G Model
CIRPJ 425 No. of Pages 12

ARTICLE IN PRESS

CIRP Journal of Manufacturing Science and Technology xxx (2017) xxx–xxx

Contents lists available at ScienceDirect

CIRP Journal of Manufacturing Science and Technology

journal homepage: www.elsevier.com/locate/cirpj

A multi-objective software tool for manual assembly line balancing using a genetic algorithm

M. Dalle Mura*, G. Dini

Department of Civil and Industrial Engineering, University of Pisa, Via Diotisalvi, 2, Pisa 56122, Italy

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Assembly line balancing
Genetic algorithm
Multi-objective optimization

ABSTRACT

This paper proposes a new genetic algorithm approach for solving a multi-objective assembly line balancing problem. The objectives concern the minimization of the number of workstations and the workload variance, typically faced by most systems presented in literature, but also the minimization of three further aspects, not simultaneously treated in literature and very important in manual assembly lines: the number of skilled workers among workstations, the number of assembly equipment (e.g.: automatic screwdrivers, pressing machines, etc.) and the number of assembly direction changes along the sequence. To demonstrate the effectiveness of the proposed method in finding optimized assembly sequences, a classical case study taken from the literature is finally discussed.

Introduction

The Assembly Line Balancing Problem (ALBP) is considered to be one of the main issues in the design and planning of manufacturing systems, because of its combinatorial complexity [1] and great impact from an economic point of view. The classical single-model version, known as the Simple Assembly Line Balancing Problem (SALBP) [2], has been widely treated in the literature from many years [3–7]. Among the different objectives to be pursued [8–10], the minimization of the number of workstations for a given cycle time refers to the first optimization problem (SALBP-1), whose main assumptions are [11]:

- mass production of one homogeneous product;
- paced line with fixed cycle time;
- deterministic execution times;
- serial line layout, one-sided stations;
- constant repositioning time throughout the workstations;
- fixed launch interval corresponding to cycle time.

Different methods are currently available for its resolution [12–15], such as heuristic techniques, which can give good results for simple problems in a short time [16,17]. A solution close to the optimal can also be found through iterative algorithms that, in a reasonable computing time, reach a gradual convergence of the objective function. In recent years, other algorithms, mostly inspired by the biological world, such as neural networks and ant colony optimization, have been employed to solve this problem. Among them, the genetic algorithm (GA) has been used for the resolution of a wide variety of combinatorial problems, because of the demonstrated success in the results it can achieve [18,19]. As a matter of fact, the GA can generate optimum solutions faster than other algorithms. Researchers also use GA in ALBP because it is able to solve complex and multiple constraint problems. On the other hand, some GA aspects are inappropriate for the ALBP and must be addressed to make it suitable for these optimization problems [20]. One characteristic is the binary string originally designed for chromosomes and another issue is the feasibility of the population, whose precedence constraints must not be violated by crossover and mutation operators [18]. The main weakness of the GA remains, however, the early convergence is susceptible to.

The rest of the paper is organized as follows: in Section "Genetic algorithm in assembly line balancing," the contributions of reviewed literature based on various solutions of GA for ALBP are discussed; in Section "Multi-objective optimization" the concept of multi-objective optimization is presented; the software tool developed to solve SALBP-1 through a multi-objective optimization is presented in Section "Proposed software tool"; the results obtained using a case study taken from the literature are

* Corresponding author.
E-mail address: m.dallemura@ing.unipi.it (M. Dalle Mura).

http://dx.doi.org/10.1016/j.cirpj.2017.06.002
1755-5817/© 2017 CIRP.

Nomenclature

d_i	assembly direction associated to task *i*
D_i	direction change between task *i* and task *i* + 1
e_{ik}	equipment *k* associated to task *i*
E_m	number of different equipment in workstation *m*
F	fitness function
F_n	normalized value of objective *n*
I_n	current value of the index of objective *n*
LB_n	lower bound of objective *n*
n_d	number of directions in the assembly sequence
n_k	number of equipment
n_t	number of assembly tasks
p_{ij}	precedence relation between task *i* and task *j*
s_i	skill level associated to task *i*
S_m	maximum value of skills requested on workstation *m*
T_{am}	idle time of workstation *m*
T_{amean}	mean idle time between workstations
T_c	cycle time
T_p	total assembly time of the product
UB_n	upper bound of objective *n*
w_n	weight of objective *n*
x_k	maximum value of e_{ik} for the task *i*

given in Section "Case study"; lastly, in "Conclusion", the conclusions and future research directions are remarked.

