Performance Analysis of a Real-Time Optimization Model for a Mixed Model Stochastic Assembly Line

Rangith Baby Kuriakose, Hermanus Jacobus Vermaak

Centre for Sustainable Cities, Central University of Technology, Free State, Bloemfontein, South Africa.

rkuriako@cut.ac.za; hvermaak@cut.ac.za

Abstract **- Mixed Model Stochastic Assembly lines are critical in the age of the 4th Industrial revolution. They are at the fore front of the shifting trend in manufacturing which sees moving from a Make-To-Stock approach to a Make-To-Order approach. Make-to-order systems are by nature stochastic, as their production cannot be pre-planned and complicated further by the fact that variants of the same product need to be manufactured on the same assembly line. This paper looks at the results of a real-time optimization model for a Mixed Model Stochastic Assembly Line. The model was tested on a water bottling plant that needs to produce 500 ml and 750 ml bottles based on client orders that are sourced from a cloud with external constraints such as the date of delivery and internal constraints like availability of clean water and number of bottles. The primary results show that in all instances, the optimization model was able to meet the delivery date set by the client. However, there are some instances where an anomaly was seen, especially when completing the first order. The aim of this paper is to do an in-depth analysis of the performance of the model to ascertain its veracity and robustness.**

Keywords – component, Modelling, Simulation, Stochastic, Perfomance Analysis

I. INTRODUCTION

The classification of Assembly Line Balancing [1], [2] (ALB) is done according to various parameters. Some of the most important parameters to be considered are product variety, task times, assembly line layout and level of automation. In terms of product variety, assembly lines can be split into single model [3], mixed model and multi model [4] assembly lines.

With respect to the processing time taken by a workstation or task time, the assembly line can be split into deterministic [5] and stochastic systems [6]. The organization of the assembly or the line layout, which can be Straight-type [7] or U-type [8] defines the third parameter. The classification parameters are rounded off by analyzing the level of automation in an assembly line. Here, an assembly line can be manual [9] or automated [10].

The study into the different Assembly Line Balancing Problems (ALBP) is centered around the afore mentioned parameters and has been widely researched [5], [11], [12]. However, research in Mixed Model Stochastic Straight type (MMSS) has been limited with focus mainly on minimizing workstations [13], maximizing cycle time, ensuring assembly line stability [14] and reducing positive drift [15].

This research aims to optimize a real-time MMSS assembly line. This will add to the existing knowledge on minimizing positive drift in MMSS assembly line systems. To achieve this goal, a Simulink model fed with real-time inputs though a Cloud Interface of an MMSS assembly line was designed [16] and tested with a MATLAB optimization model [17]. The model [18] is that of a water bottling plant which bottles 500ml and 750ml bottles of water on a straight type assembly line.

This paper is structed such that it initially looks at the limitations to existing research which necessitated this study. Secondly, the optimization model that was developed for this study is briefly described. Thirdly, the performance analysis of the model is done. Finally, a discussion is done on the obtained results with focus on what can be done in the future.

II. LIMITATIONS TO EXISTING RESEARCH

This specific study focusses on optimization of real time multi mixed make-to-order assembly line to reduce positive drift. Make-to-order systems are by nature stochastic as their production cannot be pre planned and are dependent on the order by a client, hence they can only be manufactured by using a multi model stochastic (MMSS) assembly line.

As seen from the summary in Fig 1, at the time of conducting this study, the amount of research done in Multi Model Stochastic Straight type assembly line problems was seemingly limited. The first of these studies is by McMullen and Taresewich [18]. They studied mixed model stochastic assembly line balancing problem with parallel workstations in which the objective is to design the assembly line such that the number of workstations or the cost is minimized for a given cycle time.

McMullen and Frazier [13] had earlier studied the balancing problem in which multiple product types scheduled in mixed-model fashion with stochastic task times and parallel workstations. The objectives here were the minimization of the total cost and maximization of the degree to which the desired cycle time is achieved [8].

Xu and Xiao [19] carried out a research in which an assembly line balancing problem with station lengths longer than the distance for which the conveyer moves within one cycle time is investigated in fuzzy environments, where operation times are assumed to be fuzzy variables. The objective is to minimize the positive drift time during the decision horizon.

