The International Journal of Engineering and Science (IJES)
|| Volume || 6 || Issue || 8 || Pages || PP 64-73 || 2017 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Conflicting Objectives in Assembly Line Scheduling Problems

Akın Karabulut, Serol Bulkan
Department of Industrial Engineering, Faculty of Engineering,
Marmara University, 34722 Istanbul, Turkey
Corresponding author: Akın Karabulut

ABSTRACT
Classical planning and scheduling theories are limited in satisfying requirements of today's demand driven and competitive organizations. General approach in production scheduling is related to a single criterion minimization while scheduling problems often can require more than one constraint which increases the need of multiple criteria analysis. Besides, sometimes the objectives of the scheduling problem may conflict within each other which makes several theories unusable. In this work, we set up a real conflicting case in an assembly-line production utility based on component assembly and using component production in main-item composition in assembly-line production model. Our example has WIP costs which will be defined as waiting cost in the schedule and warehouse cost which is time needed to keep components until they are used in production of main items until they are delivered. A heuristic algorithm that we have developed is compared to optimum result of the defined problem in cost and calculation time.

Keywords: *Conflicting Objectives, WIP cost, Waiting Cost, Assembly Line, Scheduling, Heuristic*

Date of Submission: 28-07-2017 Date of acceptance: 10-08-2017

I. INTRODUCTION

For many organizations, meeting rising customer expectations while lowering production costs in an environment with more products, complexity and variety is placing great stress on the effectiveness of their planning and scheduling processes. Since 1960s, the focus of the objectives has gradually changed from single-criteria optimization to multiple conflicting criteria decision making. These multiple criteria can include not only time, but also resource, capacity and cost arguments. The traditional planning and scheduling approach is no longer providing responsiveness against businesses need in order to be effective in today's competitive, demand-driven environment [27].

Assembly lines are flow-oriented manufacturing systems in production of high quantity standardized commodities, in which usually interchangeable parts are added to a product in a sequential manner. This production method is subdivided into a number of line segments generally separated by buffers and can be either a main line or a supplying line. Assembly lines consist of a set of consecutive line stations and these lines are especially important in low volume production of customized items. These systems require noticeably high investment costs due to high level of automation [9].

Traditional assembly-line definition includes work stations k=1..., m and close functional mechanical equipment which is located through a conveyor belt. Jobs are performed consecutively along the line by moving between stations. The operations of assembly-line are executed with a planned to cycle time. Along these lines, the problem which is based on balancing the load by optimally proportioning between assembly work stations is known as Assembly-line balancing problem [22].

Assembly-line systems are used for mass and large-scale production for years. Advances on assembly-line production resulted as more flexible designs, such as parallel lines, tasks and work-stations, customer-oriented mixed model, multi-model lines and U-Shaped lines including intermediate buffers [22].

Most of research in production scheduling is concerned with the minimization of a single criterion. However, real life problems usually require balancing the trade-off between conflicting objectives [16]. Under some circumstances resource-based objectives can conflict with order-based objectives. These are the most common classes of objectives which may conflict with each other. For instance, satisfying due dates is an order-based objective while efficient machine utilization is a resource-based objective [6].

There has been applied works related to conflicting objectives regarding cost minimization and profit maximization perspective. Endre Boros and Lei Lei considered the problem of coordinating the operations of two supply chain partners: a foreign shipping company and a domestic port. The two partners have conflicting business objectives, and the issue is to determine the optimal cycle time, by which the shipping company removes the empty containers from the domestic port, so that the joint profit of the two partners is maximized.

The domestic port prefers a shorter cycle time to mitigate its empty container accumulation and land use problems, while the shipping company wishes a longer cycle time to save its expensive vessel capacities. They proposed an iterative procedure to search for this optimal cycle time. In each iteration, a candidate cycle time is evaluated by solving a deterministic vessel scheduling problem and a stochastic container-yard capacity optimization problem [8].

