Research Journal of Applied Sciences, Engineering and Technology 6(15): 2846-2858, 2013
DOI:10.19026/rjaset.37956
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: January 29, 2013 Accepted: May 17, 2013 Published: August 20, 2013

Research Article

Optimization of a Single Model U-SLAB with Stochastic Duration with Integration of Genetic Algorithm and Computer Simulation

Seyed Nima Mirabedini, Hassan Mina, Seyed Hossein Iranmanesh and Babak Saleckpay
Department of Industrial Engineering, University of Tehran, Iran

Abstract: In this study, a simulation optimization method is applied in order to find the optimal design of a U-shape assembly line. The optimality criterion is the minimum number of needed stations. While many previous works use deterministic models to solve this problem, a simulation approach is applied in this study to consider the stochastic nature of the problem. On other hand, when we use a simulation method, a better understanding of system behavior can be obtained through the evolution of system. Another case that is considered in this study is the failures of conveyers which happen in the real world and the fatigue of operators is considered too. The procedure is as follows: first, an initial design of system is obtained by an optimizer (here Genetic Algorithm) with an initial given parameters. Second, the output of the optimizer is used to implement a simulation model in Visual Slam. Third, after running the model in simulator, the desired outputs are evaluated and the necessary changes will be made to optimizer parameters. Fourth, again the optimizer is used to generate new design with new parameters.

Keywords: Assembly line balancing problem, mathematical programming, simulation, single line

INTRODUCTION

Assembly line is a flow of material between a series of work stations. Basically, Assembly Line Balancing problem (ALB) answer to this question that how to assign tasks to stations subject to some constraints, like minimizing total number of stations given cycle time, minimizing cycle time given number of stations and etc. In today competitive environment, it is crucial for manager of big companies to find optimal assignment of tasks to stations, as it can reduces the number of worker we need for stations and consequently, it decreases the cost of production in some extent.

Assembly Line Balancing Problem (ALBP) is categorized into the four following classes by Ghosh and Gagnon (1989):

- Single Model Deterministic (SMD)
- Single Model Stochastic (SMS)
- Multi/Mixed Model Deterministic (MMD)
- Multi/Mixed Model Stochastic (MMS)

After that Backer *et al*. (2006) divided ALBP into these two categories: Simple Assembly Line Balancing Problem (SALBP) and General Assembly Line Balancing Problem (GALBP). Finally we decided to extend the classification done by Scholl and Backer, as presented in Fig. 1.

Considering this classification "Table 1" is devoted to present a literature review for General Assembly Line Balancing Problem (GALBP).

According to definition of ALB problem, different assignment of tasks is possible that met precedence constraint. Finding the best assignment is so time consuming that put this problem in category of NP-Hard problems (Baykasoğlu and Dereli, 2009). As we know, in NP-Hard problems, it is more desirable to have a near optimal solution in a less amount of time rather than optimal solution that is obtained in a long time (maybe days or weeks). Meta-Heuristic algorithms do such thing for us.

Noorul Haq *et al*. (2006) developed a hybrid GA with modified ranked positional method to solve a mixed-model assembly line balancing with *n* model. In fact, to lower time of search, they use ranked positional method to obtain initial solution of GA. RuiJun *et al*. (2007) have improved genetic algorithm considering new data structure to solve ALB problem. Bautista and Pereira (2007) considered time and space constraint, applied Ant Colony Optimization method to find solution of problem. In case of Multi Objective Problems, Rekiek *et al*. (2001) develop kind of GA to solve the proposed problem with different objective functions. Mcmullen and Frazier (1998) have developed a SA to apply on different cases of multi objective ALB. Pastor *et al*. (2002) is another work that considers multiple objectives in multi-model ALB problem. They used Tabu search method to find the solution. Tasan and Tunali (2008) presented a review of different GAs that has been applied to solve different kind of ALBPs.

Corresponding Author: Seyed Nima Mirabedini, Department of Industrial Engineering, University of Tehran, Iran

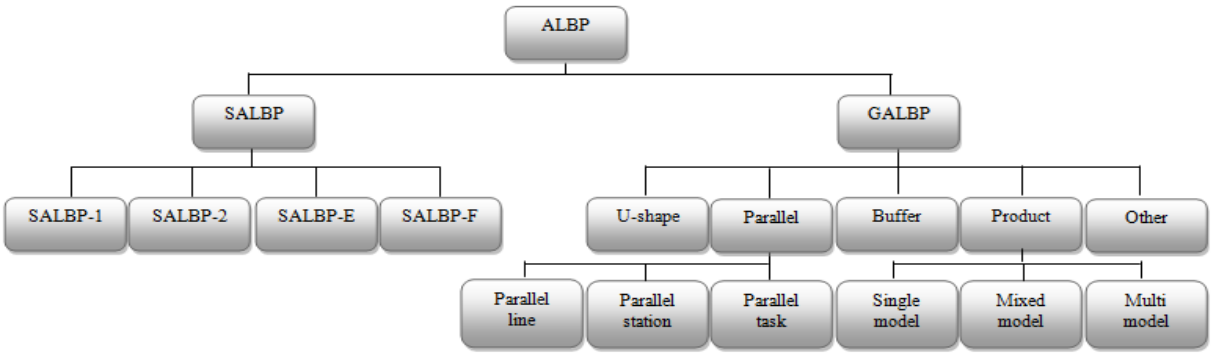


Fig. 1: ALBP classification

Table 1: Literature review for GALBP

GALBP				
	Parallel ALBP			
U-shape	Parallel lines		Parallel station	Parallel tasks
Miltenburg (2000)	Lehman (1969)		Pinto *et al*. (1981)	Arcus (1966)
Chiang and Urban (2002)	Ahmadi *et al*. (1992)		Bard (1989)	Pinto *et al*. (1975)
Guerriero and Miltenburg (2000)	Vilarinho and Simaria (2002)		Bukchin and Rubinovitz (2003)	Dashchenko (2003)
Hwang *et al*. (2008)	Gökçen *et al*. (2006)		Bukchin and Masin (2004)	Dimitriadis (2006)
Kara *et al*. (2009)	Özcan and Toklu (2009)		Chen and Plebani (2008)	Guschinskaya *et al*. (2008)
	Product ALBP			
Buffer ALBP	Single model	Mixed model	Multi model	Other
Buzacott (1968)	Scholl (1999)	Vilarinho and Simaria (2002)	Nearchou (2008)	Amen (2006)
Baker *et al*. (1990)	Gadidov and Wilhelm (2000)	Agpak and Hadi (2005)	Yagmahan (2011)	Andres *et al*. (2008)
Dolgui *et al*. (2002)	Sabuncuoglu *et al*. (2009)	Drexl *et al*. (2006)	Cakir *et al*. (2011)	Boysen *et al*. (2008)
Tempelmeier (2003)	Gökçen *et al*. (2006)	Becker *et al*. (2006)	Yoosefelahi *et al*. (2012)	Corominas *et al*. (2008)
Manitz (2008)	Sabuncuoglu *et al*. (2009)	Boysen *et al*. (2012)	Hamta *et al*. (2012)	Moreira *et al*. (2012)

The main difference of U-Shape with SALB is the direction of task's movement and the way we assign tasks to stations. In SALB, when we want assign a task to a station, only consider the precedence constraint. It means we cannot assign tasks that their predecessors weren't assign to previous stations. But in U-Shape, we consider two conditions, first is the one we have done for SALB and the second is: we check if the successors of a task are assign to one of the previous stations? Either of these conditions is met, we can assign that task. It's the main difference between SALB and U-Shape ALB and this difference does give us chance of having lower number of stations, too (mathematical modeling of U-Shape ALB is covered in next section).