Matanachai and Yano [20] also looked at the problem of positive drift on mixed model assembly lines in a multilevel production system. Their focus was mainly on the stability of the assembly line. Similar research by Tambe [21] included set-up time minimization and sequencing which was missing in the former study.

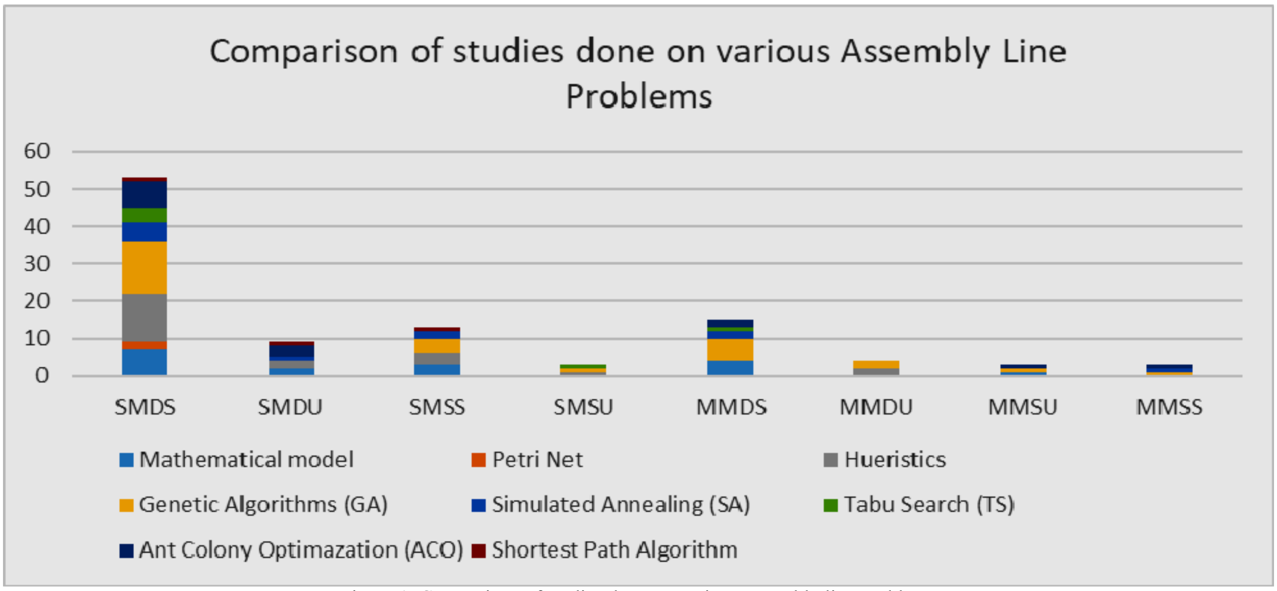


Figure 1. Comparison of studies done on various assembly line problems

III. PLANT OPTIMIZATION MODEL

The design of a customized plant model [16] with real time web enabled inputs [17] has been documented in previous journal articles. The development of an optimization model [18] with the aim of bottling 500 ml and 750 ml bottles as per user defined input requirements has also been documented.

A Real Time Optimization (RTO), such as the one described here, is formulated and solved using the following six steps.

1. Determine the process variables of interest
2. Define the objective function
3. Development of process models
4. Simplify the process model
5. Apply a suitable optimization technique
6. Check for sensitivity

These steps have been tailored to fit the water bottling plant as described below;

- Step 1: Determine process variables – The plant model has the following variables;
  - o Water stored in the tank in the Source subsystem
  - o Flow rate of water from the pump in the storage tank subsystem
  - o Initial number of 500ml bottles in the bottle manufacturing subsystem
  - o Initial number of 750ml bottles in the bottle manufacturing subsystem
  - o Expected date of delivery of customer orders

- Step 2: Defining the objective function – The objective function of the plant model is to reduce the production time for completing the customer orders. The hypothesis is that with optimization the production time can be significantly improved.

- Step 3: Development of process models - The model has considered two constraints;
  - o The water level of the tank- The water level in the tank should never go below 0% and should create an alert when below 25%.
  - o The number of 500ml and 750ml bottles available in storage- The number of bottles in storage should never go

below zero as this would result in the system crashing in a physical setup.

- Step 4: Simplify the process model – In order to simplify the process, the initial number of bottles is kept above zero. The pump flow rate from the storage tank subsystem acts as the handle which can be varied to meet the constraints.

