Design of a Footwear Assembly Line Using Simulation-based ALNS

Quang-Vinh Danga, Khoa Phama,*

aGlobal Production Engineering and Management, Vietnamese-German University, Binh Duong, Vietnam

* Corresponding author. Tel.: +84-650-222-0990; fax: +84-650-222-0980. E-mail address: pdk1188@yahoo.com

Abstract

Competitive advantages of footwear manufacturing companies today lie in the ability to improve the efficiency and effectiveness of resource utilization through constantly eliminating wastes. This paper presents an innovative approach to designing a footwear assembly line with uncertain task times and parallel workstations. The measure defined to characterize the performance of the assembly line is Pair/Person/Hour (PPH). Two operational parameters, namely buffer size and the number of resources are taken into account. The objective is to maximize the performance measure of the assembly line by determining an optimized setting of the operational parameters and optimal task assignment. The solution approach utilizes Discrete Event Simulation (DES) to model the performance measure under consideration of process constraints and variability. An Adaptive Large Neighborhood Search (ALNS) heuristic is next developed and integrated into the simulation model to search for problem domain. A case study of an industrial footwear manufacturing factory is conducted to demonstrate the effectiveness of the proposed approach.

1. Introduction

Vietnam is one of the top ten countries in producing and exporting footwear in the world. In 2012, Vietnam produced approximately 681 million pairs of shoes that were accounted for 3.2% of all produced pairs of shoes worldwide according to the General Statistics Office of Vietnam [1]. The increase of global competition forces footwear companies in Vietnam to continuously improve their manufacturing processes and to design more efficient production systems. Fundamentally, the important criteria in the footwear production are to ensure the on time delivery of customer orders and to increase machine and labor utilization to reduce production cost [2]. Designing of an efficient and effective footwear assembly line is, hence, a crucial task for competitiveness of the footwear companies.

In the footwear manufacturing, shoes are assembled with a number of given tasks. These tasks are prioritized based on their precedence relationship, i.e. a task cannot be processed unless all of its predecessors finish. Each task is performed on a specific machine, and the skill level of an operator working on that machine determines the processing time of the task. In other words, the processing time of a task varies according to each individual. Several tasks are grouped and assigned to a workstation depending on constraints of machines and labor skills [3]. A cycle time of each workstation is not allowed to exceed the required cycle time determined by a shop floor manager(s) to fulfill customer orders. Assume that there are six tasks in a footwear assembly line, Fig. 1 below depicts the precedence relationship between these tasks and the generated workstations.
In a footwear assembly line, the shop floor managers generally pay attention to how to maximize throughput of the line and minimize the number of employed resources, which also means to maximize the efficiency and minimize the cost of the line. These two conflicting objectives can be evaluated by one of the most important performance indicators that is Pair/Person/Hour (PPH), i.e. the number of pairs of shoes an operator can produce in an hour. The problem of designing of a footwear assembly line is how to group and assign a given set of tasks to a number of workstations so as to maximize the PPH while not violating a number of precedence constraints. The described problem is formulated as follows: Let \( X \) be the set of model control factors, i.e. number of workstations, number of operators, task assignment pattern as well as buffer capacity. Let \( X_i \) represent the set of model stochastic factors, i.e. processing time, availability of resources and defect rate. Let \( X_{li} \) represent the set of model logic controls, i.e. process routing, queue discipline and dispatching rule. \( X_i, X_{li} \) and \( X_{hi} \) have direct and indirect impacts on the system performance in terms of PPH. Therefore, \( f(X, X_i, X_{li}) \) can be defined as the nonlinear and stochastic objective function of the PPH. The functions \( h \), \( h(X_i, X_{li}, X_{hi}) \) represents the constraints on the model control, logic and parameters such as precedence constraints, limit on production resources and cycle time. The resulting problem formulation is presented mathematically as follows, with \( A \) are constants:

\[
\text{Max } f(X, X_i, X_{li}) : h(X_i, X_{li}, X_{hi}) \leq A \quad (1)
\]

Solving the problem defined in Eq. (1) implies determining the optimum settings of the set of control factors \( X \) so that the maximum PPH is attained under the set of control logic factors \( X, X_i \) without violating the set of constraints \( (A) \). In practice the shop floor managers often use their experience to address the problem, which is not an effective approach [4]. In addition, since there are a large amount of feasible line configurations, that empirical approach is not able to explore all possible design patterns. Furthermore, the performance of the assembly lines might be difficult to be maintained from one manager to another with different assignment preference and/or work experience [4]. Therefore, this paper focuses on developing an efficient and effective optimization approach to design a footwear assembly line.

