CIRP Template v4.0

Contents lists available at SciVerse ScienceDirect

CIRP Annals Manufacturing Technology

Journal homepage: www.elsevier.com/locate/cirp

Decomposition approach to optimal feature-based assembly planning

Csaba Kardos[a,b], András Kovács[a], József Váncza (1)[a,b]

[a] Fraunhofer Project Center PMI, Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary
[b] Department of Manufacturing Science and Engineering, Budapest University of Technology and Economics, Budapest, Hungary

The paper proposes a generic approach to assembly planning where individual tasks, with detailed technological content specified by features, must be combined into an optimal assembly plan subject to technological and geometric constraints. To cope with the complexity and variety of the constraints that refer to the overall assembly process, Benders decomposition is applied. The macro-level master problem looks for the optimal sequencing and resource assignment of the tasks, while sub-problem modules ensure plan feasibility on the micro-level from aspects of technology, fixturing, tooling, and collision. Constraints are also dynamically generated for the master problem. The approach is demonstrated in automotive assembly.

Assembly; Planning; Optimization

1. Introduction

Process planning is the essential act of production engineering which defines ways from the world of design ideas to the reality of production. The ultimate motivation of this research is to propose a novel, *generic model* for Computer-Assisted Process Planning (CAPP). The model should warrant the feasibility and even optimality of process plans, comply with the requirements of all stakeholders responsible for different aspects of plan execution, keep the complexity of the planning process in bay, and support iterative, mixed-initiative problem solving. Hence, a complete and preferably optimal solution is sought which is intuitive, tractable, extendable and also repairable.

The actual application domain is *mechanical assembly* where departing from the CAD model of the product and its parts, the specification of their relations and joints, as well as the description of available resources (tools, fixtures, human or robot operators), an executable *assembly plan* is to be generated which is best according to some criteria. More specifically, in assembly planning a number of various aspects such as product structure and variety [1], assembly technology, fixturing and tooling [2], part stability [3], handling of rigid and elastic parts [4], tolerances and quality, path of movements [5], capabilities and skills of workers, ergonomics [6], as well as costs and setup times [1] have to be considered and consolidated. While engineering tradition and common sense suggests that there could be no single model for incorporating all the above issues, some general principles apply well: (i) *Hierarchical decomposition* elevates sequencing and resource assignment decisions to the level of *macro planning* and refers the handling of other details to *micro planning* [7][8]. (ii) *Feature-based decomposition* helps define elementary tasks and semantic structuring of domain knowledge both for planning [9] and task execution [10].

Assembly planning even on the macro level is still hard since typically a complex net of various types of *constraints* has to be satisfied [11]. The admissible use of resources and ordering of operations mutually depend on each other, hence—as in any CAPP domain—the constraints are *conditional* [8][12]. Micro-level concerns are typically respected as pieces of advice in form of special constraints, giving up this way the completeness and sometimes even the satisfiability of the planning problem. All this requires custom-tailored (re)solution methods [5][12][13]. Determination of process parameters, fixture design, path planning, collision checking and, in general the *validation* of assembly plans requires micro-level activities, either done by specialized modules or manual intervention (eventually supported by advanced visualization [14]). However, if an assembly plan turns out to be non-executable due to an issue detected on the micro level, there must be an automated way back to resume macro planning and avoid the failure.

2. Problem statement

This paper addresses the development of a *hierarchical decomposition* planning methodology that allows the optimization of the *macro-level assembly process plan*, while it guarantees the *micro-level feasibility* of the plan and individual tasks from all relevant engineering aspects. In the proposed decomposition scheme, the macro-level planner is responsible for generating the assembly plan by optimizing the *task sequence* and the *resource assignments*, which are two *interrelated decisions*. A collection of micro-level sub-problem solvers ensure that the planned tasks can be implemented in physical reality by generating constraints on the macro-level plan, both before planning and dynamically, during plan generation (see Fig. 1). Sub-problem analyzers and solvers in the current implementation include *technological feasibility*, *collision detection*, *fixturing*, and *tooling* modules.

