Sequencing and scheduling in the sheet metal shop

B. Verlinden[1], D. Cattrysse[1], H. Crauwels[2], J. Duflou[1] and D. Van Oudheusden[1]
[1]K.U.Leuven, Centre for Industrial Management
[2]Hogeschool voor Wetenschap en Kunst, De Nayer Instituut
Belgium

1. Introduction

1.1 History

Sheet metal operations have been in existence since 8000 B.C. (Fries-Knoblach, 1999). Due to its long history, sheet metalworking is, unfortunately, often seen as archaic and uninteresting. That metal sheets can be transformed with the aid of robust machines into fancy consumer products with tight tolerances is inconceivable to many. Yet, sheet metal operations are used for producing both structural components and durable consumer goods. Nowadays, sheet metal parts are widely present in different daily life products. During the past decades, scientific research in the field of sheet metal operations has been booming and international conferences on different sheet metal topics attract numerous attendants. Both industry and the academic community recognize the importance of continuing improvement in sheet metal operations.

Thirty years ago, the rapid advance of computer systems triggered the introduction of automation in manufacturing environments. Also for sheet metal operations the use of computer systems has become indispensable to survive. Computer aided design (CAD) and computer aided manufacturing (CAM) systems are widely present in sheet metal production environments. At the earliest design stage, CAD systems are used to computerize the whole process of drawing and redrawing the desired part. Most modern CAD systems allow to build up a part from several re-usable 3D components, thus automating to a large extent the time consuming design process. The use of computers has also entered the manufacturing stage through computer aided manufacturing. The CAD file is converted into a sequence of processes for manufacture on a numerically controlled (NC) machine.

The use of computers helps operators in automating different steps of the production process. As far as planning is concerned, most attention is focused on the computerization of sheet metal process planning. Production planning proper received much less attention due to a long tradition of experience-based production planning.

1.2 Scope of the research

If one queries the internet for sheet metal operations, numerous entries on processes are found, for instance on laser cutting, punching, deep drawing, bending, incremental forming,

Source: Multiprocessor Scheduling: Theory and Applications, Book edited by Eugene Levner,
ISBN 978-3-902613-02-8, pp.436, December 2007, Itech Education and Publishing, Vienna, Austria

etc. Since many different sheet metal production processes and production environments can be identified, it is important to clearly delineate the scope of the research. The research as described in this chapter focuses on sheet metal operations in a flow shop environment (Figure 1). As far as the production processes are concerned, only laser cutting and air bending are considered.

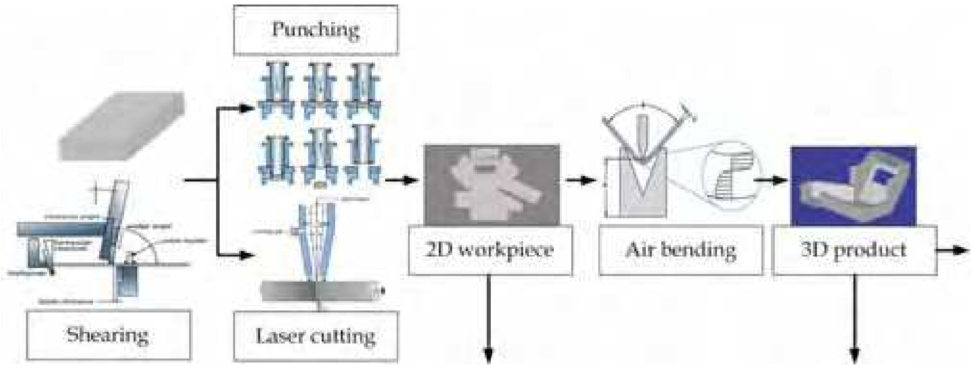


Figure 1. Sheet metal operations

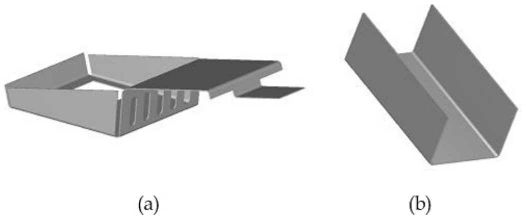


Figure 2. Typical sheet metal parts (a) complex part; (b) standard profile

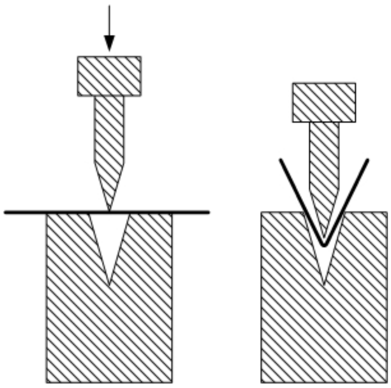


Figure 3. The air bending principle

All parts follow a unidirectional flow in the sheet metal shop. In a preliminary stage, a large standard metal sheet or coil is cut to the right dimensions by using a pair of automated scissors (the shearing operation). Next, in the cutting stage, the unfolded blank of a part is punched with a punch press or cut with a laser machine. After cutting, the parts are bent and transformed into 3D products. Air bending is used for this purpose. The air bending machine or press brake consists of a fixed bed on which the dies are fastened and a vertically movable ram (driven by hydraulic cylinders) on which the punches are mounted. A vertical force causes the ram to move down, forcing the sheet into the die, creating the bend line. The larger the vertical force, the smaller the bend angle (Figure 3).

