Mixed Model Production Sequencing Problem: A Genetic Algorithm Based Approach

Abdul Nazar K. P, V. Madhusudanan Pillai
Department of Mechanical Engineering,
NIT Calicut, Kerala, India
nazarppns@gmail.com
vmp@nitc.ac.in

Abstract— In Mixed Model Production systems a variety of similar product models are assembled in a mixed fashion. Sequencing these product models on a production line, termed as MMP sequencing problem belongs to the NP hard combinatorial optimization class. This paper proposes a genetic algorithm based approach for solving the MMP sequencing problem considering the objectives of minimizing the production rates variation and number of set-ups. The method is applied for diverse sets of problems of small and medium sizes. The performance of the algorithm is compared with the results for the same problem sets using other methods reported in the literature. The computational results reveal that the present algorithm gives better sequences in small and medium sized problems.

Keywords— mixed model production sequencing; optimization; production rates variation; set-ups; genetic algorithm.

I. INTRODUCTION

Mixed Model Production (MMP) or Mixed Model Assembly Line (MMAL) is a tool to achieve level production in Just-In-Time (JIT) manufacturing. It is the production of multiple kinds of products on a repetitive basis, in a mixed fashion and on a single line or station [1]. MMP systems enable manufacturers meet demand for a variety of products in an efficient way. The major issues to be addressed in MMP systems can be grouped into two categories: (i) Designing and balancing the line and (ii) Sequencing the different product models. In this paper, only the sequencing problem is considered for an already balanced line. This involves determining the sequence of introducing models to the production line, while considering crucial organizational goals for implementing JIT production system [2]. Researchers have addressed the MMP sequencing problem considering various objectives such as (i) leveling the load on each process within the line, keeping a constant usage rate of every part used by the line [2]; (ii) minimizing the production rates variation [3, 4], (iii) total utility work [5], (iv) total setup cost [6], (v) the risk of stopping a conveyor [7], (vi) the overall line length [8], and so on. However major emphasis was on minimizing production rates variation and minimizing total utility work. A large number of researchers have approached the mixed model sequencing problem with the objective of keeping a constant usage rate of every part used by the line. The first mathematical model of the Toyota system was proposed by Miltenburg [3]. This model formulated the MMP sequencing problem as a non-linear integer programming problem with the objective of minimizing the total deviation of actual production rates from the desired production rates.

The MMP sequencing is an NP-hard combinatorial optimization problem. Computational time is a critical factor in sequencing since it is a short term planning. Hence a number of researchers have used evolutionary algorithms and other heuristic procedures in solving such problems. Hyun et al. [9] carried out the first research on the application of Genetic Algorithm (GA) in MMP sequencing problem. Ponnambalam et al. [10] studied the performance of the selection mechanisms and showed that a GA that uses the Pareto stratum-niche cubicle performs better than a GA with other selection mechanisms.

This paper proposes a GA based approach for the sequencing problem for a given product mix in a mixed model production system. The method solves the problem under two objective scenarios such as (i) minimizing the production rates variation, and (ii) a linear combination of number of setups and production rates variation. Two sets of 10 test problems ranging from low to high possible number of solutions are solved using the proposed method for the purpose of evaluation of the method.

II. THE PROBLEM STATEMENT

The problem addressed in this paper is the optimization of MMP sequences with just-in-time objectives. The objective functions considered are described below.

A. *Minimizing the Production Rates Variation*

Continual and stable part supply can be realized when the demand rate of parts is constant over time. This objective is significant to a successful operation of the system. Under the assumption that all products require the same number and mix of parts in the model, the variation in production rates of the final product achieves minimum in all parts usage rates. Thus, the objective can be achieved by matching demand with the actual production. In this paper, the following model is used which is found in Mansouri [4].

$$U = \sum_{k=1}^{D_T} \sum_{i=1}^{a} \left(x_{i,k} - k \frac{d_i}{D_T} \right)^2 \quad (1)$$

U = Production rates variation of a production sequence

a = Number of unique products to be produced

d_i = Demand for product i, i=1,2,...,a

D_T = Total number of units for all products

$x_{i,k}$ = Total number of units of product i produced over stages 1 to k, k = 1,2,...,D_T

B. Minimizing the Number of Set-ups

The second objective considered here is the number of set-ups required as in (2)

$$S = 1 + \sum_{k=2}^{D_T} s_k \tag{2}$$

s_k=1 if set-up is required at stage k, otherwise $s_k = 0$.

