engineering*, vol. 6, pp. 18-25, 1955.
- [6] E. Ereil, S.C. Sarin, "A survey of the assembly line balancing procedures", *Production Planning & Control*, vol. 9(5), pp. 414-434, 1998.
- [7] A. Scholl, C. Becker, "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing", *European Journal of Operational Research*, vol. 168(3), pp. 666-693, 2006.
- [8] C. Becker, A. Scholl, "A survey on problems and methods in generalized assembly line balancing", *European Journal of Operational Research*, vol. 168(3), pp. 694-715, 2006.
- [9] C.L. Moodie, H.H. Young, "A heuristic method of assembly line balancing for assumptions of constant or variable work element times", *Journal of Industrial Engineering*, vol. 16, pp 23-29, 1965.
- [10] J.F. Kottas, H.S. Lau, "A stochastic line balancing procedure", *International Journal of Production Research*, vol. 19, pp. 177-193, 1981.
- [11] F. N. Silverman, J. C. Carter, "A cost-based methodology for stochastic line balancing with intermittent line stoppages", *Management Science*, vol. 32(4), pp. 455-463, 1986.
- [12] S. C. Sarin, E. Ereil, E. M. Dar-El, "A methodology for solving single-model, stochastic assembly line balancing problem", *Omega*, vol. 27, pp. 525-535, 1999.

- [13] Z. Xiaobo, D. Xu, H. Zhang, Q.M. He, "Modeling and analysis of a supply-assembly-store chain", *European Journal of Operational Research*, vol. 176(1), pp. 275-294, 2007.
- [14] Q.G. Sun, Y.D. Li, "Study on Stochastic Mixed-Model Assembly Line", *Proceeding of the International Conference on Mechanic Automation and Control Engineering (MACE)*, Wuhan, China, June 26-28, pp 3446 – 3449, 2010.
- [15] K. Ağpak, H. Gökçen, "A chance-constrained approach to stochastic line balancing problem", *European Journal of Operational Research*, vol. 180(3), pp. 1098-1115, 2007.
- [16] U. Özcan, T. Kellegöz, B. Toklu, "A genetic algorithm for the stochastic mixed-model U-line balancing and sequencing problem", *International Journal of Production Research*, vol. 49(6), pp. 1605-1626, 2011.
- [17] B. Cakir, F. Altıparmak, B. Dengiz, "Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm", *Computers & Industrial Engineering*, vol. 60(3), pp. 376-384, 2011.
- [18] P. R. McMullen, P. Tarasewich, "Using Ant Techniques to Solve the Assembly Line", *IIE Transactions*, vol. 35(7), pp 605-617, 2003.
- [19] U. Özcan, "Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm", *European Journal of Operational Research*, vol. 205(1), pp. 81-97, 2010.
- [20] R.L. Carraway, "A dynamic programming approach to stochastic assembly line balancing", *Management Science*, vol. 35(4), pp. 459-471, 1989.
- [21] G. Suresh, S. Sahu, "Stochastic assembly line balancing using simulated annealing", *International Journal of Production Research*, vol. 32(8), pp. 1801-1810, 1994.
- [22] S. B. Liu, H. L. Ong, H. C. Huang, "A bidirectional heuristic for stochastic assembly line balancing Type II problem", *The International Journal of Advanced Manufacturing Technology*, vol. 25(1-2), pp. 71-77, 2005.
- [23] K.R. Apt, *Principles of Constraint Programming*. Cambridge University Press, 2003.
- [24] G.E. Khayat, A. Langevin, D. Riopel, "Integrated production and material handling scheduling using mathematical programming and constraint programming", *European Journal of Operational Research*, vol. 175(3), pp. 1818-1832, 2006.
- [25] P.E. Hladik, H. Cambazard, A.M. Déplanche, N. Jussien, "Solving a real-time allocation problem with constraint programming", *Journal of Systems and Software*, vol. 81(1), pp. 132-149, 2008.
- [26] L.J. Zeballos, "A constraint programming approach to tool allocation and production scheduling in flexible manufacturing systems", *Robotics and Computer-Integrated Manufacturing*, vol. 26(6), pp. 725-743, 2010.
- [27] M. Trojet, F. H'Mida, P. Lopez, "Project scheduling under resource constraints: Application of the cumulative global constraint in a decision support framework", *Computers & Industrial Engineering*, vol. 61(2), pp. 357-363, 2011.
- [28] H. Li, K. Womer, "Optimizing the supply chain configuration for make-to-order manufacturing", *European Journal of Operational Research*, vol. 221(1), pp. 118-128, 2012.
- [29] D. Terekhov, M.K. Doğru, U. Özen, J.C. Beck, "Solving two-machine assembly scheduling problems with inventory constraints", *Computers & Industrial Engineering*, vol. 63(1), pp. 120-134, 2012.
- [30] G. Latouche, M.F. Neuts, "Efficient algorithm solutions to exponential tandem queue with blocking", *SIAM J Algebra Disc Method*, vol. 1(1), pp. 93-106, 1980.
- [31] E.H. Lipper, B. Sengupta, "Assembly-like queues with finite capacity: bounds, asymptotics and approximations", *Queueing systems: Theory and Applications*, vol 1, pp. 67-83, 1986.
- [32] I. Duenyas, W.J. Hopp, "Estimating the throughput of an exponential CONWIP assembly system", *Queueing Systems*, vol. 14, pp. 135-157, 1993.
- [33] A. Azaron, H. Katagiri, K. Kato, M. Sakawa, "Modelling complex assemblies as a queueing network for lead time control", *European Journal of Operational Research*, vol. 174(1), pp. 150-168, 2006.
- [34] M. Manitz, "Queueing-model based analysis of assembly lines with finite buffers and general service times", *Computers & Operations Research*, vol. 35(8), pp. 2520-2536, 2008.
- [35] R.R. Lazaro, C.J. Luis Perez, "Dynamic analysis of an automobile assembly line considering starving and blocking", *Robotics and Computer-Integrated Manufacturing*, vol. 25(2), pp. 271-279, 2009.
- [36] J. Jackman, E. Johnson, "The Role of Queueing Network Models in Performance Evaluation of Manufacturing Systems", *The Journal of the Operational Research Society*, vol. 44(8), pp. 797-807, 1993.
- [37] H.T. Papadopoulos, C. Heavey, "Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines", *European Journal of Operational Research*, vol. 92, pp. 1-27, 1996.
- [38] M.K. Govil, M.C. Fu, "Queueing theory in manufacturing A survey", *Journal of Manufacturing Systems*, vol. 18(3), pp. 214-240, 1999.
- [39] J.H. Patterson, J.J. Albracht, "Assembly-Line Balancing: Zero-One Programming with Fibonacci Search", *Operations Research*, vol. 23(1), pp. 166-172, 1975.
- [40] R. Klein, A. Scholl, "Maximizing the Production Rate in Simple Assembly Line Balancing-A branch and bound procedure", *European Journal of Operational Research*, vol. 91, pp. 367-385, 1996.
- [41] G. Bolch, S. Greiner, H. Meer, K.S. Trivedi, *Queueing Networks and Markov Chains Modeling and Performance Evaluation with Computer Science Applications*, John Wiley & Sons, New Jersey, 2006

