Int. J. Production Economics 120 (2009) 315–326

Contents lists available at ScienceDirect

Int. J. Production Economics

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

Backup strategy for robots' failures in an automotive assembly system

Tomer Kahan [a], Yossi Bukchin [a,*], Roland Menassa [b], Irad Ben-Gal [a]

[a] Department of Industrial Engineering, Faculty of Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel
[b] Manufacturing Systems Research, GM Research & Development, MC-480-106-359, 30500 Mound Road, Warren, MI 48090-9055, USA

ARTICLE INFO

Article history:
Received 23 October 2006
Accepted 16 September 2007
Available online 18 January 2009

Keywords:
Automotive industry
Assembly lines
Spot welding
Robot failures
Line balancing

ABSTRACT

Automotive assembly lines are often characterized by robots' failures that may result in stoppages of the lines and manual backup of tasks. The phenomena tend to impair throughput rate and products' quality. This paper presents a backup strategy in which working robots perform tasks of failed robots. The proposed Mixed-Integer Linear-Programming based approach minimizes the throughput loss by utilizing the robots' redundancy in the system. Two algorithms are developed to comply with stochastic conditions of a real-world environment. The performance of these algorithms is compared with several heuristics, and the downstream-backup based algorithm is found superior to all other methods.

1. Introduction: body-shop systems in the automotive industry

High-volume body-shop systems in the automotive industry often consist of a series of assembly zones that are serially connected via automated material handling (MH) systems. A zone contains several robotic cells (also called stations), each of which consist of several welding robots that are working simultaneously. The automated MH system is used for feeding the stations with parts that are assembled (welded) to the vehicle body. These MH systems are usually asynchronous where carriers can circulate if they are not blocked or starved.

Weld spots are grouped on the basis of their location in the vehicle body and performed sequentially by a single welding robot. There are two types of weld spots: dimensional control welds (DCWs) and respot welds (RSPs). In DCWs, a new part is welded to the vehicle's body to define a new geometry of the vehicle. A station which performs DCWs is usually facilitated by an automated MH system (and sometimes by another dedicated robot) which transfers the parts that have to be assembled. RSPs are performed on an existing geometry—no new part is assembled, and the sole purpose of the RSPs is to strengthen the vehicle's body.

Each robot can weld a single group of spots or multiple groups of spots in a single work cycle. The welding task, performed by a spot welding-gun, consists of the robot motion from the "Home position" to the welding area and back to the "Home" position, in addition to the time dedicated to each welding spot.

The problem addressed in this paper refers to a situation, depicted in Fig. 1, in which one robot or multiple robots fail during the operation time. The proposed recovery plan or a backup plan should then indicate which robot(s) replace the failed ones during the repair period. The backup plan aims at minimizing the failures effects on the throughput rate.

It is assumed that the *capability* of each robot, in terms of the weld spots it can perform, is known and given, as well as the *precedence relationships* among various groups of spots. The precedence relationships indicate the assembly sequence among groups of spots, and eliminate infeasible situations. For example, a situation where a

* Corresponding author. Tel.: +972 3 6407941; fax: +972 3 6407669.
E-mail address: bukchin@eng.tau.ac.il (Y. Bukchin).

0925-5273/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijpe.2007.09.015

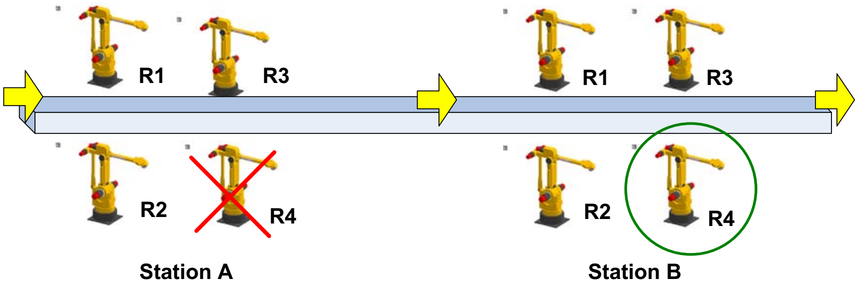

Fig. 1. The problem illustration: robot R4 in station A fails and is replaced by robot R4 in station B.

downstream robot has to backup a weld spot on a surface which has been already covered by other parts. Here we propose a backup plan only to the RSPs (about 85% of the total number of spots). For the DCWs it is assumed that the existence of the automated MH system, which handles the newly assembled parts, prevents the possibility to perform this task in another station.

The proposed approach solves a static problem, in which one or more robots fail in some observed state of the system. Given this state, the suggested procedure provides the optimal backup robot(s) to perform the groups of spots that were previously performed by the failed robot(s). In practice, the proposed procedure can be executed offline by considering in advance common failure scenarios and obtaining a list of backup robots per potential failure. Since we consider the reallocation of only those spots that belong to failed robots, while maintaining the original welding allocation, the number of decision variables is relatively small. As a result, the obtained algorithmic backup procedure is computationally tractable, and thus can be executed online. The online backup plan is executed following each robot failure while updating the system state based on the robots' status. Both types of solutions are further discussed in later sections.

The remainder of the paper is organized as follows. In the next section related literature is reviewed, particularly literature concerned with assembly line balancing. In Section 3, mixed integer linear programming (MILP) formulations are developed for the selection of the backup robot(s) in case of a failure(s). The concept of a capability matrix (CM) required for this end is also presented. In Section 4, a small scale illustrating example is given, along with the analysis of the solution characteristics as a function of the problem parameters. Section 5 deals with a real-world environment which captures the stochastic characteristics of robots' failure and repair. Several backup heuristic rules are developed and a comprehensive comparison between the proposed solution approach and these heuristics is presented. The summary and concluding remarks are given in Section 6.

2. Literature review

The considered problem can be viewed as a sub-problem of the body shop design and operation. Due to the complex nature of this problem, most of the research in this field regarding the system design and operation has been based on simulation. Several different types of simulation software packages have been used for this purpose. In general, the design process consists of the work-cell design and the system design (Moon et al., 2006). For the former, 3D simulation software can be used (e.g. IGRIP®, RobCAD®, QUEST®, Factory CAD®) in order to examine the detailed motion of the robots, collisions among parts, jigs and robots. These tools can enable us to construct and simulate virtual factory relatively fast. Such an approach is suggested by Noh et al. (2001). For the system design stage, discrete event simulation software can be used to examine system performance measures such as cycle time, work-in-process inventory, flowtime, etc. A procedure for simulation-based optimization is presented in Spieckermann et al. (2000), who combine simulation with metaheuristics such as simulated annealing and genetic algorithm. The existed gap in the above approaches concerns with the interface between the two modules (work cell and system design). Since either of the simulation tools is highly time consuming, in particular the 3D simulation, one cannot apply an efficient combinatorial optimization approach for the spot allocation problem. Such an approach is suggested in this paper, as the considered problem is viewed as a special case of the assembly line balancing problem.

