Communications in Computer and Information Science

ISSN:1865-0929

Springer

Communications in Computer and Information Science 1075

Commenced Publication in 2007
Founding and Former Series Editors:
Phoebe Chen, Alfredo Cuzzocrea, Xiaoyong Du, Orhun Kara, Ting Liu, Krishna M. Sivalingam, Dominik Ślęzak, Takashi Washio, and Xiaokang Yang

More information about this series at http://www.springer.com/series/7899

Ashish Kumar Luhach · Dharm Singh Jat ·
Kamarul Bin Ghazali Hawari ·
Xiao-Zhi Gao · Pawan Lingras (Eds.)

Advanced Informatics for Computing Research

Revised Selected Papers, Part I

Editors
Ashish Kumar Luhach
Papua New Guinea University
of Technology
Lae, Papua New Guinea

Kamarul Bin Ghazali Hawari
Universiti Malaysia Pahang
Pekan, Pahang, Malaysia

Pawan Lingras
Department of Mathematics
and Computing Science
Saint Mary's University
Halifax, Canada

Dharm Singh Jat
Computer Science Department
Namibia University of Science
and Technology
Windhoek, Namibia

Xiao-Zhi Gao
School of Computing
University of Eastern Finland
Kuopio, Finland

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-981-15-0107-4 ISBN 978-981-15-0108-1 (eBook)
https://doi.org/10.1007/978-981-15-0108-1

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

The Third International Conference on Advanced Informatics for Computing Research (ICAICR 2019) targeted state-of-the-art as well as emerging topics pertaining to advanced informatics for computing research and its implementation for engineering applications. The objective of this international conference is to provide opportunities for researchers, academics, industry professionals, and students to interact and exchange ideas, experience, and expertise in the current trends and strategies in information and communication technologies. Moreover, participants were informed about current and emerging technological developments in the field of advanced informatics and its applications, which were thoroughly explored and discussed.

ICAICR 2019 was held during June 15–16 in Shimla, India in association with Namibia University of Science and Technology and technically sponsored by the CSI Jaipur Chapter, MRK Institute of Engineering and Technology, Haryana, India, and Leafra Research Pvt. Ltd., Haryana, India.

We are very thankful to our valuable authors for their contribution and our Technical Program Committee for their immense support and motivation for making the first edition of ICAICR 2019 a success. We are also grateful to our keynote speakers for sharing their precious work and enlightening the delegates of the conference. We express our sincere gratitude to our publication partner, Springer, for believing in us.

June 2019

Ashish Kumar Luhach
Dharm Singh Jat
Kamarul Bin Ghazali Hawari
Xiao-Zhi Gao
Pawan Lingras

Contents – Part I

Contents – Part II

Software and Its Engineering

Optimization of a Real Time Web Enabled Mixed Model Stochastic Assembly Line to Reduce Production Time

Rangith Baby Kuriakose(✉) and Hermanus Jacobus Vermaak

Department of Electrical, Electronics and Computer Systems, Central University of Technology, Bloemfontein 9301, Free State, South Africa
{rkuriako,hvermak}@cut.ac.za

Abstract. The role of assembly lines has never been more critical as it is now with the world entering the 4th Industrial Revolution, commonly referred to as Industry 4.0. If the focus of the previous industrial revolution was on mass production, the focus of Industry 4.0 is on mass customization. One of the major changes mass customization brings about to an assembly line is the need for them to be autonomous. An autonomous assembly line needs to have the following key features; ability to provide a ubiquitous input, the ability to optimize the model in real time and achieve product variety. Product variety, in this context, refers to different variants of the same product as determined by the user. Assembly lines that make provision for introducing product variety are termed as mixed-model assembly lines. Mixed-model assembly lines become stochastic in nature when the inputs are customized as time cannot be predetermined in a stochastic process. The challenge, as it stands, is that there are limited discussions on real-time optimization of mixed model stochastic assembly lines. This paper aims to highlight this challenge by considering the case study of optimizing a mixed model assembly line in the form of a water bottling plant. The water bottling plant, which needs to produce two variants of the bottled water, 500 ml, and 750 ml, takes customer inputs through a web interface linked to the model, thereby making it stochastic in nature. The paper initially details how the model replicating the functioning of the water bottling plant was developed in MATLAB. Then, it proceeds to show how the model was optimized in real time with respect to certain constraints. The key results of the study, among others, showcase how the optimization of the model is able to significantly reduce production time.