The two parameters of an MMP sequence, U and S are conflicting. Lower production rates variation could be achieved by increasing number of setups. But it is not desirable to obtain reduced production rates variation at the expense of large number of setups, if set-up times are not negligible. Thus, a solution approach would be desirable that provides the decision-makers with the most appropriate sequences considering both the objectives. For example, consider a situation where 5 products as A, B, C, D and E are to be produced (a = 5), where demand for the items is as follows: $d_1 = 6$, $d_2 = 3$, $d_3 = 1$, $d_4 = 1$ and $d_5 = 1$ resulting a total demand DT = (6 + 3 + 1 + 1 + 1) = 12. Consider two possible solutions for this problem as X1 = BBBCAAAAAAED and X2 = ABACADEABABA. From the equations the number of setups (S) and production rates variation (U) X1 is 5 and 40.83 respectively. For the sequence X2, these parameters are 12 and 7.67 respectively. Therefore the decision maker has to assign suitable weights to these parameters to select the optimal sequence as per the organizational objectives. A composite objective function can be formulated as

$$Min. Z = w_s S + w_u U \tag{3}$$

where w_s and w_u are weights assigned for S and U respectively.

In this work two methods are used for solving the MMP sequencing problems. In the first method only the production rates variation U is considered as the objective function which is same as the heuristic 2 provided by McMullen (1998). In the other approach a linear composite objective function of S and U with weights 14.2755 and 3 as provided by McMullen (1998) is considered. The weight 14.2755 was determined after sampling. Several solutions were sampled across many problems and the coefficients were determined such that the number of setups and the material usage rate made the desired contributions to the objective function. So in the second method U is contributing three times as compared to the number of set-ups in the objective function.

III. PROPOSED GENETIC ALGORITHM BASED APPROACH

A. Representation and Initialization

A genetic algorithm based approach for the MMP sequencing problem is coded in Scilab version 5.4.1. In GA, each chromosome represents a repeating sequence of models (e.g. ABABACAD or AAAABBCD) and each gene in the chromosome represents the individual model A, B, C or D. Once the number of units in a cycle is known, the initial population of random feasible solutions can be generated. Each initial solution is merely a different permutation of the feasible number of models. Initially, a population of random sequences is generated whose size is equal to the chromosome length.

B. Fitness Function and Selection

A fitness or evaluation function is used to evaluate and select the better performing solutions which themselves become candidates for improvement using the genetic operations. The specific form of evaluation function depends on the objective function being considered. The fitness of an individual solution dictates the number of copies of that solution in the mating pool. The more copies an individual receives, the greater is the probability that the characteristics will be repeated in subsequent generations. Since the objective function considered in the present problem is a minimization problem, a transfer function is used to map this to a fitness function. The transfer function used in this paper is

$$F_i = T_{\max} - T_i \tag{4}$$

Where F_i is the fitness function of string i, T_i is the objective function value of a sequence i and T_{max} is the largest objective function value in the current generation.

The reproduction operator is used to select individuals from the current population to become parents of the next generation. Parents are selected according to their fitness value. Rowlette wheel selection is used as the selection process. According to this method, the probability of selection of a particular sequence p_i is calculated as

$$P_i = \frac{F_i}{\sum F_i} \tag{5}$$

Where, F_i is the fitness value of the sequence i.

C. Genetic Operations

Here three genetic operators, cross over, inversion and mutation are used. Randomly 60% of the sequences in the mating pool go for crossover and mutation operations, while 40% go for inversion and mutation operations.