In most manufacturing environments, as in the environment addressed here, there is a clear motivation to balance the work load among the system's resources. Work load balance avoids idleness of resources and often results in a higher throughput rate. This concept is explicitly applied in the assembly line balancing problem, where a set of assembly tasks are assigned to assembly stations in order to minimize the number of stations subject to a required throughput rate, or minimize the cycle time subject to a given number of stations. In both cases, the utilization of the resources is maximized and the idle time is minimized. When a single product type (single model) is concerned, the problem is defined as the simple assembly line balancing problem (SALB-P). This problem is proven to be NP-Hard (Karp, 1972), and many optimal and heuristic algorithms for this problem have been developed during the last 50 years. Baybars (1986) presents a survey on optimal procedures developed for this problem. Scholl and Becker (2006) present a state of the art survey of optimal and heuristic procedures for the

SALB-P. Amen (2000) presents a survey on heuristic approaches for assembly line balancing when cost is explicitly considered. The assembly line balancing literature consists of many variations of the basic problem, such as assembly lines with stochastic task times, mixed-model lines (where different product types are assembled on the same line), paced lines versus un-paced lines, equipment selection, etc. Ghosh and Gagnon (1989) review optimal and heuristic procedures for several variations of the problem. An up-to-date survey is provided in Becker and Scholl (2006), which addresses the generalized assembly line balancing (GALB-P). Boysen et al. (2008) addresses the gap between research and practice. They classify the variations of the line balancing problem and suggest relevant models for the real-world problems.

Although most traditional literature addresses manual assembly lines, some papers take into account the equipment required for the assembly process. Graves and Holmes Redfield (1988) consider the mixed-model assembly line design problem, where each task can be performed by one or more alternative types of equipment. Assuming a fixed sequence of the assembly tasks and large similarities among different products, they suggest a procedure for the design process, consisting of the simultaneous task assignment and equipment selection. Rubinovitz and Bukchin (1993), and later on Bukchin and Tzur (2000), address a similar problem, where a single model is concerned with a relatively flexible assembly sequence, expressed by a precedence diagram. The task assignment, along with equipment selection out of multiple alternatives, is performed by using a branch and bound optimal procedure for moderate sized problems. Another branch and bound based heuristic is proposed for solving large scale problems. Bukchin and Rubinovitz (2003) extend the above problem to address the possibility to apply parallel stations in the assembly line.

The problem considered in this paper can be viewed as another variation of the classic assembly line balancing problem, where the assembly equipment, spot-welding robots in this case, is taken into account. Nevertheless, unlike the above, an operational problem rather than a design problem is considered here, where the robots are already placed in stations. Each time one or more robots fail, the problem of assigning the group(s) of spots (tasks) of the failed robot(s) to other working robot(s) can be considered as a re-balancing problem. Fortunately, since only the failed groups (i.e., the groups that were assigned to the failed robot) are to be reassigned to the backup robots, it is found that relatively large problems can be solved in a relatively reasonable amount of time.

3. Spot re-allocation models

3.1. Preliminaries and definitions

The problem of spots reallocation due to a robot's failure can be addressed in three hierarchical levels: (level 1) single-robot backup; (level 2) group allocation based multi-robot backup; and (level 3) spot allocation based multi-robot backup. In the first level, the whole work content, consisting of one or several groups of welding spots of the failed robot(s), is reallocated to a single backup robot. In the second level, each group performed by the failed robot is reallocated as a whole; yet, different groups of spots can be allocated to different backup robots. In the third level, any of the spots in each of the failed groups can be individually allocated among different backup robots.

There is a clear tradeoff between the quality of the proposed solution and the required algorithmic complexity. The proposed solution framework is general enough to be implemented in each of these hierarchical levels. Nevertheless, the proposed solution approach focuses on the first two hierarchical levels from practical considerations. Splitting spots within a group (the third hierarchical level) might be attractive with regard to the cycle time reduction. However, the current available technology, both at the controllers and the robotics stations, do not enable an efficient execution at this level.

As discussed above, the determination of the backup robot(s) is somewhat similar to the re-balancing of an assembly line which aims at minimizing the cycle time. Consequently, the proposed MILP, on which the solution approach is based, is an enhanced version of similar formulations known in the area of assembly line balancing. In the proposed formulation, robots and groups of spots are analogous to the stations and tasks in the traditional assembly line, respectively. Consequently, we assume that the robot load is equal to the summation of all welding times of the groups of spots assigned to this robot, and the line cycle time is determined by the most loaded robot. In the proposed model we consider only the group of spots of the failed robot(s) to be reallocated, while shifting groups of working robots is not allowed. Accordingly, we expect the number of integer variables to be much smaller with respect to the assembly line balancing problems. As noted above, the reduction in the number of variables leads to a solution which can be attained in a relatively small amount of time. Next we present the used notation and formulations.

Notation

Sets and parameters

I	set of total groups of spots
I^W	set of total groups of spots assigned to working robots, $I^W \subseteq I$
I^F	set of total groups of spots assigned to failed robots, $I^F = I/I^W$
R	set of all the robots located in the assembly line
R^W	set of working robots, $R^W \subseteq R$
R^F	set of failed robots, $R^F = R \backslash R^W$
T_i	performance time of group of spots i (including setup)
IP_i^W	set of immediate predecessors of group of spots i assigned to working robots
IS_i^W	set of immediate successors of group of spots i assigned to working robots

IP_i^F set of immediate predecessors of group of spots i assigned to failed robots

IS_i^F set of immediate successors of group of sport i assigned to failed robots

Matrices

Two matrices describe the current and the potential work allocation in the system. The former is given by the initial matrix (*IM*) and the latter given by the *CM*. The entries in these matrices satisfy the following rule.

$$IM_{ir} = \begin{cases} 1, & \text{if a group of spots } i \text{ is performed by robot} \\ & r \text{ the system's initial state} \\ 0 & \text{otherwise} \end{cases}$$
$$\forall i \in I,\ \forall r \in R,$$

$$CM_{ir} = \begin{cases} 1, & \text{if a group spots } i \text{ can performed by robot } r \\ 0 & \text{otherwise} \end{cases},$$
$$\forall i \in I,\ \forall r \in R$$

Note that $CM_{ir} \geqslant IM_{ir}\ \ \forall i \in I,\ \forall r \in R$.

A third matrix, which represents the solution of the backup problem, is the recovery matrix (RM) and will be explained later.