So many researchers devote their time to study different kind of U-Shape ALB problem, as it is so important. In area of exact algorithm (Urban, 1998) presented a new integer programming model for U-Shape lines. Gokcen and Agˇpak (2006) developed a goal programming model for the simple U-Shape line balancing problem that consider simultaneously several conflicting goals. But, mathematical models take so much time to find the optimal solutions and in some situation, they can't (like SALB). That's why researcher began using meta-heuristic algorithm to solve this problem. Kazemi *et al*. (2011) have presented two stage genetic algorithms to solve a mixed model U-Shape ALB problem. The first GA's task is to provide initial population for second GA by being solved for m models. Then they take final chromosomes of first GA to form the chromosome structure of main GA. The objective function of main GA is the cost of stations and duplicated tasks.

Ajeblit and Wainwright (1998) use GA to solve U-Shape assembly line balancing problem. They used six different decoding algorithms that decode each chromosome to assigned tasks and compare their efficiency. Suwannarongsri and Puangdownreong (2008) proposed Tabu search method to solve single model U-Shape ALB problem. Baykasoğlu and Dereli (2009) solve both problem SALB and U-Shape ALB problem using ant colony optimization to make a comparison between them. Baykasoğlu (2006) presented a multiple-rule based multi-objectives Simulated Annealing algorithm to solve simple and U-shape ALB problem.

In the real world problem, it is very often that we don't have exact task duration but their value is a

stochastic variable. Bagher *et al*. (2011) considered a single model U-Shape ALB problem with stochastic task duration. They used a hybrid ICA (Imperialistic Competitive Algorithm) with COMSOAL to solve problem. Baykasoğlu and Özbakır (2007) used a multi-rule coding genetic algorithm to solve a U-Shape ALB problem with stochastic tasks duration. Chiang and Urban (2006) consider stochastic U-Line balancing due to human factors and various disruptions. They presented a hybrid heuristic algorithm to solve this problem. The proposed algorithm consists of two major components, an initial feasible solution module that gives us an initial solution and solution improvement module. Cakir *et al*. (2011) considered a multi-objective optimization of a single-model stochastic assembly line balancing problem with parallel stations. Then, they proposed a new algorithm based on Simulated Annealing (SA) to find Pareto solutions.

Simulation and computer simulation: Simulation refers to broad collection of methods and applications to mimic the behavior of real systems, usually on computer with appropriate software (in some complex systems, software fails to do simulation and we need to code by ourselves). Simulation, like most analysis methods, involves systems and models of them. People often study a system to measure its performance, improve its operation, or design it if it doesn't exist. Before developing a simulation model, primary goal is to focus attention on understanding how their system currently works, which provides great insight into what changes need to be made. In fact, simulation can be extremely general term since the idea applies across fields, industries and applications. These days, simulation is more popular and powerful than ever since computers and software are better than ever.

Computer Simulation refers to methods for studying a wide variety of models of real world systems by numerical evaluation using software design to imitate the system's operation or characteristic, often over time. From practical viewpoint, simulation is the process of designing and creating computerized model of a real or proposed system for the purpose of constructing numerical experiments to give us better understanding of the behavior of that system for a given set of conditions. While simulation may not be the only tool you could use to study the model, it's frequently the method of choice. The reason for this is that the simulation model can be allowed to become quite complex, if needed to present the system faithfully and you can still do simulation analysis. Other method may require stronger simplifying assumptions about the system to enable an analysis, which might bring the validity of the model into question (Kelton *et al*., 2009).

As it was said, computer simulation has a large area of application that makes it really difficult to talk about all of them. In this section, we briefly talk about some studies that apply this approach. Qiu-Gao and Ying-De (2010) in their study, first analyzed the stochastic factors that affect a mixed-model assembly line balancing problem, then they built an Arena model of problem to study these stochastic factors during simulation in a virtual environment. Das *et al*. (2009) developed a computer simulation model to evaluate assembly line balancing problem with variable operation time and bowl phenomenon. Das *et al*. (2010) used computer simulation model to evaluate a bowl versus inverted bowl assembly line arrangement and make some comparison between normal and exponential operation time distribution. Azadeh *et al*. (2008) applied an integrated method to improve and develop railway system. At first, computer simulation is used to model verify and validate the system being studied. The Analytical Hierarchy Process (AHP) methodology determine the weight of any qualitative criteria, finally, the Data Envelopment Analysis (DEA) is used to solve the multi objective model to identify best alternatives and also to identify the mechanism to optimize current system. When we make a simulation model of system, we can't thrust it unless we do some verification and validate our model. It means, we should put our model in different and unanticipated situation then measure the error of our model with comparison to reality and then make suitable corrections. Sargent (2007) has discussed verification and validation of a simulation model in his study by describing four different approaches. More tips and guidelines to have successful simulation modeling could be found in Sadowski (2007) and Sanchez (2007).

Simulation optimization: For a moment pay attention to the parameters that we use in a simulation model as inputs. Our simulation model evaluates these inputs by running in a fixed or unfixed period of times. Then again we repeat these steps by changing the value of inputs to get different outputs and make comparison between them. We call this process, Simulation experiment. Simulation experiment can be defined as a test or series of tests in which meaningful changes are made to the input variables of a simulation model so that we may observe and identify the reasons for changes in the output variable (s). But, what would happen if we have large number of inputs or a complex simulation model? Is it possible to try all changes? In such a situation, we need to estimate best value of inputs among all possible values, somehow (Carson and Maria, 1997). That's the objective of Simulation Optimization, that find the best inputs among all possible inputs in way other than testing all possible inputs. Like simulation applications, simulation optimization has a broad area of application too, as it is in same category of simulation. So we briefly talk about some works that use simulation optimization method.

Yan and Wang (2007) applied this approach for a job shop scheduling problem. Their approach consists of two modules: genetic algorithm based optimizer and discrete event simulation model. Candidate scheduling schemes represented by a serial of scheduling rules are suggested by GA. Simulation models are used to evaluate the performance of candidate scheduling schemes, the results of evaluation are returned to the GA to be utilized in selection of the next generation of candidate scheduling schemes to be evaluated. Mejtsky (2007) use a Meta heuristic algorithm to evaluate simultaneous simulation outputs in simulation optimization process. Keskin *et al*. (2010) built a discrete-event simulation model to evaluate the objective function of the problem (integrated sourcing and inventory decisions) that works in concert with a scatter search-based Meta heuristic optimization approach to search the solution space. Yu *et al*. (2010) in their problem, asset allocation, first construct a simulation model to simulate operations of a property-casualty insurer. Then they develop Multi-Phase Evolution Strategies (MPES) to be used with the simulation model to search for promising asset allocations for the insurer. Azadeh *et al*. (2008) have presented in their study an integrated computer simulation and genetic algorithm for optimizing operator allocation in some workstations to maximize production throughput in a large multi-product assembly shop. The strategy that they have applied is to use computer simulation model as tool for modeling and analyzing the performance of system and then GA is used to maximize the throughput of the system. Azadeh *et al*. (2010) presented an integration of computer simulation and Tabu search and Design of Experiments (DOE) to optimize the performance of production system in a large steel making workshop. Xu and Xiao (2008) integrated fuzzy simulation and genetic algorithm to design a hybrid intelligent algorithm to solve a mixed model assembly line balancing problem with fuzzy operation time. Noushabadi *et al*. (2011) used simulation optimization approach based on genetic algorithm optimizer to find the solution of a stochastic line balancing problem with reworks. Kuo and Yang (2011) employed Particle Swarm Optimization algorithm (PSO) for simulation optimization in order to optimize the managerial parameters in production system.

