

SIMPLE AND U-TYPE ASSEMBLY LINE BALANCING BY USING AN ANT COLONY BASED ALGORITHM

Adil Baykasoğlu and Türkay Dereli

Department of Industrial Engineering, University of Gaziantep, Gaziantep, Turkey {baykasoglu, dereli}@gaziantep.edu.tr

Abstract- In this paper, an Ant Colony Optimization (ACO) based heuristic algorithm is proposed for solving simple (straight line) and U-shaped assembly line balancing problems (ALBP). The paper makes one of the first attempts to show how ACO heuristic can be used to solve U-shaped ALBP. A new algorithm is proposed in this paper that seamlessly integrates COMSOAL algorithm, Ranked Positional Weight Heuristic (RPWH) and an ACO based heuristic in order to obtain good solutions to simple and U-shaped ALBPs. The result of the computational study has shown that the proposed algorithm is effective in solving simple/U-shaped line balancing problems. **Key Words-** Assembly Line Balancing, Ant Colony Optimization, Heuristics

1. INTRODUCTION

ALBP relates to a finite set of work elements or tasks, each having an operation processing time and a set of precedence relations, which specify the permissible orderings of the tasks. One of the problems in organizing mass production is how to group work tasks to be performed on workstations so as to achieve the desired level of performance. Line balancing is an attempt to allocate equal amounts of work to the various workstations along the line. The fundamental line-balancing problem is how to assign a set of tasks to an ordered set of workstations, such that the precedence relations are satisfied and some measure of performance is optimized [17]. When designing an assembly line there are two main restrictions that must be imposed on the grouping of work elements [15]. Precedence relationship; The cycle time is greater than or equal to the maximum of any station time and of the time of any work element. The station time must not exceed the cycle time.

The ALBP as defined above is known as a NP-hard problem [23] and it is formally given in Equation set 1 [19]. Equation set 1 represents the type of mathematical problem that is tried to be solved. Mathematical formulations for the Ushaped ALBPs can also be found in Gokcen *et al.* [20], Gokcen and Agpak [21]. If there are *n* tasks and *r* precedence requirements then there are roughly $n!/2^r$ possible tasks sequences in a simple ALBP [3]. With such a vast search space it is difficult to obtain an optimal solution for big size problems by using deterministic algorithms. Many attempts have been made in the literature to solve the ALB problem with exact deterministic methods. Some of these methods have proven to be efficient for the problems up to 300 tasks. But for larger problems their practical use is limited due to their computational inefficiency. Baybars [3] and Talbot *et al.* [34] reviewed and evaluated these different approaches.

$$\min \sum_{k=1}^{m_{\max}} z_{k}$$
s.t.
$$\sum_{k=LB_{i}}^{UB_{i}} x_{ik} = 1 \qquad \forall i$$

$$\sum_{i \in W_{k}} t_{i}(x_{ik}) \leq C \qquad k = 1, ..., m_{\max} \qquad (1)$$

$$\sum_{k=LB_{a}}^{UB_{a}} k(x_{ak}) - \sum_{k=LB_{b}}^{UB_{b}} k(x_{bk}) \leq 0 \quad \forall (a,b) \in P$$

$$\sum_{i \in W_{k}} x_{ik} - \|W_{k}\| z_{k} \leq 0 \qquad k = 1, ..., m_{\max}$$

$$x_{ik}, z_{k} \in \{0,1\} \forall i, k$$

 UB_i : upper bound on the station to which *i* may be assigned (S_i : set of all successors of *i*),

$$UB_i = N + 1 - \left[\left(t_i + \sum_{j \in S_i} t_j \right) / C \right]^{\top}$$

 LB_i : lower bound on the station to which *i* may be assigned (P_i : set of all predecessors of *i*),

$$LB_{i} = \left[\left(t_{i} + \sum_{j \in P_{i}} t_{j}\right) / C \right]^{+}$$

where; C: Cycle time, m_{max} : predetermined maximum number of stations; W_k : subset of all tasks that can be assigned to station k; $||W_k||$: number of tasks in set W_k ; LB_i : earliest station task i can be assigned to (given precedence relations); UB_i : latest station task i can be assigned to (given precedence relations); P: set of tasks that precedes from a task, x_{ik} : 1 if task i is assigned to workstation k, 0 otherwise; z_k : 1, if there is any task assigned to workstation k, 0 otherwise. The first constraint assures that a task cannot be assigned to more than one workstation and must be assigned to a workstation. The second constraint ensures that the cycle time is not violated. The third constraint ensures that assignments are made to the open stations. Finally, the objective function of the formulation is to minimize the number of workstations (or line efficiency).

The assembly line balancing problem (ALBP) can be divided into two groups, according to the classification proposed by Baybars [3]: Simple Assembly Line Balancing (SALBP) and General Assembly Line Balancing (GALBP). The first group determines the tasks assigned to a set of workstations with the same cycle time; each task has a deterministic duration and must be performed in only one of the workstations. Two goals can be considered in addition to the precedence relations between the tasks: the minimization of the number of workstations for a given cycle time (SALBP-I) and the minimization of the cycle time for a given number of workstations (SALBP-II). Any other variation of the problem is included in the second group. U type assembly line