## NOMENCLATURE

## Mathematical and CP Model

$x_{ik}$	decision variable of mathematical model. 1, if task $i$ is assign at the station; 0, other wise
$T$	the set of tasks
$\ T\ $	number of elements of set $T$
$W$	the set of workstations
$\ W\ $	number of elements of set $W$
$t_i$	time of task $i$
$\mu_i^t$	mean time of task $i$
$\sigma_i^t$	standard deviation of task $i$
$t(S_k)$	time of station $k$
$S_k$	the set of tasks assigned to the station $k$
$\mu_k^s$	mean time of station $k$
$\sigma_k^s$	standard deviation of station $k$
$C$	cycle time
$p$	production or output rate
$E_i$	Earliest station bound for task $i$
$L_i$	latest station bound for task $i$
$P_i^*$	the set of the predecessors of task $i$
$F_i^*$	the set of the followers of the task $i$
$Pr$	The set of precedence relations among tasks, for instance task $i$ precedes task $k$ .

## Queueing networks

$\rho_k$	utilization of station $k$
$ti$	traffic intensity
$\lambda$	arrival rate (system workload)
$e_k$	visit ratios of station $k$
$\mu_k$	service rate of station $k$
$c_A$	coefficient of variation of interarrival time
$c_B$	coefficient of variation of service time
$C_{Ak}^2$	squared coefficient of variation of interarrival time for station $k$
$C_{Bk}^2$	squared coefficient of variation of service time for station $k$
$p_{jk}$	routing probability
$\mu_k$	updated utilization of station $k$
$\bar{K}_k$	mean number of jobs of station $k$
$CV$	coefficient of variation