Mansoura Engineering Journal

Volume 22 | Issue 2 Article 3

6-1-2021

An Approach to Assembling on Fixed Position Layout.

Ahmed Abd El Hamed El-Shafie
Associate Professor., Department of Industrial Production Engineering., Faculty of Engineering., El-Mansoura University., Mansoura., Egypt.

Hassan Soltan
Assistant Professor., Industrial Production Engineering Department., Faculty of Engineering., El-Mansoura University., Mansoura., Egypt.

Recommended Citation
El-Shafie, Ahmed Abd El Hamed and Soltan, Hassan (2021) "An Approach to Assembling on Fixed Position Layout.," *Mansoura Engineering Journal*: Vol. 22 : Iss. 2 , Article 3.
Available at: https://doi.org/10.21608/bfemu.2021.150651

Mansoura Engineering Journal , (MEJ) , Vol. 22 , No. 2 , June 1997.

An Approach to Assembling on Fixed Position Layout

أسلوب للتجميع بإستخدام التخطيطى الثابت

A. A. A. Abdel-Shafi and H. A. Soltan
Department of Industrial Prod. Engng.,
Faculty of Engineering, Mansoura University, Egypt

تم تطوير أسلوب جديد لنظام التجميع وذلك كبديل للأساليب التقليدية لخطوط التجميع والأسلوب الجديد يقوم بتوزيع وتخصيص الأعمال على خلية تجميع تتكون من عدد الماكينات بإستخدام صيغة جديدة تقوم بتقليل التكلفة الكلية للمنتج المصنع بطريقة الحد الأدنى الخاص بالتجميع الأوتوماتيكى ، حيث تتبع طريقتان للتعامل مع الأعمال غير المكتملة والتى من الممكن أن تحدث فجأة فى العمليات التجميعية ، يعتمد النظام المقترح هنا أساسا على الموقع التخطيطى الثابت ، حيث يثبت الجزء الرئيسى للمنتج فى محطة متعددة الماكينات تقوم بإستكمال جميع الأعمال اللازمة .

عملية تخصيص الشغل هنا ليست موجهة كلية إلى الاتزان كما هو الحال بالنسبة للأساليب التقليدية للخطوط حيث يتم تعيين الأعمال لسلسلة من المحطات ، ولكن الأسلوب الجديد يستفيد من حالة وجود ماكينتين أو أكثر فى الخلية - تؤدى كل منها العمل مستقلة وآنيا - ويقوم بمساواة الوقت الضائع لكل الماكينات قدر الإمكان .

التكلفة الكلية للوحدة المجمعة يتم تقديرها بإستخدام طريقتين الأولى تجريبية والثانية حسابية . وقد أثبتت التطبيقات المبرمجة للأسلوب الجديد مقدرته العالية وأنه متحدى فعال للأساليب التقليدية .

As an alternative to the classical assembly lines techniques, an approach is developed here considering another assembly system. It allocates the tasks on an assembly cell consists of a number of machines. The control is conducted using a new formulation which tends to minimize the total cost of the manufactured product. The *lower-bound* approach of automatic assembly is considered, where it flags two ways in dealing with breakdowns which may occur suddenly in processes. The proposed assembly system is mainly based on the fixed position layout, where the major component is stationary at a multiple-machine station which completes whole job. The assignment procedure is not completely for balancing as occurs in known line procedures which assign the tasks into a series of work stations. It tries to equalize the idle time on all machines, and exploits the situation that two or more machines process many more independent tasks simultaneously. The total cost of unit assembly is computed through two functions, the first is experimental and the second is analytical. The computer application of the approach confirms that the examined system is an efficient challenger and exhibits the capability of the approach itself.

1. INTRODUCTION

Automatic assembly is a term that refers to use of mechanized and automated devices to replace manual assembly operations. Thus, an automated assembly machine consists typically of the following elements:1. Transfer system for transporting a partially completed assembly from work station to the next. 2. Automatic work stations to perform the various assembly steps. Included at a typical work station is a part storage facility for holding the components to be added at that station, and a parts orientation and a mechanism for feeding to present the components to the workhead in a correct position for assembly. 3. Manual assembly station where the human operators carry out the steps that are not easily mechanized.

One of the problems encountered in the design of an automated assembly machines is orienting and feeding of components at the various work stations because of different component geometries.So another problem, manufacturing variations, sometimes, may lead to a certain fraction rate of defective components which may cause a jam or add a defective component; so, this may cause a stoppage at a subsequent station or machine. Generally,in the current work, an automation is proposed, at specific positions, in order to enhance the productivity of labour, and the policy will not to replace the overall human design see Kamali et al. [9].