Decision variables

$$x_r = \begin{cases} 1, & \text{if robot } r \text{ is a backup robot} \\ 0 & \text{otherwise} \end{cases} \quad \forall r \in R,$$

c—cycle time of the system

$$x_r = \begin{cases} 1, & \text{if group spots } i \text{ is performed by robot } r \text{ after failure} \\ 0 & \text{otherwise} \end{cases},$$
$$\forall i \in I^F,\ \forall r \in R^W$$

Prior to implementing the solution procedure, the *CM* and the precedence diagram should be obtained. The CM captures the redundancy of the system and the capability of different robots with regard to performing different groups of spots. The precedence diagram consists of the precedence constraints among groups. The CM, along with the precedence diagram, establish the sets of constraints for the backup strategy.

As noted above, each element CM_{ir} of the *CM* is equal to 1 if group of spots i can be performed by robot r and 0 otherwise. The CM is generated on the basis of the initial matrix, *IM*. Each row of *IM* is examined with respect to those robots that are capable of performing the respective group of spots, in addition to the originally allocated robot. For each robot r, which is capable to backup group of spots i, we set $Cm_{ir} = 1$. The capability of a robot to become a backup robot mainly depends on its gun configuration and its physical location. The latter determines the work envelope of the robot and its feasibility to perform group of spots i. The capability of the robot remains unchanged as long as no technological changes were applied. If no redundancy exists ($CM = IM$), i.e., each group can be performed by a single robot only, no backup is available and the groups of the failed robot(s) should either wait for the robot to recover or be backed up in a manual station at the end of the zone. The other extreme, according to which $CM_{ir} = 1\ \forall i, \forall r$, represents a maximal redundancy level, which is uncommon in practice.

The precedence diagram consists of the technological precedence relationships between groups of spots or between spots within groups. These constraints result from the product structure along with the characteristics of the production system. For example, a precedence constraint may result if during the assembly process the access to a certain location in the body is avoided due to its covering by a welded part. Note, that a common precedence diagram provides much flexibility which enables numerous feasible assembly sequences. The precedence constraints can be expressed in a diagram, as shown later on in Fig. 4.

3.2. General robot backup (GRB) formulation

The proposed formulation considers a situation where one or more robots fail and their groups of spots are reallocated to multiple backup robots simultaneously; still this number can be limited subject to managerial decisions. This new allocation should satisfy the capability and precedence constraints while minimizing the throughput loss (or the cycle time). The proposed model is a general one; some special cases are derived later on, as seen in the sequel.

Model GRB:

$$\text{Minimize} \quad c \tag{1}$$

Subject to:

$$\sum_{i \in I^W} T_i \cdot IM_{ir} + \sum_{i \in I^F} T_i \cdot rm_{ir} \leqslant c \quad \forall r \in R^W \tag{2}$$

$$\sum_{r \in R^W} rm_{ir} = 1 \quad \forall i \in I^F \tag{3}$$

$$x_r \geqslant rm_{ir} \quad \forall i \in I^F,\ \forall r \in R^W \tag{4}$$

$$rm_{ir} \leqslant CM_{ir} \quad \forall i \in I^F,\ \forall r \in R^W \tag{5}$$

$$\sum_{k \in R^W} k \cdot IM_{hk} \leqslant \sum_{l \in R^W} l \cdot rm_{il} \quad \forall i \in I^F,\ \forall h \in IP_i^W \tag{6}$$

$$\sum_{k \in R^W} k \cdot rm_{ik} \leqslant \sum_{l \in R^W} l \cdot IM_{gl} \quad \forall i \in I^F,\ \forall g \in IS_i^W \tag{7}$$

$$\sum_{k \in R^W} k \cdot rm_{hk} \leqslant \sum_{l \in R^W} l \cdot rm_{il} \quad \forall i \in I^F,\ \forall h \in IP_i^F \tag{8}$$

$$\sum_{k \in R^W} k \cdot rm_{ik} \leqslant \sum_{l \in R^W} l \cdot rm_{hl} \quad \forall i \in I^F,\ \forall h \in IS_i^F \tag{9}$$

$$\sum_{r \in R^W} x_r \leqslant R_{MAX} \tag{10}$$

$$rm_{ir} \in \{0, 1\} \quad \forall i \in I^F,\ \forall r \in R^W \tag{11}$$

$$x_r \in \{0, 1\} \quad \forall r \in R \tag{12}$$

$$c \geqslant 0 \tag{13}$$

The objective function (1) minimizes the system's cycle time, i.e., maximizes the throughput rate. The cycle time constraint set (2) enforces the system's cycle time to be larger than or equal to the assembly time of the most loaded robot. The assembly time of a single operational robot consists of two components; the first component captures the constant assembly time, namely, the initial assembly time of a specific robot, prior to any robot's failure, and the second component contains the additional assembly time added to a working robot due to other robots' failures. Note that the values of T_i can be taken either directly from the line or using robotic CAD systems. According to constraint set (3), each group of spots previously done by the failing robot(s) will be backed up by some working robot. Constraint set (4) enforces the x_r variables to be equal to 1 if robot r is a backup robot. The suitability of each robot r to serve as a backup robot, based on the CM, is verified in constraint set (5).

Constraint sets (6)–(9) are precedence constraints which assure that the new assignment of the failed groups will still satisfy the technological precedence relationships. Constraint sets (6) and (8) assure that a failed group i will be performed after the completion of each of its immediate predecessor, h, as group h belongs to a working robot in the former and to a failed robot in the latter. Constraint sets (7) and (9) enforces the failed task i to be completed before starting each of its immediate successors, g, as group g below to a working robot in the former and to a failed robot in the latter.

Constraint set (10) is optional and enables to limit the number of backup robots. Constraints sets (11) and (12) are integrality constraints and (13) is the non-negativity constraint of the cycle time, c.

Several special cases can be derived from the above formulation. In case $R_{MAX} = 1$, each time a failure occurs all the groups of the failed robot(s) are performed by a single backup robot. This limitation simplifies the solution implementation; however, it will most likely result in a poor solution, especially in a relatively balanced system. In this case, the backup robot will end up with a relatively high cycle time. This model is called the single-robot backup (SRB) model. The other extreme refers to the situation where constraint sets (4), (10) and (12) are omitted. In this case, there is no limitation on the number of backup robots and the solution is expected to be much better. We refer to this model as the multiple-robot backup (MRB) model. Another difference between SRB and MRB model relates to the solution run time, as the latter model is expected to take higher computation time due to the additional integer variables.

4. Problem analysis

In this section we start with a small-scale problem, focusing on a single assembly zone, in order to illustrate the difference between the SRB and the MRB formulations. Next, the MRB formulation is tested and analyzed under a large scale environment. By using a full factorial experiment, we examine the effect of main problem parameters on the cycle time following a robot's failure and a reallocation of welding spots.