Keywords: Real Time Optimization · Cloud manufacturing · Mixed model · Assembly lines · Stochastic processes · Industry 4.0

1 Introduction

Assembly lines are critical to the operation of any successful manufacturing plant [1]. This holds value even as the world is embracing the fourth Industrial Revolution [2] commonly termed as Industry 4.0 (I.4.0). The industrial revolutions of the past focused on mass production [3, 4] and reducing production time.

However, with the advent of I.4.0, the focus has shifted from mass production to mass customization [5]. Mass customization is key to introducing product variety [6]

A. K. Luhach et al. (Eds.): ICAICR 2019, CCIS 1075, pp. 39–50, 2019.
https://doi.org/10.1007/978-981-15-0108-1_5

into manufacturing. With product variety, customers are given the opportunity to make variations in the final product.

Product variety is achieved by using Multi/Mixed model assembly lines [7, 8] with stochastic task times [9] and as a result poses problems in assembly line balancing. Assembly Line Balancing Problems (ALBP) are a study of the different types of problems [10] encountered in balancing assembly lines.

There have been many classifications of ALBP's, but the most referenced include that by Ghosh and Gagnon in 1989 [11], Scholl and Becker in 2006 [12] and finally by Sivasankaran and Shahubdeen in 2014 [13]. The later study has split Multi/Mixed model Stochastic (MMS) assembly lines in accordance to the line layout to be either S-type or U-type.

The complexity of a MMS assembly line is exacerbated if one considers real-time inputs as this would need Real-Time Optimization (RTO) [14, 15]. As it stands there is very limited research done on MMS assembly line balancing and even less literature on Real-Time Optimization of MMS assembly lines.

Therefore, in order to garner fresh perspectives in this research niche area, this paper considers optimization of a real-time mixed model assembly line. A case study of a water bottling plant that can manufacture 500 ml and 750 ml with real time inputs from a web server is considered for this purpose.

The paper initially considers the design criteria for the water bottling plant. Then it looks at how the plant was modelled in MATLAB and real-time inputs were provided through a Web Server. Thirdly, the paper focuses on the problem formulation and optimization of the model and finally the results thereof are discussed.

2 Water Bottling Plant as a Case Study

As stated in the introduction, in-order to design a multi mixed assembly line, a case study needs to be selected. In this study, a business plan [16] to realize a water bottling plant at the Bloemfontein campus of the Central University of Technology, was chosen as the case study. A three-dimensional CAD model of the proposed model is shown in Fig. 1.

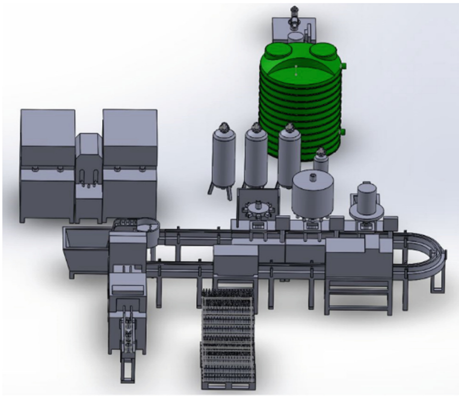

Fig. 1. 3-dimensional CAD model of the plant

The CAD model was split into three main subsystems for creating a Simulink model. The three subsystems with their specific tasks are defined as follows;

- Subsystem A – Source and Storage tank
- Subsystem B – Bottle manufacturing and storage
- Subsystem C – Water filling

The Simulink model is depicted in Fig. 2 with the different subsystems. As seen in Fig. 2, the input is provided through a web server. The input is fed to the water filling substation. Based on the inputs, the water filling subsystem triggers water and bottles from subsystem A and B respectively.

