

板金製品の加工性評価のための高速な曲げ手順生成アルゴリズム

茨城大学 工学部 システム工学科

助教授 乾 正知

(平成7年度奨励研究助成 AF - 95037)

1. 研究の目的と背景

高品質の板金製品を低コストで製造するためには、製品の加工性を十分に考慮した設計をおこなう必要がある [2]。多くの板金製品は、板金をV字型の金型に固定しプレスする曲げ加工により製造される (図1参照) [4]。したがって製品の加工性を評価するためには、その製品が逐次的な曲げ加工により適切に製造できること、具体的には、製品形状が曲げ加工により正確に実現できること、加工中に工具と板金間に干渉が生じないこと、加工に要するコストが十分に低いことなどを検証すればよい。

板金製品の加工性に最も悪い影響を与えるのは、曲げ加工中の板金間の自己干渉や板金と工具の干渉である。そこで本研究では、板金製品の幾何モデルに基づいて、干渉を生じない全ての曲げ加工手順を自動生成する問題を考える。このような手順が得られれば、その結果に基づいて加工精度の検証や作業コストの推定を行なうことで、製品の加工性のより正確な評価が実現できる。

この問題は、製品の曲げ部を全て展開した状態の幾何モデルを作成し、これに曲げ加工のシミュレーションを順に施し、干渉を生じない手順を探索すれば解くことができる。しかし板金製品の曲げ手順は、曲げ部の数を n とすると最大 $n!$ 通り存在するので、素朴なアルゴリズムでは、複雑な製品の曲げ手順を全て探索することは困難である。

本論文では、干渉を生じない全ての曲げ手順を高速に算出する新しいアルゴリズムを開発したので報告する。曲げ部の多い複雑な板金製品ほど、加工中に板金や工具間に干渉を生じやすいので、曲げ部の数が増えても可能な曲げ手

順はそれほど爆発的には増加しない。したがって、これらの干渉の有無を効率的に判定し探索空間の縮小に利用すれば、問題解決のための手間をかなり削減できる。特に本研究では、過去に行なわれた干渉判定の結果をキャッシュに記録しておき、手順の探索中にその結果を再利用することで、高速な処理を実現した。提案したアルゴリズムを用いて、干渉を生じない全ての曲げ手順を算出するプログラムを作成し、計算実験によりその有効性を検証した。

2. 板金製品の幾何モデル

曲げ加工中の板金の自己干渉や工具との干渉を判定するためには、加工にともなう板金の変形を正しく扱える必要がある。本節では、曲げによる変形を考慮した板金製品の幾何モデルについて説明する。このモデルは文献 [3] と [6] に基づいている。

2.1 板金モデル

本研究では板金の加工法として、板金をある直線にそって折り曲げる一般的な方法のみを考え、丸め加工やカール加工などの特殊な方法は考慮しない。また、既に展開状態に切り抜かれた板金を曲げていくだけの工程を想定し、曲げ加工と抜き加工を併用するような作業は考えない。

このような加工では、変形は加工力が作用する曲げ軸の近傍にのみ発生し、残りの大部分の形状は変化しない。この曲げ加工の特徴に基づいて、板金製品を曲げ軸の近傍とそれ以外の変形を生じない板状の部分に分けてモデル化した。曲げ軸近傍の領域はごく狭いので、干渉判定では無視できる。そこで曲げ加工による板金製品の形状変化を、ある仮想的な軸周りを板状の薄い立体モデルが回転移動することで近似した。

曲げ軸近傍では、板金の外側は引き延ばされる方向に、また内側は圧縮される方向に応力が作用する [1]。したがってこの部分の切断面を考えると、応力が全く作用しない「中立面」とよばれる面が存在する。この面上では変形が生じないことから、中立面は加工前の素材の長さを維持している。このことを考慮すると、仮想的な回転軸 Ax は、図2に示した式を解くことで決められる。中立面の位置を示す係数 k は板金の材質や厚さによって異なるが、通常 $k = 0.3$ 程度である。図中の回転角が $\psi = 0$ であれば展開状態、 $\psi =$

