Evolving Solutions of Mixed-model Assembly Line Balancing Problems by Chaining Heuristic Optimization Methods

Track: Evolutionary Combinatorial Optimization and Meta-heuristics

Santiago Enrique Conant-Pablos
ITESM Campus Monterrey
Monterrey, N.L., Mexico
sconant@itesm.mx

Halley Jarumy Ferrer-Soriano
ITESM Campus Monterrey
Monterrey, N.L., Mexico
halleyferrer@hotmail.com

Hugo Terashima-Marín
ITESM Campus Monterrey
Av. E. Garza Sada 2501
Monterrey, N.L., Mexico
terashima@itesm.mx

ABSTRACT

Mixed-model Assembly Line Balancing (MALB) is needed on production of a variety of models on the same assembly line as required by just-in-time manufacturing. This paper presents an approach that applies Computational Intelligence techniques for solving MALB problems. The proposed solution consists in a heuristic optimization method that works in three stages: first, it creates an initial population of based on heuristics from classic assembly line balancing methods; second, it uses a memetic algorithm to maximize the line balancing level; and finally, it uses a min-conflicts algorithm to find a solution that better conforms to a set of preferences while trying to maintain the line efficiency of the previous stage. The results yielded by this method demonstrated to be competitive solutions and very close to the optimal.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods, Scheduling*

General Terms

Algorithms

Keywords

Metaheuristics, Combinatorial optimization, Evolution Strategies, Scheduling, Hybridization

1. INTRODUCTION AND PROBLEM

Manufacturing a product in an assembly line requires the total workload to be divided in a set of elements called tasks. Each task is performed in a given time and requires certain type of equipment and machinery besides some operator habilities. These tasks are assigned in groups to different stations in the assembly line. The assembly line balancing problem consist in finding a task-to-station assignment for all the tasks, such that all the restrictions (hard constraints), like the task precedence restrictions, and most of the preferences (soft constraints), like ergonomic recomendations, be fulfilled. Because a MALB problem is a NP-hard combinatorial optimization problem, we can not guarantee to find an optimal solution for each instance, but we are trying to get as close as possible in a reasonable time.

Vast research has been performed since its mathematical formulation was established in 1955 by Salveson [4]. Earlier heuristics are still widely applied, however, other approaches have been succesfully used, such as genetic algorithms, ant techniques, tabu search and simulated annealing. A recent survey of these models was realized by Becker and Scholl [1].

The balancing problem reported in this paper involves mixed-model assembly lines, a kind of line that assembles different product models in a mixed sequence inside the same stations. This work has been focused on assembly lines that do not experiment blocked or starved conditions. The MALB problems used to test the proposed approach are constructed under the following assumptions: tasks precedence relationships and compatibility are hard constraints, the cycle time at each station is determined during the solution process, the task times are deterministic and model specific, the product models are produced on straight assembly lines, the soft constraints are related to ergonomic issues, and the tasks precedence diagrams of all models are combined into one.

2. PROPOSED APPROACH

The solution model implemented for solving the MALB problems consists of three stages that pipeline three heuristic optimization algorithms.

The heuristic algorithm of the **first stage** creates solutions for assembling M different products, assigning N assembly tasks in a maximum of K stations by a process that sequentially selects the tasks based on a random order of application of the following three heuristics from classic methods for assembly line balancing, like in [5]. *Heuristic 1* selects first the task with the maximum number of successor tasks. *Heuristic 2* selects first the task with the maximum average processing time, as in [3]. *Heuristic 3* selects first the task with the maximum positional weight, as in [2]. Each selected task is verified against possible precedence and tasks compatibility violations. A new station is created when the current chosen task can not be added to the current open

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

station without exceeding the current cycle time T_c. The T_c is increased when the obtained solution requires more than K stations. The T_c is fixed when the first solution requiring at most K stations is obtained.

The objective function for the optimization method of the **second stage** consist of a weighted sum of a normalized sum of the differences between the cycle time and the current time of each station, and the fraction of the stations with unfulfilled soft constraints. The memetic algorithm of this work is a generational genetic algorithm that includes a local search process performed during the mutation step. The local search is executed by a simulated annealing algorithm (SAA). Chromosomes that represent every individual are coded as strings of length N, one for each task, with integer values that indicate the workstation which the tasks have been assigned to. The genetic operators used are: selection by tournament of size 2, crossover of one point, and random and locally searched mutation. For recombination, all the tasks for the stations until the crossover point are taken from parent 1, while the remaining tasks for the rest of the stations are taken from parent 2 without repetition. These are reviewed in order and the task is assigned to the first station that have enough time to accomodate it without precedence and compatibility problems. Only some of the offspring are mutated by performing some steps of a simulated annealing search. The rest of the offspring to be mutated are modified randomly. Both mutation strategies randomly select a task to be re-assigned to a new workstation following a similar strategy as the used by the crossover operator. The mutation by simulated annealing use a temperature parameter that is proportionally reduced once for each generation.