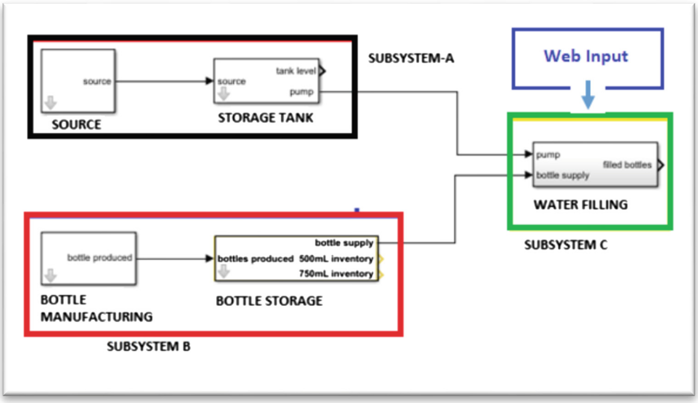

Fig. 2. Simulink model of the plant with three subsystems

3 Model Overview

This section elaborates on the Simulink model described in Fig. 2. This is done by initially focusing on the web input and how it is setup. Secondly the focus will be on the three subsystems with an aim to explain the functioning of each block in the respective subsystems. This will form the base for discussion on the problem formulation and optimization, which is discussed in Sect. 4.

3.1 Customer Inputs Using Web Apps

Web apps are MATLAB [17] applications that can run in a web browser. An interactive Graphic User Interface (GUI) is created using the App Designer and packaged using a Web App Compiler and hosted on the MATLAB Web Server App. A unique URL is created for each web app and can be accessed from a browser using HTTP or HTTPS protocols.

In this paper, the web app is designed to first collect information from customers on the number of 500 ml and 750 ml bottles of water that needs to be ordered and the required date of delivery. This information is captured and saved on the. A program written in MATLAB is used to access this saved file and process it so as to start the water filling process. A picture depicting the GUI is shown in Fig. 3.

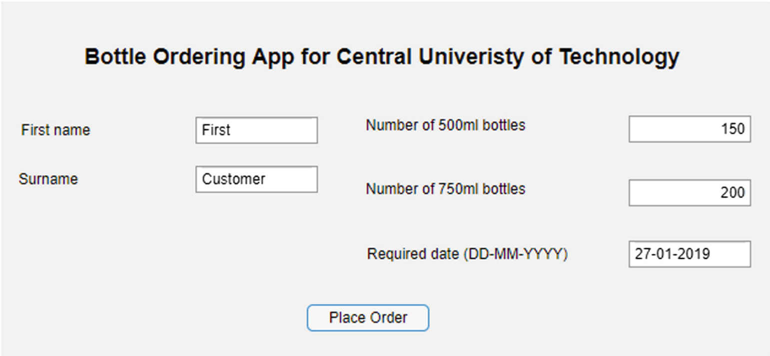

Fig. 3. GUI for bottle ordering app

3.2 Source and Storage Subsystem-Subsystem A

As depicted in Fig. 3, subsystem A acts as one of the inputs to subsystem C. Subsystem A consists of two subsystems being the water source and the storage tank. The source subsystem is designed to provide purified water to the plant. For design purposes, the source subsystem is a masked meaning that parameters such as the flow rate and upper limit of water from the source system are user defined in the mask.

The storage tank subsystem needs to take purified water from the source subsystem and pump it into the water filling subsystem. These conditions make it a continuous state subsystem. Continuous states, in the context of Simulink, refers to a variable whose value is determined through numerical integration of its derivative with respect to time.