4.1. Small-scale illustrative example

To illustrate the performance of the above formulation, a small scale example is presented, focusing on a single zone. Fig. 2 depicts the layout of the considered assembly zone which consists of four stations with a total of 14 robots. Let us assume that robot no. 7 and robot no. 12, marked by the gray color in the figure, fail and require backup by the working robots. The problem to be solved is how to reallocate the group of spots of these failed robots.

The initial matrix, *IM* and the *CM* are presented in Fig. 3(a) and (b), respectively. Each column (row) represents a robot (a group of spots). The bolded columns and rows represent failing robots and their corresponding groups of spots (i.e., $R^F = \{7,12\}$ and $I^F = \{8,9,15,16,17\}$). Note that groups of spots 1 and 2 are considered as *DCWs*, i.e., they do not have any backup. The precedence constraints among groups and the process times are given in Fig. 4, where each node represents a group of spots with its process time, T_i, written above the node. The gray cells in Figs. 3 and 4 represent the reallocated groups and their respective process times.

Note that at the initial state, prior to any failure, the bottleneck robot is robot no. 2 with a cycle time of 38 s. When failure occurs, we start by solving the SRB model, which results in robot no. 11 as the chosen backup robot. Under this scenario, the obtained cycle time is equal to $\sum_{i\in I^W} T_i \cdot IM_{i,11} + \sum_{i\in I^F} T_i \cdot rm_{i,11} = 100\,\text{s}$. Note that the "single robot constraint" yields a significant increase in the cycle time. Next, we solve the MRB model, which allows backup by multiple robots. Now the obtained

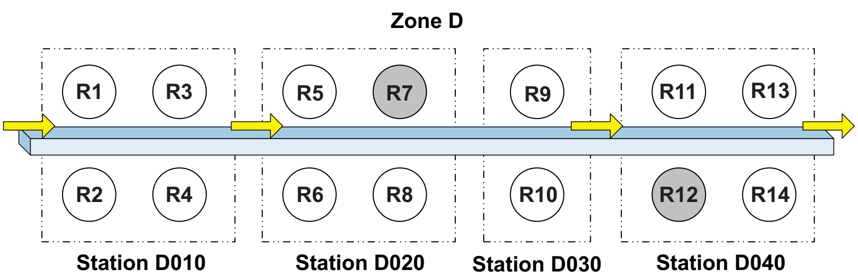

Fig. 2. The layout of the illustrative example.

a

Group of spots	Robot 1	2	3	4	5	6	**7**	8	9	10	11	**12**	13	14
1	1	0	0	0	0	0	**0**	0	0	0	0	**0**	0	0
2	0	1	0	0	0	0	**0**	0	0	0	0	**0**	0	0
3	0	0	1	0	0	0	**0**	0	0	0	0	**0**	0	0
4	0	0	1	0	0	0	**0**	0	0	0	0	**0**	0	0
5	0	0	0	1	0	0	**0**	0	0	0	0	**0**	0	0
6	0	0	0	0	1	0	**0**	0	0	0	0	**0**	0	0
7	0	0	0	0	0	1	**0**	0	0	0	0	**0**	0	0
8	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
9	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
10	0	0	0	0	0	0	**0**	1	0	0	0	**0**	0	0
11	0	0	0	0	0	0	**0**	1	0	0	0	**0**	0	0
12	0	0	0	0	0	0	**0**	0	1	0	0	**0**	0	0
13	0	0	0	0	0	0	**0**	0	0	1	0	**0**	0	0
14	0	0	0	0	0	0	**0**	0	0	0	1	**0**	0	0
15	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
16	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
17	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
18	0	0	0	0	0	0	**0**	0	0	0	0	**0**	1	0
19	0	0	0	0	0	0	**0**	0	0	0	0	**0**	0	1

b

Group of spots	Robot 1	2	3	4	5	6	**7**	8	9	10	11	**12**	13	14
1	1	0	0	0	0	0	**0**	0	0	0	0	**0**	0	0
2	0	1	0	0	0	0	**0**	0	0	0	0	**0**	0	0
3	0	0	1	0	0	0	**1**	0	0	0	0	**0**	0	0
4	0	0	1	0	0	0	**1**	0	0	0	0	**0**	0	0
5	0	0	0	1	0	0	**0**	1	0	0	0	**0**	0	0
6	1	0	0	0	1	0	**0**	0	1	0	0	**0**	0	0
7	0	0	0	0	0	1	**0**	0	0	1	0	**0**	0	0
8	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**
9	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**
10	0	0	0	1	0	0	**0**	1	0	0	0	**1**	0	0
11	0	0	0	1	0	0	**0**	1	0	0	0	**1**	0	0
12	0	0	0	0	1	0	**1**	0	1	0	1	**0**	0	0
13	0	0	0	0	0	0	**0**	0	0	1	0	**0**	0	0
14	0	0	0	0	0	0	**0**	1	0	0	1	**1**	0	0
15	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**1**
16	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**
17	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**1**
18	0	0	0	0	1	0	**1**	0	1	0	1	**0**	1	0
19	0	0	0	0	0	0	**0**	0	0	0	0	**1**	0	1

Fig. 3. Input data of the small-scale example. (a) initial matrix, *IM* (b) capability matrix, *CM*.

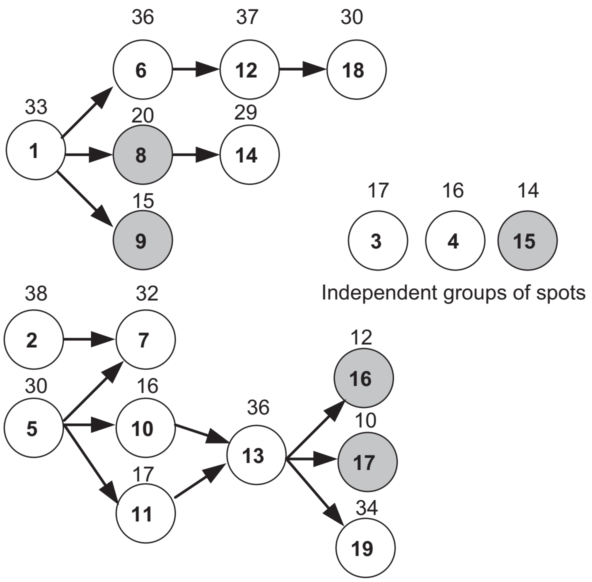

Fig. 4. Precedence diagram of the small-scale example.

