A new look at the bowl phenomenon

Pedro B. Castellucci[1] and Alysson M. Costa[2]

[1] *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo*

[2] *Department of Mathematics and Statistics, University of Melbourne*

pbc@icmc.usp.br, alysson.costa@unimelb.edu.au

Abstract

An interesting empirical result in the assembly line literature states that slightly unbalanced assembly lines (in the format of a bowl - with central stations less loaded than the external ones) present higher throughputs than perfectly balanced ones. This effect is known as the *bowl phenomenon*. In this study, we analyze the presence of this phenomenon in assembly lines with integer task times. For this purpose, we modify existing models for the *simple assembly line balancing problem* and *assembly line worker assignment and balancing problem* in order to generate configurations exhibiting the desired format. These configurations are implemented in a stochastic simulation model, which is run for a large set of recently introduced instances. The obtained results are analyzed and the findings obtained here indicate, for the first time, the existence of the bowl phenomenon in a large set of configurations (corresponding to the wide range of instances tested) and also the possibility of reproducing such phenomenon in lines with a heterogeneous workforce.

Keywords: Assembly lines, simulation, disabled workers, heterogeneous workers, bowl phenomenon.

1 Introduction

Assembly lines are productive systems specially useful for large scale standardized manufacturing. The main rationale behind such systems is labour division. This is done by partitioning the tasks to be executed among a number of workers or workstations (Scholl 1999). In each workstation, a subset of the tasks is executed. The classical assembly line balancing problem is known as SALBP (*simple assembly line balancing problem*) and considers, among other hypotheses, that all workers/workstations are equally efficient. A large amount of research has been conducted on the SALBP and includes both classical and recent surveys and research articles (Salveson 1955, Tonge 1961, Baybars 1986, Ghosh & Gagnon 1989, Scholl 1999, Scholl & Becker 2006, Boysen & Fliedner 2008, Scholl et al. 2010).

SALBP's hypotheses are rarely valid in practical contexts. This fact has motivated the study of a large number of its variants (Boysen et al. 2007, Boysen et al. 2008, Battaïa & Dolgui 2012). In particular, this research addresses, besides SALBP, the case of assembly lines in sheltered work centers for disabled. In this case, due to workers heterogeneity, task execution times depend on the worker they are assigned to. This gives origin to a problem in which, besides assigning tasks to workstations, there is also the need to assign workers to workstations. This situation has been first modelled by Miralles et al. (2007) and named ALWABP (*assembly line worker assignment and balancing problem*).

Both SALBP and ALWABP assume deterministic execution times for each task and, hence, the line's throughput is dictated by the most loaded workstation. In practice, however, task execution times are usually stochastic. This may occur due to small differences among parts, changes in worker's behavior, technical issues and many other random events. The randomness in execution times may lead to interesting effects such as the well known bowl phenomenon, in which slightly unbalanced lines in the format of a bowl (i.e., with central workstations less loaded than external ones), or with central workstations with more stable loads (less variable execution times), have higher productivities (Hillier & Boling 1966).

The theory behind the Bowl phenomenon is not yet fully understood. Nevertheless, the effect may lead to adjustments in optimization methods (heuristic and exact) in order to enhance throughput of assembly lines in practical contexts. Thereunto, simulation models are helpful, because they allow the modelling of execution times with higher precision.

In this study, an assembly line simulation model for productivity analysis is proposed. In particular, the interest is focused on recently proposed instances for the SALBP (Otto et al. 2013) and marginally on instances of the ALWABP. The objective is twofold:

Our first goal is to verify the possibility of generating solutions for the ALWABP that can take advantage of the bowl phenomenon. For this purpose, a mixed-integer programming model capable of generating solutions with unbalanced bowl-shaped profiles (with greater or lesser slopes) is proposed. These solutions are used as parameters in the simulation model, also developed in this work, and the results are discussed.

The second and more general goal is to evaluate the own existence of the bowl phenomenon. Indeed, the literature has always dealt with a reduced number of configurations (obtained from existing classical case studies). Here, we use the newly proposed instances of (Otto et al. 2013) and effect a much larger computational study. Also, since the parameters for the simulation model come from the proposed mixed-integer programming model, the indivisibility of tasks is considered for evaluation which is not usually reported in the literature.

The remainder of this article is organized as follows. In the following section, SALBP and ALWABP are discussed and their models are presented. Also in this section, these model are modified in order to generate solutions with unbalanced bowl-shaped profiles. Then, Section 3 focus on the proposed simulation model development while Section 4 presents the computational experiments and results. General conclusions end this paper in Section 5.

2 SALBP and ALWABP

The most idealized problem of assembly line balancing is the *simple assembly line balancing problem*, SALBP. In this problem, a paced assembly line with a fixed cycle time is considered. Task execution times are deterministic, task allocations are only constrained by precedence relations, lines are serial and all the workstations are equally equipped (Scholl 1999).

