Int J Adv Manuf Technol (2014) 71:973–982
DOI 10.1007/s00170-013-5515-7

ORIGINAL ARTICLE

Optimization for process plans in sheet metal forming

Silke Wagner · Madan Sathe · Olaf Schenk

Received: 29 May 2013 / Accepted: 19 November 2013 / Published online: 15 December 2013
© Springer-Verlag London 2013

Abstract Determining a process plan in the early phase of the sheet metal forming process is a mandatory task for a process planner. The objective of a process planner is to find a feasible and cost optimal process plan which, in particular, optimizes the assignment of processing elements to processing steps of the production process. We propose to find such an assignment in an automatic way for all the hole features by splitting the entire task into three subsequent steps. At each step, the combinatorial optimization problem is modeled as a bin packing problem with conflicts, and heuristically solved by a specifically designed ant colony optimizer. It is ensured that, at each step, the process plan is feasible while minimizing the tooling costs. In our computational results, we compare our approach to the existing greedy heuristic when computing a process plan for five different practice-relevant sheet metal parts, and show that we can save up to 50 % of the entire tooling costs.

Keywords Process plan · Sheet metal parts · Tooling costs · Ant colony optimization

1 Introduction

During the last decade, the car producers and their suppliers have to face increasingly harsh market conditions: in order to stay competitive, they have to react on the market's demand for more variety in the car models. New regulations about CO_2 emissions and about safety force them to develop lighter and at the same time more robust cars. The further globalization of the markets and severely increasing raw material costs lead to an enormous cost pressure. All in all, this means that the car producers have to optimize the three competing target values quality, time, and costs at the same time. The sector of development, planning, and production of the car body components is affected even stronger by this effect as the increased number of car models is realized mainly by varying the car body design of existing models. Furthermore, the traditional sectors of the car production chain have to compensate for higher costs on the growing electronic sector.

Most of the costs of a sheet metal part are generated relatively late in the process chain, namely during die building and part production. However, the costs are determined during the first planning phase where the process plan for the part is set up. Therefore, it is essential that the process planner gets feedback about the resulting costs of a potential process plan. As slight changes in the process plan may lead, for instance, to a collision of tool components in the dies, it is important for the planner to assure the feasibility of the part when setting up or changing the process plan. In this paper, we suggest an optimization method to determine a cost optimal and feasible process plan.

The remaining part of the article is organized as follows: in Section 2 we describe the general production process of sheet metal parts followed by a description of the planning process and how it is performed nowadays. Furthermore, a method for semiautomatic process planning and automatic tool cost calculation is presented. This method is the base and the reference for the optimization approach that is presented in Section 3. Results that compare the output of the reference method and of the suggested optimization approach are presented in Section 4. In Section 5, we conclude and give an outlook on future work.

S. Wagner (✉)
AutoForm Development GmbH, Technoparkstrasse 1,
8005 Zurich, Switzerland
e-mail: silke.wagner@autoform.ch

M. Sathe
Department of Computer Science, University of Basel,
Klingelbergstrasse 50, 4056 Basel, Switzerland

O. Schenk
Institute of Computational Science, University of Lugano, Via
Giuseppe Buffi 13, 6906 Lugano, Switzerland

2 Computer-aided process planning for sheet metal forming

2.1 Production of sheet metal parts

A sheet metal part is produced by a sequence of forming and trimming operations (OPs): at first, a blank has to be cut, either by a special blanking press or directly from the coil. As a second step, there is typically a deep drawing operation in which most of the shape of the part is formed. Afterwards, there is a sequence of trimming and forming operations in which the part gets its final shape. An automation system like, e.g., roboters or gripper bars transports the part from one operation to the next. Figure 1 shows the geometry of a side panel after each of the operations of the production process. Each operation is carried out in a separate press containing a complex die that performs the trimming and forming of the part. If certain regions cannot be accessed from press working direction special components called cams are needed that divert the press force to another direction. Figure 2 shows the working principle of a piercing cam: when the press closes, the upper die moves down and the cam driver pushes the cam in the lower die to the left; the piercing punch that is mounted on the cam cuts a hole in the part geometry (red); a gas spring pushes the cam back to the original position when the press opens. Figure 3 shows the complete upper and lower die for a forming operation containing cams. As cams are very expensive and need a lot of space within the die, the planner tries to avoid the use of cams by positioning the part for each operation in a way that the maximum number of regions of the part can be processed from press working direction. This positioning relative to the press working direction is called tipping. Each die has to be individually designed and built based on the content that the process planner assigns to each operation.

2.2 Planning of the production process

The task of the process planner is to first analyze the input data which consists of the part geometry and typically of further data that forms bounding parameters for the planning process like the part material, the press forces, the maximum number of operations, and the size of the press beds on which the dies have to be mounted. In the next step, a process plan is elaborated which contains the following information:

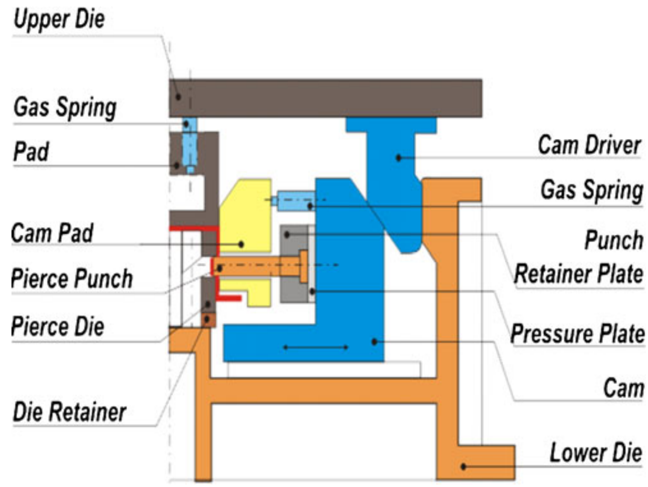


Fig. 2 Working principle of a cam [60]

- Number of operations,
- Tipping per operation,
- Part features: regions of the part geometry that have to be processed in a certain way after the drawing operation in order to get their final shape,
- The processing sequence for each feature which describes the sequence of trimming and forming steps that has to be performed on that feature,
- Assignment of the single steps within a processing sequence (so-called processing units) to the operations,
- For each processing unit: information whether it can be realized from press working direction or it requires a cam,
- Size of the die in each operation.

