TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 13/16 pp308-314
Volume 16, Number 3, June 2011

Strategic Robust Mixed Model Assembly Line Balancing Based on Scenario Planning*

XU Weida (徐炜达), XIAO Tianyuan (肖田元)**

National Laboratory for Information Science and Technology, National Computer Integrated Manufacturing Systems Engineering Research Center, Department of Automation, Tsinghua University, Beijing 100084, China

Abstract: Assembly line balancing involves assigning a series of task elements to uniform sequential stations with certain restrictions. Decision makers often discover that a task assignment which is optimal with respect to a deterministic or stochastic/fuzzy model yields quite poor performance in reality. In real environments, assembly line balancing robustness is a more appropriate decision selection guide. A robust model based on the α worst case scenario is developed to compensate for the drawbacks of traditional robust criteria. A robust genetic algorithm is used to solve the problem. Comprehensive computational experiments to study the effect of the solution procedure show that the model generates more flexible robust solutions. Careful tuning the value of α allows the decision maker to balance robustness and conservativeness of assembly line task element assignments.

Key words: mixed model; assembly line balancing; robust; scenario planning; genetic algorithm

Introduction

The assembly line balancing problem (ALBP) is the problem of assigning basic assembly task elements to different stations, pursuing specific goals in compliance with given constraints. Ever since Henry Ford's introduction of the assembly line, the ALBP has been of significant industrial importance. Due to the high cost of building and maintaining an assembly line, manufacturers often simultaneously produce one model with different features or several models on a single line. Under these circumstances, the mixed model assembly line balancing problem (MALBP) arises to smooth the production and reduce the cost.

According to Scholl[1], the MALBP can be distinguished into four versions with regard to the objective function. This paper considers MALBP-F which involves finding a feasible balance for a given number of stations and a given cycle time. This situation is very common in real assembly lines where the cycle time is typically chosen to provide the desired annual output rate and the number of stations is dictated or constrained by the existing physical infrastructure (such as facilities, conveyors, and workshops space etc.).

Assembly line balancing is inherently strategic. Planners confronted with this kind of problem often long for an assignment that will have a permanent, productive life time. While the conditions for which an assembly line will operate may be known or estimated with some degree of certainty for a very short term, the long-term operating conditions are subject to considerable uncertainties, with task processing times exhibiting noticeable variations from cycle to cycle and demand for each product model changing from day to day. As a result, more and more researchers have stated

Received: 2010-04-13; revised: 2011-03-28
∗ Supported by the National High-Tech Research Development (863) Program of China (No. 2006AA04Z160)
∗∗ To whom correspondence should be addressed.
E-mail: xty-dau@mail.tsinghua.edu.cn; Tel: 86-10-51533733

the ALBP with uncertain task times with most studies addressing the problem with stochastic task times[2,3]; while others use fuzzy task times[4,5]. Boysen et al.[6] gave a comprehensive survey of this field.

Ample evidence exists in the research literature that for decision environments with significant uncertainty, neither the deterministic optimization nor the stochasitc/fuzzy optimization approaches accurately represent the aim of the decision maker[7]. The uncertainty can be structured through scenario planning where each scenario corresponds to the assignment of plausible values to the model parameters. Robust analysis looks for solution in a context where it is impossible to attribute probabilities or possibilities to the outcomes of any decision. Unlike deterministic or stochastic/fuzzy approaches which are aimed at determining the best solution for a certain instance of values (or scenario), robust approaches try to find a solution or a set of solutions that performs well across all scenarios with hedges against the worst possible scenarios.

In combinatorial optimization problems like the MALBP, the most widely used robust criteria rely on the worst case, such as min-max, min-max regret, and min-max relative regret. These criteria were referred to as absolute robust criteria, robust deviation criteria, and relative robust criteria by Kouvelis and Yu[7]. However, use of these criteria often results in conservative decisions, because they are based on the anticipation that the worst case will happen. Further more, no tolerance is considered in these criteria. These two drawbacks of the criteria resting on the worst case suggest considering alternative robust criteria. Kouvelis et al.[8] introduced a measure called p-robustness. Daskin et al.[9] proposed a model called the α-reliable min-max regret model. Kalai et al.[10] defined a α-leximax relation and proposed an approach called lexicographic α-robustness.

