Australian Journal of Basic and Applied Sciences, 5(7): 1070-1076, 2011
ISSN 1991-8178

The Production Line Task Assignment by Considering Workload Equilibrium

[1]Vahid Yazdanparast, [2]Hamid Hajihosseini and [3]Unes Bahalke

[1]Islamic Azad University, Zahedan Branch, Department of Engineering, Zahedan, Iran.
[2]Islamic Azad University,Zahedan Branch, Department of Industrial Engineering, Zahedan, Iran
[3]Islamic Azad University, Zahedan Branch, Zahedan, Iran

Abstract: This paper tackles with a production line which includes the series of work centers dispersed in a one way straight line. The new approach of grouped task assignment restriction is considered and a simulation integrated optimization approach is also developed to cover the problem in a general status of production line. The effects of this restriction are inspected on the workload balance in a simple single product production line.

Key words: Production Line; Task Assignment restriction; Simulation Optimization; Simulated Annealing.

INTRODUCTION

Each production line consists a series of work centers names and stations and a set of buffers which are a free spaces between every two workstations. According to the commodity that the line produces the line separates to the fixed number of stations and a fixed number of tasks which should be assigned in to these stations. Allocation of these tasks is dependent to several factors, which almost of the most theoretical studies do not considers these limitations in simple assembly or production lines. One of these restrictions is the regards to the nature of the tasks and the machines that should be do them. This limitation causes to assign the specific tasks or at least not assign the grouped specific tasks to one particular workstation. This issue is exists in many production lines and almost there is not any production or assembly line that has not this limitation. Totally in order to perform an assigned task, the goal workstation must be equipped by operators and machines which have the abilities and technical capabilities required. Particularly in the case of complex products it is usually not possible to have all stations equipped equally resulting in work center regarded assignment restrictions. Besides, the assignment of tasks may be limited by task related constraints such as incompatibilities between tasks, minimum or maximum distances in terms of time or space) between stations performing a pair (or subset) of tasks. Furthermore, position regarded limitations are relevant for commodities which are heavy, large or rigid at the conveyor belt such that they cannot be changed in any status which is required for executing a task in a certain station. Another type of assignment restrictions is worker related, because operators have different levels of ability such that only particular task integrations are possible when a worker is allocated to a certain station. So according to this issue, a model should be presented to this kind of problems to consider and cover more conveniently of the production problems while the managers face to them. The tasks assignment restrictions are considered in other kind few studies and at the top of them are (Kilbridge, 1961; Ignall, 1965; Deckro, 1989; Agnetis, 1995; Buxey, 1976; Wang, 1986).

The major notification in simulation optimization methodologies are the implemented optimization techniques and presenting an optimization algorithm to simulation problems can be helpful due to its capability on comprehend domination of problems. The meta-heuristic algorithms have been used many in the simulation integrated optimization and also manufacturing related crisis's domains. In these subjects the information related the execution of simulated annealing algorithms in scheduling problems can be find in (Gen, 2009), or the Artificial neural networks (ANNs) as valid alternative to the classic job-shop simulation fields (Fonseca, 2002; Fonseca, 2003). Wang (2005) combined a genetic algorithm and a neural network strategy (GANN) for such kind of optimization problems. Lutz *et al.* (1998) considered the buffer location and storage size in a assembly line by tabu search integrated with simulation. Also the application of genetic algorithm in manufacturing problems can be found in (Paul, 1998; Pasandideh, 2006). At the end the simulated annealing algorithm which is one of the most used simple, practical and with high speed of convergence meta-heuristic that is implemented in simulation optimization approaches. In this subject Ahmed and Alkhamis (Ahmed, 2002) developed an algorithm that mixed simulated annealing and ranking and selection to solve a discrete stochastic

Corresponding Author: Vahid Yazdanparast, Islamic Azad University, Zahedan Branch, Department of Engineering, Zahedan, Iran.
Tel:+98-9371627883; E-mail:yazdanparastvahid@yahoo.com.

optimization problem of a manufacturing unit.

In this paper the main simulated annealing meta-heuristic is developed to optimize the simulation model of a production line dealing with buffers, machine setups, product failure and a new approach of grouped assignment restricted tasks by considering the workload equilibrium as the goal of optimization model.