In this study, the following gaps of previous studies have been discussed and a simulation optimization model is presented to overcome those shortages:

- The conveyer is considered to be unreliable. It would happen to fail and put the whole system in halt.
- The fatigue of operators is either considered. It means that, the more operator work, the more time he needs to perform same job because of being tired.

METHODOLOGY

Mathematical model of SALB and U-shape ALB: In this section, we present mathematical modeling of SALB and U-Shape line balancing problem.
Integer Programming for SALB:

n: *total number of tasks*

t_i: *duration of task i*

IP(i): set of immediat predecessors of task i

C : cycle time

$$x_i = \begin{cases} 1 & \text{if task } i \text{ assign to station } k \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} &\text{Min } z \\ &\text{S.t} \\ &\sum_{k=1}^{m} k \times (x_{ik} - x_{jk}) \le 0 \quad \forall i,j \text{ and } i \in IP(i) \\ &\sum_{i=1}^{n} x_{ik} \times t_i \le c \quad \forall k \\ &z \ge \sum_{k=1}^{n} k \times x_{ik} \quad \forall i \\ &z \ge 0 \text{ and is int eger} \end{aligned} \tag{1}$$

Integer Programming for U-Shape ALB:

n : total number of tasks

ti : duration of task i

IP(i) : set of immediate predecessors of task i

IS(i) : set of immediate successors of task i

C : cycle time

$$x_i = \begin{cases} 1 & \text{if task } i \text{ assign to station } k \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if immediate sucessors of task } i \\ & \text{have been previously assigned} \\ 0 & \text{if immediate predecessors of task } i \\ & \text{have been previously assigned} \end{cases} \tag{2}$$

$$\begin{aligned} &\text{Min } z \\ &\text{S.t} \\ &\sum_{k=1}^{m} k \times (x_{ik} - x_{jk}) \le M y_h \quad \forall i,j \text{ and } i \in IP(i) \\ &\sum_{k=1}^{m} k \times (x_{ik} - x_{jk}) \le M(1-y_h) \quad \forall i,j \text{ and } i \in IS(i) \\ &\sum_{i=1}^{n} x_{ik} \times t_i \le c \quad \forall k \\ &z \ge \sum_{k=1}^{n} k \times x_{ik} \quad \forall i \\ &z \ge 0 \text{ and is int eger} \end{aligned}$$

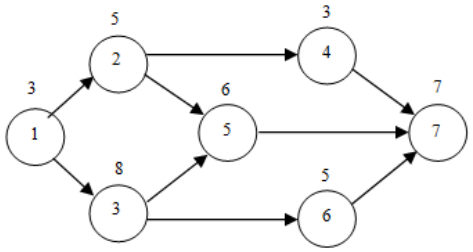


Fig. 2: Precedence diagram of example

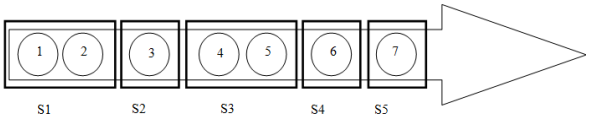


Fig. 3: Simple assembly line of example

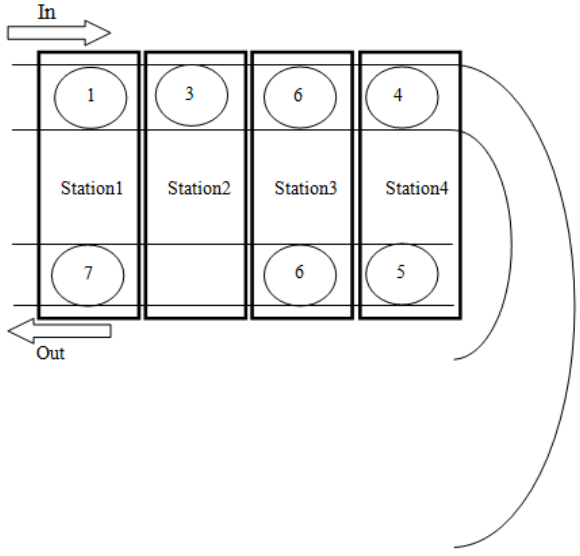


Fig. 4: U-line assembly of example

Now we solve a problem with precedence diagram in Fig. 1 with both SALB and U-Shape to have a comparison. For this problem, suppose that cycle time C = 10 min. the result of two different model is shown in Fig. 2 and 3. As we can see, total number of station in U-Shape is less than SALB. Always it happens. It means that total number of station for U-Shape ALB is at most equal to SALB for same problem.

Problem definition: The problem we consider in this study is a single model U-Shape line balancing consists of 21 tasks. Time between entrances of jobs to first station is an exponential random number with parameter α. Tasks duration is an exponential random number that its factor depends on time. It means operator efficiency comes down after hours and so do mean of random number that we used to simulate tasks duration. So, we consider two parameter, β_1 and β_2. It is probable that conveyer stop moving and if it happen, whole system stop working. Whenever this happen, it takes a uniform amount of time to fix conveyer problem. No rework is considered. There is no incompatibility between tasks, so we can assign tasks to station as long as precedence constraint is met. Cycle time is a predetermined parameter and we want to find assignment of task that maximizes efficiency of stations (or minimize total number of stations), so the number of stations is not known in advance. Another parameter that we want to determine as near to optimal as possible in this simulation optimization procedure is capacity of stations for work in processes.

Figure 4 shows precedence diagram for our problem. A schematic of our system for an arbitrary assignment of task is depicted in Fig. 5.

Method: Figure 6 shows step by step of what we have done in this process. As it was discussed in introduction, we applied simulation optimization strategy for this problem based on Genetic Algorithm as an optimizer. First, after initializing parameters that affect GA (like cycle time and task duration here) we use GA to find optimal or near optimal assignment of tasks. As a matter of fact, we are finding best possible set of inputs for our simulation model. Number of operators for stations is a parameter that we handle as another input for simulation model not GA. As GA is based on random initial population, we can have different solution by multiple runs. So we can have different strategy for simulation (different inputs). For each result of GA, we implement the structure of simulation model in Visual Slam and get multiple runs of a specific structure. Then we analyze the output of this simulation model (objectives of our problem) for important managerial parameters. Then we use this analysis as a feedback for our optimization process (here GA) to test different value of parameters affecting the problem. For example we can decrease the cycle time or increase it and then get new structure of stations and new assignment of tasks.