From quality point of view, of course while costing minimization is one of the major expectation of a profit organization, increasing quality (maximization of quality) is from a break-even point, can be evaluated as conflicting objectives. Therefore, while decreasing cost of the production may lead decrease of quality which is another major expectation. Conflicting objectives generally appear to have a result including costing effect on the organization. The costs which cause unexpected results are traditionally named as Penalty. This approach neglects storage costs due to insurance, theft, perishing, and bounded capital for the case of a completion of the job before the due date.

II. LITERATURE SURVEY

Most of the scheduling problems are complex combinatorial problems that are hard to solve while problem size increases. Constraint Programming (CP) becomes a popular approach to solve these hard problems. It provides natural modeling capabilities to describe many real-life problems via domain variables and constraints among these variables. The idea of CP is to describe the problem declaratively by using constraints over variables. Scheduling is one of the most successful application areas of constraint programming [24].

Several methods are applied using simulation and mathematical programming techniques to solve scheduling problems. The proposed approaches include network analysis, specific algorithms and also heuristics. The selection of an appropriate approach depends on the complexity of the problem, the quantity and the configuration of the machines, the production system, the scheduling system and the static or dynamic nature of job arrivals [3]. In general, the selection of scheduling policy to handle at the bottleneck resource directly affects the success of the solution.

Related to assembly-line, Marc T. Kaufman in 1973, has worked on longest-path scheduling algorithm as an "almost" optimal algorithm for the scheduling of trees in multiprocessor scheduling, the longest path algorithm assigns free processors at any time to those available tasks which are furthest from the root of the tree and processors are never left idle if they can be assigned [14].

The study of Tadeusz Sawik in 2002 included using Mixed Integer Programming for Flexible Assembly Systems (FAS). The objective of the problem was to determine an allocation of assembly tasks and component feeders among the stations and to find an assembly schedule for a mix of products with no revisiting of stations so as to complete the products in minimum time. It was pointed out that the performance of the mixed integer programming models may depend on the FAS configuration (e.g., single vs parallel assembly stations, single vs multiple in-process buffers, etc.), the cutting constraints that are included in the corresponding MIP models and the solver configuration [20].

In another study of Assembly Line Scheduling problems in 2003, Ali Tozkapan, Ömer Kırca and Chia-Shin Chung worked on two-stage assembly scheduling problem to minimize total weighted flow time. A lower bounding procedure and a dominance criterion are developed and incorporated into a branch and bound procedure supplied with an initial heuristic upper bound [23].

In 2006, Armin Scholl and Christian Becker worked on Heuristic Solutions for Simple Assembly Line Balancing and resulted in significant algorithmic developments. Though SALBP is a class of NP-hard optimization problems, effective exact and heuristic procedures are available which solve medium-sized instances sufficiently for practical use. However, further improvements are required to solve large-scale instances [22]

In 2006, Carlos Andres, Cristobal Miralles and Rafael Pastor named General Assembly Line Balancing Problem with Setups as GALBPS, defined nine basic assumptions, a binary programming model, and tested against several problems. These tests proved the requirement for heuristic approaches to solve larger instance problems in reasonable computational time. After formulating a mathematical model, eight different heuristic rules and a GRASP algorithm are designed and tested for solving the problem in reasonable computational time [4].

In 2007, Fawaz S. Al-Anzi and Ali Allahverdi worked on adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times. They addressed two-stage assembly flowshop scheduling problem with respect to maximum lateness criterion where setup times are treated as separate from processing times. They conducted extensive computational experiments to compare the performance of the proposed heuristic with those of particle swarm optimization (PSO), tabu search, and EDD heuristics. The computational analysis indicates that PSO performs much better than remaining [1].