As with the source subsystem, the storage tank subsystem is also masked, with values like the upper limit of the tank, maximum capacity and pump flow rate defined under the mask. For optimization purposes, the pump flow rate is defined as ‘x’ in the mask as it acts as the handle and needs to be varied with respect to the constraints.

3.3 Bottle Manufacturing Subsystem-Subsystem B

Subsystem B consists of the bottle manufacturing subsystem and the bottle storage subsystem. Subsystem B is independent of subsystem A and the output of the bottle storage subsystem is fed to subsystem C.

The bottle manufacturing subsystem, in keeping with the design parameters, require that bottles be manufactured in two sizes, 500 ml and 750 ml. The bottles are manufactured according to the order from the web input through the bottle ordering app depicted in Fig. 3.

The manufacturing and storage is done by retrieving the orders, saved on the server, into a one dimensional look-up table. The data in the look-up table can be transposed and flattened to appear as a row of information to the table. A relational table with a memory block can be used to check if the number of bottles in each row has been reached.

3.4 Water Filling Subsystem-Subsystem C

As explained previously, subsystem C takes inputs from subsystem A and B. Subsystem A provides the water and subsystem B provides the bottles. The data from the one dimensional lookup table described in subsystem B is used for filling the bottles.

As soon as the number of bottles in the first row has been achieved, a trigger element can be set up to first index the data and move to the next row in the look-up table. This can be continued till the last row in the look-up table has been read into the model and defined in the index.

The distinction between 500 ml and 750 ml bottles can be made by calling a function in the model which checks the index where the data has been read from. By default, the 500 ml bottles will be indexed in the odd rows while the 750 ml will be indexed in the even rows. The output of the bottle storage unit is then provided as input to subsystem C.

4 Problem Formulation and Optimization

The model that has been designed in this study has customized inputs as the inputs are fed through a web interface. This would mean that model will have real inputs. Therefore, the optimization problem needs to be formulated [20] as a Real Time Optimization (RTO) problem.

The formulation can be split into five steps described as follows;

- Step 1: Determine process variables – The plant model has the following variables;
  - Water stored in the tank in the Source subsystem
  - Flow rate of water from the pump in the storage tank subsystem
  - Initial number of 500 ml bottles in the bottle manufacturing subsystem
  - Initial number of 750 ml bottles in the bottle manufacturing subsystem
  - 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 three constraints being firstly the water level of the tank and secondly the number of 500 ml and 750 ml bottles available in storage. The water level in the tank should never go below 25%. 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 non-linear relationship. Since the modelling is done with Simulink, the optimization can be done using MATLAB.

The Optimization ToolBox™ in MATLAB [21] offers variety of functions to solve the different types of optimization problems. Step 5 described in the problem formulation localizes the problem in this paper to focus on analyzing constrained nonlinear optimization techniques.

There are mainly three functions available in MATLAB for nonlinear constrained optimization. They along with their specific purpose is listed as follows;

- *fminbnd* – Finding minimum variable function on fixed intervals
- *fmincon* – Find minimum of constrained nonlinear multivariable function
- *fseminf* – Find minimum of semi-infinitely constrained multivariable nonlinear function

It is evident that in the listed MATLAB functions, *fmincon*, is best suited for the plant model optimization as it accommodates the minimizing of nonlinear objective equations with constraints and multivariable functions. The *fmincon* function has the following syntax;

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

Where,

x0 = starting point of minimization

fun = function to minimize

b and *beq* are linear constraints

A and *Aeq* are nonlinear constraints

lb = lower boundary of constraint

ub = upper boundary of constraint

Here, the syntax needs *x0*, a starting point for the minimization. This is kept at 0.1. Next it needs a lower and upper boundary for the pump flow rate, which is the handle. This will ensure that the solution is always within the range of $lb \leq x \leq ub$. Here, *lb* = 0 and the *ub* = 1. As there are no inequality constraints, *Aeq* = [] and *beq* = [].

