META HEURISTIC BASED SIMULATED ANEALING APPROACH FOR DESIGN OF U-SHAPED MANUFACTURING ASSEMBLY LINE BALANCING

V.Thangadurai
Research scholar,
Department of Mechanical Engineering, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India
Email:aegolddurai@gmail.com

Abstract

The objective of this study is to resolve the balancing problem comprising a shared U-shaped assembly line which is exclusively designed by a Simulated Annealing Algorithm. The proposed algorithm may be utilized in medium and large scale problems. This approach envisages an efficient mechanism which associates a large solution space search for revealing an optimal solution. The existing balancing problem is just a conventional straight assembly line, limits its application to production line where every tasks are grouped into workstations. Line balancing is a process that balances the tasks among various workstations that is based upon precedence relation. The concept of shared principle enhances its efficiency by reducing the number of workstations. When compared with the conventional assembly line, the U-shaped assembly line clearly stresses on the balancing problem by allocating the tasks in forward, backward, or in both directions with respect to the precedence relation. The efficiency revealed by SAA for the shared U-shaped assembly line has proved better when compared to existing lines. The Simulated Annealing Algorithm (SAA) heuristic approach is projected to solve the medium and large sized problems by suggesting two objectives concurrently (i) To reveal the optimal number of work stations and (ii) to find the unbalance time among workstations for a fixed cycle time. The proposed approach is elaborated with a model problem and its performance is scrutinized with a set of problems after comparing the results of SAA with model test problems available in the already published literature. The results of the experiments have revealed that the proposed SA-based algorithm outperforms with great effectiveness. The Future research scope and a comprehensive bibliography are also given.

Keywords: U-shaped assembly Line, Line balancing, Sharing, Multi-objective, Simulated Annealing Algorithm (SAA).

1. Introduction

SAA is a non-traditional optimisation technique that depends on a principle of metallurgy. As it mimics the principle of Annealing or slow cooling it is called Simulated Annealing Algorithm. It is robust and accurate for discrete algorithm problem. Metropolis and his colleagues had corroborated an algorithm based on annealing principle to simulate the solid to a thermal equilibrium. Krikpatrick et al. (1983) successfully notified the application of this algorithm to optimize a combinatorial problem. The SAA is thus validated in applying in the Assembly line work stations. Assembly line comprises of workstations that are arranged in an orderly fashion where the tasks are to convert raw material to a finished product.

As per G. J Miltenburg, a U-shaped assembly line possess several advantages when compared to a one-sided line, such as reduction of (i) the total number of operators, (ii) throughput time, (iii) the working cost of tools and fixtures, when they can be shared by the operators from both sides, (iv) line length of the assembly line and (v) the total number of workstations. An example has been used to clarify the above line concepts given in Fig. 1.

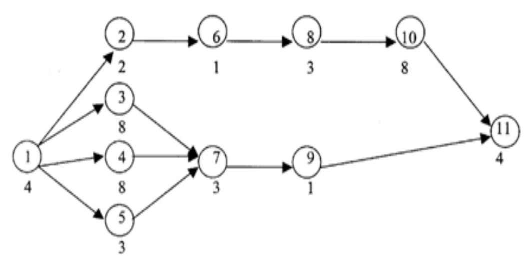

Fig. 1. Precedence diagram and the task times

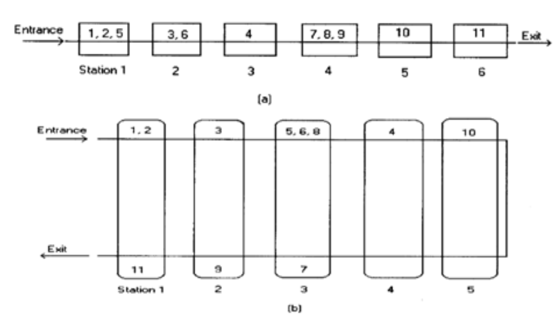

Fig. 2. Schematic view of (a) straight line and (b) U-line configurations

Depiction of Jackson's 11-task problem is given in Fig. 1 (Source-type assembly systems E.Erel et al, 2001). The numbers in and beneath the nodes predicts the tasks and the associated task times, respectively. Figure 2a shows a line assembly and Figure 2b shows a solution of the previous example in a U-type configuration.