In order to model the SALBP, let N be a set of tasks. Tasks $i \in N$ must respect a partial ordering and the execution time of each task $i \in N$ is t_i. We use the notation $i \prec j$ to indicate that task i must be completed before task j is initiated. Defining binary variables x_{ik} (equal to 1 if, and only if, task i is executed at station k). Let C be the cycle time and S the set of workstation, then a SALBP model can be written as:

$$\text{Min } C \tag{1}$$

subjected to:

$$\sum_{k \in S} x_{ik} = 1, \qquad \forall i \in N, \tag{2}$$

$$\sum_{i \in N} t_i \cdot x_{ik} \leq C, \qquad \forall k \in S, \tag{3}$$

$$\sum_{k \in S} k \cdot x_{ik} \leq \sum_{k \in S} k \cdot x_{jk}, \qquad \forall i, j \in N, i \prec j, \tag{4}$$

$$x_{ik} \in \{0, 1\}, \qquad \forall i \in N, \forall k \in S. \tag{5}$$

The goal of the model (1 - 5) is to provide an allocation of tasks to workstation so that the cycle time is minimized. Constraints (2) force every task to be executed, while constraints (3) define the assembly line's cycle time and (4) ensure the respect of the partial ordering of tasks. Model (1 - 5) assumes all SALBP hypotheses.

In the ALWABP, the hypothesis of equally equipped workstations are relaxed by considering different workers in each station. Miralles et al. (2007) have proposed the ALWABP motivated by the situation found in sheltered work centers for the disabled, in which assembly lines are operated by workers with disabilities. This problem has fomented an intense amount of research both on the original problem (Blum & Miralles 2011, Moreira et al. 2012, Mutlu et al. 2013, Vilà & Pereira 2014, Borba & Ritt 2014) and on its variants (Costa & Miralles 2009, Araújo et al. 2012, Moreira & Costa 2013, Araújo et al. 2014).

In order to model the ALWABP, let N be a set of tasks. The execution time of each task $i \in N$ depends on the worker it is assigned to and is assumed deterministic. Indeed, p_{wi} is the executing time of task i by worker w. Tasks $i \in N$ still must respect a partial ordering. Let W represent all workers available and S represent the set of workstations. Defining binary variables y_{sw} (equal to 1 if, and only if, worker $w \in W$ is allocated to station $s \in S$) and variables x_{swi} (equal to 1 if, and only if, task $i \in N$ is executed at station $s \in S$ by worker $w \in W$) and a variable C representing the cycle time (execution time of the most loaded workstation), the ALWABP can be written as:

$$\text{Min } C \tag{6}$$

subjected to:

$$\sum_{s \in S} \sum_{w \in W} x_{swi} = 1, \qquad \forall i \in N, \tag{7}$$

$$\sum_{s \in S} y_{sw} = 1, \qquad \forall w \in W, \tag{8}$$

$$\sum_{w \in W} y_{sw} = 1, \qquad \forall s \in S, \tag{9}$$

$$\sum_{s \in S} \sum_{w \in W} s \cdot x_{swi} \leq \sum_{s \in S} \sum_{w \in W} s \cdot x_{swj}, \qquad \forall i, j \in N, i \lessdot j, \tag{10}$$

$$\sum_{i \in N} \sum_{w \in W} p_{wi} \cdot x_{swi} \leq C, \qquad \forall s \in S, \tag{11}$$

$$x_{swi} \leq y_{sw}, \qquad \forall s \in S, \forall w \in W, \forall i \in N, \tag{12}$$

$$y_{sw} \in \{0, 1\}, \qquad \forall s \in S, \forall w \in W, \tag{13}$$

$$x_{swi} \in \{0, 1\}, \qquad \forall s \in S, \forall w \in W, \forall i \in N. \tag{14}$$

The the goal of model (6)-(14) is to obtain worker and task assignments to the workstations so that the cycle time is minimized. Constraints (7) force every task to be executed, while (8) and (9) guarantee that each worker is assigned to a workstation and each workstation receives a single worker, respectively. The task execution partial ordering is respected due to constraints (10), while (11) define the line's cycle time. Finally, constraints (12) ensure that a task is executed by a worker in a given workstation only if the worker is assigned to the workstation.

2.1 The bowl phenomenon and ALBPs

Manly in non-automatized assembly lines, where people, not machines, are responsible for doing the tasks, considering execution times to be deterministic might be a strong assumption. In these cases, more precise production rates can be modelled if one considers the statistical distribution dictating the task times. We consider the case of unpaced assembly lines where the pace dictated by the cycle time is relaxed. In such unpaced lines, workstations may have to wait for the following station to finish its work in order to pass the product along the line, in this case, the station waiting is said to be blocked. On the other hand, when a station is free and has to wait for the product from the previous station, it is said to be starved. In the literature, it is common to assume that the first station can never be starved and the last one can never be blocked.

According to Smunt & Perkins (1985), the first studies of unpaced lines date from 1962 when simulations evaluated initial buffer configurations (Barten 1962). Usually, analytical models of these kind of systems are complex and are only able to deal with no more than four workstations. Even then, they require simplifying hypotheses like assuming that the task times are exponentially distributed. More practical scenarios (with more stations and appropriated distributions) have been studied using simulation based approaches. Hillier & Boling (1966), after some claims that the optimum allocation would be to alternate fast and slow stations (Patterson 1964), showed that if the operation times (task times) were following an exponential distribution, then the assembly line throughput could be enhanced by introducing specific small load unbalances. This effect became known as the *bowl phenomenon*, since the graphic of total time of each workstation along the line has the shape of a bowl, with times increasing from central station(s). Figure 1 shows an example of such configuration in a line with eight workstations.