The problem of designing of a footwear assembly line has been modeled in several respects comparable to assembly line balancing problems (ALBP). Several approaches for exact or heuristic algorithms have been proposed to address problems of this type. Ağpaka and Gökçen [5] develop an integer programming model to solve the simple straight type and U-type ALBP with stochastic task time. Liu et al. [6] develop a bidirectional heuristic to deal with the type-II ALBP under stochastic task time with the objective of minimizing the cycle time for a number of workstations and the pre-determined assembly reliability. Gamberini et al. [7] propose a new heuristic to solve the assembly re-balancing problem with stochastic task time. The idea is based on an approach named “Technique for order preference by similarity to ideal solution (TOPSIS)”. Zacharia and Nearchou [8] develop a multi-objective genetic algorithm to solve the type-II ALBP under fuzzy processing time. The total fuzzy fitness function is the weighted sum of the two fuzzy objectives of minimizing the cycle time and the smoothness index and minimizing the cycle time and the balance delay of the line. Zacharia and Nearchou [9] present a meta-heuristic based on the genetic algorithms to deal with the ALBP with fuzzy processing times. Ozcan et al. [10] formulate a chance constrained piecewise linear mixed integer programming model (CPMIP) to solve the two-sided ALBP under consideration of stochastic task time. The authors also develop a simulated annealing algorithm and a heuristic based on COMSOAL algorithm. Cakir et al. [11] propose a hybrid simulated annealing and tabu search algorithm to deal with the stochastic ALBP with parallel workstations. Boysen and Fleischer [12] describe a shortest path algorithm to solve both single model and several generalized ALBP. Hamta et al. [13] develop a hybrid particle swarm optimization algorithm with variable neighborhood search as a local search to deal with the single model ALBP under uncertain task time. Kim et al. [14] propose an endosymbiotic evolutionary algorithm (EEA) for the balancing and sequencing of the mixed-model U-type assembly line. Erel et al. [15] present a beam search method, a special type of tabu search, to deal with the stochastic U-type ALBP. Bagher et al. [16] develop imperialistic competitive algorithm to solve the simple U-type ALBP with stochastic task time. Xu and Xiao [17] propose a fuzzy simulation which is embedded into a genetic algorithm and produce a hybrid intelligent algorithm to deal with a mixed-model ALBP with stochastic task time and drifting operations. McMullen and Tarasewich [18] develop an ant colony optimization algorithm to solve the mixed-model ALBP under stochastic task time.

Although much related research and various developed methods have been completed, most of the studies in the ALBP consider throughput or some indirect measures of throughput as an objective function. The most general problem (type E ALBP) [19] strives to simultaneously minimize cycle time and number of workstations by using weighted factors. To the best of the authors’ knowledge, the problem of designing of a footwear assembly line with the objective of maximizing the PPH considering both throughput and number of resources has not been found in the literature. The surveyed approaches are not well-suited and cannot be directly used to solve this problem due to the simultaneous consideration of workload allocation, buffer storage space, a new performance indicator (PPH) and with the stochastic nature in the footwear manufacturing. Therefore, in this paper an approach based on simulation and Adaptive Large Neighborhood Search (ALNS) algorithm is developed to solve the problem. The aim is to use simulation to model and evaluate the performance measure of a footwear assembly line and to use ALNS algorithm to find the optimum setting of the operational parameters. One of the main advantages of the ALNS is that the large neighborhood permits the algorithm to navigate through a solution space easily even though the instance is tightly constrained. Another advantage is that other well-performing heuristics might be reused to form the core of the algorithm. Thanks to these, the resulting algorithm which will be discussed in the next section might become significantly efficient and effective in order to solve the problem of designing of a footwear assembly line.
2. Simulation-based ALNS algorithm

In this section an algorithm based on simulation and ALNS is presented to allow converting the described problem so that near-optimal solution could be found. The pseudo code for the simulation-based ALNS algorithm is illustrated in Fig. 2.