This is new input for our simulation model and consecutively, we can have new outputs. As it is clear, modules, optimizer and simulator, both have inputs. Outputs of optimizer is an optimal input for our simulation model, among all possible input and output of simulation model affect the input of optimizer. For example, we can test optimal structures of station for different value of cycle time as discussed above

To run simulation model, we assume each station as simple queue system that use resources to do tasks. Pieces come from previous stations and stay in waiting area that is considered for every station. As we have limited space capacity, it is desired for us to minimize

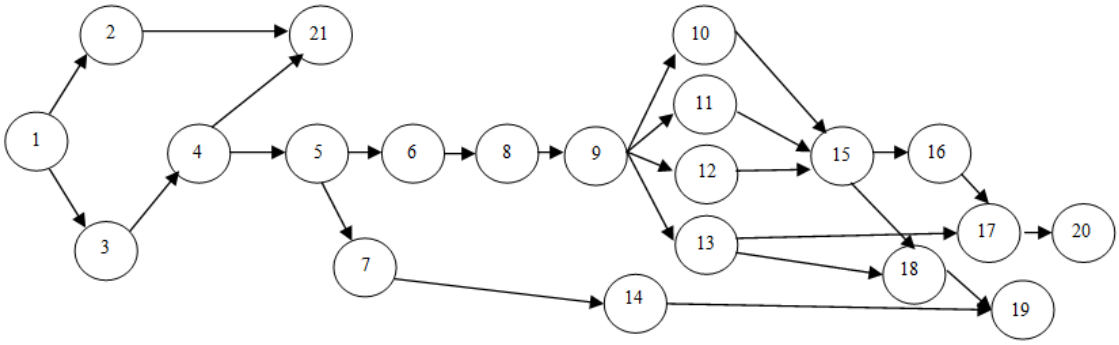


Fig. 5: Precedence diagram of our problem

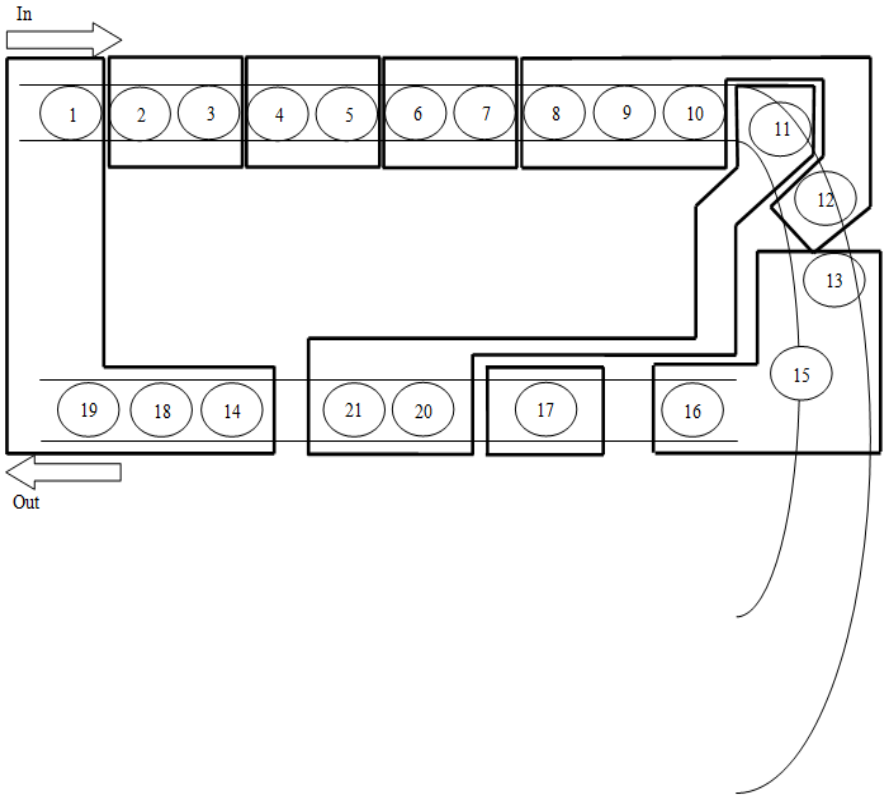


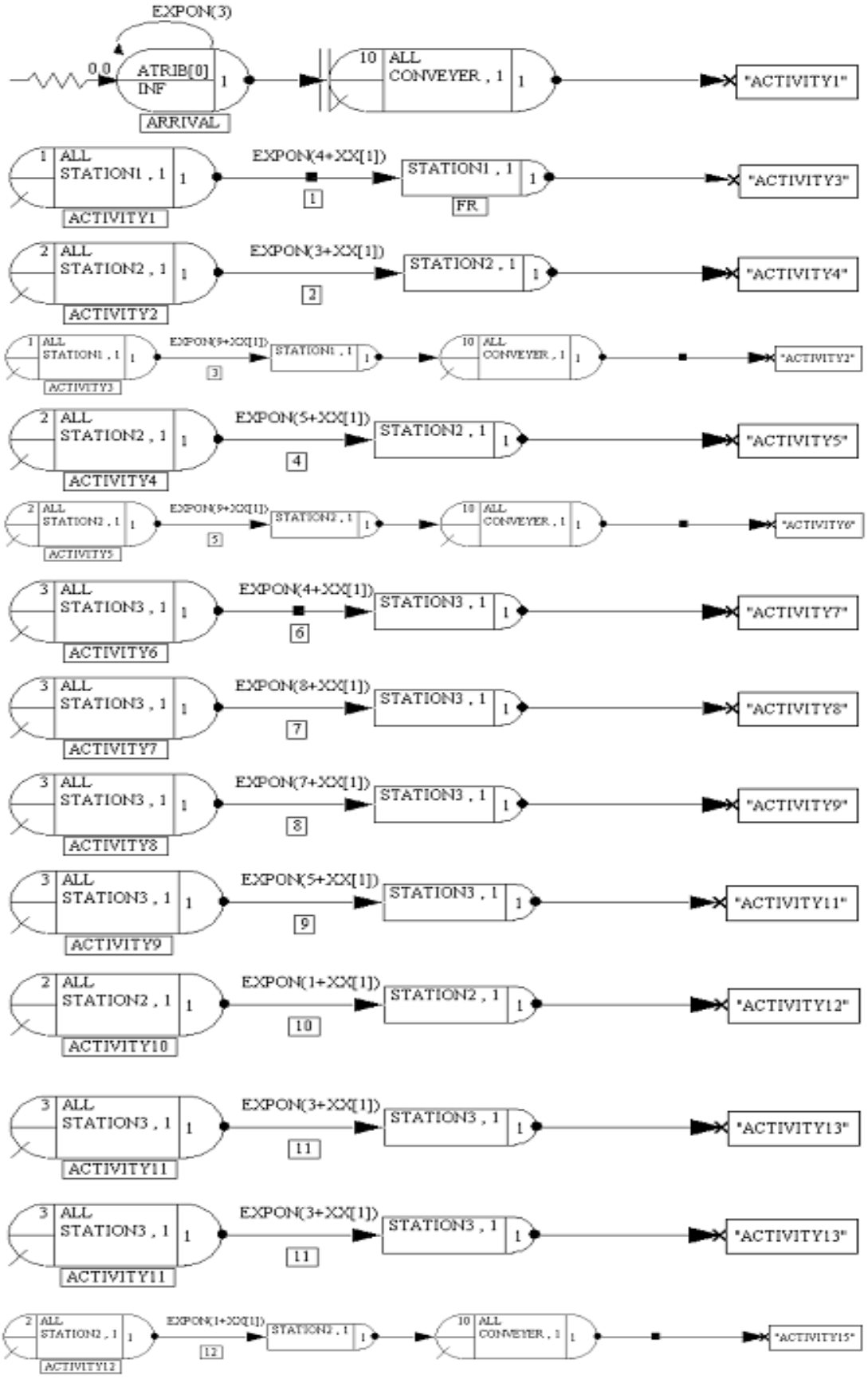
Fig. 6: U-shape arrangement of an arbitrary assignment

maximum available capacity of each station belong to work in process pieces.