In 2010, Öztürk has worked on Flexible Mixed Model Asembly Lines with Sequence Dependent Setup Times to minimize Cmax and proposed alternate MIP and CP formulations. CP models seemed more promising on small instances [18].
One example for scheduling algorithm which takes consideration of warehouse costs is with steady state genetic algorithm by Jalal Atoum and Mamoun Al Rababaa. Warehouses scheduling is the problem of sequencing requests of products to fulfill several customers' orders so as to minimize the average time and shipping costs [5]. Another genetic algorithm for Tardiness problem has been applied by Dirk C. Mattfeld and Christian Bierwirth for Tardiness objectives. First, a complexity reduction is achieved by narrowing the scope at the machine level by means of the schedule builder. Second, a problem decomposition at the shop floor level by means of a multi-stage approach is considered. The effects of both approaches are investigated independently before they are combined [17].
The effect of incorporating queuing waiting time in the calculations of job slack time and critical ratio was investigated by Berry and Finlay in 1976. They used flow time, job lateness, and work-in-process as the performance measures [2]. Waiting time has been used in scheduling in several heuristics such as Truncated Shortest Processing Time, LPT to minimize mean waiting time, SPT to minimize mean waiting time etc. In this work, we will use waiting time to calculate waiting cost for each product and subcomponents and use the heuristics to minimize total waiting cost in the schedule.

III. METHODOLOGY

The target of this study is to provide a model for decision makers in order to decide the number of objectives that should be operated and which objectives to include in the optimal set to produce the lowest operational cost, with the best overall performance, best welfare for an assembly line production model.
Master data of the model can be explained as follows. Item (product and components) definition includes warehouse cost and waiting costs in the schedule. Item type is either production item or component. Warehouse cost occurs for components and products if items stay in warehouse until deadline. Deadline for component is the date it is used in related composition task, deadline for product is the planned delivery date defined in the order. Item Route definition includes each task required to execute for product route definition includes item, sequence for the task, machine information, and time required to execute the task (processing time) in the machine. Bill of Material (BOM) definition includes the relation between product and component. Machine information is the task which related component will be issued for item production and quantity needed to use in order to produce main-item. Order Information includes customer orders from the assembly-line production company and deadlines from production. Order information is defined only for end product items.
According to all defined explanations regarding scheduling and conflicting objectives, we set up a real conflicting case in an assembly-line production utility based on component assembly and using component production in main-item composition in assembly-line production model. Our example has WIP costs which will be defined as waiting cost in the schedule and warehouse cost which is time needed to keep components until used in production of main item and main items until they are delivered. In the model, all main items have to be finished before defined due date. Orders are received from customer of production company with defined deadlines which is delivery date for the main-item in the schedule. In defined model, we have 3 main items like (A, B, C) and all items have 4 components, like (Z1, Z2, Z3, Z4) which need to be produced in the same schedule. Schedule is based on 4 machines M1, M2, M3, M4. Machines are flow-oriented in sequential manner. In the model routing is defined for all items which describe the routing of each item and in each of the task defined in the route definition, a machine definition and time needed to do the task in specified machine exists. Bill of material definition defines the relation of main-item to component and also where the component will be issued for production of main item composition process. In the schedule, time needed to produce component item is also defined. For each of the item warehouse cost and waiting cost is defined.

3.1 Restrictions and Rules of the Schedule

Restrictions and rules of the schedule can be defined as follows:
1. Schedule starts with orders from customer.
2. There is predefined item, BOM and route information regarding definitions explained above.
3. Each item has subcomponents which has to be produced before starting related route.
4. Subcomponents are issued to production on defined BOM machines which requires components has to be produced before it is required to be issued in the related machine.
5. If any item cannot be finalized before deadline, then the schedule is infeasible. The cost of the schedule is unlimited when completion time of any product is bigger than deadline.
6. Each machine can only run one item at one unit of time.

7. Warehouse-Storage cost: is the cost (including product and component cost) of any item's time staying in the warehouse after route task of item is executed which is multiplied by unit warehouse cost of the related item. Warehouse time is the time which product is finalized before deadline which is Earliness. In terms of scheduling it is calculated as Deadline subtracted from the completion time of the last task of item. Because in the schedule, there is predefined storage cost for each item, earliness value is multiplied by storage cost in order to find warehouse cost of the item. Total warehouse cost of product and components is the warehouse cost of the schedule.
8. Waiting Cost (WIP cost): The cost of waiting in the schedule until the last task of the related item is finalized. Every item has a standard processing time, a starting time, and a completion time in the schedule. Standard time for product is calculated with assumption that no other task out of the product tasks is executed in the same schedule. In another word time needed to produce only product in the same schedule. According to standard time (processing time), the time between standard time and completion time is the waiting time. In terms of scheduling, waiting time is calculated as completion time subtracted from processing time (standard time of item). This waiting time is multiplied by the waiting cost for each product and component. Schedules waiting cost is total cost of all items in the schedule.