The **third stage** min-conflicts algorithm improves the best solution of the previous stage by reassigning some of the tasks in different workstations trying to reduce some soft constraints inconsistencies detected. The algorithm determines which workstations have conflicts with soft constraints, and which tasks in these stations participate in the conflicts. Then, one of these stations and one of its conflictive tasks are randomly chosen for its reassignment to a new station. The assignment that offers the biggest reduction of soft constraints, without significant reduction of the balance level, is chosen, if any. The soft constraints for this work come from ergonomic restrictions resulting from undesired concentrations of stressing tasks in certain workstations.

3. EXPERIMENTAL RESULTS

In general, the complexity for solving a MALB problem greatly depends on the *number of tasks (N)* involved, the quantity and structure of the *precedence restrictions*, the *number of stations (K)* to use, the *West ratio (WR)* (which measures the average number of tasks by station), the *Time Variability Ratio (TVR)* (which is the ratio of longest task time over the shortest task time). The features of some of the MALB problems used in the experiments are described in Table 1 where the number of product models to assemble and their production share are the same for all the instances ($M = 2$ and $q_m = [0.5, 0.5]$, respectively). T_o represents the optimal cycle time for the instance, while C represents the number of tasks that cause an ergonomic conflict.

To compare the efficiency and accuracy of the proposed approach, Table 2 presents the results obtained by three implemented variations of it. The difference among the variations is only in the second stage algorithm: HS-GA-MC use a genetic algorithm, HS-SA-MC use a simulated annealing algorithm, and HS-MA-MC use the memetic algorithm. All used empirically fine tuned search parameters. This table displays the averages of the results on the executed trials. The t column shows the execution time in minutes. The *Bal* column list the line eficiency. Finally, the *Tc* column presents the average cicle time obtained. The problems are listed by order of complexity. Each problem instance was solved 10 times by each algorithm.

Table 1: Features of solved MALB problems

Instance	N	K	T_o	WR	TVR	C
1	35	8	100	4.4	10	21
2	97	10	100	9.7	48	20
3	107	20	150	5.4	95	40
4	195	15	150	13.0	38	30
5	405	20	150	20.3	26	38

Table 2: Balancing obtained from MALB algorithms

	HS-GA-MC			HS-SA-MC			HS-MA-MC		
I	t	Bal	Tc	t	Bal	Tc	t	Bal	Tc
1	2	93.4	104	5	92.4	106	26	92.9	104
2	7	94.9	105	5	94.2	112	7	96.6	101
3	7	93.3	165	4	88.7	177	25	95.2	159
4	15	93.9	163	13	93.5	180	12	97.1	158
5	41	93.7	168	16	92.8	194	40	94.8	162

As it can be seen, the HS-MA-MC algorithm, which uses the memetic algorithm, produces competitive results and commonly better than the other two algorithms.

4. CONCLUSIONS

The results produced by the proposed approach in this paper have proved to be competitive and close to the optimal solution. The solutions it finds commonly surpases the 94% of the line efficiency when its common to accept solutions with 80% of line efficiency obtained manually with a lot of effort or using a classical ALB method. Adittionally this approach allows to manage zoning constraints and ergonomic risks; the second ones as preferences represented through soft constraints of the problem.

5. REFERENCES

[1] C. Becker and A. Scholl. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 127(3):694–715, February 2006.

[2] W. P. Helgeson and D. P. Birnie. Assembly line balancing using the ranked positional weight technique. *Journal of Industrial Engineering*, 12(6):384–398, 1961.

[3] C. L. Moodie and H. H. Young. A heuristic method of assembly line balancing for assumptions of constant or variable work element times. *Journal of Industrial Engineering*, 16(1):23–29, 1965.

[4] M. E. Salveson. The assembly line balancing problem. *Journal of Industrial Engineering*, (6):18–25, 1955.

[5] P. M. Vilarinho, A. S. Simaria and M. E. Salveson. A genetic algorithm based approach to the mixed-model assembly line balancing problem of type 2. *Computers & Industrial Engineering*, 47(4):391–407, 2004.