A SIMULATED ANNEALING APPROACH FOR BALANCING AND SEQUENCING OF MIXED-MODEL U-LINES

Yakup Kara, Uğur Özcan, Ahmet Peker
Selçuk University, Department of Industrial Engineering, Campus 42075, Konya

Abstract: This study proposes a simulated annealing-based approach for simultaneously solving balancing and sequencing problems of mixed-model U-lines. The proposed approach is capable of minimizing the number of workstations required on the mixed-model U-line as well as minimizing absolute deviation of workloads (*ADW*) among workstations. An experiment to evaluate the performance of the algorithm is also presented.

1. Introduction

U-lines have become very popular in manufacturing and scientific environments as a consequence of continuous improvement and cost reduction efforts of Just-In-Time (JIT) production. Many benefits of U-lines utilized in a JIT environment were reported in the literature (Hall, 1983; Monden, 1993). U-lines improve visibility and communication skills between operators, reduce the number of operators, and facilitate problem-solving and efforts to adjust to changes in the external environment of the firm (Miltenburg, 1998). Miltenburg (2001) studied 114 Japanese and American real U-lines and reported impressive benefits of U-line implementations, including increasing productivity, reduced work-in-process inventory, shorter throughput, and improved quality. Miltenburg (2001) also stated that almost three-quarters of real U-lines produce more than one product. This type of production is called as *mixed-model production* and U-lines on which mixed-model production is performed are called as *mixed-model U-lines.* A successful implementation of a mixed-model U-line requires solutions for two important problems called Mixed-Model U-line Line Balancing (MMU/LB) and Mixed-Model U-line Model Sequencing (MMU/MS). These two problems are tightly interrelated with each other and very different compared with the mixed-model straight lines (Sparling and Miltenburg, 1998; Kim et al., 2000). The former problem, MMU/LB, is the problem of assigning tasks to an ordered sequence of workstations on the mixed-model U-line in such a way that some performance measures are optimized. The latter problem, MMU/MS, is the problem of determining production sequence of models produced on the mixed-model U-line. This paper proposes an approach to solve MMU/LB and MMU/MS problems simultaneously.

2. Problem Statement

U-lines considered in the study work under the following assumptions:

- Product models with similar production characteristics are produced on the same U-line.
- Precedence diagrams of different models are known. The combined precedence diagram concept of Macaskill (1972) is employed.
- Task completion times are deterministic and independent of the assigned workstation.
- Common tasks among different models exist. A task’s completion time may differ from one model to another.
- Parallel tasks and parallel workstations are not allowed.
- No work-in-process inventory is allowed.
- The travel times of workers are ignored.

The problem considered in the current study is to find such a U-line balance and a model sequence that some performance measures are optimized. This combined problem is called as Mixed-Model U-line Balancing and Sequencing problem (MMU/BS) in the literature (Kim et al., 2000; Miltenburg, 2002). As other types of line balancing, this problem is also NP-hard. The combinatorial nature of this problem makes it difficult to solve when the problem size increases and forces us to develop approximate algorithms to obtain optimal or near-optimal solutions to the problem. This study proposes a simulated annealing-based heuristic approach to MMU/BS. The proposed approach is capable of minimizing the number of workstations and the *ADW* of the line balance while all of the constraints are satisfied.

3. The Proposed Simulated Annealing Approach

The main idea of Simulated Annealing (SA) is the analogy between physical annealing process of solids and solution process of large-scale combinatorial optimization problems. SA is initialized with choosing a proper energy (objective) function (E), an initial solution (S_0), an initial temperature (T_0), a cooling rate (R), and the number of iterations for each level of temperature (IT). Then the annealing process is started and maintained until the stopping criteria (T_{min}) is met (i.e. the system is frozen). The initial solution becomes the first current solution (S_c) and the first best solution (S_{best}); the cost of this solution (E_0) becomes the current value of objective function (E_c) and the best value of objective function (E_{best}). Then, a neighbour solution (S_n) of S_c is generated randomly; the cost of S_n (E_n) and the difference (Δ) between E_n and E_c is calculated. If Δ is negative then S_n is accepted and replaced with S_c. If Δ is positive then the Metropolis criterion is applied, S_n is accepted and replaced with S_c with the probability of exp ($-\Delta/T_c$). Otherwise, S_c remains unchanged. This searching process is repeated IT times for each level of temperature (T_c) until the temperature level being less than T_{min}. The proposed SA-based algorithm is described by the following steps:

Step 0. Specify the problem parameters. Generate the first temporary line balance (LB_0) and an initial model sequence (MS_0). Call this solution as current temporary solution (S_t) and feasible solution (S_f). Calculate the cost of current temporary solution (E_t). Set $E_f = E_t$, $k_1 = 0$.