![Algorithm: Simulation-based ALNS](image)

The solution consists of three parts. Part 1 refers to the task assignment to workstations where the location of $X_o$ indicates task $i$ and the value of $X_o$ represents the workstation to which task $i$ is assigned with $i \in \{1, 2, \ldots, n\}$, $n$: number of tasks. Part 2 represents the allocation of number of operators where the location of $X_o$ indicates workstation $j$ and the value of $X_o$ indicates the number of operators assigned to workstation $j$ with $j \in \{1, 2, \ldots, p\}$, $p = \max\{X_o | i \in \{1, 2, \ldots, n\}\}$. Part 3 represents the buffer space allocation in which the location of $X_b$ indicates workstation $k$ and the value of $X_b$ indicates the buffer size at workstation $k$, with $k \in \{1, 2, \ldots, p\}$.

2.2. Initialization

A heuristic method is proposed to create an initial solution as follows:

**Step 1:** Create a list of unassigned tasks (list $A$), a two-dimensional matrix with immediate predecessors of all tasks (matrix $IP$), the current workstation $j$ and the current cycle time of the workstation.

**Step 2:** Check precedence feasibility: Check if there exist task in list $A$ that has no immediate predecessor $IP(i) \in \{0\}$. If so, go to Step 3; otherwise, go back to Step 1.

**Step 3:** Create a list of tasks with no immediate predecessor from list $A$ (list $B$).

**Step 4:** Check machine type feasibility: check if there exists a task in list $B$ that does not violate machine constraints. If so, go to Step 5; otherwise, go back to Step 3.

**Step 5:** Create a list of tasks without violating machine constraint (list $C$).

**Step 6:** Check cycle time feasibility: check if there exists a task in list $C$ that does not violate cycle time constraints. If so, go to Step 7; otherwise, go back to Step 5.

**Step 7:** Create a list of tasks without violating cycle time constraints (list $D$).

**Step 8:** Check if there exists any task in list $D$. If so, go to Step 10; otherwise, go to Step 9.

**Step 9:** Create new workstation if there is no task that can satisfy the cycle time of the current workstation.

**Step 10:** Assign task to workstation: select a task randomly from list $D$ and assign it to current workstation, then recalculate the cycle time of the current workstation and update list $A, B, C, D$ and matrix $IP$.

**Step 11:** Check if all of the tasks have been assigned to workstations. If so, go to Step 12; otherwise, go back to Step 3.
Step 12: Assign the number of operators and buffer sizes to all workstations randomly.

2.3. Destroy operators

Three destroy operators are used to remove tasks from a solution. These are random removal, sequential removal and related removal. The random removal randomly removes a task currently assigned to a workstation and moves it to the list of unassigned tasks (list A). This is repeated until a number of tasks have been removed. The sequential removal randomly selects a number of contiguous tasks in the middle of the solution and moves them to list A. The related removal removes tasks similar to each other, e.g. tasks that perform on the same type of machine or tasks that have the same priority.

2.4. Repair operators

There are three repair operators employed to reassign tasks to workstations, namely random insertion, longest processing time insertion and largest number of successors insertion. As stated in their names the first operator randomly selects a task, which is used to diversify the search domain while the second operator selects a task with the longest processing time and the third operator selects a task having the largest number of successors. Note that a selected task has to satisfy all of the constraints.

2.5. Acceptance criteria

A solution $s'$ is accepted if it is better than solution $s$. In case $s'$ is worse than $s$, $s'$ replaces $s$ with a probability of $e^{-\frac{1}{T(t)}},$ with $t$, the current temperature initialized at the beginning of the search [20].

2.6. Choosing destroy and repair heuristics

This paper employs destroy-repair pairs which mean every combination out of the set of destroy operators and the set of repair operators [6]. The probability of selecting an operator pair is proportional to $\rho_{dr}$ for each destroy-repair operator with $\rho_{dr}$: the weight of a destroy-repair pair. The formulation to compute $\phi_{dr}$ is as follows:

$$\phi_{dr} = \rho_{dr} \left( \sum_{dr} \sum_{dr'} \rho_{dr'} \right)$$

where $\phi_{dr}$: probability of selecting a given pair
$n_{dr}, n_{dr'}$: number of employed destroy-repair operators

A destroy-repair pair is selected in every iteration of the ALNS using the roulette selection [20]. In the beginning, the weights of all $\rho_{dr}$ are set to be 1, and the scores $\Psi_{dr}$ are set to be 0 with $\Psi_{dr}$: the score of the destroy and repair heuristic. At the end of each iteration, the scores $\Psi_{dr}$ of the employed pairs are updated in the following:

$\Psi_{dr} = \Psi_{dr} + \sigma_1$: if a destroy-repair pair creates a solution that improve the global best solution.

