Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 63 (2017) 459 – 464

www.elsevier.com/locate/procedia

50th CIRP Conference on Manufacturing Systems (CIRP-CMS 2017)

Scheduling and operator control in reconfigurable assembly systems

Dávid Gyulai[a,b,*], Botond Kádár[a], László Monostori[a,b]

[a]*Fraunhofer Project Center at Institute of Computer Science and Control, Hungarian Academy of Sciences, Kende str. 13-17, H-1111 Budapest, Hungary*
[b]*Department of Manufacturing Science and Engineering, Budapest University of Technology and Economics, Egry J. str. 1, H-1111 Budapest, Hungary*

[*] Corresponding author. Tel.: +36-1-279-6181; fax: +36-1-466-7503. *E-mail address:* david.gyulai@sztaki.mta.hu

Abstract

Pushed by the recent market trends, companies need to adapt to changeable demands, regarding both mix and volume, in order to keep their competitiveness. Modular and reconfigurable assembly systems offer an efficient solution to these changes, providing economies of scale and also economies of scope. In the previous works of the authors, novel methods were presented to solve strategic level system configuration, and tactical mid-term production planning problems related to modular, reconfigurable assembly systems. The paper relies on these results, and aims at extending the previously proposed planning hierarchy on the short-term, daily production scheduling. The objective is to minimize the total operator headcount, considering the production lot sizes calculated on a higher, planning level on a working shift basis. The analyzed scheduling problem requires novel models, as important constraints in the scheduling problem are the reconfigurations consuming time as well as resources. In the paper, constraint programming and metaheuristics models are formulated and compared, resulting in production schedules that specify the production sequences, and the operator allocations. Conclusively, the operator controls can be also obtained from the results, specifying a work plan and tasks for a given operator within a working shift. The proposed methods are compared by using real industrial problem instances.

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems
Keywords: reconfiguration; scheduling; assembly

1. Introduction and motivation

The greatest recent challenge in production management is to match production capacities with the market conditions, characterized by increasing complexity in product variety, as well as diversity in volume. This leads to the fragmentation of orders that are to be handled by careful production planning in order to keep the internal efficiency of the company at a desired level, and stay competitive in the market. Reconfigurable production systems provide a cost-efficient option to match production with fragmented order stream, by offering changeable structure and scalable capacity. Although their efficiency is proven for years now, their industrial application requires special production planning and control approaches to utilize their structural and technological advantages. These approaches must consider the ever changing structure of the applied reconfigurable system's structure, in order to determine proper production plans and assign orders to capacities while keeping the target level of the production performance indicators. In the paper, a two-level production planning and control methodology is proposed to calculate cost-optimal production plans and the corresponding schedules for modular reconfigurable assembly systems.

1.1. Modular reconfigurable assembly systems

In product variety management, changeability of the production systems is a key concept towards efficient synchronization of production processes and customer orders' stream [1]. Changeability is an umbrella concept, encompassing key enablers, among which modularity plays an important role both on the logical and the physical system level. On the latter, the concept stands for the application of so-called *plug and produce* production resources with standardized design and interfaces, as well as with the capability of autonomous operation [2]. Focusing on the assembly processes, modular configuration enables organizations to adjust the physical structure of the system to the assembly processes with low effort considering both time and resources [3–5]. Besides, in planning and control of assembly systems, balancing the operators' workload is of crucial importance to keep the efficiency [6]. Though the literature of reconfigurable production and assembly systems is rather extended, there are a few papers only with the special focus on the production planning and scheduling of these systems [7–9]. Among this limited set of papers, fast reconfigurable assembly systems with modular resource constraints in planning and scheduling are not considered, therefore, the paper and the presented research is aimed at filling this gap by introducing a

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems
doi:10.1016/j.procir.2017.03.082

two level capacity management framework for these systems.

