(REFEREED RESEARCH)

MINIMIZING MACHINE CHANGEOVER TIME IN PRODUCT LINE IN AN APPAREL INDUSTRY

HAZIR GİYİM SANAYİNDE ÜRETİM HATLARINDA MAKİNE DEĞİŞİM SÜRESİNİN EN AZA İNDİRİLMESİ

Aykut KENTLİ[1], Vedat DAL[2], Ali Fuat ALKAYA[3]

[1] Marmara University, Department of Mechanical Engineering, İstanbul, Turkey
[2] Marmara University, Department of Textile Engineering, İstanbul, Turkey
[3] Marmara University, Department of Computer Engineering, İstanbul, Turkey

Received: 18.01.2013 Accepted: 11.02.2013

ABSTRACT

This study deals with a scheduling problem with machine changeovers in a apparel company and presents a heuristic to minimize the total setup time subject to machine changeovers in models ordered. Commonly used heuristics such as shortest processing time and earliest due date can be used to calculate a feasible schedule quickly, but usually do not produce schedules that are close to optimal in these environments. A solution approach for the problem is developed by dividing it into two subproblems. Solution of the first problem is given as an input for the second problem. In this process, an originally developed heuristic is applied for the first problem and the second problem is formulated as an open and asymmetric traveling salesman problem. The second problem is solved by Genetic (GA) and Simulated Annealing (SA) algorithms. The experimental results demonstrate the effectiveness of the proposed algorithm to solve the scheduling problem with machine changeovers.

Key Words: Apparel Industry, Combinatorial optimization, Changeover time, SA, GA.

ÖZET

Bu çalışmada bir hazır giyim firması için makine değişimlerinin düzenleyen çizelgeleme sorunu ele alınmıştır ve sipariş edilen ürünlerin üretimi için gerekli makine değişimleri dikkate alarak, toplam kurulum süresini en aza indirmek için bir sezgisel metot kullanılmıştır. En kısa işlem süresi ve en erken teslim tarihi gibi sık kullanılan sezgisel metotlar ile geçerliliği olan bir çizelgeleme için kullanılabilir, ancak genellikle bu durumda en iyi sonuca yakın çizelgeler üretilemez. Bu çalışmada sorunu iki alt problem bölerek bir çözüm yöntemi geliştirdik. İlk problemin çözüm sonuçları ikincisinin giriş verileri olarak kullanılmıştır. Bu süreçte, birinci problemin çözümünde yeni geliştirilmiş bir sezgisel metot kullanıldı. İkinci problem açık ve asimetrik bir seyyar satıcı problem olarak formüle edildi. İkinci problem Genetik (GA) ve Benzetilmiş Tavlama (BT) algoritmaları kullanılarak çözüldü. Deney sonuçları önerilen algoritmanın makine değişimlerini göz önüne alan çizelgeleme sorununun çözümünde etkin olduğunu göstermiştir.

Anahtar Kelimeler: Hazır Giyim Sanayi, Kombinatoryal optimizasyon, Değişim zamanı, BT, GA.

[*] **Corresponding Author:** Vedat Dal, vedat@marmara.edu.tr, Tel:+90 216 336 57 70, Fax:+90 216 337 89 87

1. INTRODUCTION

The present garment industry is experiencing an increase in product variety. In other words, the companies produce variety of models rather than constructing single model as before. Currently, the companies have been planting at least 50 models throughout the same product group. Although 5000 product was ordered for any model formerly, nowadays, products for different models at low quantity, such as 100-1000 pieces, have been ordered due to an increase in variety of models. Namely, even though there has been an increase in variety of models, the quantity of orders has decreased that triggered the necessity of having flexible production lines. Even one has to produce different models within the same day in the same production line.

Accordingly, the time of the model variation becomes important when different models are to be produced since either its tool or setting changes, an absolute change is occurred eventually in any way, when it switches from one model to another one. As long as the period of model change extends, the efficiency of firm falls down as the change is time-consuming. Contingently, there is another loss of efficiency as a result of change: the employee's ability is also reduced throughout switching from a working model to another one. Because the employee is doing a task and specializes on it and after a while, becomes swift. Afterwards, as the model is changed, the one has to perform a new task on the new model. Therefore, a loss of efficiency is occurred until the one adapts to the new task. This situation is a major problem for companies that should be solved as the change cannot be prevented.

In this study, the firm solved this problem by both analysing the models ordered and checking which model will enter into which production line formerly. It is noticed that there is a gap in the literature on this subject. One of the reasons is that this is an emerging problem which revealed along last four years. Accordingly, the concept of lean production has already been widespread in the last 4-5 years in Turkey. Also, there is a new trend in Turkey: Fast Fashion. European companies give orders to Chinese ones and the products are reaching within 120 days; however, the product is available in stores within 21 days when they are ordered to Turkish companies.

Since Turkey is closer to the European market, this kind of production has been turned to Turkey. Turkey has to respond it immediately. Therefore, it is necessary to ensure flexible manufacturing production lines and to optimize them. Those who solved this problem get high income. There are a few attempts, however, there is very few study on pre-optimizing logic in literature. A solution heuristic is developed and used in order to optimize the production throughput in this study. This study provides the logic that is getting orders of models, optimizing their production lines and predefining which products will go which production line before production begins.

In the next section, a brief literature survey about related problems is given. The problem is described in Section three. In section four, solution methodology developed for the problem are given in detail. Section five explains and gives the related data of case study Section six presents the results. Section seven includes conclusive remarks as well as future works.

2. LITERATURE REVIEW

Scheduling multi machine multi-product assembly systems is necessary for the apparel industry due to rapid market changes. A company produces a number of different types of products. During production, it must be processed in several machines sequenced as an assembly line. After production of a product type is finished, the production line must be reorganized for producing the next type of product. This organization problem is a job shop scheduling problem. Normally, managers plan the work flow on shop floor according to their experience on the system. But, they could be limited by that the assembly line become larger and many operations should be scheduled. As a result, implementing an optimization method to solve the job shop scheduling problem effectively becomes significant.

