Hybrid Multiobjective Evolutionary Algorithm for Assembly Line Balancing Problem with Stochastic Processing Time

Wenqiang Zhang\textsuperscript{a}, Weitao Xu\textsuperscript{a} and Mitsuo Gen\textsuperscript{b,c,}\textsuperscript{*}

\textsuperscript{a}College of Information Science and Engineering, Henan University of Technology, China
\textsuperscript{b,Fuzzy Logic Systems Institute and \textsuperscript{c}Tokyo University of Science, Japan

Abstract

An assembly line (AL) is a typical manufacturing process consisting of various tasks in which interchangeable parts are added to a product in a sequential manner at a station to produce a final product. Most of the work related to the ALs concentrate on the assembly line balancing (ALB) which deals with the allocation of the tasks among stations so that the precedence relations among them are not violated and a given objective function is optimized. From the view point of the real ALB systems, multiobjective ALB with stochastic processing time (S-moALB) is an important and practical topic from traditional ALB problem involving conflicting criteria such as the cycle time, variation of workload, and/or the processing cost under uncertain manufacturing environment.

This paper proposes a hybrid multiobjective evolutionary algorithm (hMOEA) to deal with such S-moALB problem with stochastic processing time considering minimization of the cycle time and the processing cost, given a fixed number of stations available. The special fitness function strategy is adopted and a hybrid selection mechanism is designed to improve the convergence and distribution performance. Experimental results with various instances show that hMOEA could get the better convergence distribution performance than existing MOEAs.

Keywords: Assembly line balancing (ALB), Stochastic processing time, Multiobjective optimization problem, Multiobjective evolutionary algorithm (MOEA);

1. Introduction

The assembly line balancing (ALB) problem optimizes one or more objectives by determining the assignment of various tasks to an ordered sequence of stations without violating restrictions imposed on the line in a manufacturing system. The traditional ALB problem processes one model with tasks by allocating the tasks into stations to generate an optimal solution by considering the deterministic task time and/or cost. However, in the modern industries,
The problem of balancing of assembly lines has been extensively examined in the literature and a number of review studies have been published [1-9]. Both exact and heuristic procedures have been developed to solve this problem. As for stochastic processing time, single objective optimization [10-15] and multiobjective optimization [16-19] have been investigated extensively. Multiobjective optimization problem (MOOP) is a practical, important but very intractable optimization problem in which two or more conflicting objectives should be considered together, and many Pareto-optimal solutions with incommensurable quality are generated for decision makers. Multiobjective evolutionary algorithms (MOEAs) have been recognized to be well-suited for solving MOOPs [5,26]. As two classical MOEAs, non-dominated sorting genetic algorithm-II (NSGA-II) [20] and strength Pareto evolutionary algorithm 2 (SPEA2) [21] have been proven to be able to get better quality in solving MOOPs. NSGA-II can get better quality owing to its Pareto ranking and crowding distance mechanism. SPEA2 depends on raw fitness assignment mechanism and density mechanism. Zhang et al proposed an improved vector evaluated genetic algorithm with archive (IVEGA-A) that combined VEGA and Pareto-based scale-independent fitness function (GPSI-FF)-based archive mechanism for solving the process planning and scheduling problem [22,27]. Furthermore, under the consideration of the multiobjective characteristic of S-moALB with simultaneously optimizing the two conflicting objectives, the cycle time and processing cost, seeking an optimal solution rapidly and effectively from all of the permutations, combinations of all of the tasks, manufacturing resources is very difficult. As for the MOEA, the special mechanisms need to be designed to increase the quality (both of convergence and distribution) so that the heuristic algorithm can be applied easily on the real world ALB problem. In this study, we consider the S-moALB and propose a new hybrid multiobjective evolutionary algorithm (hMOEA). The hMOEA could converge to the center and two edges areas of Pareto front strongly and could both preserve the convergence rate and guarantee the better distribution performance.

The paper is organized as follows: Section 2 formulates the mathematical model for S-moALB problem; Section 3 presents the detailed hMOEA approach; Section 4 gives a discussion and analysis of numerical experiments results; finally, the conclusion and future work are given in Section 5.

2. Mathematical Formulation

Difference to the conventional ALB, the duration of processing time may take any real value from the uniform distribution $U(\xi, \bar{t}_j)$ where $\xi$ and $\bar{t}_j$ are the given lower and upper bounds respectively. Therefore, the duration uncertainty can be constructed through the concept scenario $\xi$ which corresponds to an assignment of reasonable duration on task $j$. This finite set of scenarios $\Xi$ is sampled as input of S-moALB which the processing time satisfies the intervals $[\xi, \bar{t}_j]$.