Generally, existing robust models can be distinguished into two families. The first family looks for solutions which optimize the objective function for the worst case scenario (e.g., min-max related approaches), whereas the second one imposes conditions that solutions must satisfy to be considered as robust (e.g., p-robustness, α-reliable min-max regret model, and lexicographic α-robustness). This paper presents a lexicographic-order based robust approach which is similar to the first family of approaches where the objective function to be optimized is the α worst case scenario rather than the worst case scenario.

1 Robust MALBP

1.1 Problem statement

First outline some assumptions that apply for most practical mixed model assembly lines.

(1) The line is connected by a conveyor belt which moves at a constant speed. Consecutive workpieces are equispaced on the line by launching each after a cycle time.

(2) The cycle time, number of stations and models to be assembled within the decision horizon and their own precedence graphs are predetermined.

(3) The operating time of every task and the demand for each model are not exactly known, and no probabilities or possibilities can be attached to them. However, a set of realizable scenarios can be given for these uncertainties by the decision makers or domain experts.

(4) Precedence graphs for all models can be accumulated into a single combined precedence graph, with similar operations for different models having different operating times; zero operating time indicates that an operation is not required for a model.

(5) Every task should be assigned to exactly one station, and common tasks between models must be assigned to the same stations.

The notations used in this paper are as follows:

C, cycle time;

K, number of stations; index: $k=1,2,\cdots,K$;

M, number of models; index: $m=1,2,\cdots,M$;

J, number of tasks; index: $j=1,2,\cdots,J$;

P, combined precedence relation matrices corresponding to the combined precedence graph. $p_{ij}=1$ means task i should be finished before task j starts;

S, number of potentially realizable input data scenarios over a pre-specified planning horizon; index $s=1,2,\cdots,S$;

t_{mj}^{s}, operating time for task j for one unit of model m for scenario s;

T_{mk}^{s}, workload (i.e., station time, total task time) for model m on station k for scenario s;

q_{m}^{s}, demand for model m for scenario s;

q^{s}, total demand for all models for scenario s;

TM_{k}^{s}, average workload for all models on station k

$(=\sum_{m=1}^{M} T_{mk}^{s} \frac{q_m^s}{q^s})$ for scenario s;

X, vector of decision variables, i.e., task assignments. $X=(x_1, x_2, \cdots, x_J)$, x_j means task j is assigned to the station with index x_j;

X_s^*, optimal decision for scenario s.

The MALBP-F has basically two objectives for smoothing the station workload as horizontal balancing (which means smoothing varying station times caused by different models) and vertical balancing (which means smoothing station times over all stations of the line). Emde et al.[11] presented a series of objective functions for workload smoothing with numerical evaluations. In this paper, according to the requirement of the real projects, we choose vertical balancing. One of the measurements for vertical balancing called workload variance, $WV_s(X)$, for a given decision X for scenario, s, is straightforward and is thus employed in this study,

$$WV_s(X) = \sqrt{\sum_{k=1}^{K}\left(TM_k^s - \sum_{i=1}^{K}\frac{TM_i^s}{K}\right)^2} \tag{1}$$

Maximizing the vertical workload smoothness is equivalent to minimizing workload variance. This methodology can easily be applied to other objective functions.

1.2 *α* worst case scenario-based robust MALBP model

Since the reasoning and results are valid for workload variances and for regrets and relative regrets, the term "observation" is used here with the notation f for all three. For the original workload variance,

$$f_s(X) = WV_s(X) \tag{2}$$

for the workload variance regret,

$$f_s(X) = WV_s(X) - WV_s(X_s^*) \tag{3}$$

and for the workload variance relative regret,

$$f_s(X) = \frac{WV_s(X) - WV_s(X_s^*)}{WV_s(X_s^*)} \tag{4}$$

Definition 1 The α worst case scenario is the scenario for which the system performs equally or better than for $\alpha\times100\%$ of all scenarios.

Here α $(0<\alpha\leqslant1)$ is a tolerance threshold that is specified by the decision maker (for $\alpha=1$, the α worst case scenario is none other than the worst case scenario).