To produce a bend line of a part, a punch and die are required (called a tool set). The geometrical properties of the bend line (i.e. internal radius, bend allowance, type of profile, etc.) determine the characteristics of the required tool set. Punches and dies are supplied in sections (segments) for easy handling and quick set-up. If a long tool is needed, different segments are required. A combination of different segments to produce a single bend line is called a station. If a part consists of multiple bend lines, multiple stations are usually required (e.g. one station per bend line). Some stations may be used to produce several bend lines, while other stations can only be used for very specific lines. If a part requires different stations, all stations are positioned on the press brake with the required space between them, resulting in a production layout (Figure 4). A production layout can be used for bending several parts (depending on the geometrical properties of the part).

This chapter mainly focuses on combinatorial optimization models and algorithms of use to production planning in Belgian small to medium-sized enterprises. Typically, in a Belgian SME, a single laser machine and press brake are available for production. Batch sizes range from a single workpiece to larger series (200 parts).

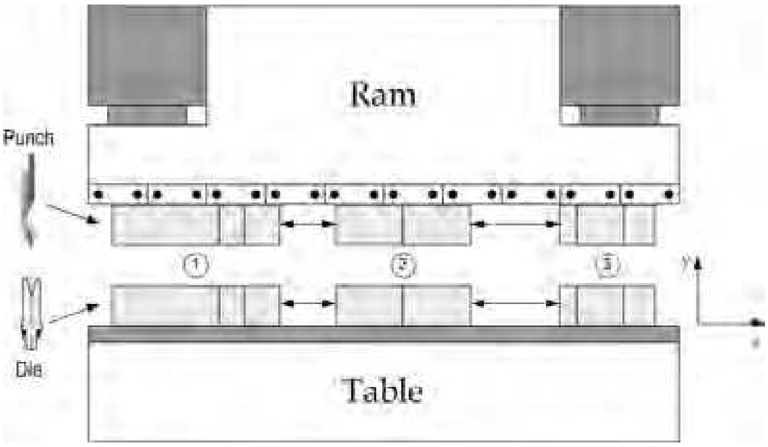


Figure 4. Press brake with a production layout consisting of three stations (every station is built up with different segments)

1.3 Outline

The research as described in this chapter evolved from a project in cooperation with a large sheet metal machine constructor and has been carried out by members of the Centre for Industrial Management, K.U.Leuven. The main goal of the project was to tackle a series of practical problems in a typical sheet metal shop.

The chapter has six sections. Section 2 discusses some process planning issues for sheet metal laser cutting and air bending. An overview is given and relevant literature is mentioned. Section 3 elaborates on the tooling layout problem for air bending. The problem is analyzed and several solution procedures are proposed. Section 4 analyzes the production planning problem for air bending. A traveling purchaser problem approach (TPP) and a generalized traveling salesperson problem approach (GTSP) are proposed to reformulate the air bending production planning problem. Section 5 discusses production planning integration of laser cutting with air bending. The problems are highlighted and mathematical models are proposed. Computational results are discussed for a number of test cases. Section 6 concludes the chapter and formulates future research topics.

In the chapter, mathematical models are formulated for the different planning issues, taking into account numerous problem-specific constraints. Typically, in a first stage, optimization procedures are used to obtain overall optimal solutions, but the computational effort required to solve combinatorial problems to optimality appears to grow very fast with the size of the problem. Local search, a class of heuristic methods, is applied at that point. It allows in a straightforward way to determine good, but not necessarily optimal solutions after a limited computational effort. In the problems discussed in this chapter two classes of local search techniques are implemented: simple neighborhood search and guided local search.

Neighborhood search starts with a known feasible solution and tries to improve this solution by making well-defined moves in the solution space, shifting from one neighbor to another. A move is evaluated by comparing the objective function value of the current solution to that of its neighbor. In a pure descent method, only improving moves are allowed. When no further improvement can be found, the procedure ends. All feasible moves need to be defined and the search has to be initialized with a first solution. Although good choices for the different implementation issues can improve the performance of a descent algorithm, the resulting solution is, most likely, a local optimum, not a global optimum. A classical remedy for this drawback is to perform multiple runs of the procedure starting from different initial solutions and to take the best result as final solution (multi-start descent). Another possibility is to use a combination of different neighborhood structures (variable neighborhood search), as introduced by Mladenović & Hansen (1997). In this way, a local optimum relative to a number of neighborhoods is determined.

A more specialized type of local search is ***guided local search*** (GLS). The main feature of this approach is the iterative use of local search sequences. GLS penalizes, based on a utility function, unwanted solution features at the end of a local search sequence. In this way the solution procedure may escape from a local optimum and continue the search. The interested reader is referred to Voudouris & Tsang (1999).

2. Process planning issues

Besides production planning, the authors have used combinatorial optimization techniques also for a number of sheet metal process planning aspects. Next section discusses the state of the art regarding process planning issues and summarizes some of the process planning research conducted by the authors.

2.1 Process planning for laser cutting

Past decades steel prices have been increasing, resulting in a higher material cost for sheet metal parts. Hence, the amount of waste material needs to be minimized. In order to reduce the waste material, sheet metal parts with identical characteristics (material and thickness) are, if possible, combined on a large standard sheet to obtain high sheet utilization ratios.

The problem of combining different patterns on a flat blank or coil (apart from metal, the material can be paper, cardboard, leather, wood, plastic, textile, etc.) is known as the cutting stock problem and has received much attention. Alternatively, the name nesting is used. Numerous publications and applications can be found on this topic and a wide variety of solution techniques is proposed.