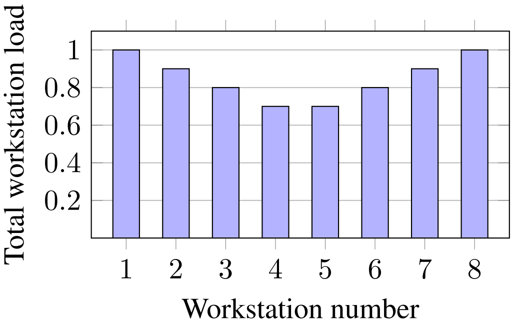

Figure 1: example of a bowl profile. Stations in the center are less loaded than the peripheral ones.

Results presented by Hillier & Boling (1966) considered lines with at most four stations with buffer capacity from one to four units. However, using the exponential distribution does not allow one to tell if the enhancement in the line's throughput is due to unbalancing the mean or the variability of the task times, since both parameters are the same for that distribution. The investigation of this gap, also considering exponential distribution and assuming some deterministic variables, showed the optimum strategy to be a combination of both unbalances (Rao 1976). The use of *Erlang* distribution also produces a bowl load profile as optimum solution and Hillier & Boling (1979) concluded that the effect of the bowl configuration became larger as the number of stations increased. By this time, the first experiments supporting the bowl phenomenon as proposed by Hillier & Boling (1966) were performed.

At one moment, conflicting results related to the production rate and the idle time of the workstations were presented. Some of them argued that an increasing workload profile would be better than a bowl shaped workload. In this sense, (Kottas & Lau 1981) proposed an approach for studying and monitoring the transient behavior of an assembly line and showed that the conflicting results were, in fact, underestimated transient properties read as steady-state behavior.

Another controversy happened when Smunt & Perkins (1985), using some results from (Dudley 1963), argued that the assembly lines considered in previous researches, with few stations and high variability, were not of practical interest. They then presented results for more realistic scenarios that supported a degradation of the line's throughput if a bowl workload profile was used. Nevertheless, (Karwan & Philipoom 1989) and (So 1989), independently argued that (Smunt & Perkins 1985) had some flaws in their experimental design. All the group of researchers agreed that more realistic scenarios should be explored.

Hillier & So (1996) examined the robustness of the bowl phenomenon, namely, the effect of incorrect estimate the optimum bowl allocation of workload. They concluded that even errors of 50% in the optimum allocation are still better than the perfectly balanced load. Moreover, a 10% error in the workstations' load produces better results than the balanced case. Therefore, the authors concluded that is better to aim at a bowl profile than at a balanced one.

Hillier & Hillier (2006) optimized workload and buffers capacity simultaneously and the results showed that, if the buffers are small, their optimum allocation is balanced and the workload should follow a bowl pattern. Considering one wants to minimize total time, idle time or mean time of a product in the system, (Das, Garcia-Diaz, MacDonald & Ghoshal 2010) and (Das, Sanchez-Rivas, Garcia-Diaz & MacDonald 2010), using simulation, explored bowl, inverted bowl and crescent workload profiles. Also, in (Das et al. 2012) there is a comparison between crescent and decrescent load profile. Furthermore, optimizing buffer capacity between stations and work allocation using exact and heuristic methods was studied by Hillier (2013).

It is important to state that none of the articles cited above considered the indivisibility of the tasks, even though this is a strong requirement in most practical situations. In other words, all authors have assumed that a task with a specific size is always available to fulfill the required workload of a station. This flaw has been already pointed out by Tempelmeier (2003) but, to the best of our knowledge, no investigation was conducted to settle this question. This is one of the main research gaps explored in the remainder of this article.

In general, the innovation presented in this paper is twofold. First, we use a simulation model to try to conclude on the existence of the bowl effect in larger lines and with more diversified configurations. For this, we use the set of instances recently proposed by Otto et al. (2013) while considering the integrality of the tasks. We also propose a simple MIP methodology to generate bowl format configurations for lines with homogeneous and heterogeneous workforce, either by introducing load unbalances or by analysing the effect of unbalances in deviation. These two cases are described in the following.

2.1.1 Load unbalance

Constraints (3) and (11) are appropriate for the deterministic scenario because they induce a uniform load distribution among all workstations, hence minimizing the cycle time C. In order to obtain bowl-like solutions, these constraints are modified in the corresponding models. For SALBP the new constraints are:

$$\sum_{i\in N} t_i \cdot x_{is} \leq \alpha_s C, \qquad \forall s \in S, \tag{15}$$

and for ALWABP, the cycle time constraints are replaced by:

$$\sum_{i\in N} \sum_{w\in W} p_{wi} \cdot x_{swi} \leq \alpha_s C, \qquad \forall s \in S, \tag{16}$$

in both cases, α_s are parameters whose values respect the following equations:

$$\alpha_1 > 0, \tag{17}$$

$$\alpha_s = \alpha_{|S|-s+1}, \qquad s = 1 \ldots \lfloor |S|/2 \rfloor. \tag{18}$$

$$\alpha_s = \beta \alpha_{s-1}, \qquad s = 2 \ldots \lfloor |S|/2 \rfloor - 1. \tag{19}$$

Constraints (17) guarantee positive loads for workstations while constraints (18) induce symmetry in the load configuration. Finally, constraints (19) define the bowl slope level which is controlled by parameter β, $0 < \beta \leq 1$. Note that if $\beta = 1$ the proposed models are equivalent to the original ones. Figure 2 shows examples of resulting profiles for some values of α and β.