The history of the job shop scheduling problem can be traced back to more than 40 years ago (1-2). Many researchers have formulated different job shop scheduling models. GAs (3), dispatching rules (4), and simulation (5) models have been applied to job shop scheduling problem having sequence dependent setups (SDS). More studies could be found in the study of Zhu and Wilhelm (6). They reviewed the literature related to the class of scheduling problems that involve sequence-dependent setup times. Allahverdi et.al presented a survey of more than 300 studies for the scheduling problem (7). In recent years, evolutionary algorithms have been extensively applied, such as, simulated annealing (SA) method (8-10) and genetic algorithm (GA) (11-14).

Several studies about production scheduling in the apparel industry have been published. Bowers et.al proposed a 3-tiered (long-term, short-term, and daily planning tasks) hierarchical production planning and scheduling model (15). Kwong et.al used genetic algorithms and fuzzy-set theory to generate fault-tolerant fabric-cutting schedules in a just-in-time production environment (16). Chan et.al used a GA to solve an assembly line balancing problem in the apparel industry (17). Tucci and Rinaldi presented the application of the tabu search metaheuristic to production scheduling in weaving (18). Wong et.al also developed a GA to balance an apparel assembly line of UPS and investigated the impact of different level of skill inventory on the assembly makespan (19). Guo et.al constructed a universal mathematical model of the job shop scheduling problem for apparel assembly process is constructed. A genetic optimization process is presented to solve this model (20). Karabuk involved in the yarn production planning problem. He developed a stochastic programming model that explicitly includes uncertainty in the form of discrete demand scenarios (21). Alkaya has solved a sequence dependent job shop scheduling problem, similar to our but for a smaller case, by using GA algorithm (22). Chen has dealt with a scheduling problem with machine maintenance in a textile company. He presented a heuristic to minimize the completion time or subject to periodic maintenance and due dates. He compared the performance of heuristic with of the branch-and-bound algorithm (23). Hsu et al. presented a scheduling approach for yarn-dyed textile manufacturing. They formulated as a mixed integer programming (MIP) model, solved by a GA (24).

However, the above research focused on solving a particular problem using a specific method in a well-defined environment with various constraints, a simplified method concerning machine changeovers in the apparel industry has attracted little attention. Furthermore, most of the studies for the problem in the apparel industry used GA methodology and to best our knowledge, no one has used SA. In this study, in order to solve this multi-machine multi-product scheduling problem, SA algorithm is also developed beneath GA to compare the efficiencies. The algorithms are then used to solve the real-world job shop scheduling problem.

3. PROBLEM DEFINITION

Assembly forms for mass production of apparel manufacture are composed of a certain number of sewing workstations. Figure 1 show a sample assembly form.

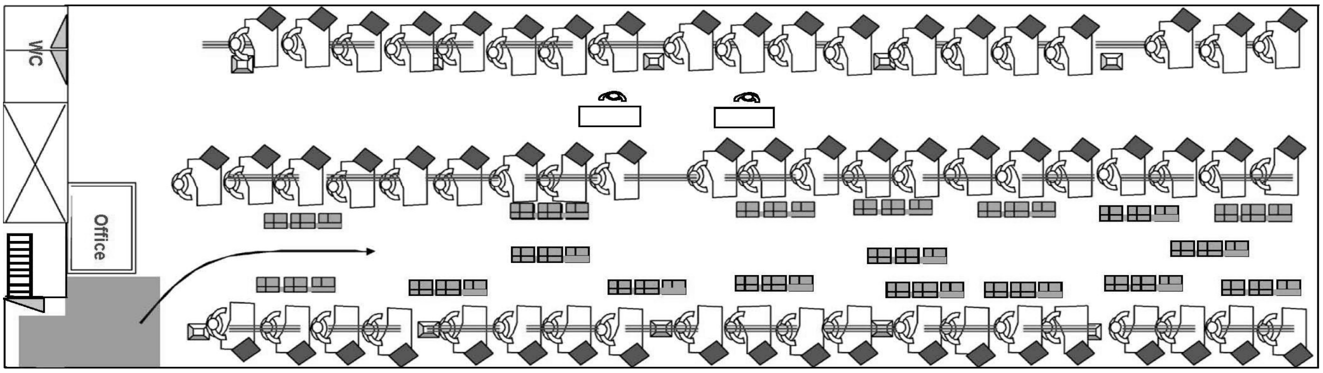

Figure 1. Sample layout of machine in an apparel production company

The assembly process of each garment consists of multiple operations. During production, it must be processed in several machines sequenced as an assembly line. After production of a product type is finished, the production line must be reorganized for producing the next type of product. Setup cost for a production line is linearly proportional with the number of machines moved while it is being built. During the reorganization of a production line, moving a machine requires a certain amount of time and movements of machines cannot be processed at parallel. The required number of machines of each machine type for a product is known beforehand. If production of next product type requires more number of machines of any type than current production setup, then the necessary amount of required type of machines can be brought from the depot within the same amount of time. The number of machines for each machine type is available as much as needed in the depot.

The problem under consideration is to find a sequence that minimizes the total completion time subject to machine changeovers. Finding exact solutions requires extensive amount of time for such problem instances. For this reason, a common approach for solving these problems is developing heuristic methods. In heuristic methods, the guarantee of finding optimal solutions is sacrificed for the sake of getting good solutions in a significantly reduced amount of time. Commonly used heuristics such as shortest processing time (SPT) and earliest due date (EDD) can quickly create feasible solutions, but as problem complexity increases the solutions may be far from optimal.

In conclusion, the problem could be defined as follows

3.1 Objective

In the apparel industry, meeting promised delivery dates, or due dates, is the most desirable objective which management wants to achieve. It is described that a current sequence dependent job shop scheduling problem with machine changeovers. The objective of this model is implemented by selecting appropriate production starting time of each order and generating optimal operation assignment, which is equivalent to minimize the makespan.