Research on the cutting stock problem started with Eisemann (1957) discussing the problem of minimizing losses when cutting rolls of paper, textile, metal and other materials. This trim problem aims at allocating raws to machines and setting up the different cuts in such a way that the demanded rolls are cut to the right width, minimizing the overall trimming loss. Small problems are solved by enumerating the different possibilities. Most cutting stock problems were at that time formulated as integer programming problems to be solved with standard solution procedures. An overview is given by Haessler & Sweeney (1991). Research also focused on the determination of optimal sheet layouts. For such problems one has a number of rectangular sheets that need to be filled with two-dimensional regular and irregular shapes, minimizing the waste material. In the earliest days, researchers proposed to work with encaging rectangles. Adamowicz and Albano (1976) propose a two-stage approach. In a first stage, optimal rectangular modules are produced encaging one or more parts.

In the second stage those modules are positioned on a sheet, minimizing the waste material. Working with encaging rectangles can only produce a rough plan since the exact shape of the parts is not considered.

When personal computers with larger computational power became available, research shifted towards the nesting of the complex parts themselves instead of encaging rectangles. Lee & Woo (1988) developed a method to seek the minimal area convex hull of two convex polygons P and Q with Q being allowed to be translated along the boundary of P. If both translation and rotation operations are allowed, the problem of finding the minimal area convex hull becomes more difficult. The interested reader is referred to Nye (2001).

Nesting remains an important issue in sheet metal cutting. Different peculiarities inherent to sheet metal production have to be taken into account when nesting different unfolded blanks on a large standard sheet. Prasad (1994) proposes some heuristic algorithms for nesting irregular sheet metal parts, taking into account grain orientation, minimum bridge width (i.e. the distance between two parts should be strong enough to withhold the bending force) and maximal material utilization. Tang and Rajesham (1994) propose an algorithm taking into account the rolling direction of the part. The rolling direction influences the brittleness of the part in both traverse and rolling direction. When bending the part this can cause the bend to crack at the bending edge. Generally it is assumed that a bend made at 30° to the grain flow is enough to avoid breakage. To overcome this problem, rolling direction information is taken into account at the nesting procedure. The method proposed approximates the part by a polygon with a sufficient number of sides to represent closely the original geometric shape. A more recent overview of sheet metal nesting is given by Valvo and Licari (2007).

Besides the nesting proper, other related process planning issues received attention as well. After optimal nestings are generated, the cutting technology (laser power, cutting speed, position of the focal point, cutting gas pressure, stand-off distance between nozzle and material, lead-in, piercing method, etc.) and the cutting path need to be determined (taking into account common cuts, cutting distance minimization, heat build up in the material when cutting, etc.). Fortunately for the end-user, those different issues have been satisfactorily studied and are implemented in commercial nesting software. Such dedicated software tools allow the user to generate good nestings and to determine the cutting technology automatically.

2.2 Process planning for air bending

As for the laser cutting process, research on the bending process focuses on process planning. When a (3D) sheet metal part is produced with air bending, one starts from the unfolded blank of the required part (Figure 5). Gradually, this 2D flat blank is transformed into a 3D final workpiece by producing the different required bend lines. Different sequences are possible for producing the bend lines. Some bend sequences will cause collisions between the part, the machine and the tools while other sequences create no problems. The main goal of process planning is to determine an executable bend sequence and to select the different tools to use for each bend line, the gauging positions and the punch displacements.

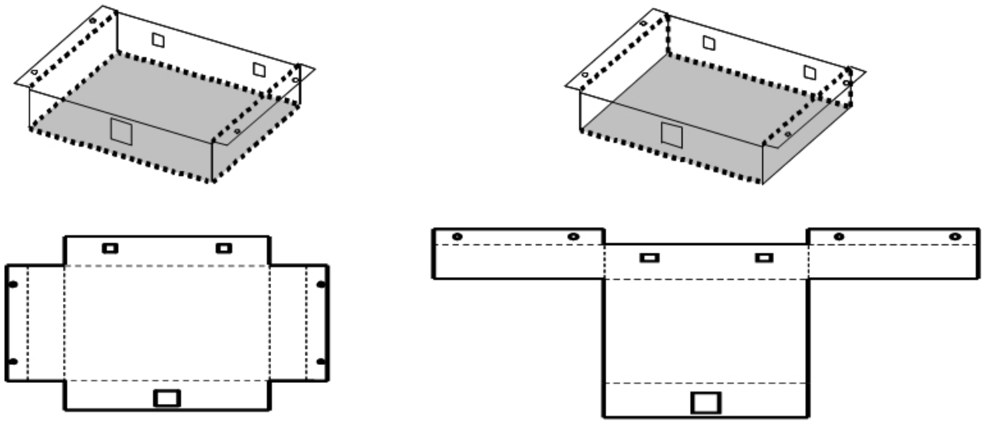


Figure 5. Different unfoldings of a part

In order to determine the bend sequence, the different interactions between the part, the machine and the operator need to be investigated. For this purpose, a model is required representing the part during each of the consecutive process steps. If all geometric specifications, topological information, material characteristics and process related information would be included in such a representation model, the memory requirements could become very large. Some information, however, is redundant and only key information needs to be included. Duflou (1999) proposes a reduced foil model that contains only specific geometric information. This model also contains information regarding the topological relationships between the flange and the bend features of a part. For this purpose the part is represented by a graph with n nodes.