balancing is one of the generalizations of the simple assembly line balancing problem that is studied in this paper. The U-shaped ALBP is introduced and modeled by Miltenburg and Wijngaard [27]. SALBP-I can be solved with greedy heuristics, based on the application of priority rules to assign the tasks to the workstations. The rules consider such aspects as the duration of the tasks, the number of tasks after a given one, the constraints on the minimum number of workstations to assign tasks, etc., or even a mixture of these aspects. The rules are used to establish an ordered list with the possible tasks, so in each selection the most suitable task (according to the chosen rule) can be chosen. A traditional line organizes stations and the tasks that comprise them sequentially along a straight line. In a U-shaped line, tasks are arranged around a U shape line and are organized into stations that can cross from one side of the line to the other. The assignment of the tasks to the stations on a U-line exploits the geometry of the line to keep the return and crossover distances as small as possible. Consequently, total travel distance and, hence, travel time is less on a U-shaped line. This is shown with the U-shaped layout in Figure 1 where operator one (WS-2) performs tasks (2, 3) on the front side of the line, travels to the back side to complete tasks (9, 8), and then returns to the front side of the line to begin the next cycle. As it can be seen from Figure 1, the U-shaped line allows more possibilities on how to assign tasks to workstations; the number of stations needed for a U-shaped line layout is never more than the number of stations needed for the traditional straight line [1]. The reason for this is that in the traditional assembly line balancing problem, for a given workstation, the set of possible assignable tasks is conformed by those tasks whose predecessors have already been assigned to workstations, whereas in the U-shaped line problem the set of assignable tasks is determined by all those tasks whose predecessors or successors have already been assigned [1]. Note that a traditional line can be considered a special case of the Uline. The geometry of U-shaped line is presented in Figure 1.



Figure 1. U-shaped assembly line

As mentioned previously, solving large scale ALBP optimally by using exact and classical optimization procedures is very difficult. To overcome the difficulties imposed by the classical techniques, researchers recently developed effective heuristics to solve the ALBPs. Most of these heuristics are based on genetic algorithms, tabu search and simulated annealing. In each of these heuristics, different strategies are used to represent ALBP solutions and neighbor generation mechanisms. Kim *et al.* [24], Chan *et al.* [6], Sabuncuoglu *et al.* [28], Goncalves and Almeida [18] developed genetic algorithms based heuristics for simple ALBP. Chiang [7], Scholl and Voss [30] applied tabu search heuristic to solve simple ALBP. Boctor [4] proposed several heuristic procedures for solving simple ALBP. Helgeson and Birnie [22] proposed the well known ranked positional weight task assignment heuristic for ALB problems. Ajenblit and Wainwright

[1] developed genetic algorithms based heuristics for solving U-shaped ALBPs. Erel *et al.* [16] proposed a simulated annealing procedure for U-shaped ALBP. Scholl and Klein [32] proposed several effective heuristic procedures for balancing U-shaped lines. McMullen and Tarasewich [26] recently proposed an ant colony based algorithm for ALBPs with parallel workstations, stochastic task durations, and mixed-models. After the literature review, it is observed that genetic algorithms are more frequently used in modeling and solving simple and U-shaped ALBPs than the other meta-heuristics. The number of studies on U-shaped ALBPs is limited in comparison to simple ALBPs.

In this research paper, an ACO based algorithm is proposed for solving simple and U-shaped ALBPs with the aim of maximizing the line performance (or minimizing the number of workstations). The proposed method integrates COMSOAL algorithm [2] and Ranked Positional Weight Heuristic [22] within an ACO based heuristic. The proposed algorithm is tested and compared with several literature test problems with different cycle times. The proposed algorithm found promising solutions for each problem in short computational times.

2. ANT COLONY BASED ALGORITHM FOR SIMPLE AND U-TYPE ALBP

ACO algorithms are becoming popular approaches for solving combinatorial optimization problems in the literature. They were first introduced by Dorigo et al. [10-12]. Several sophisticated versions have been proposed to improve its performance. For a comprehensive review on ant algorithms, refer to Dorigo, Bonabeau and Theraulaz [14]. The ant system has been applied to the job shop scheduling problem by Colorni et al. [8], to the graph colouring problem by Costa and Hertz [9], to the quadratic assignment problem by Maniezzo [25] and to the vehicle routing problem by Bullnheimer et al. [5]. The fundamental idea of ACO heuristics is based on the behaviour of natural ants that succeed in finding the shortest paths from their nest to food sources by communicating via a collective memory that consists of pheromone trails. Due to ant's weak global perception of its environment, an ant moves essentially at random when no pheromone is available. However, it tends to follow a path with a high pheromone level when many ants move in a common area, which leads to an autocatalytic process. Finally, the ant does not choose its direction based on the level of pheromone exclusively, but also takes the proximity of the nest and of the food source respectively into account. This allows the discovery of new and potentially shorter paths. The described behavioural mechanism can be used to solve combinatorial optimization problems by simulation, with artificial ants searching the solution space instead of real ants searching their environment. In addition to this, the objective values correspond to the quality of the discovered food and an adaptive memory is the equivalent of the pheromone trails [13]. To guide their search through the set of feasible solutions, the artificial ants are furthermore equipped with a local heuristic function. The ant colony heuristic approach, like, genetic algorithms and simulated annealing, is attractive to many researchers since its optimization scheme is based on natural metaphors. But we should mention here that for successful application of the ant algorithms to the optimization problems, it should be easy and/or possible to define (or present) the solution space as a network (or graph). Luckily solution spaces of most of the combinatorial optimization problems especially the ones with binary decision

variables like the ALBP can be described (and/or mapped) graphically and solution procedures based on networks can be developed. In the following sub-sections, we will present the application of the ACO to the simple and U-shaped ALBP. In the proposed algorithm search mechanisms of simple and U-shaped lines are nearly same. The main difference is in the selection of task for possible workstation assignment. In the following sub-section the simple case is described first, afterwards the differences between simple and U-shaped lines are explained. The following notation is used in developing the ACO based algorithm:

| n | : | Number of task |
|------------|----|---|
| CS | : | Ant colony size |
| т | : | Index for ant $(1 \le m \le cs)$ |
| i | : | Index for task $(1 \le i \le n)$ |
| k | : | Index for station $(1 \le k \le z)$ |
| $cl={}$ | : | Candidate task(s) list. |
| пс | : | Number of candidate task in candidate list. |
| tf | : | Total pheromone quantity |
| gf(i,k) | : | Global pheromone quantity for assigning i^{th} task to k^{th} station |
| pw(i) | : | <i>i</i> th task positional weight |
| r(i) | : | <i>i</i> th task selection probability |
| p(i) | : | Cumulative probability $(l \rightarrow i)$ |
| x(m,i,k) | : | Binary decision variable (1, If i^{th} task is assigned to k^{th} station by m^{th} |
| | | ant; 0, otherwise) |
| xb(i,k) | : | Binary decision variable (representing the best solution) |
| of(m) | : | m^{th} ant's objective function value |
| α | : | Important rates of global pheromone quantity of decision $0 < \alpha \le 1$ |
| β | : | Important rates of decision without pheromone $0 < \beta \le 1$ |
| δ | : | Important rates of positional weight* $1 \le \delta < 10$ |
| * Position | al | weight of a task is determined based on the cumulative assembly time associated |

* Positional weight of a task is determined based on the cumulative assembly time associated with itself and its successors. Tasks are then assigned in this order to the lowest numbered feasible workstation [34].

Steps of the ACO based algorithm

The steps of the proposed algorithm are given in the following pseudo code.

End

Only the step 4 of the proposed algorithm that is outlined above is modified in order to solve U-shaped line. In the case of U-shaped lines task assignment to workstations can be made from both ends of the precedence graphs. Therefore, in forming candidate list for task assignment, (in step 4) immediate predecessors and immediate successor are determined by scanning both ends of the precedence diagram. Other steps are identical. Opening a station and forming the feasible candidate lists for possible assignment in steps 3 and 4 is the COMSOAL process. In the original COMSOAL process task selection for station assignment is made randomly, however in ACO process this is achieved by pheromone quantity update and ranked positional weight values as shown in step 5. This is the process of integrating ACO with COMSOAL [2] and Ranked Positional Weight [22] heuristics. An example is given here in order to show some of the solution steps of the proposed algorithm. The precedence diagram for example problem is shown in Figure 2. The cycle time is set to 70 and optimal solution is 3 workstations. The positional weights are shown in Table 1.



Figure 2. Precedence diagram for the 12-task example problem

| Task No | Positional Weight | Task No | Positional Weight | Task No | Positional Weight |
|---------|----------------------|---------|----------------------|------------|----------------------|
| 1 | 138 | 5 | 85 | 9 | 77 |
| 2 | 118 | 6 | 97 | 10 | 67 |
| 3 | 112 | 7 | 102 | 11 | 62 |
| 4 | 123 | 8 | 87 | 12 | 16 |

Table 1. Positional weights of the tasks

Some of the steps of the proposed algorithm for the simple line case are given as follows:

```
Steps 1-3: \alpha=1, \beta=1, \delta=10; m=1; k=1,
Step 4
            : Candidate list: cl={1, 4, 5, 6}, nc=4
Step 5
             : qf(1,1) = qf(4,1) = qf(5,1) = qf(6,1) = 0 \implies tf = 0
                tpw =138+123+85+97=443
               r(1) = [qf(1,1)*1+[138*10/443]+1]/[(1*0)+(1*4)+10]=0.29,
               r(4)=0.27, r(5)=0.21, r(6)=0.23
Step 6
             : 1 \rightarrow p(1) = 0.29, 4 \rightarrow p(2) = 0.56, 5 \rightarrow p(3) = 0.77, 6 \rightarrow p(4) = 1
                q=0.705 \Rightarrow p(2) < q < p(3) \Rightarrow p(3) \rightarrow 5^{th} task is assigned to 1^{st}
                station
Step 7
             : x(1, 5, 1) = 1
             : gf(5,1) = gf(5,1) + x(1,5,1) = 1
Step 8
```

Assume that tasks 5,4,1,2,3,10 are assigned to the 1st station by continuing the same procedure;

```
x(1,4,1)=1, gf(4,1)=1; x(1,1,1)=1, gf(1,1)=1; x(1,2,1)=1, gf(2,1)=1; x(1,3,1)=1, gf(3,1)=1; x(1,10,1)=1, gf(10,1)=1
```

 1^{st} workstation remaining time=70-(8+21+20+6+5+5)=5, there isn't any task left which can be assigned to the 1^{st} station, so a new station is opened. k=k+1, k=2. The other tasks are assigned by using the same procedure. After all tasks are assigned, the best solution is updated if an ant finds a better solution than the global optimum solution and so on.