3.2 Notation of the Model

In this work the subsequent mathematical model is used to represent conflicts of objectives in Assembly Line Scheduling Problem.

k: Number of products
z: Number of sub components
l: Number of sequence (route for each item)
S_{kl}: Starting time of product k route l
Cw_k: Unit waiting cost of item
Cw_z: Unit waiting cost of subcomponent
Cs_k: Unit warehouse- storage cost of item
Cs_z: Unit warehouse- storage cost of subcomponent
C_k: Completion time of each product k
C_{kl}: Completion time of each product k route l
C_{klz}: Completion time of each product k route l subcomponent
ICw_k: Waiting cost of product item (A, B, C)
ICw_{klz}:Waiting cost of processing time of each product k route l sub-component
ICs_k: Storage cost product item (A, B, C)
$ICs_{k\,lz}$: Storage cost item each product k route l sub-component
W_k: Waiting time which is the difference between completion time and processing time
W_{klz}: Waiting time of product k route l subcomponent
P_k: Processing time of each product
p_{kl}: Processing time of each item on each sequence
p_{klz}: Processing time of each product k route l subcomponent
d_k: Deadline of products
d_{klz}: Deadline of subcomponent
(Due date for products, start time for related product task for component)
a1: Weight of waiting costs
a2: Weight of warehouse costs

3.3 Mathematical Model

In order to compare Scheduling results of Heuristic applied above, closed form of IP model is defined as below.

Minimize Z;

$$Z = a1\left(\sum_k ICw_k + \sum_k \sum_l ICw_{klz}\right) + a2\left(\sum_k ICs_k + \sum_k \sum_l ICs_{klz}\right) \quad (1)$$

Subject to;

$C_k <= d_k$ (2)

$ICwk >= 0$ (3)

$ICsk >= 0$ (4)

$E_k = \max(0, d_k - C_k)$ (5)

$ICs_k = Cs_k E_k$ (6)

$ICw_k = Cw_k (C_k - P_k)$ (7)

$d_{klz} = S_{kl}$ (8)

$E_{klz} = \max(0, d_{klz} - C_{klz})$ (9)

$ICs_{klz} = Cs_z E_{klz}$ (10)

$ICw_{klz} = Cw_z (C_{klz} - P_z)$ (11)

$S_{kl} - S_{(k+n)}l + p_{kl} <= M * (1 - y_{k(k+n)l})$ (12)

$S_{(k+n)l} - S_{(k)l} + p_{(k+n)l} <= M * (y_{k(k+n)l})$ (13)

$S_{kl} - S_{(k-n)l} + p_{kl} <= M * (1 - y_{k(k-n)l})$ (14)

$S_{(k-n)l} - S_{(k)l} + p_{(k-n)l} <= M * (y_{k(k-n)l})$ (15)

$S_{kl} - S_{(k+n)lz} + p_{kl} <= M * (1 - y_{k(k+n)lz})$ (16)

$S_{(k+n)lz} - S_{(k)l} + p_{(k+n)lz} <= M * (y_{k(k+n)lz})$ (17)

$S_{kl} - S_{(k-n)lz} + p_{kl} <= M * (1 - y_{k(k-n)lz})$ (18)

$S_{(k-n)lz} - S_{(k)l} + p_{(k-n)lz} <= M * (y_{k(k-n)lz})$ (19)

$S_{klz} - S_{(k+n)lz} + p_{klz} <= M * (1 - y_{k(k+n)lz})$ (20)

$S_{(k+n)lz} - S_{(k)lz} + p_{(k+n)lz} <= M * (y_{k(k+n)lz})$ (21)

$S_{klz} - S_{(k-n)lz} + p_{klz} <= M * (1 - y_{k(k-n)lz})$ (22)

$S_{(k-n)lz} - S_{(k)lz} + p_{(k-n)lz} <= M * (y_{k(k-n)lz})$ (23)

$S_{kl} - S_{klz} >= p_{klz}$ (24)