3.2 Constraints

The actual apparel production has many irregular characteristics and must be subject to some constraints. Time constraints define the due dates of the orders. Operation precedence constraints state that an operation cannot be started before its preceding operation is completed and it is transported to the corresponding machine. In this study, the following characteristics are assumed:

- Once an operation is started, it cannot be interrupted.
- There is no shortage of materials, machine breakdown and absence of operators in the job shop.
- The job shop used for modelling is empty initially, in other words, there is no WIP in each workstation.

For solving the problem mentioned above, following solution methodology is developed.

4. SOLUTION METHODOLOGY

The mentioned scheduling problem is decomposed into a sequence of sub-problems. In solution of first subproblem, setup costs are calculated by using our heuristic and a distance matrix is obtained. In solution of the second problem, this matrix is used to define the problem as a Traveling Salesman Problem (TSP). In other words, the minimization of total production time problem is divided into subproblems of constructing distance matrix and finding minimum distance. Solution methodology used is explained in detail at the following subsections.

4.1 Constructing Distance Matrix

This subproblem is determining the minimum number of machine changes between each pair of production lines. It is called as minimization of machine changes throughout the paper.

Let us explain the solution approach through an example. Table 1 gives the sequence of operations that are required for producing two product types: 103333 and 103334. If the product 103333 is to be produced then an assembly line is set up with the machines listed in second column of Table 1 (explanation of abbreviations is given in Table 3). If product 103334 is to be produced after the production of 103333, then the production line should be redesigned as given in the third column of the table. In other words, production line 103333 should be converted into 103334 in possible minimum time. For building the new line, moving a machine increases the setup time. Thus, for converting a production line into another one, a smart method should make use of the former line as much as possible. This idea is designed as a solution method. It is assumed that the necessary amount of machines of type DDM can be brought from the depot within the same amount of time if production of next type of product requires more number of machines of any type than current production setup.

Table 1. Production line data for two sample product types

Operation No	103333	103334
1	IRON	DDM
2	RCM	RCM
3	4IOM	IRON
4	4IOM	RCM
5	RCM	4IOM
6	4IOM	4IOM
7	RCM	RCM
8	DDM	RCM
9	RCM	DDM
10	DDM	RCM
11	RCM	
12	DDM	
13	RCM	

In this study, a deterministic solution heuristic for minimizing the machine changes between production lines is proposed. The proposed heuristic is called Find Maximum Match (FMM). The FMM is explained as a pseudo code in Figure 2.

Using FMM, all combinations between each product are tested and obtained minimum values of machine changeovers are saved in a distance matrix. This matrix will represent the setup costs of machine changeover and it will be used in solving second subproblem.

4.2 Finding Minimum Distance

Once minimization of machine changes problem is solved efficiently, the solution is ready to be used as an input for the second subproblem. This subproblem is finding an optimum sequence of production lines. That is, to decide on producing which product type after another. The problem of prescribing a sequence, even for a single machine with SDS with makespan as the objective, is equivalent to the Traveling Salesman Problem (TSP) and is therefore NP-hard (25). Actually, it turns out to be a variant of the Traveling Salesman Problem (TSP). TSP or its variants can be applied to solve many realistic problems within our daily lives. For a complete survey and summary about TSP, it is recommended to look up (26).

If production lines are considered as cities and setup costs as the distance between cities, then it turns out to be a TSP. However, it is an Open Traveling Salesman Problem (OTSP) because it is needed not to build the first production line after all of the product types are produced. On the other hand, it is an Asymmetric Traveling Salesman Problem (ATSP) because setup costs between two production lines is based on the order of production. Therefore, our second problem is an Open and Asymmetric TSP. For solving this subproblem, two meta-heuristics are implemented: Genetic Algorithm and Simulated Annealing.

4.2.1 Genetic algorithm

Genetic algorithms are a population based method inspired by the principles of natural evolution (27). The algorithm starts the search with a population of individual chromosomes (solutions) generated randomly or heuristically (28).

Classical GA operators such as crossover and mutation may not work efficiently for TSP. To describe the tour of *n*-cities, three modes of string representation, namely adjacency representation, ordinal representation and path representation are available. A comparison of these techniques was reported in the study of Michalewicz (29). Path representation is adopted. Based on path representation, a tour sequence is denoted by a string comprising the identification numbers of cities. For example, the traveling salesman has visited cities 5, 7, 4, 3, 1, 2 and 6, his tour can be described as (5 7 4 3 1 2 6). The fitness of this solution is the total cost of travel on this path (30), (31).

- Assume that an assembly line A requires n number of machines and assembly line B requires m number of machines.
- Assume that assembly line A is set up after assembly line B.
- Place assembly line A and B as two rows where B is the top row and A is the bottom row in the way that last element of A corresponds to the first element of B.
- mnom = 0 //mnom is the maximum number of matches
- nom = 0 //nom is the number of mathces
- for m + n - 2 times
    - slide the line A by one step to its right and calculate number of matches (nom) with B
    - if nom > mnom
        - mnom = nom
        - save this position
- return mnom and position

Figure 2. Pseudo code of FMM heuristic

Table 2. A sample run of Find Maximum Match (FMM) Heuristic

Matches																
…	…															
3	103333	IRON	**RCM**	4IOM	4IOM	RCM	**4IOM**	**RCM**	DDM	RCM	DDM	RCM	DDM	RCM		
	103334	DDM	**RCM**	IRON	RCM	4IOM	**4IOM**	**RCM**	RCM	DDM	RCM					
5	103333	IRON	RCM	4IOM	4IOM	**RCM**	**4IOM**	RCM	DDM	**RCM**	**DDM**	**RCM**	DDM	RCM		
	103334		DDM	RCM	IRON	**RCM**	**4IOM**	4IOM	RCM	**RCM**	**DDM**	**RCM**				
1	103333	IRON	RCM	4IOM	4IOM	RCM	4IOM	RCM	DDM	**RCM**	DDM	RCM	DDM	RCM		
	103334			DDM	RCM	IRON	RCM	4IOM	4IOM	**RCM**	RCM	DDM	RCM			
5	103333	IRON	RCM	4IOM	4IOM	**RCM**	4IOM	**RCM**	DDM	RCM	DDM	**RCM**	**DDM**	**RCM**		
	103334				DDM	**RCM**	IRON	**RCM**	4IOM	4IOM	RCM	**RCM**	**DDM**	**RCM**		
1	103333	IRON	RCM	4IOM	4IOM	RCM	4IOM	RCM	DDM	RCM	DDM	**RCM**	DDM	RCM		
	103334					DDM	RCM	IRON	RCM	4IOM	4IOM	**RCM**	RCM	DDM	RCM	
3	103333	IRON	RCM	4IOM	4IOM	RCM	4IOM	**RCM**	DDM	**RCM**	DDM	RCM	DDM	**RCM**		
	103334						DDM	**RCM**	IRON	**RCM**	4IOM	4IOM	RCM	**RCM**	DDM	RCM
…	…															