2. Literature review

Colossal numbers of literatures have mooted the single model and multi model straight line assembly line balancing. However, the main study on U assembly line balancing problem was corroborated by Miltenburg and Wijigaard (1994).

Mitsuo Gen et al. (2017) have presented the proceeding of hybrid metaheuristic based optimization and proved that it was powerful tool to find optimal system settings to the stochastic manufacturing scheduling problems. Evolutionary algorithm (EA) in hybrid metaheuristics was a generic population-based metaheuristic, which can find compromised optimal solutions well for a complicated manufacturing scheduling problem. By using the hybrid sampling strategy-based EA (HSS-EA) and the multi-objective estimation of distribution algorithm (MoEDA), they surveyed several case studies such as stochastic multi-objective job shop scheduling problem (S-MoJSP), stochastic multi-objective assembly line balancing (S-MoALB) problem and stochastic multi-objective resource constrained project scheduling problem (S-MoRcPSP) with numerical experimental results and has proved that better efficacy and efficiency than existing NSGA-II, SPEA2 and awGA algorithms.

Yılmaz Delice et al. (2014) proposed a new modified particle swarm optimization algorithm with negative knowledge as proposed to solve the mixed-model two-sided assembly line balancing problem. The approach comprised new procedures such as generation procedure which is based on combined selection mechanism and decoding procedure. These new procedures enhance the solution capability of the algorithm while enabling it to search at different points of the solution space, efficiently. Performance of the proposed approach is tested on a set of test problem. The experimental results have revealed that the proposed approach have acquired distinguished results than the existing solution approaches.

Mukund Nilakantan et al. (2014) had worked in an assembly line that can be achieved using robots. An U-shaped assembly line balancing (RUALB), robots were assigned to workstations to perform the assembly tasks on a U-shaped assembly line. The robots were expected to perform multiple asks, because of their capabilities. U-shaped assembly line problems were derived from additional assembly line problems and are relatively new. Tasks were assigned to the workstations when either all of their predecessors or all of their successors have already been assigned to workstations. Finally, they have revealed that robotic U-shaped assembly lines perform better than robotic straight assembly lines with respect to the cycle times.

Ming Li et al. (2016) had made an approach based on multiple rules and an integer programming model. Three rules were systematically grouped together, which were task selection rules, task assignment rules and task exchange rules. The sufficient conditions for implementing exchange rules were proposed and proved. Thirteen small or medium scale benchmark issues including 63 instances were solved and the computational results has revealed that it offers very good performance in efficiency and effectiveness compared with that by integer programming. The computational results of eighteen examples including 121 instances shows that the task exchange rules have significantly improve computational accuracy of the traditional heuristic. Finally, 30 new standard instances produced by a systematic data generation were also effectively solved by the proposed approach.

The literature on the U-lines is very limited and new relations to the traditional straight lines. The research on U-lines can be divided into two groups: line balancing (ULB) and production flow lines. In the ULB group, the researchers study the problem of balancing U-type assembly systems to minimize the cycle time or the total number of stations. In the Production flow lines, the emphasis is on recognizing some important design factors and their impacts on the performance of U-type flow lines. The other literatures deliberately gives formidable platform for this research are presented briefly in this section.

N. Jawahar et al. (2014) claimed that the two, minimum number of workstations and minimum unbalance time among workstations, have been considered for balancing the assembly line. There are two approaches to solve multi-objective optimization problem: first approach combines all the objectives into a single composite function or moves all but one objective to the constraint set; second approach determines the Pareto optimal solution set. Also, proposed a comparison with a set of set problems which were illustrated with examples.

