(REFEREED RESEARCH)

ON THE FLEXIBILITY CONSTRAINED LINE BALANCING PROBLEM IN LEAN MANUFACTURING

ESNEKLİK KISITLI YALIN ÜRETİM HAT DENGELEME PROBLEMİ ÜZERİNE

Arif GÜRSOY[1], Necla KIRCALI GÜRSOY[2]

[1]Ege University, Department of Mathematics, İzmir, Turkey
[2]Ege University, Aliağa Vocational School, İzmir, Turkey

Received: 28.09.2015 Accepted: 08.12.2015

ABSTRACT

In this study, a line balancing problem, which considers flexibility constraints of operations, has been presented to respond fast changing sectoral market demands in textile and apparel industry. This problem, called as the flexibility constrained line balancing problem (FCLBP) in lean manufacturing, takes into account operation flexibilities and assigns them to the available operators using the flexibility boundaries. The material of the study is the operation details that will be balanced the line in the sewing department as in (1). The method of the study is integer mathematical programming and meta-heuristic algorithm. Using the integer model, minimum idle time per operator in a given production range will be obtained and, furthermore, using the genetic algorithm, market demands will be responded in different balanced number. Lastly, using the algorithm, software has been programmed in C# to be used in the textile and apparel industry. Also, the high-efficiency balancing results will be obtained by means of the software.

Keywords: Line balancing, lean manufacturing, integer mathematical programming, genetic algorithm, software.

ÖZET

Bu çalışmada, tekstil ve konfeksiyon sektöründe hızla değişen sektörel talepleri karşılamak üzere operasyonların esneklik kısıtlarını dikkate alan bir hat dengeleme problemi sunulmuştur. Esneklik kısıtlı yalın üretim hat dengeleme problemi (FCLBP) olarak isimlendirilen bu problem, operasyonların esnekliklerini dikkate alır ve onları esneklik kısıtları çerçevesinde uygun operatörlere atar. Çalışmanın materyali, (1) de olduğu gibi, dikim bölümünde hat dengelemesi yapılacak ürünün operasyon bilgileridir. Çalışmanın metodu, tamsayılı matematiksel programlama ve meta-sezgisel algoritmadır. Bu tamsayılı model kullanılarak, verilen bir üretim miktarı aralığında operatör başına en küçük boş zaman elde edilebilecek, bunun yanında, genetik programlama tekniği kullanılarak farklı hat dengeleme durumlarıyla piyasa taleplerine cevap verilebilecektir. Son olarak, tekstil ve hazırgiyim sektörü için, bu algoritmayı temel alan, C# dilinde programlanmış bir yazılım geliştirilmiştir. Bu yazılım vasıtasıyla da, yüksek verimlilikteki hat dengeleme sonuçları elde edilebilecektir.

Anahtar Kelimeler: Hat dengeleme, yalın üretim, tamsayılı programlama, genetik algoritma, yazılım.

Corresponding Author: Arif Gürsoy, arif.gursoy@ege.edu.tr

1. INTRODUCTION

Assembly lines consist of the operators, the operations and the machines. The operators have been considered to have equally over operations and machines during the line balancing. Whereas most of operations are consecutively launched down the line and are moved from a station to another station, some operations have flexibility property, that is, they can be replaced in the flexibility boundaries. Operations are repeatedly performed at the related stations. (2)

Because of seasonal change and plenty of models, the line balancing has great importance in textile and apparel industry. Besides, the standard time of operations are generally different from each other. Also, an operator may work on multiple operations or multiple operators can work on an operation during the balancing. (3, 4)

Moreover, lean manufacturing, a method to eliminate wasting, is a production practice. Main strategy of the lean is to improve quality, cost and delivery performance while reducing the flow time. In addition, the lean production gets rid of the need to keep stocks and aims to enable the low-cost and high-quality production. (4, 5, 6)

Kara et al. intended to achieve main benefits of just-in-time production regarding line balancing and model sequencing with their proposed approach (5). Time and Space constrained assembly line balancing problem is suggested in (7) and this study has focused on the application of a procedure based on ant colonies to solve an assembly line balancing problem.