- Step 5: Apply a suitable optimization technique – On analyzing the objective function, process variables and the constraints, it is noted that they exhibit a nonlinear relationship. Since the modelling is done on Simulink, the optimization can be done using MATLAB.

- Step 6: Check for sensitivity – The sensitivity check will be discussed in the results.

MATLAB optimization function *fmincon* is used in this study. The syntax for the *fmincon* function is as follows:

x=fmincon(fun,x0,A,b,Aeq,Beq,lb,ub,nonlcon,opt)

Where,

x0=starting point of minimization
fun=function to minimize
b and beq are vectors linear constraints
A and Aeq are matrices
lb=lower boundary of constraint
ub=upper boundary of constraint

In this specific optimization example, the syntax is as follows:

[xOpt, TTMOpt]=fmincon(fun,[0.1;0.1],[ ],[ ],[ ],[ ],[0;0],[1;1],funConstr,opt)

IV. PERFORMANCE ANALYSIS OF OPTMIZATION MODEL

This section of the paper initially analyses a section of the results which were seen to contradict the proper functioning of the optimization model. Secondly, the status of the constraints is examined to establish the cause of the contradictory results.

A total of 50 special instances were tested with varying values of customer inputs and required delivery. In all instances, a comparison between the optimized and non-optimized delivery date was done. It was seen that the required delivery date was met on all 50 instances. However, in four instances the non-optimized delivery date was quicker or on par with the optimized delivery date for the first customer order.

In the first of these instances, a random input of eleven customer orders were placed with the required delivery date on the same day and time. This is depicted in Fig 2. In the second instance, the customer orders were in input in opposite gradients with delivery date two hours apart from each other. This is depicted in Fig 3.

#	500 ml orders	750 ml orders	Same date and time delivery
1	100	100	2019/06/16 15:00
2	85	95	2019/06/16 15:00
3	120	120	2019/06/16 15:00
4	100	150	2019/06/16 15:00
5	120	150	2019/06/16 15:00
6	79	69	2019/06/16 15:00
7	25	30	2019/06/16 15:00
8	25	30	2019/06/16 15:00
9	45	50	2019/06/16 15:00
10	100	200	2019/06/16 15:00
11	150	250	2019/06/16 15:00

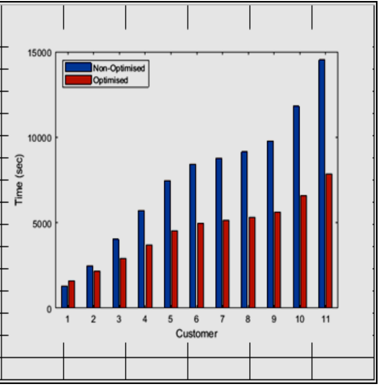


Figure 2. Comparison between the optimized and non-optimized date of delivery for customer orders with required delivery date on the same day

#	500 ml orders	750 ml orders	Opposite gradient, 2 hours apart
1	100	50	2019/06/16 08:00
2	90	60	2019/06/16 10:00
3	80	70	2019/06/16 12:00
4	70	80	2019/06/16 14:00
5	60	90	2019/06/16 16:00
6	50	100	2019/06/16 18:00
7	40	110	2019/06/16 20:00
8	30	120	2019/06/16 22:00
9	20	130	2019/06/17 00:00
10	10	140	2019/06/17 02:00
11	5	150	2019/06/17 04:00

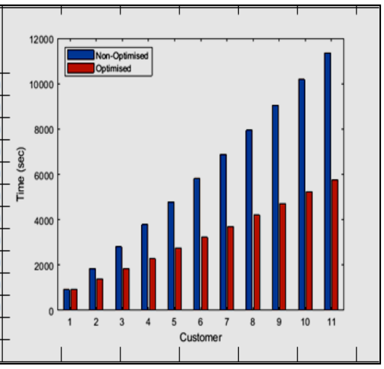


Figure 3. Comparison between the optimized and non-optimized date of delivery for customer orders with opposite gradients two hours apart

In the third instance, a set of eleven inputs which require same date delivery thirty minutes apart from each other is done. This is depicted in Fig 4. The last instance showcases twelve customer orders with delivery required on the same date at the same time. This is depicted in Fig 5.