5 Results and Discussion

This section presents the results of the work described. This is done by comparing the non-optimized and optimized pump flow rate against the constraints defined in the problem formulation. Firstly, the customer requirements as obtained from the cloud server for this set of tests are shown in Table 1.

Table 1. Customer requirements table as per the input from the cloud server.

Customer number	Number of 500 ml bottles required	Number of 500 ml bottles required	Required date and time of delivery
1	100	100	16-Jan-2019 15:00
2	85	95	16-Jan-2019 15:00
3	120	120	16-Jan-2019 15:00
4	100	150	17-Jan-2019 15:00
5	120	150	17-Jan-2019 15:00
6	79	69	17-Jan-2019 15:00

Secondly, the pump flow rate, which acts as the handle, is varied. A set of three tests are done on the inputs shown in Table 1. The first two being with manual, non-optimized pump flow rates and the third one a model generated, optimized flow rate. The flow rate can be varied between 0 m/s to 1.0 m/s.

As a first test, the model is provided with a manual pump flow rate of 0.6 m/s. The assumption here is that with a high flow rate the orders will be completed at a fast rate. The result of this test is given Table 2.

Table 2. Customer requirements table with non-optimized delivery date with pump flow rate at 0.6 m/s

Customer number	Number of 500 ml bottles required	Number of 500 ml bottles required	Required date and time of delivery	Non-optimized date and time of delivery
1	100	100	16-Jan-2019 15:00	16-Jan-2019 02:54
2	85	95	16-Jan-2019 15:00	16-Jan-2019 03:14
3	120	120	16-Jan-2019 15:00	16-Jan-2019 03:42
4	100	150	17-Jan-2019 15:00	17-Jan-2019 12:07
5	120	150	17-Jan-2019 15:00	**17-Jan-2019 16:37**
6	79	69	17-Jan-2019 15:00	**17-Jan-2019 17:01**

It can be deduced from Table 2 that first four orders (Customer's 1, 2, 3 and 4) are completed before the required date of delivery. However, the last two orders (Customer's 5 and 6) are not completed on time. This is owing to the fact that the high water flow rate has resulted in the water in the tank being depleted. Therefore, the assembly line process has to be halted to allow for the tank to be replenished.

After the tank has been replenished, the assembly line process continues, but the time lost during the replenishing cannot be made up and hence the customer orders 5 and 6 fall behind of schedule. A GUI depicting the status of the constraints using gauges is shown in Fig. 4. It can be seen here that the gauge showing tank level has gone below 25%. This is a fail as in the problem formulation the first condition was that the tank level should not go below 25%.

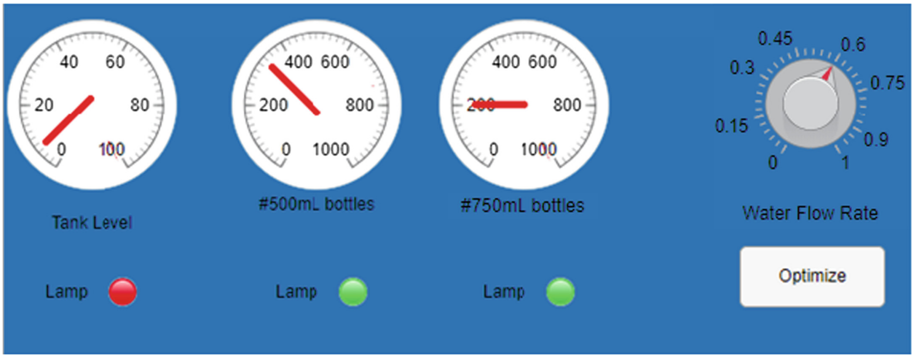

Fig. 4. GUI showing the status of the constraints at 0.6 m/s pump flow rate

The second tests had the model being provided with a manual pump flow rate of 0.1 m/s. The assumption here is that the slower pump flow rate would result in the constraints being met and therefore the orders will be completed in time. The result of this test is given Table 3.