Nuriyev et al. have presented a mathematical programming for determination of optimal production quantity for minimum idle time and a case study has been illustrated in a garment industry. (8)

The line balancing problem (LBP) is a combinatorial optimization problem. The LBP is in NP-Hard class and takes exponential time to be solved optimally. (9)

Gürsoy has presented an integer mathematical programming, a heuristic algorithm having polynomial complexity, and software for flexible line balancing problem which considers flexible operations without flexibility bounds (1). After that, Gürsoy has also presented performance-based line balancing problem with a mathematical model and a greedy algorithm. (10)

Moreover, Güner et al. have showed that the line balancing is of great importance in apparel manufacturing processes having a large number of employees. (11)

In this paper, the integer programming also takes into consideration whether the non-flexible operations in the balancing process are in their order as in (1). In addition to that, the flexible operations are assigned to the work groups considering their flexibility boundaries in the integer programming. Furthermore, a meta-heuristic strategy, genetic algorithm, has been presented and new software, based on the genetic algorithm, considering flexibilities of the operations to solve the FCLBP with lean production in textile and apparel industry, has been programmed.

2. MATERIAL AND METHOD

Sewing department is one of the most important departments in textile and apparel industry and an efficient line balancing improves motivation of the workers and benefit of the establishment (12). Also, the lean production technique for the balancing has been used. In this paper, some details of the operations of a product, which are the operation names and their machine names to run, the flexibilities, the flexibility bounds and the unit times, were used as the material for the line balancing in the sewing department (Table 1).

The flexible operations assigned to the operators according the boundaries have flexibility and there can be a wide range of the flexible operations. The remaining, non-flexible, operations have to be performed to the order, that is, the non-flexible operations have a sorting among themselves so that a non-flexible operation is depended on the previous non-flexible one.

In the Flexibility Constrained Line Balancing Problem (FCLBP), for the line balancing the operations must be assigned to the operators using the flexibility boundaries so that residual idle time per operator is minimized. In this paper, the FCLBP was interpreted like the Bin Packing Problem (BPP) and an integer mathematical programming was developed to find a solution to the FCLBP as optimal. Because of the NP-Hardness of the FCLBP, the optimal solution cannot be mostly detected in acceptable time. To solve the FCLBP near optimal, a meta-heuristic algorithm was generated and useful software based on the algorithm was programmed in C# programming language.

2.1 THE 0/1 INTEGER MATHEMATICAL MODEL OF THE FLEXIBILITY CONSTRAINED LINE BALANCING PROBLEM WITH LEAN PRODUCTION

The formulation of the Bin Packing Problem (BPP) has been used to construct the mathematical model of The Flexibility Constrained Line Balancing Problem (FCLBP) with lean manufacturing (13). The Bin Packing Problem can be described as given *n* items and *n* bins. Each item must be assigned to one bin so that the total weight of the items in each bin cannot exceed the capacity (c) and the number of bins used has to be the minimum. In the FCLBP, operators instead of bins, operations instead of items and standard times instead of item weights are used. Here, operators will be mentioned as work groups which can include one, two or three operators because one-operator work time in a day cannot be enough to operate. (14)

Table 1. Details of the operations of the sample model

No	Operation Name	Unit Time (cmin.)	Machine Name	Flexibility	Flexibility start	Flexibility end
1	Sewing dart	88	Lockstitch	False		
2	Fusing interlining for pocket	10	Iron	False		
3	Fusing interlining to the facing	8	Iron	True	1	6
4	Making chain stitch to the facing and the other part with overlock	38	Overlock machine with three thread	True	2	8
5	Sewing the facing	90	Lockstitch	False		
6	Making a notch to the facing	56	Manuel	False		
7	Sewing the end of the facing and making top stitch	130	Lockstitch	False		
8	Finishing back pocket	78	Lockstitch	False		
9	Sewing back pockets bag	24	Overlock machine with five thread	False		
10	Making chain stitch to back with overlock	37	Overlock machine with five thread	False		
11	Making top stitch to back	39	Lockstitch	False		

In the FCLBP, we have n work groups and n operations (Figure 1 and Figure 2). All operators have the same work time per a day as c. Nevertheless, some operations in the line can have flexibility boundaries, namely $s_j < j < o_j$, the j^{th} operation has flexibility between operation s_j and operation o_j. (8, 14, 15)

It is known that the BPP is in NP-complete and the FCLBP can be reduced the BPP in polynomial time as above. Therefore, the FCLBP is in NP-complete class.