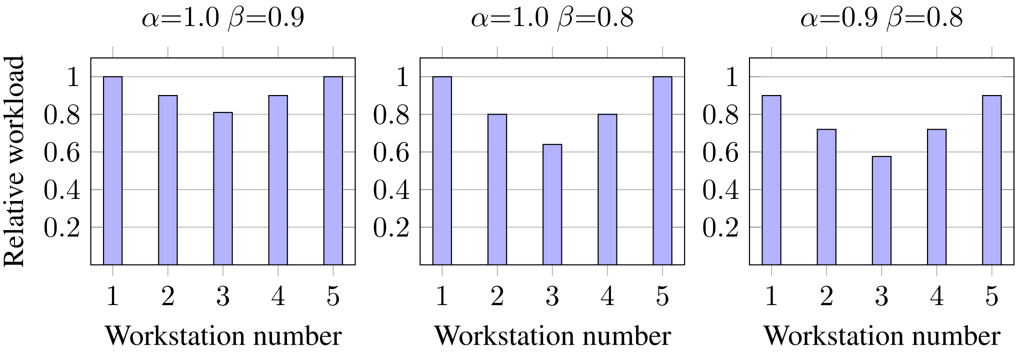

Figure 2: load configurations for different values of parameters α and β.

2.1.2 Deviation unbalance

Other characteristic that may induce a bowl effect is related to task times deviations. Let $c_s = \frac{\sigma_s}{\mu_s}$ indicate the level of variability in a given station execution's time $s \in S$. A bowl profile is obtained by maintaining the station loads fixed and reducing the task deviation times in central stations. The deviation was computed considering solutions without restrictions (15) or (16) and by artificially adjusting task deviation times from the base case.

The base case considers $\frac{\sigma}{\mu} = 0.1$, $s = 1, ..., S$, while bowl profiles are obtained with $c_s = \theta c_{s-1}$, $s = 1, ..., \lfloor |S|/2 \rfloor$ and $c_s = c_{|S|-s+1}$, $s = 1, ..., \lfloor |S|/2 \rfloor$, where $0 < \theta \leq 1$ is a parameter related to the depth of the bowl. Note that these equations are similar to the equations to generate bowl solutions for the mean case. Hence, the bowl profiles for mean and deviation have the same shapes.

3 Simulation Model

Simulation is a technique for performance and reaction analysis of a system. It can be useful in planning stages or when the system is already in operation if careful validation on practical contexts is desired (Leal et al. 2011). Most benefits of simulation appear in contexts in which the system under study has many features that can not be modelled properly with the use of other analytical techniques due, for example, to its complexity.

When building a simulation model, it is important to define the inputs and outputs so the results may be analyzed. This analysis depends on the nature of the model. If the model is deterministic, for example, only one run is needed for each input one may want to analyze; on the other hand, in a stochastic model, it is possible to use a Monte Carlo method: the essence of it is to use stochastic distributions, each one representing a process in the model. Then, a statistical sampling is used to summarize the result for each input. In other words, simulation models can be used to analyze processes having a degree of uncertainty (randomness) in their variables (Kroese 2008).

In this work, the simulation model built in order to evaluate assembly lines has the number of products which left the line as a state-variable, it is of continuous time, discrete state, stochastic, linear, dynamic, open and unstable (Ross 2006). Also, it is a next-event analysis model whose events are triggered when a station finishes its work.

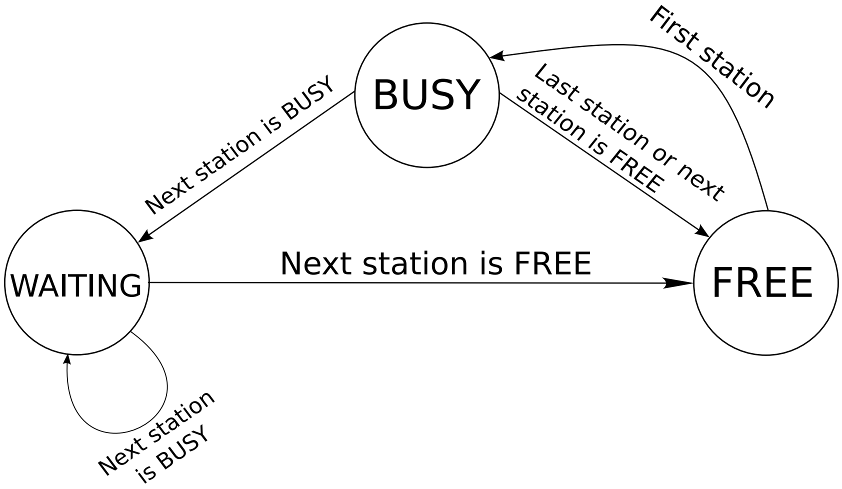

Figure 3: simplified machine state for the workstations. When a workstation is in BUSY state it is performing some work, in WAITING state it is blocked and in FREE it is starved.

Actually, besides the starting event, there is just one more type of event that happens during a simulation: when all the work in a station has been done, a previously scheduled event is triggered and the product is delivered to the next station or the current station gets blocked, following the diagram in figure 3. There are three possible workstation states. Any transition can only happen when the referred event is triggered. Namely, whenever the event happens, if the current station is in BUSY state it checks the following station, if it is in FREE state, the finished piece of product is delivered to it and the state of the current station changes to FREE, which means it can receive another piece to work on, if it is the first station it returns to BUSY state (there is always work available for the first station); otherwise, if the next station is in BUSY state when the current finishes its work, the current station changes its state to WAITING, blocking itself until the next station is FREE. If the current BUSY station is the last station of the line it changes its state to FREE, since there is no need to check for the next station's state. There is also one action, not shown in figure 3, that takes place when the transition from BUSY to FREE happens: the current station changes the state of the next station to BUSY, translating the fact that a new piece of product arrived in the following station. Hence, the event triggered when a workstation finishes its work controls the flow of products in the line.