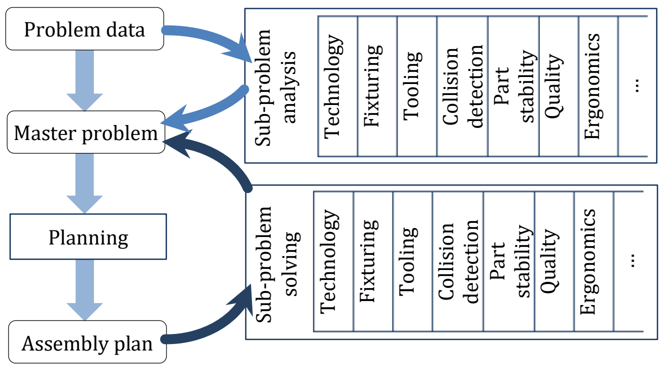

Fig. 1. Benders decomposition for the assembly process planning problem.

The following, typical assumptions are made:

- The set of assembly and auxiliary tasks are given as input.
- Binary (two-handed) monotonous assembly is assumed, i.e., each assembly task joins two parts (or composites) and no interim disassembly is involved.
- The liaison graph of the product is a tree (arbitrary graphs can be handled at the price of moderate extensions).
- Parts are non-deformable.
- Task durations are independent of the task sequence and the resources assigned.

In the application of the generic approach to assembly process planning, there is a set of assembly tasks (and potentially some auxiliary tasks) that must be sequenced and assigned to resources [1][9][10]. The detailed technological content of assembly task t is specified by the assembly feature $F_t : \langle \rho_t, a_t, b_t, \Theta_t, P_t \rangle$, where ρ_t is the feature type (placing, insertion, and screwing are handled in the current implementation), a_t and b_t are the two parts joined by the task, while Θ_t is the homogeneous transformation matrix that defines the motion joining the involved parts. Composite features are allowed (e.g., entering multiple parallel-axis screws in a single screwing task), in which case the parts can be compounds (e.g., multiple identical screws). While a_t and b_t are initially interchangeable, after the assignment of fixtures and tools they can be distinguished as *base* and *moved* parts. For some features, e.g., screwing, this differentiation can be evident. P_t is a vector of numeric feature parameters according to the given feature type. The standard *liaison graph* representation is adopted, in which vertices denote parts, and they are connected by edges labelled by the task joining the two parts (see [1]).

An *auxiliary task t*, e.g., an inspection task, is associated to a single part, potentially after it is joined into a sub-assembly. For the sake of uniform representation of assembly and auxiliary tasks, this is captured by $a_t = b_t$ and a dummy feature type $\rho_t = \emptyset$.

Each task requires an appropriate *fixture* and a *tool* (e.g., screwdriver or gripper) as resources for its execution, whereas efficiency is addressed by minimizing the total processing time, which, given that task durations are fixed, is equivalent to minimizing the resource changeover times. The feasibility of the assembly process plan on the micro-level is verified from the technological, fixturing, tooling, and the collision points of view.

The proposed approach is illustrated on an automotive supercharger assembly consisting of 29 individual parts (some of them belonging to composites), joined in 17 assembly tasks and one auxiliary task. Illustrations in the paper are derived on a sub-assembly of this product, the so-called inlet by-pass sub-assembly, consisting of 12 parts, see Fig. 2.

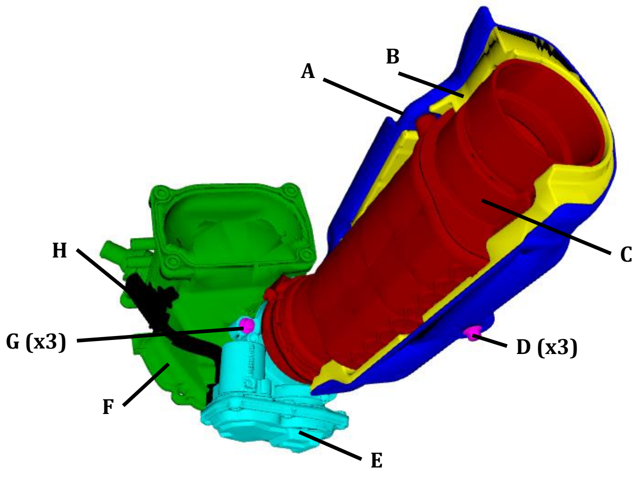

Fig. 2. CAD model of the product in the working example: inlet by-pass sub-assembly of an automotive supercharger.

Table 1. Notation applied in the master problem.