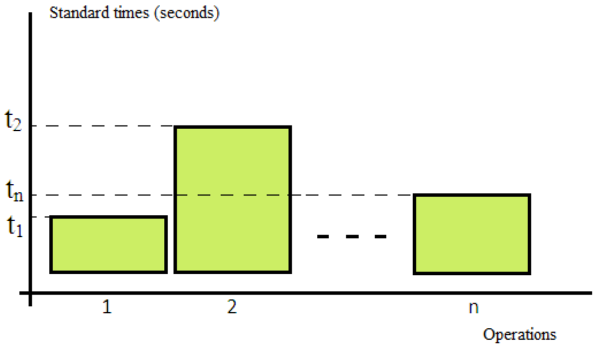

Figure 1. Operations and their standard times as seconds

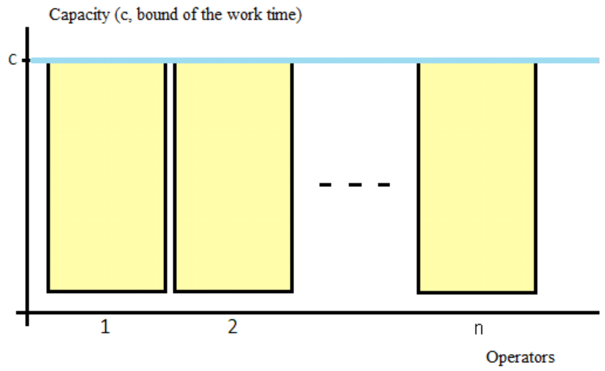

Figure 2. Operators and bound of their work time (c)

2.2 THE 0/1 INTEGER MATHEMATICAL MODEL OF THE FCLBP WITH LEAN PRODUCTION

Assume that a sample model (Table 1) to produce has a number of operations (n), including non-flexible and flexible operations having flexibility boundaries. t_j is the unit time of the j^{th} operation and each operator can just work $T + ot$ time in minute in a day, where T is daily work time and ot is daily overtime. The j^{th} operation takes $p \cdot t_j$ minutes during the process.

All operations in the process are performed at work stations and the work stations are run by the operators. For some number of products, the production time of some operations can be larger than the work time of one operator. Hence, these operations cannot be performed by an operator in the same day. In this situation, second or third operator is attended to these operations to solve the time problem. Therefore, the operator is named as the work group including one, two or three operators during the balancing. (1)

If a work group has $l\left(l \in \mathbb{Z}^+\right)$ operators, then the capacity of the related group is $l \cdot (T + ot)$ minutes, that is,

$$(l-1)\cdot(T+ot) < p \cdot t_j \le l \cdot (T+ot),\ l \in \mathbb{Z}^+ \qquad (2.1)$$

Each operation must be appointed to a certain work group and each work group can include l operators, at most. l is bounded by 3 to provide more useful and realistic programming. (1)

The FCLBP can be considered like the BPP in mathematical terms, where the operations and the work groups become the objects and the bins. Although the bins have same capacity in the BPP, the work groups having different capacity will be used in the LBP. (8)

Let m be the number of work groups, n be the number of operations and f be the number of the flexible operations:

$t_j \in \mathbb{Z}^+,\ j = \overline{1,n}$, the unit time of the j^{th} operation,

$\bar{e}_j \in \mathbb{Z}^+,\ j = \overline{1,n-f}$, the index of the j^{th} nonflexible operation,

$e_j \in \mathbb{Z}^+,\ j = \overline{1,f}$, the index of the j^{th} flexible operation,

$s_j \in \mathbb{Z}^+,\ j = \overline{1,f}$, the index of the beginning of flexibility for the j^{th} flexible operation,

$o_j \in \mathbb{Z}^+,\ j = \overline{1,f}$, the index of the end of flexibility for the j^{th} flexible operation,

$T \in \mathbb{Z}^+$, common daily work time,

$ot \in \mathbb{Z}^+ \cup \{0\}$, daily overtime,

$$x_{ij} = \begin{cases} 1, \text{ if operation } j \text{ is assigned to work group } i; \\ 0, \text{ otherwise.} \end{cases},\ i = \overline{1,m},\ j = \overline{1,n},$$

$l_i \in \mathbb{Z}^+,\ i = \overline{1,m}$, the number of operators of the i^{th} work group.