For a given solution X, rearrange the observations in non-decreasing order to obtain $f_{[1]} \leqslant f_{[2]} \leqslant \cdots \leqslant f_{[S]}$. Then, $f_{[\lfloor \alpha\times S \rfloor+1]}$ is the observation for the α worst case scenario. Here, $\lfloor \alpha\times S \rfloor$ means the largest integer less than $\alpha\times S$.

With Definition 1, the α worst case scenario-based robust MALBP model can be formulated as follows.

$$\text{Minimize} \quad y \tag{5}$$

$$\text{s.t.} \quad y = f_{[\lfloor \alpha\times S \rfloor+1]} \tag{6}$$

$$\forall p_{ij}=1,\ x_i \leqslant x_j \tag{7}$$

$$1 \leqslant x_i, x_j \leqslant K,\ x_i, x_j \text{ are integers} \tag{8}$$

$$1 \leqslant i, j \leqslant J,\ \ i, j \text{ are integers} \tag{9}$$

The objective function (5) minimizes the objective value y, which is defined in constraint (6). Constraint set (7) ensures that the precedence relations are fulfilled. Constraint sets (8) and (9) guarantee that every task is assigned to exactly one station. The criteria in the model (5)-(9) are called α-min-max, α-min-max regret, and α-min-max relative regret corresponding to min-max, min-max regret, and min-max relative regret in the traditional robust model.

2 Robust Genetic Algorithm

The mathematical complexity of the ALBP for a given scenario is NP-complete in the strong sense because the NP-complete bin-packing problem can be easily transformed to this in polynomial time[4]. The model (5)-(9) is even more complicated due to the consideration of decision robustness.

Many algorithms have been developed to solve the ALBP. One well-known method is Jackson's algorithm, which enumerates possible combinations of operations based on their precedence relationships. Another method is the Branch and Bound method which cuts combinations having performance index values lower than some bound value. Other methods include integer programming and dynamic programming. However the ALBP is difficult to solve by these traditional methods for problems involving large numbers of tasks and stations, so these methods are mainly applied to get the minimum number of stations for a given cycle time. Heuristic algorithms have also been applied to the ALBP. They are very practical; however, they can not guarantee the optimal solution. In recent years, numerous research efforts have been directed towards the development of intelligent algorithms to provide an alternative to traditional optimization techniques, such as neural networks, simulated annealing (SA), and genetic algorithms (GA). Hashimoto et al.[12] applied the Hopfield neural network to the ALBP, and showed that moderate scale ALBP can be solved efficiently. McMullen and Frazier[13] developed simulated annealing algorithms to solve the MALBP. Scholl and Becker[14], Becker and Scholl[15], and Boysen et al.[6] surveyed various ALBP solution procedures.

The GA is a stochastic search method for optimization problems based on the mechanics of natural selection and natural genetics (i.e., survival of the fittest). When the objective functions to be optimized are multimodal or the search spaces are particularly irregular, the algorithms need to be very robust to avoid getting stuck in a local optimal solution. The advantage of the GA is that it can easily obtain the global optimal solution. During the past three decades, the GA has demonstrated considerable success in providing good solutions to many combinational optimization problems, such as the travelling salesman problems, flow-shop and job-shop scheduling problems, and so on. The GA has also been applied to the MALBP[16-18]. The major difference between all these studies and the present work is the consideration of decision robustness.

A GA integrating traditional robust criteria has already been used to solve the robust mixed model assembly line balancing problem[19]. Here, this algorithm will be applied to the new robust model (5)-(9) which is based on the α worst case scenario. To be self-contained and to avoid unnecessary repetition, we only briefly outline the main steps with details available from Xu and Xiao[19].

Step 1 Initialize chromosomes.

Step 2 Calculate the objective values for all chromosomes.

Step 3 Compute the fitness of each chromosome by rank-based evaluation function based on their objective values.

Step 4 Select the chromosomes by spinning the roulette wheel.

Step 5 Update the chromosomes by crossover and mutation operations.

Step 6 Repeat Steps 2 to 5 a given number of times or until a suitable solution is found.