The scheduling of an event occurs when the station changes its state from FREE to BUSY. In the model, this transition starts a sampling procedure of the station tasks execution times and the event associated with the completion of the work in that station is scheduled to the proper time. The handling of scheduled events is performed according to the diagram shown in figure 4 which has been adapted from (Robinson 2004).

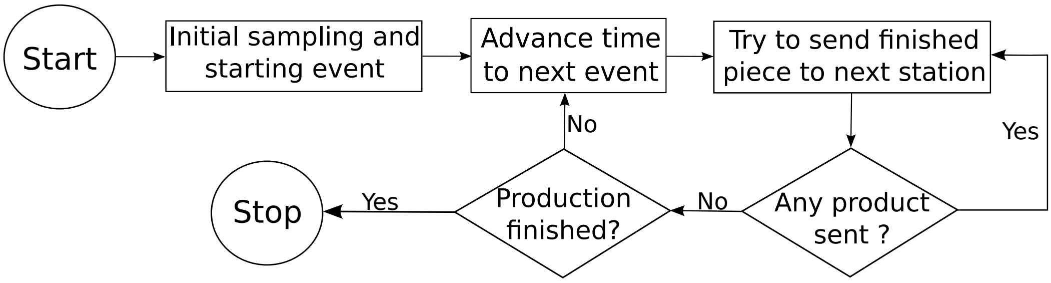
Start
Initial sampling and starting event
Advance time to next event
Try to send finished piece to next station
No
Yes
Stop
Yes
Production finished?
No
Any product sent ?

Figure 4: flowchart of a simulation. The simulation advances its time to the next scheduled event and process the corresponding events. If none of them can be processed it advances time again until at least one station is not blocked. Simulation stops when a specified number of completed products is reached.

When a simulation starts, every station state is set to FREE, but the first one whose state is set to BUSY, hence an event corresponding to finishing the first job in the first station is scheduled. Then, time is advanced to the next event which is processed according to diagram in figure 3. If no station can pass the product to the next station and the desired number of products is not yet reached, the simulation advances to the next scheduled event until a workstation becomes able to deliver its work to the following one. The simulation stops when the specified production is reached.

The allocation of tasks to workstations and, in ALWABP case, workers to workstations is given by the respective deterministic mathematical model whose objective is to minimize the assembly line's cycle time. Since, in the simulation model of this work, every task is supposed to be performed in a time following a Normal distribution, two parameters are needed in order to fully specify the task times to the simulation model: mean μ and standard deviation σ. The mean is assumed to be the deterministic time of the task and the standard deviation is $\sigma = \frac{\mu}{10}$ for load unbalance simulations and is given according to subsection 2.1.2 for deviation unbalance simulations.

In summary, we propose a stochastic simulation model using next-event analysis and deterministic mathematical programming models to study the bowl phenomenon considering integer tasks and instances proposed by (Otto et al. 2013) and (Chaves et al. 2009), respectively for SALBP and ALWABP. The results obtained in the experiments are detailed in the next section.

4 Computational results

The computational results were obtained with a series of stochastic simulations using a Monte Carlo method. Each instance was run 300 times in order to obtain statistically relevant results, and each simulation run until a production of 150 items was reached. The time to accomplish this production is, therefore, the result to be analyzed. For each run the results obtained for the first 50 items were discarded in order to account for a possible transient state of the line. This warm-up time was obtained using Welch's method as described in (Mahajan & Ingalls 2004). All the task times were assumed to follow a Normal distribution, $N(\mu, \sigma^2)$, and separated experiments were performed to evaluate the mean and standard deviation unbalances on total workstation load. In order to analyze mean unbalances, experiments with $\alpha_1 = 1$ and $\beta \in B = \{1.00, 0.99, 0.98, 0.97, 0.96, 0.95, 0.94\}$ were performed (as defined in constraints (17) and (19)); for deviation unbalances the experiments considered $\theta \in \{\theta_i = \beta_i | \forall \beta_i \in B\}$.

Both SALBP and ALWABP cases were considered independently. The benchmark problems for the SALBP simulations are a subset of the dataset proposed by (Otto et al. 2013). We use the instances for which the known upper bound on the number of stations is seven or less. The benchmark for the ALWABP simulations are the instances of type Heskia and Roszieg from (Chaves et al. 2009). Both, SALBP and ALWABP instances, were solved to optimality. The results' analysis, performed in the following, focus on the SALBP problems since their instances are more representative than their ALWABP counterparts. Tables 1 through 6 summarize results for the SALBP simulations. Each table presents the total number of instances in the corresponding categories, the number of instances in which the bowl phenomenon could be observed (i.e., the number of instances in which the bowl configuration was statistically superior to their balanced counterparts) and their mean β (or θ) for mean (or standard deviation) unbalances. The categories are related to different precedence graph structures and how the task times are distributed: graphs may have *bottleneck* tasks and *chain* tasks and, according to the relevance of one or other type of tasks, Otto et al. (2013) classified the graph structure as *BN* (for *bottleneck*), *CH* (for *chain*) or *MIXED* when there is no prominent characteristic. Task time's distributions may be bimodal or with peak at small tasks (bottom peak) in the considered instances. Also a statistical t-test with 0.05 threshold in the p-value was used to ensure statistical difference from the best bowl solution to the balanced one.