Mutation is applied to each child individually after crossover. The mutation operator incorporated in this study simply chooses two genes at random and swaps their position in the chromosome. Mutation probability used in this study is 0.05. Roulette wheel selection method is preferred in the implementation. The pseudo code for our GA implementation is given in Figure 3.

The cost matrix representing the setup costs between production lines is given as input to the GA. Let the size of matrix (number of production lines) be *n* for a generic representation. In the genetic algorithm, firstly initial solutions are produced. The initial population mainly consists of individuals that are constructed by using the Nearest Neighbour Heuristic. Using each point as a starting point will result in *n* different solutions. Assuming *noi (population size)* >*n*, the rest *noi-n* solutions will be generated randomly.

In our implementation, *noi/2* offspring are produced from the current generation by using order-based crossover and are added to the population set (*ps*) after being mutated. Then, best *noi* solutions are chosen for constituting the next generation. Parameters of this algorithm are *noi* and *nog,* which represents how many generations are built within the algorithm and parameter *nog* is used as the termination condition. Their values are explicitly given in the Results and Discussion section.

4.2.2 Simulated annealing

Second method exploited for solving the second subproblem is Simulated Annealing (SA). SA is commonly said to be the oldest among the meta-heuristics and surely one of the first algorithms that had an explicit strategy to escape from local minima. The fundamental idea is to allow moves resulting in solutions of worse quality than the current solution (uphill moves) in order to escape from local minima. The probability of doing such a move is decreased during the search (32).

5. CASE STUDY

In this section, the solution methodology is applied to a case study in which a textile manufacturing company has faced. The company studied is a medium-sized company among the largest international ready-to-wear companies, producing garments made of knitted fabrics. The company produces garments within six production lines. Company managers plan the work flow of the production line for models. In this study, the production information belonging to the models of 30 different T-shirt required to produce within one month of operation is used. One employee is responsible for each machine. Employee and machine shifts together in model changes. The key feature of employees is being multi-skilled. Multi-skills means that an employee is able work in more than one machine and operation. Type of machinery used in and work flow of production lines are given below. Explanation of abbreviations used is given in Table 3.

In this example, there are 30 different product types to be produced. The required machines for the production of each product type are given in Table 4.

- generate noi solutions and put them in a population set, ps
- while termination condition is not satisfied
  - for noi/4 times
    - select two solutions from ps, s1 and s2
    - determine two crossover sites randomly
    - apply order-based crossover to s1 and s2 by using the crossover sites and call the new offspring as s3 and s4
    - mutate s3 and s4
    - add s3 and s4 into ps
  - end for
  - remove worst noi/2 solutions from ps
- end while

Figure 3. GA algorithm implemented for solving the second subproblem

Table 3. Abbreviation of machine used in the case

DDM	Lockstitch machine
Manual	Manual
4İOM	4 Thread overlock sewing machine
CITMAK	Metal button machine
RCM	Coverstitch sewing machine
ILMAK	Buttonholing machine
PNTRZ	Bartacking Machine
IRON	Iron
BANT	Binding sewing machine
KRYK	Three Needles Coverstitch Sewing Machine
DUGME	Button sewing machine
BRM	Top and Bottom Coverstitch Machine
KM	Flat-bed Coverstitch machine
BM	Piping Machine
1. SAH	Three Needles Coverstitch over waist-band
2. SAH	Coverstitch (Les)
ZINCIR	Chain Stitch Machine
GER	Coverstitch (side seam) machine
ZKZK	Zigzag stitch machine
OT	Belt machine
EL	Scissors Cutting