1) Crossover: The modified order crossover (modOX) developed by Hyun et al. [10] is used. The elements from the mating pools are selected in pairs and they undergo crossover with a crossover probability 0.8. Two crossover points are randomly selected from both the parents. The elements between crossover points in one parent(P1) are copied into an offspring O1 in the same position as they appear in P1. Then the copied elements are randomly deleted from the other parent P2 and the remaining elements in P2 are copied into the undetermined positions in the offspring in the same order as they appear in P2. The second offspring is created by alternating the roles of the two parents. The modOX crossover would create offspring that would preserve the relative order in parents. An example is given in fig.1

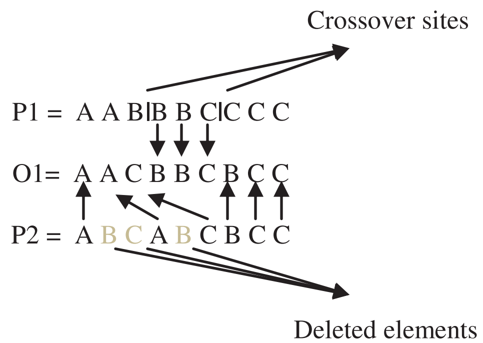
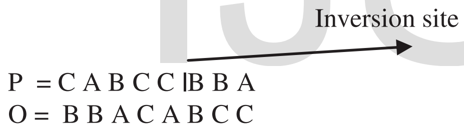


Figure 1. modOX crossover

2) Inversion: Individual chromosomes are selected from the mating pool to undergo inversion operation with an inversion probability 0.8. A single inversion site is generated randomly in the parent string. Then the offspring is formed by inverting the string about the inversion site. The elements after the inversion site in the parent string are copied to the offspring to start from the first position. Then the remaining positions are filled with the elements of the parent string coming before the inversion site in the same order. An example is given below.



3) Mutation: After crossover or inversion operations, each string is selected for the mutation operation. A position in a string is selected for mutation with a mutation probability of 0.1. The element in the selected position of the string is randomly exchanged with another element in the same string. The mutation operation helps to extend the search into previously unexplored areas of solution space.

D. Replacement Strategy and Termination

When the genetic operations are completed for the present generation, the offspring are evaluated based on the objective function and compared with the existing solutions in the mating pool. Those solutions which are well performing compared with the current solution is admitted to the new population. When succeeding iterations which do not improve the solution for a user defined number (UDN) of generations, the algorithm is terminated. Here it is taken as 25.

IV. TEST PROBLEMS

Two problem sets (see Tables I and II) from McMullen [11] are used as test problems in this paper. These problem sets belong to the category of small and medium size MMP sequencing problems, and are widely used by the researchers to compare the performance of different algorithms. Each problem set consists of 10 different problems with varying demands for the individual products.

TABLE I: PROBLEM SET 1

Problem	Demand for Product Type					Possible No. of Solutions
	1	2	3	4	5	
A	20	0	0	0	0	1
B	16	1	1	1	1	1.163×10^5
C	15	2	1	1	1	9.302×10^5
D	15	2	1	1	1	1.628×10^7
E	10	5	2	2	1	1.397×10^9
F	8	7	2	2	1	2.993×10^9
G	6	6	5	2	1	1.995×10^{10}
H	5	5	5	3	2	1.173×10^{11}
I	5	4	4	4	3	2.444×10^{11}
J	4	4	4	4	4	3.055×10^{11}

TABLE II: PROBLEM SET 2

Problem	Demand for Product Type										Possible No. of Solutions
	1	2	3	4	5	6	7	8	9	10	
A	20	0	0	0	0	0	0	0	0	0	1
B	11	1	1	1	1	1	1	1	1	1	6.095×10^{10}
C	10	2	1	1	1	1	1	1	1	1	3.352×10^{11}
D	9	3	1	1	1	1	1	1	1	1	1.117×10^{12}
E	8	4	1	1	1	1	1	1	1	1	2.514×10^{12}
F	7	5	1	1	1	1	1	1	1	1	4.023×10^{12}
G	6	5	2	1	1	1	1	1	1	1	1.408×10^{13}
H	5	5	3	1	1	1	1	1	1	1	2.816×10^{13}
I	4	4	4	2	1	1	1	1	1	1	2.112×10^{15}
J	2	2	2	2	2	2	2	2	2	2	2.376×10^{15}