Step 1. Generate a new temporary solution (S_t') by combining two successive workstations of S_f. Calculate the cost of new temporary solution (E_t').

Step 2. If S_t' is feasible then set $S_t = S_t'$, $S_f = S_t'$, $E_t = E_t'$, $E_f = E_t'$ and go to Step 1. Otherwise set $k_2 = 0$ and go to Step 3.

Step 3. Initialization of SA: Set $S_0 = S_t'$. Calculate the cost of initial solution (E_0); set $S_c = S_0$, $S_{best} = S_0$, $E_c = E_0$, $E_{best} = E_0$; set $n = 1$, $T_0 = T_c = 1000$, $T_{min} = 1.00$, $IT = 10$, $R > 0$.

Step 4. Generate a neighbour solution (S_n) and calculate the cost of this neighbour solution (E_n).

Step 5. Calculate the difference (Δ) between cost of neighbour solution (E_n) and cost of current solution (E_c) by the following equation:

$$\Delta = E_n - E_c \tag{1}$$

Step 6. If $\Delta \le 0$ then accept the neighbour solution as new current solution, set $S_c = S_n$, $E_c = E_n$ and go to Step 8.

Step 7. If $\Delta > 0$ then accept the neighbour solution as new current solution with the probability of exp ($-\Delta/T_c$) and set $S_c = S_n$, $E_c = E_n$. Otherwise S_c and E_c are not changed.

Step 8. If $E_c < E_{best}$ then $S_{best} = S_c$; $E_{best} = E_c$. Otherwise S_{best} and E_{best} are not changed.

Step 9. If S_{best} is feasible then set $S_f = S_{best}$, $E_f = E_{best}$ and go to Step 10. Otherwise go to Step 11.

Step 10. If $k_1 = 0$ then terminate SA and go to Step 1. If $k_1 = 1$ AND $k_2 = 0$ then set $k_1 = 0$, terminate SA and go to Step 1. If $k_1 = 1$ AND $k_2 = 1$ then go to Step 11.

Step 11. If $n = IT$ then go to Step 12. Otherwise, set $n = n + 1$ and go to Step 4.

Step 12. Set $T_c = T_c \times R$, $n = 1$.

Step 13. If $T_c \ge T_{min}$ then go to Step 4. Otherwise go to Step 14.

Step 14. If $k_1 = 0$ then set $k_1 = 1$, $k_2 = 1$, $S_t' = S_f$ AND go to Step 3. If $k_1 = 1$ and $k_2 = 1$ then go to Step 1. If $k_1 = 1$ AND $k_2 = 0$ then Stop.

The essential purpose of the proposed SA-based algorithm is to find a feasible line balance by distributing the workloads among workstations smoothly. An infeasible solution with a fixed number of stations and a model sequence is used as an input to SA algorithm for smoothing. If the smoothing process yields a feasible solution then another infeasible solution is generated by using modified version of Erel et al.'s (2001) Solution Generator (SG) mechanism. We modified the SG of Erel et al. (2001) by combining successive workstations that yield the minimum total average workload to obtain a temporary U-line balance. The initial model sequence of mixed-model U-line is generated randomly.

A neighbour solution can be either a new line balance or a new model sequence. A new line balance is generated by reallocating task among workstations while a new model sequence is generated by changing positions of models in model sequence. The neighbourhood generation logic adopted in the proposed SA algorithm is depicted in Figure 1.

The neighbour solution generator logic given above allows us to consider line balancing and model sequencing problems of mixed-model U-lines simultaneously.

In mixed-model U-lines, *ADW* is a measure of model imbalance (or smoothness) and also used by Sparling and Miltenburg (1998) and Kim et al. (2000) as an objective function. In our SA-based algorithm, *ADW* is used as the cost function (E) to evaluate the quality of solutions generated.

Since SA is a part of the proposed approach; it can start and stop more than once. SA stops when a feasible solution is found or the system is frozen, then the proposed algorithm is maintained or terminated depending on k_1 and k_2 values.