1.2. Operation of modular assembly systems

In the paper, a modular, reconfigurable assembly system is under investigation, which consists of lightweight, *plug and produce* assembly workstations (modules). Each module is dedicated to a single assembly process, and has standardized design including standard connectors and docking interfaces. The modules have a mobile, lightweight frame design enabling fast, short term reconfigurations. They are equipped with assembly tools that can be adjusted to perform assembly processes with different parameters (e.g. screwing torque, screw size etc.). Each of the products assembled in the system is supposed to have assembly tasks that can be performed by applying the standard modules. Therefore, the assembly process of a certain product can be split up into a sequence of standardized assembly tasks (e.g. screwing, pressing) that can be matched with the sequence of the corresponding standard assembly modules. The lines are configured manually on the shop-floor by operators, so as the mobile workstations are placed sequentially according to the successive assembly operations. The configuration is always performed based on the product type to be assembled, and the lines are reconfigured when the assembled product type is changed. The simplified operation (reconfiguration cycle) of the system is the following:

- Configuration: First, the assembly line is built-up by means of the standard modules (which are required by the actual product), by moving them next to each other according to the assembly process steps.
- Setup: The operator performs the necessary setup tasks, e.g., plugs in the air connectors, and places the necessary fixtures on the modules. The operator prepares the necessary parts required by the given assembly processes.
- Assembly: The operator assembles the products in the required volume.
- Deconfiguration: After an assembly process is finished, the operator dismantles the lines, by moving back the excess workstations to the resource pool.

The above described dynamically changing system structure enables flexible production —especially regarding the mix of products assembled—, however, it also requires flexibility in the human workforce, to be capable of performing the reconfigurations as well as the assembly processes. On the operational level of the production planning hierarchy, flexibility in human workforce means that the operators can be assigned to different tasks within their working time (production shift). Technically, this means that each operator is assigned to multiple tasks to perform within the same production shift, and the operator changes task once he/she performed the previous one. The operational level scheduling in this case stands for the operator-task assignments including the starting times of the tasks. In the following sections, the formal definition of the problem in question is provided, applying the notation summarized in Table 1. The input data of the scheduling is provided by the solution of the higher level production planning process, specifying the assembly tasks to be performed within a given time period $t \in T$, therefore, the production planning model and its solution are introduced first.

Table 1. Notation applied in the paper

	Sets
T	set of production time periods
P	set of products
H	set of operator headcounts
N	set of orders
J	set of modules
L	set of lines
	Parameters
t^w	length of a planning period
t_p^s	setup time of product p
t_p^p	total manual processing time of product p
$o_p^{\max}$	maximum operator headcount of product p
r_{jp}	required number of modules from type j by product p
t_{ph}	cycle time of product p when assembled by h operators
c^{op}	cost of an operator per period
q_j	amount of modules from type j
c^h	inventory holding cost [cost/part/period]
c^l	late delivery cost [cost/part/period]
c_{nt}	deviation cost of order n if executed in period t
v_n	volume of order n [pcs.]
t_n^d	due date of order n
p_n	product of order n
v_p^{min}	minimal lot size of prodct p
	Variables
x_{ntlh}	assemble order n in period t and line l with h operators
r_{jlt}	number of modules from type j required at line l in period t
O	total headcount of operators
t_n^{START}	execution start time of task n
t_n^{END}	execution end time of task n

1.3. Production planning problem

In the production planing model, the objective is to determine the production lot sizes x_{ntlh} by matching the available capacities (human and machine) with the customer demands. The planning horizon T is divided into equal length time buckets $t \in T$, and a given set of orders $n \in N$ corresponding to products $p \in P$ need to be completed. The assembly processes are performed by applying $j \in J$ different module types, each type is capable of performing a single process type. The amount of modules from each type j is limited by the resource pool q_j. It is assumed, that the number of simultaneously operating reconfigurable lines is limited along the horizon by introducing the set of lines $l \in L$. These lines are "virtual", as they have no static parts but only composed of reconfigurable modules, however, it is supposed that they are placed on a finite set of segments on the shop floor, and each line occupies a single segment. This assumption is required to manage the machine resources in the production planning model, as the module-line assignment can be constrained in this way. Similarly to the modules, the human resource requirements are also constrained in the production planning model by introducing a set of headcounts $h \in H$ that can be applied to assemble a given product type. In the analyzed problem, skills are not considered, thus an operator can perform any assembly task. Based on the above assumptions, the production planning model is specified as follows. The production lot executions are to be determined with the binary decision variables x_{ntlh}, specifying if order n is executed in period t at line l by the headcount of operators h. Each order n is associated with a product type p specified by p_n, the order volume v_n and a due date t_n^{d}. The parameters c^{h} and c^{l} respectively express that both early and late execution of the orders are penalized with extra costs, with the following formula:

$$c_{nt} = \begin{cases} c^{\mathrm{h}} v_n (t_n^{\mathrm{d}} - t) & \text{if } t < t_n^{\mathrm{d}} \\ c^{\mathrm{l}} v_n (t - t_n^{\mathrm{d}}) & \text{otherwise} \end{cases} \tag{1}$$

The products are characterized with their total manual processing time t_p^{p}, setup time t_p^{s}, minimal economical lot size v_p^{min} (from reconfiguration perspective) and the number of modules r_{jp} required by type j. The objective of the planning is to minimize the cost that is the sum of operator costs c^{op} per periods and the deviation costs c_{nt}.

