TM-0874

歯車変速機構の構造設計問題
—設計問題に対する制約問題解決に基づくアプローチ—

堀内 英一

April, 1990

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

歯車変速機構の構造設計問題
—— 設計問題に対する制約問題解決に基づくアプローチ ——

堀内 英一

ＩＣＯＴ　第５研究室

１．はじめに

１．１　中期研究の経緯

第５研究室では中期の研究課題の一つとして機械設計向け制約指向知識コンパイラであるＭＥＣＨＡＮＩＣＯＴ[1]の研究開発を行ってきた。ＭＥＣＨＡＮＩＣＯＴの開発は，制約問題解決システムの開発という位置付けで行なわれたものであり，工作機械の一つである普通旋盤の主軸部分の設計[2]を例題とした実験システムである。後期の研究課題は，開発したシステムであるＭＥＣＨＡＮＩＣＯＴにおいて初期の研究課題には含まれなかったがその後の研究の進展で研究の必要性が認められた領域が対象となる。

１．２　後期の研究課題

中期の研究結果に関する問題点に基づき，次のような研究課題を挙げることができる。第１の研究課題はシステムで利用できる設計知識の拡大である。これまでのＭＥＣＨＡＮＩＣＯＴでは主軸例題をパラメータ設計問題として扱ってきた。パラメータ設計問題では設計に必要な設計式，ＪＩＳ規格表，図表などの制約がすべてまえもって与えられていて，これらの制約を充足するようなパラメータを求めることが問題であった。そのためＭＥＣＨＡＮＩＣＯＴでは制約が陽に与えられる問題しか扱えないことが問題となった。例えば，主軸の構造を決定するような制約は物理的な常識的知識から決まってしまうような側面があり，機械設計の教科書や現場の設計者とのインタビューから獲得することが困難である。その結果，ＭＥＣＨＡＮＩＣＯＴでは主軸の構造は固定して，まえもって決めてしまうことで妥協していた。従って，構造を決定する知識を新たに追加して設計仕様に応じて主軸の構造を変化できるようにすることが課題となる。

第２の研究課題は機械設計問題を並列処理のアプリケーションとしてみたアプローチである。ＭＥＣＨＡＮＩＣＯＴにおいては機械設計問題を扱うのに適した知識表現を開発することが一つの課題であったが，その知識表現の処理の並列化の検討は行われなかった。しかし，現実のシステムで高速な処理が要求されるような場合には，処理の並列化の検討がどうしても必要となる。ここでは第１の研究課題として述べたＭＥＣＨＡＮＩＣＯＴの拡張部分についての処理の並列化の検討を行う。すなわち設計仕様が与えられたときに，主軸部分の構造を決定する処理をより高速に実行する並列化技術を考える。すぐれた並列プログラムはすぐれた並列アルゴリズムなしでは作れないので，対象とする問題を並列処理に適した形で定式化し，並列に独立して処理ができる部分を見つけることを課題とする。

１．３　研究の位置付け

ここで論じる構造設計の問題解決の研究は一つの機械設計システムに含まれる構造設計サブシステムの開発に必要な研究として位置付けることができる。機械設計システム

は構造設計／制約生成サブシステムおよび制約解析／制約処理サブシステム[3]から構成される。構造設計／制約生成サブシステムは設計対象である機械部品の構造を決定し，必要であれば新たな制約を設計問題に直接適用できるような浅い知識として機械設計の領域に関する深い知識から導く。対象とする設計問題にとって必要十分な設計知識を表す制約集合が得られたならば，この制約集合は制約解析／制約処理サブシステムに受け渡される。制約解析／制約処理サブシステムはこの制約集合を解析して変数の依存関係などを調べ，制約集合を充足するような変数の組合せを求めるという手続きにより設計問題を解くことができる。

２．問題設定

２．１　問題の定義

この報告は，まず第１の研究課題に基づき，工作機械主軸用の歯車変速機構[4][5]の構造を設計するために必要な制約を明らかにして，ある設計要求あるいは設計仕様が与えられたときに，別に定義した最適性を満たす構造が順次枚挙できるような定式化を行う。また第２の研究課題に基づき，構造を決定する処理について並列性を考慮したアルゴリズムの開発を行う。そこで工作機械主軸の歯車変速機構の構造を決定する問題を以下のように設定することにする。