Depending on the company-specific structure, the planner might also have to estimate the costs of dies but often, this is the task of a separate cost planner in another department who gets the process plan as input.

The general practice to perform these tasks consists of adapting a process plan, respectively, a tooling concept, of a similar part to the new part geometry [2]. Typically, this is done paper-based without connection to the 3D part geometry.

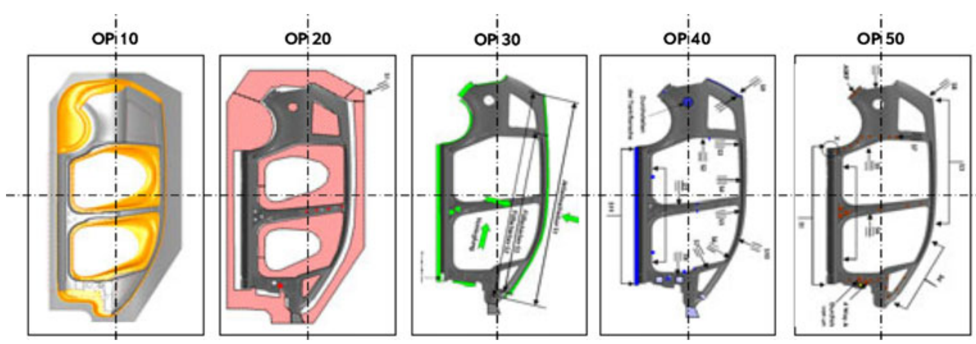


Fig. 1 Side panel of Opel Zafira that is produced in 6 operations (blanking OP is not shown explicitly) [1]

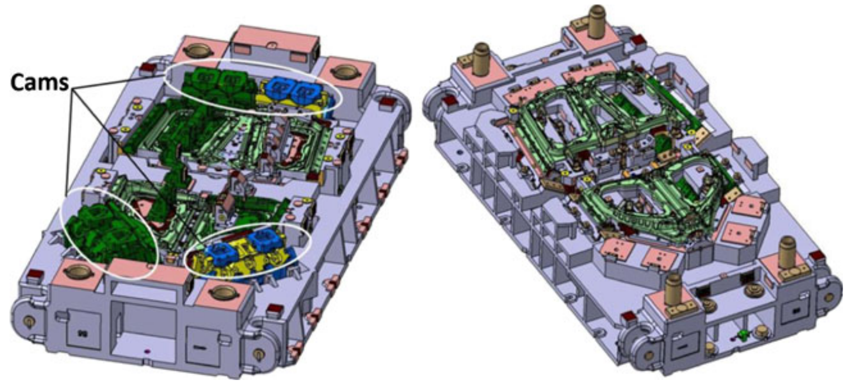


Fig. 3 Upper and lower die of a forming operation with cams. Source: Gessler and Weimann, Weingarten

Therefore, the planner has to first measure the parameters of the part geometry in a CAD system or just on a printed picture of the part in order to "guesstimate" in the next step the information provided within the process plan. The decisions are based on personal experience; however, no feedback about potential better solutions is gathered. The same applies to the cost calculation: at a very early stage in the process, the total costs are often estimated based on the planner's experience [3], later on, when a more detailed calculation is required, it is performed within a commercially or personally developed spreadsheet program without any connection to the part geometry or the process plan.

2.3 Computer-aided process planning and cost calculation

Until recently, there was no standardized and computer-aided method for process planning of sheet metal parts. An integrated approach for semiautomatic process planning and fully automatic tooling cost calculation based on 3D part geometry is proposed in [4, 5]. It consists of the following steps:

- Feature detection: analysis of the 3D part geometry in order to find the part features. Figure 4 shows the features of a bumper.
- Process plan: assignment of appropriate processing sequences to the features and distribution of the processing units to the available operations taking into account bounding conditions of the process and the press (see Fig. 5).
- Tooling concept: derivation of a parameterized list of tool components that are associatively linked to the part geometry and the process plan.
- Costs: for each tool component, calculation of the resource consumption (labor and material) and the costs are based on the component parameters.

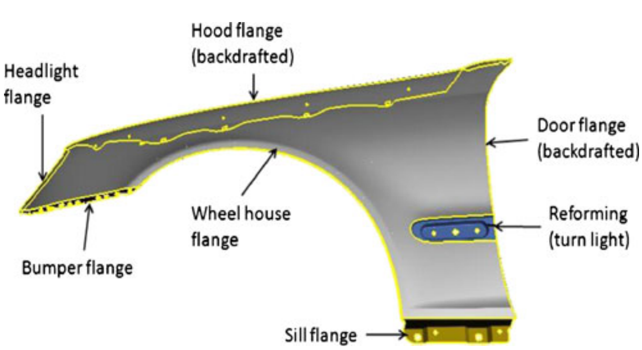


Fig. 4 Features of a fender (holes and trim lines are not labeled) [5]

No other approach is known that integrates the process plan and the tooling costs with the part geometry in a way that the costs are updated automatically if the part geometry or the process plan changes. For the single steps of this procedure, similar approaches are known:

- Feature detection: most of the work focuses on 2D or 2.5D milling features, the latest approaches try to combine different techniques [6–9]. Only a few articles can be found for features of sheet metal parts [10] which are typically 3D and contain free-form surfaces [11, 12].
- Associative process plan/tooling concept: in [11] a hybrid system of knowledge-based engineering and case-based reasoning is suggested which tries to retrieve similar cases from a database. Approaches with an associative parametric linkage between the part geometry and the die faces can be found in [13, 14].
- Tool cost calculation: literature about geometry-based cost calculation typically refers to costs for the 3D object in consideration [15–17] but in our case, the object in consideration is the sheet metal part whereas the costs refer to the dies that are needed to produce that part. Verlinden et al. [18] compares the suitability of artificial neural networks and multiple regression for the costs of sheet metal parts. In [19], a top-down calculation based on tolerance features is presented. Other approaches base on a search for similar cases via case-based reasoning–the search can be based on the part geometry [20] or on an alphanumeric manufacturing data set (e.g., bill of material) [21] or on a combination of both [22, 23].