1.4. Scheduling and operator control problem

As scheduling corresponds to a lower, execution level, its time horizon is shorter than the one of planning. In this case, the scheduling horizon is a single time bucket $t \in T$ with the length of t^w, thus an individual scheduling problem instance can be defined for each time period of production planning. The main input parameters of scheduling are the lot sizes x_{ntlh} (decision variables of the planning model), specifying the assembly tasks, the corresponding operator headcount and assembly lines. The objective of production scheduling task is to minimize the total headcount of operators O working in period t, by calculating the execution start time t_n^{START}, and end time t_n^{END} corresponding to a task n assembled in t. A proper schedule means that the task execution times are distributed over the period enabling the operators to switch between the lines they are working at, when the executed task is finished. The applied resolution of the scheduling horizon is much higher (e.g. minutes) than that of the planning, as the horizon length and problem size allow it. One can distinguish human and machine resources in the scheduling problem, constraining the solution in a different way. As for the machines, a single virtual line L_n and the assigned assembly modules —determined by the planning model— are capable of processing a single task n at any point of time (disjunctive resource constraint). Besides, as many operators O_n need to be assigned to a task n, that is specified by the solution of production planning with the parameter h.

2. Capacity management framework

Based on the above problem specifications, one can identify that a two-stage planning and scheduling problem is to be solved, in which the solution of the higher level planning problem provides the input of the lower level scheduling. While the production planning is responsible for matching the internal capacities with the customer orders, the lower level production schedule specifies the execution times and minimizes the headcount of operators within a time period. In order to solve the overall problem, a two-level capacity management framework

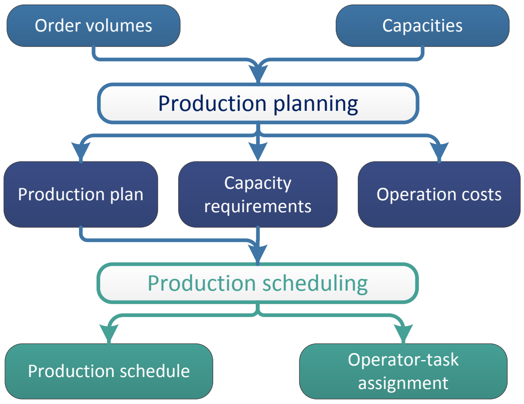

Fig. 1. Decision hierarchy of the applied capacity management framework

is proposed, consisting of production planning and scheduling stages (Figure 1).

2.1. Production planning and scheduling models

The production planning model is formalized as an integer programming model as it follows.

$$\text{minimize} \sum_{l \in L} \sum_{t \in T} \sum_{h \in H} \sum_{n \in N} x_{ntlh} (c^{\mathrm{op}} h + c_{nt}) \tag{2}$$

$$r_{jlt} \geq r_{jp_n} x_{ntlh} \qquad \forall l \in L, t \in T, j \in J, n \in N, h \in H \tag{3}$$

$$\sum_{l \in L} r_{jlt} \leq q_j \qquad \forall t \in T, j \in J \tag{4}$$

$$\sum_{\substack{n \in N \\ p_n = p}} \sum_{h \in H} x_{ntlh} (t_p^{\mathrm{s}} + t_{ph} v_n) \leq t^w \qquad \forall l \in L, t \in T \tag{5}$$

$$\sum_{\substack{n \in N \\ p_n = p}} \sum_{h \in H} x_{ntlh} v_n \geq v_p^{min} \qquad \forall l \in L, t \in T \tag{6}$$

$$\sum_{h \in H} x_{ntlh} \leq 1 \qquad \forall l \in L, t \in T, n \in N \tag{7}$$

$$\sum_{t \in T} \sum_{l \in L} \sum_{h \in H} x_{ntlh} \geq 1 \qquad \forall n \in N \tag{8}$$

$$x_{ntlh} \in [0, 1] \qquad \forall n \in N, l \in L, t \in T, h \in H \tag{9}$$