【問題】歯車軸数がＭ軸，変速段数がＮ段の歯車変速機構について，与えられた最適性を満足するような構造を設計する。ここで構造を設計するとは各歯車軸上の歯車の並びの順序関係と歯車間の適切な距離とを決定することを意味する。最初に与えられる情報は，歯車軸数Ｍ，変速段数Ｎ，および歯車軸を移動させた場合に互いに衝突が生じるような相異なる隣り合う歯車軸上にある歯車の組である。

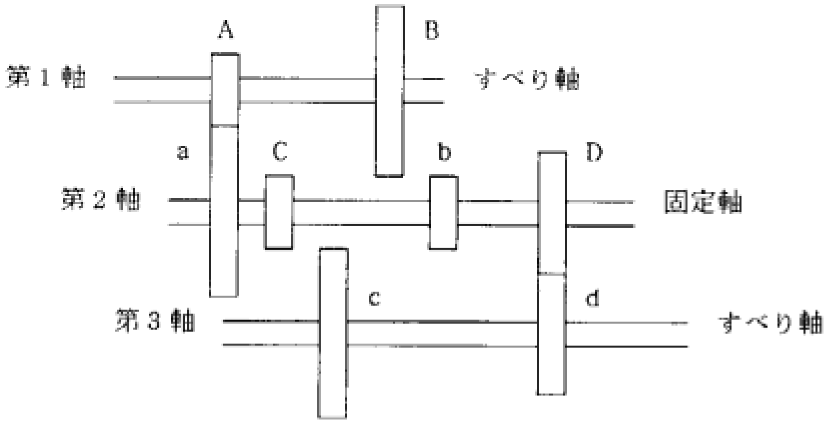


図１　３軸４段変速歯車変速機構

図１は歯車軸が３軸，変速段数が４段の歯車変速機構の例である。この例では同じアルファベットの大文字と小文字で表された歯車どうしが噛み合うことを表している。このように歯車変速機構では歯車軸のなかのいくつかの軸を固定しないで軸方向にすべり運動ができるようにして，歯車と歯車の噛み合いを変更して望ましい速度比を得るようになっている。すべり軸上にある歯車はすべり歯車と呼ばれるが，現実の設計ではすべり歯車が３個までであればそれらは一体として設計され，４個以上になれば２組に分割されることが多い。

２．２ 問題設定における仮定

この報告では問題設定にあたって次のようないくつかの仮定を置くことにする。これらの仮定は問題解決の便宜上の仮定であり問題自体の一般性を失うようなことはない。

・第１軸を外部から駆動トルクが加えられるような入力軸とし，最終軸を外部にトルクを伝達する出力軸とする。
・歯車はすべて等しい幅をもつ。
・構造の左右を逆転して互いに一致するような鏡像関係にある２つの構造は同一の構造を表すとする。

歯車変速機構の構造を決めるためにはまず各歯車軸上の歯車の個数を決める必要がある。そのためには変速段数Ｎを素因数分解して歯車軸数Ｍを考慮して各歯車軸に分配する速度列を決めなければならない。例えば変速段数が１２段であれば，１２＝２×２×３なので，歯車軸数は４とし，第１／第２軸で２段変速，第２／第３軸で２段変速，第３／第４軸で３段変速とすればよい。あるいは歯車軸数を３に減らしたければ１２＝４×３から第１／第２軸で４段変速，第２／第３軸で３段変速とすることも可能である。このような配分は設計者の意図によるところが大きいのでここでは次の仮定を置く。

・歯車軸数Ｍ，減速段数Ｎより決まる各歯車軸に対する速度列の配分は設計仕様で与えられる。

図１からわかるように歯車Ｂと歯車ｂとを噛み合わせるためには第１軸を右方向にすべらせる必要があるが，歯車Ｂと歯車ｂとの接触を越えてさらに右方向に第１軸をすべらせると歯車Ｂと歯車Ｄとが衝突することがわかる。図１の例では望ましい速度比を実現するための通常のすべり操作の途中では衝突が生じないので問題はないが，必要なすべり操作の間に本来噛み合うべきでない歯車どうしが接触したり，衝突したりするような構造は不適な構造である。