V. COMPUTATIONAL EXPERIMENTS

The proposed algorithm is coded in Scilab 5.4.1 and run on a core i5 processor at 2.60 GHz with Windows 7 and 4 GB RAM. The program is validated by solving the example problems in Mansouri [4] which resulted in getting the same optimal values.

The problems given in Tables I and II are solved for the two objective functions separately resulting in a total of 40 runs. Objective function 1 is Minimise $Z = U$ and objective function 2 is Minimise $Z = 14.2755\ S + 3\ U$. For each of the problems, the program outputs are the best sequence, its objective function value, corresponding values for the sequence parameters (*S* and *U*) and the CPU time. Table III and Table IV show the results with objective function 1 for problem set 1 and 2 respectively. Table V and Table VI show the respective values with objective function 2. For evaluating the proposed algorithm, the results (mean and standard deviations of *S* and *U* for each problem set) are compared with those obtained in McMullen [11]. Table VII shows the comparison for both the objective functions separately. The algorithm is also tried for a large sized problem set, but the performance was not good with respect to computational time.

VI. EXPERIMENTAL RESULTS

We used the Genetic Algorithm based approach to solve small and medium sized sequencing problems (Table I and II) in mixed model production systems. The computational experiments show that the present method finds better sequences in terms of both the objective function considered. The first objective is to minimise the production rates variation (U) alone, and the second objective is to minimize a linear combination of both production rates variation and the number of set-ups required for the sequence(s).

Tables III – VI show the best sequence identified by the method, the objective function values, U and S for these sequences and the computational time. The mean values of the parameters U and S for each problem set is used for comparison study. The comparison of the results with those reported in McMullen (1998) reveals that the proposed method is efficient to solve MMP sequencing problems of small and medium size (Table VII). The method performs equally well with objective functions 1 and 2 with a comparable computational time.

TABLE III: SOLUTIONS FOR PROBLEM SET 1 WITH OBJECTIVE FUNCTION1

Problem	Best Sequence Found	PRV (U)	Setups (S)	CPU time (sec.)
A	1	0	1	0
B	1 1 1 1 3 1 1 1 2 1 1 5 1 1 1 1 4 1 1 1	13.50	9	13.67
C	1 1 2 1 1 1 3 1 1 1 5 1 1 4 1 1 1 2 1 1	11.00	11	19.50
D	1 2 1 1 4 1 1 2 1 3 1 1 2 1 5 1 1 1 2 1	11.70	15	21.09
E	1 2 1 3 4 1 2 1 1 2 5 1 1 2 1 4 3 1 2 1	9.85	18	37.67
F	1 2 3 4 1 2 3 2 1 5 3 1 2 3 1 2 4 3 2 1	10.25	20	38.14
G	2 1 3 4 2 1 3 1 2 3 5 2 1 3 2 1 4 3 1 2	10.25	20	33.73
H	3 2 1 4 5 3 2 1 4 3 2 1 3 2 1 5 4 1 3 2	11.80	20	42.90
I	1 3 4 2 5 1 3 4 2 5 1 4 3 2 1 5 4 3 2 1	11.35	20	28.83
J	1 3 4 5 2 5 3 4 1 2 5 1 4 3 2 5 3 4 1 2	16.00	20	9.25
	Mean	10.57	15.4	23.68
	Standard deviation	4.14	6.50	

However the performance of the algorithm for large sized problems is poor, in terms of the computational time. Since it is an operational level problem, computational time is a critical factor in mixed model production sequencing. Hence the application of the proposed method is limited to small and medium sized MMP sequencing problems.