The objective function (2) minimizes the overall costs of production. Constraint (3) defines the minimal amount of assembly modules to be assigned to line l within a period t, while the total number of modules cannot be exceeded (4). Constraint (5) states that the total amount of processing and setup times of the tasks must be less than the length of the time period t^w, for each line l. Reconfigurations are economical only if applied lot sizes are greater than the minimal quantity as constrained by (6). The last constraints state that only a single operator headcount h can be applied for the execution of each task (7), and each order need to be fulfilled (8), while (9) express that the decision variables x_{ntlh} are boolean type.

The production planning model introduced above is the modified version of the model, presented by the authors in a preceding publication [10]. In the previous version, the headcount of operators was determined on the production planning level, therefore, its solution cannot be applied as the input of the scheduling model to minimize the total headcount with the scheduling of the tasks. Therefore, the decision variable of the planning model was modified to determine the headcount on a task basis, instead of a period basis. This modification requires some pre-calculations, to define the applicable headcount scenarios $h \in H$ for the different tasks, and related headcount-dependent processing times t_{ph}.

The applicable operator headcount of the products' assembly processes is bounded by both the required number of modules r_{jp} and the processing times of the different elementary assembly operations. The resultant maximal operator headcount is the minimum of these two values (10). On the one hand, the operator headcount cannot exceed the number of modules when assembling a product. On the other hand, the operator headcount is also limited by the assembly operations' processing times: if more operators are assembling a given product type p, the resultant cycle time is the linear function of the operator headcount. In the simplest case, one can expect half cycle time for a product when it is assembled by two operators instead of one. This linear correlation is valid until a certain operator headcount is reached, as the resultant cycle time cannot be higher than the longest elementary operation time t^{op}_{pk}, where k is an assembly operation of product p that has $k \in K$ operations in total. The maximum operator headcount in this case is the nearest lower integer of the fraction of total processing time t_{ph} and the longest operation time $\max_{k \in K} t_{pk}$.

$$o^{\max}_p = \min\left(\sum_{j \in J} r_{jp}; \left\lfloor \frac{t^{\mathrm{p}}_p}{\max_{k \in K} t^{\mathrm{op}}_{pk}} \right\rfloor\right) \qquad \forall p \in P \tag{10}$$

As stated above, the assembly cycle times are inversely proportional with the operator headcount. If one would represent the human capacity constraints in a mathematical model, the following equation would needed.

$$\sum_{\substack{n \in N \\ p_n = p}} x_{ntl} \left(\frac{t^{\mathrm{p}}_p v_n}{h_n}\right) \leq t^w \qquad \forall l \in L, t \in T \tag{11}$$

where h_n is a decision variable, expressing the headcount of operators completing the assembly tasks of order n, and x_{ntl} binary variable determines if order n is processed on line l in period t. As it is seen, the fraction term with the decision variable in the denominator would lead to a non-linear model, which is avoidable in this case. Therefore, in order to keep the linearity of the planning model, a new decision variable x_{ntlh} with and additional dimension h is proposed in the planning model instead of x_{ntl}. The above relations are valid only in case of approximated line balances, when the structure of the line as well as the operator task assignments are unknown. Otherwise, if line balances of different operators headcount scenarios are known a-priori, the headcount-dependent processing times t_{ph} can be replaced by the values given by the different line balances. Therefore, the above pre-calculations are needed to be performed for each product type $p \in P$ and possible operator headcount $h \in H$ to calculate the values of t_{ph}. Using the formula (10), one can calculate the set of possible operator headcounts: $H = \{1 \ldots h^{\max}\} \;\; | \;\; h^{\max} = \max_{p \in P} o^{\max}_p$.