一般にどの歯車とどの歯車とが衝突する可能性があるかは，歯車の径が与えられなければわからない。例えば図１では歯車Ａはより大きな速度比を実現し，歯車Ｂはより小さい速度比を実現することが設計仕様で与えられれば，歯車Ｂの径は歯車Ａの径より大きいことがわかる。第１軸と第２軸の軸間距離は一定であるので，歯車Ａと噛み合う歯車ａの径は，歯車Ｂと噛み合う歯車ｂの径よりも大きいことがわかり，第１軸を無制限にすべらせると歯車Ｂは歯車ａと衝突する可能性があることがわかる。しかし歯車Ｂが第２軸上にある歯車Ｃ，Ｄと衝突するかどうかは，歯車の径が与えられない限り全く未知である。なぜなら，歯車Ｃ，Ｄの径は第２軸と第３軸の間の関係で決まるので，第１軸上にある歯車Ｂの径とは無関係だからである。そこで次の仮定を置く。

・相異なる歯車軸上にあって，径の大小関係により互いに衝突する可能性がある歯車の対がまえもって与えられる。

ただし，上に述べたように歯車軸数が２軸の場合は，歯車の径が与えられていなくても，衝突する可能性がある歯車の対を歯車に割り当てられた速度比から推測することができる。

次に設計を行う上での指針となる最適性を考える。この問題では歯車変速機構の全体

幅をより小さくする構造をより最適な構造であると定義する。現実の機械設計を調べてみると，変速機構を製作するための材料費や加工費，変速機構が工作機械内部で専有する空間，材料力学的な強度などが考慮されて，変速機構の全体幅をできる限り小さくするような設計が行われている。従って，この最適性の定義は一般的なものであると考えてよい。これ以外の最適性の定義を与えることも可能であるが，この報告ではそれは議論しない。

機構の全体幅を考える場合には，どの歯車軸をすべり軸／固定軸とするかも考慮しなけばならない。一般に，すべり軸の数が多いほど全体幅をより小さくできるので全部の歯車軸をすべり軸にすることも考えられる。しかし，現実の機械設計では可動部分を減らして保守・管理を容易にする理由などからそのような設計は行わず，すべり軸の数をできる限り少なくするような設計が行われている。そこで，ここでは議論を簡単にするため，すべり軸と固定軸の設定に制限を設けることにする。

・歯車軸数が奇数の場合は，奇数軸（偶数軸）をすべり軸（固定軸）とする。
・歯車軸数が偶数の場合は，奇数軸（偶数軸）をすべり軸（固定軸）とした構造と，偶数軸（奇数軸）をすべり軸（固定軸）とした構造の２つを調べて，全体幅をより小さくするような構造を選択する。

３．問題解決

３．１　問題解決の戦略

問題を解くにあたって，設定された問題を２つの部分問題に分けて考えることにする。まず第１段階では歯車変速機構のそれぞれの歯車軸の上の歯車の並びの順列だけを決定し，第２段階で歯車軸上の歯車の並びの順列を基にして，適切なスペースを歯車の間に挿入した構造を求めることにする。歯車軸上の歯車の並びの順列は具体的な数値は問題としないで空間上の相対的な位置関係のみを考えるので，以後，これを位相的構造と呼ぶことにし，位相的構造に歯車の間のスペースを付加した構造はスペースの大きさについて幾何学的な寸法が考慮されるので，以後，これを幾何学的構造と呼ぶことにする。

構造問題をこのように２つの部分問題に分割したのは，各段階で解の候補の効率的な刈り込みを行うためである。なぜなら，いま考えている歯車変速機構の構造設計問題は典型的な組合せ問題であり，少し複雑な設計仕様が与えられただけで組合せ爆発が生じて探査すべき構造の候補の数が手に負えなくなってしまうからである。

３．２　位相的構造

第１段階では歯車変速機構の位相的構造の候補を表す歯車の並びの順列を枚挙する。この段階では位相的構造を調べることにより，歯車の衝突なしに仕様で指定された歯車の対を噛み合わせることができるかどうかを検査して，条件を満たさない位相的構造の候補は棄却する。例えば図２に示すように，すべり軸である第１軸上のある歯車Ａが第２軸上の歯車ａと噛み合う場合を考える。このとき第２軸上に歯車Ｂがあって歯車Ａと衝突する可能性があるとする。この場合，水平方向の相対的な位置に関して，歯車Ａと歯車ａの両方が歯車Ｂからみて同じ側にあれば，すべり歯車Ａを移動させて歯車ａと噛み合わせることができるが，歯車Ａと歯車ａが歯車Ｂからみて互いに反対側にあるときには両者を噛み合わせることができない。