$C_{klz} - S_{klz} = p_{klz}$ (25)

$S_{kl} - S_{k(l-1)} >= p_{kl}$ (26)

$P_k = \sum_l (P_{kl} + P_{klz})$ (27)

$W_k = C_k - P_k$ (28)

$$S_{klz} \leq S_{kl} - P_{klz} \quad (29)$$

Equation (1) represents the objective function. Constraint (2) defines a deadline constraint that for every product has to finish before or equal to deadline defined in the order. Constraint (3) and (4) provide that waiting cost and warehouse-storage cost for every product must be positive or zero respectively. Equation (5) represents that warehouse time is earliness which is deadline of subtracted from completion time of product. Completion time of the product is the completion time of last route of product k, and equation (6) defines that warehouse cost (is the multiplication of unit warehouse-storage cost with warehouse-storage time (deadline subtracted from completion time of k). Waiting cost is calculated by Equation (7) with the multiplication of unit waiting cost with waiting time (completion time of last task of the item subtracted from total processing time of the item). For every sub-component z produced to be used for product k route l. Equation (8) represents that deadline of subcomponent is starting time of product k route l. Equation (9) and (10) indicates that storage time cannot be negative and warehouse cost is the multiplication of unit warehouse-storage cost of sub-component with warehouse-storage time (sub-component stays in warehouse after produced, sub-component warehouse-storage time is time between completion time of sub-component until starting of the related route operation for the product S_{kl}). Similarly, for every sub-component z produced to be used for product k route l, waiting cost calculated by Equation (11) with the multiplication of unit waiting cost of sub-component with waiting time (time needed to wait until z is produced which is the difference between completion time C_{klz} and processing time of P_z). Constraints (12), (13), (14) and (15) are concerned with product item route sequences (One product operation, can allocate one machine at a time). Operations which is defined for the same machine is defined as below to prevent overlap n # of product items, either S_{kl} or $S_{(k+n)l}$ is scheduled at a time, y can be 0 or 1. Constraints (16), (17), (18) and (19) are related to product item route, sub-component sequences (One product operation, can allocate one machine at a time). Operations which is defined for the same machine is defined as below to prevent overlap n # of product items, either S_{kl} or $S_{(k+n)l}$ is scheduled at a time, y can be 0 or 1. Constraints (20), (21), (22) and (23) are concerned with the relation between sub-components. Constraint (21) provides starting time of product k route l S_{kl} must be more than S_{klz} at least p_{klz}. Equation (25) and (26) are concerned with relation of every sub-component operation and every product operation respectively. Equation (27) indicates the processing time of each product k task l and processing time of subcomponent z which is produced before product k task l. Constraint (28) includes definition of waiting time of each product which is the completion time of last task of the item subtracted from total processing time of the item. Constraint (29) represents that component start time can be any time before starting of product k route l.

3.4 Heuristic Method

The process of heuristic algorithm applied to solve problem is defined as follows:

1. Applied heuristic is selecting the lowest schedule cost in each iteration.
2. Number of iteration is calculated by order and route combination. For example, if 3 orders will be processed, each item having 4 route tasks defined, 12 iterations are processed in order to decide which item and task has the lowest storage and waiting cost.
3. In every iteration, schedule cost (waiting and warehouse cost) is calculated and compared. Each iteration results with a decision, and selected item and task (job) in previous selection is eliminated.
4. In each iteration, comparison is made between same work center tasks based on the routing definition. For example, if 3 products have routing task in the same work center, the decision is made between the 3 tasks.
5. Costing comparisons in the same work center also includes component cost which has to be produced before starting the task. For example, if Z1 is required for A1, Z1 waiting and warehouse cost assumption is added to cost of A1 to compare with B1 and C1.
6. Waiting cost of A1 for comparison is calculated as;

- if A1 is selected the shifting cost of B and C is calculated,
- if A1 is selected which means Z1 has to be produced before which results shifting cost of B and C schedule including sub-components is calculated and added to comparison as A1 cost.

7. Warehouse cost of A1 is calculated by finding best available time to finalize A with assuming all tasks of A is finished with priority, warehouse cost of A is calculated with this assumption.