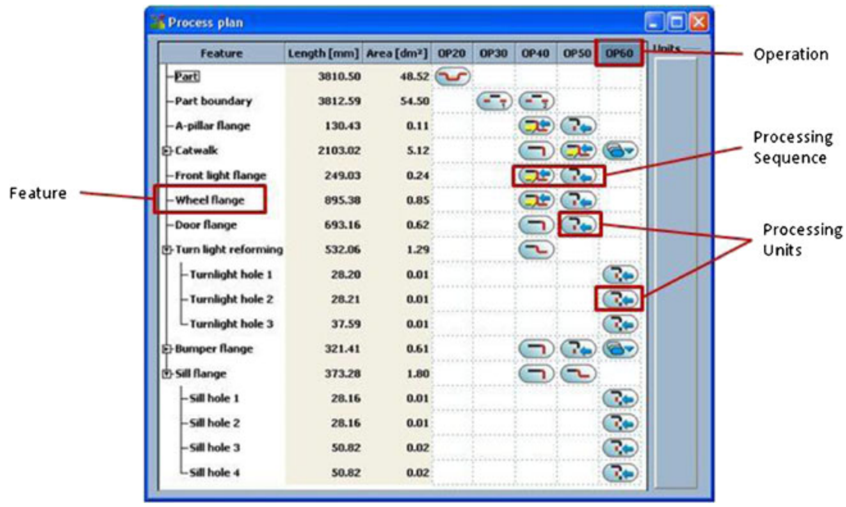


Fig. 5 Process plan for the fender in Fig. 4

The approach in [5] contains the possibility to manually adjust and optimize the process plan that is automatically derived from the part geometry. The most important options are the change of the tipping for certain operations, the grouping of holes, the sharing of cams, and the removing of a complete operation. Each of these steps leads to a reduction of the number of tooling components and thus to a cost reduction. However, there are constraints that have to be respected; otherwise the process plan might not be feasible any more. Furthermore, these steps strongly interact, e.g., removing an operation may cause the additional usage of cams in the remaining operations (which then might be shared between different features). For a planner, it is impossible to manually find a best and feasible solution within this huge amount of possible combinations of tippings, groupings, shared cams, and potentially removed operations. In the next section, we suggest an optimization method for this problem.

3 Optimization of the process planning

The main goal is to find a cost optimal and feasible process plan: produce a valid sequence of operations with a feasible assignment of processing units to each operation while minimizing the total costs of the manufacturing process. We limit our optimization to “hole” features, specified by the types round, rectangular, long, special shape, opening, large opening, and large plunging. The geometry of a feature is given as 3D Cartesian coordinates, the center of gravity, and the processing direction. Each feature is assigned to exactly one OP, in which the feature and the OP are linked via a processing unit that specifies how the feature is processed in that operation. For hole features, there is only one valid processing unit which is piercing. For the optimization, the constant cost per processing unit can be ignored, only the variable costs, i.e., the cam costs, are considered for the processing unit. Holes that are close to each other and that have the same working direction can be processed together, i.e., they share certain components. This is represented by grouping the features, respectively, their processing units. Furthermore, a cam can be shared between features (or processing units, respectively) that are processed within the same operation and that have working directions that are close to each other (within a certain limit). The task is to assign processing units to groups, assign groups to OPs, and minimize the total costs—minimize the usage of cams and minimize the number of OPs—for the full assignment. We find an ordering of the OPs for saving cams such that the operation direction between two consecutive OPs is smaller than a certain value, here 20 °. To make the final process plan feasible, approximating collision detection is implemented by considering distance constraints within and between OPs and processing units. The tipping of OPs is also implemented which has an influence on the decision to use cams or not.

The optimization of a process plan can be stated as a combinatorial problem—a bin packing problem (BPP) with conflicts. The general bin packing problem [24, 25], with the item set $I=\{1,\ldots,m\}$ and the bin set $J=\{1,\ldots,n\}$, models the task of packing a given number of items into a minimal number of bins subject to the capacity restriction for the bins. The mathematical program looks as

$$\min \quad \sum_{j=1}^{n} y_j, \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} y_j = 1\,, i \in I \qquad \text{(Singularity)} \tag{2}$$

$$\sum_{i=1}^{m} w_i x_{ij} \leq W y_j, j \in J \qquad \text{(Weight Capacity)} \tag{3}$$

$$x_{ij}, y_j \in \{0,1\}, \quad i\in I, j\in J \tag{4}$$

where $x_{ij} = \begin{cases} 1 & \text{if item } i \text{ is packed into bin } j, \\ 0 & \text{otherwise}. \end{cases}$

and $y_j = \begin{cases} 1 & \text{if bin } j \text{ is used}, \\ 0 & \text{otherwise}. \end{cases}$

The objective function (see Eq. 1) minimizes the number of used bins j. The singularity constraints (see Eq. 2) ensure that each item i is assigned to exactly one bin j. Each bin j has a weight capacity restriction W. The sum over the weights w_i of items i in a bin should be smaller than the capacity W of each bin (see Eq. 3). The decision variables are restricted to be binary valued (see Eq. 4).

Due to the fact that the bin packing problem is NP-hard, several exact heuristics and approximation algorithms [26–36] are developed to solve the problem accurately. Metaheuristics [37] are also quite successful applied to BPP based on

- Tabu search [27, 36],
- Simulated annealing [38],
- Ant colony optimization [39, 40],
- Variable neighborhood search [41, 42], and
- Evolutionary algorithms [43–51].