Table 1: mean unbalance results for SALBP instances grouped by number of workstations.

	Workstations				
	3	**4**	**5**	**6**	**7**
Instances	163	38	89	50	6
Bowl	0.5399	0.8421	0.7416	0.7800	0.8333
β_{mean}	0.9781	0.9781	0.9788	0.9777	0.9900

Tables 1, 2 and 3 contain results related to the mean unbalance simulations. Table 1 presents the results grouped by number of stations. Most of the instances (163) considered have three workstations in the optimal solution and in 53.99% the best bowl solution was statistically different from the balanced solution leading to $\beta_{mean} = 0.9781$. The deepest mean bowl was also found in instances with four workstations which was, as well, the case that present greater relative number of observable bowl phenomenons (84.21%).

Table 2 summarizes the results grouped by graph structures and task time distributions. In the graph structure group, the deepest mean bowl was found in *MIXED* and the *CH* class presented the greater relative number of bowl-shaped solutions performing better than balanced one. The bottom peak task time distribution presented the deepest bowl and bimodal instances shown a greater number of solutions for which the bowl configuration was more productive. Also, it can be seen that *CH* graphs

Table 2: mean unbalance results for SALBP instances grouped by graph structure and task time distribution.

	Graph structure			**Task time distribution**	
	BN	**CH**	**MIXED**	**Bimodal**	**Bottom peak**
Instances	98	99	149	174	172
Bowl	0.6531	0.6970	0.6510	0.7644	0.5640
β_{mean}	0.9820	0.9781	0.9764	0.9788	0.9780

and bimodal task distribution seem to favor the occurrence of the bowl phenomenon.

Table 3: mean unbalance results for SALBP instances combining number of workstation with graph structure or task time distribution.

		Graph structure			**Task time distribution**	
Workstations		**BN**	**CH**	**MIXED**	**Bimodal**	**Bottom peak**
3	**Instances**	47	46	70	0	163
	Bowl	0.5319	0.5435	0.5429	-	0.5399
	β_{mean}	0.9832	0.9772	0.9753	-	0.9781
4	**Instances**	11	12	15	29	9
	Bowl	0.6364	1.0000	0.8667	0.7931	1.0000
	β_{mean}	0.9800	0.9792	0.9762	0.9783	0.9778
5	**Instances**	28	24	37	89	0
	Bowl	0.7857	0.7083	0.7297	0.7416	-
	β_{mean}	0.9818	0.9770	0.9774	0.9788	-
6	**Instances**	10	15	25	50	0
	Bowl	0.8000	0.8667	0.7200	0.7800	-
	β_{mean}	0.9788	0.9785	0.9767	0.9777	-
7	**Instances**	2	2	2	6	0
	Bowl	1.0000	1.0000	0.5000	0.8333	-
	β_{mean}	0.9900	0.9900	0.9900	0.9900	-

Table 3 presents the results considering both number of stations and graph structure and number of stations and task time distribution. In some of these scenarios there was no available instance: three workstations with bimodal task time distribution and five workstations with bottom peak time distribution, for example. Therefore, there are table cells marked with ”-” where the corresponding result is not applicable. Again, for three and four workstation, it is possible to see *CH* graphs favouring the bowl phenomenon, nevertheless, this is not true for five workstations. It is hard to draw conclusions for instances with seven workstations since there are fewer instances in this scenario. Likewise, for task time distribution case the instances are not well suited for comparing bimodal and bottom peak distributions.

Tables 4, 5 and 6 present results regarding bowl profiles in $c_s = \frac{\sigma_s}{\mu_s}$. Table 4 shows results grouped

Table 4: deviation unbalance results for SALBP instances grouped by number of workstations.

	Workstations				
	3	**4**	**5**	**6**	**7**
Instances	163	38	89	50	6
Bowl	0.5337	0.9737	1.0000	1.0000	1.0000
θ_{mean}	0.9480	0.9462	0.9402	0.9410	0.9400

by number of workstations. Instances with five, six and seven workstations achieved 100% of *bowl phenomenon occurrence* (i.e., in all cases the lines with the obtained bowl configurations were more productive than their balanced counterparts). Also, the mean depth of the bowl for seven workstation reached the maximum value among all the experiments conducted.

Table 5: deviation unbalance results for SALBP instances grouped by graph structure and task time distribution.

	Graph structure			**Task time distribution**	
	BN	**CH**	**MIXED**	**Bimodal**	**Bottom peak**
Instances	98	99	149	174	175
Bowl	0.7653	0.8080	0.7651	0.9943	0.5581
θ_{mean}	0.9437	0.9446	0.9445	0.9424	0.9478

Table 5 presents results grouped by graph structures and task time distributions. In the graph structure group, the deepest mean bowl was found in the *bottleneck* class and the class that presented the greater relative number of bowl-shaped solution performing better than balanced solutions was the *chain* one. The bimodal task time distribution presented deepest and greater number of better bowl solutions. Four out of five classes presented more than 75% of occurrence of the bowl phenomenon, reaching 99.43% in the bimodal class.