3.5 Pseudo Code of Algorithm

Initialize number of iterations by order X number of routes decision
While iteration<number of max iterations
Selection of unscheduled smallest route sequence
Find machine of that selected sequence
Find all unscheduled products routes for selected machine with selected sequence
While not all selected product routes calculated
Select of any product route in Selected machine and sequence set
Calculate estimated warehouse cost of selected product route
Calculate estimated warehouse cost of sub-component
Calculate estimated waiting cost of selected item route
Calculate waiting cost of subcomponent for the route
Calculate estimated total cost for product route
endwhile
Compare calculated route costs and Schedule least cost product route
increase iteration
endwhile

IV. COMPUTATIONS AND RESULTS

The computations have been processed with Heuristic which is described in Heuristic Method section 3.4 and IP in the mathematical model. Table 1 and Table 2 includes master data example of randomly selected problem while Table 3 and Table 4 include solution of the same problem with 3 products with 4 machines, 4 subcomponents. Table 5 includes IP and Heuristic solution comparison of randomly selected 10 different problems

Table 1: Product and components data

Item	**Type**	**Warehouse Cost**	**Waiting Cost**
A	Product	3	15
B	Product	2	2
C	Product	1	3
Z1	Component	2	2
Z2	Component	2	4
Z3	Component	3	5
Z4	Component	4	2

Table 2: Item Route definition

Item	**Sequence**	**Machine**	**Processing Time**
A	1	M1	6
A	2	M2	8
A	3	M3	4
A	4	M4	3
B	1	M1	7
B	2	M2	9
B	3	M3	9
B	4	M4	5
C	1	M1	5
C	2	M2	1
C	3	M3	5
C	4	M4	7
Z1	1	M1	4
Z2	2	M2	5
Z3	3	M3	8
Z4	4	M4	4

Table 3: IP results (3 products, 4 machines, 4 subcomponents)

Item	Machine	Due Date	Release Time	Finish Time	Ware. Time	Unit War. Cost	Ware. Cost	Standard Time	Waiting Time	Unit Waiting Cost	Waiting Cost
A		60.00	4.0	25.0	35.0	3.0	105.0	25.0	0.0	15.0	0.0
Z1	M1	4.00	0.00	4.00	0.0	2.0	0.0	4.0	0.0	2.0	0.0
Z2	M2	10.00	0.00	5.00	5.0	2.0	10.0	5.0	0.0	4.0	0.0
Z3	M3	18.00	0.00	8.00	10.0	3.0	30.0	8.0	0.0	5.0	0.0
Z4	M4	22.00	18.00	22.00	0.0	4.0	0.0	4.0	18.0	2.0	36.0
B		60.00	23.0	53.0	7.0	2.0	14.0	34.0	19.0	2.0	38.0
Z1	M1	23.00	19.00	23.00	0.0	2.0	0.0	4.0	19.0	2.0	38.0
Z2	M2	30.00	20.00	25.00	5.0	2.0	10.0	5.0	20.0	4.0	80.0
Z3	M3	39.00	27.00	35.00	4.0	3.0	12.0	8.0	27.0	5.0	135.0
Z4	M4	48.00	44.00	48.00	0.0	4.0	0.0	4.0	44.0	2.0	88.0
C		60.00	14.0	36.0	24.0	1.0	24.0	22.0	14.0	3.0	42.0
Z1	M1	14.00	10.00	14.00	0.0	2.0	0.0	4.0	10.0	2.0	20.0
Z2	M2	19.00	5.00	10.00	9.0	2.0	18.0	5.0	5.0	4.0	20.0
Z3	M3	22.00	8.00	16.00	6.0	3.0	18.0	8.0	8.0	5.0	40.0
Z4	M4	29.00	25.00	29.00	0.0	4.0	0.0	4.0	25.0	2.0	50.0
Total Cost							241.0				587.0
Total sum											828.0

Table 4: Heuristic result (3 products, 4 machines, 4 subcomponents)