Table 4. Production Line Data For 30 Different Product Types

Operation No	63646	103332	103333	103334	103337	103361	103470	107862	107863	237714	263137	1971248	1972098	AEWZ8063	AEWZ8070	AEWZ8080	AEWZ8083	AR211047	AR211060	AR211067	ARHSH20	ARHSH201	B68373TENI	BANTBALAS	CH103361	MNSSHEBA	POLO107861	POLO107862	POLO107863	POLO107864
1	DDM	IRON	IRON	DDM	4IOM	4IOM	RCM	4IOM	4IOM	DDM	DDM	4IOM	4IOM	2.SAH	4IOM	4IOM	4IOM	IRON	MANUAL	4IOM	4IOM	4IOM	IRON	4IOM	4IOM	DDM	4IOM	IRON	4IOM	4IOM
2	MANUAL	RCM	RCM	RCM	4IOM	RCM	4IOM	4IOM	DDM	DDM	MANUAL	4IOM	4IOM	4IOM	4IOM	4IOM	4IOM	IRON	DDM	ZKZK	DDM	DDM	4IOM	RCM	RCM	DDM	4IOM	MANUAL	DDM	DDM
3	4IOM	4IOM	4IOM	IRON	4IOM	4IOM	DDM	4IOM	RCM	DDM	DDM	DDM	4IOM	4IOM	1.SAH	1.SAH	1.SAH	MANUAL	4IOM	4IOM	EL	EL	DDM	4IOM	4IOM	DDM	DDM	DDM	RCM	RCM
4	DDM	4IOM	4IOM	RCM	RCM	RCM	4IOM	RCM	4IOM	4IOM	KM	DDM	RCM	GER	MANUAL	MANUAL	MANUAL	DDM	DDM	ZKZK	DDM	DDM	4IOM	RCM	RCM	DDM	4IOM	DDM	4IOM	4IOM
5	MANUAL	RCM	RCM	4IOM	4IOM	DDM	KRYK	4IOM	IRON	DDM	DDM	4IOM	DDM	2.SAH	4IOM	GER	GER	DDM	4IOM	IRON	DDM	DDM	4IOM	DDM	DDM	DDM	RCM	DDM	IRON	IRON
6	DDM	4IOM	4IOM	4IOM	ILMAK	RCM	RCM	IRON	IRON	4IOM	DDM	DDM	4IOM	BM	1.SAH	MANUAL	MANUAL	MANUAL	ZKZK	4IOM	4IOM	4IOM	DDM	RCM	RCM	DDM	4IOM	DDM	IRON	IRON
7	DDM	RCM	RCM	RCM	DDM	DDM	4IOM	DDM	DDM	DDM	4IOM	4IOM	DDM	4IOM	4IOM	2.SAH	DDM	DDM	4IOM	DDM	4IOM	4IOM	MANUAL	DDM	DDM	4IOM	4IOM	DDM	DDM	DDM
8	MANUAL	DDM	DDM	RCM	4IOM	4IOM	RCM	4IOM	DDM	DDM	KM	4IOM	RCM	DDM	BM	IRON	4IOM	MANUAL	ZKZK	4IOM	RCM	RCM	DDM	4IOM	4IOM	RCM	RCM	DDM	DDM	DDM
9	4IOM	4IOM	RCM	DDM	DDM	DDM	4IOM	DDM	RCM	DDM	4IOM	RCM	DDM	4IOM	4IOM	ILMAK	IRON	DDM	4IOM	RCM	4IOM	4IOM	MANUAL	DDM	DDM	DDM	RCM	4IOM	RCM	RCM
10	DDM	4IOM	DDM	RCM	RCM	RCM	DDM	DDM	MANUAL	4IOM	KM	RCM		DDM	BM	DDM	ILMAK	4IOM	4IOM	DDM	RCM	RCM	4IOM	RCM	RCM	IRON	DDM	4IOM	MANUAL	MANUAL
11	CITMAK	MANUAL	RCM		RCM	RCM	4IOM	DDM	MANUAL	MANUAL	BRM	DDM		PNTRZ	4IOM	ZINCIR	OT	4IOM	ZKZK	DDM	RCM	RCM	DDM	RCM	RCM	DDM	DDM	MANUAL	MANUAL	MANUAL
12	DDM	RCM	DDM		RCM	DDM	DDM	RCM	RCM	BM	4IOM				DDM	4IOM	DDM	ZKZK	IRON	DDM	DDM	DDM	DDM	DDM	DDM	4IOM	DDM	ILMAK	RCM	RCM
13	DDM	DDM	RCM		MANUAL	4IOM	BANT	RCM	DDM	DDM	MANUAL				DDM	IRON	4IOM	4IOM	4IOM	4IOM	IRON	IRON	DDM	4IOM	4IOM	DDM		MANUAL	DDM	DDM
14	4IOM	RCM				4IOM	DDM	RCM	DDM	4IOM	4IOM				PNTRZ	PNTRZ	IRON	ZKZK	DDM	ZKZK	IRON	IRON	DDM	4IOM	4IOM	4IOM		DUGME	DDM	DDM
15	RCM	RCM				DDM	DDM	MANUAL	DDM	4IOM	KM				2.SAH	MANUAL	1.SAH	DDM	4IOM	RCM	DDM	DDM	MANUAL	DDM	DDM	RCM		RCM	DDM	DDM
16	DDM					IRON	KRYK	4IOM	DDM	DDM	4IOM				4IOM	DDM	PNTRZ	4IOM	RCM	RCM	DDM	DDM	DDM	IRON	IRON	DDM		DDM	DDM	DDM
17	DDM					DDM	4IOM	DDM	RCM	4IOM	4IOM				DDM		MANUAL	RCM	DDM	DDM	DDM	DDM	4IOM	DDM	DDM	4IOM		DDM	RCM	RCM
18	DDM					DDM	KRYK	RCM	DDM	4IOM	BM							DDM	DDM		DDM	DDM	4IOM	DDM	DDM	DDM		DDM	DDM	DDM
19	RCM						DDM	DDM	RCM	4IOM	DDM							DDM	DDM		DDM	DDM	DDM			DDM		RCM	RCM	RCM
20	DDM						4IOM	RCM	DDM	BRM	BRM							4IOM	RCM				MANUAL			4IOM			DDM	DDM
21	4IOM						DDM	MANUAL	4IOM	BRM	DDM							ZKZK	RCM				4IOM			RCM			4IOM	4IOM
22	DDM						BANT	DUGME	MANUAL	BRM	DDM							RCM	DDM				DDM			DDM			MANUAL	RCM
23	ILMAK						DDM		RCM	DDM								RCM					DDM			DDM			RCM	4IOM
24	4IOM						DDM		4IOM	MANUAL								DDM					DDM						4IOM	RCM
25	DDM						PNTRZ		RCM	CITMAK													DDM						RCM	4IOM
26	RCM								DDM														4IOM						DDM	DDM
27	4IOM								4IOM														DDM						4IOM	MANUAL
28	4IOM								RCM														DDM						RCM	ILMAK
29	RCM								4IOM														DDM						4IOM	MANUAL
30	RCM								DDM														4IOM						DDM	DUGME
31	CITMAK								MANUAL														DDM						MANUAL	
32	DDM								ILMAK														DDM						ILMAK	
33	PNTRZ								MANUAL														DDM						MANUAL	
34	MANUAL								DUGME														PNTRZ						DUGME	
35																							MANUAL							
36																							CITMAK							

6. RESULTS AND DISCUSSION

The data given in Table 4 is used as the case study for the problem. The above mentioned algorithms are implemented in Java programming language in an object oriented fashion. FMM heuristic reaches the solution given in Table 5. This matrix is an input for the GA and SA implemented for solving the second subproblem. Among the two implemented meta-heuristics, the best solution of the second subproblem is realized by SA by the cost of time equal to moving 257 machines. Explicitly, this solution corresponds to producing the product types in the order of POLO107863, 107863, POLO107864, 107862, 103337, AR211067, AR211060, AR211047, POLO107862, 63646, B68373TENI, 237714, 263137, MNSSHEBA, 103361, CH103361, BANTBALAS, 103333, 103332, 1971248, POLO107861, 103470, ARHSH201, ARHSH20, 103334, 1972098, AEWZ8063, AEWZ8070, AEWZ8083, and AEWZ8080.