A transfer line comprises of a number of automatic stages are arranged in series, and often with planned buffer. These whilst in action are liable to fail and are repairable . Automatic transfer lines is similar to that which has been analyzed in the literature by Buxy et al. [1], Buzacott [2], Groover [8], Gershwin and Schick [6], Law [12], and others. Also, automated assembly has been dealt with by Smith and Daskalaki [14], Kuula et al.[11], Feldmann and Roth [3], Shin and Zheng [13], and Gaimon [4,5].

A different approach, will be described, to allocate all job tasks to two or more automatic machines fixed to one station . It is mainly based on the fixed position layout in which the product unit is a fixed item. Also this is similar to modular assembly by manning the station by a group of workers. Each of machines must be assigned a group of tasks. The repair of breakdowns is based on the lower-bound approach which is implemented for transfer lines. The procedure is not a balancing procedure, in that sense, but it tries to equalize the idle time between the machines and minimize the total operating cost including the incompletion component. An incompletion due to a machine breakdown will be dealt with in two different ways according to the product type and material cost. If the product is small and destructive failure occurs, scrap action may be taken otherwise it must be reworked. Actually, this type of facilities arrangement is often associated with job shops. Thus to explain the assignment procedure, some definitions must be made.

2. THE APPROACH

In addition to precedence matrix, two matrices were designed and called "relation matrices", which can be developed and defined as follows:

1. *Task-Machine matrix*-Is a numerical representation to the relation between the layout machines and the job tasks, it is used to indicate that which task can be carried out by which machine. A matrix entries include zero's and one's; entry "1" means that the machine can do the task, and entry "0" otherwise, as shown example in Fig. (2-1) (a).

2. *Concurrent matrix*–Also it is a numerical representation to the relation between the tasks each others, it is used to indicate if there are two or more tasks can be carried out in the same time by using, also, zero's and one's entries. Entry "1" between tasks means that they can be carried out concurrently, and entry "0" is found otherwise, as shown example in Fig. (2–1) (b).

M/C Task	M1	M2	M3	M4	M5	M6
1	0	1	1	0	1	1
2	1	1	1	1	1	1
3	1	1	0	1	1	0
4	1	0	1	1	0	1
5	1	0	0	1	0	0
6	0	0	1	0	1	1
7	1	1	0	1	0	1
8	1	1	0	1	1	0
9	1	0	1	1	0	1
10	1	1	1	1	0	0

(a) Task-Machine matrix.

Task	1	2	3	4	5	6	7	8	9	10
1		0	1	1	0	1	0	1	1	0
2			0	0	1	1	1	0	0	1
3				1	1	0	1	0	1	1
4					1	1	0	1	0	1
5						1	0	1	1	0
6							0	1	1	0
7								1	0	1
8									1	1
9										1
10										

(b) Concurrent matrix.

Fig. (2-1) Relation marices.

2.1 Formulation of the Problem

2.1.1 Assumptions and Restrictions

The assumptions imposed here are different from those made in the case of balancing with a series of work stations because the restriction of cycle time will be in another sense. However, those can be summarized as follows:

1. The central objective is to minimize the total variable operating cost including such component, the incompletion cost.
2. The assembly layout is by fixed position in which all the machines are arranged independently.
3. Each task must be defined to one machine or more.
4. Each machine can carry out a group of tasks with available facilities.
5. The units come to the system in a uniform distribution.
6. Units produced must be identical for the same setting up and the same cycle and can be changed by changing the setting (multi model assembly.)
7. Task times are independent random variables with estimable means and variances.
8. Task time distributions are unknown and may be different.
9. The precedence restrictions are specified and satisfied.
10. There are two actions, based on lower-bound approach, can be taken when a machine breaks down and fails to complete a given task, the system will be stopped and,
    a. If the product is small and its material is not expensive, it will be scrapped (incompletions of scrap type.)
    b. If the product is large and its material can not be scrapped, it will be reworked out the station

(incompletions of rework type.)

Each of the types of incompletions incurred has a function used to evaluate the total unit operating cost.

11. No buffer is allowed.
12. The task failure events are equally likely.
13. Different products can be manufactured with low rates.

2.1.2 Mathematical Model

The assumptions cited before are used to help in developing a mathematical stochastic form, according to the same set theory, it must satisfy the following:

1. Min. TCF= MCF + ICF .... objective function

Subjected to

2. $\cup_{f=1}^{F} S_f = E$ .... equivalence of system and product
3. $\forall E_i \in E \ \exists \ m/c \in S_f \leftrightarrow \forall \ m/c \in S_f \ \exists \ E_i \in E$ ...... matching
4. $S_i \cap S_j = \Phi, \ i \neq j$ .... mutually exclusive machines
5. If E_n pr E_m and $E_n \in S_i$, $E_m \in S_j$, then $i \leq j$ ..... precedence
6. $\{\Gamma(S_f) = \sum_{i \in S_f} \mu_i\} \leq T_f$ ..... machine time to control output

where F : is the total number of machines in the system.
E : is the set of all tasks of the job.
TCF: is the expected total operating cost of one unit.
MCF: is the normal operating cost, on system, of one unit.
ICF: is the expected incompletion cost of one unit.
μ_i : is the mean time of task i.
S_f : is the set of tasks assigned to machine f.
T_f : is the maximum time allowed to machine f.