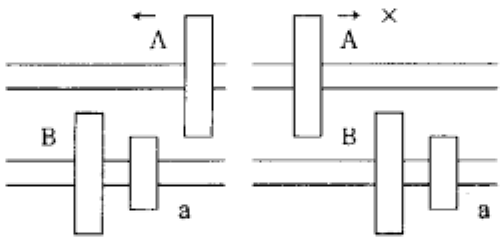


図２　衝突する可能性がある歯車の対

歯車変速機構の一つの位相的構造を与える歯車の並びの順列を枚挙するためには，一つの歯車軸上の歯車の並びの順列を生成することと，互いに隣り合う２つの歯車軸の間の組み合わせを作ること，という２つの要素を考える必要がある。例えば図３に示すように，ある歯車軸Ｉ上に歯車Ａ，Ｂ，Ｃが並ぶ場合には，その歯車軸上の歯車の並びの順列は，ＡＢＣ，ＢＡＣ，ＣＡＢなどとなる。その歯車軸と隣り合う別の歯車軸Ｊがあり，歯車Ｘ，Ｙ，Ｚがその上に並ぶとき，同様にＸＹＺ，ＺＸＹなどの歯車の並びの順列が可能である。ところが，歯車軸ＩとＪの組合せ方を考える場合には互いに衝突する可能性がある歯車の対が必要になる。第Ｉ軸の歯車Ａが第Ｊ軸の歯車Ｘと衝突する可能性がある場合には，水平方向の相対的な位置に関して，歯車Ａは歯車Ｘの右側にあるか左側にあるかのいずれかである，というのが基本的な考え方である。さらに歯車Ｂが歯車Ｘ，Ｙと衝突する可能性があり，歯車Ｘが歯車Ｙの左側にある場合には，組合せの仕方として歯車Ｂが歯車Ｘの左側にある，歯車Ｂが歯車Ｘと歯車Ｙの間にある，歯車Ｂが歯車Ｙの右側にある，の３つの場合が考えられる。このように歯車の並びの順列を枚挙しただけでも歯車変速機構の位相的構造としては多数の候補が存在することになり，組合せの爆発が起こることがわかる。

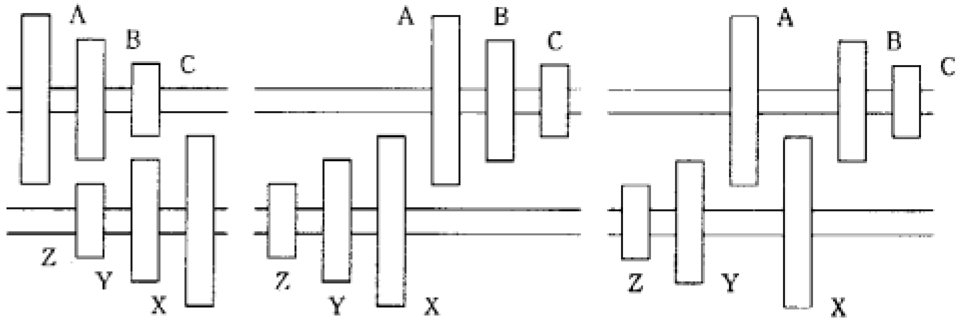


図３　２つの歯車軸の相対的な位置の組合せ

３．３　幾何学的構造

第２段階では，第１段階で枚挙され構造の候補の検査を通過した，歯車変速機構の位相的構造に基づき，歯車と歯車の間に適当なスペースを挿入して，幾何学的に具体化された幾何学的構造を枚挙する。この段階で歯車の空間的な位置関係が具体化されるので，実際にすべり歯車を移動させて，設計仕様で指定された歯車の対を噛み合わせることができるかどうかを検査する。この段階では枚挙された幾何学的構造について，指定された歯車の対が噛み合っている状態，および速度比を変更するためにすべり歯車が移動している途中で他の歯車の対が同時に噛み合ったり，接触したりしないこと（同時噛み合いの禁止）が検査される。これを図４に示す。