Using above notations and without loss of generality, we will suppose that $p \cdot t_j \le 3 \cdot (T + ot),\ \forall j$, and accordingly $1 \le l_i \le 3,\ \forall i$.

$$C_i = \sum_{j=1}^{n} p \cdot t_j \cdot x_{ij},\ \forall i \qquad (2.2)$$

$$\Delta C_i = l_i \cdot (T + ot) - C_i,\ \forall i \qquad (2.3)$$

$$\Delta C = \sum_{i=1}^{m} \Delta C_i \qquad (2.4)$$

$$\Delta ID = \frac{\Delta C}{\sum_{i=1}^{m} l_i} \quad (2.5)$$

Expression (2.2) gives the work time of the i^{th} work group, expression (2.3) gives the idle time of the i^{th} work group, expression (2.4) gives the total idle time of the production process, and expression (2.5) gives the idle time per operator over the process.

Using above expressions, the Integer Mathematical Program of the FCLBP with lean manufacturing, can be modelled as below:

$$Min\ \Delta ID \quad (2.6)$$

subject to

$$\sum_{j=1}^{n} p \cdot t_j \cdot x_{ij} \le l_i \cdot (T + ot),\ \forall i \quad (2.7)$$

$$\sum_{i=1}^{m} x_{ij} = 1,\ \forall j \quad (2.8)$$

$$\sum_{j=1}^{n} x_{ij} \ge 1,\ \forall i \quad (2.9)$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = n \quad (2.10)$$

$$l_i = \left\lceil \max_{\forall j}\left\{ p \cdot t_j \cdot \frac{x_{ij}}{T + ot} \right\} \right\rceil,\ \forall i \quad (2.11)$$

$$k \cdot x_{k\bar{e}_{j-1}} - i \cdot x_{i\bar{e}_j} \ge 0, i = \overline{1,m}, k = \overline{1,m}, j = \overline{1, n - f - 1}, x_{i\bar{e}_j} = 1 \quad (2.12)$$

$$i \cdot x_{ie_j} - k \cdot x_{ks_j} \ge 0, i = \overline{1,m}, k = \overline{1,m}, j = \overline{1,f}, x_{ie_j} = 1 \quad (2.13)$$

$$k \cdot x_{ko_j} - i \cdot x_{ie_j} \ge 0, i = \overline{1,m}, k = \overline{1,m}, j = \overline{1,f}, x_{ie_j} = 1 \quad (2.14)$$

$$x_{ij} = 0 \vee 1, \forall i, \forall j \quad (2.15)$$

$$l_i \in \{0,1,2,3\}, \forall i \quad (2.16)$$

The integer mathematical model of the FCLBP is (2.6-2.16), where (2.7) shows that operations assigned to the work groups can't exceed the time capacity of the related work group, (2.8) shows that each operation can just be made in one work group, (2.9) shows that each work group has to operate one operation, at least, (2.10) shows that all operations must be performed during the balancing process, (2.11) finds the number of operators of i^{th} work group, (2.12) checks whether the non-flexible operations are depended on their order, and (2.13) and (2.14) ensure that the j^{th} operation cannot be performed before operation s_j and after operation o_j.Under the constraints (2.7-2.16), the goal (2.6) is to minimize the idle time per operator.

2.3. A GENETIC ALGORITHM FOR THE FCLBP

Let m be the number of the work groups and n be the number of operations in the FCLBP. In industrial practice, a director has to generate a feasible solution in acceptable time, that is, not optimal but near optimal to satisfy the industrial demands. To solve the FCLBP with lean production in reasonable time and satisfy the demands, a metaheuristic algorithm has been constructed.