In this formulation, there are elements similar to elements in the formulation made before in the balancing approach, but the linking between the six elements gives a different problem.

2.2 Input Data and Computational Parameters

2.2.1 Data Required

The necessary data to complete the assignment process include the task time observations, the precedence matrix, on system rate, off system rate, the number of machines in the system, the maximum time can be assigned to each machine, the task-machine matrix, and the concurrent matrix.

2.2.2 Assignment Parameters

The mean time is the principal parameter computed and used to assign the job tasks to the system machines.

2.3 Assignment and Evaluation Procedure

The procedure, whose flow chart is shown in Fig. (2-2), which is proposed for the purpose of assigning the tasks to the machine system and evaluating total operating cost, builds up the process

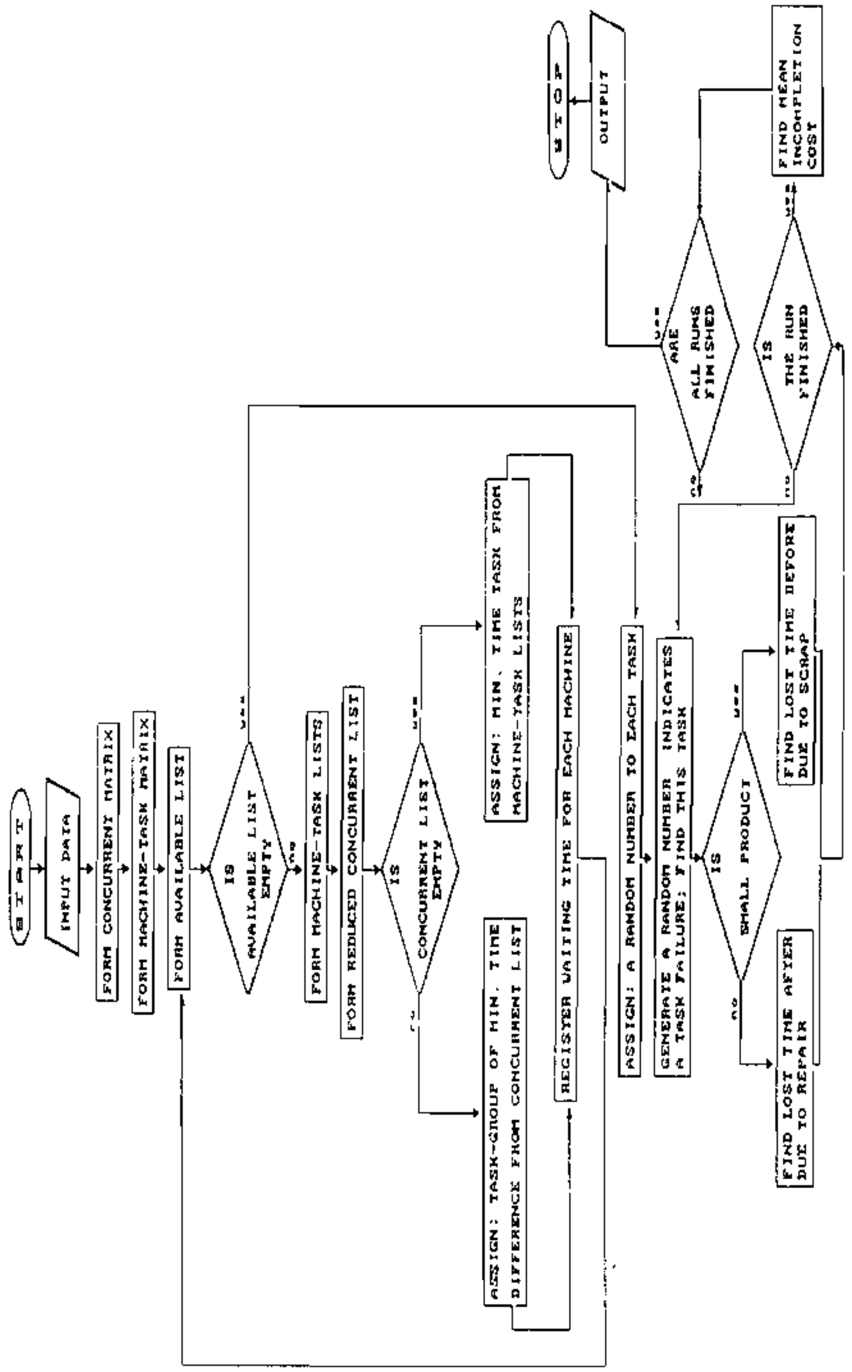

Fig. (2-2) Flow chart for the job shop heuristic.

until allocation is finished. The evaluation segment is based on Monte-Carlo method. The steps can be clearly explained as follows:

(1) Input data:

All the data mentioned in section (2.2.1) must be prepared to make a complete conception about the problem. Go to step (2).