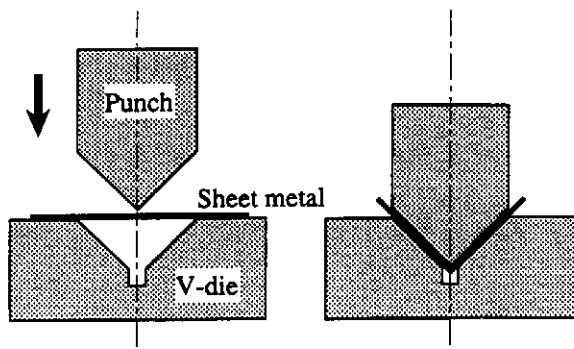
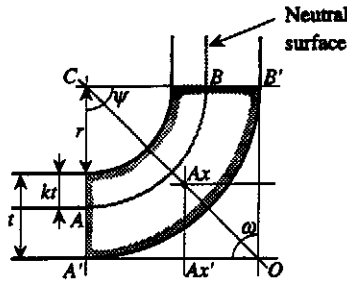


図1 Bending method with a V-shaped die and a punch



$$\begin{aligned}
 AB &= (r + kt) \frac{2\pi\psi}{360} \\
 A'Ax' &= \frac{AB}{2} \\
 A'O &= (r + t) \tan\left(\frac{\psi}{2}\right) \\
 CO &= \frac{r+t}{\cos\left(\frac{\psi}{2}\right)} \\
 CAx &= CO \frac{A'Ax'}{A'O}
 \end{aligned}$$

図2 Definition of the centerline of bending rotation

180 - ω (ωは曲げ角度)であれば曲げ加工された状態が表現される。

以後の議論では、仮想的な回転軸をベンドとよび、板状の薄い立体モデルをプレートとよぶ。各ベンドの左右には二枚のプレートが接続する。このベンドとプレートの接続関係はループを作らない。したがって板金モデルSは、プレートの集合P={P_i}と、それらを結合するベンドの集合B={B_i}から構成される木構造S=(P, B)となる [6]。

2.2 座標変換による曲げ加工の扱い

板金モデルS=(P, B)の形状を表現するためには、モデルを構成する全てのプレート∀P_i∈Pを、基準座標系F_W中の適切な場所に配置すれば良い。そこで各プレートP_i上にプレート固有の座標系F_{P_i}を定義し、F_WにおけるF_{P_i}の位置姿勢を表す同次座標変換H_{P_i}^Wを与えることでP_iを位置決めする [5]。ベンドB_iに2枚のプレートP_jとP_kが接続しているだけの単純な構造の板金モデルを用いて、各プレートの座標変換の計算手順を説明する。

まずB_iが展開状態の場合を考える。プレートと同様に、ベンドB_iにも固有の座標系F_{B_i}を与える。ただしF_{B_i}は、そのz軸が仮想的な回転軸と一致するように定義されている。P_jの座標系F_{P_j}における、B_iの座標系F_{B_i}の位置姿勢を表す同次座標変換をH_{B_i}^{P_j}とする。同様にP_kの座標系F_{P_k}における、F_{B_i}の位置姿勢を表す座標変換をH_{B_i}^{P_k}とする。これら二つの座標変換は製品の展開図から容易に得られる。H_{B_i}^W、H_{P_k}^W、H_{B_i}^{P_k}、H_{B_i}^{P_j}の間には以下の関係式が成り立つので、H_{P_j}^WかH_{P_k}^Wのどちらかが分かれば、他方の変換は式を変形することで計算できる。

$$H_{P_j}^W H_{B_i}^{P_j} = H_{P_k}^W H_{B_i}^{P_k} \quad (1)$$

次にベンドB_iに曲げ加工が施され、B_iに接続する2枚の

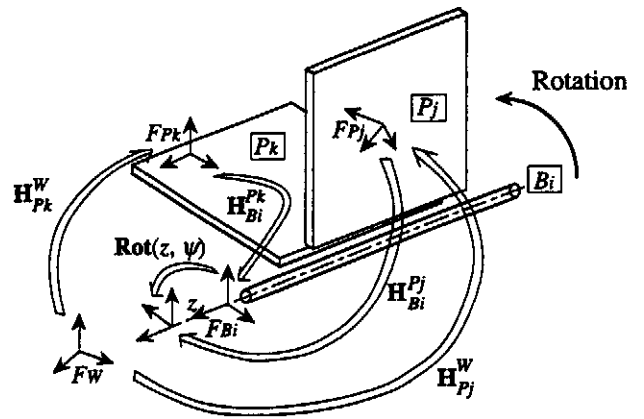


図3 Spatial relationships between coordinate frames specified on a bend B_i and its connecting plates P_j and P_k