In practical applications, metaheuristics can be more successful than exact heuristics since they do not require any kind of derivative information and also work well for problems with a moderate size (up to 1,000 decision variables). Additionally, it is easy to extend the meta-heuristic algorithm to solve multi-objective bin packing problems where two conflicting objectives are simultaneously optimized [52–55]. However, even in the single-objective case, there will be no guarantee to find the global best solution for the process. For solving the many-constraint bin packing problems, motivated by optimization of process plans, heuristics based on ant colony optimization are proposed.

3.1 Ant colony optimization

Ant colony optimization (ACO) was first introduced as ant system and later renamed to ACO [56–59]. The idea comes from real ants in nature finding the shortest path between their nest and food source. Ants constantly deploy pheromones. In general, an ant chooses a path having higher pheromone concentration with a higher probability; therefore at the beginning, all possible paths are chosen with the same probability. The shorter paths take less time than the longer ones; therefore the pheromone concentration on the shorter paths grows faster than on the longer ones. Hence, more ants choose the shorter paths so that the pheromone concentration there grows even faster. Finally, almost all ants of the colony take the shortest path when a trail has emerged. The ants find the shortest path from the nest to the food source only by communicating indirectly via the pheromones.

The behavior of ants is mapped to a graph with vertices $1, \ldots, n$. The path from vertex i to vertex k is represented by the edge (i, k) and τ_{ik} denotes the pheromone concentration on edge (i, k). ACO consists of two main parts: the solution generation and the pheromone update as shown in algorithm 1.

Algorithm 1 Pseudocode of the ACO metaheuristic

```
repeat
    generateSolutions();      ▷ fill bins according to pheromones
    pheromoneUpdate();        ▷ update information
    smoothing();              ▷ reinitialize pheromones if required
until terminationCriteriaMet
```

To generate a single solution, an ant chooses an edge from one vertex to another until a path from start to end is found. The probability p_{ik} of an ant going from vertex i to vertex k is calculated with:

$$p_{ik} = \frac{\tau_{ik}^{\alpha} \cdot \eta_{ik}^{\beta}}{\sum_i \sum_k \tau_{ik}^{\alpha} \cdot \eta_{ik}^{\beta}}, \tag{5}$$

where η_{ik}, the a-priori knowledge, is the desirability for taking edge (i, k) and $\alpha, \beta \geq 0$ are constants controlling the influence of τ and η. The pheromone update comprises pheromone evaporation and deployment:

$$\tau_{ik} = \rho \cdot \tau_{ik} + \Delta_{ik},$$

where $\rho \in (0, 1)$ is the evaporation speed and Δ_{ik} is the update increment. If edge (i, k) is not part of the generated solution, then $\Delta_{ik}=0$. Otherwise, Δ_{ik} is >0 and is calculated based on the solution quality. It is a valid statement that the better the solution, the bigger is Δ_{ik}.

Over the years, several variations of ACO were invented. The MAX-MIN Ant System significantly reduced the probability of getting stuck in a local optimum and additional parameters were introduced. The pheromone concentrations are bounded by τ_{max} and τ_{min}. At the beginning the pheromones are initialized with τ_{max}, here $\tau_{max} = \frac{1}{1-\rho}$, and then evaporated. Smoothing is a mechanism to escape local optima after stagnation was detected without losing all the information gathered in the earlier iterations. To smooth, the pheromone concentrations are recalculated:

$$\tau_{ik} = \tau_{ik} + (\tau_{max} - \tau_{ik}) \cdot \delta$$

with $\delta \in [0,1]$ controls the influence of the pheromone concentrations before smoothing, whereas $\delta = 1$ equals a reinitialization of the pheromone concentrations with τ_{max}. Smoothing takes place if the modulo of iteration count and stagnation is 0. During iteration, a constant number of solutions will be generated. The best solution of this iteration is called the local best solution. The best solution found since the last smoothing is called the global best solution. To decide whether to update with the global or local best solution after iteration, the parameter γ was introduced. It updates with the local best solution before an update with the global best solution is performed (control exploration versus exploitation).

3.2 Three-step ant colony optimization for process planning

The solution approach is implemented in a three-step approach to find an optimal process plan. Each step can be modeled as a BPP with conflicts where either some constraints are added or changed in the original formulation, or the objective function is reformulated as a cost function. The mapping from bin packing to the process plan design is that items are considered as features and bins as groups of items, or in the final process plan as OPs. Recall that in the process plan, each feature should be assigned to exactly one OP. Thus, the decision is binary valued if a feature is placed in a group or not. The objective function can be interpreted as a minimization function over the number of OPs, or substituted as a cost function to find a cost optimal process plan.

A specific modification is made on the generic ant optimizer to solve each step accurately. We have designed for each step an ant colony optimizer:

1. Grouping–Ant: grouping of holes based on geometrical data,
2. No-Cam–Ant: combining groups of holes to super groups without using cams,
3. OP–Ant: merging of the super groups to a feasible process plan.

The first two steps combine features to geometrically connected components. The grouping in the first step considers costs and in the second step minimizes the number of super groups. The cost optimal process plan is then generated from the super groups to build OPs implemented by the third step.

To adapt ACO to bin packing problems, the pheromone trail (see Eq. 5) is reinterpreted: τ_{ik} encodes the favorableness to pick feature i and k for the same bin. Thus, pairing of features is the major goal of this pheromone trail. Note that the information is not taken into account to which group these two features are assigned. In general, we set the a-priori knowledge η_{ik} equal to 1. In the original bin packing formulation, each item has a weight, and in the process plan, each feature can be described by a bounding box, which consists of two three-dimensional vectors $x, y \in \mathbb{R}^3$. The weight of a feature is given as the distance of its bounding box to the centers of gravity of all other features. Thus each feature holds a distance vector of size the number of features.