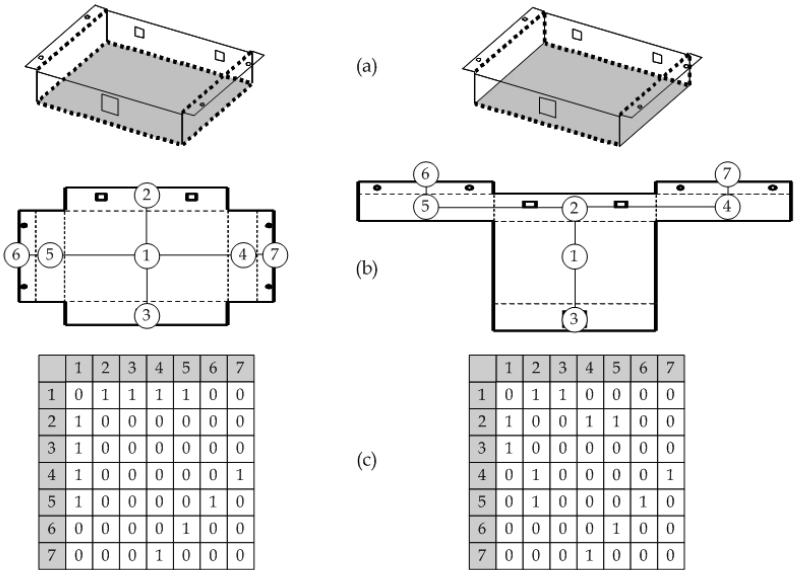
The nodes represent the flanges, while the arcs represent the connecting features. The part can then be represented by a binary matrix, indicating whether two flanges are connected or not (Figure 6). For most parts the graph representation has to correspond with a spanning tree in the complete nxn graph.

If loops are present in the graph representation scheme, a number of possibilities occur:

- the part can still be produced with air bending, but dedicated tooling will be required;

- the part can still be produced if a set of bends is simultaneously executed (so-called compulsory combined bends);
- no producible part's unfolding can be determined: one or more bend connections need to be converted in weld connections or open seams.

The interested reader is referred to Duflou (1999). Additionally to the geometric and topological constraints, also technological constraints are enforced on the part. Non-compliance with those constraints causes the part to be infeasible to manufacture.



	1	2	3	4	5	6	7
1	0	1	1	1	1	0	0
2	1	0	0	0	0	0	0
3	1	0	0	0	0	0	0
4	1	0	0	0	0	0	1
5	1	0	0	0	0	1	0
6	0	0	0	0	1	0	0
7	0	0	0	1	0	0	0

	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	1	0	0	1	1	0	0
3	1	0	0	0	0	0	0
4	0	1	0	0	0	0	1
5	0	1	0	0	0	1	0
6	0	0	0	0	1	0	0
7	0	0	0	1	0	0	0

Figure 6. Part representation schemes (Duflou, 1999): (a) the 3D sheet metal part; (b) the unfolded blank of the part and the spanning tree; (c) the incidence matrix

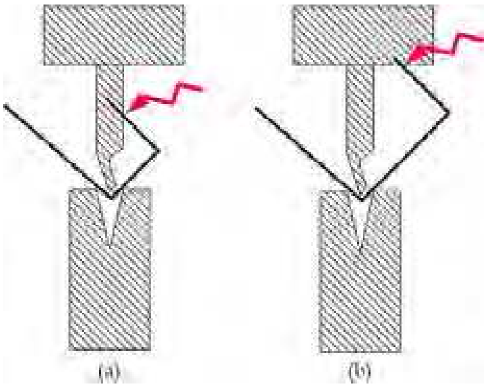


Figure 7. Possible collision with (a) the punch and (b) the ram, when the bend sequence is not optimal

Thus the part representation scheme is used to determine the bend sequence and the gauging positions. The number of bend lines of the part, the number of gauging positions of the press brake and the different possible tools to produce the part tremendously increase the number of possible bend sequences. Duflou (1999) proposes a traveling salesperson problem based method to identify feasible (near-optimal) bend sequences. To limit in this TSP approach the number of possible candidate solutions, a number of criteria is used. Amongst others, ergonomic factors and productivity criteria are taken into account when verifying the feasibility of the different bend sequences. Also hard reject criteria (unavailability of proper gauging edges, collisions, non-compliance of the part dimensions with the specified tolerances, etc.) and soft criteria (e.g. looking at the ease of workpiece handling and total resource utilization) are used to reduce the number of candidate solutions. Each criterion is assigned a weight to be included in the TSP objective function. As such, a feasible (near optimal) bend sequence is determined from the remaining candidates.

Bend sequencing is often integrated with tool selection (Nguyen, 2005). For each bend line a short list of possible tools is made based on technological and geometrical constraints. The technological constraints verify if the part can technically be made with the specific tool. Geometrical constraints check if there are any foreseeable collisions (Figure 7). In general a two-phase approach is followed. In the first phase (preselection phase) all nonconforming toolsets are eliminated for each bend line, based on the technological and geometric constraints. In the second phase (refined selection phase) the preselected tools are adjusted whenever a collision occurs to suit the stricter conditions imposed by this collision. Based on this data, the bend sequence is determined. The search procedure starts at the root node of the decision tree (the first bend line to be produced). Gradually, other bend lines are added to the intermediate bend sequence. At each node of the decision tree, collision detection tests are executed. If a collision occurs, all nodes from the same level are tested to detect a collision-free path. If no collision-free path can be determined, a tool change is executed if possible (Figure 8). To select a new tool (from the short list for that bend line), a number of rules are followed:

- minimize the number of tool profiles to be used for producing the part;
- maximize common preference for tools in a company;
- minimize the chance for sub-collision due to selection of a certain tool.

A penalty system ensures that no redundant changes are made. The final result of the approach generates a collision-free bend sequence together with the different tools to be used for each bend stroke. The interested reader is referred to Nguyen (2005). For a comprehensive overview of other process planning issues, see Duflou et al. (2005).