Table 5. Output of FMM When Applied To Case Study

Line	63646	103332	103333	103334	103337	103361	103470	107862	107863	237714	263137	1971248	1972098	AEWZ8063	AEWZ8070	AEWZ8080	AEWZ8083	AR211047	AR211060	AR211067	ARHSH20	ARHSH201	B68373TENI	18BALAS	CH103361	MNSSHEBA	POLO107861	POLO107862	POLO107863	POLO107864
63646	--	10	7	5	8	9	17	14	23	14	16	5	3	7	12	11	12	16	13	10	11	11	21	9	9	11	6	9	23	19
103332	29	--	5	5	6	11	19	14	28	19	18	4	4	8	13	14	13	18	15	10	13	13	31	11	11	17	6	13	28	24
103333	28	7	--	5	5	9	21	14	28	21	18	7	2	8	13	13	15	19	16	11	13	13	31	9	9	17	6	13	28	25
103334	29	10	8	--	9	13	22	15	28	22	20	5	5	9	15	14	15	19	18	13	14	14	33	13	13	19	6	16	28	25
103337	29	8	5	6	--	10	21	13	29	21	18	5	3	6	13	14	13	19	16	10	14	14	31	10	10	17	7	15	29	25
103361	25	8	4	5	5	--	17	11	26	18	16	4	3	7	12	12	12	18	15	8	11	11	27	0	0	12	6	14	26	23
103470	26	9	9	7	9	10	--	14	27	18	16	6	4	6	10	12	13	17	15	11	11	11	26	10	10	15	5	12	27	23
107862	26	7	5	3	4	7	17	--	24	19	17	5	3	6	12	12	13	16	14	10	12	12	28	7	7	15	5	13	24	20
107863	23	9	7	4	8	10	18	12	--	16	16	6	4	8	13	12	13	14	15	10	12	12	25	10	10	12	6	11	0	8
237714	23	9	9	7	9	11	18	16	25	--	13	6	5	7	12	13	13	16	15	12	12	12	23	11	11	13	6	12	25	22
263137	28	11	9	8	9	12	19	17	28	16	--	7	5	8	12	13	13	16	15	12	13	13	28	12	12	13	6	13	28	24
1971248	28	8	9	4	7	11	20	16	29	20	18	--	4	7	13	14	13	19	16	10	12	12	30	11	11	16	4	14	29	26
1972098	28	10	6	6	7	12	20	16	29	21	18	6	--	6	13	14	14	19	16	12	15	15	31	12	12	17	6	15	29	26
AEWZ8063	30	12	10	8	8	14	20	17	31	21	19	7	4	--	12	14	13	21	18	13	16	16	31	14	14	19	9	16	31	27
AEWZ8070	29	11	9	8	9	13	18	17	30	20	17	7	5	6	--	11	12	19	16	12	14	14	29	13	13	18	7	16	30	26
AEWZ8080	29	13	10	8	11	14	21	18	30	22	19	9	7	9	12	--	11	20	19	13	15	15	30	14	14	20	9	15	30	26
AEWZ8083	29	11	11	8	9	13	21	18	30	21	18	7	6	7	12	10	--	20	18	12	14	14	31	13	13	19	9	14	30	27
AR211047	26	9	8	5	8	12	18	14	24	17	14	6	4	8	12	12	13	--	12	8	13	13	26	12	12	13	6	9	24	23
AR211060	25	8	7	6	7	11	18	14	27	18	15	5	3	7	11	13	13	14	--	7	12	12	28	11	11	15	6	12	27	23
AR211067	27	8	7	6	6	9	19	15	27	20	17	4	4	7	12	12	12	15	12	--	13	13	29	9	9	14	6	13	27	23
ARHSH20	26	9	7	5	8	10	17	15	27	18	16	4	5	8	12	12	12	18	15	11	--	0	27	10	10	15	5	12	27	23
ARHSH201	26	9	7	5	8	10	17	15	27	18	16	4	5	8	12	12	12	18	15	11	0	--	27	10	10	15	5	12	27	23
B68373TENI	19	10	8	7	8	9	15	14	23	12	14	5	4	6	10	10	12	14	14	10	10	10	--	9	9	10	6	9	23	19
18BALAS	25	8	4	5	5	0	17	11	26	18	16	4	3	7	12	12	12	18	15	8	11	11	27	--	0	12	6	14	26	23
CH103361	25	8	4	5	5	0	17	11	26	18	16	4	3	7	12	12	12	18	15	8	11	11	27	0	--	12	6	14	26	23
MNSSHEBA	22	9	7	6	7	7	17	14	23	15	12	4	3	7	12	13	13	14	14	8	11	11	23	7	7	--	6	11	23	21
POLO107861	28	9	7	4	8	12	18	15	28	19	16	3	3	8	12	13	14	18	16	11	12	12	30	12	12	17	--	14	28	24
POLO107862	24	9	7	7	9	13	18	16	26	18	16	6	5	8	14	12	12	14	15	11	12	12	26	13	13	15	7	--	26	23
POLO107863	23	9	7	4	8	10	18	12	0	16	16	6	4	8	13	12	13	14	15	10	12	12	25	10	10	12	6	11	--	8
POLO107864	23	9	8	5	8	11	18	12	12	17	16	7	5	8	13	12	14	17	15	10	12	12	25	11	11	14	6	12	12	--

On the other hand, GA finds a solution with a cost of time equal to moving 265 machines. These results are the best solutions of each meta-heuristic. It is believed that the performance analysis of different values of parameters is worth to mention. In the below discussions, any value appearing is the average of 10 trials of the algorithm with the same parameter settings.