Table 6 summarizes results combining number of workstations and graph structure and number of workstations and task time distribution for deviation analysis. For instances with three workstations the graph structure *chain* seems to favor the effect. For instances with more than three stations the phenomenon could be observed in most of the problems and the maximum bowl depth considered in the experiments was in a number of scenarios.

Regarding the ALWABP, the results were analyzed in a much simpler way since the available instances are not as representative as those for the SALBP. Table 7 shows the results. They are grouped by variance among workers: low variance means that task execution time varies less among workers than in the high variance scenario (Chaves et al. 2009). The table presents the fraction of instances where the bowl phenomenon could be observed and β_{mean} for mean unbalance or θ_{mean} for deviation unbalance. For each class (low and high) 80 instances were analyzed. It is possible to see that β_{mean} had smaller values here than in the SALBP case. Furthermore, in the deviation case the bowl phenomenon was observed in fewer instances if compared to SALBP. Nevertheless, the bowl phenomenon could be observed in more than 50% of the cases for every class (and case) considered.

As a final note, we remark that the SALBP and ALWABP results show that the bowl phenomenon

Table 6: deviation unbalance results for SALBP instances combining number of workstation with graph structure or task time distribution

		Graph structure			**Task time distribution**	
Workstations		**BN**	**CH**	**MIXED**	**Bimodal**	**Bottom peak**
3	**Instances**	47	46	70	0	163
	Bowl	0.5319	0.5869	0.5000	-	0.5337
	θ_{mean}	0.9472	0.9478	0.9489	-	0.9480
4	**Instances**	11	12	15	29	9
	Bowl	0.9090	1.0000	1.0000	0.9655	1.0000
	θ_{mean}	0.9430	0.9492	0.9460	0.9464	0.9456
5	**Instances**	28	24	37	89	0
	Bowl	1.0000	1.0000	1.0000	1.0000	-
	θ_{mean}	0.9421	0.9413	0.9424	0.9420	-
6	**Instances**	10	15	25	50	0
	Bowl	1.0000	1.0000	1.0000	1.0000	-
	θ_{mean}	0.9410	0.9413	0.9408	0.9410	-
7	**Instances**	2	2	2	6	0
	Bowl	1.0000	1.0000	1.0000	1.0000	-
	θ_{mean}	0.9400	0.9400	0.9400	0.9400	-

Table 7: mean and deviation unbalances for ALWABP instances.

	Low		**High**	
	mean	**deviation**	**mean**	**deviation**
Bowl	0.6125	0.8625	0.7500	0.8000
β_{mean} or θ_{mean}	0.9684	0.9423	0.9670	0.9430

regarding variability occurred in more instances than the effect regarding mean. This may have happened because there is no integrality constraint in the standard deviation whereas, for the mean case, the proposed MIP formulation considers the integrality of the tasks.

5 Conclusions

This study introduced mixed-integer mathematical models to obtain slightly unbalanced load profiles for the simple assembly line balancing problem and the assembly line worker assignment balancing problem. These profiles were used in a simulation model to verify that the efficiency of the lines could be improved if they were slightly unbalanced. This was done using recently proposed benchmark problems, which are more representative than those used in previous bowl phenomenon studies, favoring the generalization of the conclusions obtained. These conclusions indicate that assembly lines can indeed benefit from bowl shaped configurations even if the more realistic scenario with indivisible tasks is considered.

6 Acknowledgments

This work was developed with financial support from CNPq, CAPES/DGU (258-12) and FAPESP.

References

Araújo, F. F. B., Costa, A. M. & Miralles, C. (2012). Two extensions for the assembly line worker assignment and balancing problem: parallel stations and collaborative approach, *International Journal of Production Economics* **140**: 483–495.

Araújo, F. F. B., Costa, A. M. & Miralles, C. (2014). Balancing parallel assembly lines with disabled workers, *European Journal of Industrial Engineering (In press)* .

Barten, K. (1962). A queuing simulator for determining optimum invetory levels in a sequential process, *Journal of Industrial Engineering* **14**: 703–710.

Battaïa, O. & Dolgui, A. (2012). A taxonomy of line balancing problems and their solution approaches, *International Journal of Production Economics* .

Baybars, I. (1986). A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem, *Management Science* **32**: 909–932.

Blum, C. & Miralles, C. (2011). On solving the assembly line worker assignment and balancing problem via beam search, *Computers & Operations Research* **38**: 328–339.

Borba, L. & Ritt, M. (2014). A heuristic and a branch-and-bound algorithm for the Assembly Line Worker Assignment and Balancing Problem, *Computers & Operations Research* **45**: 87–96.

Boysen, N. & Fliedner, M. (2008). A versatile algorithm for assembly line balancing, *European Journal of Operational Research* **184**: 39–56.

Boysen, N., Fliedner, M. & Scholl, A. (2007). A classification of assembly line balancing problems, *European Journal of Operational Research* **183**: 674–693.

Boysen, N., Fliedner, M. & Scholl, A. (2008). Assembly line balancing: Which model to use when?, *International Journal of Production Economics* **111**: 509–528.

Chaves, A. A., Lorena, L. A. N. & Miralles, C. (2009). Hybrid metaheuristic for the assembly line worker assignment and balancing problem, *Lecture Notes on Computer Science* **5818**: 1–14.

Costa, A. M. & Miralles, C. (2009). Job rotation in assembly lines employing disabled workers, *International Journal of Production Economics* **120**: 625–632.