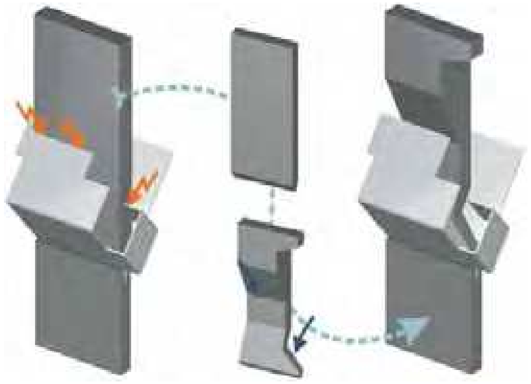


Figure 8. Selecting the appropriate bending tooling to avoid collisions (Nguyen, 2005)

After determining the bend sequence and the tools, the physical layout of those tools on the press brake needs to be generated. A shrewd choice of the position of the different stations minimizes the total distance traveled for the operator during the bending process. Next section elaborates on this topic.

3. Tooling layout on a press brake for sheet metal air bending

3.1 Problem formulation

This section discusses the practical planning problem of locating all the required stations for a specific workpiece on a press brake. Process constraints and objectives regarding efficiency are taken into account. The n stations are mounted on a single press brake, their configuration is called the production layout (Figure 4). A station i has a length w_i and l_i and r_i are the free spaces required respectively on the left and on the right side of station i. This free space depends on the bending operations assigned to the station, the intermediate shapes and the changing dimensions of the sheet metal part throughout the bend sequence. With the bend sequence S, a row of m station numbers is linked, specifying the order of the stations on which the m bending operations have to be executed. The actual location of all stations required for a specific workpiece on a press brake is determined by minimizing the total distance an operator has to travel during the processing of a specific work order. The main objective is to construct a fast algorithm for solving this problem because in a typical industrial environment, only a few seconds of computational time can be made available. Since exhaustive enumeration requires too much computational time, a neighborhood search method is suggested.

The stations have to be located on a single line: the stations will be placed along the z-axis of the press brake. Experience with a large number of sheet metal parts allows to conclude that n=10 might be a safe upper limit for the number of required stations. Looking at the stations along the line, from the left to the right, the sequence of station numbers is a permutation of the n stations. Once a permutation is determined, the z-coordinate z_i of each station, indicating the position of the middle of the station on the z-axis, is easily calculated.

For any two stations i and j where i is located to the left of j, the following constraints have to be satisfied:

$$z_j \geq z_i + w_i/2 + w_j/2 + l_j \tag{1}$$

$$z_i \leq z_j - w_j/2 - w_i/2 - r_i \tag{2}$$

The quality of a specific layout can be calculated by the total distance traveled by an operator when he is executing a bend sequence:

$$\sum_{i=1}^{m-1} | z_{S_i} - z_{S_{i+1}} | \tag{3}$$

where S_i and S_{i+1} indicate the station numbers of two consecutive bending operations. Because this total distance traveled should be as small as possible, the stations will be placed as close as possible to each other, leaving no unnecessary space.

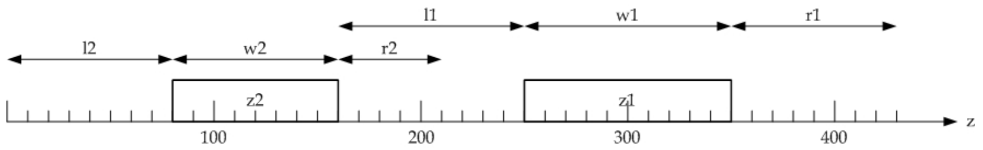


Figure 9. Example of a station layout on a press brake

Figure 9 shows an example with n=2 stations. The parameters for station 1 and 2 are w_1=100, l_1=90, r_1=70 and w_2=80, l_2=80, r_2=50, respectively. In this layout, station 2 is placed at the first position and station 1 at the second position. From these relative positions, the absolute positions z_i can be calculated:

$$z_2 = l_2 + w_2 / 2$$

$$z_1 = z_2 + w_2 / 2 \ + \ \max(r_2, l_2) + w_1 / 2$$

For a bend sequence S=[1,2,1], the total distance traveled is equal to |300-120| + |120-300| or 360. Note that when the two stations exchange positions, the z_i coordinates have to be recalculated (z_1=140, z_2=310). This is because the free space needed on the left of a station can be different from the free space required on the right side. Hence, the total distance traveled is not necessarily equal to the one of the first layout (|140-310| + |310-140| or 340).

3.2 Solution approaches

In the traditional approach no computer is used for determining the relative positions of the different stations on a press brake, only machine deformation considerations are taken into account. For the sake of symmetry and because both the table and the ram of a press brake are deformed during bending operations, the station for the longest bend line is located as centrally as possible. The second and third largest station are placed next to this largest station, one to the left and the other to the right, and this process is continued until all stations are placed. As a result, when going along the line of the stations from the left to the right, the stations become longer and longer until the longest station is reached; then they become smaller again until the end of the line is reached. This placing technique results in $2^{|n/2|}$ possible solutions, because for each next pair of largest stations two alternatives exist: one station to the left of the partial layout and the other to the right and vice versa. The number of stations on a press brake is limited to 10, thus the maximum number of different solutions is equal to 2^5 = 32. This is a very small number and all the solutions can easily be generated by a computer by exhaustive enumeration. This approach is called the technical approach (EnumT).

In a second approach, the length of the stations is not taken into account for determining the sequence of the stations. All possible solutions are listed by generating the n permutations of the set {1, 2, …, n}. For this optimal approach (EnumP) the minimal change order algorithm is used (Kreher & Stinson, 1999).

In the third approach, the deformation of both the table and the ram of the press brake is again taken into account. This deformation is primarily caused by the largest stations and consequently, they should be placed as centrally as possible. The other, shorter stations can be placed freely along the line aside these few large stations. Parameter n_w indicates the number of largest stations that have to be fixed in the middle of the line. For practical use, the number of largest stations that should be placed as centrally as possible, can be calculated taking into account the geometrical aspects of the workpiece and the derived required forces during the bending operations.