#	500 ml orders	750 ml orders	Same delivery, 30 minutes apart
1	65	40	2019/06/16 08:00
2	95	95	2019/06/16 08:30
3	45	65	2019/06/16 09:00
4	30	70	2019/06/16 09:30
5	80	80	2019/06/16 10:00
6	70	35	2019/06/16 10:30
7	110	65	2019/06/16 11:00
8	90	95	2019/06/16 11:30
9	45	70	2019/06/16 12:00
10	25	85	2019/06/16 12:30
11	150	40	2019/06/16 13:00

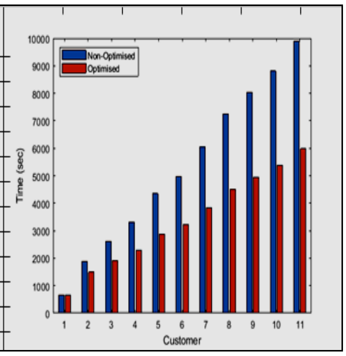


Figure 4. Comparison between the optimized and non-optimized date of delivery for customer orders with required delivery date thirty minutes apart

#	500 ml orders	750 ml orders	Same delivery with 12 customers
1	100	100	2019/06/16 15:00
2	85	95	2019/06/16 15:00
3	120	120	2019/06/16 15:00
4	100	150	2019/06/16 15:00
5	120	150	2019/06/16 15:00
6	79	69	2019/06/16 15:00
7	25	30	2019/06/16 15:00
8	25	30	2019/06/16 15:00
9	45	50	2019/06/16 15:00
10	100	200	2019/06/16 15:00
11	150	250	2019/06/16 15:00
12	140	120	2019/06/16 15:00

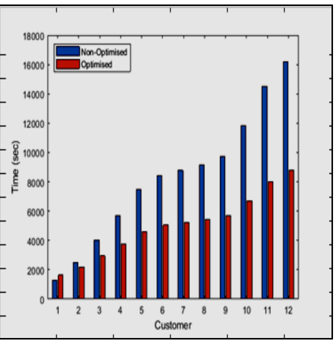


Figure 5. Comparison between the optimized and non-optimized date of delivery for twelve customer orders with required delivery on same day

As seen from Fig 2-5, in all four instances, the first customer order has a quicker delivery time for the non-optimized date of delivery as opposed to the customized delivery date of delivery. It is also important to note that this phenomenon is only visible for the first customer order, as all other customer orders are completed quicker in the optimized state of operation.

As this would be a demerit on the model, a test was done to check the status of the constraints as these two instances were input to the model. The status of the constraints was programmed into MATLAB in the form of a Graphic User Interface (GUI).

In the GUI, the status of all three constraints were shown prior to and after optimization. As mentioned in section 3, the water level in the storage tank had a threshold of 25%, while the 500ml and 750ml bottles available in the storage unit should not be less than the number of customer orders.

The GUI is depicted Fig 6 and it can be observed that in both instances, the water level in the main storage tank had gone below 25%, which was set as a threshold.

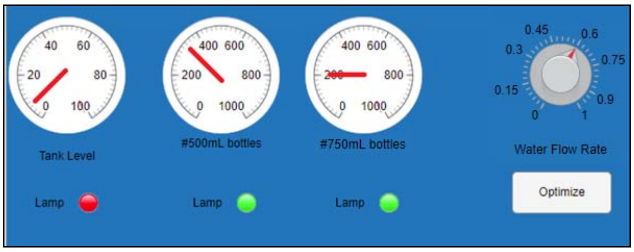
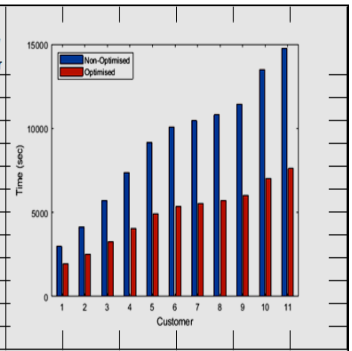


Figure 6. GUI showing status of constraints in non-optimized operation state

A further two instances were tested to check if the number of inputs or the same date of delivery had an impact on the functioning of the model. This notion was negated as shown in Fig 7.