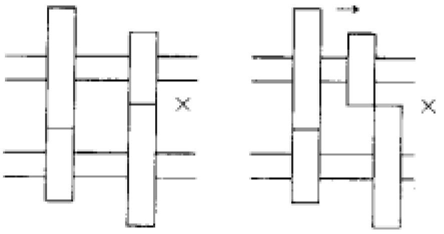


図4　同時噛み合いの禁止

3．4　生成検査法

第１段階で利用する基本的な問題解決戦略は生成検査法である。生成検査法とは解の候補を順次，枚挙していく生成部と，枚挙された解の候補が解として満足すべき条件に適合するかどうかを調べる検査部の２つのプロセスを用いて問題解決を行う戦略である。生成検査法を自然な形で並列処理する一つの方法として，検査部を並列実行する方法が考えられる。この方法では，一つ一つの解の候補の検査はそれぞれ独立に実行できるので通信のためのコストはほとんどかからないという利点があり，また検査部には同一の効率的な逐次実行のアルゴリズムが利用できるので開発も容易である。この場合，解の候補の生成部は逐次に実行することになるが，もし一つの解の候補を生成するための計算量が，一つの解の候補を検査する計算量と比べて十分に小さければ，この方法は効率的であると考えられる。

4．位相的構造の決定

4．1　問題の記述

ここでは歯車軸がＭ軸で変速段数がＮ段の歯車変速機構の位相的構造の候補を枚挙し，検査する具体的なアルゴリズムを論じる。

4．1．1　生成部

まず各歯車軸上の歯車の並びは順列を順に生成するアルゴリズムで実現できるので，Ｍ個の歯車軸についてこの順列のすべての組合せを作る。いま，ある歯車Ａの水平方向の相対的な位置をｐ（Ａ）で表すとする。一つの歯車軸上に歯車Ａ，Ｂ，Ｃ，．．．Ｚが並ぶとするとき，歯車の並びの順列を次のような順序関係で表すことにする。

ｐ（Ａ）＜ｐ（Ｂ）＜ｐ（Ｃ）＜ ．．． ＜ｐ（Ｚ）

ここで不等式Ａ＜Ｂ（Ａ＞Ｂ）はＡがＢの左側（右側）にあることを表すとする。これは同一の歯車軸上の歯車の位置の順序関係である。

次に各歯車軸の並びの順列を一つづつ選んで一つの組合せを作り，この組合せについて，隣り合う歯車軸上にあって互いに衝突する可能性がある歯車の対に基づき，Ｍ個の歯車軸の相対的な組合せの仕方をすべて求める。ここでは第Ｉ軸上の歯車Ａと衝突する可能性がある第Ｉ＋１軸上の歯車の集合

Ｏｂ（Ａ）＝｛Ｘ，Ｙ，．．．｝（１≦Ｉ≦Ｍ－１）

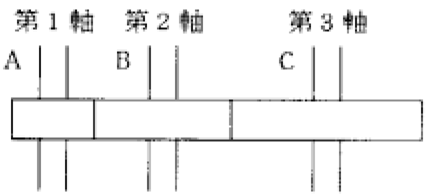
が与えられると仮定する。Ｘ，Ｙ，．．．は第Ｉ＋１軸上の歯車であり，その第Ｉ＋１軸上の相対的な位置がすでにｐ（Ｘ）＜ｐ（Ｙ）＜．．．のように与えられている。従って，第Ｉ軸上にある歯車Ａの第Ｉ＋１軸に対する相対的な組合せの仕方として，次のような順序関係を作ることができる。

ｐ（Ａ）＜ｐ（Ｘ），ｐ（Ｘ）＜ｐ（Ａ）＜ｐ（Ｙ），ｐ（Ｙ）＜ｐ（Ａ），...
ここでは異なる相対的な組合せをカンマで区切っている。これらの不等式は相異なる隣り合う軸上にある歯車の位置の順序関係を表している。

各歯車軸上の歯車の並びと，隣り合う歯車軸の相対的な組合せが指定されれば，歯車変速機構の位相的構造が一意に決まる。一つに位相的構造には同一軸上および異なる軸上にある歯車の相対的な位置に関する順序関係の一つの集合が一意に対応する。次に行うべきことは，各位相的構造で歯車の衝突により仕様で決められた歯車の対の噛み合いができなくなることがないこと，の検査である。ここでは歯車Ａと歯車Ｘが噛み合うべき歯車の対であるとき，それを次の等式で表現する。

ｐ（Ａ）　ｐ（Ｘ）

これは噛み合うべき歯車の対が噛み合うとき，その水平方向の相対的な等しくなることを表している。この等式は隣り合う相異なる歯車軸上の歯車の位置の関係を表す。この関係を使えば，図５に示すように隣り合う２本の歯車軸だけでなく，隣り合う３本の歯車軸で３個の歯車が２重に噛み合うような仕様も表すことができる。