The solution procedure for this hybrid approach (EnumC) is a combination of the two procedures of the first two approaches. Permutations (Σ) for the n_w largest stations are generated by a procedure similar to EnumT and these stations are placed as centrally as possible. For each generated permutation, a procedure similar to EnumP is used to generate all permutations (Π) for the ($n-n_w$) shorter stations. The first part of a permutation Π is placed to the left of Σ and the rest to the right. In total, $2^{nw/2}$ $(n-n_w)!$ alternative layouts have to be generated, except when n is even and n_w is odd. In that case, the number of alternatives is twice this value.

The computational effort required to solve the problems related to the optimal and the hybrid approach grows very fast with the size of the problem. Therefore, descent methods are developed to solve these problems. For the optimal approach, the natural representation is a permutation of the integers (1, …, n) with n the number of stations. On this representation, two basic neighborhoods can be defined. With insert (INS) a station is removed from one position in the sequence and inserted at another position (either before or after the original position). General pairwise interchange (GPI) or "swap" swaps two stations. A special case is API adjacent pairwise interchange where the two stations are adjacent. For the order in which the neighborhood is searched, a fixed natural lexicographic ordering is used, i.e. (1,2), (1,3), …, (1,n), (2,1),(2,3),…, (i,j), …, (n-1,n), with i and j the two station numbers (for GPI only pairs where i<j are considered). Each time an improving move is executed, the next iteration restarts at the beginning of the ordering.

Different initial solutions were investigated. With at random, the first solution is equal to the permutation of the different stations in numerical order, i.e. (1,2, …, n). In the most visited initial solution, the station that is visited most frequently is placed in the middle of the line. The others are placed to the left and to the right of this station in order of descending frequency of use. The start bend sequence seed is created in accordance to the sequence of bending operations. This can be a promising initial solution when most stations are visited only once. The end bend sequence seed is analogous to the previous method, but the search is started at the end of the bend sequence and continues in reverse order through the sequence. For each of these four initial solutions, an additional initial seed is considered by using the generated solution in reverse order. By performing multiple runs of the procedure starting from the eight different initial seeds and taking the best sequence as final solution, we get a multi-start descent solution approach.

For the hybrid approach, only a descent method for the placement of the shorter stations is considered. In practical situations, the number of largest stations n_w is small and all corresponding permutations can easily be generated by exhaustive enumeration. The neighborhood is a variation of the swap neighborhood defined for the optimal approach: only stations from the first part (before the fixed largest stations) and from the last part (after the fixed largest stations) are considered for a pairwise interchange. The initial seed is a layout that satisfies the specifications of the technical approach. The fixed largest stations are already positioned as centrally as possible. Around this fixed part, the other stations are added in an order from long to short.

3.3 Computational experience

For the computational tests, the enumeration algorithms and the descent techniques were coded in C and run on a HP 9000/L1000 computer. The data set (with 48 instances) used to test the different procedures is at random data offering a good approximation for real instances. An optimal solution value to each problem is obtained with the enumeration algorithm EnumP.

For problems with up to n=8 stations, this exhaustive enumeration requires less than one second of computational time. The longest computational time is used for problems with n=10 stations, i.e. nineteen seconds. The performance of the technical approach and the hybrid approach and of the heuristics for the optimal approach is compared by listing the number of times (out of 48) that an optimal solution is found (NO), the average relative percentage deviation (ARD) of the solution value from the optimal value and the maximum relative percentage deviation (MRD) from the optimal value.

Table 1 compares the results when using the technical approach or the hybrid approach (with n_w the number of fixed largest stations) instead of the more ideal station layout based on the optimal approach. It is clear that a lot more distance has to be traversed when a layout based on the first approach is used. Note that the ARD and MRD performance measures are not an indication for the bad performance of the solution procedures. They just give information about how much the layouts based on the technical or hybrid approach deviate from the optimal layout because of the additional process constraints. For some instances, this distance is more than twice the distance resulting from a layout based on the optimal approach.

The results of Table 1 indicate that it is worthwhile to consider the hybrid approach, where only a few large stations are fixed as centrally as possible. In most practical situations, fixing the largest station in the middle is adequate to prevent an asymmetrical deformation within one bend line. The column with label n_w=1, shows that the total distance traveled is on average raised with 10%.

	EnumT	EnumC		
		n_w=1	n_w=2	n_w=3
NO	6	14	6	6
ARD(%)	54.84	10.43	24.36	37.60
MRD(%)	259.67	46.60	66.30	135.60

Table 1. Results of the enumeration algorithms

When more large stations have to be fixed in the middle of the line, the additional distance that has to be traversed, increases. Hence, it is important that stations that cannot cause any significant deformation, are freely placed on the line in order to minimize the total distance traveled as much as possible.

The left part of Table 2 shows the performance of the multi-start descent method on the optimal approach considering the three neighborhoods, adjacent pairwise interchange (API), general pairwise interchange (GPI) and insert (INS). This method requires less than one second for each instance of the data set.

	Optimal approach			Hybrid approach		
	API	GPI	INS	n_w=1	n_w=2	n_w=3
NO	29	39	47	29	35	42
ARD(%)	1.84	0.33	0.02	1.66	1.07	0.22
MRD(%)	13.71	5.01	0.85	13.51	14.44	4.63

Table 2. Neighborhood search results

Note that in this table the ARD and MRD performance measures are an indication for the effectiveness of the heuristic solution procedure. One of the characteristics of a neighborhood for determining the effectiveness is its size, i.e. the number of neighbors for a single solution. The sizes of the used neighborhoods are (n-1) for API, n(n-1)/2 for GPI and $(n-1)^2$ for INS. With a larger neighborhood, more diversity can be introduced in the search and this generally results in a better performance as can be observed in the left part of Table 2. The number of problems solved to optimality increases and the average relative deviation decreases when a larger neighborhood is used.