To have a reliable simulation model, we divide whole simulation time of every run to two parts, beginning of simulation to time t_0 and rest of simulation. We don't consider the output data for first time interval, as during this period, system is warming up and is in a transient status. As matter of fact, the output of steady state is important for us.

Implementing simulation network in visual slam: As mentioned previously, we used Visual Slam software Package to construct and run our simulation model. Table 2 lists all nodes and resources and gates that we use in constructing model and in Fig. 7, we bring full schematic of our networks with appropriate control statement.

Structure of genetic algorithm: Figure 8 shows steps of Genetic Algorithm that we used to find optimal inputs. First step is initialize parameters that is influenced by feedback from simulation. Then we need to make an initial population of individuals (chromosome). The structure of chromosome in our work is as follow. We consider a random string of numbers from 1 to 21. Each of this numbers is showing


EXPON(3)
0,0
ATRIB[0]
INF
1
ARRIVAL
10 ALL CONVEYER , 1 1
"ACTIVITY1"
1 ALL STATION1 , 1 1
ACTIVITY1
EXPON(4+XX[1])
1
STATION1 , 1 1
FR
"ACTIVITY3"
2 ALL STATION2 , 1 1
ACTIVITY2
EXPON(3+XX[1])
2
STATION2 , 1 1
"ACTIVITY4"
1 ALL STATION1 , 1 1
ACTIVITY3
EXPON(9+XX[1])
3
STATION1 , 1 1
10 ALL CONVEYER , 1 1
"ACTIVITY2"
2 ALL STATION2 , 1 1
ACTIVITY4
EXPON(5+XX[1])
4
STATION2 , 1 1
"ACTIVITY5"
2 ALL STATION2 , 1 1
ACTIVITY5
EXPON(9+XX[1])
5
STATION2 , 1 1
10 ALL CONVEYER , 1 1
"ACTIVITY6"
3 ALL STATION3 , 1 1
ACTIVITY6
EXPON(4+XX[1])
6
STATION3 , 1 1
"ACTIVITY7"
3 ALL STATION3 , 1 1
ACTIVITY7
EXPON(8+XX[1])
7
STATION3 , 1 1
"ACTIVITY8"
3 ALL STATION3 , 1 1
ACTIVITY8
EXPON(7+XX[1])
8
STATION3 , 1 1
"ACTIVITY9"
3 ALL STATION3 , 1 1
ACTIVITY9
EXPON(5+XX[1])
9
STATION3 , 1 1
"ACTIVITY11"
2 ALL STATION2 , 1 1
ACTIVITY10
EXPON(1+XX[1])
10
STATION2 , 1 1
"ACTIVITY12"
3 ALL STATION3 , 1 1
ACTIVITY11
EXPON(3+XX[1])
11
STATION3 , 1 1
"ACTIVITY13"
3 ALL STATION3 , 1 1
ACTIVITY11
EXPON(3+XX[1])
11
STATION3 , 1 1
"ACTIVITY13"
2 ALL STATION2 , 1 1
ACTIVITY12
EXPON(1+XX[1])
12
STATION2 , 1 1
10 ALL CONVEYER , 1 1
"ACTIVITY15"

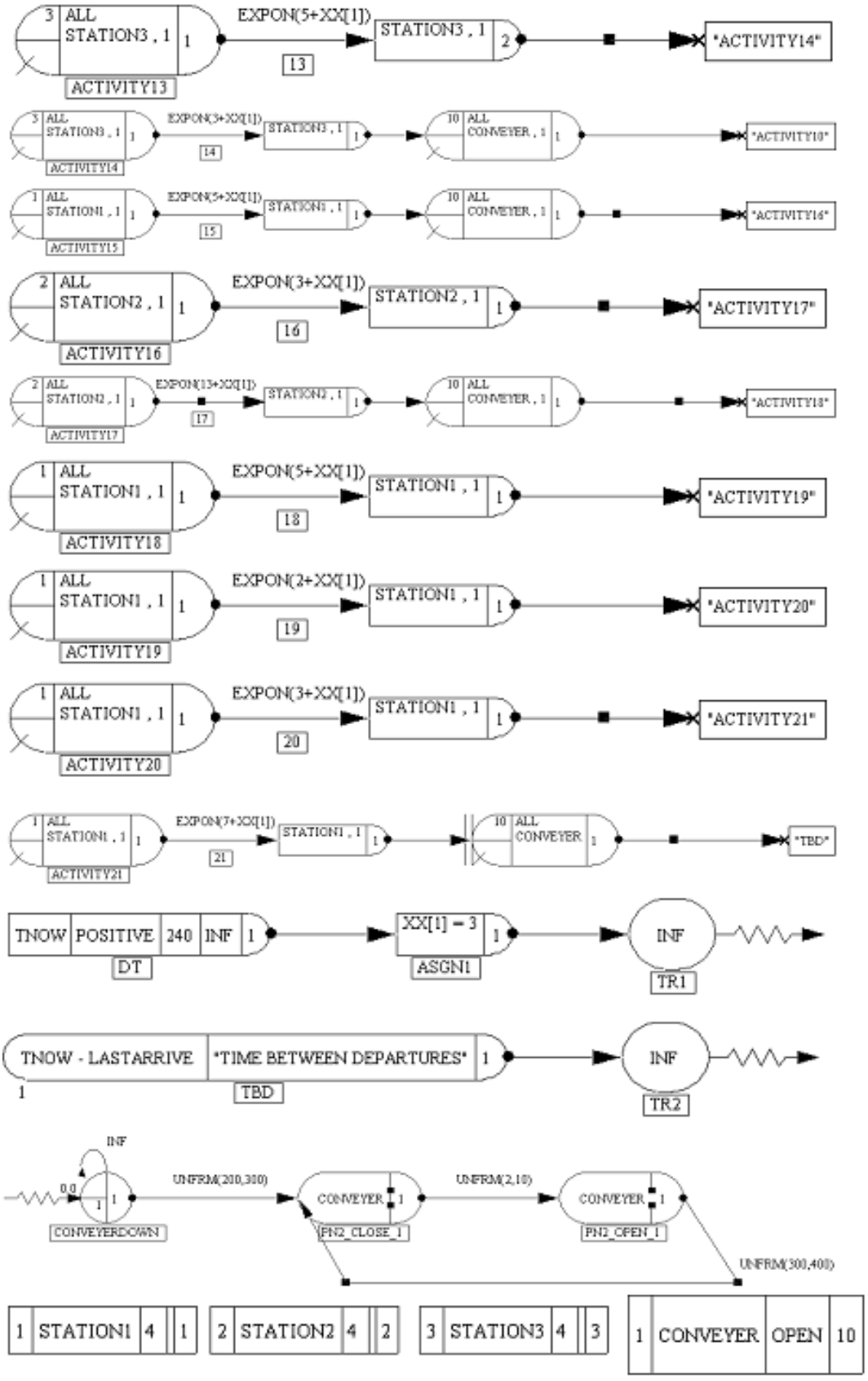


Fig. 7: Visual slam network (for the case of Table 7)