Miltenburg et al. (1994) corroborates a new problem derived from the traditional ALB problem where any production lines are allocated as U-type lines instead of straight lines. The U-type assembly line is a useful alternative for assembly production systems as the operators become multiskilled by outperforming tasks assigned on different parts of the assembly line. Moreover, the U-type line disposition enable the possibilities on how to allocate tasks to different workstations, the number of workstations required for a U-type line layout should not be greater than the number of workstations required for the conventional straight line. Miltenburg et al. (1995) presented three exact algorithms to solve the U assembly line balancing problem. However, in lean thinking of U line, it is essential to add walking time for operation times. The probability of increasing number of workers is obviously relying on walking time, but there is no such evidence available on how much walking time is required for changing the number of workers. Ajenblit et al. (1998) elaborated a Genetic Algorithm, proposed Simulated Annealing methodologies for large size U-line. In 1998, Urban presented an integer linear programming formulation for solving small to medium sized UALBP with up to 45 tasks. In 1999, Scholl et al developed a branch and bound procedure to resolve, either optimally or sub optimally problem with up to 297 tasks.

Ajenblit and Erel (1998) revealed a Genetic Algorithm and simulated annealing technique that depends on solution methodology for larger U -line. Nuchsara Kriengkorakot (2007) develops a balancing problem by computing the U-line balancing problem to a straight-line balancing problem.

In 1998, Sparling corroborated on a heuristic solution algorithms for a U-line facility comprising individual U-lines that operated sequentially at the same cycle time and connected with multiline stations. Travel time between tasks and U-lines were also considered and minimized. All these studies have revealed that the ULB is an predominant problem for current modern assembly systems. However, when considering the table (Source-type assembly systems E.Erel et al, IJPR 2001), up to 45-task problems can be optimally resolved as it is a complex nature of the problem.

. In this paper, such a procedure is proposed which can easily handle more than 100-task problems. After thoroughly gleaning the literatures, it may be emphasized that U assembly line balancing becomes a significant problem in the current system of production assembly. Though several methods have been proposed for solving the balancing problem that comprises of small number of tasks, it is imperative to hold an effective heuristic procedure for large sized problems that predicts greater intricacies with increase in number of tasks that leads to complex determining of optimal solution. SAA thus is bridging the gap prevailed in the U-shaped assembly line for quicker processing of jobs in respective workstations.

3. Problem descriptions

3.1. Problem statement

The proposed work considered here is a U-shaped assembly line (Ozcan, 2010). Products such as cars, trucks and heavy machineries that are heavier in size and shape are manufactured using L, R, and E tasks. The line is arranged as in-line U assembly and the workers/automatic processing heads can be arraigned on either sides of the line. The number of tasks employed in the assembly relies on the product structure and is taken as ‘n’. Each task ‘i’ is constrained with a particular predecessor tasks. In addition to the precedence constraints, some of the tasks were restricted to any one side (Left or Right) of the assembly line in the case of TSALBP and other remaining tasks can be assigned to either side (E) of the line. The time (ti) for processing task ‘i’ is known and deterministic. The main aim of the paper is to develop an algorithm and to find the solution methodology under 2 stages as follows:

- Case (i) Optimal Solution considered as a two-sided line assembly
- Case (ii) Optimal Solution considered as a U line assembly

3.2. Objective criterions

The objectives arrived from the literature review are as follows:

- Minimize the entire cycle time for a given number of mated station.
- Minimize the total number of workstation for a given cycle time (i.e., the number of operators).
- Minimize the number of mated stations for a given cycle time (i.e., line length and number of positions).
- Minimize the number of tasks allocated to each workstation.
- Maximizing the Work Relatedness and Slackness.
- Assigning the tasks from left station to the right station of the position that depends on the start time of the tasks.

4. Proposed methodology

4.1. Simulated annealing algorithm

SAA is a non-traditional optimisation technique that depends on a principle of metallurgy. As it mimics the principle of Annealing or slow cooling it is called Simulated Annealing Algorithm (Van Laarhoven and Aarts, 1987).