The S-moALB problem concerns with the assignment of the tasks(each task $j=1,2,\ldots,n$ with processing time $t^j$ and processing cost $d^j$ under the scenario $\xi$ ) to stations (each station $i=1,2,\ldots,m$) for each station in order to minimize the cycle time and minimize the processing cost under the constraint of precedence relationships. The Suc($j$) is the set of direct successors of task $j$ and Pre($j$) is the set of direct predecessors of task $j$. $S^i$ is the set of tasks assigned to station $i$ under the scenario $\xi$. $i(S^i) = \sum_{j=1}^{n} t^j x_{ij}$ $\forall i$ is the processing time at station $i$ under the scenario $\xi$. The decision variable $x_{ij}=1$ means if the task $j$ is assigned to station $i$, otherwise, $x_{ij}=0$. Therefore, the mathematical model of S-moALB can be summarized as follows:

$$\min \quad c^i = \max \left\{ \sum_{j=1}^{n} t^j x_{ij} : \sum_{i=1}^{m} d^j \right\}$$  \quad (1)

$$\min \quad d^j = \sum_{i=1}^{m} x_{ij}$$  \quad (2)

$$\text{s.t. } \sum_{i=1}^{m} x_{ij} \geq 1, \forall k \in \text{Pre}(j), \forall j$$  \quad (3)

$$\sum_{i=1}^{m} x_{ij} = 1, \forall i, j$$  \quad (4)

$$x_{ij} \in \{0,1\}, \forall i, j$$  \quad (5)
The first objective function (1) of the model is to minimize the cycle time of the assembly line. The second objective function (2) is to minimize the total processing cost. Inequity (3) states that all predecessor of task \( j \) must be assign to a station, which is in front of or the same as the station that task \( j \) is assigned in. Equation (4) ensures that task \( j \) must be assigned to only one station. Equation (5) represents the nonnegative restrictions.

3. Hybrid Multiobjective Evolutionary Algorithm

The main framework of hMOEA is shown in Fig. 1 where \( A(t) \) represents the archive at generation \( t \) and \( P(t) \) is the population at generation \( t \).

In the main framework, the fitness function plays the key issue to evaluate the individuals and update the archive according to this fitness function. In this paper, a Pareto dominating and dominated relationship-based fitness function (PDDR-FF) -based fitness function is proposed to evaluate the individuals. The PDDR-FF of an individual \( s_i \) is calculated by the following function [27]:

\[
\text{eval}(s_i) = q(s_i) + 1/(p(s_i) + 1), \quad i = 1, 2, ..., \text{popSize}
\]  

where \( p(s_i) \) is the number of individuals, that can be dominated by the individual \( s_i \), \( q(s_i) \) that can dominate the individual \( s_i \). The smaller value is better. If the individual belongs to nondominated one, its fitness value will not exceed one. The fitness value of individual which is dominated by other will exceed one. It is obvious that the nondominated individuals locating around the central area of Pareto font with bigger domination area will have smaller values (near to 0) than the edge points (near to 1). Therefore, PDDR-FF gives the sensible difference values between the nondominated and dominated individuals. Moreover, ones with different numbers of dominating are also set as different fitness values even though they are all nondominated individuals. The individuals locating around the central region of Pareto font will have smaller values than the edge points.

In selection phase of hMOEA, the PDDR-FF based sampling strategy has the advantage with the tendency converging toward the canter area of the Pareto font, but drawback to the edge region. It causes bad distribution performance. The VEGA prefers the edge rather than certer regions of Pareto font that it causes VEGA cannot achieve better distribution performance. So we hybridize these two mechanisms to improve the overall performance and reduce the computation time.

The solution procedure of one generation includes 4 phases.
Phase 1: Sampling the individuals into the mating pool for the edge points by VEGA

Selecting individuals into subpopulations 1 and 2 by VEGA (good for the edges area of Pareto front). In this step, individuals are selected with replacement according to objective 1 into subpopulation 1 while ignoring objective 2 until the size of the subpopulation 1 (half of population size) is reached. In the same manner, individuals are selected for objective 2 into subpopulation 2 without considering objective 1 until subpopulation 2 is full.

Phase 2: Combining the individuals of archive into the mating pool

Combining the subpopulations and archive \(A(t)\) to form the mating pool (good for the central area of Pareto front).

Phase 3: Reproducing the new population

Arithmetical crossover (for task priority vector) and swap mutation are used to reproduce new individuals.

Phase 4: Updating the Archive by the fitness function

The individuals of \(A(t)\) and \(P(t+1)\) are combined to form a temporary archive \(A'(t)\). Thereafter, the PDDR-FF values of all individuals in \(A'(t)\) are calculated and sorted. The smallest \(|A(t)|\) individuals in \(A'(t)\) are copied to form \(A(t+1)\). This archive updating mechanism likes an elitist sampling strategy to keep the better individuals with better PDDR-FF values.