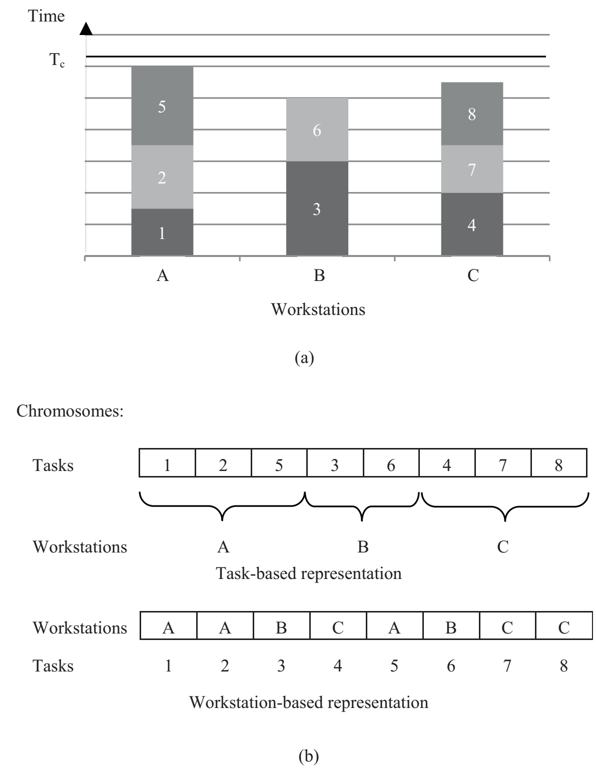

Fig. 1. Chromosome representation schemes used for ALBP: (a) Example of balancing problem solution and (b) Corresponding chromosome structures.

Genetic algorithm in assembly line balancing

The different approaches used by the authors in implementing the main GA features and parameters are emphasized below.

Initial data and parameters

The initial data for a GA in assembly line balancing are the precedence relationships and the algorithm parameters (i.e.: population size, number of iterations, etc.).

Precedence relationships are always first represented by a topology network where tasks are expressed by nodes and direct priority constraints are expressed by arcs [21], all this then codified by a binary square matrix.

Among algorithm parameters, the termination criterion has been noted to correspond, in most of cases, to the maximum number of generations [22–26]. In other research works the algorithm stops when a convergence of the fitness function, combined to a predetermined number of iterations, has been reached [27–29]. This termination criterion makes the GA more efficient if no improvement in the best solution occurs after a certain amount of generations.

The population size strictly depends on the problem to be solved, namely on the number of tasks needed to assemble a specific product: the range found in the reviewed literature goes roughly from a minimum of 20 individuals to a maximum of 100.

Chromosome structure

The genetic representation uniquely identifies an individual of the population in order to convert the solution of the problem into a string called chromosome. Different chromosome structures have been proposed by the researchers. The most used chromosome structure in the literature for the ALBP is the task-based representation [24,26,27], in which a chromosome is defined as a sequence of task. The workstation-based representation, used in Refs. [25,28,29], consists of a sequence of labels indicating workstations to which the tasks are assigned. These encoding methods are both represented in Fig. 1. Other chromosome structures, less widely used, are the grouping-based representation, where the workstations are represented by augmenting the workstation based chromosome with a group part that lists all the workstations [30], and the heuristic-based representation, where the chromosome length is defined by the number of heuristics used to assign the tasks to the workstations [31,32].

An appropriate chromosome representation scheme is fundamental for GA, since the application of genetic operators may result in solutions that violate precedence constraints. Heuristic-based chromosomes have the advantage to achieve feasibility [19], but in case of genetic operators specifically designed to solve this problem, the task-based representation is the most suitable to describe the assembly sequence.

Fitness function

The fitness function is used to provide a measure of performance for each individual of the population, using the predetermined objectives. The minimization of the number of workstations is pursued in many researches [33], most of which try to obtain better balanced solutions at the same time through the maximization of the workload smoothness (i.e. minimization of the workload variance) [24,34,35]. The balance efficiency is maximized in Ref. [36], whereas in Ref. [37] the cycle time is minimized, simultaneously with the workload variance and the frequency of tool changes. Contributions relative to other line parameters can be found in Ref. [38], where a method to choose the type of equipment to place in every station of the assembly line is proposed, in order to minimize the total equipment cost. In Ref. [39] the problem to minimize the number of temporary workers in favor of permanent employees is discussed, in particular for