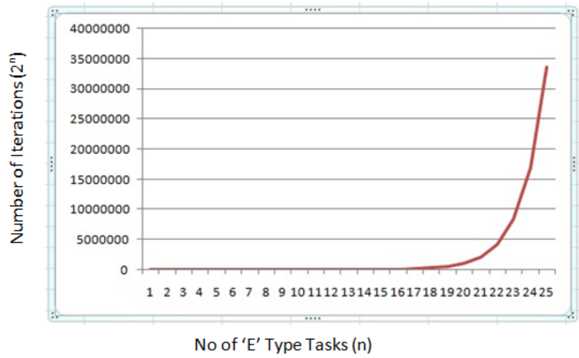

Fig. 3. Number of Iteration of EHA Vs No of 'E' Type Tasks

The acceptance of inferior solution is roughly determined by the Metropolis Criterion (P) as given by Eq. (1).

$$P = e^{(-(Xp-X)/T)} \qquad (1)$$

Where X is the solution at current state, X_p is the perturbated solution of the current system at new state and T is the control parameter (temperature).

4.2. Framework of SAA

The Fig. 4 depicts the framework of the proposed SAA. This section elaborates the details of the various stages of the SAA that is proposed to evolve the pareto front with the objectives of minimum number of workstations and lowest of maximum unbalance time among workstation.

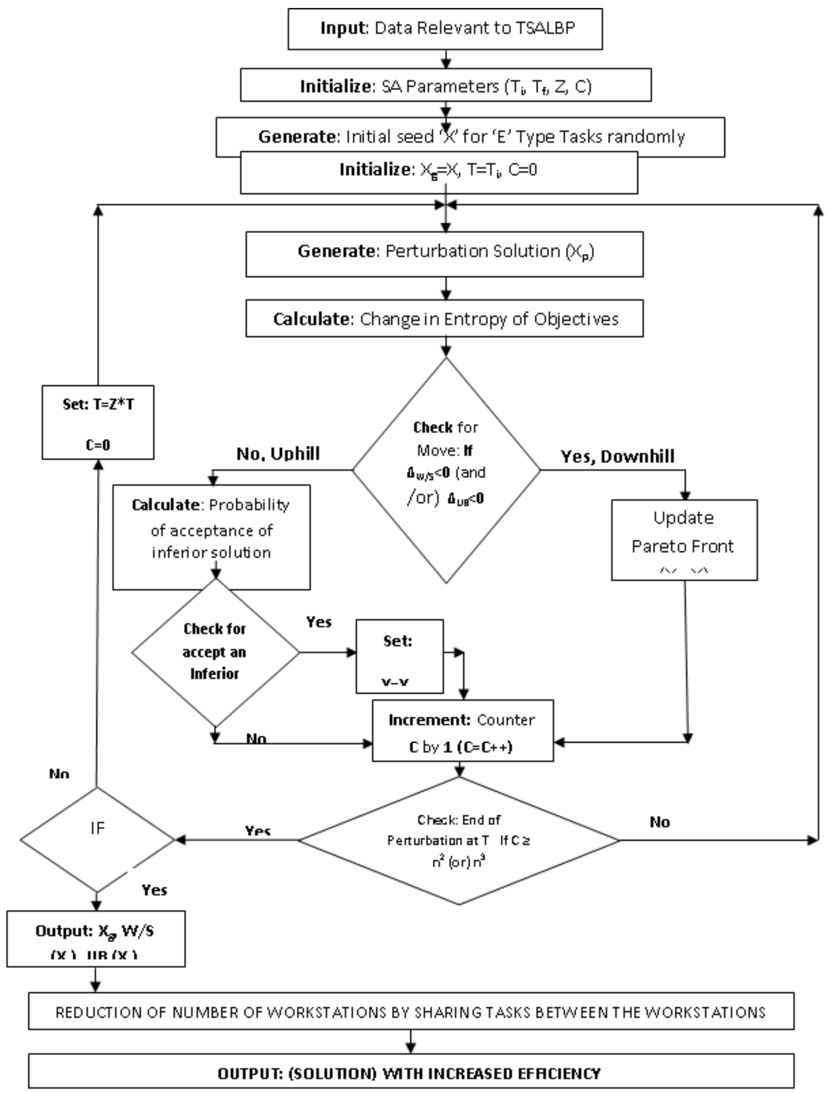

Fig. 4. Framework of the Proposed Simulated Annealing Algorithm for USALBP

4.3. Procedural Steps of the Proposed SAA

Jawahar et al. (2014) had addressed the feasibility of two sided assembly line using SAA and have considered two approaches to solve multi-objective optimization problem: first approach combines all the objectives into a single composite function or moves all but one objective to the constraint set; second approach determines the Pareto optimal solution set and had revealed that Enumerative Heuristic Algorithm (EHA) would be most suitable to handle problems of small and medium size and Simulated Annealing Algorithm (SAA) would be used for large-sized problems. Hence an attempt has been made to utilize SAA to apply in U Shaped assembly work stations with large sized problems with the same steps corroborated by them. The steps are as follows.