Problem Definition:

As it defined the considered production line structured by a set of workstations allocated in line format series which furthered by buffers between every two stations. The line produces a single product which means that in every cycle time just only one complete product exits the line, moreover the work pieces may undergo to the failure options that occurs after exiting from the previous station through checking process. Via the checking process the products or work pieces will fail by a probabilistic percent. Also the machines all need to consume time to prepare their selves to perform tasks known as setup times. For above described problem a simulation model is designed which is combined with an optimization tool known as popular Simulated annealing algorithm that is practical and well-liked for its speed and convergence. So for above described line the work load equilibrium is optimized which the goal investigates the minimizing the working time differences of the stations, in other words it seeks a balanced work load in all the work centers. The other objective is finding an optimum time of production cycle which has the maximum number of production at a defined working time.

Simulation and Optimization Model:

***3.1. Simulation Model*:**

Due to the characteristics of the considered problem the Discrete Event Simulation (DES) methodology is employed. In *discrete-event simulation*, the process of a system is represented as a sequential series of events. Each event happens at a moment in time and marks a change of condition in the arrangement. While classifying the system solver techniques regarding to its requirement, the DES is categorized as Fig. 1.
The discrete event simulation includes three following specifications (Banks, 1984).

- Stochastic: some state variables are random.
- Dynamic: time evolution is important.
- Discrete event: significant changes occur at discrete time instances.

The simulation model clarified and the model specifications are also as follows:

System status: A binary variable of $LE_{[t]}^{[i]}$, where $\begin{cases} 1 & \text{if workstation i is busy} \\ 0 & \text{otherwise} \end{cases}$, and positive integer variable of $LQ_{[t]}^{[i]} \geq 0$ which shows the queue length in Buffer *i*.

***Entities*:**

The tasks and work pieces are entities.

***Events*:**

Every occasion that changes the system status is known as event in which the entrance to workstations and leavings of them and also the simulation completion are known as events in the current problem.

***Activity*:**

However the state variable is defined for workstation status but its process does not equivalent to activity, because the activity is a process that its finish time is known by its starting, but while a work piece enters to a workstation its processes finish time is unknown due to existence of failure issue. Thus the workstation process is not known as activity and it is reports as a *delay* in simulation model, on the other hand the tasks that are assigned inside of the sub model of each station are account as activities.

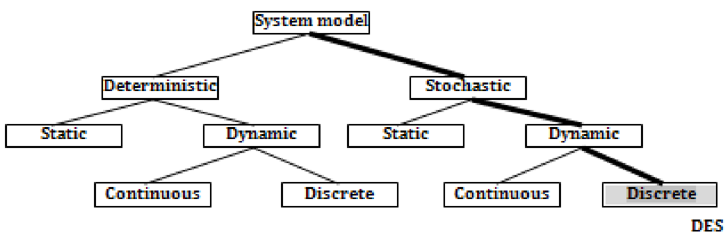


Fig. 1: System Taxonomy.

3.2. Optimization Combined with Simulation Model:

Simulated annealing (SA) is a generic probabilistic meta-heuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global optimum of a given function in a large search space. It is often used when the search space is discrete. The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one. SA was introduced by Kirkpatrick *et al*. (1983) as a technique to solve combinatorial optimization problems. SA starts with an initial solution (X), an initial temperature (T), and an iteration number (L). Temperature (T) manages the possibility of the acceptance of a worsening solution, and an iteration number (L) decides the number of repetitions until a solution finds a steady position under the temperature. The T may have the subsequent hidden meaning of a flexibility manifestation (Baker., 1995). At high temperature (early in the search), there is some flexibility to shift to a worse solution; but at lower temperature (later in the search) less of this flexibility exists. A new neighborhood solution (Y) is produced based on these T and L via a heuristic perturbation on the existing solutions. The neighborhood solution (Y) becomes a new solution if the change of a fitness function $(\Delta = (C(Y)) - C(X))$ is got better (i.e., $\Delta < 0$ for the considered minimization *problem).* Even though it is not improved, the neighborhood solution becomes a new solution with an appropriate probability based on $e^{-\Delta/T}$. This leaves the possibility of finding a global optimal solution out of a local optimum. The proposed algorithm stops if there is no improvement after L repetitions. Otherwise, the iteration goes on with a new temperature *(T).*

3.2.1. Initial Solution and Neighborhood Generation:

Since the all considered algorithm's solution representations are the same, neighborhood creation in SA is also similar to local search in TS, i.e., The positions of two tasks are replaced or the station positions are shifted a genome to forward or backward. The proposed algorithm's pseudo-code is presented as following (Fig.2).

```
Begin;
INITIALIZE(X;T;L);
Repeat
      For i = 1 to L do
      Y=PERTURB(X); {generate new neighborhood solution}
      Δ =C(Y)−C(X);
     If ((C(Y) ≤C(X)) or (exp (− Δ/T )> RANDOM (0, 1))
       Then X=Y; {accept the movement}
     End (if);
     End (for);
UPDATE (T; L);
Until (Stop-Criterion)
End;
```

Fig. 2: Pseudo-code of proposed SA.