3. COMPUTATIONAL STUDY

In order to present the efficiency of the proposed ACO based algorithm, two sets of test problem are solved. In the first set, there are 64 test problems and these problems are collected by Talbot *et al.* [34]. In the second set there are 168 test problems and

these problems are collected by Scholl [29]. The proposed algorithm is programmed in Visual Basic in a Pentium III-MMX model PC at 450 MHz (128 MB RAM). All of these test problems are available in <u>http://www.assembly-line-balancing.de/</u>. The ACO parameters α , β , δ are usually set to 1,1,10. The results for the first set of problem are tabulated in Table 2. The results for the second set of problems are tabulated in Table 3. In Table 4 a comparison is also given with the branch-and-bound algorithm of Scholl and Klein [31]. There are several other papers that also solved U-type ALB problems but these papers tested their algorithms with only Talbot *et al.*'s data set [34]. We exclude these comparisons from this paper as most of the algorithms are successful with Talbot *et al.*'s data set [34].

| Problem | Cycle | Known | ACO | ACO | ACO Simple | ACO U-line |
|-------------|-------|-------------|----------|--------|---------------|---------------|
| Name | Time | optimum for | Simple | U-line | Computational | Computational |
| | | simple line | Result | Result | Time | Time |
| Mitchell | 15 | 8 | 8 | 8 | 0,1 | 0,6 |
| | 14 | 8 | 8 | 8 | 0,1 | 0,01 |
| | 21 | 5 | 5 | 5 | 0,1 | 3,29 |
| | 26 | 5 | 5 | 5 | 0,1 | 0,01 |
| | 35 | 3 | 3 | 3 | 1 | 0,11 |
| | 39 | 3 | 3 | 3 | 0,1 | 0,6 |
| Arcus 111 | 5755 | 27 | 27 | 27 | 4 | 40,1 |
| | 8847 | 18 | 18 | 18 | 2 | 2,97 |
| | 10027 | 16 | 16 | 16 | 2 | 2,91 |
| | 10743 | 15 | 15 | 15 | 5 | 2,96 |
| | 11378 | 14 | 14 | 14 | 3 | 2,91 |
| | 17067 | 9 | 9 | 9 | 4 | 2,91 |
| Arcus 83 | 5048 | 16 | 16 | 16 | 2 | 1,31 |
| | 5853 | 14 | 14 | 14 | 0,1 | 1,27 |
| | 6842 | 12 | 12 | 12 | 1 | 1,26 |
| | /5/1 | 10 | 10 | 10 | 1 | 1,32 |
| | 8412 | 10 | 10 | 10 | 0,1 | 1,26 |
| | 8898 | 9 | 9 | 9 | 0,1 | 1,32 |
| Tanga | 10816 | 8 | <u>8</u> | 8 | 0,1 | 1,20 |
| Tonge | 264 | 10 | 10 | 21 | 0.1 | 0,82 |
| | 410 | 0 | 10 | 10 | 0,1 | 0,83 |
| | 410 | 8 | 9 | 9 | 0,1 | 0.82 |
| | 527 | 7 | 7 | | 0,1 | 0,82 |
| K & Wester | 57 | 10 | 10 | 10 | 0,1 | 0.22 |
| R. & Wester | 79 | 7 | 7 | 7 | 0,1 | 154 |
| | 92 | 6 | 6 | 6 | 3 | 36.2 |
| | 110 | 6 | 6 | 6 | 0.1 | 0.28 |
| | 138 | 4 | 4 | 4 | 5 | 0,76 |
| | 184 | 3 | 3 | 3 | 1 | 2,8 |
| Mansoor | 48 | 4 | 4 | 4 | 0,02 | 0,42 |
| | 62 | 3 | 3 | 3 | 0,03 | 0,44 |
| | 94 | 2 | 2 | 2 | 0,01 | 0,01 |
| Sawyer | 25 | 14 | 14 | 14 | 1 | 0,49 |
| | 27 | 13 | 13 | 13 | 1 | 0,6 |
| | 30 | 12 | 12 | 12 | 0,1 | 0,11 |
| | 36 | 10 | 10 | 10 | 0,1 | 0,11 |
| | 41 | 8 | 8 | 8 | 0,1 | 2,89 |
| | 54 | 7 | 7 | 7 | 0,1 | 0,11 |
| | 75 | 5 | 5 | 5 | 0,1 | 0,6 |
| Heskiaoff | 138 | 8 | 8 | 8 | 0,1 | 0,11 |
| | 205 | 5 | 5 | 5 | 3 | 6,98 |
| | 216 | 5 | 5 | 5 | 0,1 | 0,5 |
| | 256 | 4 | 4 | 4 | 0,1 | 9,34 |

Table 2. Results for the simple and U type ALBP (Talbot *et al.*'s data set [34])

| | 324 | 4 | 4 | 4 | 0,1 | 0,6 |
|----------|-----|---|---|---|-----|------|
| | 342 | 3 | 3 | 3 | 0,1 | 2,8 |
| Jackson | 7 | 8 | 8 | 7 | 0,1 | 0,2 |
| | 9 | 6 | 6 | 6 | 0,1 | 0,01 |
| | 10 | 5 | 5 | 5 | 0,1 | 0,01 |
| | 13 | 4 | 4 | 4 | 1 | 0,01 |
| | 14 | 4 | 4 | 4 | 0,1 | 0,01 |
| | 21 | 3 | 3 | 3 | 0,1 | 0,01 |
| Jaeschke | 6 | 8 | 8 | 8 | 0,1 | 0,01 |
| | 7 | 7 | 7 | 7 | 0,1 | 0,01 |
| | 8 | 6 | 6 | 6 | 0,1 | 0,5 |
| | 10 | 4 | 4 | 4 | 0,1 | 0,01 |
| | 18 | 3 | 3 | 3 | 0,1 | 0,01 |
| Bowman | 20 | 5 | 5 | 4 | 0,1 | 0,3 |
| Merten | 6 | 6 | 6 | 6 | 0,1 | 0,01 |
| | 7 | 5 | 5 | 5 | 0,1 | 0,01 |
| | 8 | 5 | 5 | 5 | 0,1 | 0,6 |
| | 10 | 3 | 3 | 3 | 0,1 | 0,01 |
| | 15 | 2 | 2 | 2 | 0,1 | 0,01 |
| | 18 | 2 | 2 | 2 | 0,1 | 0,01 |