Our GA implementation has two parameters. Analysis of their effect on finding the optimal value is given in Figure 4. It is observed in the figure that, low *noi* and *nog* values present worse solutions. Increasing any or both of them brings better results. However, increasing *noi* parameter gives faster response for better results than increasing *nog* parameter. This phenomenon is best seen when *nog* is 1 and *noi* increases from 10 to 1000. So it can be concluded that diversification brings better solutions

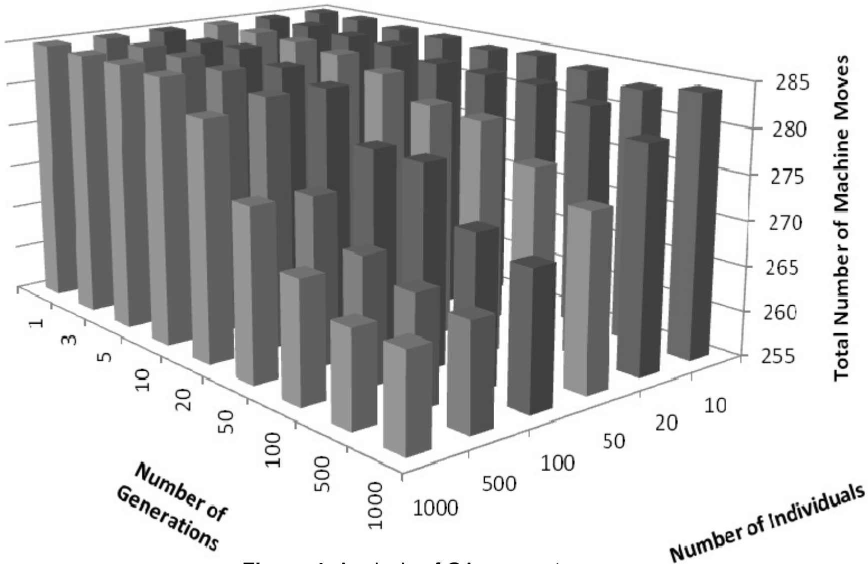

Figure 4. Analysis of GA parameters

Our SA implementation has five parameters each having two values. Therefore, number of different parameter settings becomes 32. Result of each parameter setting is given in Table 6.

It is observed that best performing parameter set is 100, 20, 1.1, 1.1 and N^3 for the values of *T(Initial temperature)*, *R(Number of iterations at each temperature setting)*, *a (Temperature decrease ratio)*, *b(Increase ratio in iteration number at each setting)*, and *nox(Maximum number of exchanges)* respectively. Best solution of SA (with a cost of 257) is also obtained by applying this parameter setting.

When the performance of these two meta-heuristics on our data set is compared, it is observed that SA displays a better performance both in terms of makespan and run time. Best solution is realized by SA and it is better than GA's best solution by %3. On the other hand, run time for SA is only 5 seconds for all of 320 runs, whereas it takes 165 seconds for GA to complete its 540 runs. Hence, on the average SA finds its result in 0.015 seconds while GA finds in 0.30 seconds. Thus, it can be concluded that SA gives better results in much more less amount of time to our problems.

Table 6. Analysis of SA parameters

T	R	a	B	nox	Number of machine moves
100	5	1,1	1,1	N^3	265,5
100	5	1,1	1,1	$N^3/3$	267,1
100	5	1,1	1,5	N^3	301,6
100	5	1,1	1,5	$N^3/3$	314,7
100	5	1,5	1,1	N^3	267,6
100	5	1,5	1,1	$N^3/3$	266,5
100	5	1,5	1,5	N^3	265,9
100	5	1,5	1,5	$N^3/3$	266,7
100	20	1,1	1,1	N^3	263,2
100	20	1,1	1,1	$N^3/3$	274,4
100	20	1,1	1,5	N^3	308,6
100	20	1,1	1,5	$N^3/3$	319,5
100	20	1,5	1,1	N^3	267,2
100	20	1,5	1,1	$N^3/3$	267,5
100	20	1,5	1,5	N^3	263,5
100	20	1,5	1,5	$N^3/3$	267,0
1000	5	1,1	1,1	N^3	269,4
1000	5	1,1	1,1	$N^3/3$	288,4
1000	5	1,1	1,5	N^3	330,2
1000	5	1,1	1,5	$N^3/3$	338,9
1000	5	1,5	1,1	N^3	268,4
1000	5	1,5	1,1	$N^3/3$	266,9
1000	5	1,5	1,5	N^3	264,3
1000	5	1,5	1,5	$N^3/3$	272,9
1000	20	1,1	1,1	N^3	288,3
1000	20	1,1	1,1	$N^3/3$	316,8
1000	20	1,1	1,5	N^3	328,7
1000	20	1,1	1,5	$N^3/3$	338,5
1000	20	1,5	1,1	N^3	267,5
1000	20	1,5	1,1	$N^3/3$	266,4
1000	20	1,5	1,5	N^3	268,3
1000	20	1,5	1,5	$N^3/3$	290,9

7. CONCLUSION

In this study, optimizing the throughput of a textile manufacturing company is realized. Specifically, optimization of setup times between production lines is analysed. The emerging problem is partitioned into two subproblems where the output of first subproblem is an input for the second. The subproblems are identified and an original heuristic is developed for the first subproblem.

It is also shown that the second subproblem is an instance of Open and Asymmetric Traveling Salesman Problem. For solving the second subproblem two meta-heuristics (genetic algorithm and simulated annealing) are implemented and their performances are compared on a case study with real data. Furthermore, their parameter performance is also presented. As a result, it is seen that simulated annealing has given better results. It outperformed GA by %3 in a less run time (20 times faster).

As a future work, more realistic problem constraints could be added to the problem. Moreover, uncertainty in scheduling times and production lines should be taken into account in deriving problem setup. It is believed that considering uncertainty in determining optimum solution will give much more realistic results.