プレートP_jとP_kのなす角度がω度となった場合を考える。われわれのモデルでは、この変形をP_j (もしくはP_k) が、B_iの座標系F_{B_i}のz軸周りをψ = 180 - ω度回転移動することで表現する。z軸周りの回転変換をRot(z, ψ)と書くことにすると、座標変換間に以下の関係式が成り立つ (図3参照)。

$$H_{P_j}^W H_{B_i}^{P_j} = H_{P_k}^W H_{B_i}^{P_k} \text{Rot}(z, \psi) \quad (2)$$

したがって展開状態の場合と同様に、H_{P_j}^WかH_{P_k}^Wのどちらかが分かれば、他方の変換は式 (2) から計算できる。

一般の板金モデルS=(P, B)は、プレートの集合Pとベンドの集合Bをそれぞれ頂点と辺とする木構造である。そこでまず一枚のプレートP_i∈Pを選び、H_{P_i}^Wとして適切な座標変換を与えることでその位置姿勢を確定する。次にベンドを介してP_iに接続する他のプレートをたどる。その際ベンドが展開状態であれば式 (1)、曲げ状態であれば式 (2) を用いて、たどった先のプレートの位置姿勢を確定する。以上の処理を、全てのプレートをたどり終えるまで繰り返せば、任意の曲げ状態の板金製品の形状が得られる。

3. 曲げ手順の算出アルゴリズム

板金製品の曲げ手順を算出するアルゴリズムとしては、製品の展開状態の幾何モデルを入力し、ベンドを一つずつ折り曲げては干渉の有無を確認し可能な手順を調べる、前向き探索に基づく方法がまず考えられる。同じことは、完成状態の幾何モデルのベンドを一つずつひろげ、可能な展開手順を調べる後ろ向き探索でも実現できる。後ろ向き探索の場合には、得られた手順の逆が曲げ手順となる。