As it can be seen from Table 4 the proposed ACO algorithm is very successful with Talbot *et al.*'s data set [34]. The algorithm found all of the optimal solutions with small computational times. The ACO algorithm is also effective with Scholl's data set [29]. Although the number of optimal solutions found by ACO for the simple ALB is relatively smaller than FABLE, FEUR and FSLM, the computational time requirements are smaller than the compared algorithms. The performance of ACO for U-lines is also comparable to branch-and bound algorithms. ACO U-line is able to reach good solutions with small smaller computational times.

| Problem Name | Cycle Time | Known optimum for simple line | ACO Simple Result | ACO U-line Result | ACO Simple Comp. Time | ACO U-line Compt. Time | Problem Name | Cycle Time | Known optimum for simple line | ACO Simple Result | ACO U-line Result | ACO Simple Comp. Time | ACO U-line Compt. Time |
|-----------------|---------------|---|-------------------------|-------------------------|--------------------------------|---------------------------------|-----------------|---------------|---|-------------------------|-------------------------|--------------------------------|---------------------------------|
| BARTHOL2 | 84 | 51 | 53 | 53 | 27,93 | 364,89 | WARNECKE | 54 | 31 | 33 | 32 | 2,11 | 23,12 |
| | 85 | 50 | 53 | 53 | 27,94 | 364,41 | | 56 | 29 | 31 | 31 | 2,08 | 22,92 |
| | 87 | 49 | 52 | 51 | 28,02 | 93,74 | | 58 | 29 | 31 | 30 | 2,10 | 22,94 |
| | 89 | 48 | 50 | 50 | 28,01 | 93,43 | | 60 | 27 | 30 | 29 | 2,09 | 22,91 |
| | 91 | 47 | 49 | 49 | 27,92 | 93,51 | | 62 | 27 | 29 | 28 | 2,08 | 22,88 |
| | 93 | 46 | 49 | 48 | 28,05 | 93,39 | | 65 | 25 | 28 | 27 | 2,09 | 22,91 |
| | 95 | 45 | 47 | 47 | 27,89 | 93,40 | | 68 | 24 | 26 | 25 | 2,08 | 22,90 |
| | 97 | 44 | 46 | 46 | 27,91 | 93,42 | | 71 | 23 | 25 | 24 | 2,08 | 22,94 |
| | 99 | 43 | 45 | 45 | 27,93 | 93,37 | | 74 | 22 | 23 | 23 | 2,07 | 22,89 |
| | 101 | 42 | 45 | 44 | 28,02 | 93,35 | | 78 | 21 | 22 | 21 | 2,08 | 8,52 |
| | 104 | 41 | 43 | 43 | 27,93 | 93,48 | | 82 | 20 | 21 | 20 | 2,08 | 1,91 |
| | 106 | 40 | 42 | 42 | 27,98 | 93,39 | | 86 | 19 | 20 | 19 | 2,08 | 3,50 |
| | 109 | 39 | 41 | 41 | 27,91 | 93,38 | | 92 | 17 | 19 | 18 | 2,09 | 22,87 |
| | 112 | 38 | 40 | 39 | 27,89 | 93,43 | | 97 | 17 | 17 | 17 | 0,96 | 1,67 |
| | 115 | 37 | 39 | 39 | 27,94 | 93,48 | | 104 | 15 | 16 | 16 | 2,09 | 22,92 |
| | 118 | 36 | 38 | 37 | 27,98 | 93,46 | | 111 | 14 | 15 | 15 | 2,08 | 22,92 |
| | 121 | 35 | 37 | 37 | 28,02 | 93,58 | WEEMAG | 28 | 63 | 63 | 63 | 0,46 | 0,76 |
| | 125 | 34 | 35 | 35 | 27,91 | 93,39 | | 29 | 63 | 63 | 63 | 0,39 | 0,62 |
| | 129 | 33 | 34 | 34 | 27,90 | 93,57 | | 30 | 62 | 62 | 62 | 0,80 | 8,42 |
| | 133 | 32 | 33 | 33 | 27,96 | 93,59 | | 31 | 62 | 62 | 62 | 0,33 | 0,63 |
| | 137 | 31 | 32 | 32 | 27,95 | 93,44 | | 32 | 61 | 61 | 61 | 0,53 | 2,80 |
| | 142 | 30 | 31 | 31 | 27,93 | 93,51 | | 33 | 61 | 61 | 61 | 0,14 | 1,12 |
| | 146 | 29 | 30 | 30 | 27,86 | 93,57 | | 34 | 61 | 61 | 61 | 0,15 | 0,62 |
| | 152 | 28 | 29 | 29 | 27,97 | 93,48 | | 35 | 60 | 60 | 60 | 0,21 | 2,57 |
| | 157 | 27 | 28 | 28 | 27,97 | 93,44 | | 36 | 60 | 60 | 60 | 0,14 | 0,63 |
| | 163 | 26 | 27 | 27 | 28,00 | 93,53 | | 37 | 60 | 60 | 60 | 0,14 | 0,62 |
| | 170 | 25 | 26 | 26 | 28,12 | 93,50 | | 38 | 60 | 60 | 60 | 0,14 | 0,62 |
| BUXEY | 27 | 13 | 14 | 13 | 0,37 | 0,39 | | 39 | 60 | 60 | 60 | 0,14 | 0,63 |