Table I Input Data of P19 TALBP

Task 'i'	Processing Time 't_i' (min)	Task Direction 'k_i'	Code of Task Directions	Number of Precedence 'nop_i'	List of Immediate Precedence 'p_i'
1	2.8	L	1	0	-
2	3.1	R	2	0	-
3	2.5	E	3	1	1
4	3.4	R	2	2	1, 2
5	3.2	L	1	1	3
6	2.7	E	3	1	4
7	2.6	L	1	1	4
8	3.3	L	1	1	4
9	5.9	E	3	2	5, 6
10	3.7	R	2	2	6, 7
11	4.1	L	1	2	7, 8
12	2.2	E	3	2	9, 10
13	1.8	R	2	1	10
14	1.2	R	2	2	10, 11
15	2.3	E	3	1	12
16	2.4	L	1	2	12, 13
17	5.8	R	2	1	14
18	3.8	E	3	3	15, 16, 17
19	2.1	L	1	1	18

Table II Possible Assignment of 'E' type task to P19 Problem

Assignment No	'E' Type Tasks					
	3	**6**	**9**	**12**	**15**	**18**
1	L	L	L	L	L	L
2	R	L	L	L	L	L
3	L	R	L	L	L	L
4	R	R	L	L	L	L
5	L	L	R	L	L	L
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
17	**L**	**L**	**L**	**L**	**R**	**L**
.	.	.	.	.	.	.
21	**L**	**L**	**R**	**L**	**R**	**L**
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
64	R	R	R	R	R	R

Step 1: First the input data i.e. the task, direction, processing time and precedence relation were arrived for the given problem.

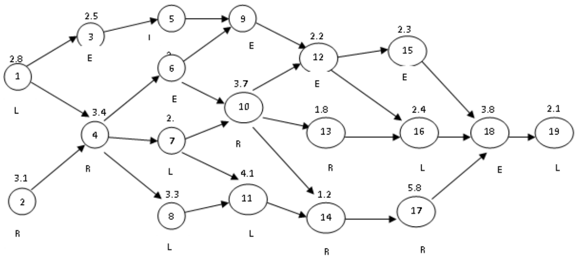

Fig. 5. Precedence Diagram along with Task Time and Operation Directions of P19 – UALBP

Step 2: The directions of the given tasks that were chosen and were limited to either left or right, for which the SAA algorithm was used. Then either type tasks were chosen and arranged in the ascending order with respect to the task number.

Table III Either type tasks arranged based on their task number

Title	**Either type tasks arranged based on their task number**					
Task 'i'	**3**	**6**	**9**	**12**	**15**	**18**
Position 'j'	1	2	3	4	5	6

Step 3: Initially limit the direction of all the **Either** type tasks as **Left** and this combination of the **Either** type tasks constrained to left is called **"X" or initial seed.**

Table IV Initial Seed 'X' of the Illustration Problem P19

Title	**Initial Feasible String for 6 'E' type tasks 'X' which are randomly assigned to Left/out or Right/Inside of U line**					
Task 'i'	**3**	**6**	**9**	**12**	**15**	**18**
Position 'j'	**1**	**2**	3	**4**	5	**6**
k_i (X)	**1**	**1**	1	**1**	1	**1**

Step 4: The parameters of SAA are initialised as follows:

1. Counter (C), C=n^2 (for small size) or n^3 (for large size problem where n is number of tasks in that problem)
2. Initial Temperature T_i =450
3. Final temperature T_f=20
4. Reduction factor Z=0.90

Step 5: Pareto front (X_g), Temperature and counter were initialized as X_g = X (j, i, k_i), T = T_i (450o) and C=0 respectively.