Input parameters:

p, production quantity

c, work time per a day of one operator (minute)

m, the number of work groups

n, the number of operations

f, the number of flexible operations ($1 \le f \le n$),

$t_j \in \mathbb{Z}^+,\ j = \overline{1,n}$, the unit time of the j^{th} operation,

$\bar{e}_j \in \mathbb{Z}^+,\ j = \overline{1, n - f}$, the index of the j^{th} nonflexible operation,

$e_j \in \mathbb{Z}^+,\ j = \overline{1, f}$, the index of the j^{th} flexible operation,

$s_j \in \mathbb{Z}^+,\ j = \overline{1, f}$, the index of the beginning of flexibility for the j^{th} flexible operation,

$o_j \in \mathbb{Z}^+,\ j = \overline{1, f}$, the index of the end of flexibility for the j^{th} flexible operation,

$T \in \mathbb{Z}^+$, common daily work time,

$ot \in \mathbb{Z}^+ \cup \{0\}$, daily overtime.

These input parameters are used in *CreateIndividual()* subprocedure and *GeneticFCLBP()* main procedure.

***CreateIndividual()*:**

S1. List non-flexible operation as consecutively.

S2. *for j=1 to (n-f) do*

S3. Insert the j^{th} flexible operation (e_j) to the open interval (s_j, o_j) randomly.

S4. In the new individual, calculate $p \cdot t_j,\ j = \overline{1,n}$.

S5. Determine the number of operators using Next-Fit algorithm and set as fitness value.

***GeneticFCLBP()*:**

S1. *for j=1 to n do*

S2. *CreateIndividual()*

S3. Insert new individual to the population pool according fitness value non-decreasingly.

S4. *crosn=0,9*n; mutn=0,1*n*

*S5. until (*best fitness value was repeated *n* times*)*

S6. *for i=1 to crosn do*

S7. Select two individuals from the population based on the roulette wheel

S8. Crossover the selected individuals

S9. If both of offsprings don't satisfy the flexibility boundaries then goto S7

S10. For the offsprings satisfying the flexibility boundaries, run the Next-Fit algorithm, set the fitness values, and insert them to the population non-decreasingly.

S11. *for i=1 to mutn do*

S12. Select an individual from the population based on the roulette wheel.

S13. Determine a flexible operation (gene) from the individual randomly.

S14. Replace the gene randomly under its flexibility boundaries.

S15. For the offspring, run the Next-Fit algorithm, set the fitness value, and insert to the population non-decreasingly.

S16. Print the first individual of the population.

CreateIndividual subprocedure having $O(n)$ complexity creates offsprings using Next-fit algorithm from BPP. *GeneticFCLBP* having crossover and mutation operands, also, continues to produce new generations until repeated n times the best fitness value.

3. SOFTWARE AND COMPUTATIONAL EXPERIMENTS

This software designed in C# uses the above genetic algorithm having crossover and mutation operands. The software lists all solutions in the given interval of number of productions where lower bound of production quantity is *P_low* and the upper bound is *P_up*. In the software, it is needed to common daily work time in minute (*T*), daily overtime in minute (*ot*), lower bound of efficiency (*e_low*) as standard model input and population size (*Pop_size*), mutation and crossover rates (*Mut_rate*, *Cross_rate*) as genetic algorithm input (Figure 1). In the software results, all available numbers of productions (*p*) with the total idle time $\left(\sum id\right)$, the total work time $\left(\sum C_i\right)$, the total numbers of operators $\left(\sum l_i\right)$, the work time per operator $\left(\sum C_i / \sum l_i\right)$, and efficiency (ef) as percent, are listed. (1)

Table 2. Available solutions, higher than 90% efficiency, based on Table 1

p	$\sum id$ (min)	$\sum C_i$ (min)	$\sum l_i$	$\sum C_i / \sum l_i$ (min)	ef %
500	250	2990	6	498,33	92%
501	242	2994	6	499,00	92%
502	236	3000	6	500,00	93%
503	230	3006	6	501,00	93%
504	224	3012	6	502,00	93%
505	218	3018	6	503,00	93%
506	212	3024	6	504,00	93%
507	206	3030	6	505,00	94%
508	200	3036	6	506,00	94%
509	194	3042	6	507,00	94%
510	189	3047	6	507,83	94%
511	183	3054	6	509,00	94%
512	177	3059	6	509,83	94%
513	170	3066	6	511,00	95%
514	164	3073	6	512,17	95%
515	158	3078	6	513,00	95%
516	152	3084	6	514,00	95%
517	146	3090	6	515,00	95%
518	140	3096	6	516,00	96%
519	134	3102	6	517,00	96%
728	505	4350	9	483,33	90%
729	498	4357	9	484,11	90%
730	493	4363	9	484,78	90%
731	485	4370	9	485,56	90%
732	480	4375	9	486,11	90%
733	475	4380	9	486,67	90%
734	468	4387	9	487,44	90%
735	462	4393	9	488,11	90%
736	456	4399	9	488,78	91%
737	449	4406	9	489,56	91%
738	445	4410	9	490,00	91%
739	438	4417	9	490,78	91%