Table 3. Customer requirements table with non-optimized delivery date with pump flow rate at 0.1 m/s

Customer number	Number of 500 ml bottles required	Number of 500 ml bottles required	Required date and time of delivery	Non-optimized date and time of delivery
1	100	100	16-Jan-2019 15:00	16-Jan-2019 12:50
2	85	95	16-Jan-2019 15:00	16-Jan-2019 13:23
3	120	120	16-Jan-2019 15:00	**16-Jan-2019 21:25**
4	100	150	17-Jan-2019 15:00	17-Jan-2019 07:14
5	120	150	17-Jan-2019 15:00	17-Jan-2019 13:26
6	79	69	17-Jan-2019 15:00	**17-Jan-2019 15:53**

It can be seen from Table 3 that the rate of completing the orders is much slower in this instance. This is evident in that customer orders 3 and 6 are not completed within the required time. However, unlike with the previous test, all constraints are met in this test. This is depicted in Fig. 5.

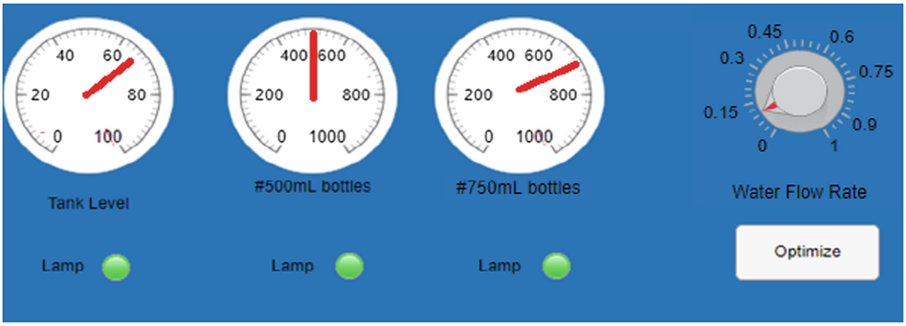

Fig. 5. GUI showing the status of the constraints at 0.1 m/s pump flow rate

The instances where production time exceeds the allotted time is termed as positive drift [3]. These tests and results put a strong emphasis for the need for a robust optimization function for this model. The optimization should be able to meet all constraints while ensuring that the required delivery dates are met.

As described in the problem formulation in Sect. 4, the MATLAB optimization function, *fmincon*, is applied on the model. The syntax defined in the problem formulation is expanded out as follows;

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

Here, the starting point of the pump flow rate and the lower and upper boundaries of the constraint are defined. The starting point is defined as 0.1 and the lower boundary is 0, while the upper boundary is 1. *XOpt* is the optimized pump flow rate which will meet the defined constraints and *TTMOpt* is the 'Time To Manufacture' with optimized pump flow rate.

After completing the optimization, the *fmincon* function does a further check to test the robustness of the model by adding 0.001 to the optimized pump flow rate. The check should meet the constraints as before to ensure the model is robust and not prone to even the minutest of changes. The result of the optimization is summarized in Table 4.

Table 4. Customer requirements table with optimized time and date of delivery

Customer number	Number of 500 ml bottles required	Number of 500 ml bottles required	Required date and time of delivery	Optimized date and time of delivery
1	100	100	16-Jan-2019 15:00	16-Jan-2019 02:54
2	85	95	16-Jan-2019 15:00	16-Jan-2019 03:30
3	120	120	16-Jan-2019 15:00	16-Jan-2019 04:26
4	100	150	17-Jan-2019 15:00	16-Jan-2019 05:17
5	120	150	17-Jan-2019 15:00	16-Jan-2019 06:42
6	79	69	17-Jan-2019 15:00	16-Jan-2019 08:47

The optimized pump flow rate for this specific set of inputs was 0.36. It can be seen here that at the optimized pump flow rate, all the customer orders are met well before the required date and time of delivery.