Table 3. Results for the simple and U type ALBP (Scholl's data set [29])

| | | 30 | 12 | 12 | 12 | 0,03 | 0,70 | | 40 | 60 | 60 | 60 | 0,14 | 0,62 |
|---|----------|------|----|----|----|--------------|-------|----------|------|----|----|----|--------|--------|
| | | 33 | 11 | 11 | 11 | 0,03 | 0,12 | | 41 | 59 | 59 | 59 | 0,21 | 0,63 |
| | | 36 | 10 | 10 | 10 | 0.02 | 0.50 | | 42 | 55 | 55 | 55 | 0 15 | 1 10 |
| | | 41 | 8 | 8 | 9 | 0.29 | 3 23 | | 43 | 50 | 50 | 50 | 0.26 | 0.62 |
| | | 47 | 7 | 8 | 7 | 0.37 | 2 44 | | 45 | 38 | 30 | 30 | 4 30 | 48 48 |
| | | 54 | 7 | 7 | 7 | 0,01 | 2,44 | | 46 | 24 | 26 | 27 | 4,55 | 40,40 |
| | | 34 | 14 | 14 | 14 | 0,01 | 0,50 | | 40 | 34 | 30 | 37 | 4,17 | 40,00 |
| | GUNTHER | 41 | 14 | 14 | 14 | 0,45 | 0,14 | | 47 | 33 | 34 | 34 | 4,18 | 48,53 |
| | | 44 | 12 | 12 | 12 | 0,03 | 0,30 | | 49 | 32 | 33 | 32 | 4,23 | 4,00 |
| | | 49 | 11 | 11 | 11 | 0,03 | 0,15 | | 50 | 32 | 32 | 32 | 0,12 | 1,80 |
| | | 54 | 9 | 10 | 10 | 0,57 | 5,43 | | 52 | 31 | 32 | 31 | 4,22 | 24,37 |
| | | 61 | 9 | 9 | 9 | 0,03 | 0,80 | | 54 | 31 | 31 | 31 | 0,17 | 0,60 |
| | | 69 | 8 | 8 | 8 | 0,03 | 0,14 | | 56 | 30 | 30 | 31 | 0,69 | 48,72 |
| | | 81 | 7 | 7 | 7 | 0,01 | 0,80 | BARTHOLD | 403 | 14 | 15 | 15 | 28,06 | 364,52 |
| | LUTZ1 | 1414 | 11 | 11 | 11 | 0,10 | 0,80 | | 434 | 13 | 14 | 14 | 28,05 | 365,27 |
| | | 1572 | 10 | 10 | 10 | 0,06 | 0,70 | | 470 | 12 | 13 | 13 | 28,07 | 365,28 |
| | | 1768 | 9 | 9 | 9 | 0.03 | 0.70 | | 513 | 11 | 12 | 12 | 28.10 | 365.32 |
| | | 2020 | 8 | 8 | 8 | 0.02 | 0.80 | | 564 | 10 | 11 | 11 | 28.06 | 365.29 |
| | | 2357 | 7 | 7 | 7 | 0.01 | 0.60 | | 626 | 9 | 10 | 10 | 28.07 | 365,30 |
| | | 2828 | 6 | 6 | 6 | 0,01 | 0,00 | | 705 | 8 | 9 | 9 | 28.05 | 365.27 |
| | 111772 | 11 | 40 | 51 | 50 | 6,63 | 70.76 | | 805 | 7 | 8 | 8 | 28,00 | 365 30 |
| | LUIZZ | 10 | 49 | 40 | 10 | 0,00 | 79,70 | | 2004 | , | 0 | 0 | 20,09 | 0.00 |
| | | 12 | 44 | 40 | 40 | 0,02 | 79,70 | ПАПИ | 2004 | 0 | 0 | 0 | 0,00 | 0,26 |
| | | 13 | 40 | 43 | 41 | 0,00 | 79,75 | | 2330 | / | 0 | 1 | 1,65 | 0,25 |
| | | 14 | 37 | 39 | 37 | 6,61 | 36,79 | | 2806 | 6 | 6 | 6 | 0,05 | 0,26 |
| | | 15 | 34 | 35 | 34 | 6,60 | 10,59 | | 3507 | 5 | 5 | 5 | 0,06 | 0,26 |
| | | 16 | 31 | 34 | 33 | 6,64 | 79,73 | | 4676 | 4 | 4 | 4 | 0,06 | 0,26 |
| | | 17 | 29 | 31 | 30 | 6,60 | 79,60 | SCHOLL | 1394 | 50 | 53 | 52 | 209,07 | 2982 |
| | | 18 | 28 | 29 | 29 | 6,60 | 79,81 | | 1422 | 50 | 52 | 51 | 209,03 | 2982 |
| | | 19 | 26 | 27 | 27 | 6,60 | 79,58 | | 1452 | 48 | 51 | 50 | 208,88 | 789,80 |
| | | 20 | 25 | 26 | 25 | 6,60 | 49,53 | | 1483 | 47 | 50 | 49 | 208,91 | 789,00 |
| | | 21 | 24 | 25 | 24 | 6,61 | 71,82 | | 1515 | 46 | 48 | 48 | 208,71 | 789,10 |
| | LUTZ3 | 75 | 23 | 24 | 23 | 6,57 | 2,10 | | 1548 | 46 | 47 | 47 | 208,63 | 788,97 |
| | | 79 | 22 | 23 | 22 | 6,57 | 2,66 | | 1584 | 44 | 46 | 45 | 208,51 | 789,11 |
| | | 83 | 21 | 22 | 21 | 6,57 | 1,87 | | 1620 | 44 | 45 | 44 | 208,08 | 286,10 |
| | | 87 | 20 | 21 | 20 | 6,57 | 2,64 | | 1659 | 42 | 44 | 43 | 208,44 | 788,53 |
| | | 92 | 19 | 19 | 19 | 3,59 | 1,85 | | 1699 | 42 | 43 | 42 | 208,58 | 19,16 |
| | | 97 | 18 | 18 | 18 | 0.21 | 1.70 | | 1742 | 40 | 42 | 41 | 208.53 | 788.47 |
| | | 103 | 17 | 17 | 17 | 0.41 | 1 70 | | 1787 | 39 | 41 | 40 | 208 64 | 788 80 |
| | | 110 | 15 | 16 | 16 | 6.55 | 79.68 | | 1834 | 38 | 40 | 39 | 208.95 | 788.97 |
| | | 118 | 14 | 15 | 15 | 6 58 | 79.68 | | 1883 | 37 | 30 | 38 | 208.87 | 788.98 |
| | | 127 | 14 | 14 | 14 | 0,00 | 1.60 | | 1035 | 36 | 38 | 37 | 208,07 | 780.12 |
| | | 137 | 13 | 13 | 13 | 0,40 | 1,00 | | 1901 | 35 | 37 | 36 | 208,70 | 780 10 |
| | | 150 | 10 | 10 | 10 | 0.21 | 1 70 | | 2040 | 34 | 35 | 25 | 200,07 | 780 10 |
| | | 130 | 12 | 12 | 12 | U,ZI 7,70 | 1,70 | | 2049 | 34 | 30 | 35 | 200,40 | 700,10 |
| | WUKHERJE | 1/0 | 25 | 20 | 25 | 1,13 | 37,71 | | 2111 | 33 | 34 | 34 | 208,46 | 789,30 |
| | | 183 | 24 | 25 | 24 | 7,81 | 4,89 | | 21// | 32 | 33 | 33 | 208,57 | 789,40 |
| | | 192 | 23 | 24 | 23 | 7,82 | 1,16 | | 2247 | 31 | 32 | 32 | 209,18 | 19,30 |
| | | 201 | 22 | 23 | 22 | 7,80 | 2,10 | | 2322 | 30 | 31 | 31 | 208,63 | 19,21 |
| | | 211 | 21 | 22 | 21 | 7,82 | 2,11 | | 2402 | 29 | 30 | 30 | 209,24 | 19,26 |
| | | 222 | 20 | 21 | 20 | 7,80 | 2,10 | | 2488 | 28 | 29 | 29 | 209,39 | 19,20 |
| | | 234 | 19 | 19 | 19 | 2,51 | 1,17 | | 2580 | 27 | 28 | 28 | 209,35 | 19,20 |
| | | 248 | 18 | 18 | 18 | 0,67 | 1,17 | | 2680 | 26 | 27 | 27 | 209,36 | 19,21 |
| | | 263 | 17 | 17 | 17 | 0,32 | 1,17 | | 2787 | 25 | 26 | 26 | 209,15 | 19,20 |
| | | 281 | 16 | 16 | 16 | 0,32 | 1,17 | ROSZIEG | 14 | 10 | 10 | 10 | 0,02 | 0,70 |
| | | 301 | 15 | 15 | 15 | 0,21 | 1,17 | | 16 | 8 | 9 | 8 | 0,25 | 0,28 |
| | | 324 | 14 | 14 | 14 | 0,21 | 1,17 | | 18 | 8 | 8 | 8 | 0,01 | 0,30 |
| | | 351 | 13 | 13 | 13 | 0,21 | 1,17 | | 21 | 6 | 6 | 6 | 0,02 | 0,70 |
| | | | | | | | | | 25 | 6 | 6 | 6 | 0,00 | 0,40 |
| | | | | | | | | | 32 | 4 | 4 | 4 | 0,02 | 0,40 |
| J | | | | | | | | | | | | | | |

