

Balancing a Mixed-Model Assembly System in the Footwear Industry

Parisa Sadeghi^{1,2}(✉), Rui Diogo Rebelo², and José Soeiro Ferreira^{1,2}

¹ Faculty of Engineering, University of Porto,
Campus da FEUP, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

² INESC TEC - Technology and Science,
Campus da FEUP, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
p.sadeghi@fe.up.pt

Abstract. Portuguese footwear industry has improved dramatically to become one of the main world players. This work is part of a project in cooperation with a large footwear company, operating a new automated assembly equipment, integrating various lines. Balancing such lines implies going from an almost manual preparation executed by experienced operators, to a planning supported by optimisation systems. These complex mixed-model lines have distinctive characteristics, which make balancing a unique problem. The paper proposes the ASBsm – Assembly System Balancing Solution Method, a new method that integrates a constructive heuristic and an improvement heuristic, which takes inspiration from Tabu Search. The solutions obtained, based on real instances, are quite encouraging when compared with other effected factory solutions. Consequently, the balances obtained by ASBsm are now being implemented and articulated with sequencing methods.

Keywords: Footwear industry · Mixed-model assembly line balancing · Optimisation · Heuristics

1 Introduction

Portugal is one of the major world players in the footwear industry. This is the case because the country has chosen to invest in technology innovation, research, manpower qualification, innovative design and internationalisation. Much has changed, from low-cost mass production to serving clients consisting of small retail chains, where orders are small and models are varied. Consequently, work plans vary frequently and traditional flow lines are steadily being replaced by more flexible and sophisticated systems. The case presented in this paper applies to this context of progress and intends to contribute to solving complex balancing problems arising from the new mixed-model flexible assembly system of an important footwear factory.

Footwear manufacturing typically involves cutting, stitching and assembly processes. This work focuses on stitching lines, which are frequently a bottleneck in this industrial sector. Shoe components are placed inside boxes, usually

not more than 10 pairs, which move along the lines, between any appropriate workstation. The conveyors also transport them from and to the warehouses.

Each model inside a box requires a set of different tasks, with processing times and specific sequences, represented in graphs – the routings. A workstation (operator and machine) should be assigned to each task. In rapidly changing production plans, managing the lines, is a critical issue. Unsuitable plans will create bottlenecks and starvations; appropriate plans will contribute to better sequences, to a better use of the resources and to competitive advantages.

These problems are known in the literature as Assembly Line Balancing Problems (ALBP). As mentioned in Scholl (1999), these are crucial optimisation problems because they are highly complex and important in real world industries. Saif et al. (2014) divide ALBP into different categories. The problem dealt with here is a Mixed-model ALBP (MALBP), with extra complexities. Scholl and Becker (2006), Tasan and Tunali (2008), Chen et al. (2012) refer to this type of problems and illustrate various objectives such as, minimising the number of workstations, reducing the cycle time, maximising the line efficiency and finding a feasible solution.

The paper is organised as follows: Sect. 2 briefly describes the assembly system and the type of balancing problem. Section 3 proposes a solution method -ASBsm, which incorporates a constructive heuristic and an improvement heuristic inspired in Tabu Search (TS). Section 4 includes the computational results based on real data. Finally, some conclusions are provided in Sect. 5, regarding the feasibility of the method to improve current solutions.

2 The Assembly System Balancing Problem

The Balancing Problem described here is being addressed in ongoing research projects in a large footwear company. The assembly system is a new automated production and transportation system especially designed to deal with the current small amounts of very diverse models and types of demand. It is composed of two separate parts located in distinct places. One of them has four stitching lines, with a set of workstations, and conveyors transporting boxes with the components. The other is a U-shape line without an automated warehouse. Although there are differences between the two types of lines, what is important to emphasize is that the boxes can directly move from a workstation to another, according to the planned sequence of tasks, and, in case of the larger system, between the warehouses and any workstation. Figure 1 illustrates these lines. The main purpose of this work is to devise and implement a new balancing method.