ｐ（Ａ）＝ｐ（Ｂ）＝ｐ（Ｃ）

図５　２重噛み合い

４．１．２　検査部

検査では位相的構造の順序関係の集合に対して，噛み合うべき歯車の対を表す等式を付加して，順序関係に矛盾が生じないかどうかを調べる。順序関係の集合における矛盾とは，例えば，ｐ（Ａ）＜ｐ（Ａ）のように，ある歯車Ａが自分自身よりも右側（左側）にあるというような関係が導かれた場合である。例えば図６の例では歯車の並びの順列は｛ｐ（Ａ）＜ｐ（Ｂ），ｐ（ａ）＜ｐ（ｂ）｝，歯車Ａと歯車ｂとが衝突する可能性があり，第１軸と第２軸との相対的な組合せは｛ｐ（Ａ）＜ｐ（ｂ）｝となっている。これらの不等式の集合が順序関係の集合を構成する。図６に示すように歯車Ａと歯車ａ，歯車Ｂと歯車ｂとが噛み合うべき歯車の対として設計仕様で与えられれば，これから決まる等式は(1) ｐ（Ａ）＝ｐ（ａ），(2) ｐ（Ｂ）＝ｐ（ｂ）の２つである。検査は順序関係の集合に(1) を付加した集合，順序関係の集合に(2) を付加した集合について行い，両方とも矛盾が生じなければ，この順序関係の集合は構造の位相的構造の妥当な候補と判断される。この例では矛盾は生じない。

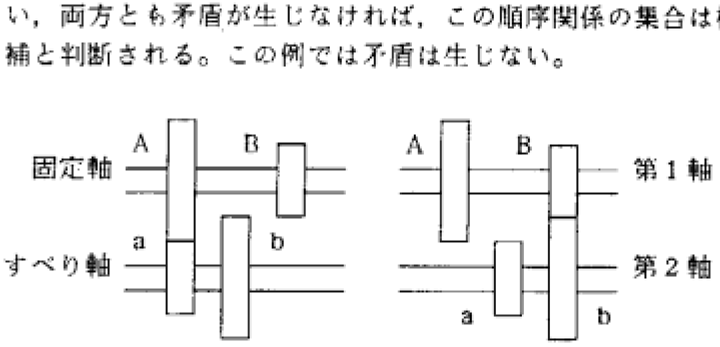


図６　噛み合うべき２組の歯車の対

ここでは，順序関係の集合に既に矛盾がある場合と，等式を加えて初めて矛盾が生じる場合とがある。いずれの場合でも順序関係の集合に矛盾があれば，その順序関係が表す位相的構造をユークリッド空間で実現することは不可能であり，構造の候補から除くことができる。この検査でもし矛盾が生じなければ，その順序関係が表す位相的構造は実現可能であると判断して第２段階に受け渡す。

順序関係の集合の矛盾の有無はグラフ理論の手法を利用して調べる。ここでは順序関係を表す不等式の不等号を有向弧とみなすことにする。不等式の両辺にある項をグラフのノードと考えると，順序関係の集合は一つの有向グラフで表すことができる。例えば，{p(A)＞p(B)，p(B)＞p(C)} という順序関係の一つの集合は {A→B，B→C} という有向弧の集合で表される。以後，不等式を有向弧に変換した表現では簡単のために p()を省いた表記を用いる。このとき順序関係の集合に矛盾が生じることは，順序関係の集合が表す有向グラフが閉路をもつこと，すなわち有向グラフに強連結成分が存在することと等価[6]である。従って，検査は有向グラフの強連結成分の有無を調べるアルゴリズムがそのまま利用できる。

４．２　処理手順

以上をまとめると次の処理手続きを得る。

（１．１）第１軸の歯車軸の歯車の並びの順列を生成するループ ┐
　　　………………………………………　　多重ループ
（１．M）第M軸の歯車軸の歯車の並びの順列を生成するループ ┘
　（２）すでに枚挙した組合せと鏡像関係にある組合せの除去 → (3)
　（3.1）　第１軸と第２軸の相対的な組合せを生成するループ ┐
　　　………………………………………　　多重ループ
　（3.M-1）第M-1軸と第M軸の相対的な組合せを生成するループ ┘
　　（４）噛み合うべき歯車の対を一つづつ枚挙するループ ┐
　　　（５）順序関係の集合を有向グラフに変換 → (6)　　並列実行
　　　（６）有向グラフの強連結成分の有無の検査 ┘
（７）検査をパスした位相的構造の候補を出力