(2) Compute assignment parameters:

The assignment parameters, here, will be the task mean times and the task time variances. Those parameters are constant and can not be changed during assignment. Go to step (3).

(3) Form relation matrices:

If they are not included in data, they can be formed by using discrete information about them. Go to step (4).

(4) Form available list:

Available list, as defined before, is the list of tasks which currently satisfy the precedence restrictions. Go to step(5).

(5) Check if available list is empty:

If it is empty, the assignment is completed, go to step (10); otherwise, go to step (6).

(6) Form machine-task lists:

This list comprises available list tasks which are currently available for each machine tasks. Go to step (7).

(7) Form reduced concurrent list:

This list comprises all possible combinations of tasks which can be carried out in the same time. This can be done by reviewing and updating the concurrent matrix with the precedence matrix each time at such decision point. Go to step (8).

(8) Check if concurrent list is empty:

If it is empty, assign the task of minimum time from all the machine-task lists to its machine; select the machine which has minimum accumulated time, thus, minimizing the system and machines idle time. If it is not empty, assign the task combination; each task is assigned to the corresponding machine with minimum accumulated time. The minimum accumulated time is calculated as

$$[\Sigma_i \Sigma_j (\mu_i - \mu_j)^2]^{\frac{1}{2}} = \text{Min.},\ i \neq j; (i,j) \in \text{Com}_b \ \ldots\ldots \quad (2.3\text{-}1)$$

where $\text{Com}_b = (X_1,\ X_2, \ldots, X_F),\ b = 1, 2, \ldots, B$ .... (2.3-2)

where Com_b : is the combination set b of tasks X_f.

B : is the maximum number of current combinations.

F : is the total number of machines in the system.

And then, go to step (9).

(9) Register waiting time for each machine:

Each time a task assigned one of two conditions may be found. First, all machines assigned tasks with different times. Second,at least one machine is not assigned a task. In both cases, a machine waiting time exists and it must be registered. Return to step (4).

(10) Assign a random number to each task:

Thus, an uniformly distributed random number generator can be used. Starting from that point, each task in the job is identified by its assigned random number. Go to step (11).

(11) Generate a random number:

After completing the assignment, a random number indicating task failure is generated and the corresponding task is drawn. Go to step (12).

(12) Check product size:

If the rework is destructive, an incompleted unit is scrapped and the time lost, TLB_j, before a task j in sequence which causes the failure, can be approximated as

$$TLB_j = \sum_{i\in E} \mu_i - PW_{j\in S_f} \quad \ldots\ldots\ldots\ldots \quad (2.3\text{-}3)$$

$$PW_j = \sum_{i\geq j} \mu_i \quad \ldots\ldots\ldots\ldots \quad (2.3\text{-}4)$$

where E : is the set of all tasks.

μ_i : is the mean time of task i in the set of all tasks.

PW_j : is the ranked positional weight of incompleted task j.

S_f : is the set of tasks assigned to machine f.

Quantity TLB is accumulated each time a random incompleted task is encountered in a each run.

But if the rework is not destructive, an incompleted unit is taken out the the system and repaired manually and the time lost after task j, TLA_j, can be estimated as

$$TLA_j = PW_j \quad \ldots\ldots\ldots\ldots \quad (2.3\text{-}5)$$

Quantity TLA is accumulated each time a random incompleted task is encountered in each run. Go to step (13).

(13) Check if the current run is finished:

If it is not finished, return to step (11). Otherwise, it can be possible to find the expected unit incompletion cost, ICF, as

$$ICF = \begin{cases} \ddot{a} \sum_i (TLB_j)_i \,/\, I & \text{if scrapped} \\ \quad \text{or} & ,i=1,2,\ldots,I \\ \ddot{A} \sum_i (TLA_j)_i \,/\, I & \text{if reworked} \end{cases} \quad \ldots\ldots\ldots \quad (2.3\text{-}6)$$

Also the normal operating cost, MCF, can be evaluated as

$$MCF = \ddot{a} \sum \mu_{i\in E} + \bar{\ddot{a}} \sum W_f \quad \ldots\ldots\ldots\ldots \quad (2.3\text{-}7)$$

where $\ddot{A}$: is the off-system labour rate.