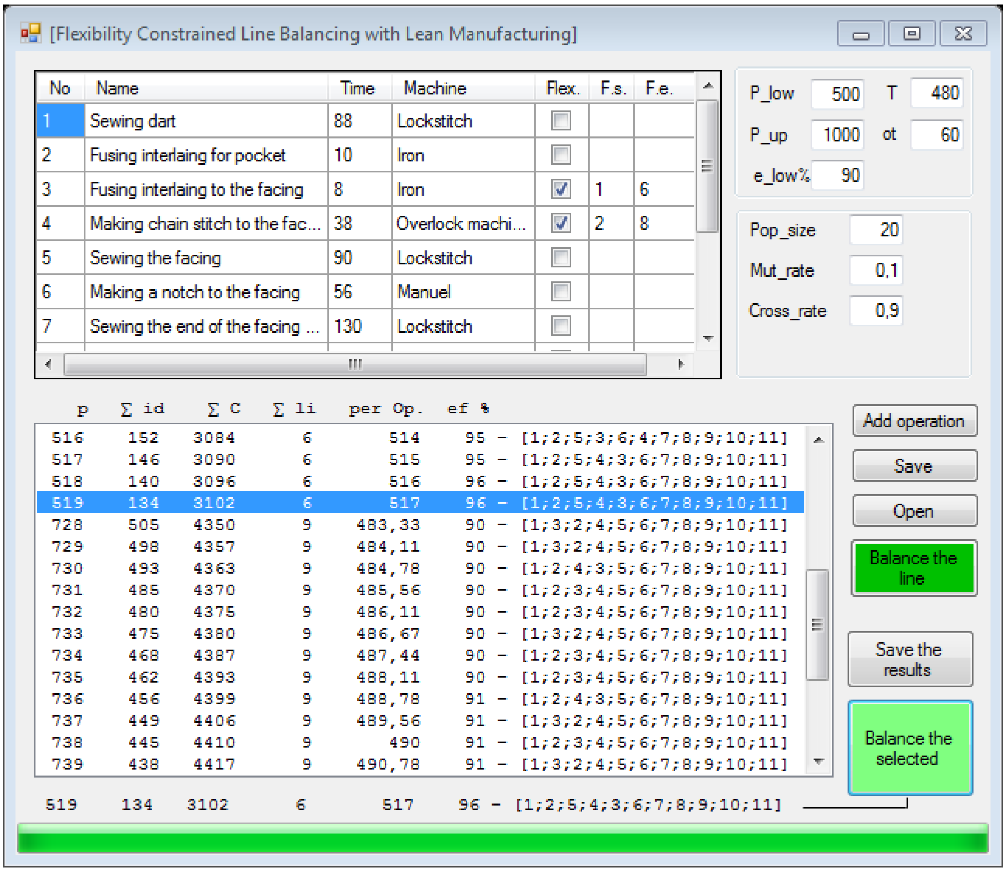

Figure 3. The Software of the FCLBP with Lean Production

One of the best efficiencies is %96 and 519 productions in Table 2 and average work time 517 minutes per operator along the process (Table 3 and Table 4). Owing to the large-scale sectoral market demands, balancing with small quantities cannot generally satisfy them. Moreover, using Table 2, *p=739* having %91 can be acceptable among different feasible solutions in Table 2.