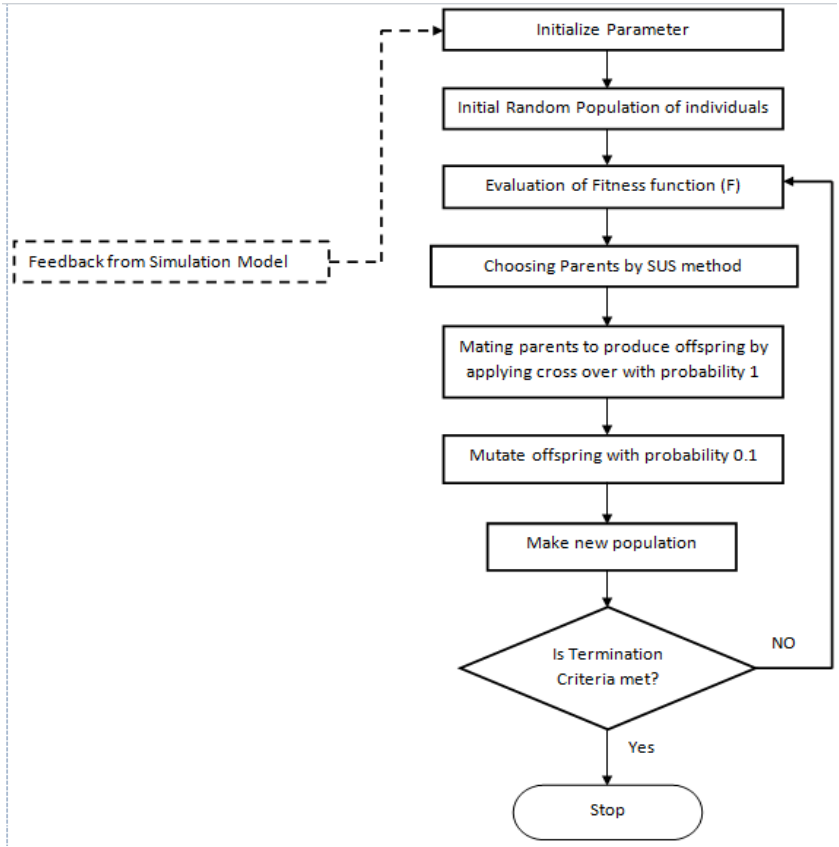


Fig. 8: Structure of genetic algorithm we used for optimization

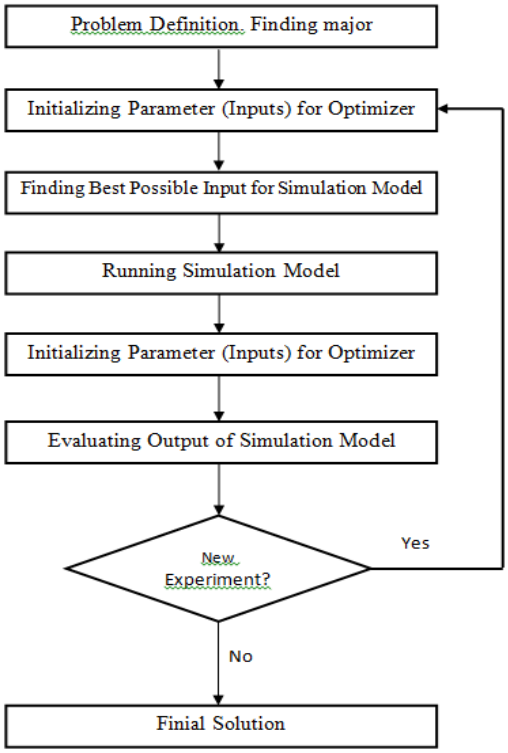


Fig. 9: Step by step of our procedure

Table 2: List of nodes used in visual slam network

Name	Nodes	Description
Activity 1-activity 21	Await	Pieces wait to be processed by operator
Arrival	Creat	Arrival of pieces
FR1-FR21	Free	To free seized resource
Conveyerdown	Creat	To simulation conveyer failures
PN2_close_1	Close	To simulation conveyer failures
PN2_open_1	Open	To simulation conveyer failures
DT	Detect	For duration depend to time
Asgn1	Assign	For duration depend to time
TR1-TR2	Terminate	
TBD	Colct	Time between departures
Station#	Resoure	Operators of stations
Conveyer	Gate	To simulate conveyer

Table 3: Experiment A

Cycle time = 14		Number of stations = 8	
Station	Avg. length	Max. length	Avg. waiting time
1	10.210	47	16.468
2	17.002	29	66.531
3	38.514	66	128.382
4	0.928	5	10.437
5	0.951	7	7.376
6	0.260	2	3.547
7	0.341	3	4.643
8	0.034	1	1.381

Avg.: Average; Max.: Maximum

tasks. To evaluate fitness of these chromosomes we need to decode them to assigned task. The procedure is as follow. We open first station by task 1. Then we

Table 4: Experiment B

Cycle time = 14		Number of stations = 8	
Station	Avg. length	Max. length	Avg. waiting time
1	0.874	7	2.150
2	67.265	129	125.080
3	10.829	17	69.616
4	0.673	4	6.725
5	4.077	12	26.206
6	0.739	5	9.504
7	0.054	2	1.011
8	0.007	1	0.377

Avg.: Average; Max.: Maximum

Table 5: Experiment C

Cycle time = 21		Number of stations = 6	
Station	Avg. length	Max. length	Avg. waiting time
1	48.950	100	51.466
2	0.005	1	0.050
3	0	1	0
4	33.346	55	138.943
5	0.074	1	1.654
6	0.004	1	0.264

Avg.: Average; Max.: Maximum

Table 6: Experiment D

Cycle time = 21		Number of stations = 6	
Station	Avg. length	Max. length	Avg. waiting time
1	19.581	56	21.439
2	0	1	0
3	0.320	6	1.411
4	55.090	94	174.581
5	0.064	1	1.608
6	0.010	1	0.720

Avg.: Average; Max.: Maximum

Table 7: Experiment E

Cycle time = 21		Number of stations = 3	
Station	Avg. length	Max. length	Avg. waiting time
1	12.165	41	15.776
2	52.049	94	75.071
3	0.008	1	0.022

Avg.: Average; Max.: Maximum

Table 8: Experiment F

Cycle time = 21		Number of stations = 3	
Station	Avg. length	Max. length	Avg. waiting time
1	12.165	41	15.776
2	52.049	94	75.071
3	0.008	1	0.022

Avg.: Average; Max.: Maximum

Table 9: Experiment G

Cycle time = 21		Number of stations = 3	
Station	Avg. length	Max. length	Avg. waiting time
1	13.523	45	19.505
2	33.983	64	44.978
3	26.001	40	38.363

Avg.: Average; Max.: Maximum

update a list of tasks can be assigned according to precedence constraint. It means the list contains those tasks with all predecessors or successors assigned previously. Then we determine remained time of current station. Now we update last list to have task with compatible duration for current remained time. We choose first task of last list and assign it to current station and then we repeats these steps for current station until no more tasks is available. It's time to open new station and repeat this for all chromosomes. After this and determining fitness of chromosomes, we need to choose parents. Stochastic Universal Sampling (SUS) is applied here for selection procedure. Then by mating parents, two point cross over is used for producing offspring. Offspring is mutated with probability 0.1. Then we make new generation by combining old generation and new offspring and again using SUS to select among all chromosomes.