5. CONCLUSIONS

In this paper, a new algorithm is developed for solving simple and U-shaped ALBPs. The proposed algorithm integrates COMSOAL method, Ranked Positional Weight heuristic and an ACO based heuristic. The algorithm is able quickly search effective solutions for simple and U-shaped ALBPs. The performance of the proposed algorithm is tested with several test problems from the literature. In most of the runs, the proposed algorithm found optimal solutions in short computational times. It is concluded after this research that the proposed ACO is an eligible meta-heuristic for solving ALBPs. The ACO algorithm can also be used to optimize other types of

assembly lines like parallel lines, two sided assembly lines with some problem specific modifications. This is scheduled as a future work. Application of the proposed heuristic to stochastic and mixed model assembly line balancing problems can also be considered as a useful future work.

| Table 4. Comparison of results with Scholl and Klein's Branch-and-Bound algorithms; |
|---|
| FABLE, FEUR and FSLM [29] |

| | | Talbot | et al. [34] | (64 inst.) | | Scholl [29] (128 inst.) | | | | |
|----------------------|-------|--------|-------------|---------------|---------------|-------------------------|-------|-------|---------------|---------------|
| | FABLE | FEUR | FSLM | ACO Simple | ACO U-Line | FABLE | FEUR | FSLM | ACO Simple | ACO U-Line |
| # of Opt. Results | 64 | 63 | 64 | 64 | 64 | 80 | 69 | 100 | 59 | 82 |
| Avg. CPU(sec.) | 0.2 | 9 | 0.5 | 0.77 | 4.6 | 267.1 | 333.4 | 209.3 | 39.53 | 153.3 |