Step 6: Perturbed seed X_p was generated randomly by generating 4 random numbers representing the positions of the initial seed X and then switching the choice of assignment (Left or Right) to opposite.

Table V Perturbed Seed 'X_p'

Task 'i'	3	6	9	12	15	18
Position 'j'	1	2	3	4	5	6
K_i (X)	1	1	1	1	1	1
K_i (Xp)	2	2	1	2	1	2

Step 7: The total number of workstations and related unbalance time (maximum idle time) for the Initial seed (X) and perturbed seed (X_p) combinations were calculated after sharing the tasks among workstations.

Step 8: Then change in entropies in the workstation ΔEW/S and the unbalance time ΔEUB were calculated using the Eq. (2) and (4) respectively. The change in entropies is given below.

$$\Delta EW/S = W/S(Xp) - W/S(X) = 15 – 15 = 0$$

$$\Delta EUB = UB(Xp) – UB(X) = 3.80 - 4.80 = -ve$$

Step 9: if the changes in entropies are negative ($\Delta E_{W/S}$ <0 or ΔE_{UB}<0) then go with downhill move else switch to uphill move.

$$P_a = e^{(- \Delta EW/S /T)} + e^{(- \Delta EUB /T)})/2$$

Step 9.1: Calculate the minimum probability of accepting 'P_a' for the inferior 'X_p' with the formula given in Eq. (4). Generation of random number 'r' (0 to 1)

Step 9.2: If r ≤ P_a, then change X = X_p, otherwise X=X

Step 10: When the downhill move or uphill move is accomplished the counter C (initially initialized as 0) is incremented one by one. Then its value is compared with n^2 (where n is the number of tasks) and if C>= n^2, then T (initial temperature set as 450^{o}C) is compared with the final temperature T_f set as 20^{o}C and if T<=T_f the program ends, else the initial temperature is multiplied with reduction factor z (say T=450*0.90 and the multiplied value reduced to the current temperature T) and then looped to the step 6 where the perturbed seed is once again generated randomly changing the choice of assignment of four positions in the initial seed ‘X’(in this case initial seed may be the one replaced in the downhill move or uphill move). If C<= n^2 then it is looped back to step 6 and the procedure is is to be repeated.

Step 11: Finally, when C>= n^2 && T<=T_f (ie; Temperature (T) reaches final temperature say 200C) the program terminates and the current seed (X) also called as updated pareto front and its assignments becomes the output of SAA.

Xg	= [112121]
W/S [Xg]	= 12
UB [Xg]	= 2.90 second

4.4. Solution methodology (SAA based heuristic approach)

4.4.1. Stage I: optimal solution by SAA for U-line assembly

FOR P19 [CT=6]

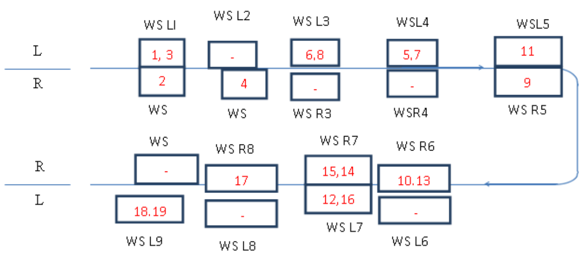

Fig. 6. Optimal Solution by SAA for U-Line Assembly

In this juncture, tasks were arranged into the respective workstations in a U – shaped assembly line based on the precedence relation with the cycle time of 6min. taking all these constraints into consideration tasks are assigned into their respective workstations in a chronological manner. The workstation is common to both inside and outside of the U line with a partition of right and left for a single workstation. The first workstation WS 1 may have partition within it as WS L1 and WS R1.

4.4.2. Stage II: allocation of tasks for the optimal solution obtained by SAA for p19 [CT=6]

The workstation comprising partition within it for example WS L1 and WS R1 are deliberated as two distinct workstations. These two distinct workstations may be assigned into U line assembly based on precedence relation and number of the tasks they hold. Then WS L1 and WS L2 may now be taken as WS1 and WS2 respectively.