To generate a solution with an ant, a feature is randomly picked from the feature set and placed into a new feature group. Then, to select feasible candidates for this group, the available constraints are checked for the un-grouped features. A feature is feasible for this group if the constraints are fulfilled with respect to all features in the group. If there is a feasible candidate, the feature is added to the group. If there is no feasible feature available, a new group is created and the procedure starts from the beginning with the remainder of the feature list. When all features are grouped together, the objective function values are evaluated and the pheromone trail is updated with the best solution in this iteration. It is also possible to update with the best solution over all iterations after $\kappa\gamma$ iterations with $\kappa, \gamma > 0$ and γ is a predefined constant. The update is performed either by the normalized costs or by the number of used bins. If the ant optimization process is stagnating, smoothing will be activated and the pheromone concentration is reset. We will describe each step in more detail.

3.2.1 Grouping–Ant

Instead of the capacity constraint in BPP, we use several other constraints to check for feasibility. Additionally, the objective function is given by the cost function in relation to the cam costs.

The constraints are five geometrical distances for grouping of features. These constraints can be expressed by using the decision variable $x_{ij} \cdot x_{kj}$ having value either 1 (both features i and k are in the group j) or 0 (otherwise). Clearly, $x_{ij} + x_{kj} - 1$ is the linearization of $x_{ij} \cdot x_{kj}$. In the following, the term distance d_{ik} is defined as $d_{ik} = \|(x_i, y_i, z_i)^T - (x_k, y_k, z_k)^T\|$ where $\|\cdot\|$ is an arbitrary norm. The maximum distance constraint is based on the limited size of tooling components to process a large group of features (see Eq. 6). But features should not be too close to each other in the same group because dies cover some space to manufacture one feature (see Eq. 7). Within a group a feature should also not be too far located from the next feature according to the sharing properties of dies (see Eq. 8). If the difference in the height is too much, dies are not able to process both features at the same time, where d_z measures the distance between the z coordinates of the two items (see Eq. 9). The last limitation factor is the processing direction of a feature. The angle between the processing directions of two features should be smaller than a certain degree, otherwise the resulting features might not have the desired shape, where $\sphericalangle_{ik}$ calculates the angle between features i and k (see Eq. 10).

$$\text{Max}: \quad d_{ik}\left(x_{ij} + x_{kj} - 1\right) \le D_{max} y_j \tag{6}$$

$$\text{Min}: \quad \left|d_{ik}\left(x_{ij} + x_{kj} - 1\right)\right| + \left|D_{min}\left(x_{ij} + x_{kj} - 2\right)\right| \ge D_{min} y_j \tag{7}$$

Table 1 Geometrical properties of five different parts

Part	Surface [dm^2]	Length [mm]	Width [mm]	No. of holes
A pillar	34.27	1,059.83	1,138.13	16
C pillar	46.54	1,504.63	1,450.51	67
Hood inner	186.61	1,499.97	1,662.37	145
Door inner	60.92	1,125.17	754.92	76
Side panel	364.18	3,442.62	1,245.71	52

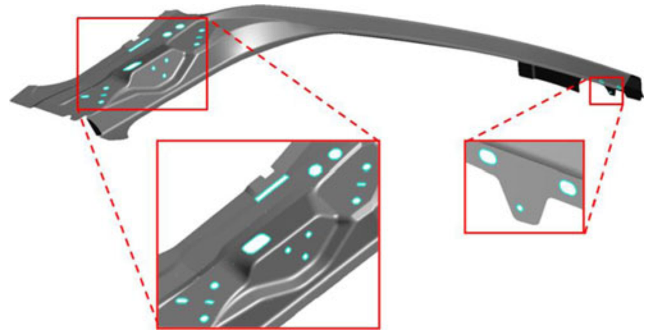


Fig. 7 A pillar geometry containing various holes with different working directions

$$\text{Next}: \quad \exists k : d_{ik}\left(x_{ij} + x_{kj} - 1\right) \leq D_{next} y_j \tag{8}$$

$$\text{Height}: \quad d_z\left(x_{ij} + x_{kj} - 1\right) \leq D_{height} y_j \tag{9}$$

$$\text{Angle}: \quad \sphericalangle_{ik}\left(x_{ij} + x_{kj} - 1\right) \leq P y_j \tag{10}$$

where D_{max}=380, D_{next}=70, D_{height}=12, D_{min}=4, and P=15.

The objective function is $\min \sum_{j=1}^{n} c_j y_j$ where c_j is computed by the cam costs for the group j.

Now, a number of groups of features are clustered which can be processed almost cost optimal; however, the number of groups gets quite large. Thus, the next step must be to further merge the groups to super groups.

3.2.2 No-Cam–Ant

The No-Cam–Ant combines the groups of the Grouping–Ant to super groups. It is not yet known if a group will be processed with a cam or can be processed from the press working direction. The groups of features are represented as the items of the BPP and the super groups are the bins of the BPP. The constraints are checked during filling into the super groups. The constraint types which were active to find feasible super groups are again the minimum distance constraint by the same argumentation. As a second constraint, the angle between the processing directions of the two groups of features is limited, because otherwise, a cam is required, causing additional costs which should be decided by the last ant optimizer.

$$\text{Min}: \left|d_{ik}\left(x_{ij} + x_{kj} - 1\right)\right| + \left|D_{min}\left(x_{ij} + x_{kj} - 2\right)\right| \geq D_{min} y_j$$

$$\text{Angle}: \sphericalangle\left(v_j, v_{j+1}\right) \leq Q y_j, j \in J, v_j \in \mathbb{R}^3, Q = 20$$

The objective function minimizes the number of super groups to achieve a smaller number of candidates for the final process plan.