Parameters	
a_t, b_t	Two parts assembled by task *t*
T_{rs}	Set of tasks along the $\overline{rs}$ path in the liaison graph
r_f	Part grasped by fixture *f*
Z_t	Set of candidate tools for task *t*
N_t	Set of forbidden tool/fixture combinations for task *t*
w_r	Weight of part *r*
g_f	Weight limit for fixture *f*
W	Total weight of the assembly
d_f	Changeover time related to fixture *f*
e_z	Changeover time related to tool *z*
Variables	
x_{tp}	Indicates if task *t* is located at position *p* of the plan
φ_{pf}	Indicates if fixture *f* is applied at position *p* of the plan
τ_{pz}	Indicates if tool *z* is applied at position *p* of the plan
y_{tu}	Indicates if task *t* precedes task *u* in the plan
q_{prs}	Indicates if parts *r* and *s* belong to the same sub-assembly after executing the task at position p of the plan
c_p	Changeover time before position *p* of the plan

3. Solution approach

3.1 Benders decomposition framework

The proposed approach exploits the inherent decomposition between the macro-level process planning problem, involving task sequencing and resource assignment decisions, and the micro-level problems related to the detailed implementation of the individual tasks. Computationally, the proposed methodology can be described as a *Benders decomposition* approach [15]: the master problem is a relaxation of the original planning problem, since initially it contains no constraints to enforce micro-level feasibility, e.g., it overlooks the aspect of collision avoidance. Consequently, a solution of the master problem may be infeasible on the micro-level. In such a case, the corresponding sub-problem solver discards this solution and generates a constraint, a so-called *feasibility cut*, which excludes this solution, and potentially other solutions infeasible for a similar reason, from subsequent iterations. The master problem is re-solved with the added cut, and this procedure is iterated until an optimal master solution is found to be feasible also on the micro-level, or infeasibility is proven in the master problem. The approach guarantees that the resulting solution is an optimal solution to the overall planning problem, and hence, the proposed Benders decomposition approach is an *exact* solution procedure.

3.2 Master problem: sequencing and resource assignment

The master problem in the decomposition approach is responsible for task sequencing and resource assignment. It can be formulated as a mixed-integer linear programming (MILP) model as follows, and subsequently, solved by standard MILP packages. The applied notation is summarized in Table 1.

Minimize

$$\sum_p c_p$$

subject to

$$\sum_p x_{tp} = 1 \qquad \forall t \qquad (1)$$

$$\sum_t x_{tp} = 1 \qquad \forall p \qquad (2)$$

$$y_{tu} + \sum_{p'=1}^{p} x_{up'} \le \sum_{p'=1}^{p-1} x_{tp'} + 1 \qquad \forall t \ne u, p \qquad (3)$$

$$q_{prs} \geq \sum_{p'=1}^{p} \sum_{t \in T_{rs}} x_{tp'} - |T_{rs}| + 1 \qquad \forall p, r \neq s \tag{4}$$

$$q_{prs} \leq \left(\sum_{p'=1}^{p} \sum_{t \in T_{rs}} x_{tp'} \right) / |T_{rs}| \qquad \forall p, r \neq s \tag{5}$$

$$\sum_{f} \varphi_{pf} = 1 \qquad \forall p \tag{6}$$

$$\sum_{z} \tau_{pz} = 1 \qquad \forall p \tag{7}$$

$$c_p \geq d_f \left(\varphi_{pf} - \varphi_{(p-1)f} \right) \qquad \forall p, f \tag{8}$$

$$c_p \geq e_z \left(\tau_{pz} - \tau_{(p-1)z} \right) \qquad \forall p, z \tag{9}$$

$$y_{tu} + y_{uv} \leq y_{tv} + 1 \qquad \forall t \neq u \neq v \tag{10}$$

$$q_{0rs} = 0, \quad q_{prr} = 1, \quad q_{prs} = q_{psr} \qquad \forall p, r \neq s \tag{11}$$

$$y_{tt} = 0, \quad \tau_{0z} = 0, \quad \varphi_{0f} = 0 \qquad \forall t, z, f \tag{12}$$

$$x_{tp}, \varphi_{pf}, \tau_{pz}, y_{tu}, q_{prs} \in \{0,1\} \qquad \forall t, u, p, f, z, r, s \tag{13}$$

The objective is minimizing the total changeover time. Constraints (1,2) ensure that the assembly sequence is a permutation of the given tasks. Equality (3) connects the task precedence variables y_{tu} to the task position variables x_{tp}. Constraints (4,5) maintain the connectivity relations between parts by stating that parts r and s are connected if and only if all the tasks along the path $\overline{rs}$ are completed. Equations (6,7) ensure that exactly one fixture and tool is selected for each task. Inequalities (8,9) define the changeover time between subsequent positions as the maximum of the fixture changeover and the tool changeover times. The transitivity of the precedence relations is encoded in the redundant constraint (10). Finally, constraints (11-13) define the variables' ranges and set the variables' values in some specific cases.