Workstations are composed of machines and operators, whose skills must be taken into consideration. There are special operators who are the only ones with the ability to perform some tasks, which should be preassigned to such operators. Machines are classified according to their types and capability to execute certain tasks. There are different types of machines. Each machine has only one operator, but an operator can work with various machines. Shoe components

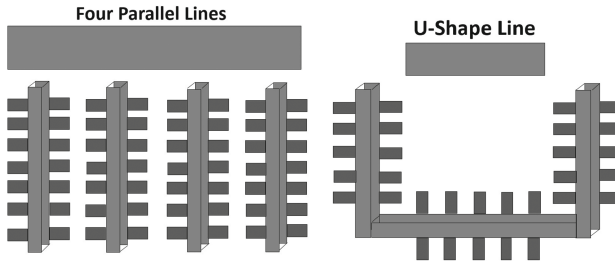


Fig. 1. Different system of stitching lines.

are inside boxes, which are moved by the conveyors. Each box has an associated routing, represented by a graph, defining the operations to be completed in the components and the precedence relations between them. According to the company, balancing involves two main purposes:

Each box has an associated routing, which defines the operations that must be completed in the components and the precedence relations between them. Routings are represented by graphs. In this paper, and according to the company, balancing involves two main purposes:

1. reducing the number of workstations required to complete a known production plan;
2. smoothing operators' workloads.

In the objective function, the weight of each part is analysed taking into consideration their relative importance. This MALBP is unique due to the automated transportation system and the industrial sector.

3 Solution Method: ASBsm

ASBsm integrates two heuristics: a constructive heuristic to generate an initial solution and an improvement heuristic, which is a neighbourhood search procedure inspired by TS. The venture into this method is a consequence of various types of considerations: a new assembly system composed of various stitching lines; reaching a good equilibrium between practical solutions and real modelling limitations; the intrinsic combinatorial complexity of the MALBP; the large dimension of the instances and their unclear data. ASBsm details are presented in the following sections.

3.1 ASBsm: Constructive Heuristic

Production orders of various footwear models are known in advance. The first phase implies the creation of boxes with the shoe parts, usually containing a maximum of 10 sets. The following example with two models *A* and *B*, with order quantities of 25 and 20 pairs, respectively, will be used to illustrate the heuristic.

Table 1. Creating boxes and quantities.

Model (j)	Box _{ij}		
	Box (i = 1)	Box (i = 2)	Box (i = 3)
A (j = 1)	10	10	5
B (j = 2)	10	10	-

Table 2. Routings.

Model	Task	Sequence
A	10	2
A	20	1
A	30	3
B	10	1
B	30	2
B	40	3

Table 3. TasModBox matrix.

Box	Model	Task
1	A	20
1	A	10
1	A	30
1	B	10
1	B	30
1	B	40
2	A	20
2	A	10
2	A	30
2	B	10
2	B	30
2	B	40
3	A	20
3	A	10
3	A	30

A possible set of corresponding boxes is in Table 1. Inside each box there are items of the same model, size and colour. Each box has two identifications: The box counter (*i*) and the model type (*j*).

The next phase assigns tasks of each box to suitable operators and machines. The heuristic will sort all tasks of all boxes. For that, at the beginning, boxes of all models are ordered. Afterwards, the prior model of the first box is collected (in the example model *A* of box 1, with the quantity of 10). Then the boxes undergo different tasks according to their sequence and model type. Table 2 shows the sequences. After, next model of the first box is selected (box 1 of *B* with the quantity 10, in the example) and, when all tasks of all first boxes are listed, then the procedure is repeated for the next boxes until all tasks are sorted in a matrix called *TasModBox*. Table 3 shows the corresponding matrix. For some tasks, there is only one resource available, which leads to a pre-assignment. There are two types of pre-assignment, the first taking into account the operators and the second the machines, as explained below:

1. If only one operator has the ability to perform a specific task, such task must be assigned to him, the workload being updated taking into consideration the processing time of the task multiplied by the amount of boxes. Other tasks can be assigned to that operator if there is enough free time.
2. If a task requires a specific type of machine and only one of those machines exist, it must be selected for that task. Afterwards, if that machine still has enough free time, it may be assigned to other adequate tasks.

After allocating special tasks to the operators and/or machines capable of executing them, the heuristic allocates the remaining tasks. It starts by assigning tasks of *TasModBox* matrix according to their order and respecting the constraints below:

1. Each task of a model of a box should be assigned to a required operator;
2. Each task of a model of a box should be assigned to exactly one operator;
3. Each task of a model of a box should be assigned to a required machine type;
4. Each task of a model of a box should be assigned to exactly one machine of the required type;
5. Each operator should not work more than an available period of time or defined percentage of the available time;
6. Each machine can only be allocated to one operator;
7. N.er of machines used is less than or equal to a maximum n.er of machines.

After picking a task, the heuristic finds the first operator with the required skill and available time to work on the components in the box; otherwise, it searches for another operator. Next step, it searches for a required type of machine, not yet allocated, repeating the procedure until all tasks are routed. When a task is assigned to an operator and a machine, then a corresponding workstation is created. Quite often Work In Process (WIP) has to be taken into consideration, as there are boxes in the lines, with some tasks already executed. Consequent adaptations of the input data are performed such as updating routings.