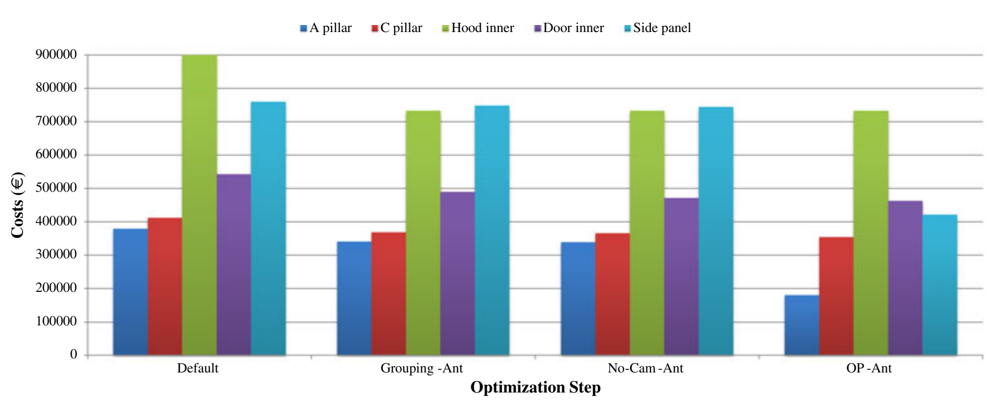
3.2.3 OP–Ant

The super groups of the previous step are merged to OPs to find a feasible and cost optimal process plan. Here, the ACO approach is slightly changed. The OP–Ant generates orderings for the super groups which are then evaluated by the objective function. The pheromone concentration τ_{ik} tells how favorable it was to choose operation k after operation i. The objective function minimizes the cam costs.

When an operation is chosen, the following steps are performed: the selected super group creates a new OP, and the processing direction of the super group becomes the operation direction. The next super group checks if it is feasible to process the super groups on the current OP:

$$\sphericalangle\left(v_j, v_{j+1}\right) \leq R y_j, j \in J, v_j \in \mathbb{R}^3, R = 20 \tag{11}$$

Fig. 6 Cost comparison of the default configuration and the multi-step ant colony approach



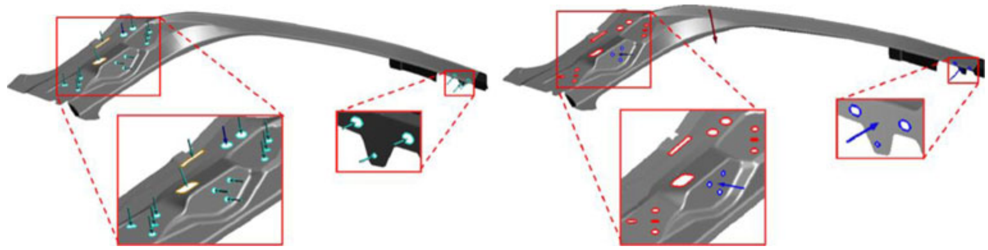


Fig. 8 Solution of the greedy heuristic (*left*) and the ACO optimization (*right*): the number of cams (*arrows*) can be reduced from 17 to 2

If this is cheaper than creating a new OP, the operation becomes a sub OP of the current operation. Otherwise, a new OP will be created. We have to determine the operation direction v of the new OP. If the angle between the operation direction of the previous OP and the current processing direction is smaller than 20 °, the processing direction becomes the operation direction of the new OP. Otherwise, the operation direction v^* of the new OP will be determined by moving the operation direction of the previous OP by 20 ° in direction to the current processing direction:

$$v^*_{j+1} = v_j + q \cdot v_{j+1}, q \in \mathbb{R}^3. \quad (12)$$

This mechanism generates always a feasible process plan. However, the three-step approach is based on a metaheuristic such that the global optimality cannot be guaranteed, but a high quality of the solution is expected.

Due to the complexity of the practical optimization problem, i.e., finding a feasible assignment of processing elements to processing steps, a three-step ACO approach has been chosen as the adequate solution process to find an optimal process plan. The advantages when following a three-step ACO approach are twofold:

On the one hand, there are natural dependencies between the intermediate optimization problems which we think cannot be modeled as a single optimization problem. For instance, the third ACO optimizer, the ordering of processing steps, needs as a mandatory input the set of processing elements on each processing steps. The elements are finally determined by the second ACO optimizer. The input for the second step requires groups of elements which are geographically categorized into the same group to decide if a cam can be shared among groups. The groups are identified by the first step.

On the other hand, there is a decision-making advantage for the process planer as after each step, the results of the ACO optimizer can be visualized and interpreted by a process planer. For instance, the Grouping–Ant determines geographical groups on the sheet. As this solution itself might not be the only optimal solution, a process planer can possibly justify the result and prioritize certain assignments of elements to groups.

4 Computational results in computer-aided process planning

The multi-step ACO approach is applied to five different parts of a car (see Table 1, Fig. 6). The parts differ in length and width, and also the number of holes changes drastically. For instance, the part hood inner has a large number of holes but all holes can be processed from the same processing direction. In contrast to this part, the part A pillar has only a few holes (see Fig. 7) but grouping and tipping will have a strong influence on the costs.

Currently, as the default configuration, a greedy heuristic is implemented which consists of several steps: first, the part geometry is tipped into an average normal position, i.e., the weighted average vector of all part face normal is opposing the press working direction. This tipping is applied to all OPs where the number of OPs has to be defined by the user. Based on this position and based on the tolerances for the working directions, features are assigned to the processing operations by the following scenario: starting from the original geographical location, all features next to this location are one-by-one

Table 2 Comparison between the default configuration and the optimized variant when applied to different parts

	No. of cams		No. of OPs		No. of groups		Costs (in 10^4)	
Part	Default	ACO	Default	ACO	Default	ACO	Default	ACO
A pillar	17	2	2	1	0	4	37.9	18.0
C pillar	41	6	1	1	0	14	41.2	35.4
Hood inner	64	8	2	2	0	19	54.2	46.2
Door inner	0	0	2	2	0	27	90.3	73.2
Side panel	50	7	2	1	0	7	76.0	42.1

assigned to the first possible processing operation. Then, cams are assigned to the features (or processing units, respectively). In the last step, the processing units are assigned to the OPs based on a set of constraints. Note that there is no change in the costs when the assignment is changed because the number of operations is fixed and all operations have the same tipping such that a reassignment of the processing units does not lead to a lower number of cams. Furthermore, cam sharing is not considered for the automatic assignment. The optimization can only be done manually by changing the tipping of an operation, grouping features, or sharing cams between features. Hence, the greedy heuristic generates a good start assignment with an upper bound for the cost but not a cost optimal assignment.