$\ddot{a}$: is the normal rate (on-system) of operating the system.

$\bar{\ddot{a}}$: is the penalized rate due to waiting.

I : is the run length.

W_f: is the waiting time incurred of machine f.

The expected total cost, TCF, of producing a unit is the sum of the two components. Go to step (14).

(14) Check if all runs are finished:

If all runs are processed, go to step (15). Otherwise, begin a new run, return to step (11). For a review of random number generators and runs, see Gottfried [7] and Kleijnen [10].

(15) [illegible]

The step outputs the skeleton of assignment showing the tasks assigned to each machine in their sequence, the components of cost function per produced unit and their total at each run, and system efficiency and efficiency of each machine.

2.4 Alternative Stochastic Cost Function

Another stochastic cost function is developed, to approximate the total operating cost subjected to the assumptions cited above. Two statistical theorems can be used, first multiplication theorem for independent events and second, summation theorem for mutually exclusive events. Let p_j to be the probability that a task j will fail, then the probability, P_j, that the system failure will occur during carrying out the task j will be

$$P_{j\in E} = p_j(1-p_{j-1})(1-p_{j-2})\ldots(1-p_2)(1-p_1) \quad \ldots\ldots\ldots \quad (2.3\text{-}8)$$

If $p_1 = p_2 = \ldots\ldots = p_j = p$, then,

$$P_{j\in E} = p(1-p)^{j-1} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.3\text{-}9)$$

Note that such probability follows a geometric distribution. If it is assumed that the system may fail at any task in the set, E, of a job of tasks, the expected incompletion cost component, ICF, can be evaluated, for a given sequence, by

$$ICF = \begin{cases} \ddot{a}p\ \sum_j (TLB_j)(1-p)^{j-1} & \text{if scrapped} \\ \quad \text{or} & ,j=1,2,\ldots,N \quad \ldots\ldots\ldots \quad (2.3\text{-}10) \\ \ddot{A}p\ \sum_j (TLA_j)(1-p)^{j-1} & \text{if reworked} \end{cases}$$

In a special case, all tasks may have near mean times $\mu_i = \mu$, (TLB_j) and (TLA_j) will follow a straight line. Then, the equation (2.3-10) will be

$$ICF = \begin{cases} \ddot{a}p\mu\ \sum_j (j-1)(1-p)^{j-1} & \text{if scrapped} \\ \quad \text{or} & ,j=1,2,\ldots,N \quad \ldots\ldots \quad (2.3\text{-}11) \\ \ddot{A}p\mu\ \sum_j (N-j+1)(1-p)^{j-1} & \text{if reworked} \end{cases}$$

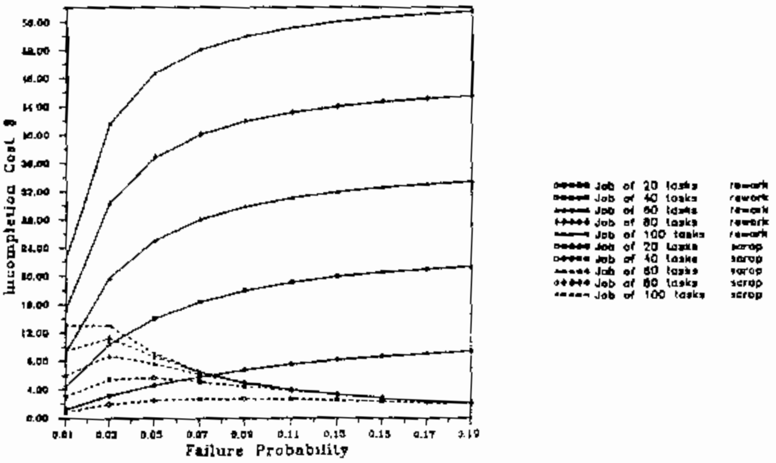

Fig. (2-3) Analytical incompletion cost function.

It is evident that ICF, in scrap type incompletion, is an indirect function of, N, the size of assembly job. Equation (2.3-11) mixes a linear distribution multiplied by a geometric distribution. That can be caught from Fig. (2-3).

Let ICF_s and ICF_r represent scrap type and rework type and if $\ddot{a} = \ddot{A} = a$, then,

$$ICF_r = ap\mu \sum_j [N-(j-1)](1-p)^{j-1}$$

$$= ap\mu \sum_j N(1-p)^{j-1} - ap\mu \sum_j (j-1)(1-p)^{j-1}$$

$$= ap\mu N \sum_j (1-p)^{j-1} - ICF_s \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2.3\text{-}12)$$

Thus, under condition of equal times for the same job, ICF_r always records quantities greater than ICF as shown in Fig. (2-3).