Reallocated Group of spots	Robot 1	2	3	4	5	6	8	9	10	11	13	14
8	0	0	0	0	0	0	0	0	0	1	0	0
9	0	0	1	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	1	0	0	0	0	0
16	0	0	0	0	0	0	0	0	1	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	1

Fig. 5. Recovery matrix (*RM*) for the small-scale example.

solution, consisting of the reallocation of the failed groups, is given in the *RM*. This matrix consists of the failed groups only (rows) and the robots (columns), as can be seen in Fig. 5. Consequently, each column containing an element with a value of 1 represents a backup robot. As we can see, the five failed groups, 8, 9, 15, 16 and 17 are now performed by five different backup robots no. 11, 3, 8, 10 and 14, respectively. The new bottleneck robot, is still robot no. 11 (resulting from $\arg\max_{r\in R^W}(\sum_{i\in I^W} T_i \cdot IM_{ir} + \sum_{i\in I^F} T_i \cdot rm_{ir}) = 11$, consequently with a much lower cycle time of $\sum_{i\in I^W} T_i \cdot IM_{i,11} + \sum_{i\in I^F} T_i \cdot rm_{i,11} = 49\,\text{s}$.

4.2. Analysis of the solution approach

In the following experimentation, we examine the effect of some of the problem parameters on the cycle time after the reallocation of failed spots, using the MRB model. The examined environment is based on a full body-shop assembly line, consisting of four zones, in which 128 groups of spots are performed. The cycle time of the line before failure is 42 s. The problem parameters that are used as the experimental factors for the analysis are the following:

1. The number of failed groups.
2. The flexibility ratio (F-ratio) as defined by Dar-El (1973). This factor provides a quantitative expression for the level of flexibility in the assembly sequence, as represented by the precedence diagram. In particular, F-ratio $= 1-H/B$, where H is the actual number of precedence constraints and B is the maximal possible number of precedence constraints. Note that $0 \leqslant$ F-ratio $\leqslant 1$, where F-ratio $= 1$ denotes a maximal flexibility where no precedence constraints among groups exist (see, for example, the part of the precedence diagram in Fig. 4, which consists of groups of spots no. 3, 4 and 15), while an F-ratio $= 0$ denotes a

		Robot					
Group of spots	N_i	1	2	3	4	5	6
1	5	1	0	1	0	1	0
2	7	1	0	1	0	1	0
3	4	0	1	0	1	0	1
4	8	1	0	1	0	1	0
5	5	0	0	0	1	0	1
6	5	0	0	1	0	1	0
7	6	0	0	1	0	1	0
8	4	0	0	0	1	0	1

Fig. 6. Capability matrix (*CM*) for redundancy calculation.

situation with no flexibility at all, where the assembly sequence is strict (see for example the groups of spots no. 6, 12 and 18 in Fig. 4).

3. The redundancy factor, RDF, reflects the percentage of redundancy in the CM. This proposed factor, has two forms—a weighted and an un-weighted expression given by $RDF=(\sum_{i\in I}\sum_{r\in R}CM_{ir})-|I|/(|I|(|R|-1))$ and $RDF_W=\sum_{i\in I}N_i(\sum_{r\in R}CM_{ir}-1)/((|R|-1)\sum_{i\in I}N_i)$, respectively. The former expression gives the amount of redundancy that exists in the system, as a zero RDF value stands for a "no backup" situation in the system, while an RDF value of one stands for maximal backup, where each group of spots can be performed by all robots. In the latter expression, the groups of spots are weighted by the number of spots in each group. Based on the sample data which appears in Fig. 6, $RDF=12/40=0.3$, and while considering the number of spots in each group i, we get $RDF_W=66/220=0.309$.

A full factorial experiment is conducted by determining three levels for the F-ratio and the RDF factors resulting in scenarios of failed robots. In each scenario, six replications have been performed, where in each time the number of failed robots was randomly generated between 1 and 4 resulting in a number of failed groups, which is between 1 and 8. Consequently, an overall of 54 experiments are conducted in total. The system configuration was given and remained unchanged along the experimentation. An ANOVA with a confidence level of 95% has been constructed for the cycle time, and the results are described below. The analysis of variance, presented in Table 1, indicates that the model, as well as each of the three factors, is significant. The obtained results show that the cycle time increases significantly with the number of failed groups and decreases significantly with the F-ratio and the RDF_W factors, as can be seen in Fig. 7. One can see that these results are quite intuitive; the cycle time is supposed to increase in the amount of work to be reallocated. The number of backup options is expected to increase in the values of the F-ratio and the RDF_W, resulting in an improved cycle time. Nevertheless, one can see a 'diminishing return' of the F-ratio and the RDF_W factors with regard to the cycle time, as the major part of the improvement is associated with

Table 1
ANOVA table for the cycle time.

Source	Sum of squares	DF	Mean square	F value	Prob > F
Model	2376.96	7	339.57	15.944	<0.0001
# failed	1913.86	3	637.95	29.955	<0.0001
F-ratio	292.44	2	146.22	6.866	0.0028
RDF w	555.10	2	277.55	13.032	<0.0001
Residual	809.28	38	21.30		
Lack of fit	512.06	24	21.34	1.005	0.5127
Pure error	297.22	14	21.23		
Cor total	3186.25	45			

the F-ratio (RDF_W) increase from the lower level, F1, (R1) to the medium level, F2, (R2), while a smaller improvement is associated with an increase from the medium level, F2, (R2) to the higher level, F3, (R3).

5. Performance analysis of a "real-world" setting

5.1. Aspects of real-world implementation

The implementation of the above model in a stochastic industrial environment is not straight-forward. The model presented above focuses on a particular system state after a failure occurs, and aims at finding the best backup solution for this occurrence without taking into consideration future trajectories of failures and repairs. However, in a real-world environment, failures occur randomly over time, the repair time is a random variable and the number of simultaneously failed robots may change. In such an environment, the effectiveness of our myopic solution is no longer guaranteed. In this section, we aim at analyzing the effectiveness of implementing the MILP model to a real world setting. The implementation of the proposed procedure should consider the following two aspects:

1. *Avoiding the reallocation of 'working' groups*: the chronology of the decision-making process may imply the reallocation of groups of spots of working robots. Assume, for example, that robot X has failed, and a backup solution was applied. While robot X is down, robot Y also fails. Clearly, the backup solution for both failed robots, X and Y, may require an additional reallocation of the groups of robot X. However, from practical considerations, we avoid applying such an action in the proposed procedure since it would increase the variability in the system, which has to be avoided in real settings. Consequently, we allow the reallocation of groups of failed robots only.
2. *Online versus offline procedure*: each time a robot state is changed, the solution is either calculated online, or a-priori generated solution is implemented. The implementation of an online procedure depends on the capabilities of the online computing system in the shop floor and the expected algorithmic run time (note that