Performing the above modifications on the model and calculating the operator-dependent task times and possible headcounts, the mathematical programming model of the considered scheduling problem can be formulated as it follows:

$$\text{minimize } O \tag{12}$$

$$t^{START}_n, t^{END}_n \in \left\{t^s_p \ldots t^w\right\} \quad | \quad p_n = p \qquad \forall n \in N \tag{13}$$

$$\left(t^{END}_m \leq t^{START}_n\right) \vee \left(t^{END}_n \leq t^{START}_m\right) \vee (L_n <> L_m) \qquad \forall n \neq m \tag{14}$$

$$\sum_{n:\left(t^{START}_n \leq t\right) \wedge \left(t^{END}_n > t\right)} O_n \leq O \tag{15}$$

The objective function (12) states that the total headcount of operators working in the period is to be minimized. The first constraint (13) defines that the execution start t^{START}_n and t^{END}_n times of task n (also considering the setup time of the assembled product) are bounded by the duration of a working shift. The second constraint (14) states that only a single product type can be assembled on any given virtual line $l \in L$ at any point of time. The last constraint (15) specifies that the total operator headcount must be greater or equal to the sum of operator headcounts assigned to the executed tasks at any point of time. In (15), the headcount O_n of operators assigned to task n is defined as $O_n = \sum_{h \in H} \sum_{l \in L} x_{ntlh}$, if $t \in T$ is the time period of the scheduling problem to be solved.

2.2. Solution with constraint programming

Production scheduling problems —similar to the presented one in Section 2.1— are often solved by constraint programming (CP) techniques, enabling to find feasible schedules in a reasonable time. The strength of constraint programming relies in the high level, descriptive modeling approach, and the efficient handling of various constraints even in large scale problem instances. Constraint programming has two core elements: a set of predefined constraint types (constraint store) and a built-around programming language to instantiate and combine the constraints [11]. In practice, CP solvers combine constraint reasoning and non-deterministic search approaches to find the solution for a specific problem [12]. Constraint reasoning involves various filtering steps for domain reduction, in order to consider and satisfy multiple constraints that share common variables, this procedure is called constraint propagation [13]. For scheduling problems, constraint programming solvers offer various domain-specific filtering algorithms, called constraint propagators.

The scheduling problem —introduced in the previous section— can be solved by using the cumulative and disjunctive resource propagators. Cumulative resources are represented by their capacity, and the tasks need to be scheduled so as their consumption of the cumulative resources cannot exceed their capacity C at any point of time. Therefore, the operators (15) in the formulated CP model are represented as cumulative resources of a single type, and their capacity is exactly the objective function O of the model. The second, called disjunctive resource propagator is a special cumulative resource, whose capacity is $C = 1$. In the considered scheduling problem this means that any two tasks assigned to the same line $l \in L$ cannot be scheduled so as their executions overlap in time (14), therefore, lines are disjunctive resources. Concluding the above, one can infer that the formulation of the problem with CP techniques applying cumulative and disjunctive resource propagators is straightforward, however, neither possible stochastic nature of the manual processing times, nor the random events can be handled with this modelling technique.

2.3. Genetic algorithm based solution

For the above reasons, the problem is also solved by genetic algorithm (GA), which is one of the most fundamental approaches to solve stochastic optimization problems. Genetic algorithms are classified as search metaheuristics, belonging to the class of evolutionary algorithms. Applying bio-inspired genetic operators on a set (population) of candidate solutions (individuals), GAs try to improve the solutions and move toward the global optima. As in general GAs cannot be applied for constrained optimization problems, hurt of the constraints in the solutions are mostly penalized in the objective (fitness) function. Generally, genetic algorithms are capable of handling stochastic parameters if one can evaluate a solution considering them, therefore, they can be applied to solve the considered scheduling problem where stochasticity characterize the parameters due to the manual processing times with certain deviations, and other possible random events like scrap products entailing rework. In the paper, we propose a simulation-based method for solution evaluation: the fitness function of a given schedule is determined by executing a discrete-event simulation analysis. This approach allows for the detailed analysis of stochastic parameters, that often characterize manual assembly processes. The greatest benefit of using a simulation model relies in the opportunity of representing the stochasticity of parameters in detail. In each iteration of the GA, simulation experiments are executed to evaluate the fitness of the individuals, therefore, the time consumption of a single experiment is of crucial importance to keep the overall running time of the algorithm on a reasonable level. The simulation applies an automated model building process, enabling the dynamic model creation and realistic handling of resource constraints. [14].

3. Numerical results

In order to evaluate and compare the efficiency of the applied solution methods (CP and GA), a real case study from the automotive industry was selected.