#	500 ml orders	750 ml orders	Same date delivery with high first order
1	200	250	2019/06/08 15:00
2	85	95	2019/06/09 15:00
3	120	120	2019/06/10 15:00
4	100	150	2019/06/11 15:00
5	120	150	2019/06/12 15:00
6	79	69	2019/06/13 15:00
7	25	30	2019/06/14 15:00
8	25	30	2019/06/15 15:00
9	45	50	2019/06/16 15:00
10	100	200	2019/06/17 15:00
11	100	100	2019/06/18 15:00



#	500 ml orders	750 ml orders	Same date delivery with all 200 orders
1	200	200	2019/06/08 15:00
2	200	200	2019/06/08 15:00
3	200	200	2019/06/08 15:00
4	200	200	2019/06/08 15:00
5	200	200	2019/06/08 15:00
6	200	200	2019/06/08 15:00
7	200	200	2019/06/08 15:00
8	200	200	2019/06/08 15:00
9	200	200	2019/06/08 15:00
10	200	200	2019/06/08 15:00
11	200	200	2019/06/08 15:00

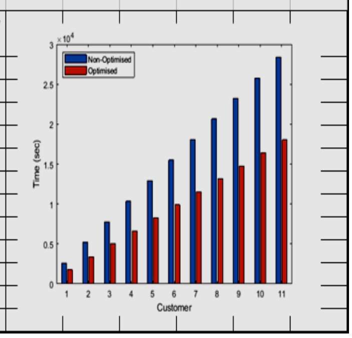


Figure 7. Comparison between the optimized and non-optimized date of delivery for customer orders with high inputs and same date of delivery

V. CONCLUSION

A set of fifty instances with diverse values of date and time of delivery and customer inputs were provided to the model. The results show that the model was successful in meeting the required date of delivery for all instances. However, in four instances (depicted in Fig 2, 3, 4 and 5), the optimized date and time of delivery was slower than the non-optimized date and time of delivery.

This anomaly was only evident for the first customer order. Thereafter, for every subsequent customer order, the optimized date and time of delivery was quicker than the non-optimized date and time of delivery.

A closer examination of the constraints, while these two instances were running, showed that the pump flow rate for the first customer order in the non-optimized state was very high and started using the water from the tank at a quick rate to complete the order. This resulted in the first order being completed quickly, but all subsequent orders being delayed because the tank had to be replenished.

The optimized delivery time, meanwhile, factored in the total number of orders and determined a suitable pump flow rate which would ensure that all customer orders were completed at an ideal time while also ensuring that the constraints were met.

This study also shows that the optimization model developed in this study is versatile and robust as it can handle very large customer orders (depicted in Fig 7) while negating the impact of positive drift. Therefore, the model could be used as a baseline in future study into Mixed-Model

Stochastic assembly line balancing and possibly also in a virtual commissioning environment.

VI. FUTURE WORK

This research has been presented as part of an ongoing research project at the Central University of Technology, Free State, studying the different types of Assembly Line Balancing Problems and how the assembly lines can be optimized to reduce production time, thereby increasing productivity.

The conclusions arrived in this specific study form a platform from which various other studies can be launched. This study specifically focused on mixed model stochastic assembly lines as they were the least researched, however there are many aspects that deem further introspection.

The first of which is to examine if the developed model can be altered so that it can be used as a digital twin for an existing or proposed physical plant. The advantage of such an approach is that, real time optimization can be done prior to plant operation and allow for digital monitoring of plants. This can open research paths into virtual/hybrid commissioning.

Another possible extension of this research is the scope of study into SMART manufacturing with specific focus on CLOUD manufacturing. The protocols that define communication between the CLOUD and the Smart Manufacturing Units (SMU's) have not been standardized due to the fast pace of development in this field and in the governing field of Industry 4.0. Using this research as a base, study can be done on defining standard protocols.

The model developed in this study can also be used create three separate SMU's. These can be used to investigate the decentralized operation of SMU's. Decentralized operation is one of the enabling factors of interoperability, a key characteristic of a SMART manufacturing environment. Decentralization allows for direct communication between the SMU's as opposed to communication between SMU's through a cloud server which can result in data latency.

ACKNOWLEDGMENT

The author would like to thank Prof HJ Vermaak who, as mentor of the student and leader of RGEMS group, has provided guidance and funding for this ongoing project.

REFERENCES

[1] N. Boysen et al., "A classification of assembly line balancing problemsA classification of assembly line balancing problems," Jenaer Schriften zur Wirtschaftswiss., pp. 1–27, 2006.