For the α-min-max criterion, the objective value is equal to the workload variance for the α worst case scenario; while for the α-min-max (relative) regret criterion, it is equal to the (relative) regret for the α worst case scenario. In addition, for the last two criteria, the optimal solution for each scenario is needed in advance, which can be obtained using this GA or other approaches mentioned in the beginning of this section.

3 Computational Experiments

3.1 Experiments on test problems

The desirability of this methodology is assessed using five combined precedence relation metrics from the standard ALBP lib[20]: the 21-task problem MITCHELL(21), the 35-task problem GUNTHER(35), the 45-task problem KILBRID(45), the 58-task problem WARNECKE(58), and the 83-task problem ARC(83).

Each problem assumes that 50 scenarios are outlined by decision makers in advance. Each scenario is composed of the demand for every model and the processing times for every task. Four different values of α are used (i.e., 0.5, 0.7, 0.9, 1.0), and the corresponding expectation model (viz., all scenarios have the same probability to appear) is also used to compare the long-term performance of solutions from the model. Due to space considerations, this section only illustrates results for the α-min-max criterion. The other two criteria (i.e., α-min-max regret and α-min-max relative regret) generate similar results.

Box plots[21] are used to graphically compare the work variances in the 50 scenarios with the best solution found for each approach. Box plots are used as a convenient way of depicting groups of numerical data through their five-number summaries: the smallest observation, lower quartile, median, upper quartile, and largest observation. These will indicate which observations, if any, might be outliers.

The results are shown in Figs. 1-5. In these figures, 0.5, 0.7, 0.9, and 1.0 denote the robust model results for different values of α, and EXP denotes the results for the expectation model. The results show that the α worst case scenario based robust model can hedge against the risk of poor system performance in bad scenarios. α=1.0, which is equivalent to the original min-max robust criterion, always obtains the best worst case scenario performance, but sometimes it may be too conservative. Relatively optimistic solutions are found for smaller α.

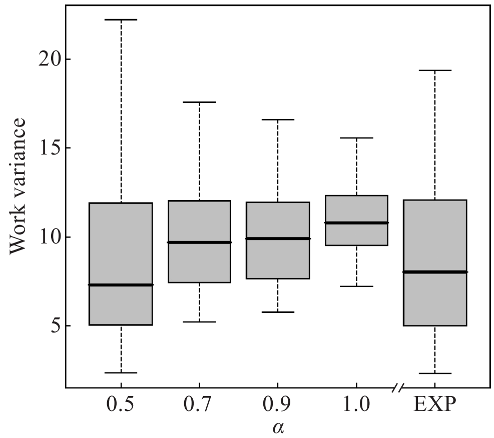

Fig. 1 Box plot for MITCHELL (21)

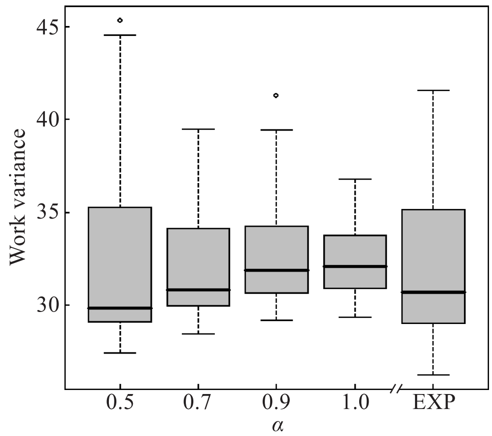

Fig. 2 Box plot for GUNTHER (35)

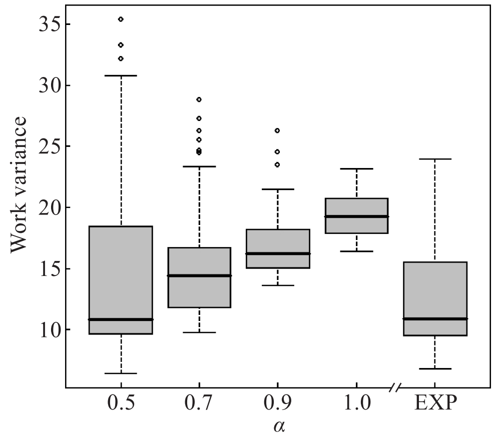

Fig. 3 Box plot for KILBRID (45)