We compare the costs obtained by the multi-step ACO approach with the costs of the default greedy configuration (see Fig. 6). Using the Grouping–Ant, a maximum reduction in costs is achieved in case of Hood inner by an amount of 19 %, in case of side panel by an amount of 1.6 %. In the next step, the No-Cam–Ant may reduce the costs further if cams can be shared among groups. The reduction percentage ranges from 4 % (door inner) to 0.05 % (A pillar). In hood inner, the processing direction of the holes do not require any cams, thus, no groups are sharing cams. The OP–Ant finds the optimal tipping for the part and merges cams among groups. In the parts, A pillar and side panel, a final cost reduction of around 50 % can be achieved where in part C pillar, a reduction of 14 % is attained. Figure 8 shows the solution of the ACO optimization (right) in comparison to the result of the greedy heuristic (left) for the A pillar: due to grouping and better tipping the amount of cams can be reduced from 17 to 2.

In Table 2, geometrical aspects of parts by considering cams, OPs, and groups, and the total costs are compared to each other using the default and the multi-step ACO heuristic. The number of cams can be considerably minimized by grouping the holes and sharing cams among holes. Additionally, the number of OPs can be reduced. On average, a total reduction of the costs by 29 % compared to the greedy heuristic is obtained.

5 Conclusions and outlook

We have presented a three-step approach to obtain fully automatic a feasible and cost optimal process plan. At each step, the optimization problem is heuristically solved by an ant colony optimizer which outperforms the existing greedy approach regarding tooling costs and practicability of the process plan. At maximum, about 50 % of the tooling costs compared to the greedy heuristic can be saved.

Future work will focus on the enlargement to the processing of forming features like e.g., flanges or forming areas. The challenge consists of the preprocessing of the features: a forming feature might be formed in several steps and the intermediate geometries are not known; therefore, the computation of working directions for the features is difficult. An approach consists in analyzing the sequence of radii in the intersection of the form feature.

Acknowledgments This work was supported by the Swiss Commission for Technology and Innovation (CTI) under grant CTI No. 8582.1 ESPP-ES.

References

1. Emrich A, Compensation of springback based on the formability analyses results, in: The Press Plant facing shifting requirements, volume 8 of Automotive Circle International Conference, p. 160
2. Schrempp G, Werteveränderung durch modifizierte Allokationen auf der Prozesskette Blech, Vortrag, 2007. Available online at http://logistics.de/downloads/4f/2a/i_file_45370/ allokation-prozesskette-dlk-2007-schrempp-pwo.pdf; visited on April 24th 2012
3. Schuh G, Boos W (2004) Werkzeugfunktionen als Kalkulationsbasis, Plastverarbeiter 6
4. Kubli W, Prabhakar V (2008) Digital process planning for automotive body panels, in: Proceedings of the 7th International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes (Numisheet)
5. Pietsch M, Wagner S, Computer aided method for an associative process layout generation, in: Proceedings of the 7th International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes (Numisheet) 2008
6. Li W, Ong S, Nee A (2003) A hybrid method to recognizing interacting machining features. International Journal of Production Research 41:1887–1908
7. Owodunni O, Hinduja S, Systematic development and evaluation of composite methods for recognition of three-dimensional subtractive features, in: Proceedings of Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, volume 219, pp. 871–890
8. Rahmani K, Arezoo B (2007) A hybrid hint-based and graph-based framework for recognition of interacting milling features. Computers in Industry 58:304–312
9. Verma A, Rajotia S (2009) Hybrid machining feature recognition system. International Journal of Manufacturing Research 4:343–361
10. Nasr EA, Kamrani AK (2007) Computer-based design and manufacturing: an information-based approach. Springer, Berlin
11. Sunil V, Pande S (2008) Automatic recognition of features from freeform surface cad models. Computer-Aided Design 40:502–517
12. Zhang C, Zhou X, Li C (2009) Automatic recognition of intersecting features of freeform sheet metal parts. Journal of Zhejiang University Science A 10:1439–1449
13. Danzberg J (2001) Einsatz hierarchischer parametrisierter Konstruktionsmetodik zur Planung und Optimierung von Karosserieblechteilen, Ph.D. thesis, ETH Zurich
14. Hillmann M, Kubli W (2005) Bestimmung von Prozessoperationen zur Beschreibung von Umformprozessen an einem Umformteil
15. Ehrlenspiel K, Kiewert A, Lindeman U (2005) Kostengünstig Entwickeln und Konstruieren, Springer Verlag Berlin Heidelberg
16. Fischer J (2008) Kostenbewusstes Konstruieren, Springer Verlag Berlin Heidelberg
17. Pahl G, Beitz W, Feldhusen J, Grote KH (2007) Konstruktionslehre, Springer Verlag Berlin Heidelberg
18. Verlinden B, Duflou JR, Collins P, Cattrysse D (2008) Cost estimation for sheet metal parts using multiple regression and artificial

neural networks: a case study. International Journal of Production Economics 111(2):484–492