3.2.3. Temperature:

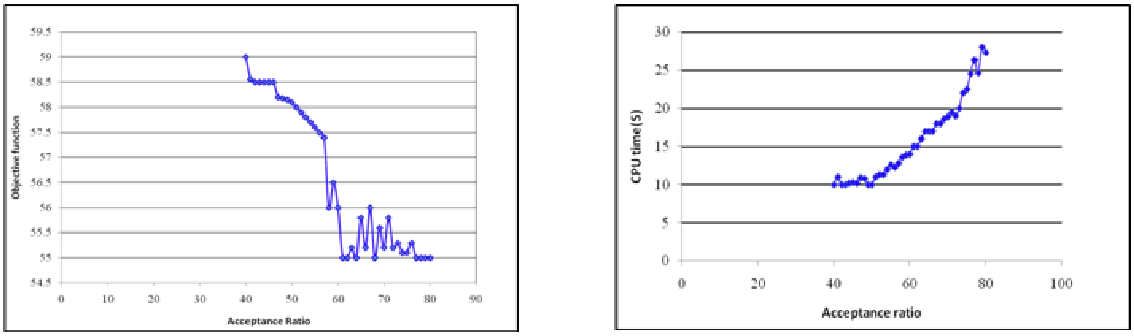
As told the initial temperature directly affects the accepted solution percentage. Selecting of high initial temperature will lead to consume more computational time and on the other side the low initial temperature will limits the search space.

Thus this temperature has important role in the algorithm. Therefore, this temperature is mainly firmed by the minimum temperature that is larger than or equal to the acceptance ratio, the number of accepted neighbors divided by the number of total neighbors, tested by preliminary trials before actual SA (Johnson, 1989). So determining the acceptance ratio consequently confirms the initial temperature. After several testing of variant number of acceptance ratios starting from 40% to 80% the ratio of 60% resulted in better outputs in comparison with other ratios. This testing is in both aspects of CPU running time and solution qualities shows in Fig. 3 (A) & (B).

Fig. 3 shows that the solution quality has not significant difference by passing through the 60% to 80% and is in lowest (best) values and chart in Fig. 3 (B) satisfies us to select the 60% ratio of acceptance, since the computational time of this ratio (13 seconds) is about less than the half of ratio of 80% (28 seconds).

3.2.3. Cooling Ratio:

The determined initial solution should be decreased along the algorithm's proceed to comprise the convergence status of the algorithm.



(A) Solution Quality (B) Cpu Time

Fig. 3: Acceptance ratio determination.

A cooling ratio which is based on a geometric formula is used in this paper (Kim, 2002):

$T_k = \alpha T_{k-1}$, $k=1, 2, \ldots$. $0<\alpha<1$

The ratio of α is selected randomly from distribute interval of [0.55, 0.9]. Also the complete strategy of the proposed Simulation Optimization technique is portrayed in Fig. 4.

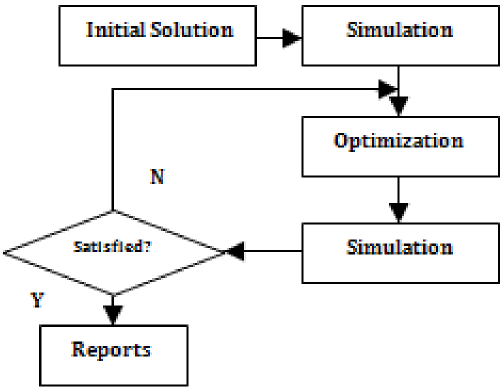


Fig. 4: The strategy of Optimization simulation and popular Simulation optimization.

RESULTS AND DISCUSSIONS

In this section the combination of simulation and optimization methodology is presented and its results and outcomes are discussed. The all experiments are done on the problem of Mendes *et al.* (precedence diagram).

This precedence relation is presented in Fig. 5. As accordance to the descriptions, the introduced procedure starts from an initial solution and proceeds its solution by the simulation, the simulation iterates until a determined termination criterion met, next the optimization algorithm get into work and the average value of fitness function or the objective function is directly transferred to the optimization algorithm, after that the SA proceeds as what trend is described and SA results in new solution and the cycle goes around and continue until the SA algorithm reaches a satisfying results.