TABLE IV: SOLUTIONS FOR PROBLEM SET 2 WITH OBJECTIVE FUNCTION 1

Pr.	Best Sequence Found	PRV (U)	Set-ups (S)	CPU time (sec.)
A	1	0	1	0
B	1 8 1 3 1 10 1 9 1 1 6 7 1 1 5 1 4 1 2 1	30.75	18	21.81
C	1 2 1 6 1 9 1 8 1 7 4 1 5 1 10 1 3 1 2 1	26.80	20	40.47
D	1 2 1 4 9 1 3 1 6 2 1 10 1 5 7 1 8 1 2 1	27.15	20	25.27
E	1 2 1 10 5 1 2 7 1 4 6 1 9 2 1 3 8 1 2 1	27.20	20	72.07
F	1 2 9 1 2 10 1 6 4 2 1 5 8 1 3 2 1 7 2 1	27.55	20	38.13
G	1 2 3 1 5 2 8 7 1 2 10 1 9 4 6 2 1 3 2 1	25.00	20	40.28
H	2 1 3 4 2 1 7 8 10 1 2 3 5 6 2 1 9 3 1 2	25.75	20	34.82
I	2 3 1 4 8 3 1 2 5 9 10 6 3 1 2 7 4 1 3 2	24.15	20	32.78
J	1 3 2 8 4 7 5 9 6 10 7 3 10 4 2 9 6 1 5 8	33.00	20	12.47
	Mean	24.73	17.9	31.81
	Standard deviation	9.08	5.97	

TABLE V: SOLUTIONS FOR SET 1 WITH OBJECTIVE FUNCTION 2

Pr.	Best Sequence Found	Z	PRV (U)	Set-ups (S)	CPU time (sec.)
A	1	14.28	0	1	0
B	1 1 1 1 1 2 5 1 1 1 1 1 1 3 4 1 1 1 1 1	146.43	15.50	7	14.60
C	2 1 1 1 1 1 1 1 3 5 4 1 1 1 1 1 1 1 1 2	154.53	18.20	7	17.39
D	1 1 1 2 2 1 1 1 4 3 1 1 1 5 2 2 1 1 1 1	184.28	18.60	9	20.84
E	2 3 1 1 1 1 4 2 2 5 1 1 1 1 4 2 2 3 1 1	213.88	18.95	11	28.92
F	3 2 2 1 1 1 1 2 2 4 4 5 1 1 2 2 2 1 1 3	221.51	26.25	10	49.86
G	2 1 1 4 3 3 3 2 2 2 1 1 1 5 4 3 3 2 2 1	234.28	25.75	11	29.36
H	5 2 2 1 1 3 3 3 4 4 1 1 2 2 2 5 3 3 1 4	253.63	32.20	11	24.91
I	3 5 1 1 2 2 4 4 3 5 1 1 2 4 4 3 3 5 2 1	267.21	22.45	14	40.59
J	4 2 3 3 1 1 5 5 4 4 2 2 1 5 5 3 3 1 4 2	263.58	26.00	13	15.98
	Mean	195.36	20.39	9.4	24.25
	Standard deviation	76.54	8.74	3.71	