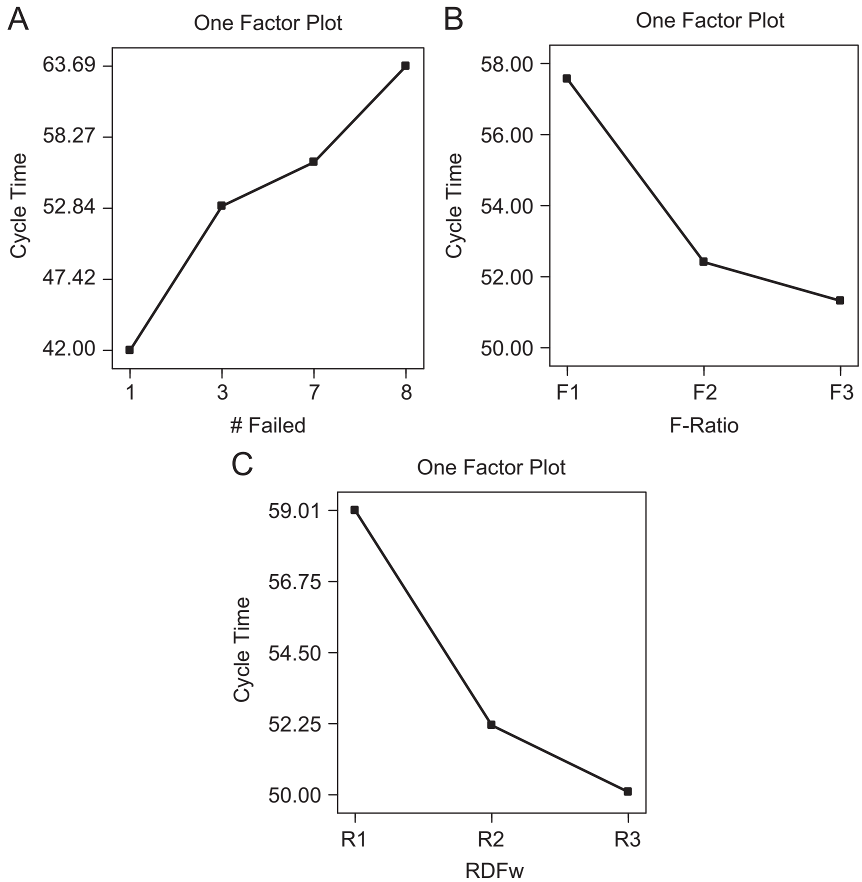

Fig. 7. The cycle time versus: the number of failed groups (A); F-ratio (B); and RDF_W (C).

experiments resulted in negligible computation time for real size problems). In a case where the online procedure cannot be implemented, an offline procedure is applied instead. In this case, all the solutions of all possible (or reasonable) scenarios are generated a-priori and stored in the system. As a consequence, each time a robot's state changes, the a-priori generated solution is retrieved from the database and implemented. Clearly, the number of possible scenarios depends on the number of robots that can break down simultaneously. For example, when two out of sixty-five robots can break down simultaneously, the total number of failure scenarios is given by $\binom{65}{1} + \binom{65}{2} = 2,145$. This number should be kept to a relatively small value to enable the implementation of the offline procedure. Another reasonable scenario which may occur in such a stochastic system is the failure of a robot while another robot is down (rather than a simultaneous fall of two robots). In order to avoid reallocation of 'working' groups (as may happen when applying the above solution for two failed robots), one should generate the solutions of such scenarios in the offline procedure.

The implementation process of the proposed procedure is summarized in Fig. 8. As the system's state changes, two options arise. If a failed robot is up again, the original allocation is resumed, regardless of whether there are currently other failed robots. The reason relies on our assumption that the initial matrix represents the best work allocation when all robots are up, and this way, the work allocation of the initial matrix will be resumed each time the failed robots will be repaired. In case the change is a result of a newly failed robot, we either use the MILP formulation to generate a backup solution (online procedure) or retrieve the solution from a set of a-priori generated and stored solutions (offline procedure). In case such a solution does not exist, the groups of the newly failed robot are allocated to the MR station.

5.2. Performance evaluation

Although the proposed formulation can be solved to optimality, obtaining a global optimum is not guaranteed due to the stochastic nature of the failures and repairs process, which is not taken into account by the MILP formulation. Consequently, the performance of the proposed procedure should be validated in a real-world stochastic environment. To this end, two variations of the proposed procedure are compared with various online heuristic allocation rules, by relying on two performance measures, *cycle time* and *quality*, while the latter is mainly

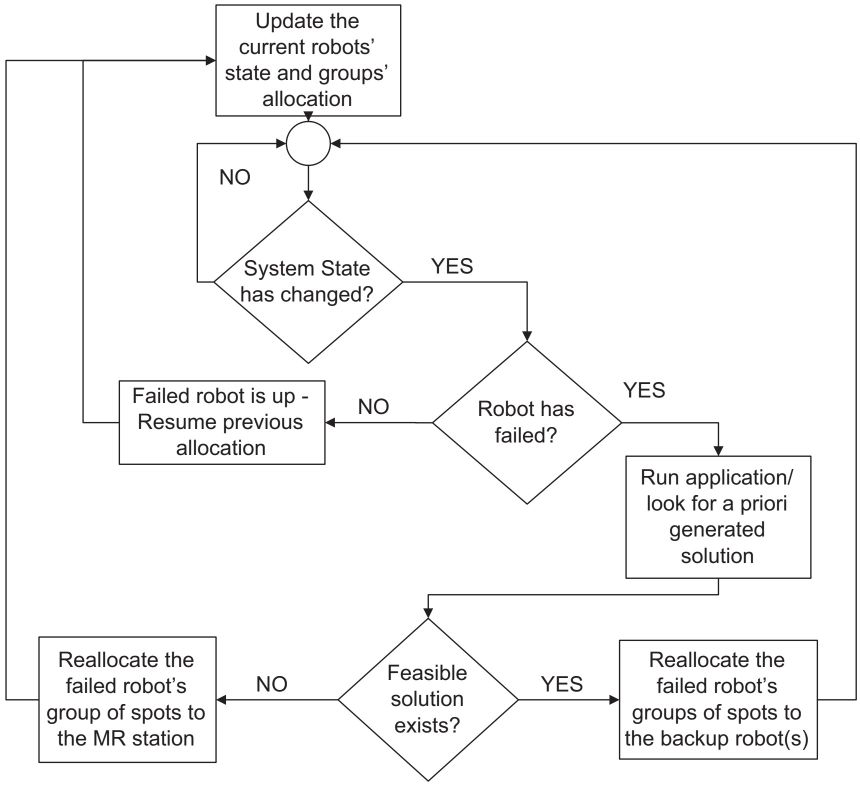

Fig. 8. Backup solution procedure: practical considerations.