To use this analytical method efficiently, an estimate for p must exist; it can be found from past experience of assembling the similar jobs. An error of p, of course, will lead to erroneous and bad conclusion, therefore, If the planner is not sure about p, he may resort to Monte-Carlo method. A computer program is developed in FORTRAN 77 to handle the proposed system and approach.

Generally, this approach is not concerned with flow lines but it can be extended to manipulate such lines by developing another formulation. The application is made for an engine by using a two -machine cell which is loaded as shown in Table (2-1).

M/C 1 Process		M/C 2 Process		Waiting
Time min.	Task	Time min.	Task	Time min.
3.969	1	-------------		3.969
-------------		7.871	2	7.871
4.143	11	4.170	4	0.07
3.293	7	3.505	6	0.212
2.784	5	2.339	8	0.445
3.890	13	4.589	14	0.699
4.735	17	4.626	10	0.109
4.367	21	4.355	24	0.012
13.725	23	13.647	22	0.120
3.739	9	3.494	20	0.245
4.592	29	4.639	30	0.047
3.135	3	3.397	32	0.262
5.127	31	4.759	12	0.769
2.757	15	3.179	16	0.423
3.035	19	3.007	18	0.031
2.541	27	4.052	26	1.511
2.626	25	-------------		2.626
-------------		2.776	36	2.776
3.891	37	3.657	40	0.234
-------------		4.256	28	4.256
4.387	33	3.282	38	1.105
6.293	35	-------------		6.293
-------------		6 606	34	6.606
10.793	39	-------------		10.793
6.089	41	-------------		6.089
-------------		3.154	42	3.154
5.487	43	5.072	44	0.415
3.817	47	3.965	48	0.147
2.967	45	2.862	56	0.105
2.131	55	2.463	52	0.333
5.413	57	5.395	60	0.017
3.056	67	3.001	66	0.055
5.230	53	5.606	64	0.376
2.966	61	3.075	62	0.109
3.991	65	4.749	46	0.758
2.336	49	2.384	70	0.048
2.997	59	3.103	50	0.105
[illegible]	[illegible]	3.120	54	0 595
3.609	63	2.274	50	1.335
3.745	69	2.291	68	1.455
4.555	51	4.730	72	0.175

Table (2-1) Assignments on two machines.

The engine consists of 72 tasks which are observed from the original line considering stochastic performance as shown in the enclosed Appendix. Then, standard times are set up to accommodate deterministic performance of the automatic assembly. The original line consists of 17 series work stations on the main assembly and 5 work stations on the rear assembly.

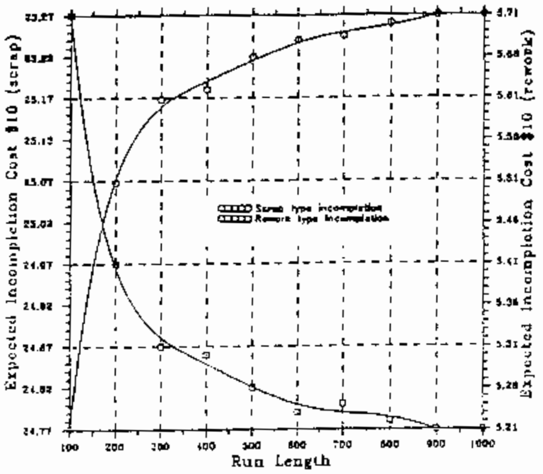

Fig. (2-4) Incompletion cost at different runs.

Fig. (2-4) shows the decay and augmenting of incompletion cost It is clear that the scrap type begins with a smaller cost value then, it goes bigger until reaching its steady state; contrary the rework type does. The scrap type is more costly due to one of two; the failure occurs late (near end tasks), or times of the before tasks are greater than the following task times. Also, the trends are close and behave in acceptable fashion.

3. CONCLUSION

This study involves a complete analysis to assembly processes using a different assembly system which is based on the fixed position layout--through a relevant approach developed to solve such problem. The experiment is carried out using a two-machine system to assemble an engine observed from the Egyptian industry. Also it confirms that the examined system can replace efficiently the flow line. The assignment procedure exhibits an efficient and sensitive routine showing the task scheduling and times including waiting.

The lower bound approach which has been considered to solve the problem of incompletions (breakdowns) results in costly scrap type. Hence, the assignment procedure is powered in retarding such expected failure. Therefore, it is recommended to use the proposed assignment procedure in case of the products which can be reworked off-system, i.e. the failures are not destructive.

The developed cost functions are comprehensive. The analytic stochastic cost function is recommended for the special task times and probabilities of failure because it fully predicts the system response in the two types of incompletions. Furthermore such study opens an attractive interest of research.