19. A. Wittmann, A.-W. Scheer (2000) FIT Featurebasiertes Integriertes Toleranzinformationssystem, Technical Report, Universitat Saarbrücken, Institut für Wirtschaftsinformatik
20. Tangelder JW, Veltkamp RC (2008) A survey of content based 3d shape retrieval methods. Multimedia Tools and Applications 39: 441–471
21. Romanowski RNCJ (2005) On comparing bills of materials: a similarity/distance measure for unordered trees. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 35: 249–260
22. Karadgi S, Müller U, Metz D, Schafer W, Grauer M Cost estimation of automotive sheet metal components using knowledge-based engineering and case-based reasoning, in: IEEE International Conference on Industrial Engineering and Engineering Management, pp. 1518–1522.
23. Grauer M, Metz D, Müller U, Karadgi S, Sch¨afer W, Barth T (2010) Towards an integrated virtual value creation chain in sheet metal forming. Lecture Notes in Business Information Processing 46: 186–197
24. Martello S, Toth P (1990) Knapsack problems: algorithms and computer implementations. Wiley, Chichester
25. Wächer G, Häusner H, Schumann H (2007) An improved typology of cutting and packing problems. European Journal of Operational Research 183:1109–1130
26. Alves C, de Carvalho JMV (2008) A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem. Computers & OR 35:1315–1328
27. Alvim ACF, Ribeiro CC, Glover F, Aloise DJ (2004) A hybrid improvement heuristic for the one-dimensional bin packing problem. Journal of Heuristics 10:205–229
28. Anily S, Bramel J, Simchi-Levi D (1994) Worst-case analysis of heuristics for the bin packing problem with general cost structures. Oper Res 42:287–298
29. Belov G (2003) Problems, models and algorithms in one and twodimensional cutting, Ph.D. thesis, Technical University of Dresden
30. Coffman J, Garey EG, Johnson DS et al (1997) Approximation algorithms for bin packing: a survey. In: Approximation algorithms for NP-hard problems. PWS, Boston, pp 46–93
31. Fekete SP, Schepers J New classes of lower bounds for bin packing problems, in: IPCO: 6th Integer Programming and Combinatorial Optimization Conference, volume 1412 of Lecture Notes in Computer Science, pp. 257–270
32. Gendreau M, Laporte G, Semet F (2004) Heuristics and lower bounds for the bin packing problem with conflicts. Computers & Operations Research 31:347–358
33. Gent I (1998) Heuristic solution of open bin packing problems. Journal of Heuristics 3:299–304
34. Gupta JND, Ho JC (1999) A new heuristic algorithm for the one dimensional bin-packing problem. Production Planning & Control 6: 598–603
35. Kao C-Y, Lin F-T (1992) A stochastic approach for the one-dimensional bin-packing problems. IEEE International Conference on Systems, Man and Cybernetics 2:1545–1551
36. Scholl A, Klein R, Jürgens C (1997) Bison: a fast hybrid procedure for exactly solving the one-dimensional bin packing problem. Computers & OR 24:627–645
37. Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Computing Surveys 35: 268–308
38. Loh K, Golden BL, Wasil EA (2008) Solving the one-dimensional bin packing problem with a weight annealing heuristic. Computers & OR 35:2283–2291
39. Brugger B, Doerner K, Hartl RF, Reimann M, Antpacking—an ant colony optimization approach for the one-dimensional bin packing problem, in: EvoCOP, pp. 41–50.
40. Levine J, Ducatelle F (2004) Ant colony optimisation and local search for bin packing and cutting stock problems. Journal of the Operational Research Society 55:705–716
41. Fleszar K, Hindi KS (2002) New heuristics for one-dimensional bin-packing. Computers & OR 29:821–839
42. Mladenovic N, Hansen P (1997) Variable neighborhood search. Computers & OR 24:1097–1100
43. Falkenauer E (1996) A hybrid grouping genetic algorithm for bin packing. Journal of Heuristics 2:5–30
44. Falkenauer E (1994) A new representation and operators for genetic algorithms applied to grouping problems. Evol Comput 2:123–144
45. Iima H, Yakawa T (2003) A new design of genetic algorithm for bin packing, in: Evolutionary Computation, CEC'03. The 2003 Congress on, volume 2, pp. 1044–1049
46. Jing X, Zhou X, Xu Y (2006) A hybrid genetic algorithm for bin packing problem based on item sequencing. Journal of Information and Computing Science 1:61–64
47. Khüri S, Schütz M, Heitkotter J (1995) Evolutionary heuristics for the bin packing problem. In: Pearson DW, Steele NC, Albrecht RF (eds) Artificial Neural Nets and Genetic Algorithms. Proceedings of the International Conference in Ales, France, pp 285–288
48. Reeves C (1996) Hybrid genetic algorithms for bin-packing and related problems. Annals of Operations Research 63:371–396
49. Rohlfshagen P, Bullinaria JA, A genetic algorithm with exon shuffling crossover for hard bin packing problems, in: GECCO, pp. 1365–1371.
50. Singh A, Gupta A (2007) Two heuristics for the one-dimensional bin-packing problem. OR Spectrum 29:765–781
51. Stawowy A (2008) Evolutionary based heuristic for bin packing problem. Computers & Industrial Engineering 55:465–474
52. Sathe M, Schenk O, Burkhart H (2009) Solving bi-objective many-constraint bin packing problems in automobile sheet metal forming processes. In: Ehrgott M, Fonseca C, Gandibleux X, Hao J-K, Sevaux M (eds) Evolutionary Multi-Criterion Optimization, volume 5467 of Lecture Notes in Computer Science. Springer Berlin, Heidelberg, pp 246–260
53. D. S. Liu, K. C. Tan, C. K. Goh, W. K. Ho, On solving multiobjective bin packing problems using particle swarm optimization, in: G. G. Y. et al. (Ed.), Proceedings of the 2006 I.E. Congress on Evolutionary Computation, IEEE Press, Vancouver, BC, Canada, 2006, pp. 2095–2102
54. Geiger MJ, Bin packing under multiple objectives a heuristic approximation approach, in: The Fourth International Conference on Evolutionary Multi-Criterion Optimization: Late Breaking Papers, Matsushima, Japan, pp.53–56.
55. Lara O, Labrador M (2010) A multiobjective ant colony-based optimization algorithm for the bin packing problem with load balancing, in: Evolutionary Computation (CEC), IEEE Congress on, pp. 1–8.
56. Dorigo M, Birattari M, Stützle T (2006) Ant colony optimization. Computational Intelligence Magazine, IEEE 1:28–39
57. Dorigo M, Blum C (2005) Ant colony optimization theory: a survey. Theor Comput Sci 344:243–278
58. Dorigo M, Stützle T (2004) Ant colony optimization. MIT Press, Cambridge, USA
59. Stützle T, Hoos H (2000) Max-min ant system. Future Gener Comput Syst 16:889–914
60. AutoForm Development GmbH, CostCalculator Reference Manual, 2009