Das, B., Garcia-Diaz, A., MacDonald, C. A. & Ghoshal, K. K. (2010). A computer simulation approach to evaluating bowl versus inverted bowl assembly line arrangement with variable operation times, *The International Journal of Advanced Manufacturing Technology* pp. 15–24.

Das, B., Garcia-Diaz, A., MacDonald, C. A. & Ghoshal, K. K. (2012). Evaluation of alternative assembly line arrangements with stochastic operation times: A computer simulation approach, *Journal of Manufacturing Technology Management* **23**: 806–822.

Das, B., Sanchez-Rivas, J. M., Garcia-Diaz, A. & MacDonald, C. a. (2010). A computer simulation approach to evaluating assembly line balancing with variable operation times, *Journal of Manufacturing Technology Management* **21**: 872–887.

Dudley, N. (1963). Work-time distributions, *International Journal of Production Research* **2**: 137–144.

Ghosh, S. & Gagnon, R. J. (1989). A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems, *International Journal of Production Research* **27**: 637–670.

Hillier, F. S. & Boling, R. W. (1966). The effects of some design factors on the efficiency of production lines with variable operation times., *Journal of Industrial Engineering* **17**: 651–658.

Hillier, F. S. & Boling, R. W. (1979). On the Optimal Allocation of Work in Symmetrically Unbalanced Production Line Systems with Variable Operation Times, *Management Science* **25**: 721–728.

Hillier, F. S. & So, K. C. (1996). On the robustness of the bowl phenomenon, *European Journal Of Operational Research* **2217**: 496–515.

Hillier, M. (2013). Designing unpaced production lines to optimize throughput and work-in-process inventory, *IIE Transactions* **45**: 516–527.

Hillier, M. S. & Hillier, F. S. (2006). Simultaneous optimization of work and buffer space in unpaced production lines with random processing times, *IIE Transactions* **38**: 39–51.

Karwan, K. & Philipoom, P. (1989). A note on “Stochastic unpaced line design: Review and further experimental results”, *Journal of Operations Management* **8**: 48–54.

Kottas, J. F. & Lau, H.-S. (1981). Some problems with transient phenomena when simulating unpaced lines, *Journal of Operations Management* **1**: 155–164.

Kroese, R. Y. R. D. P. (2008). *Simulation and The Monte Carlo Method*, 2nd edn, John Wiley & Sons, Inc.

Leal, F., Costa, R. F. d. S., Montevechi, J. A. B., de Almeida, D. A. & Marins, F. A. S. (2011). A practical guide for operational validation of discrete simulation models, *Pesquisa Operacional* **31**: 57–77.

Mahajan, P. S. & Ingalls, R. G. (2004). Evaluation of methods used to detect warm-up period in steady state simulation, *Proceedings of the 2004 Winter Simulation Conference*, pp. 663–671.

Miralles, C., Garcia-Sabater, J. P., Andrés, C. & Cardos, M. (2007). Advantages of assembly lines in Sheltered Work Centres for Disabled. A case study, *International Journal of Production Economics* **110**: 187–197.

Moreira, M. C. O. & Costa, A. M. (2013). Hybrid Heuristics for planning job rotation in Assembly Lines with disabled workers, *International Journal of Production Economics* **141**: 552–560.

Moreira, M. C. O., Ritt, M., Costa, A. M. & Chaves, A. A. (2012). Simple heuristics for the assembly line worker assignment and balancing problem, *Journal of heuristics* **18**: 505–524.

Mutlu, O., Polat, O. & Supciller, A. A. (2013). An iterative genetic algorithm for the assembly line worker assignment and balancing problem of type-II, *Computers & Operations Research* **40**: 418–426.

Otto, A., Otto, C. & Scholl, A. (2013). SALBPGen - A systematic data generator for (simple) assembly line balancing, *European Journal of Operational Research* **228**: 33–45.

Patterson, R. (1964). Markov processes occurring in the theory of traffic flow through an N-Stage stochastic service system, *Journal of Industrial Engineering* **15**: 188–193.

Rao, N. P. (1976). A generalization of the 'bowl phenomenon' in series production systems, *International Journal of Production Research* **14**: 437–443.

Robinson, S. (2004). *Simulation - the practice of model development and use*, John Wiley & Sons.

Ross, S. M. (2006). *Simulation*, Elsevier.

Salveson, M. E. (1955). The assembly line balancing problem, *Journal of Industrial Engineering* **6**: 18–25.

Scholl, A. (1999). *Balancing and sequencing of assembly lines*, Physica-Verlag.

Scholl, A. & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, *European Journal of Operational Research* **168**: 666–693.

Scholl, A., Fliedner, M. & Boysen, N. (2010). Absalom: Balancing assembly lines with assignment restrictions, *European Journal of Operational Research* **200**: 688–701.

Smunt, T. L. & Perkins, W. C. (1985). Stochastic unpaced line design: Review and further experimental results, *Journal of Operations Management* **5**: 351–373.

So, K. C. (1989). On the efficiency of unbalancing production lines, *International Journal of Production Research* **27**: 717–729.

Tempelmeier, H. (2003). Practical considerations in the optimization of flow production systems, *International Journal of Production Research* **41**: 149–170.

Tonge, F. M. (1961). *A heuristic program for assembly line balancing.*, Prentice-Hall., Englewood Cliffs, NJ.

Vilà, M. & Pereira, J. (2014). A branch-and-bound algorithm for assembly line worker assignment and balancing problems, *Computers & Operations Research* **44**: 105–114.