板金製品では、完成状態に近付くほど狭い空間内にプレートが密に集まるので、折り曲げや展開の際に干渉が生

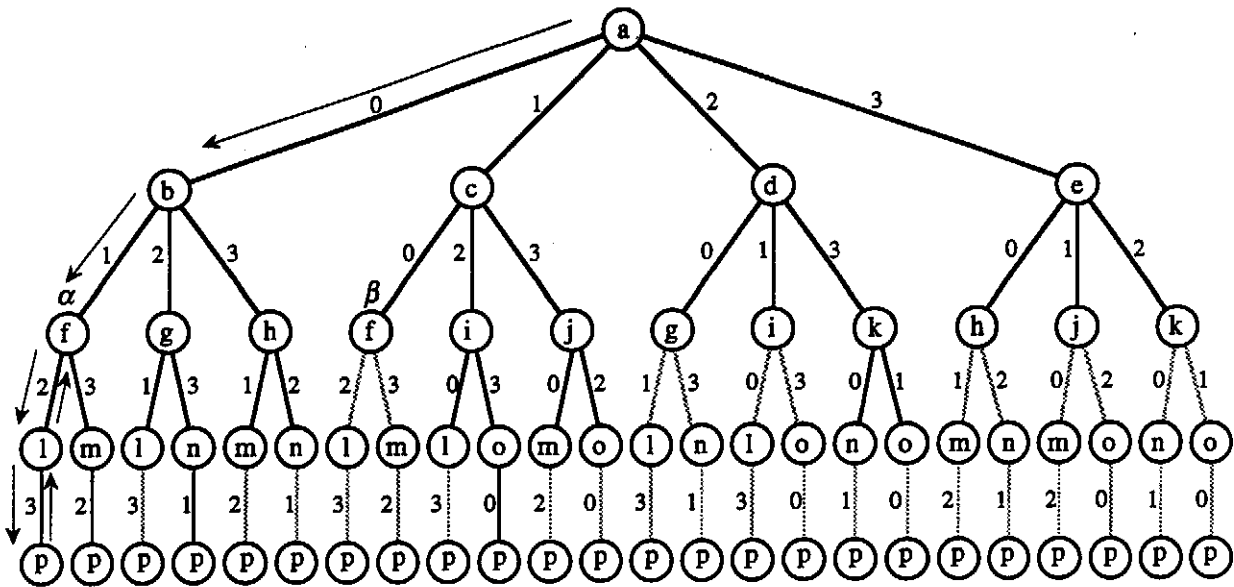


図4 A backward search tree of a sheet metal part with 4 bending portions

じやすい。したがって後ろ向き探索を採用したほうが、探索の初期の段階で問題を生じる手順を検出し破棄できるので効率が良い。そこで以後は、後ろ向き探索に基づいて干渉を生じない全ての展開手順を調べるアルゴリズムについて議論する。

n 個のバンド $B_i, i \in [0, n-1]$ からなる板金製品の全展開手順は、展開されるバンドが各辺に対応する木構造で表現できる。図4には $n=4$ の場合の木構造を示した。ただし図中の各辺には、展開されるバンド B_i の添字のみを示した。木構造の根は製品の完成状態を表し、末端の節は完全に展開された状態の板金を表す。残りの各節は、根からその節に至るまでの辺に対応するバンドを順に展開することで得られる、中間状態の板金を表す。図中の節に付けられている記号や辺の濃淡の意味については後述する。

干渉を生じない展開手順を得るには、この木構造の全ての節を深さ優先の規則にしたがって探索すればよい(図4中の矢印参照)。辺を下向きにたどる時には、対応するバンドを展開し自己干渉や板金と工具の干渉を調べる。もしも干渉が検出された場合には、展開を中止しその辺以下の節の探索を止める。逆に辺を上向きにたどる時には、対応するバンドを折り曲げ、展開前の状態を復元する。木構造の末端の節にたどりついた時、根からその節までの経路が可能なバンドの展開手順を表す。例えば図の左端の経路は、 $B_0 \rightarrow B_1 \rightarrow B_2 \rightarrow B_3$ という手順でバンドが展開可能なことを意味する。以上の処理を繰り返すことで、干渉を生じない全ての展開手順が得られる。

4. 干渉情報のキャッシュによる高速な干渉判定

板金モデルのバンドの一つを展開すると、前述の座標変

換の手順にしたがい、モデルを構成するプレート群がそのバンドの周りを回転移動する。展開手順を高速に計算するためには、移動するプレートと他のプレートの衝突や、プレートと曲げ工具の干渉を効率的に判定する必要がある。本節では、過去の干渉判定の結果をキャッシュに記録し再利用することで、干渉判定の回数を減らし処理を高速化する手法を説明する。

4.1 基本的な考え方

プレート間の衝突やプレートと工具の干渉を厳密に判定するためには、回転移動するプレートの掃引形状に相当する立体モデルを生成し、これと他のプレートや工具の立体モデルの交差を調べればよい。しかし掃引立体の生成には大きな手間を要するので、この方法は高速な判定には向かない。そこで本研究では、 $\psi=0$ (展開状態)と $\psi=180-\omega$ (折り曲げ状態)間の幾つかの離散的な回転角 ψ において、回転移動したプレートと他のプレートや工具モデルの交差を調べる、より簡易な判定法を採用した。

図4を見ると、曲げ手順の探索中に同じ干渉判定を何回も繰り返していることが分かる。たとえば探索木中の記号 α を付した節は、バンドを $B_0 \rightarrow B_1$ の順序で展開した状態の板金を表している。一方、同じ段の左から4番目の節(記号 β を付した)は、バンドを $B_1 \rightarrow B_0$ の順序で展開した板金に対応する。 B_0 を展開し次に B_1 を展開しても、先に B_1 を展開し次に B_0 を展開しても、得られる板金の形状は同じである。したがってこれらの状態で、さらにバンド B_2 や B_3 を展開した場合の干渉判定の結果も同一となる。そこで本研究では、バンドを展開する際の干渉判定の結果をキャッシュに記録しておき、以後、幾何的に同一な条件でのバンドの展開が要求された時には、記録してある判定結果を再利用するこ