TABLE VI: SOLUTIONS FOR SET 2 WITH OBJECTIVE FUNCTION 2

Pr.	Best sequence found	Z	PRV (U)	Set-ups (S)	CPU time (sec.)
A	1	0	0	1	0
B	7 1 1 1 1 5 8 2 9 1 1 1 4 10 3 1 1 1 1 6	278.56	35.75	12	17.79
C	2 1 1 1 6 3 9 4 1 1 1 1 8 7 10 5 1 1 1 2	282.78	32.4	13	29.88
D	2 1 1 1 9 5 6 8 7 1 1 1 1 3 2 2 10 4 1 1	297.03	37.15	13	19.76
E	6 1 1 1 2 2 9 4 7 3 1 1 1 10 8 5 2 2 1 1	298.38	37.6	13	38.92
F	1 1 2 2 10 3 9 1 1 4 8 5 2 2 6 1 1 1 7 2	304.71	34.95	14	42.04
G	3 1 1 2 2 6 9 8 7 1 1 5 2 2 4 10 1 1 3 2	307.73	31.2	15	35.64
H	2 10 1 1 3 3 9 2 2 8 6 5 4 7 1 1 1 2 2 3	316.11	38.75	14	29.18
I	4 1 1 2 2 3 3 8 10 9 5 7 6 3 1 1 2 2 4 3	320.78	35.55	15	42.13
J	9 1 6 8 10 4 3 7 2 2 5 5 8 7 3 10 4 1 9 6	361.96	35	18	13.37
	Mean	276.80	31.83	12.8	26.87
	Standard deviation	99.99	11.41	4.47	

TABLE VII: COMPARISON OF SOLUTIONS WITH THOSE IN MCMULLAN (1998)

Problem Set (Objective Function)	Results Obtained by GA Method Mean (Std. Deviation)		Results in McMullan Mean (Std. Deviation)	
	PRV	Set-ups	PRV	Set-ups
1 (Objective function1)	10.57 (4.14)	15.4 (6.5)	11.08 (4.2)	15.6 (6.2)
1 (Objective function 2)	20.39 (9.40)	9.4 (3.71)	22.73 (9.6)	9 (3.2)
2 (Objective function 1)	24.73 (9.08)	17.9 (5.97)	25.1 (8.7)	17.8 (5.6)
2 (Objective function 2	31.83(11.41)	12.8 (4.47)	32.6 (11.2)	12.8 (4.2)

VII. CONCLUSIONS

The developed algorithm performs well with respect to the objective functions considered for small and medium sized MMP sequencing problems. The computational efficiency of the algorithm is also good for these problems. Comparison with the results reported in literature demonstrates that the algorithm can identify better sequences in mixed model production. The combinatorial nature of the problem makes the traditional optimization techniques impractical because of the limitation on the computational time. The developed algorithm uses genetic operators such as selection, crossover, inversion, mutation and replacement strategy. However the algorithm has to be improved so that it can address large sized problems also.

References

[1] Nicholas, J. M. (1998). *Competitive manufacturing management: continuous improvement, lean production, customer-focused quality*. Irwin/McGraw-Hill.

[2] Monden, Y. (1983). *Toyota production system: practical approach to production management* Norcross, GA: Industrial Engineering and Management Press, Institute of Industrial Engineers.

[3] Miltenburg, J. (1989). Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science*, *35*(2), 192-207.

[4] Mansouri, S. A. (2005). A multi-objective genetic algorithm for mixed-model sequencing on JIT assembly lines. *European Journal of Operational Research*, *167*(3), 696-716.

[5] Yano, C. A., & Rachamadugu, R. (1991). Sequencing to minimize work overload in assembly lines with product options. *Management Science*, *37*(5), 572-586.

[6] Burns, L. D., & Daganzo, C. F. (1987). Assembly line job sequencing principles. *International Journal of Production Research*, *25*(1), 71-99.

[7] Okamura, K., & Yamashina, H. (1979). A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the conveyor. *International Journal of Production Research*, *17*(3), 233-247.

[8] Bard, J. F. & Shtub, A. (1992). An analytic framework for sequencing mixed model assembly lines. *The International Journal of Production Research*, *30*(1), 35-48.

[9] Hyun, C. J., Kim, Y., & Kim, Y. K. (1998). A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines. *Computers & Operations Research*, *25*(7-8), 675-690.

[10] Ponnambalam, S. G., Aravindan, P., & Subba Rao, M. (2003). Genetic algorithms for sequencing problems in mixed model assembly lines. *Computers & Industrial Engineering*, *45*(4), 669-690.

[11] McMullen, P. R. (1998). JIT sequencing for mixed-model assembly lines with setups using Tabu search. *Production Planning & Control*, *9*(5), 504-510.