It is worthwhile to investigate the performance of the neighborhood search technique for solving problems based on the hybrid approach. For instances with n=10 stations, exhaustive enumeration requires about four seconds of computational time. In the right part of Table 2, the results of the descent method based on a swap neighborhood are compared with the values calculated with the enumeration procedure for the third approach. Quite good results are obtained: a lot of instances are solved to optimality and the average relative percentage deviation is small. The fact that only a single-start version is used, is probably the cause for the large MRD value. It is remarkable that the performance of the heuristic increases when the number of fixed stations n_w increases. With a larger n_w value, the solution space is smaller and the swap neighborhood is probably capable to search this space adequately.

3.4 Summary

This section presents some practical methods for placing a number of stations consisting of a punch and a die, on a press brake for sheet metal air bending. Comparing the best values of the total distance traveled according to the technical approach and to the optimal approach, indicates that a lot of traveling distance can be saved when the tooling layout is based on the optimal approach. Yet, when very large stations have to be used for producing a workpiece, one cannot ignore the asymmetric machine deformation. Therefore, a third approach is suggested. In this approach only the largest stations are placed as centrally as possible and the other stations are freely added to the left and to the right of these largest stations. The computational results indicate that layouts based on this hybrid approach are to be preferred, especially when only one large station has to be fixed in the middle. Apparently, a simple neighborhood search technique gives good results.

After the physical layout of the different stations on the press brake is determined, the part can be produced with air bending. Production planning for air bending is usually carried out by an experienced operator. Makespan reduction is the most important criterion in this process. Since interchanging production layouts is time consuming, such set-ups should be avoided as much as possible. Section 4 discusses the authors' effort to automate production planning for press brakes and to improve on the schedules of an experienced production planner.

4. Automated production planning of press brakes for sheet metal bending

4.1 Problem formulation

For bending, every part requires a specific production layout and the time to produce this part depends on the properties of the layout. Changing production layouts is time consuming and should be avoided. Fortunately, some production layouts can be used for several parts. For instance a bend line with length (k) can be made with a tool set length (k+x) if that bend line does not contribute to forming a box-type part. The other way around is impossible: a bend line with length (k) cannot be made with a tool set length (k-x). Table 3 presents a small example with four jobs and six production layouts. Job 1 can be processed using production layout a in 100 seconds, using production layout d in 130 seconds, or using production layout f in 120 seconds. With production layout d it is possible to produce jobs 1, 2 and 3. Table 4 summarizes the set-up times between the different production layouts; the matrix is inherently an asymmetric one.

The different possible production layouts per job and the different possible production layouts for combinations of jobs can be generated by a computational procedure (see e.g. Duflou et al. 2003) starting from the complete (initial) set of jobs. The different manipulations during set-up from a particular layout to another one can be analyzed; standard time and motion studies will provide set-up time estimates. Furthermore, time and motion analysis allows for the calculation of production times (Vansteenwegen & Gheysens, 2002).

	Production layout					
Job	a	b	c	d	e	f
1	100	-	-	130	-	120
2	-	60	-	90	-	80
3	-	40	-	70	50	-
4	140	-	120	-	60	-

Table 3: Feasible production layouts per job and corresponding production times (seconds)

Production layout	end	a	b	c	d	e	f
start	-	72	58	48	79	53	48
a	55	-	74	136	38	324	0
b	30	128	-	22	184	90	210
c	36	40	40	-	70	50	164
d	63	140	32	38	-	60	112
e	35	200	152	34	30	-	110
f	35	20	90	38	118	18	-

Table 4: Set-up times between the production layouts (seconds)

Currently, production planning practices for bending are based on experience. The press brake operator receives a work list with parts to bend. Based on his knowledge and skill, the production layouts are selected and the parts scheduled. Until now, no assisting software packages are available on the market. Typically, the task of a production planner should be:

- select for each job a production layout, minimizing the total production time;

- sequence the different jobs at the press brake, minimizing the makespan for the pool of jobs.

For the example (Table 3 and Table 4), a feasible solution is layout sequence [e - f] with jobs 1 and 2 assigned to production layout f, while jobs 3 and 4 are assigned to production layout e. The makespan for this small set of orders equals 53 + (50+60) + 110 + (120+80) + 35 = 508 seconds. Apparently, the Press Brake Planning problem (PBP) has a very specific structure. Two well known models from the literature, the traveling purchaser problem (TPP) and the generalized traveling salesperson problem (GTSP) can capture this structure. Both modeling approaches are investigated to verify whether production planning for air bending can be automated, minimizing the makespan, or not. The computational requirements and quality of the final solution are determinant factors when comparing the two methods. Next subsections discuss both lines of action.

4.2 The traveling purchaser problem (TPP)

Ramesh (1981) describes the TPP as a generalization of the traveling salesperson problem (TSP). An agent must visit a number of markets/cities in order to buy, at minimum cost, a set of items. The cost consists of two elements: the travel cost between the markets and the purchase cost of the items.