REFERENCES

- 1. D.A. Ajenblit, R.L. Wainwright, Applying genetic algorithms to the U-shaped assembly line balancing problem. Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, pp. 96-101, 1998.
- 2. A.L. Arcus, COMSOAL: A computer method of sequencing operations for assembly lines, *International Journal of Production Research* **4**, 259-277, 1966.
- 3. I. Baybars, A survey of exact algorithms for the simple assembly line balancing Problem, *Management Science* **32**, 909-932, 1986.
- 4. F.F. Boctor, A multiple-rule heuristic for assembly line balancing, *Journal of Operational Research Society* **46**, 62–69, 1995.
- 5. B. Bullnheimer, R.F. Hartl, C. Strauss, A new rank-based version of the ant system: a computational study, *Central European Journal of Operations Research and Economics* 7, 25-38, 1999.
- 6. K.C.C. Chan, P.C.L. Hui, K.W. Yeung, F.S.F. Ng, Handling the assembly line balancing problem in the clothing industry using a genetic algorithm, *International Journal of Clothing Science and Technology* **10**, 21-37, 1998.
- 7. W-C. Chiang, The application of a tabu search metaheuristic to the assembly line balancing problem, *Annals of Operations Research* **77**, 209-227, 1998.
- 8. A. Colorni, M. Dorigo, V. Maniezzo, M. Trubian, Ant system for job-shop Scheduling, *JORBEL-Belgian Journal of Operations Research, Statistics and Computer Science* **34**, 39-53, 1994.
- 9. D. Costa, A. Hertz, Ants can color graphs, *Journal of the Operational Research Society* **48**, 295-305, 1997.
- 10. M. Dorigo, V. Maniezzo, A. Colorni, Positive feedback as a search strategy, Technical Report, 91-016, Politecnico idi Milano, 1991.
- 11. M. Dorigo, Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale (Optimization, Learning and Natural Algorithms), Ph.D.Thesis, Politecnico di Milano, Italy, in Italian, 1992.
- M. Dorigo, V. Maniezzo, A., Colorni, The ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics-Part B* 26, 29-41, 1996.

- 13. M. Dorigo, G. Di Caro, L.M. Gambardella, Ant algorithms for discrete optimization, *Artificial Life* **5**, 137-72, 1999.
- 14. M. Dorigo, E. Bonabeau, G. Theraulaz, Ant algorithm and stigmergy, *Future Generation Computer Systems* **16**, 851-871, 2000.
- 15. E.A. Elsayed, T.O. Boucher, *Analysis and Control of Production Systems*, New Jersey: Prentice Hall, 1994.
- 16. E. Erel, I. Sabuncuoglu, B.A. Aksu, Balancing of U-type assembly systems using simulated annealing, *Int. Journal of Production Research* **39**, 3003-3015, 2001
- 17. S. Ghosh, R.J. Gagnon, A comprehensive literature review and analysis of the design, balancing and scheduling of assembly lines, *International Journal of Production Research* **27**, 637-670, 1989.
- 18. J.F. Goncalves, J.R.D. Almeida, A hybrid genetic algorithm for assembly line Balancing, *Journal of Heuristics* **8**, 629–642, 2002.
- 19. H. Gokcen, E. Erel, Binary integer formulation for mixed model assembly line balancing problem, *Computers and Industrial Engineering* **34**, 451-461, 1998.
- H. Gokcen, K. Agpak, C. Gencer, E. Kizilkaya, A shortest route formulation of simple U-type assembly line balancing problem, *Applied Mathematical Modelling* 29, 373-380, 2005.
- 21. H. Gokcen, K. Agpak, A goal programming approach to simple U-line balancing Problem, *European Journal of Operational Research* **171**, 577-585, 2006.
- 22. W. Helgeson, D. Birnie, Assembly line balancing using the ranked positional weight Technique, *Journal of Industrial Engineering* **12**, 394–398, 1961.
- 23. R.M. Karp, Reducibility Among Combinatorial Problems, New York, 1972.
- 24. Y.J. Kim, Y.K. Kim, Y. Cho, A heuristic-based genetic algorithm for workload smoothing in assembly lines, *Computers & Operations Research* **25**, 99-111,1998.
- 25. V. Maniezzo, *Exact and approximate non-deterministic tree-search procedures for the quadratic assignment problem*, Research Report, CSR 98-1, Scienze dell'Informazione, University Di Bologna, Sede Di Cesena, Italy, 1998.
- 26. P.R. McMullen, P. Tarasewich, Using ant techniques to solve the assembly line balancing problem, *IIE Transactions* **35**, 605–617, 2003.
- 27. G.J. Miltenburg, J. Wijngaard, The U-line line balancing problem, *Management Science* **40**, 1378-1988, 1994.
- 28. I. Sabuncuoglu, E. Erel, M. Tanyer, Assembly line balancing using genetic Algorithms, *Journal of Intelligent Manufacturing* **11**, 95-310, 2000.
- 29. A. Scholl, *Data of Assembly Line Balancing Problems*, Schriften zur Quantitativen Betriebswirtschaftslehre 16/93, TU Darmstadt, 1993.
- 30. A. Scholl, S. Voss, Simple assembly line balancing-heuristic approaches, *Journal* of *Heuristics*, **2**, 217–244, 1996.
- 31. A. Scholl, R. Klein, SALOME: A bidirectional branch-and-bound procedure for assembly line balancing, *INFORMS Journal on Computing* **9**, 319-334, 1997.
- 32. A. Scholl, R. Klein, ULINO: Optimally balancing U-shaped JIT assembly lines, *International Journal of Production Research* **37**, 721-736, 1999.
- 33. A. Scholl, C. Becker, State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, *Eur. J. of Oper. Research* **168**, 666-693, 2006.
- 34. F.B. Talbot, J.H. Patterson, W.V. Gehrlein, A comparative evaluation of heuristic Line balancing techniques, *Management Science* **32**, 430-454, 1986.