These initial constraints are complemented by constraints generated by the sub-problem modules during the problem analysis phase and during search in the frame of the Benders decomposition approach.

3.3 Coordination between master problem and sub-problems

The constraints added to the master problem during search are feasibility cuts, i.e., they encode constraints that all master problem solutions must respect to be feasible on the micro-level. In the simplest case, the feasibility cut is a no-good cut that indicates that an earlier found master solution is infeasible, and hence, future solutions must differ from that. Other, problem-specific cuts can be designed that exclude a larger set of master solutions that are infeasible for similar reasons. Since these cuts are of crucial importance for computational efficiency, a particular goal is to identify such problem-specific cuts.

In addition to feasibility cuts, so-called *optimality cuts* also have a natural application in CAPP. These cuts express that a given master solution, though might be feasible, incurs a higher cost than assumed in the master problem. An example is that the solution of a robot trajectory planning sub-problem for a given task may point out that a planned task is feasible, but it takes more time than its assigned duration in the master problem. Nevertheless, in this paper, the focus is on solving the macro-level planning problem, addressing feasibility on the sub-problems, and hence, this attractive but challenging opportunity is ignored.

In the decomposition scheme, the generic structure of the cut fed back from a sub-problem to the master problem is as follows:

$$\left\langle pred, \quad \left\langle succ_1, \ldots, succ_K \right\rangle, \quad fixt, \quad tool \right\rangle \tag{14}$$

indicating that predecessor task *pred* must either precede at least one of the successor tasks *succ*$_1$, …, *succ*$_K$, or it must not be processed in fixture *fixt*, or it must not be processed using tool *tool*. If the fixture and the tool do not take part in the collision (i.e., *fixt* = ∅ and *tool* = ∅), then the generated cut is the disjunctive precedence constraint as follows:

$$\sum_{k=1}^{K} y_{pred, succ_k} \geq 1 \tag{15}$$

Otherwise, the following set of constraints is added (replacing $\varphi_{p,fixt}$ or $\tau_{p,tool}$ by +1 if *fixt* = ∅ or *tool* = ∅, respectively):

$$\sum_{k=1}^{K} y_{pred, succ_k} \geq x_{pred,p} + \varphi_{p,fixt} + \tau_{p,tool} - 2 \quad \forall p \tag{16}$$

3.4 Sub-problem: technology

The module ensures the technological feasibility of the computed plans by adding classical precedence constraints (i.e., constraints with a single successor task and no fixtures or tools in (14)) to the master problem in the problem analysis phase using a *rule-based* approach. The set of rules are characteristic to the application domain. An example from the currently supported mechanical assembly domain is the rule that parts joined by a screwing feature must be attached first by a placing or an insertion feature.

3.5 Sub-problem: fixturing

The fixturing module is responsible for assigning a valid fixturing option (combination of a fixture device, a grasped part, and an orientation) to each task. It is emphasized that the fixture device can be either a physical fixture characterized by a geometrical model or a conceptual fixture, depending on the relative position of process planning and fixture design in the applied workflow. The fixturing module generates the following constraints for the master problem during problem analysis:

$$\varphi_{pf} + x_{tp} - 1 \leq q_{(p-1)r_f a_t} + q_{(p-1)r_f b_t} \qquad \forall p, t, f \tag{17}$$

$$g_f + \left(1 - \varphi_{pf}\right) W \geq \sum_{s} \left(w_r q_{pr_f s} \right) \qquad \forall p, f \tag{18}$$

Constraint (17) states that a given fixturing option can be used for a given task only if one of the corresponding part sets, a_t or b_t, is already assembled with the part grasped by the fixture. Inequality (18) encodes the fixture weight limit.

3.6 Sub-problem: tooling

Likewise, the tooling module generates constraints in the problem analysis phase to ensure a feasible tool-to-task assignment. It manages tool compatibility data in the form of Z_t and N_t, and adds the following constraints to the master problem:

$$x_{tp} + \tau_{pz} \leq 1 \qquad \forall t, p, z \notin Z_t \tag{19}$$

$$x_{tp} + \tau_{pz} + \varphi_{pf} \leq 2 \qquad \forall t, p, (z, f) \in N_t \tag{20}$$

Constraint (19) excludes all invalid tool assignments, whereas (20) eliminates infeasible combinations of tools and fixtures.