Table 3. The balancing for *p=519*

p	$\sum id$	$\sum C_i$	$\sum l_i$	$\sum C_i / \sum l_i$	ef %
519	134	3102	6	517	96%

Table 4. The detailed balancing results for *p=519*

No	l_i	Operations	C_i	Idle	ef %
1	1	1; 2	508	32	94%
2	1	5	467	73	86%
3	1	4; 3; 6	529	11	98%
4	2	7; 8	1079	1	99%
5	1	9; 10; 11	519	21	96%

4. CONCLUSION

In this paper, the flexibility constrained line balancing problem with lean production, which is important to improve quality and to reduce transaction costs in the textile and apparel industry and, has flexibility bounds for flexible operations, is considered. An integer mathematical model has been presented whose goal is to find the minimum idle time per operator. Then, a mate heuristic algorithm has been presented for the FCLBP with lean production which is in NP-Hard class. The algorithm balances the operations in the model to the work groups and finds the minimum number of operators using genetic algorithm strategy. Finally, the software which solves the FCLBP with lean production based on the genetic algorithm has been programmed. The software coded in C# finds the minimum idle time per operator and lists all results for the calculated number of productions (Figure 1).

As a consequence, the integer mathematical programming, the genetic algorithm and the software may be used to balance the assembly lines in various industries which contain different operations and have flexibility boundaries. Moreover, the software can list the results which are above the given efficiency. Using the software, a user can readily select any number of productions from the list and balance the line in textile and apparel industry as in Figure 3, Table 2, Table 3 and Table 4.

As the future work, new algorithms, software and mathematical models will be created to solve the performance based line balancing problem with lean production in textile and apparel industry.

REFERENCES

1. Gürsoy, A., 2012, "An Integer Model and a Heuristic Algorithm for the Flexible Line Balancing Problem", *Tekstil ve Konfeksiyon*, Vol. 22 (1), 58-63
2. Becker, C., Scholl, A., 2006, "A survey on problems and methods in generalized assembly line balancing", *European Journal of Operational Research*, Vol. 168 (3), 694-715.
3. Güner, M., 2001, "Konfeksiyon İşletmelerinde Dikim Hattının Dengelenmesi", *Konfeksiyon ve Teknik*, Temmuz.
4. Holweg, M., 2007, "The genealogy of lean production", *Journal of Operations Management*, Vol. 25 (2), 420–437.
5. Kara Y., Ozcan U., Peker A., 2007, "Balancing and sequencing mixed-model just-in-time U-lines with multiple objectives", *Applied Mathematics and Computation*, Vol. 184 (2), 566-588.
6. Keytack H., 1997, "Expert Line Balancing System (ELBS)", *Computers & Industrial Engineering*, Vol. 33 (3/4), 303-306.
7. Bautista J., Pereira J., 2007, "Ant algorithms for a time and space constrained assembly line balancing problem", *European Journal of Operational Research*, Vol. 17 (3), 2016-2032.
8. Nuriyev U., Guner M., Gursoy A., 2009, "A model for determination of optimal production quantity for minimum idle time: a case study in agarment industry", *Tekstil*, Vol. 58 (5), 214-220.
9. Garey, M.R., Johnson, D.S, 1979, *Computers and Intractability*, W. H. Freeman and Company, San Francisco.
10. Gürsoy, A., 2012, "Performance-Based Line Balancing Problem: A New Model and A Greedy Algorithm", *Ege University Journal of the Faculty of Science*, Vol. 36 (1-2), 44-53
11. Güner, M., Yücel, Ö., Ünal, C., 2013, "Applicability of Different Line Balancing Methods in the Production of Apparel", *Tekstil ve Konfeksiyon*, Vol. 23 (1), 77-84
12. Yücel, Ö., Güner, M., 2008, "Analyzing the factors affecting garment sewing times", *Tekstil ve Konfeksiyon*, Vol. 18 (1), 41-48.
13. Martello, S., Toth, P., 1990, *Knapsack Problems, Algorithms and Computer implementations*, England, Chichester: John Wiley & Sons.
14. Gürsoy A., Nuriyev U.G., 2010, "A Mathematical Model For Assembly Line Balancing Problem With Lean Manufacturing", *Proceeding of the third International Conference Problems of Cybernetics and Informatics, PCI'2010*, Vol. III, 156-159
15. Nuriyev, U.G., Güner, M., Berberler, M.E., Gürsoy, A., 2008, "Optimum Line Balancing of a Complex Production Process with Consecutive and Parallel Operations in Textile", *International Conference on Control and Optimization with Industrial Applications*, p.142