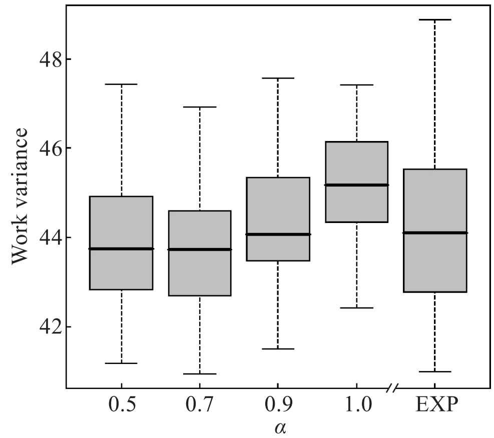

Fig. 4 Box plot for WARNECKE (58)

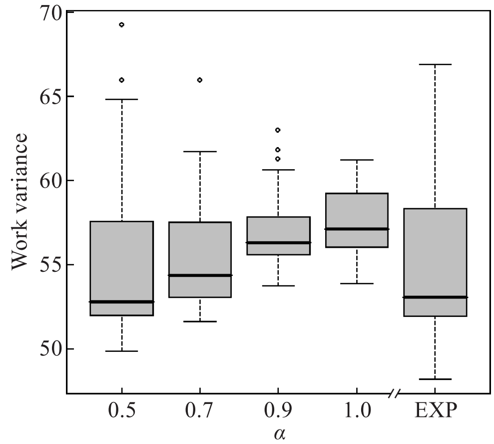

Fig. 5 Box plot for ARC (83)

With respect to long-term (expected) performance, robust solutions always generate less fluctuant system performance with an insignificant sacrifice in its value. As α increases from 0.5 to 1, the fluctuations of the workload variance between each scenario become smaller and smaller. Table 1 lists the means and standard deviations of the work variances for each problem to detail these findings.

3.2 Experiments on real problems

This section applies the approach to a real assembly line of an automobile producer in Beijing. There are 28

Table 1 Means (and standard deviations) of work variances for each test problem

α	Mean (SD)				
	MITCHELL (21)	GUNTHER (35)	KILBRID (45)	WARNECKE (58)	ARC (83)
0.5	9.07 (4.89)	32.62 (5.08)	14.97 (7.77)	43.92 (1.53)	55.12 (4.79)
0.7	10.00 (3.04)	32.13 (3.16)	15.47 (5.11)	43.68 (1.44)	56.31 (4.47)
0.9	10.16 (2.86)	32.59 (2.56)	16.89 (2.72)	44.29 (1.28)	56.88 (2.08)
1.0	10.47 (2.45)	32.40 (1.94)	19.41 (1.80)	45.23 (1.23)	57.99 (1.57)
EXP	8.95 (4.55)	32.11 (4.11)	12.68 (4.07)	43.16 (1.79)	54.66 (4.00)

stations in the line and the assembly tasks are divided into 97 elements by the engineers. The processing times of each task element and the number of cars of each model to be produced are uncertain.

Like for the tests in Section 3.1, four different values of α (i.e., 0.5, 0.7, 0.9, 1.0) with the corresponding expectation model (i.e., EXP) also considered to compare long-term performance. The resulting box plot shown in Fig. 6 indicates similar conclusions as for the tests in the previous section.

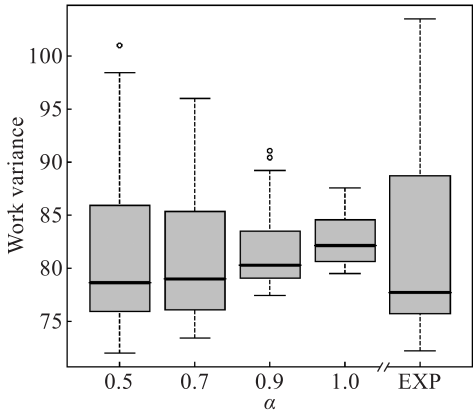

Fig. 6 Box plot for the real automobile assembly line problem

4 Conclusions

This paper considers the robust mixed model assembly line balancing problem with uncertain task processing times and demands. The scenario planning technique is employed to describe these uncertainties. A robust model based on the α worst case scenario is used instead of focusing on the worst case scenario to compensate for the conservativeness of traditional robust approaches. A robust GA is then designed to solve this model.