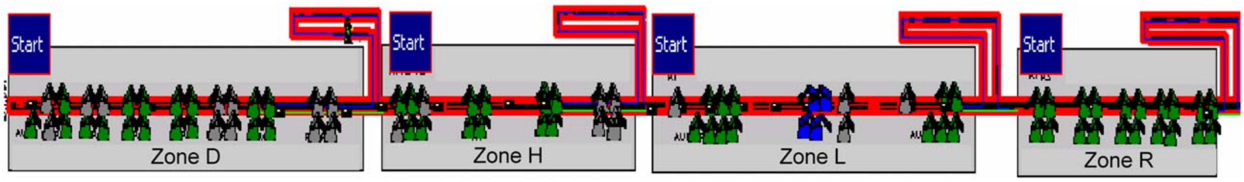

Fig. 9. Body-shop illustration.

affected by the manual welding backup. A discrete event simulation based on a real sized body-shop, (implemented via Arena 5.0 software) is utilized for the validation process.

The body-shop, which is used in this set of experiments, is depicted in Fig. 9. It consists of four zones, each of which contains several stations and one MR station at the end of the zone. In each station several robots are working simultaneously. This production environment is similar to automotive plants founded in the 90's across the US and Europe. The time-between-failure and the time-to-repair are assumed to follow the exponential distribution. This assumption is quite common in the literature, however, one should note that some cases the time to repair may not follow the exponential distribution. The mean values (MTBF and MTTR) are based on a real dataset. Among the 65 robots located on the line, we assume that only 20 robots are prone to failure, while at the maximum two robots can fail simultaneously. Further support for this assumption was obtained by running a simulation of the line with the actual MTBF and MTTR that resulted with a negligible number of scenarios with more than two failed robots simultaneously. The problem is tested on a relatively high redundancy and medium flexibility environment. The MR station is utilized in cases where no backup solution exists. Note that in order to avoid situations of line stoppage, we assume that the MR is always feasible as a last resort when no robotic backup exists, regardless of the precedence constraints.

The solution approaches compared in this experimentation are described as follows.

1 Backup by MR only—this solution approach characterizes a common default situation where no backup strategy is applied. When a failure occurs, the MR station, which is located at the end of each zone, backup all the spots that were not performed as planned.
2 MRB Formulation based algorithm—the optimal MRB formulation is applied, according to the implementation

scenarios that were described in Fig. 8. If no solution has been found, the groups are allocated to the MR station.

3 MRB downstream formulation based algorithm (MRBD)—the MRBD is a variation of the MRB formulation, in which a backup robot is searched for only among downstream stations. If no solution downstream has been found, the groups are allocated to the MR station. The motivation behind this approach is to decrease the use of manual stations which have negative effect on the product's quality, as well as the cycle time (downstream stations are preferred since they can backup the entire failed groups since the time of the failure. In comparison, upstream backup requires a manual backup for all the bodies that are between the backup station and the failed station at the time of the failure).

4 Heuristic allocation rules—the following heuristics are state-dependent in nature, by which the backup robots are chosen online. Each time a robot fails, a feasible set of backup robots (candidates) is obtained taking into consideration the precedence constraint and the CM. The reallocation of groups is performed in a descending order of process times, namely, the group with the highest process time is reallocated first, the one with the second highest time is reallocated next, etc. If the set of feasible robots is empty, the groups of spots will be sent to the MR station. The backup robot is selected based on one of the following rules:

4.1 Nearest capable robot (*NCR*)—the nearest capable robot is selected to perform the failing groups of spots. This heuristic approach searches first for a downstream candidate and only then, if such a robot does not exist, it searches for an upstream candidate.

4.2 Most reliable/least loaded robot (*RLR*)—this rule combines two characteristics of each robot: its reliability, defined by its MTBF, and its current load. Clearly, a robot with higher MTBF and lower current load is preferred for serving as a backup robot. The weighted sum of these measures, RLR_r, is calculated for each capable robot, r:

$$RLR_r = \alpha_1 \cdot \left[\frac{MTBF_r}{\sum_{r \in RC_i(t)} MTBF_r}\right] + \alpha_2 \cdot \left[\frac{\frac{1}{p_r(t)}}{\sum_{r \in RC_i(t)} \frac{1}{p_r(t)}}\right], \quad \forall r \in RC_i(t),\ \forall i \in I^F$$

where $RC_i(t)$ is the set of robots capable of performing group of spots i at time t; $MTBF_r$ is the mean time between failures of robot r; $p_r(t)$ is the process time already allocated to robot r at time t; and α_1 (α_2) is the weight of the reliability measure (load measure). Note that the values of the two measures are normalized between zero and one for scaling purposes. The closest candidate serves as a tie breaker in this rule. In this experiment three rule combinations were chosen, defined by three different weights combinations: $(\alpha_1,\alpha_2) = (0.5, 0.5)$, (1,0) or (0,1).

4.3 Random capable robot (*RCR*)—the backup robot is randomly chosen out of the set of possible candidates $RC_i(t)$.

As noted above, the two performance measures considered here are the cycle time and the quality. The quality measure is associated with the percentage of groups of failed robots that are backed up in the MR station. The reason is that the quality of spots performed by robots is much higher than those that were performed manually, and hence, one prefers to minimize the use of the MR stations. Moreover, the MR also has a negative effect on the cycle time, since the performance time in the manual station is about three times higher than the robotic time.

5.3. Simulation results

A comparison of the commonly implemented backup policy ("MR only"), the MRB, the MRBD and the five proposed heuristics (NRC, three variations of RLR and RCR) involves in total eight different solution approaches for the reallocation problem. A large scale discrete event simulation based on a real body-shop was implemented and analyzed via Arena Software 5.0. Every configuration was simulated by running 50 replications, each of which of two eight-hour shifts and a warm-up period of 2 hours (statistics were not collected during this period). Since the cycle time is around 1 minute, each run consists of approximately 1,000 cycles. The number of 50 replications was large enough to guarantee a relatively tight confidence interval, as seen next.

The average cycle time, its standard deviation (among replications) and the corresponding half width 95% confidence level were collected from the simulation study. The number of groups reallocated to the MR stations' and particularly the percentage of this number with respect to the total number of reallocated groups, were collected and used to define the quality measure of the line. The simulation results are presented in Table 2. One can see that the MRBD (based on the proposed MILP formulation where only downstream backup is allowed) outperforms all other methods in both the cycle time and the quality performance measures. These results are statistically significant with a confidence level of 95%. In particular, the MRBD provides an average cycle time of 56.69 s versus a cycle time of 63.21 s of the default situation—a reduction of 10.3%. In addition, only 3.2% of the groups which require backup are allocated to the MR station.