$\Psi_{dr} = \Psi_{dr} + \sigma_2$: if a destroy-repair pair creates a solution that is better than the current one.

$\Psi_{dr} = \Psi_{dr} + \sigma_3$: if a destroy-repair pair creates a solution that is worse than the current solution. The ALNS is divided into segments in which each consists of $m$ iterations. At the beginning of every new time segment, the scores are reset to be 0. At the end of every time segment, the weight is updated as follows:

$$\rho_{dr} = \lambda \rho_{dr} + (1 - \lambda) \Psi_{dr}$$

where $\lambda \in [0,1]$ is a parameter to control how sensitive the weight is to changes in the performance of the destroy and repair heuristic.

3. Case study

To examine the performance of the proposed simulation-based ALNS algorithm, a case study with real data taken from a footwear manufacturing factory in Vietnam is conducted in this section. The chosen area for the case study is the line P1 shown in Fig. 4 which produces men sport shoes. The line consists of 62 tasks, and data is collected from real studies on the shop floor of the footwear manufacturing factory and used as input for the case study.

The proposed algorithm has been programmed in SimTalk of Tecnomatix® Plant Simulation (TPS) software developed by Siemens and run on a PC that has AMD Turion II Dual-Core Processor 1.5 GHz and 4 GB RAM. Since the simulation model is an example of the non-terminating simulation, it is evaluated in two stages to consider the effect of warm-up. In the first stage, the model is run for 800 hours (5 months of working days) to find the warm-up period. Then to calculate the average PPH of the system for a working day, the model is run 10 times in which each lasts for 8 hours of the simulated time including the warm-up period. The PPH is estimated with a relative error of 0.05 and a confidence level of 95%. The parameters of the ALNS $\sigma_1, \sigma_2, \sigma_3, \lambda$ and $m$ are set to be 33, 9, 13, 0.7 and 200, respectively.

The best solution of the case study is summarized in Table 1 and depicted in Fig. 5. The result generally shows that 62 tasks are grouped and assigned to 41 workstations, and one operator is able to produce 119 pairs of shoes per hour. Table 1 gives the number of operators and buffer sizes allocated to each workstation, e.g. at workstation 5 there are two operators and the buffer size is 2. As compared to the manual planning and designing, the PPH achieved by the proposed algorithm shows a remarkable increase of approximately 16% under the same investigated case. The total computation time to solve the case study with 62 tasks is approximately 420 seconds (or 7 minutes). This amount of time is acceptable considering the medium- to long-term nature of the solution as well as the importance of the decision to be made.
Fig. 4. Precedence diagram of the case study.

Table 1. Result of the case study.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Tasks assigned</th>
<th>Number of operators</th>
<th>Buffer size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1; 3B-1; 8A-1; 14A3A-1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3A-2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3A-1; 6A-1; 13A-1; 14A-1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

PPH = 119
This paper presents the results of a study of designing of a footwear assembly line under stochastic task time and parallel operations. It is important for shop floor managers to make a design of the assembly line which maximizes the PPH, one of the key performance indicators considering two important aspects of throughput and the number of employed resources. It is important for shop floor managers to make a design of the assembly line which maximizes the PPH, one of the key performance indicators considering two important aspects of throughput and the number of employed resources. This must be done while taking into account a number of designing parameters such as machine reliability and/or product rework.

4. Conclusions

This paper presents the results of a study of designing of a footwear assembly line under stochastic task time and parallel operations. It is important for shop floor managers to make a design of the assembly line which maximizes the PPH, one of the key performance indicators considering two important aspects of throughput and the number of employed resources. This must be done while taking into account a number of designing parameters such as machine reliability and/or product rework.

References

[17] Xu W, Xiao T. Mixed model assembly line balancing problem with fuzzy operations times and shifting operations. Proceedings of the 46th Winter Simulation Conference; 2008 Dec 7-10; Miami FL, US.