3.7 Sub-problem: collision detection

A central condition of the feasibility of an assembly plan is the ability to execute the tasks without any collision between parts or the resources assigned. Following the approach proposed earlier in [16], the investigation of potential collisions is performed in two steps: (i) for the core movement captured by the assembly feature, and (ii) for the approach of the moved parts and the tool to the region of interest. In both cases, collision detection is performed using the *Flexible Collision Library* (FCL) [17] on the triangle mesh models of the involved objects.

For the motion fully defined in the assembly feature in case (i), potential collisions can be unambiguously identified and captured in the *Extended Liaison Graph* (ELG) corresponding to the assembly configuration at the time of the given task *t*. The ELG contains the parts plus the tool and the fixture as vertices, as well as their liaisons, including also tool-to-part and fixture-to-part contacts, as edges. In case the ELG is not connected, only the component containing the edge of task *t* is considered. Each individual collision can be associated to a pair of vertices in the ELG (the involved two parts, a tool, or a fixture), and it defines a unique path π between these vertices in the tree-structured ELG. Observe that an identical collision is prevented in subsequent iterations of planning if at least one of the edges along π is removed from the ELG, by executing the corresponding task only after *t* (part-to-part edges) or modifying the corresponding resource assignment (fixture/tool-to-part edges). Accordingly, the sub-problem solver generates the feasibility cut for each detected collision in which, using the notation of (14), the predecessor task is the current task *t*, successor tasks are all other tasks along π, while the fixture and tool are present if they are involved in the collision. An illustration from the working example is presented in Fig. 3 for a collision between the tool and part A when executing task 6. Vertices corresponding to objects involved in the collision and the path connecting them are highlighted. The generated cut is $\langle 6, \ \langle 1,2,4,5 \rangle, \ \varnothing, \ Tool \rangle$.

For potential collisions during approach, in case (ii), infeasibility is identified by observing that no collision-free path exists between a remote point and the *near position* defined in the feature. In this case, the generated cut encodes that an *arbitrary edge* must be removed from the ELG.

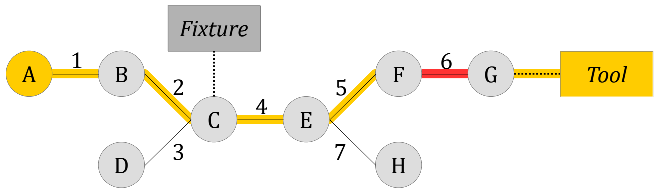

Fig. 3. Extended Liaison Graph representation of a collision.

4. Case study in automotive assembly

The proposed solution approach was implemented in Phyton, using FICO Xpress 8.0 for solving the master problem and FCL as a collision detection engine [17]. The algorithms were validated on different variants of the automotive supercharger assembly problem introduced above. The steps of the assembly process plan for the small-scale working example (12 parts, 7+1 tasks) are displayed in Fig. 4, omitting fixtures and tools. For the original industrial problem (29 parts, 17+1 tasks), the planner required only 2 iterations to find a feasible and optimal assembly plan. In the first iterations, the collision detection module discovered 8 different collisions and generated the corresponding cuts to eliminate these from subsequent iterations. The assembly plan built in the final iteration was found to be feasible by all sub-problem solvers, and hence, it is the optimal solution to the assembly planning problem. The complete solution process took 602 s on a virtual machine corresponding to the computational power of an average PC: 90 s for initializing data (e.g., part geometries), 507 s for solving the master problem in two iterations, and 5 s for the sub-problem solvers.

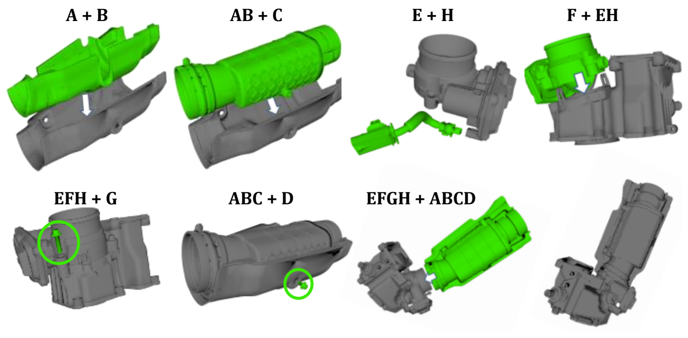

Fig. 4. Assembly plan for the working example. Base parts are displayed in gray, moved parts in green. Fixtures and tools are omitted from the figure.