On further analysis, the constraints such as the level of water in the tank and the number of bottles available in the inventory are met on completion of the orders. The status of the constraints is depicted in Fig. 6.

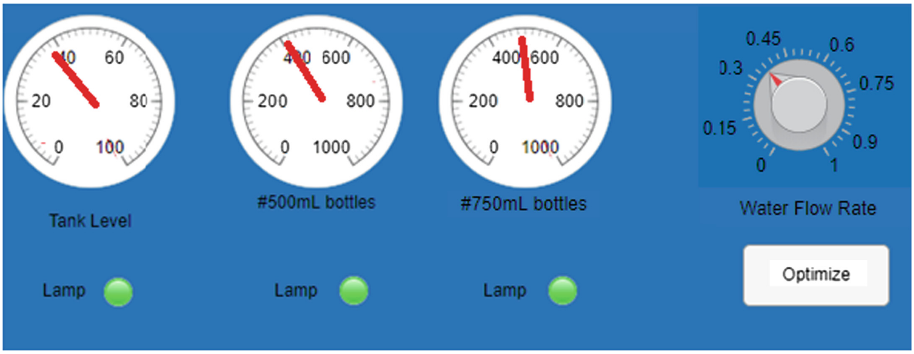

Fig. 6. GUI showing the status of the constraints at 0.1 m/s pump flow rate

6 Conclusion and Future Work

This study was necessitated due to the limited research done on Real Time Optimization of Multi Mixed Stochastic assembly lines. In order to facilitate this study, a case study was chosen.

The case study was on a water bottling plant that can manufacture 500 ml and 750 ml with real time customer inputs provided through a web server. The paper initially describes how the plant was modelled on Simulink. Secondly, the paper shows how a web app was developed on MATLAB and finally how the optimization problem was formulated.

The results of the paper initially discuss how a set of orders received through the web server is completed using a manual pump flow rate. Two sets of pump flow rates were chosen for test purposes. The first pump flow rate was chosen to be 0.6 m/s and the second pump flow rate was chosen as 0.1 m/s.

The first test instance showed that the first few orders were completed at a rapid pace. However, this resulted in the water in the tank being depleted at a fast rate. This meant that after the first few orders had been completed, the process had to be halted for the water in the tank to be replenished. This eventually resulted in two orders not meeting the required delivery date and time. These results are summarized in Table 2 and Fig. 4.

The second test, which used a slower pump flow rate, resulted in all constraints being met as depicted in Fig. 6. However, like in the first instance, a few orders were not met as per the required date of delivery and time. This is evident from the data shown in Table 3.

The required date of delivery was not met in both instances and necessitated the need for optimization. The MATLAB optimization function *fmincon* was used to optimize the production time based on the customer inputs and the required date of delivery. It was seen that the required delivery date and the constraints were met upon optimization. This is depicted in Table 4 and Fig. 6.

This proves that optimization technique used in this study was successfully able to reduce the production time for a mixed model stochastic assembly line with real time inputs. As part of future work, to establish the veracity and robustness of the model, numerous sets of inputs with various scenarios as far as the required date of delivery and number of orders needs to be applied to the model. The number of constraints, three in this study, could also be increased to see the stress it creates on the model.

The results of the suggested tests, as part of the future work, can add valuable insight into the field of mixed model stochastic assembly line balancing. The results could also determine if this model can be tested on an existing manufacturing plant and form part of creating an industry standard that can be used in future to combat problems related to positive drift.

Acknowledgements:. The authors would like to thank the RGEMS group of Central University of Technology, Free State for providing the lab resources to pursue this project. The authors would also like to thank MerSETA for financially contributing to the successful completion of the project.