3.2 ASBsm: Improvement Heuristic

ASBsm integrates an Improvement Heuristic, which evolves by using different problem adequate neighbourhoods and local search. It takes some inspiration from TS, which has also been applied to solve ALBP (Özcan and Toklui 2009), but not specifically in the footwear industry. The heuristic improves the solutions, not only by local search, but also by promoting diversification. The concept of tabu list shows up when these neighbourhoods are introduced along the procedure.

In order to create appropriate solutions, a careful attempt is made to devise effective neighbourhood structures (N_i), by transferring and swapping a task, or by performing some tasks at once between workstations. All neighbours of a

current solution are evaluated and the best one, regarding the objective function, is chosen. The dimension of each neighbourhood structure is related to the instances used. The essentials are explained below.

N1: transfer all tasks of a machine and operator to another machine and the assigned operator;

N2: transfer one task of a machine and operator to another machine and the assigned operator;

N3: transfer tasks of a machine and operator to other machines and the assigned operators one by one (could be more than one machine and operator);

N4: if the first operator works on more than one machine, swap all tasks of a machine and operator with another machine and the assigned operator.

The search progression for all neighbours has similarities and dissimilarities. Two situations may occur for all neighbourhood spaces: in one situation, an occupied machine is selected and if it happens, it is of one type only. The heuristic then looks for another occupied machine of the same type; following this, the goal is to find operators who are assigned to each of the machines. In a second situation, an occupied machine is taken again but it has different types. Then, another occupied machine is chosen, regardless of its types. After that, the operators assigned to both machines are established.

About *N1*, and considering the first situation, if the operator of the second machine has the ability to perform every task in the first machine and has sufficient free time, it is possible to transfer tasks. In the second situation, all assigned tasks of the first machine and operator, one by one, are considered and if the second machine and operator have the ability to perform them and they also have enough free time, tasks can be transferred as well.

In *N2*, the difference to *N1* is that only one task transferred at each step.

Regarding *N3*, tasks are transferred one by one and if the selected operator and machine do not have enough time, another machine and operator will be chosen to perform the remaining tasks. Therefore, tasks assigned to a machine and operator may be transferred to different machines and operators.

N4 essentially follows the previous processes. However, in this case all tasks of both machines will be controlled to see if each machine and operator have the ability and enough free time for new tasks. At the beginning the heuristic looks for operators who have worked on more than one machine.

The computational experiments showed that the neighbourhood structures can have different impacts on both objectives. Table 4 illustrates the best impact for each objective.

Tabu list and Diversification: ASBsm includes a diversification phase to drive the search into new regions by changing the neighbourhood types. After exploring a neighbourhood and the best solution is found, such neighbourhood enters a tabu list, being forbidden for a number of iterations. In the next step, a new neighbourhood is searched. It is expected that other neighbourhoods, with diverse effects, may have the chance to improve the current solution. As there are 4 neighbourhood structures, the length of the tabu list is 2 iterations.

Table 4. Neighbourhoods and objectives.

Reducing N.er worksations	Smoothing workload
N1	–
N2	N2
N3	–
–	N4

Aspiration criterion: If a neighbourhood structure (non-tabu) does not improve the current solution, then an aspiration criterion is employed, overriding a tabu state by resorting to the older element of the list. If again an improvement is not met, then another tabu neighbourhood (first, not recently becoming tabu, then recently being tabu) is applied and examined.

Termination condition: The termination condition is 30 iterations or no improvement occurs during these iterations; then the algorithm ends sooner.

4 Computational Results

This section includes the computational results obtained with ASBsm. The program was implemented in C++ and compiled with Microsoft Visual Studio 2008 and the tests were run on an Intel (R) Core (TM) i5-5200U CPU, with 8 Gb of random access memory. Nine data instances were selected for the tests, taken from the real data available regarding some days. Medium-size (MS) and large-size instances (LS) are respectively related to the U-line and to the system of parallel lines.

Gaps between initial solutions and final solutions were calculated, in order to get an idea of the improvements achieved. Moreover, and as optimal solutions are not known, rough Lower Bounds (LB) on the number of workstations were also obtained. For this, the processing time of each task is calculated, given the requested quantity and, as each task needs a unique machine type, the time required for each type of machine is measured. Without going into detail, since there are machines of different types all possible combinations of machine types are considered. Finally, the LB for the number of workstations is derived by summing the number of workstations required for all machine types.