REFERENCES

1. Buxy, G.M.; Slack, N.D.; and Wild, R.; "production flow line systems design-a review," AIIE Trans., Vol. 5, March, 1973, pp. 37-48.
2. Buzzcott, J.A.,"Automatic transfer lines with buffer stocks," Int. J. Prod. Res., Vol. 5, No. 3, 1967, pp. 183-200.
3. Feldmann, K., and Roth, N., "Optimization of set-up strategies for operating automated SMT assembly lines," Manufa. Tech. CIRP Annals, Vol. 40, No. 1, 1991, pp. 433-436. Publ. by Int. Inst. for prod. Engng. Res., Switz.
4. Gaimon, C., "Optimal times and levels of acqusition of automation," Ohio State University, Proceedings, TIME-ORSA Conference on Flexible Manufacturing Systems, August, 1984a.
5. Gaimon, C., "The optimal acquisition of automation to enhance the productivity of labour," Mgmt. Sci., Vol. 31, No. 9, 1985, pp. 1175-1190.
6. Gershwin, S.B., and Schick, I.C., "Modeling and analysis of three-stage transfer lines with unreliable machines and finite buffers," Ops. Res., Vol. 31, No. 2, 1983, pp. 354-380.
7. Gottfried, B.S., "Elements of stochastic process simulation," Prentice Hall, Englewood Cliffs, New Jersey, U.S.A., 1984.
8. Groover, M.P., "Automation, Production Systems, and Computer-Aided Manufacturing," Prentice-Hall, Englewood Cliffs, New Jersey, U.S.A., 1980.
9. Kamali, J.; Moodie, C.L.; and Salvendy, G.; "A framework for integrated assembly systems: humans, automation, and robots," Int. J. Prod. Res., Vol. 20, No. 4, 1982, pp. 431-448.
10. Kleijnen, J.P.C., "Analyzing simulation experiments with common random numbers," Mgmt. Sci., Vol. 34, No. 1, 1988, pp. 65-74.
11. Kuula, M.; Stam, A.; Leino, S.; and Ranta, J.; "Workload balancing in the manufacturing environment-a multicriteria tradeoff analysis," Mgmt. systems, Helsinki School of Economics and Business Administration, helsinki,Finland, Working Paper, May, 1993.
12. Law, S.S., "A factorial analysis of automatic transfer line systems," Int. J. Prod. Res., Vol. 21, No. 6, 1983, pp. 827-834.
13. Shin, K.G., and Zheng, Q., "Scheduling job shop operations in an automatic assembly line," IEEE Trans. on Robotics and Automation, Vol. 7, No. 3, 1991, pp. 333-341.
14. Smith, J.M., and Daskalaki, S., "Buffer space allocation in automated assembly lines," Ops. Res., Vol. 36, No. 2, 1988, pp. 343-358.