References

1. Tun, U., Onn, H., Sulaiman, S., Ismail, N.: Assembly Line and Balancing Assembly Line, January 2015
2. Baldassarre, F., Ricciardi, F., Campo, R.: The Advent of Industry 4.0 in manufacturing industry: Literature review and growth opportunities. In: DIEM: Dubrovnik International Economic Meeting, pp. 632–643 (2017)
3. Hu, S.J., et al.: Assembly system design and operations for product variety. CIRP Ann. Manuf. Technol. **60**(2), 715–733 (2011)
4. Kuriakose, R.B., Vermaak, H.J.: 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. **24**(11), 8846–8850 (2018)
5. Um, J., Lyons, A., Lam, H.K.S., Cheng, T.C.E., Dominguez-pery, C.: Product variety management and supply chain performance: a capability perspective on their relationships and competitiveness implications. Int. J. Prod. Econ. **187**, 15–26 (2017)
6. Wang, H., Hu, S.: Manufacturing complexity in assembly systems with hybrid configurations and its impact on throughput. CIRP Ann. Manuf. Technol. **59**(1), 53–56 (2010)
7. Kuriakose, R.B., Vermaak, H.J.: Optimization of a customized mixed model assembly line using MATLAB/Simulink. J. Phys.: Conf. Ser. **1201**, 012017 (2019)
8. Reginato, G., Anzanello, M.J., Kahmann, A., Schmidt, L.: Mixed assembly line balancing method in scenarios with different mix of products. Gestão Produção **23**(2), 294–307 (2016)
9. Baykasoglu, A., Ozbakir, L.: Stochastic U-line balancing using genetic algorithms. Int. J. Adv. Manuf. Technol. **32**, 139–147 (2007)

10. Kumar, N., Mahto, D.: Assembly line balancing: a review of developments and trends in approach to industrial application. Glob. J. Res. Eng.: Ind. Eng. **13**(2) (2013)
11. Ghosh, S., Gagnon, R.: A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. Int. J. Prod. Res. **27**(4), 637–670 (1989)
12. Becker, C., Scholl, A.: A survey on problems and methods in generalized assembly line balancing. Eur. J. Oper. Res. **168**(3), 694–715 (2006)
13. Sivasankaran, P., Shahabudeen, P.: Literature review of assembly line balancing problems. Int. J. Adv. Manuf. Technol. **73**(9–12), 1665–1694 (2014)
14. Bonvin, D.: Preface to Real-Time Optimization Processes, Special edition, pp. 1–5 (2017)
15. Francois, G., Bonvin, D.: Real-time optimization: optimizing the operation of energy systems in the presence of uncertainty and disturbances. In: 13th International Conference on Sustainable Energy technologies (SET2014), pp. 1–12 (2014)
16. Kuriakose, R.B., Vermaak, H.J.: Customized mixed model stochastic assembly line modelling using Simulink. Int. J. Simul. Syst. Sci. Technol. **20**(1) (2019). ISSN 1473-804X
17. MATLAB: MATLAB Web Apps, MATLAB Documentation (2018). https://www.mathworks.com/help/compiler/web-apps.html. Accessed 28 Apr 2019
18. Kumar, D.N.: Introduction and basic concepts - classification of optimization problems, National Program on Technology Enhanced Learning. http://nptel.ac.in/courses/Webcourse-contents/IISc-ANG/OPTIMIZATION%20METHODS/pdf/Module_1/M1L3slides.pdf. Accessed Apr 2019
19. Kumar, D.N.: Optimization problem and Model formulation, Optimization Methods. http://msulaiman.org/onewebmedia/Lecture2mphil.pdf. Accessed 28 Mar 2018
20. Edgar, T., Himmelblau, D., Ladson, L.: Optimization of Chemical Processes, 2nd edn. McGraw-Hill Higher Education, New York (2001)
21. MATLAB: MATLAB Optimization Toolbox: User's Guide, MATLAB Documentation (2018). https://www.mathworks.com/help/pdf_doc/optim/optim_tb.pdf. Accessed 02 Jan 2019