After generating the subpopulations in mating pool, the all individuals in archive \(A(t)\) are also as parts of mating pool. As shown in Fig. 1, the subpop-1 stores the good individuals for one objective, and subpopulation 2 holds the good individuals for the other objective. The archive saves the individuals with good PDDR-FF values. Therefore, in the mating pool, one-third of the individuals serve one objective, one-third the other objective, and the left one-third both the two objectives. The archive mechanism tries to cover the selection bias of VEGA. These three parts of the mating pool make the solutions converge to the Pareto front evenly. Therefore, the strong convergence capability of VEGA and PDDR-FF ensures that the hMOEA has the ability to converge to the true Pareto front both in central and edge areas.

As for the genetic representation, a real number based encoding and decoding routine is adopted to describe the chromosome. Randomly generate a real number for each task as task priority and to obtain the task priority vector, then decode the task priority vector as the task sequence vector according to task priority and the precedence relationship constraints. After generating the task sequence, a bisection searching method could be used to divide (group) the task sequence into several parts (a breakpoint vector could be generated) that mean the grouped task could be processed on this station.

The genetic operators include the crossover, mutation and selection is designed for adapting the above three vectors. The arithmetical crossover and the swap mutation are used the task priority vector and binary tournament selection operator is used for selecting the individuals.

4. Experiments and Discussion

We employed Gunther’s problem data with 35 tasks and 6 stations, which is widely used in the ALB problem literature \([23]\). For the duration uncertainty, the expected duration \(t^E_j\) is to be equal to duration of that task in original instances. The upper and lower processing time bounds of a task affected by uniform variation in its duration is calculated by:

\[
[L_j, T_j] = t^E_j \times [1, 1 + \gamma] \tag{7}
\]

In this paper a linear function is considered between the cost and the processing time. Therefore, the problem is to make a balance between the costs and the processing time as two incomparable objectives \([19]\).

The processing cost of each task is calculated by:

\[
d_j = \max_{l_{ij} \in A} \{t^E_j + (1 - t^E_j)\} \tag{8}
\]

If 30 percent of operations is under uncertainty, the disturbance ratio will be assigned to 0.3. For measuring the uncertainty, a set of 150 scenarios \(\Xi\) is sampled as the input of S-moALB

All the simulation are performed on Pentium Dual-Core processor (2.70 GHz clock) and 2GB memory. The adopted parameters are listed as follows: population size, 200; maximum generation, 1000; archive size, 100; crossover probability, 0.80 and 0.3; mutation probability, 0.40 and 0.1. hMOEA, NSGA-II, and SPEA2 are run 30 times to compared the results with each other. It should be noted that the parameters of all 3 methods are the same, except for the size of archive. The archive sizes of hMOEA is set to be half the population size, 100, while of NSGA-II and SPEA2 are set to be the same as the population size, 200.
Let $S_j$ be a solution set for each method. $PF^*$ is a known reference Pareto solutions. In this study, $PF^*$ in this study comes from combining all of the obtained Pareto set (combining all the scenarios) with 30 runs by 3 methods. The following two performance measures are considered.

![Fig. 2 Coverage C and Spacing SP by three methods](image)

Coverage $C(S_1, S_2)$ is the percent of the individuals in $S_2$ which are weakly dominated by $S_1$ [24]. The larger $C(S_1, S_2)$ means that $S_1$ outperforms $S_2$ in convergence.

Spacing $SP(S_j)$ is the standard deviation of the closest distances of individuals by $S_j$[25]. Smaller $SP$ means better distribution performance.

The $C$ is used to verify convergence performance while $SP$ is used to check the distribution performance.

The Fig. 2 show the numerical comparison of the box-and-whisker plots for $C$ and $SP$ by 3 methods. The Fig. 2a and Fig. 2b demonstrate that the hMOEA is better than other 2 methods on $C$ measure. The distribution performance $SP$ indicates that hMOEA is slightly better than NSGA-II and SPEA2 methods (Fig. 2c). In general, the convergence and distribution performance of hMOEA is better than famous NSGA-II and SPEA2.

5. Conclusions

In this study, a hybrid multiobjective evolutionary algorithm (hMOEA) approach proposed to solve multiobjective assembly line balancing with stochastic processing time (S-moALB). This approach mainly used the selection mechanism of vector evaluated genetic algorithm (VEGA) and the Pareto-based scale-independent fitness function (GPSI-FF)-based archive mechanism. These two mechanisms could converge to the multi-area of Pareto front strongly and obtained even distribution performance without special crowding distance mechanism. Meanwhile, the hMOEA could reduce the CPU time rather than traditional MOEA. Numerical comparisons demonstrated that hMOEA could achieve better performance in efficacy.

Acknowledgments

This research work is partly supported by the National NSF of China (No. U1304609), the Education Dept. of
References