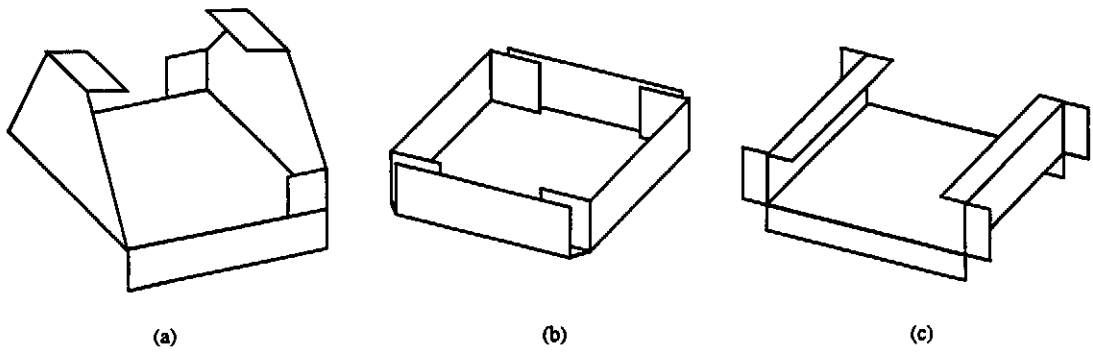


図5 Example sheet metal parts

とで、干渉判定そのものを省略することにした。

4.2 キャッシュの効果

図4に示した展開手順の木構造を用いて、キャッシュの効果を検討する。図中の各節には、対応する板金の状態を識別するために、a (完成状態) から p (展開状態) までの記号を付した。板金の状態は、根からその節までの経路に対応する展開済みのベンドの組み合わせにより決まり、展開順序には依存しない。したがってベンドの組み合わせが同一の節には、同一の記号が付けられている。

この木構造を深さ優先で探索する時に、幾何的に同一な条件でのベンドの展開が繰り返される場合の、2回目以降の展開に対応する辺を灰色で表示した。このような展開では、キャッシュに記録された過去の干渉判定の結果が再利用できるので、干渉判定そのものを省略できる。過去の判定結果を利用しない時には、最悪の場合、図中の全ての辺において干渉判定を行なうことになるので、その回数は64回となる。一方再利用する場合には、図4中の実線の辺でのみ干渉判定が必要となるので、その回数は32回となり半減する。

キャッシュの利用による高速化の割合は、曲げ部の多い複雑な板金製品ほど顕著である。 n 個のベンドからなる板金製品の場合、その展開手順の木構造の根を0段目とすると、 i 段目の節からのびる辺は合計 $n! / (n-i-1)!$ 本となる。干渉情報のキャッシュを使わない場合、干渉判定の回数 c は、木構造の辺の総数と等しいので、

$$c = \sum_{i=0}^{n-1} \frac{n!}{(n-i-1)!}$$

となる。

木構造の i 段目における板金の展開状態の種類は、 n 個中の i 個のベンドを展開した場合の数と等しいので $\binom{n}{i}$ 種類となる。干渉状態のキャッシュを使う場合には、それぞれの状態で残りの $n-i$ 個のベンドを展開する際に一度だけ干渉判定を行ない、後はその結果を再利用する。したがって干

渉判定の回数 c' は、

$$c' = \sum_{i=0}^{n-1} \frac{n!}{(n-i)! i!} (n-i) = \sum_{i=0}^{n-1} \frac{n!}{(n-i-1)! i!}$$

となる。

c と c' の各項を比較すれば、 $n > 2$ のとき $c > c'$ となること、また n がある程度大きくなると、 c が c' よりもはるかに大きくなることが分かる。例えば $n = 10$ の板金製品の場合、 $c = 9,864,100$ 回に対して $c' = 5,120$ 回なので、キャッシュを利用することで2,000倍近く高速化されることになる。実際には、最後のベンドの展開時の干渉判定は常に不要であり、干渉が検出され探索を途中で止めることも多いのでこれほどの差は開かないが、それでも数十倍の高速化が期待できる。