APPENDIX

Task Time Observations as a Computer Input

NO.	15 OBSERVATIONS (Time in Minutes)
1-	4.503.493.504.523.514.534.483.404.603.504.513.473.444.493.60
2-	8.706.807.818.697.757.796.608.758.847.617.666.807.827.708.74
3	2.802.812.902.803.912.003.852.813.863.782.902.003.003.823.78

```
 4-   4.003.103.112.805.793.504.203.155.183.005.804.804.005.904.22
 5-   3.601.502.512.602.703.613.551.583.802.812.751.503.402.453.40
 6-   3.802.902.912.883.842.704.602.803.864.814.003.102.694.882.80
 7-   2.602.592.603.624.703.504.662.553.722.502.772.594.582.803.61
 8-   3.102.903.002.102.401.123.221.152.182.162.153.132.203.111.16
 9-   4.403.504.513.412.304.353.334.114.103.223.004.503.664.802.90
10-   5.803.903.603.773.804.005.104.115.864.705.814.844.004.106.00
11-   5.103.004.203.102.904.913.105.104.153.183.226.304.993.905.00
12-   3.305.306.304.443.606.203.115.224.003.403.365.333.004.604.22
13-   4.202.006.103.305.334.122.383.234.295.503.603.403.554.223.13
14-   3.204.005.003.603.805.334.436.504.104.004.903.705.664.616.00
15-   2.801.701.712.601.662.652.682.643.811.883.862.903.813.832.82
16-   3.001.912.234.191.903.882.893.982.302.884.103.134.114.003.19
17-   5.006.904.883.504.225.343.903.883.704.664.005.104.336.115.50
18-   3.802.661.703.882.601.811.902.833.852.783.752.903.912.863.82
19-   2.202.253.292.203.213.182.194.303.302.224.282.213.204.253.24
20-   4.003.902.884.304.123.222.804.183.152.903.002.983.004.883.10
21-   5.203.804.603.715.165.184.334.184.503.153.405.304.004.005.00
22-   13.514.313.815.014.012.814.612.012.913.013.014.515.013.412.9
23-   12.714.012.513.014.115.315.014.912.613.313.012.014.713.212.6
24-   5.803.904.883.905.803.884.883.884.004.204.104.004.203.914.00
25-   2.503.001.502.202.302.202.333.602.901.882.773.602.503.512.60
26-   4.303.313.324.004.403.415.503.355.514.413.324.333.205.213.21
27-   2.202.203.182.222.152.184.213.162.302.253.242.232.212.202.18
28-   5.303.314.603.503.405.203.324.505.413.404.605.303.503.305.20
29-   4.106.105.103.913.904.923.885.904.104.005.894.963.924.004.20
30-   5.503.504.405.513.524.493.593.585.604.516.404.413.565.505.51
31-   5.306.334.324.325.306.606.503.335.314.414.425.446.414.424.50
32-   4.204.003.402.604.602.133.222.194.233.333.153.003.204.403.30
33-   3.605.514.544.005.603.614.704.714.005.103.903.884.404.603.66
34-   7.406.005.506.336.405.906.007.556.346.007.407.806.777.606.10
35-   6.405.225.107.006.606.007.907.186.005.106.406.007.006.805.70
36-   2.903.982.002.102.802.882.911.902.922.952.802.812.802.893.00
36-   4.803.803.805.203.153.123.003.904.003.605.104.403.413.193.90
37-   2.903.883.923.902.913.882.812.862.852.902.862.913.882.843.93
38-   10.010.711.111.010.910.611.011.410.810.710.911.010.210.611.0
39-   3.304.554.602.533.544.522.533.603.603.802.704.613.642.804.54
40-   6.005.106.405.905.005.117.006.306.185.806.896.506.185.987.00
41-   2.802.703.003.802.882.803.863.003.103.102.703.783.002.893.90
42-   6.205.105.205.005.256.245.205.106.006.225.235.195.185.196.00
43-   6.205.005.104.994.905.005.225.203.994.005.155.155.184.906.10
44-   2.503.443.492.603.553.392.502.402.503.552.553.442.502.603.50
45-   6.005.104.225.003.904.813.894.005.104.905.224.804.884.205.22
46-   4.704.663.704.712.752.783.003.704.693.683.803.803.753.743.80
48-   4.704.615.803.764.503.603.402.553.614.703.803.764.502.553.63
49-   2.601.591.592.612.622.591.582.602.622.632.612.582.562.601.66
50-   1.602.662.651.612.621.592.612.632.602.552.591.572.601.612.62
51-   4.904.104.003.903.914.925.003.984.105.205.213.905.914.205.10
52-   2.902.912.881.921.901.851.842.792.802.812.911.802.921.812.91
53-   6.504.515.555.524.605.395.496.405.524.554.514.515.494.395.52
54-   3.502.503.503.902.342.332.552.903.512.563.543.542.343.903.89
55-   1.901.881.871.901.881.861.902.861.812.822.801.881.872.821.91
56-   3.502.623.402.503.562.413.592.602.412.402.602.632.612.593.51
57-   4.505.505.505.905.714.666.405.414.905.715.605.405.445.664.90
58-   2.802.903.812.822.862.853.803.812.902.803.812.832.872.822.86
59-   2.403.403.392.403.393.412.382.393.363.383.412.423.402.423.41
```

```
0-    5.505.505.444.504.604.615.556.485.534.526.445.505.605.615.55
1-    2.503.512.553.512.493.472.502.492.523.532.502.543.523.423.44
2-    3.502.482.523.503.513.533.542.523.512.483.652.662.692.553.48
3-    4.703.003.102.792.783.753.764.003.703.104.663.663.104.153.89
4-    6.505.606.495.515.535.554.524.534.606.495.515.535.556.485.70
5-    4.703.632.802.813.704.653.793.802.884.005.004.704.723.884.81
6-    3.503.512.493.513.592.602.523.562.533.512.502.593.492.602.52
7-    2.503.603.502.603.512.523.493.483.492.513.492.602.532.543.48
8-    2.401.412.392.501.501.442.402.492.422.452.392.382.382.403.41
9-    4.704.604.622.593.003.712.703.593.603.634.653.903.982.914.00
0-    2.902.812.801.901.901.881.822.811.802.831.832.851.862.902.87
1-    3.704.004.883.614.603.003.663.703.712.883.612.903.913.913.66
2-    5.803.824.813.795.804.813.865.884.853.815.805.004.104.004.82
```

cknowledgement-The authors appreciate the valueadding cooperation ound by the staff supervise ward of "6-Cylinder V Type Engine" in asr Cars Manufacturing Co.; also, the management of the company. lso, we are indebted to the referees for their advices.