Item	Machine	Due Date	Release Time	Finish Time	Ware. Time	Unit War. Cost	Ware. Cost	Standard Time	Waiting Time	Unit Waiting Cost	Waiting Cost
A		60.00	4.0	25.0	35.0	3.0	105.0	25.0	0.0	15.0	0.0
Z1	M1	4.00	0.00	4.00	0.0	2.0	0.0	4.0	0.0	2.0	0.0
Z2	M2	10.00	0.00	5.00	5.0	2.0	10.0	5.0	0.0	4.0	0.0
Z3	M3	18.00	0.00	8.00	10.0	3.0	30.0	8.0	0.0	5.0	0.0
Z4	M4	22.00	18.00	22.0	0.0	4.0	0.0	4.0	18.0	2.0	36.0
B		60.00	23.0	57.0	3.0	2.0	6.0	34.0	23.0	2.0	46.0
Z1	M1	23.00	19.00	23.0	0.0	2.0	0.0	4.0	19.0	2.0	38.0
Z2	M2	30.00	24.00	29.0	1.0	2.0	2.0	5.0	24.0	4.0	96.0
Z3	M3	43.00	35.00	43.0	0.0	3.0	0.0	8.0	35.0	5.0	175.0
Z4	M4	52.00	48.00	52.0	0.0	4.0	0.0	4.0	48.0	2.0	96.0
C		60.00	14.0	42.0	18.0	1.0	18.0	22.0	20.0	3.0	60.0
Z1	M1	14.00	10.00	14.0	0.0	2.0	0.0	4.0	10.0	2.0	20.0
Z2	M2	23.00	18.00	23.0	0.0	2.0	0.0	5.0	18.0	4.0	72.0
Z3	M3	30.00	22.00	30.0	0.0	3.0	0.0	8.0	22.0	5.0	110.0
Z4	M4	35.00	31.00	35.0	0.0	4.0	0.0	4.0	31.0	2.0	62.0
Total Cost							171.0				811.0
Total sum											982

Table 5: Heuristic and IP results comparison

Example	Heuristic Results	IP Results	Difference	Difference Percentage
1	982	828	154	%18,59
2	1250	1105	145	%13,12
3	1117	952	165	%17,33
4	1169	1054	115	%10,9
5	771	658	113	%17.17
6	711	616	95	%15.42
7	1204	1114	90	%8.07
8	823	705	118	%16.74
9	462	420	42	%10.0
10	422	374	10	%12.83

V. CONCLUSION

This paper represents an implementation of a new heuristic model for assembly line scheduling problem having conflicting constraints. The heuristic developed in this work is under special conditions and applicable with predefined rules and restrictions. The results indicate that solution can supply significant time and resource advantage especially in assembly line production environment. The algorithm can be enhanced by applying this problem to different and wide scope real life cases and algorithm can be developed to decrease total cost by shifting the tasks/routing activities by considering conflicting constraints.

REFERENCES

[1]. F. Al-Anzi, A. Allahverdi, A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times. *European Journal of Operational Research ,182,* 2007, 80–94.

[2]. I.M. Alharkan, *Algorithms for Sequencing and Scheuling* (Industrial Engineering Department, King Saud University, 2007, 26–29)

[3]. M.A. Allouche, B. Aouni, J.M. Martel, T. Loukil, A. Rebai, Solving multi-criteria scheduling flow shop problem through compromise programming and satisfaction functions, *European Journal of Operational Research, 192*, 2009, 460-467.

[4]. C. Andres, C. Miralles, R. Pastor, Balancing and scheduling tasks in assembly lines with sequence-dependent setup times, *European Journal of Operational Research, 187*, 2008, 1212–1223.

[5]. J. Atoum, M. Rababaa, Multiple Warehouses Scheduling Using Steady State Genetic Algorithms, *The International Arab Journal of Information Technology, 7*, 2010, 310–316.

[6]. P.M. Berry, Satisfying Conflicting Objectives in Factory Scheduling, *Artificial Intelligence Applications*, *1*, 1990, 101-107.

[7]. P.M. Berry, A Predictive Model for Satisfying Conflicting Objectives in Scheduling Problems, *AI Magazine*, *13,* 1992, 13-15.