The extensive company data per day, (an average 230 excel lines for the medium-size up to 1800 for the large size problems) required an algorithm for reading the data and adapting it to the format required by ASBsm. It was also necessary to adapt WIP, which is usually included in the data. The maximum size of the boxes created is 10 (pairs of components). Table 5 summarises the results. The available time depends on the line, varying from 8 h for instances $MS - 1$, $MS - 1$ and $MS - 3$ to 9 h the others.

From Table 5, it is possible to conclude that ASBsm could improve its initial solutions by 22% on average, with a minimum improvement of 9% and a

Table 5. Computational results of MS and LS instances.

Instance	N.er Model					Initial Solution				Improved Solution				Lower Bound			
	N.er Model	N.er Box	N.er Task	N.er Operator	N.er Machine	N.er Operator	N.er Machine	Objective.	Time(s)	N.er Operator	N.er Machine	Objective	Time(s)	Improvement	N.er Workstation	Objective	Gap
MS-1	15	127	15	28	41	14	14	16.14	0.15	9	9	11.14	0.58	31%	8	10.13	9%
MS-2	11	171	10	26	40	9	9	11.06	0.15	8	8	10.04	1.04	9%	7	9.04	10%
MS-3	16	159	15	28	41	12	12	13.96	0.17	9	9	10.89	2.50	22%	8	9.90	9%
MS-4	41	209	30	27	48	15	17	19.33	0.31	10	11	13.29	0.76	31%	9	11.23	16%
LS-1	303	334	70	86	146	59	61	62.81	5.10	46	48	49.80	104.49	21%	42	43.79	12%
LS-2	320	356	62	86	152	58	58	59.81	5.21	48	48	49.79	127.78	17%	42	43.78	12%
LS-3	298	335	68	86	148	64	64	65.77	4.91	45	45	46.76	131.71	29%	43	44.76	4%
LS-4	326	377	73	86	143	64	64	65.67	6.5	52	52	53.64	186.46	18%	45	46.62	13%
LS-5	344	399	69	86	148	63	63	64.81	6.8	48	48	49.80	152.97	23%	43	44.80	10%

maximum of 31%. Concerning LB, the average gap is 11%, with a minimum of 4% and a maximum gap of 16%. ASBsm is fast and can be used for real cases, whenever necessary; the running times are shown in Table 5.

It should be mentioned that some of the results were also tested by simulation to guarantee a successful implementation. However, what is more relevant from the results is the confidence acquired in the new optimisation method ASBsm to deal with the balancing problems accompanying the novel assembly systems of the company.

5 Conclusions

The Portuguese footwear industry combines tradition with cutting-edge technologies, innovative design and an extraordinary export growth. However, much more can be done to optimise some procedures of the main production phases. That is the case addressed in this paper – one of the major companies benefiting from flexible assembly systems but still managing large stitching lines, certainly supported by some automatisms and experienced operators, although without optimisation methods.

This paper specifically addresses the balancing of stitching lines and proposed a solution method – ASBsm, which integrates two heuristics: a constructive heuristic to generate an initial solution and an improvement heuristic, which is a neighbourhood search procedure that takes some inspiration from Tabu Search. The method improves the solutions, not only by local search, but also by promoting diversification by performing alterations on neighbourhoods. Some lower bounds were also considered to help evaluate the solutions. In conclusion, the balancing method has succeeded in computational tests and its results are under implementation and evaluation at the factory to avoid most manual planning.

References

- Chen, J.C., Chen, C.-C., Su, L.-H., Wu, H.-B., Sun, C.-J.: Assembly line balancing in garment industry. *Expert Syst. Appl.* **39**(11), 10073–10081 (2012)
- Özcan, U., Toklui, B.: Balancing of mixed-model two-sided assembly lines. *Comput. Ind. Eng.* **57**(1), 217–227 (2009)
- Saif, U., Guan, Z., Wang, B., Mirza, J., Huang, S.: A survey on assembly lines and its types. *Front. Mech. Eng.* **9**(2), 95–105 (2014)
- Scholl, A.: *Balancing and Sequencing of Assembly Lines*. Physica-Verlag, Heidelberg (1999)
- Scholl, A., Becker, C.: State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *Eur. J. Oper. Res.* **168**(3), 666–693 (2006)
- Tasan, S.O., Tunali, S.: A review of the current applications of genetic algorithms in assembly line balancing. *J. Intell. Manuf.* **19**(1), 49–69 (2008)