3.1. Description of the production environment

The company under study is a *Tier-1* supplier, producing mechatronics components to several OEMs. The product portfolio is rather diverse, however, the whole set of assembly processes can be clustered in eight main process types, therefore, the processes can be covered by a module set of $|J| = 8$. In the assembly segment, $|P| = 67$ main product types are assembled, and the total yearly volumes of products are diverse. As for the production planning problem, the objective is to calculate the production lot sizes based on the customer order stream and available capacities. The length of the planning horizon is $|T| = 10$ working shifts, and the length of a shift is $t^w = 480$ minutes. The total number of orders to be considered in the analyzed problem instances varies in a range $|N| \in [120, 150]$ for the whole planning horizon T. The available shop-floor space in the assembly segment enables to operate $|L| = 8$ modular assembly lines simultaneously. Calculating the headcount-dependent processing times for each product type p, the maximal headcount of operators and thus the size of their set is $|H| = 10$. As for the scheduling problem, the task is to determine the task execution and end times within the production shifts, considering that the setup times of the products are $t^s_p \in [15, 30]$. Resulting from the production planning level, the average size of a scheduling problem instance is $|N| \in [15, 20]$ within a given time period t. In order to prove the validity of the proposed mathematical models and compare the solutions provided by the two solution methods, eight different test problem instances were solved by both methods. First, the production planning problem is solved, afterwards eight different production periods from the results were selected to solve the production scheduling problem.

3.2. Results with constraint programming

The CP production scheduling model —specified in Section 2.1— was implemented in *FICO Xpress* applying its *Kalis* constraint programming library with a scheduling toolbox. In order to handle the resource constraints properly, the assembly lines $l \in L$ were set as disjunctive, while the operators are cumulative resources with the capacity of O. By default, the constraint solver cannot be set to optimize the production schedule respecting the capacity of resources as an objective function. Therefore, the optimization procedure was solved by an iterative approach with interval halving, where the value of O was adjusted in each iterations. Starting with and arbitrarily large value, the problem was solved in each iteration, and the value of O was halved a solution was found. Otherwise, the headcount was set to the median of the current value and the previous one. In this way, the objective function value converged to the solution, while feasible schedules were calculated for each values. In order to boost the computations, the CP solver ran until a feasible schedule was found. In this way, all problem instances could be solved by CP, calculating the minimal required operator headcount and the corresponding feasible schedule, however, all the parameters in the problem were deterministic as CP solver could not tackle their possible stochasticity.

3.3. Results with genetic algorithm

For this reason, the scheduling problem was also solved by GA, to consider the possible variability of the manual processing times, resulted by the *human factor*. Therefore, the focus was on this effect by setting 10% deviation for the manual processing times with a normal distribution. This could be done in the simulation model of the assembly system, that was also responsible for the evaluation of the solution in each iteration of the GA. In order to get a more realistic solution, each individual (schedule) in the population was evaluated by running the simulation multiple times, simulating different processing times generated with a normal distribution function with 10% deviation by the simulation model.

The schedules were created by the algorithm applying genetic operators. In the GA, the main settings were the probability of crossover and inversion steps, set to 0.8 and 0.2, respectively. The number of iterations was set to 20, and the population sizes were 15. The simulation model of the assembly system was implemented in *Siemens Tecnomatix Plant Simulation*. The resources were represented by objects in the model, each having disjunctive feature enabling to tackle the capacity constraints in the GA-solution.

3.4. Evaluation of the results

In order to evaluate the quality of the solutions and the feasibility of the schedules, the results provided by both methods were executed with the simulation model of the system, representing the 10% deviation of the processing times. In order to represent this stochasticity in the CP scheduling model, and try to calculate feasible schedules with it, the processing times were increased by 10% in the CP, while in the GA, all the evaluations are performed by the simulation model applying the same deviation. The results provided by both methods for the 8 problem instances are included in Table 2. As the results show, the running time of the GA is significantly higher than that of the CP, however, it results in the same objective function values except in SC#1. The GA based solution provides schedules that are feasible in most of the cases, even in case of stochastic processing times, whereas CP fails to provide executable schedules in more cases if parameters are stochastic, although the schedules were calculated with extra capacities. In each cases, the CP could provide a schedule that would be feasible with deterministic parameters, however, lateness occur in the simulation, representing realistic production environment.