Tests show that as α approaches 1, the model generates more and more robust solutions and degenerates to the traditional model when α=1 with the most robust solutions. For smaller α, the model generates less robust but more optimistic solutions. Thus, careful tuning of α enables decision makers to balance the robustness and conservativeness of the assembly line task elements assignments.

References

[1] Scholl A. Balancing and Sequencing of Assembly Lines. 2nd edn. Heidelberg, Germany: Physica-Verlag, 1999.

[2] McMullen P R, Tarasewich P. Using ant techniques to solve the assembly line balancing problem. *IIE Transactions*, 2003, **35**(7): 605-607.

[3] Zhao X, Liu J, Ohn K, et al. Modeling and analysis of a mixed-model assembly line with stochastic operation times. *Naval Research Logistics*, 2007, **54**(6): 681-691.

[4] Hop N V. A heuristic solution for fuzzy mixed-model line balancing problem. *European Journal of Operational Research*, 2006, **168**(3): 798-810.

[5] Xu W, Xiao T. Mixed model assembly line balancing problem with fuzzy operation times and drifting operations. In: Proceeding of Winter Simulation Conference (WSC 2008). Miami, FL, USA, 2008: 1752-1760.

[6] Boysen N, Fliedner M, Scholl A. A classification of assembly line balancing problems. *European Journal of Operational Research*, 2007, **183**(2): 674-693.

[7] Kouvelis P, Yu G. Robust Discrete Optimization and Its Applications. Boston, MA, USA: Kluwer Academic Publishers, 1997.

[8] Kouvelis P, Kurawarwala A A, Gutierez G J. Algorithms for robust single and multiple period layout planning for manufacturing systems. *European Journal of Operational Research*, 1992, **63**(2): 287-303.

[9] Daskin M S, Hesse S M, Revelle C S. α-reliable p-minimax regret: A new model for strategic facility location modeling. *Location Science*, 1997, **5**(4): 227-246.

[10] Kalai R, Aloulou M A, Vallin P, et al. Robust 1-median location problem on a tree. In: Proceeding of Euro Conference for Young OR Researchers and Practitioners (ORP3 2005). Valencia, Spain, 2005.

[11] Emde S, Boysen N, Scholl A. Balancing mixed-model assembly lines: A computational evaluation of objectives to smoothen workload. *International Journal of Production Research*, 2010, **48**(11): 3173-3191.

[12] Hashimoto Y, Nishikawa I, Tokumaru H. Line balancing problems using a Hopfield network. In: Proceeding of Japan-US Symposium on Flexible Automation. Kobe, Japan, 1994: 1369-1375.

[13] McMullen P R, Frazier G V. Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations. *International Journal of Production Research*, 1998, **36**(10): 2717-2741.

[14] Scholl A, Becker C. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 2006, **168**(3): 666-693.

[15] Becker C, Scholl A. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 2006, **168**(3): 694-715.

[16] Simaria A S, Vilarinho P M. A genetic algorithm based approach to the mixed-model assembly line balancing problem of type II. *Computers & Industrial Engineering*, 2004, **47**(4): 391-407.

[17] Haq A N, Rengarajan K, Jayaprakash J. A hybrid genetic algorithm approach to mixed-model assembly line balancing. *The International Journal of Advanced Manufacturing Technology*, 2006, **28**(3): 337-341.

[18] Su P, Lu Y. Combining genetic algorithm and simulation for the mixed-model assembly line balancing problem. In: Proceeding of Third International Conference on Natural Computation. Haikou, China, 2007: 314-318.

[19] Xu W, Xiao T. Robust balancing of mixed model assembly line. In: Proceeding of Asia Simulation Conference 2008/the 7th International Conference on System Simulation and Scientific Computing (ICSC 2008). Beijing, China, 2008.

[20] Scholl A. Homepage for assembly line optimization research, 2007. Available at http://www.assembly-line-balancing.de/. Accessed July 5, 2007.

[21] Frigge M, Hoaglin D C, Iglewicz B. Some implementations of the boxplot. *The American Statistician*, 1989, **43**(1): 50-54.