[2] R.B Kuriakose and H. Vermaak, "A Review of the Literature on Assembly Line Balancing Problems, the Methods Used to Meet These Challenges and the Future Scope of Study," Adv. Sci. Lett., vol. 24, no. 11, pp. 8846–8850, 2018.

[3] C. Merengo, F. Nava, and A. Pozetti, "Balancing and sequencing manual mixed-model assembly lines," Int. J. Prod. Res., vol. 37, no. 1, pp. 2835–2860, 1999.

[4] N. Kumar and D. Mahto, "Assembly Line Balancing : A Review of developments and trends in approach to industrial application," Glob. J. Res. Eng. Ind. Eng., vol. 13, no. 2, 2013.

[5] P. Sivasankaran and P. Shahabudeen, "Literature review of assembly line balancing problems," Int. J. Adv. Manuf., vol. 73, no. 9–12, pp. 1665–1694, 2014.

[6] H. Tempelmeier, "Practical considerations in the optimization of flow production systems. International," J. Prod. Res., vol. 41, no. 1, pp. 41, 149–170., 2003.

[7] Y. Kara, T. Paksoy, and C. Ter Chang, "Binary fuzzy goal programming approach to single model straight and U-shaped assembly line balancing," Eur. J. Oper. Res., vol. 195, no. 1, pp. 335–347.

[8] P. Sivasankaran and P. Shahabudeen, "Genetic Algorithm for Concurrent Balancing of Mixed-Model Assembly Lines with Original Task Times of Models," Intell. Inf. Manag., vol. 5, pp. 84–92, 2013.

[9] P. M. Vilarinho and A. N. A. Sofia, "A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations," Int. J. Prod. Res., vol. 40, no. 6, pp. 1405–1420, 2002.

[10] G. R Esmaeilian, M. Mahmad, S. Sulaiman, and N. Ismail, Assembly Line and Balancing Assembly Line. 2018.

[11] C. Becker and A. Scholl, "A survey on problems and methods in generalized assembly line balancing," Eur. J. Oper. Res., vol. 168, pp. 694–715, 2006.

[12] S. Ghosh and R. . Gagnon, "A Comprehensive Literature Review and Analysis of the Design, Balancing and Scheduling of Assembly Systems," Int. J. Prod. Res., vol. 27, no. 4, pp. 637–670, 1989.

[13] P. McMullen and G. Frazier, "Using Simulated Annealing to Solve a Multiobjective Assembly Line Balancing Problem With Parallel Workstations," Int. J. Prod. Res., vol. 36, no. 10, 2001.

[14] W. Xu and T. Xiao, "Mixed model assembly line balancing problem with fuzzy operation times and drifting operations," in Proceedings of the 2008 Winter Simulation Conference, 2008, pp. 1752–1760.

[15] R.B Kuriakose and H. . Vermaak, "Customized Mixed Model Stochastic Assembly Line Modelling Using Simulink," Int. J. Simul. Syst. Sci. Technol., vol. 20, no. 1, pp. 61–69, 2019.

[16] R.B Kuriakose and H. Vermaak, "Optimization of a Real Time Web Enabled Mixed Model Stochastic Assembly Line to Reduce Production Time," Commun. Comput. Inf. Syst., vol. 3, no. 1, pp. 29–50, 2019.

[17] R.B Kuriakose and H. J. Vermaak, "Optimization of a Customized Mixed Model Assembly Using MATLAB/Simulink," J. Phys. Conf. Ser., vol. 1201, no. 1, 2019.

[18] P. McMullen and P. Tarasewich, "Using Ant Techniques to Solve the Assembly Line Balancing Problem," IIE Trans., vol. 35, no. 7, pp. 605–617, 2003.

[19] W. Xu and T. Xiao, "Mixed model assembly line balancing problem with fuzzy operation times and drifting operations," in Proceedings of the 2008 Winter Simulation Conference, 2008, pp. 1752–1760.

[20] S. Matanachai and C. A. Yano, "Balancing mixed-model assembly lines to reduce work overload," IIE Trans., vol. 33, no. 1, pp. 29–42, 2001.

[21] P. Y. Tambe, "Balancing mixed-model assembly line to reduce work overload in a multi-level production system," Louisiana State University, 2006.