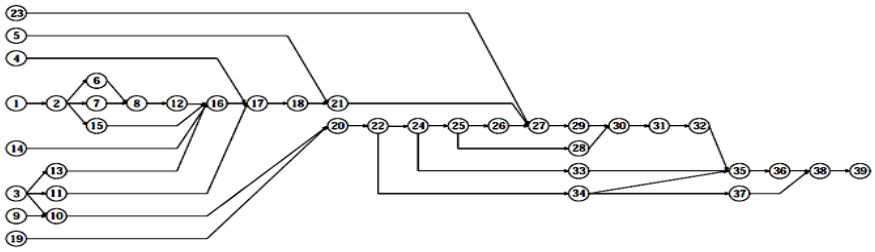


Fig. 5: The precedence diagram

As told this paper considers work load equilibrium as objective function. For the considered problem also some additional assignment restrictions are considered. Since in assembly lines always there are some taboos in assigning sub permutations of some grouped tasks, these limitations emerges due to some technical restrictions, however the precedence relations allows the assignment. These grouped tasks for the considered problem are depicted in Table. 1. for example the group 1 (3, 11) means that any permutation of these two tasks ((3→11 or 11→3)) are taboo in the same workstation and this issue is dependent from the precedence relations.

Table. 1: Assignment restricted grouped tasks.

Group 1	Group 2	Group 3	Group 4	Group 5
(3,11)	(20,22)	(20,33,29)	(15,16)	(26,31,37)

The rest of the input data of the simulation model are considered from the uniform distributions, which are presented in Table. 2. The all task times (P_i), Setup times ($S_{i,j}$), product failure probability ($P_{F(i)}$) are considered from the uniform distributions as what exhibited in Table.2. This Table explains that the processing time of each task is bounded from below by discrete uniform interval of (Agnetis, 1995; Fonseca, 2003) and from upper is bounded by (Kim, 2002).

Table 2: Input parameters.

P_i	~[5,10] up to[20,25]
$S_{i,j}$	~[1,3] up to [4,6]
$P_{F(I)}$	~[0,0.1]

The values of $S_{i,j}$ are exactly explains in the same order and failure probabilities are determined from the uniform distribution of [0.1, 0.3]. The simulation running time is considered 10000 time unit and the problem is solved considering 15 workstations.

The results of experiments are exhibited in Table. 3. As it described the workload is the time of the station, and the goal of the balance is to minimize the following formula:

$$\frac{\sum_{i=1}^{M}\left(\left|T_{\max}-T_i\right|\right)}{M} \tag{1}$$

Where M is the total number of workstations, T_i is the total working time of the station i and T_{max} is the cycle time. Fig. 5 shows the assignment and schedule of the tasks in workstations for the presented problem with 39 tasks and as portrayed in Table 3, this Table includes all the information of the simulation optimization results.

Table 3:

Stations	Assigned Tasks	Average waiting time	Average queue length
1	9,19	Infinite	Infinite
2	3,37,23	56.00	0.73
3	5,36,11	10.59	0.021
4	13,38	0	0
5	14,1,4	0	0
6	2,15	0	0
7	39,7,10	2.05	0.002

Table 3: Continue.

8	20,6	0	0
9	8,22	0	0
10	34,12,16	0.99	0
11	24,25,17	24.73	0.02
12	28,33,26	5.24	0.01
13	18,21,27	10.036	0.01
14	29,30	0	0
15	31,32,35	0	0
Work load equilibrium	31.31	Cycle time	64.73

In above table these two Average queue length and average waiting time factors are planned based on the following equations (2) and (3), respectively which are calculated separately for every buffer.

$$L = \frac{\sum_{i=0}^{\infty} i.T_i}{T} = \sum_{i=0}^{\infty} i\left[\frac{T_i}{T}\right] \tag{2}$$

$$W = \frac{1}{N}\sum_{i=1}^{N} W_i \tag{3}$$

Where, L is the average queue length, T_i is the sum of times during the total simulation time (T) where exactly i products exist in queue. And where W is the average waiting time of all products in each buffer and N is the number of all WIPs (products) that entered to line. W_i is the waiting time of product i.

The final results show that based on the assigned limitations and taboos for the grouped assignment restrictions, there is not any taboo groups assigned together in any workstations. And the achieved workload equilibrium is 31.31 which is calculated based on the introduced formula in equation (1) and the cycle time is achieved 64.73, which is the maximum task time that a specific workstation belongs to. At the end it can be extracted that the designed SA algorithm is proper for the introduced production problem.