The production planning problem for air bending is now reformulated, using following parameters and variables:

- i,j: indices for markets
- k: index for the product
- c_{ij}: travel cost when traveling from market i to market j
- h_{ik}: purchase cost for product k at market i
- x_{ij}: binary variable indicating whether or not the agent travels from market i to j
- y_{ik}: binary variable indicating whether product k is bought at market I
- I: all cities/markets, where 0 is the starting position of the agent (I_0 = I \ {0})
- K: all products

$$Min \quad \sum_{i=0}^{|I|}\sum_{j=0}^{|J|} c_{ij}x_{ij} + \sum_{i=1}^{|I|}\sum_{k=1}^{|K|} h_{ik}y_{ik} \tag{4}$$

$$s.t.$$

$$\sum_{i=1}^{|I|} y_{ik} = 1 \;\; \forall k \in K \tag{5}$$

$$y_{ik} \leq \sum_{j=0}^{|I|} x_{ij} \quad , \forall i \in I_0 ; \forall k \in K \tag{6}$$

$$\sum_{j=0}^{|I|} x_{ij} = \sum_{j=0}^{|I|} x_{ji} \quad , \forall i \in I \tag{7}$$

$$\sum_{i=1}^{|I|} x_{0i} = 1 \tag{8}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \le |S| - 1,\ S \subseteq I_0 \tag{9}$$

$$x_{ij} \in \{0,1\},\ \forall (i,j) \in I, \forall k \in K \tag{10}$$

In formulation (4)-(10), the objective function (4) minimizes the total cost. Constraints (5) ensure that all products are purchased. Constraints (6) allow to buy a product only when a market is visited. Constraints (7) are the balancing constraints: a market entered has to be left as well. Constraint (8) indicates the tour starts at the starting point. Constraints (9) are the subtour elimination constraints and constraints (10) limit the variables to Boolean values. The example from Table 3 is represented as a TPP instance in Figure 10. The squares (cities) represent the production layouts, the circles (items) the jobs. The traveling costs between the cities are the set-up times between the production layouts, while the purchase costs are the production times of the parts, given a specific production layout. Some parts or jobs (circles) are connected by arrows to several production layouts (squares) since some jobs can be produced with a few production layouts. The dotted arrows represent a feasible job allocation: only two production layouts are chosen, e and f. Jobs 3 and 4 are produced with layout e; jobs 1 and 2 are produced with production layout f. In total, the production time equals 310 seconds and the total set-up time is 198 seconds. This results in a makespan of 508 seconds. Several procedures are available to solve the TPP. The interested reader is referred to Ramesh (1981), Pearn (1991) and Singh & Van Oudheusden (1997).

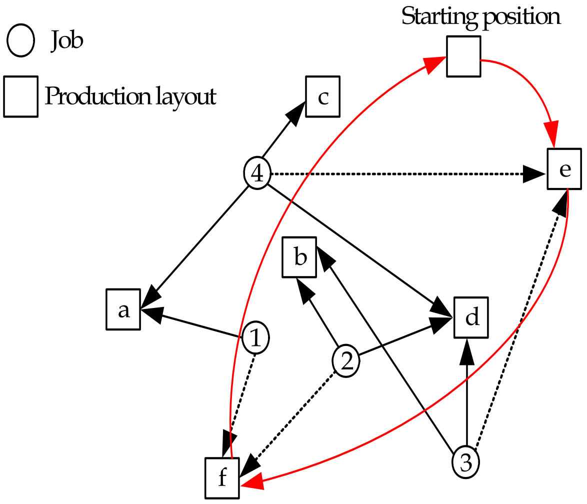


Figure 10. TPP presentation of the example

Heuristic solutions of the TPP can be generated in different ways, depending on the emphasis put on either production or set-up times. If one only wants to minimize the number of set-ups a set covering formulation can address the problem (E. El-Darzi & G. Mitra, 1995). Generally speaking this approach will yield poor results with regard to makespan since production and set-up times are not taken into account. For the example presented in Tables 3 and 4, many solutions with only two set-ups appear, i.e. -a-b-, -b-a-, -a-d-, -d-a-, -e-f-, -f-e-, -c-d-, and -d-c-. But the makespan varies quite a lot: respectively 516, 581, 573, 674, 508, 411, 591 and 563 seconds.

A much better approach, fitting the intrinsic TPP structure, would be a hierarchical decomposition procedure. In a first step a simple plant location problem (SPLP) is solved for instance by means of the procedure of Erlenkotter (1978), using average set-up times between the production layouts as the fixed costs for the different plants. In this way each workpiece is assigned a production layout, minimizing the total number of layouts for the pool of workpieces. In a second step a TSP is solved to determine the sequence for bending the different parts. Real set-up costs between the production layouts are used as the traveling costs.

4.3 The generalized traveling salesperson problem (GTSP)

The GTSP (Srivastava et al., 1969) is the problem of finding a minimum length tour through a predefined number of subsets of customers while visiting at least one customer in each subset. The node set N consists of m mutually exclusive node sets S_I such that N = $S_1 \cup S_2 \cup \ldots S_m$ and $S_I \cap S_J = \varnothing$ for all i and j (i ≠ j). Assume that arcs are defined only between nodes belonging to different sets. The objective is to find a minimum length tour visiting a node in every set. Following parameters and variables will be used in the formulation:

- i,j: indices for the cities
- c_{ij}: travel cost when traveling from city i to city j
- x_{ij}: binary variable indicating whether or not the salesperson travels from city i to j
- A: all arcs connecting two cities
- N: all nodes

$$Min \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{11}$$

$$s.t.$$

$$s.t. \sum_{i\in S_I} \sum_{\substack{j\notin S_I \\ (i,j)\in A}} x_{ij} = 1,\ \forall S_I \tag{12}$$

$$\sum_{i\notin S_I} \sum_{\substack{j\in S_I \\ (i,j)\in A}} x_{ij} = 1,\ \forall S_l \tag{13}$$

$$\sum_{\substack{i\in N \\ (i,j)\in A}} x_{ij} - \sum_{\substack{k\in N \\ (i,j)\in A}} x_{jk} = 0,\ \forall j\in N \tag{14}$$