EXPERIMENTAL RESULTS

In this section we bring the results of numerous simulation experiments of problem we define in previous sections. We test our model for 3 different cycle time. For each cycle time, we chose two different structures of stations. For all six structures, we construct a network and run the simulation model twenty times. The final results are depicted through Table 2 to 9 (Fig. 9).

CONCLUSION

This study is devoted to analyze a single model U-Shape assembly line balancing problem with a conveyer that maybe failed to work. We integrated computer simulation method with Genetic Algorithm to find best strategy among all possible. GA help us to find inputs of simulation and by running simulation model, we get insights of system that also affect the parameters of our optimizers.

REFERENCES

Agpak, K. and G. Hadi, 2005. Assembly line balancing: Two resource constrained cases. Int. J. Prod. Econ., 96(1): 129-40.

Ahmadi, R.H., S. Dasu and C.S. Tang, 1992. The dynamic line allocation problem. Manage. Sci., 38(9): 1341-1353.

Ajeblit, D.A. and R.L. Wainwright, 1998. Applying genetic algorithms to the U-Shaped assembly line balancing problem. Proceedings of the International Conference on Evolutionary Computation, pp: 96-101.

Amen, M., 2006. Cost oriented assembly line balancing: Model formulations, solution difficulty, upper and lower bounds. Eur. J. Oper. Res., 168(3): 747-770.

Andres, C., C. Miralles and R. Pastor, 2008. Balancing and scheduling task in assembly lines with sequence dependent set up times. Eur. J. Oper. Res., 187(3): 1212-1223.

Arcus, A.L., 1966. COMSOAL: A computer method of sequencing operations for assembly lines. Int. J. Prod. Res., 4(4): 259-277.

Azadeh, A., S.F. Ghaderi and H. Izadbakhsh, 2008. Inegrating of DEA and AHP with computer simulation for railway system improvement and optimization. Appl. Math. Comput., 195: 775-785.

Azadeh, A., M.R. Skandari and B. Maleki-Shoja, 2010. An integrated ant colony optimization approach to compare strategies clearing market in electricity markets: Agent-based simulation. Energ. Policy, 38: 6307-6319.

Bagher, M., M. Zandieh and H. Farsijani, 2011. Balancing of stochastic U-type assembly lines: An imperialistic competitive algorithm. Int. J. Adv. Manuf. Technol., 54(1-4): 271-285.

Baker, K.R., S.G. Powell and D.F. Pyke, 1990. Buffered and unbuffered assembly systems with variable processing times. J. Manufact. Oper. Manage., 3(3): 200-223.

Bard, J.F., 1989. Assembly line balancing with parallel workstations and dead time. Int. J. Prod. Res., 27(6): 1005-1018.

Bautista, J. and J. Pereira, 2007. Ant algorithm for a yime and space constrained assembly line balancing problem. Eur. J. Oper. Res., 177(3): 2016-2032.

Baykasoğlu, A., 2006. Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems. Int. J. Adv. Manuf. Technol., 17(2): 217-232.

Baykasoğlu, A. and L. Özbakır, 2007. Stochastic U-line balancing using genetic algorithm. Int. J. Adv. Manuf. Technol., 32: 139-147.

Baykasoğlu, A. and T. Dereli, 2009. Simple and U-Type assembly line balancing by using ant colony based algorithm. Math. Comput. Appl., 14(1): 1-12.

Becker, C., A. Scholl and S. Armin, 2006. A survey on problems and methods in generalized assembly line balancing. Eur. J. Oper. Res., 168(3): 694-715.

Boysen, N., M. Fliedner and A. Scholl, 2008. Assembly line balancing: Which model to use when? Int. Prod. Eco., 111: 509-528.

Boysen, N., A. Scholl and N. Wopperer, 2012. Resequencing of mixed-model assembly lines: Survey and research agenda. Eur. J. Oper. Res., 216(3): 594-604.

Bukchin, J. and J. Rubinovitz, 2003. A weighted approach for assembly line design with station paralleling and equipment selection. IIE Trans., 35(1): 73-85.

Bukchin, J. and M. Masin, 2004. Multi-objective design of team oriented assembly systems. Eur. J. Oper. Res., 156(2): 326-352.

Buzacott, J.A., 1968. Prediction of the efficiency of production systems without internal storage. Int. J. Prod. Res., 6(3): 173-188.

Cakir, B., F. Altiparmak and B. Dengiz, 2011. Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm. Comput. Ind. Eng., 60(3): 376-384.

Carson, Y. and A. Maria, 1997. Simultion optimization: methods and applications. Proceedings of the 1997 Winter Simulation Conference.

Chen, S. and L. Plebani, 2008. Heuristic for balancing U-shaped assembly lines with parallel stations. J. Oper. Res. Soc. Jpn., 51(1): 1-14.

Chiang, W.C. and T.L. Urban, 2002. A Hybrid Heuristic Forthestochasticu-line Balancing Problem. Working Paper, University of Tulsa, Oklahoma, USA.

Chiang, W.C. and T.L. Urban, 2006. The stochastic U-line balancing problem: A heuristic procedure. Eur. J. Oper. Res., 175(3): 1767-1781.

Corominas, A., R. Pastor and J. Plans, 2008. Balancing assembly line with skilled and un skilled workers. Omega, 36(6): 1126-1132.

Das, B., J.M. Sanchez-Rivas, A. Garcia-Diaz and C.A. MacDonald, 2009. A computer simulation approach to evaluateing assembly line balancing with variable operation times. J. Manufact. Tech. Manage., 21(7): 872-887.

Das, B., A. Garcia-Diaz, C.A. MacDonald and K.K. Ghoshal, 2010. A computer simulation approach to evaluating bowl versus inverted bowl assembly line arrangement with variable operation times. Int. J. Manuf. Technol., 51(1-4): 15-24.

Dashchenko, A.I., 2003. Manufacturing Technologies for Machines of the Future. 21st Century Technologies, Springer, Berlin.

Dimitriadis, S.G., 2006. Assembly line balancing and group working: A heuristic procedure for workers' groups operating on the same product and workstation. Comput. Oper. Res.,33(9):2757-2774.

Dolgui, A., A. Eremeev, A. Kolokolov and V. Sigaev, 2002. A genetic algorithm for allocation of buffer storagecapacities in production line with unreliable machines. J. Math. Model. Algor., 1(2): 89-104.

Drexl, A., A. Kimms and L. Matthießen, 2006. Algorithms for the car sequencing and the level scheduling problem. J. Sched., 9(2): 153-176.

Gadidov, R. and W. Wilhelm, 2000. A cutting plane approach for the single-product assembly system design problem. Int. J. Prod. Res., 38(8): 1731-1754.

Gokcen, H. and K. Agˇpak, 2006. A goal programming approach to simple U-line balancing problem. Eur. J. Oper. Res., 171: 577-585.

Gökçen, H., K. Ağpak and R. Benzer, 2006. Balancing of parallel assembly lines. Int. J. Prod. Econ., 103(2): 600-609.

Guerriero, F. and J. Miltenburg, 2003. The stochastic U-line balancing problem. Naval Res. Logist., 50(1): 31-57.

Guschinskaya, O., A. Dolgui, N. Guschinsky and G. Levin, 2008. A heuristic multi-start decomposition approach for optimal design of serial machining lines. Eur. J. Oper. Res., 189(3): 902-913.

Hamta, N., S.M.T. Fatemi Ghomia, F. Jolaib and M. Akbarpour Shirazia, 2012. A hybrid PSO algorithm for a multi objective assembly line balancing problem with flexible operation times, sequence-dependent set up times and learning effect. Int. J. Econ.,141(1): 99-111.