Conclusions:

This paper considers the problem of workload equilibrium in a production line while the tasks trapped in a grouped assignment restriction. This limitation occurs due to the technical limitations such as the operator ability, while an operator has not the ability of working with a specific group of tasks, or the lack of equipment in a workstation may lead to this limitation, or the characteristics of tasks, lack of station space and etc, are the main causes of this problem. Due to the complexity of this kind of production lines the mathematical models cannot convey the problem solutions and their computational solution times extremely high, so presenting a mathematical model will not be efficient, since its lack in giving the solutions in an appropriate time. For this reason this paper seeks an optimization procedure integrated with a simulation model. The simulation optimization methodology has shown its compatibility and effectiveness in recent years, due to its exponential speed of practical exploitations. The optimization tool proceeds along the simulation and the popular very adapted meta-heuristic algorithm named as simulated Annealing is designed to the problem. The results showed its effectiveness and compatibility.

REFERENCES

Agnetis, A., A. Ciancimino, M. Lucertini, M. Pizzichella, 1995. Balancing flexible lines for car components assembly. International Journal of Production Research, 33: 333-350.

Ahmed, M.A. and T.M. Alkhamis, 2002. Simulation-based optimization using simulated annealing with ranking and selection, Computers & Operations Research., 29(4): 387-402.

Bukchin, J., M. Tzur, 2000. Design of flexible assembly line to minimize equipment cost. IIE Transactions 32: 585-598.

Banks, J. and Carson, J.S. 1984. Discrete-Event Simulation. Prentice-Hall INC., pp: 4-12.

Buxey, G.M., D. Sadjadi, 1976. Simulation studies of conveyor- paced assembly lines with buffer capacity. International Journal of Production Research, 14: 607-624.

Baker, K.R., 1995. Elements of sequencing and scheduling. Amos Tuck School of Business Administration, Dartmouth College, N.H. Hanover. 19.

Deckro, R.F, 1989. Balancing cycle time and workstations. IIE Transactions 21: 106-111.

Fonseca, D.J. and D. Navaresse, 2002. Artificial neural networks for job shop simulation. Advanced Engineering Informatics., 16(4): 241-246.

Fonseca, D.J., D.O. Navaresse and G.P. Moynihan, 2003. Simulation metamodeling through artificial neural networks, Engineering Applications of Artificial Intelligence., 16(3): 177-183.

Gen, M., D. Green and O. Katai, 2009. Intelligent and Evolutionary Systems, Springer., Publishing Company, 2009.

Ignall, E.J., 1965. A review of assembly line balancing. Journal of Industrial Engineering, 16: 244-254.

Johnson, D.S., C.R. Aragon., L.A. Mageoch and C. Schevon, 1989. Optimization by simulated annealing: an experimental evaluation; part 1, graph partitioning. Oper Res, 37: 865-892.

Kilbridge, M.D., L. Wester, 1961. The balance delay problem.Management Science, 8: 69-84.

Kirkpatric, S. J.C.D. Gelatt and M.P. Vecci, 1983. Optimization by simulated annealing. Science, 220: 671-80.

Kim, D.W., K.H. Kim, W. Jang and F.F. Chen, 2002. Unrelated parallel machine scheduling with setup times using simulated annealing. Robotics and Computer Integrated Manufacturing., 18: 223-231.

Lutz, C.M.K., Davis, and R. Sun, 1998. Determining buffer location and size in production lines using Tabu search. European Journal of Operational Research., 106(2–3): 301-316.

Mendes, A.R., A.L. Ramos, A.S. Simaria and P.M. Vilarinho, Combining heuristic procedures and simulation models for

Paul, R.J., and T.S. Chanev, 1998. Simulation optimization using a genetic algorithm. Simulation Practice and Theory., 6(6): 601–611.

Pasandideh, S.H.R. and S.T.A. Niaki, 2006. Multi-response simulation optimization using genetic algorithm within desirability function framework. Applied Mathematics and Computation., 175(1): 366-382.

Wang, F., R.C. Wilson, 1986. Comparative analyses of fixed and removable item mixed model assembly lines. IIE. Transactions, 18: 313-317.

Wang, L., 2005. A hybrid genetic algorithm—neural network strategy for simulation optimization, Applied Mathematics and Computation., 170(2): 1329-1343.