The MRB is the second best with regard to the cycle time performance, however, significantly inferior to MRBD in this measure. Moreover, it suffers from low quality grade due to an extensive usage of the MR station for backup (49.7%). This is due to the use of upstream backup robots by the MRB approach. In this case, all the groups of spots located between the backup robot and the failed robot at the moment of failure are reallocated to the MR station. As for the quality measure, we can see that the NCR approach is the second best with only 6.3% of the backup groups being allocated to the MR station.

Table 2
Results summary of the experimentation.

	MR only	MRB	MRBD	NCR	RLR (0.5,0.5)	RLR (1,0)	RLR (0,1)	RCR
Cycle time								
Average cycle time	63.21	57.9	56.69	58.32	58.3	58.16	58.34	60
STD cycle time	1.414	1.131	1.202	0.955	1.025	1.025	1.061	1.096
Cycle time half width (95%)	0.4	0.32	0.34	0.27	0.29	0.29	0.3	0.31
Failure data								
Groups MR counter	487.5	260.9	16.3	34.2	144.76	86.9	170.46	183.4
Groups backup counter	0	264.1	498.7	506.5	382.14	453.7	357.6	328.6
MR Groups (%)	100.0	49.7	3.2	6.3	27.5	16.1	32.3	35.8
Backup Groups (%)	0.0	50.3	96.8	93.7	72.5	83.9	67.7	64.2

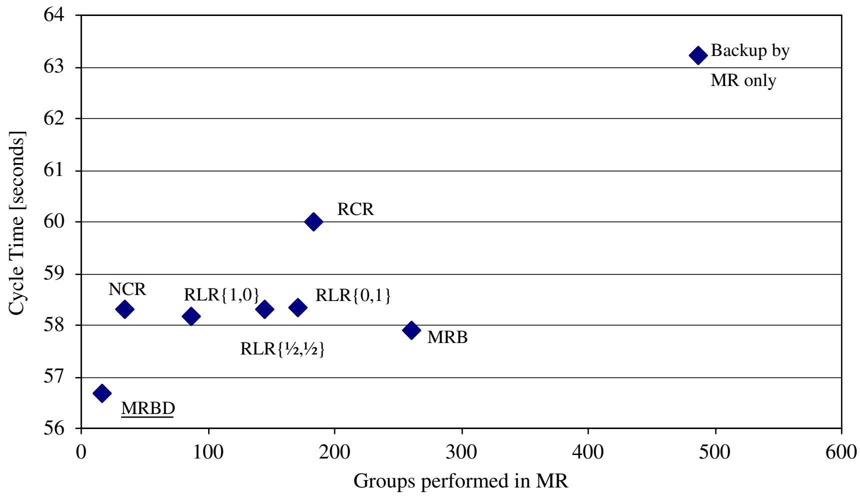

Fig. 10. Cycle time versus quality measure.

This result is not surprising since in this approach we first look for downstream backup and only in cases where such a backup is infeasible, an upstream backup is adopted.

Regarding the RLR rules one can see that the cycle time performance of these three rules is quite comparable, yet, allocating groups to robots with lower MTBF (RLR(1,0)) yields a better quality measure than the other two combinations. Note that the effect of the MR station on the cycle time results in a higher cycle time even when allocating groups to the least loaded robots (RLR(0,1)). As could be expected, the RCR is the worst heuristic both in terms of its cycle time and its quality measure.

The comparative results are also illustrated in Fig. 10, which shows the values of the two performance measures for each allocation method. It is evidently seen that all methods are dominated by the MRBD. In addition, one can see that the commonly implemented backup policy, where backup is performed only manually, is fully dominated by all the other policies, and in particular by the MRBD. Hence, we recommend using the MRBD as the backup approach.

6. Summary and concluding remarks

In this paper we consider a practical spot welding reallocation problem due to robots' failures. Two MILP formulations have been suggested to solve the problem where a single robot or multiple robots are chosen as the backup robot(s). Note that the number of integer variables is relatively small due to the fact that only the groups of spots of the failed robots are considered as decision variables. Consequently, relatively large real-sized problems can be solved via the proposed formulations. Moreover, although the proposed mathematical model was designed for a deterministic environment, which does not take into account future failures and repairs of robots, a slight variation of it (the MRBD policy) has been found to dominate various other backup policies under stochastic conditions.

Future research may include a generalization of the proposed approach to support a mixed-model environment; a dynamic reallocation model—in order to increase the system's robustness; and a comparison between the suggested approaches and other heuristic allocation rules.

Another possible direction may be associated with developing a solution approach which allows group splitting. In this case, a development of easy to use welding time estimation tools will be needed.

References

Amen, M., 2000. Heuristic methods for cost-oriented assembly line balancing: a survey. International Journal on Production Economics 68, 1–14.

Baybars, I., 1986. A survey of exact algorithm for the simple assembly line balancing problem. Management Science 32, 909–932.

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

Boysen, N., Fliedner, M., Scholl, A., 2008. Assembly line balancing: which model to use when? International Journal on Production Economics 111, 509–528.

Bukchin, J., Rubinovitz, J., 2003. A weighted approach for assembly line design with station paralleling and equipment selection. IIE Transactions 35, 73–85.

Bukchin, J., Tzur, M., 2000. Design of flexible assembly line to minimize equipment cost. IIE Transactions 32, 585–598.

Ghosh, S., Gagnon, R.J., 1989. A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. International Journal of Production Research 27, 637–670.

Graves, S.C., Holmes Redfield, C., 1988. Equipment selection and task assignment for multiproduct assembly system design. The International Journal of Flexible Manufacturing Systems 1, 31–50.

Karp, R.M., 1972. Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (Eds.), Complexity of Computer Computation. Plenum Press, New York, pp. 85–103.

Moon, D.H., Cho, H.I., Kim, H.S., Sunwoo, H., Jung, J.Y., 2006. A case study of the body shop design in an automotive factory using 3D simulation. International Journal of Production Research 44, 4121–4135.

Noh, S.D., Hong, S.W., Kim, D.Y., Sohn, S.Y., Hahn, H.S., 2001. Virtual manufacturing for an automotive company (II)—Construction and operation of a virtual body shop. IE Interface 14, 127–133.

Rubinovitz, J., Bukchin, J., 1993. RALB—a heuristic algorithm for design and balancing of robotic assembly lines. Annals of the CIRP 42, 497–500.

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

Spieckermann, S., Gutenschwager, K., Heinzel, H., Vob, H., 2000. Simulation-based optimization in the automotive industry—A case study on body shop design. Simulation 75, 276–286.