REFERENCES

1. Manne, A.S.,1960, "On the job shop scheduling problem", *Operations Research*, 8(2), pp. 219-223.
2. Wagner, H., 1959, "An integer linear-programming model for machine scheduling", *Naval Research Logistics Quarterly*, 6(2), pp. 131-140.
3. Candido, M.A.B., Khator, S.K., Barcia, R.M., 1998, "A genetic algorithm based procedure for more realistic job shop scheduling problems", *International Journal of Production Research*, 36(12), pp. 3437-3457.
4. Ovacik, I.M., Uzsoy, R., 1994, "Exploiting shop floor status information to schedule complex job shops", *Journal of Manufacturing Systems*, 13(2), pp. 73–84.
5. Vinod, V., Sridharan, R., 2009, "Simulation-based metamodels for scheduling a dynamic job shop with sequence-dependent setup times", *International Journal of Production Research*, 47(6), pp. 1425-1447.
6. Zhu, X., Wilhelm, W.E., 2006, "Scheduling and lot sizing with sequence-dependent setup: A literature review", *IIE Transactions*, 38(11), pp. 987-1007
7. Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyovc, M.Y., 2008, "A survey of scheduling problems with setup times or costs", *European Journal of Operational Research*, 187(3), pp. 985-1032.
8. Diaz-Santillan, E., Malave, C.O., 2004, "Simulated annealing for parallel machine scheduling with split jobs and sequence-dependent set-ups", *International Journal of Industrial Engineering – Theory Applications and Practice*, 11(1), pp. 43-53.
9. Lin, S.W., Ying, K.C., 2009, "Applying a hybrid simulated annealing and tabu search approach to non-permutation flowshop scheduling problems", *International Journal of Production Research*, 47(5), pp. 1411-1424.
10. Azizi, N., Zolfaghari, S., Liang, M., 2010, "Hybrid simulated annealing with memory: an evolution-based diversification approach", *International Journal of Production Research*, 48(18), pp. 5455-5480.
11. Hasan, S.M.K., Sarker, R., Essam, D., 2011, "Genetic algorithm for job-shop scheduling with machine unavailability and breakdowns", *International Journal of Production Research*, 49(16), pp. 4999-5015.
12. Jia, Z., Lu, X., Yang, J., Jia, D., 2011, "Research on job-shop scheduling problem based on genetic algorithm", *International Journal of Production Research*, 49(12), pp. 3585-3604.
13. Ali, M.B., Sassi, M., Gossa, M., Harrath, Y., 2011, "Simultaneous scheduling of production and maintenance tasks in the job shop", *International Journal of Production Research* 49(13), pp. 3891-3918.
14. Chan, F.T.S., Wong, T.C., Chan, L.Y., 2006, "Flexible job-shop scheduling problem under resource constraints", *International Journal of Production Research*, 44(11), pp. 2071-2089.
15. Bowers, M.R., Agarwal, A., Knoxville, T.N., 1993, "Hierarchical production planning: scheduling in the apparel industry", *International Journal of Clothing Science and Technology*, 5(3/4), pp. 36-43.
16. Kwong, C.K., Mok, P.Y., Wong, W.K., 2006, "Scheduling flexible manufacturing systems for apparel production", *International Journal of Production Research*, 44(21), pp. 4465-4490.
17. Chan, C.C., Hui, C.L., Yeung, K.W., Ng, S.F., 1997, "Handling the assembly line balancing problem in the clothing industry using a genetic algorithm", *International Journal of Clothing Science and Technology*, 10(1), pp. 21-37.
18. Tucci, M., Rinaldi, R., 1999, "From theory to application: tabu search in textile production scheduling", *Production Planning & Control*, 10(4), pp. 365-374.
19. Wong, W.K., Mok, P.Y., Leung, S.Y.S., "Developing a genetic optimization approach to balance an apparel assembly line", *International Journal of Advanced Manufacturing Technology*, 28(3-4), pp. 387-394.
20. Guo, Z.X., Wong, W.K., Leung, S.Y.S., Fan, J.T., Chan, S.F., "Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment: A case study based on the apparel industry", *Computers & Industrial Engineering*, 50(3), pp. 202-219.
21. Karabuk, S., "Production planning under uncertainty in textile manufacturing", *Journal of the Operational Research Society*, 59, pp. 510-520.
22. Alkaya, A.F., 2009, "Optimizing the Throughput of a Textile Manufacturing Company", *YA\EM 09 Tam Bildiri Metinleri*, Ankara, Turkiye.
23. Chen, W.J., 2009, "Scheduling with dependent setups and maintenance in a textile company", *Computers & Industrial Engineering*, 57(3), pp. 867-873.
24. Hsu, H.-M., Hsiung, Y., Chen, Y.-Z., Wu, M.-C., "A GA methodology for the scheduling of yarn-dyed textile production", *Expert Systems with Applications*, 36(10), pp. 12095–12103.
25. Pinedo, M, 2002, *Scheduling: Theory, Algorithms and Systems*, Prentice Hall, Englewood Cliffs, NJ.
26. Gutin, G., Punnen, A., 2002, *The Traveling Salesman Problem and its Variants*, Kluwer Academic Publishers.
27. Holland, H., 1990, "Escaping brittleness: the possibilities of general purpose learning algorithms applied to parallel rule-based systems", in *Machine Learning: An Artificial Intelligence Approach*, Volume II, R. S. Michalski et al., Eds. Los Altos, CA, Morgan Kaufmann, pp. 593-623.
28. Beasley, D., Bull, D.R., Martin, R.R., 1993, "An overview of genetic algorithms: Part I, Fundamentals", *University Computing*, 15, pp. 58-69.
29. Michalewicz, Z., 1994, *Genetic Algorithms+Data Structures= Evolution Programs*, Springer, New York, USA.
30. Lancaster, J., Ozbayrak, M., "Evolutionary algorithms applied to project scheduling problems-a survey of the state-of-the-art", *International Journal of Production Research*, 45(2), pp. 425-450.
31. Aytug, H., Khouja, M., Vergara, F.E., "Use of genetic algorithms to solve production and operations management problems: A review", *International Journal of Production Research*, 41(17), pp. 3955-4009.
32. Blum, C., Roli, A., "Metaheuristics in combinatorial optimization: Overview and conceptual comparison", *ACM Computing Surveys*, 35, pp. 268-308.