[8]. E. Boros, L. Lei, Y. Zhao, H. Zhong, Scheduling Vessels and Container-Yard Operations with Conflicting Objectives, *Rutcor and Rutgers Business School, Rutgers University: SUNY– Oneonta, State University of New York*, 2006, 1–26.

[9]. N. Boysen, M. Fliedner, A. Scholl, A classification of assembly line balancing problems, *European Journal of Operational Research, 183*, 2007, 674–93.

[10]. Y. Chang, N. Lee, A Multi-Objective Goal Programming airport selection model for low-cost carriers' networks, *Transportation Research Part E*, *46*, 2010, 709–718.

[11]. Z.Chen, N.G. Hall, Supply Chain Scheduling: Conflict and Cooperation in Assembly Systems, *Operations Research*, *55,* 2007, 072-1089.

[12]. J. J.Durillo, A.J. Nebro, jMetal: a Java Framework for Developing Multi-Objective Optimization Metaheuristics, *Departamento de Lenguajes y Ciencias de la Computaci´on E.T.S. Ingenier´ıa Inform´atica TECH-REPORT: ITI-2006-10*, *9*, 2006, 1–12.

[13]. H. Hoogeveen, Multicriteria scheduling, *European Journal of Operational Research, 167*, 2005, 592-623.

[14]. Mark T. Kaufman, An Almost-Optimal Algorithm for the Assembly Line Scheduling Problem, *Stanford Electronic Laboratories Technical Report NO. 53*, *9*, *1973*, 9–12.

[15]. T.O. Lee, Y. Kim, Y.K. Kim, Two-sided assembly line balancing to maximize work relatedness and slackness, *Computers and Industrial Engineering, 40,* 2001, 273–92.

[16]. T. Loukil, J. Teghem., D. Tuyttens, Solving multi-objective production scheduling problems using metaheuristics, *European Journal of Operational Research*, *161*, 2005, 42-61.

[17]. D.Mattfeld., C. Bierwirth, An efficient genetic algorithm for job shop scheduling with tardiness objectives, *European Journal of Operational Research, 155*, 2004, 616–630.

[18]. C. Ozturk, S. Tunali, B. Hnich, A.M. Ornek, Simultaneous Balancing and Scheduling of Flexible Mixed Model Assembly Lines with Sequence-Dependent Setup Times. *Electronic Notes in Discrete Mathematics, 36*, 2010, 65–72.

[19]. M. Peeters, Z. Degraeve, A linear programming based lower bound for the simple assembly line balancing problem, *European Journal of Operational Research, 168*, 2006, 716–731.

[20]. T. Sawik, Loading and scheduling of a flexible assembly system by mixed integer programming, *European Journal of Operational Research*, *154*, 2004, 1–19.

[21]. A. Scholl, C. Becker, A survey on problems and methods in generalized assembly line balancing, *European Journal of Operational Research, 168***,** 2003, 694–715.

[22]. A. Scholl, C. Becker, State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, *European Journal of Operational Research*, *168*, 2006, 666–693.

[23]. A. Tozkapan, Ö. Kırca, C. Chung, A branch and bound algorithm to minimize the total weighted flowtime for the two-stage assembly scheduling problem, *Computers & Operations Research, 30,* 2003, 309–320.

[24]. P. Vilim, *Global Constraints in Scheduling*. Phd. Thesis, Charles University, Prague, 2012

[25]. M. B. Wall, *A Genetic Algorithm for Resource-Constrained Scheduling*, Doctor of Philosophy in Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 1996
[26]. S.Wang, C. Ko, Constraint programming approach to precast production scheduling, *Journal of Construction Engineering and Management*, *128*, 2002, 513–521.
[27]. S.Wang, C. Ko, Precast production scheduling using multi-objective genetic algorithms, *Expert Systems with Applications*, *38*, 2011, 8293-8302.
[28]. A. Wierman, *Scheduling for Today's Computer Systems: Bridging Theory and Practice*, PHD Thesis School of Computer Science Carnegie Mellon University, Pittsburgh, PA, 2007

Akın Karabulut " Conflicting Objectives in Assembly Line Scheduling Problems" The International Journal of Engineering and Science (IJES) 6.8 (2017): 64-73.