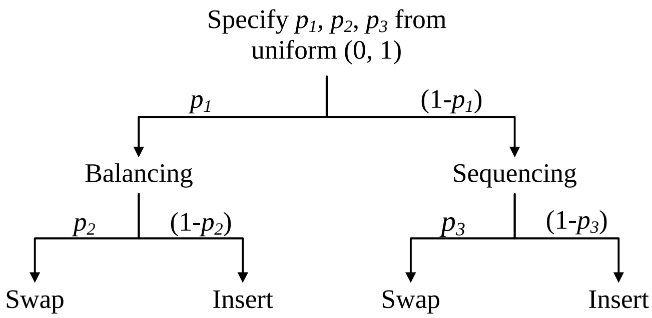

Figure 1. Neighbourhood generation logic of proposed SA

4. Experimental Results

The performance of the proposed SA-based algorithm may vary in terms of some problem factors and algorithm parameters. In this section, an experiment is conducted to evaluate the performance of the proposed algorithm considering different values of parameters and several problem factors. Some parameters of the proposed algorithm are fixed to: T_0 = 1000, T_{min} = 1.00, IT = 10, p_2 = p_3 = 0.5. The cooling rate (R) and the probability (p_1) that determine the new neighbour solution is a new line balance or a new model sequence are tested in terms of number of workstations (K) and *ADW*. Five levels of R (0.75, 0.80, 0.85, 0.90, and 0.95) and six levels of p_1 (0.5, 0.6, 0.7, 0.8, 0.9, and 1.0) are tested. Each problem is solved five times for each level of parameters. An analysis of variance (ANOVA) was performed for K and *ADW*, and the variabilities were partitioned under the main effects of length of model sequence (L), cooling rate (R), and neighbour probability (p_1). Results show that K is influenced by L, R, and p_1 main effects for almost all problem sizes. The ANOVA results for *ADW* indicate that it is also influenced by L, R, and p_1 main effects for almost all problem sizes. Some interactions were also found to be significant for K and *ADW*. There is a significant relation between the level of R values and quality of solutions. Both K and *ADW* decrease when R increases. The minimum values of K and *ADW* are obtained the R value of 0.95. It can easily be said that R values of 0.95 is appropriate for the proposed SA-based algorithm. The second parameter tested in the experiment, p_1 is also effective on the quality of solutions. For the p_1 values of 0.9, 1.0, 0.8, 0.7, there is no significant difference in terms of K. On the other hand, in terms of *ADW*, best solutions are obtained at the p_1 values of 1.0 and 0.9. Finally we can conclude that p_1 value of 0.9 can be used to increase the performance of the algorithm.

5. Conclusions

In today's market conditions, mixed-model production lines are used by many manufacturers to produce different products so that various demands of costumers can be met effectively. As a result of great attention to applications of JIT principles in manufacturing, suitability of U-lines to mixed-model production have been realized. Two important problems need to be solved for a successful mixed-model U-line implementation. These are the mixed-model U-line balancing and model sequencing problems. Both line balance and model sequence of the U-line can affect the performance of the line. Therefore, these problems should be handled simultaneously. We have to find such solutions to these problems that the number of workstations required on the U-line is minimized while the total workload of the line is distributed among workstations smoothly. In this paper, we proposed a simulated annealing-based heuristic approach for solving these problems simultaneously. The proposed approach always yields feasible solutions with respect to cycle time and precedence relationships. The neighbourhood generation mechanism of the algorithm enables us to consider both balancing and sequencing problems simultaneously.

References

Erel, E., Sabuncuoglu, I. and Aksu, B.A., 2001, Balancing of U-type assembly systems using simulated annealing. *International Journal of Production Research,* **39**, 3003-3015.

Hall, R., 1983, Zero Inventories (Homewood: American Production and Inventory Control Society/Dow Jones-Irwin).

KIM, Y.K., KIM, S.J. and KIM, J.Y., 2000, Balancing and sequencing mixed-model U-lines with a co-evolutionary algorithm. *Production Planning and Control*, **11**, 754-764.

Macaskıll, J. L. C., 1972, Production-line balances for mixed-model lines. *Management Science*, **19**, 423-434.

Mıltenburg, J., 1998, Balancing U-lines in a multiple U-line facility. *European Journal of Operational Research*, **109**, 1-23.

Mıltenburg, J., 2001, U-shaped production lines: a review of theory and practice. *International Journal of Production Economics*, **70**, 201-214.

Mıltenburg, J., 2002, Balancing and scheduling mixed-model U-shaped production lines. *International Journal of Flexible Manufacturing Systems,* **14**, 119-151.

Monden, Y., 1993, Toyota Production System. *Engineering and Management Press,* Georgia, USA.

Sparlıng, D. and Mıltenburg, J., 1998, The mixed-model U-line balancing problem. *International Journal of Production Research*, **36**, 485-501.