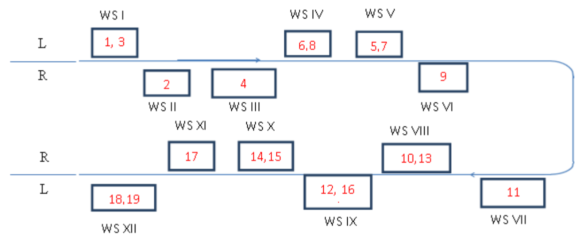

Fig. 7. Allocation of tasks for the optimal solution obtained by SAA for P19

Efficiency=∑t / no. of workstations X cycle time) X 100 =81.80%

Balance delay = 100-Efficiency =18.20 %

Unbalance time = 2.90.

Number of Workstation = 12

4.4.3. STAGE III: Sharing of tasks 15 & 14 by workstations 2 & 3 for the optimal solution obtained by SAA for P19 [CT=6]

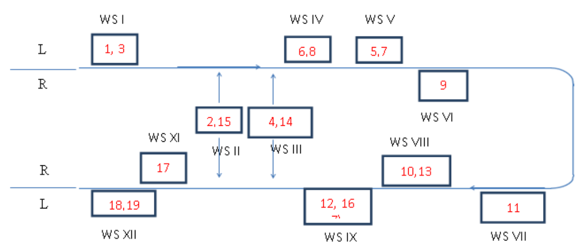

Fig. 8. Sharing of tasks 15 & 14 by workstations 2 & 3 for the Optimal Solution obtained By SAA for P19 [CT=6]

Efficiency=∑t / no. of workstations X cycle time) X 100=89.24%

Balance delay = 100-Efficiency =10.76 %

Unbalance time = 1.90

Number of Workstation = 11.

A sequence of workstation presented on the left and right side of the assembly line. Then divide the total number of workstations by 2 (if even or n+1/2 if odd).

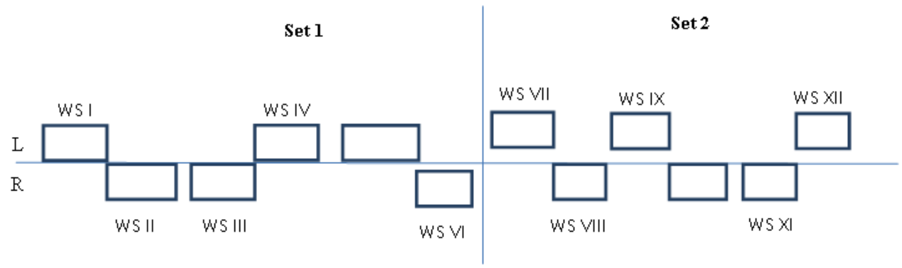

Fig. 9. Dividing the total number of workstations into two sets

Set 1 containing workstation from 1-6 Where 2,3,4 are right and 1,4 ,5 are left	**Set 2** containing workstation from 7-12 Where 8,10,11 are right and 7,9,12 are left

W/S number	**Proximate W/S**
6	X+1 ie.,7
5	X+2,
4	X+3
3	X+4
2	X+5
1	X+6
7	X
8	X-1
9	X-2
10	X-3
11	X-4
12	X-5

Table VII Results of SAA with and without Sharing for UALBP

Problem Identifier	**CT (min)**	**SAA**				**SAA with Sharing**			
		Minimum No. of Workstations	**Minimum Unbalance Time (min)**	**Efficiency %**	**Balance delay %**	**Minimum No. of Workstations**	**Minimum Unbalance Time (min)**	**Efficiency %**	**Balance delay %**
P9 (Kim et al.)	3.01	6	**1.00**	94.13	5.87	6	**0.99**	NC	NC
P19	6	12	2.90	81.80	18.20	**11**	**1.90**	**89.24**	**10.76**
	6.5	12	3.40	75.51	24.49	**11**	**2.40**	**82.37**	**17.63**
P24 (Kim et al.)	15	11	**6.97**	84.84	15.16	11	**3.99**	NC	NC
P47 (Almanza and Ovelle, 2009)	80/60	11	**2567**	87.23	12.77	11	**2452**	NC	NC
	100/60	8	**1181**	95.96	4.04	8	**575**	NC	NC
	120/60	7	**2500**	91.39	8.61	7	**2085**	NC	NC

Case2: To reduce the maximum idle time and to bring equality among the workstations. It is independent of first case. The steps involved are (first you can initialize a counter say p=0 and the following

case 2 can be made to quit when the value of p=n/2 where n is the total number of tasks in given in the problem).