4.3 実装法

最後に干渉情報のキャッシュの実装法を検討する。干渉判定の結果を再利用するためには、任意の展開状態の板金に関する過去の判定結果が即座に検索できればよい。そこで板金モデルの各展開状態に対して、検索のキーとなるユニークなビット列を割り当てることにする。このようなビット列は、各ベンドに0から $n-1$ までの番号を与え、展開済みのベンドの番号に対応するビットには1をセットし、それ以外のビットには0をセットすることで生成できる。複雑な板金製品でも曲げ部の総数は10前後なので、ビット列には2バイト程度を用意すれば十分である。

途中まで展開された状態の板金モデルが与えられたときには、各ベンドの展開済み/未展開という情報から上述の手順でビット列を生成し、これをキーにキャッシュの内容を検索する。検索の結果、過去の判定結果が見つければそれを再利用する。見つからない場合には幾何計算により干渉の有無を判定し、その結果をビット列と関連付けて、将来の再利用のためにキャッシュに格納しておく。

5. 計算実験の結果

議論した手法に基づいて、板金製品の干渉を生じない全ての曲げ手順を算出するプログラムを、C言語を用いて実装した。実際に幾つかの板金モデルを入力し、その全曲げ手順を自動決定することで干渉情報のキャッシュの有効性を検証した。

図5には、実験に用いた3種類の板金製品を示した。開発したプログラムを用いて各例題を処理した結果を下表に示す。表中の最初の列は例題の曲げ部の数を示し、次の列は算出された曲げ手順の総数を示す。3列目と4列目は、それぞれキャッシュを利用しない場合と、利用した場合の処理に要した時間（CPU秒）を示す。なお計算には、R4600（100MHz）をCPUとする主記憶64MBのワークステーションを用いた。最後の列には、キャッシュの利用による計算時間短縮の比率を示した。

これらの結果から、キャッシュを用いることで計算時間が大幅に短縮できること、短縮の度合は曲げ部の多いものほど大きいこと、複雑な板金製品の数千種類にもおよぶ曲げ手順が、対話的な利用でも支障を感じない程度の時間で計算できることなどが分かる。これらの例以外にも、様々な形状の板金製品に関して評価実験を行ない、いずれの場合にも満足できる結果を得た。

表1 Vending sequence computation results of example parts

	No. of Bends	No. of Seq.	W/O Cache t_1 (s)	W Cache t_2 (s)	t_2/t_1
a	7	327	26.68	5.93	0.22
b	8	4704	386.65	23.75	0.061
c	10	1392	253.28	11.32	0.045

6. 結び

本論文では、板金製品の干渉を生じない全曲げ手順を高速に算出するアルゴリズムについて議論した。このアルゴリズムは、過去に行なわれた干渉判定の結果をキャッシュに記録しておき、曲げ手順の探索中にその結果を再利用することで高速な処理を実現する。提案したアルゴリズムを用いて、全ての曲げ手順を算出するプログラムを作成し、計算実験を行なうことでその性能を検証した。このプログラムは、曲げ部の個数が10を越える複雑な板金製品でも、対話的な利用で支障を生じない時間で全曲げ手順を生成できる。

参考文献

- [1] D. F. Eary and E. A. Reed : Techniques of Pressworking : Sheet Metal, Prentice - Hall, (1958).
- [2] 穂坂 衛、佐田登志夫 : 統合化CAD/CAMシステム、オーム社、(1994).
- [3] M. Inui and F. Kimura : Design of Machining Processes with Dynamic Manipulation of Product Models, in Artificial Intelligence in Design, Springer - Verlag, (1991) 195.
- [4] 大久保正雄、塩川昌男、三坂浩和 : 5章プレス加工、自動車の製造法、自動車工学全書19、山海堂、(1980) 113.
- [5] R. P. Paul : Robot Manipulators : Mathematics, Programming, and Control, The MIT Press, (1981).
- [6] C. H. Wang and D. A. Bourne : Using Features and Their Constraints to Aid Process Planning of Sheet Metal Parts, Proc. of IEEE International Conference on Robotics and Automation, (1995) 1020.