5. Conclusions

The paper proposed a *Benders decomposition scheme* to assembly process planning, departing from a feature-based task model, and part, fixture and tool geometries. Contrary to other methods, the approach ensures the *feasibility* and *optimality* of the computed plans by the detailed validation of the planned tasks by sub-problem solvers dedicated to assembly technology, fixtures, tools and collisions. Constraints are also generated on the fly to eliminate any infeasibility in an iterative solution process. Hence, planning is performed by a systematic interplay of combinatorial optimization and geometric reasoning. The implemented algorithms proved to be efficient on medium-sized real life mechanical assembly problems from the automotive industry. Future work will concentrate on richer process models (e.g., task durations depending on resource assignment) and on extension to human-robot cooperation [18].

Acknowledgement

This research has been supported by the EU H2020 SYMBIO-TIC No. 637107 and the GINOP-2.3.2-15-2016-00002 grants.

References

[1] Hu SJ, Ko J, Weyand L, ElMaraghy HA, Lien TK, Koren Y, Bley H, Chryssolouris G, Nasr N, Shpitalni M (2011) Assembly system design and operations for product variety. *CIRP Annals - Manufacturing Technology* 60(2):715-733.

[2] Fantoni G, Santochi M, Dini G, Tracht K, Scholz-Reiter B, Fleischer J, Lien TK, Seliger G, Reinhart G, Franke J, Hansen HN, Verl A (2014) Grasping devices and methods in automated production processes. *CIRP Annals - Manufacturing Technology* 63(2):679-701.

[3] Su Q, Lai S, Liu J (2009) Geometric computation based assembly sequencing and evaluating in terms of assembly angle, direction, reorientation, and stability. *Computer-Aided Design* 41(7):479-489.

[4] Jiménez P (2012) Survey on model-based manipulation planning of deformable objects. *Robotics and Computer-Integrated Manufacturing* 28(2):154-163.

[5] Morato C, Kaipa KN, Gupta SK (2013) Improving assembly precedence constraint generation by utilizing motion planning and part interaction clusters. *Computer-Aided Design* 45(11):1349-1364.

[6] Ding Z, Hon B (2013) Constraints analysis and evaluation of manual assembly. *CIRP Annals - Manufacturing Technology* 62(1):1-4.

[7] ElMaraghy HA (1993) Evolution and future perspectives of CAPP. *CIRP Annals - Manufacturing Technology* 42(2):739–751.

[8] Nonaka Y, Erdős G, Kis T, Kovács A, Monostori L, Nakano T, Váncza J (2013) Generating alternative process plans for complex parts. *CIRP Annals - Manufacturing Technology* 62(1):453-458.

[9] van Holland W, Bronsvoort WF (2000) Assembly features in modeling and planning. *Robotics and Computer Integrated Manufacturing* 16(4):277-294.

[10] Wang L, Keshavarzmanesh S, Feng HY (2011) A function block based approach for increasing adaptability of assembly planning and control. *International Journal of Production Research* 49(16):4903-4924.

[11] Jones RE, Wilson RH, Calton TL (1998) On constraints in assembly planning. *IEEE Transactions on Robotics and Automation* 14(6):849-863.

[12] Márkus A, Váncza J, Kovács A (2002) Constraint-based process planning in sheet metal bending. *CIRP Annals - Manufacturing Technology* 51(1):425-428.

[13] Wang Y, Liu JH (2010) Chaotic particle swarm optimization for assembly sequence planning. *Robotics and Computer-Integrated Manufacturing* 26(2):212-222.

[14] Leu MC, ElMaraghy HA, Nee AYC, Ong SK, Lanzetta M, Putz M, Zhu W, Bernard A (2013) CAD model based virtual assembly simulation, planning and training. *CIRP Annals - Manufacturing Technology* 62(2):799-822.

[15] Hooker JN, Ottosson G (2003) Logic-based Benders decomposition. *Mathematical Programming* 96(1):33-60.

[16] Kardos C, Kovács A, Váncza J (2016) Towards feature-based human-robot assembly process planning. *Procedia CIRP* 57:516-521.

[17] Pan J, Chitta S, Manocha D (2012) FCL: A general purpose library for collision and proximity queries. *IEEE International Conference on Robotics and Automation*, pp. 3859-3866.

[18] Pellegrinelli S, Moro FL, Pedrocchi N, Tosatti LM, Tolio T (2016) A probabilistic approach to workspace sharing for human–robot cooperation in assembly tasks. *CIRP Annals - Manufacturing Technology* 65(1):57-60.