Table 2. Scheduling results provided by the CP and GA methods. The first column (SC) indicates the scenario number, $|N|$ is the number of orders to be scheduled. The columns O (superscripted with the method) give the resulted headcount, t and t is the running time in seconds. The last columns t_m are the makespan values (minutes) of the methods, and t_m^{CP} is the calculated whereas t_{mr}^{CP} is the simulated makespan of CP

SC #	$\|N\|$	O^{CP}	O^{GA}	t^{CP}	t^{GA}	t_m^{CP}	t_{mr}^{CP}	t_m^{GA}
1	15	11	12	3	172	471	488	427
2	14	8	8	2	567	469	502	433
3	11	7	7	601	328	476	476	448
4	16	7	7	5	175	475	477	471
5	15	7	7	4	558	480	470	469
6	14	8	8	3	158	477	506	508
7	11	6	6	2	247	470	466	433
8	11	7	7	603	457	457	493	497

4. Conclusion and outlook

In this paper, a novel, two-stage framework was introduced for the capacity management of modular, manually operated assembly systems. On the higher level, the production planning problem was solved in order to determine the production lot sizes and the corresponding operator headcount. On the lower level, the detailed production schedule was determined, specifying the operator-task assignments, as well as the execution start times of the production lots. The formulated scheduling model was solved by constraint programming and genetic algorithm (combined with simulation), and the resulted schedules were executed by a simulation model. Although CP-based schedules satisfy the constraints considering deterministic values, they tend to be infeasible in a realistic environment if processing times are non-deterministic. In contrast, simulation based GA scheduling provides robust schedules against the deviation of the processing times, thus the schedules remain feasible, even though the processing times are stochastic. As for the future work, the authors' plan is the further detailed analysis of simulation and GA based schedules, to determine the robustness of the plans.

Acknowledgment

Work for this paper was supported by the European Commission through the H2020 project EXCELL (http://excell-project.eu) under grant No. 691829; and the Hungarian Scientific Research Fund (OTKA) under Grant No. 113038.

References

[1] ElMaraghy, H., Schuh, G., ElMaraghy, W., Piller, F., Schönsleben, P., Tseng, M., et al. Product variety management. CIRP Annals-Manufacturing Technology 2013;62(2):629–652.
[2] Wiendahl, H.P., ElMaraghy, H.A., Nyhuis, P., Zäh, M.F., Wiendahl, H.H., Duffie, N., et al. Changeable manufacturing-classification, design and operation. CIRP Annals-Manufacturing Technology 2007;56(2):783–809.
[3] ElMaraghy, H., ElMaraghy, W.. Smart adaptable assembly systems. Procedia CIRP 2016;44:4–13.
[4] Hu, S.J., Ko, J., Weyand, L., ElMaraghy, H., Lien, T., Koren, Y., et al. Assembly system design and operations for product variety. CIRP Annals-Manufacturing Technology 2011;60(2):715–733.
[5] Michalos, G., Makris, S., Chryssolouris, G.. The new assembly system paradigm. International Journal of Computer Integrated Manufacturing 2015;28(12):1252–1261.
[6] Michalos, G., Makris, S., Rentzos, L., Chryssolouris, G.. Dynamic job rotation for workload balancing in human based assembly systems. CIRP Journal of Manufacturing Science and Technology 2010;2(3):153–160.
[7] Bruccoleri, M., Nigro, G.L., Perrone, G., Renna, P., La Diega, S.N.. Production planning in reconfigurable enterprises and reconfigurable production systems. CIRP Annals-Manufacturing Technology 2005;54(1):433–436.
[8] Abbasi, M., Houshmand, M.. Production planning of reconfigurable manufacturing systems with stochastic demands using tabu search. International Journal of Manufacturing Technology and Management 2009;17(1-2):125–148.
[9] Abbasi, M., Houshmand, M.. Production planning and performance optimization of reconfigurable manufacturing systems using genetic algorithm. The International Journal of Advanced Manufacturing Technology 2011;54(1-4):373–392.
[10] Gyulai, D., Kádár, B., Kovács, A., Monostori, L.. Capacity management for assembly systems with dedicated and reconfigurable resources. CIRP Annals-Manufacturing Technology 2014;63(1):457–460.
[11] Hentenryck, P.V.. The OPL optimization programming language. MIT Press; 1999.
[12] Hentenryck, P.V., Michel, L.. Constraint-based local search. MIT press; 2009.
[13] Bessiere, C.. Constraint propagation. Foundations of Artificial Intelligence 2006;2:29–83.
[14] Gyulai, D., Vén, Z., Pfeiffer, A., Váncza, J., Monostori, L.. Matching demand and system structure in reconfigurable assembly systems. Procedia CIRP 2012;3:579–584.