Hwang, R.K., H. Katayama and M. Gen, 2008. U-shaped assembly line balancing problem with geneticalgorithm. Int. J. Prod. Res., 46(16): 4637-4649.

Kara, Y., T. Paksoy and C.T. Chang, 2009. Binary fuzzy goal programming approach to single model straight and U-shaped assembly line balancing. Eur. J. Oper. Res., 195(2): 335-347.

Kazemi, S.M., R. Ghodsi, M. Rabbani and R. Tavakkoli-Moghaddam, 2011. A novel two-stage genetic algorithm for a mixed-model U-line balancing problem with duplicated tasks. Int. H. Adv. Manuf. Technol., 55(9): 1111-1122.

Kelton, W.D., R.P. Sadowski and N.B. Swets, 2009. Simulation with Arena. McGraw-Hill Higher Education, Boston.

Keskin, B.B., S.H. Melouk and I.L. Meyer, 2010. A simulation-optimization approach for integrated sourcing and inventory decisions. Comput. Oper. Res., 37(9): 1648-1661.

Kuo, R.J. and C.Y. Yang, 2011. Simulation optimization using particle swarm optimization algorithm with application to assembly line design. Appl. Soft Comput., 11(1): 605-613.

Lehman, M., 1969. On criteria for assigning models to assembly lines. Int. J. Prod. Res., 7(4): 269-285.

Manitz, M., 2008. Queueing-model based analysis of assembly lines with finite buffers and general service times. Comput. Oper. Res., 35(8): 2520-2536.

McMullen, P.R. and G.V. Frazier, 1998. Using simulated annealing to solve a multi-objectives assembly line balancing problem with parallel workstations. Int. J. Prod. Res., 36(10): 2717-2741.

Mejtsky, G.J., 2007. A meta-heuristic algorithm for simultaneous simulation optimization and applications to traveling salesman and job shop scheduling with due dates. Proceedings of the 2007 Winter Simulation Conference. Washington, DC, pp: 1835-1843.

Miltenburg, J., 2000. The effect of breakdowns on U-shaped production lines. Int. J. Prod. Res., 38(2): 353-364.

Moreira, M.C.O., M. Ritt, A.M. Costa and A.A. Chaves, 2012. Simple heuristic for the assembly line worker assignment and balancing problem. J. Heurist., 18(3): 505-524.

Nearchou, A.C., 2008. Multi-objective balancing of assembly lines by population heuristics. Int. J. Prod. Res., 46(8): 2275-2297.

Noorul Haq, A., K. Rengarajan and J. Jayaprakash, 2006. A hybrid genetic algorithm approach to mixed-model assembly line balancing. Int. J. Adv. Manuf. Technol., 28: 337-341.

Noushabadi, M.E., U. Bahalke, K. Dolatkhahi and A.M. Yolmeh, 2011. A simulation optimization approach to Un-paced assembly line balancing problem-II with additional reworking issue. Proceeding of the 4th International Conference on Modeling, Simulation and Optimization. Kuala Lumpur, pp: 1-6.

Özcan, U. and B. Toklu, 2009. Balancing of mixed-model two-sided assembly lines. Comput. Ind. Eng., 57(1): 217-227.

Pastor, R., C. Andrés, A. Duran and M. Pérez, 2002. Tabu search algorithms for an industrial multi-product and multi-objective assembly line balancing problem, with reduction of the task dispersion. J. Oper. Res. Soc., 53(10): 1101-1108.

Pinto, P., D.G. Dannenbring and B.M. Khumawala, 1975. A branch and bound algorithm for assembly line balancing with paralleling. Int. J. Prod. Res., 13(2): 183-196.

Pinto, P.A., D.G. Dannenbring and B.M. Khumawala, 1981. Branch and bound and heuristic procedures for assembly line balancing with paralleling of stations. Int. J. Prod. Res., 19(5): 565-576.

Qiu-Gao, S. and L. Ying-De, 2010. Study on stochastic mixed-model assembly line balancing based on arena. Proceeding of the International Conference on Mechanic Automation and Control Engineering (MACE). Wuhan, pp: 3446-3449.

Rekiek, B., P. De Lit, F. Pellichero, T. L'Eglise, P. Fouda, E. Falkenauer and A. Delchambre, 2001. A multipleobjective grouping genetic algorithm for assembly line design. J. Intell. Manufact.,12: 467-485.

RuiJun, Z., C. Dingfang, W. Yong, Y. Zhonghua and W. Xinxin, 2007. Study on line balancing problem based on improved genetic algorithms. Proceeding of the International Conference on Wireless Communications, Networking and Mobile Computing. Shanghi, China, pp: 2033-2036.

Sabuncuoglu, I., E. Erel and A. Alp, 2009. Ant colony optimization for the single model U-type assembly line balancing problem. Int. J. Prod. Econ., 120(2): 287-300.

Sadowski, D.T., 2007. Tips for successful practice of simulation. Proceedings of the 2007 Winter Simulation Conference. Baltimore, MD, pp: 87-94.

Sanchez, P.J., 2007. Fundamental of simulation modeling. Proceedings of the 2007 Winter Simulation Conference. Washington, DC, pp: 54-62.

Sargent, R.G., 2007. Verification and validation of simulations models. Proceedings of the 2007 Winter Simulation. Baltimore, MD, pp: 166-183.

Scholl, A., 1999. Balancing and Sequencing of Assembly Lines. 2nd Edn., Physica-Verlag, Heidelberg.

Suwannarongsri, S. and D. Puangdownreong, 2008. Balancing of U-shape assembly lines using tabu search method. Proceedings of International Conference on Electrical Engineering/Electronics. Krabi, pp: 609-612.

Tasan, S.O. and S. Tunali, 2008. A review of the current applications of genetic algorithms in assembly line balancing. J. Intell. Manufact., 19(1): 49-69.

Tempelmeier, H., 2003. Simultaneous buffer and workload optimization for asynchronous flow production systems. Proceedings of the 4th Aegean International Conference on Analysis of Manufacturing Systems. Samos.

Urban, T.L., 1998. Optimal balancing of U-shaped assembly lines. Manage. Sci., 44(5): 738-741.

Vilarinho, P.M. and A.S. Simaria, 2002. A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations. Int. J. Prod. Res., 40(6): 1405-1420.

Xu, W. and T. Xiao, 2008. Mixed model assembly line balancing problem with fuzzy operation times and drifting operations. Proceedings of the 2008 Winter Simulation Conference. Austin, TX, pp: 1752-1760.

Yagmahan, B., 2011. Mixed-model assembly line balancing using a multi-objective ant colony optimization approach. Expert Syst. Appl., 38(10): 12453-12461.

Yan, Y. and G. Wang, 2007. A job shop scheduling approach based on simulation optimization. Proceeding of the International Conference on Industrial Engineering and Engineering Management. Singapor, pp: 1816-1822.

Yoosefelahi, A., M. Aminnayeri, H. Mosadegh and H. Davari Ardakani, 2012. Type II robotic assembly line balancing problem: An evolution strategies algorithm for a multi-objective model. J. Manuf. Syst., 31(2): 139-151.

Yu, T.Y., C. Tsai and H.T. Huang, 2010. Applying simulation optimization to the asset allocation of a property-casualty insurer. Eur. J. Oper. Res., 207(1): 499-507.