5. Result and discussions

Effective Sharing was observed in case of P19 resulting in reduction of both unbalance time and number of workstation was possible. Furthermore, most processing time of tasks of P19 is one third of cycle time. In order to reduce the number of workstation, tasks allocated in them should be minimal. Thus, in other problems we were able to only reduce the unbalance time but number of workstations remains unchanged.

6. Conclusion

In this paper, balancing problem comprising a shared U-shaped assembly line exclusively designed by a Simulated Annealing Algorithm which may be utilized in medium and large scale problems has been presented. This approach envisages an efficient mechanism that associates a large solution space search for revealing an optimal solution. The existing balancing problem is just a conventional straight assembly line, has limited application to production line where every task are grouped into workstations. The efficiency revealed by SAA for the shared U-shaped assembly line has proved better when compared to existing lines. The Simulated Annealing Algorithm (SAA) heuristic approach was projected to solve the medium and large sized problems by suggesting two objectives concurrently (i) For revealing the optimal number of work stations and (ii) For finding the unbalance time among workstations for a fixed cycle time. The results of the experiments have revealed that the proposed SA-based algorithm outperforms with great effectiveness. This approach would pave a way for future research with various problems involving different workstations.

Reference

1. Cheng G.J. Miltenburg J and Motwani k, IEEE Transactions on Engineering Management, Vol. 47, No. 3, pp. 321-334, 2000.
2. Debora A. Ajenblit, Roger L. Wainwright in Proceedings of IEEE International conference on Evolutionary Computation. Anchorage, AK: pp. 96-101, 1998.
3. Miltenburg G. J.and Wijngaard J. Management Science, Vol. 40, No. 10, pp. 1378-1388, 1994.
4. Groover M.P, Prentice Hall of India, Second Edition, 2002.
5. Miltenburg G. J. International Journal of Flexible Manufacturing Systems, Vol.14(2), pp. 119-151. 2002
6. Miltenburg J. IIE Transactions, Vol. 33(4), pp. 303-321, 2001.
7. Kanagaraj. G and Jawahar. N, International Journals of Procurement Management, Vol.2, pp.244-266, 2009.
8. Kim. J.H Ed., Computer and Operations Research, Vol.36, pp.853-865, 2009.
9. Krikpatrick. S Ed., Science, Vol.220, pp.671-680, 1983.
10. Lee. T.O Ed., Computers and Industrial Engineering, Vol.40, pp.273-292, 2001.
11. Miltenburg, J. International Journal of Production Economics. Vol.70(3), pp.201-214. 2001.
12. Ozcan. U and Toklu. B, Computer and Operations Research, Vol.36, pp.1955-1965, 2009.
13. Ozcan. U, European Journal of Operation Research, Vol.205, pp.81-97, 2010.
14. Purnomo. H.D Ed., Mathematical and Computer Modelling, Vol.57. pp.189-199, 2013.
15. Scholl, A., and Klein, R. International Journal of Production Research. Vol. 37(4), pp.721-736, 1999.
16. Simaria. A.S and Vilarinho. P.M, Computers and Industrial Engineering, Vol.47, pp.391-407, 2004.
17. Simaria. A.S and Vilarinho. P.M, Computers and Industrial Engineering, Vol.56, pp.489-506, 2009.
18. Van Laarhoven, P.J.M. and Aarts, E.H.L, Dordrecht, Holland: D. Reidel Publishing Company, 1987.
19. Yeo Keun Ed., Production Planning and Control: The Management of Operations, Vol.11(1), pp.44-53, 2000.