ここで(1.1)から(1.M)はネストされた多重ループであり，ループ内部(2)～(6)では各軸の歯車の並びの順列の組合せが一つづつ枚挙される。次に(2) で鏡像関係にある候補が除かれる。(3.1)から(3.M-1)はネストされた多重ループであり，ループ内部(4)～(6)では衝突する可能性がある歯車の対に基づいて，隣り合う２本の歯車軸の相対的な組合せの仕方が一つづつ枚挙される。従って，２つの多重ループの内部となる(4)～(6)では変速機構の位相的構造の候補が一つづつ枚挙されることになり，この部分を並列実行する。また(4) では設計仕様で決められた噛み合うべき歯車の対を一つづつ選び，位相的構造が表す順序関係の集合に付け加える。次に(5) で順序関係の集合を有向グラフで表現し，(6) で順序関係の集合に矛盾があるかどうかを調べる。最後に，すべての噛み合うべき歯車の対に対して矛盾が生じなかった位相的構造の候補は，幾何学的構造を生成する次の段階に出力される。

次にステップ(6) の有向グラフの強連結成分の有無を調べるアルゴリズムを示す。この方法は，まず有向グラフの深さ優先探査を行い，有向グラフのすべてのノードに深さ優先探査番号をマーク付けする。次にグラフの有向弧の中に後退弧が含まれるかどうか

を調べる。この方法は「有向グラフGが与えられたとき，Gについて深さ優先探査を行ったときに後退弧に出会えばGは必ず閉路をもつ，逆に，ある有向グラフに閉路があれば，その有向グラフにどのような深さ優先探査を行っても必ず後退弧に出会う」という性質を利用した方法である。

【強連結成分の有無を調べるアルゴリズム】

ｎｏｄｅ（歯車）の識別番号：０から開始

```
/* 定数 */

#define NODE              /* node の総数 */
#define CYCLIC  1         /* cycle あり   tester() の返す値 */
#define ACYCLIC 0         /* cycle なし   tester() の返す値 */

/* 外部変数 */

int     mark[NODE];       /* mark[X]はnodeX の深さ優先探査番号，初期値０     */
int     tree[NODE];       /* tree[X]はnodeX の属する木の識別番号，初期値０   */
int     arc[NODE,NODE];   /* arc[X,Y] はnodeX からnodeY へのarc，存在しない */
                          /* arcは0，存在すれば1，tree_arc には2 を設定     */
int     counter = 1;      /* 深さ優先探査番号のカウンター，初期値１ */
int     t_counter = 1;    /* 木の識別番号，初期値１ */

main(){                               /* トップレベル */
int     i;
int     tester();                     /* tester()は整数値を返す関数    */

    for (i = 0; i < NODE; i++){ /* 任意のnode を一つ選択          */
        if (mark[i] == 0){      /* mark０のnode はunvisited       */
            mark[i] = counter;  /* unvisited node に番号付け      */
            counter++;          /* カウンターを１増やす            */
            tree[i] = t_counter;        /* node の属する木        */
            t_counter++;
            dfs(i);             /* 選んだnode に対して深さ優先探査   */
        }
    }
    if (tester() == CYCLIC)     /* 閉路検査の結果を出力 */
        printf("閉路があります");
    else
        printf("閉路はありません");
}

dfs(i)                                /* 深さ優先探査ルーチン */
int     i;                            /* i は深さ優先探査を適用するnode  */
{
```

```
        int     j;              /* j はi のdescendant node  */

        for (j = 0; j < MODE; j++){             /* arc があり, unvisited  */
            if (arc[i,j] /= 0 && mark[j] == 0){ /* descendant node を選択 */
                mark[j] = counter;      /* ここで選択された弧がtree_arc   */
                counter++;
                tree[j] = t_counter;
                arc[i,j] = 2;   /* tree_arc をマーク付け */
                dfs(j);         /* 深さ優先探査を再帰的に適用 */
            }
        }
}

int     tester(){               /* 閉路の検査ルーチン */
        int     i, j;

        for (i = 0; i < NODE; i++){
            for (j = 0; j < NODE; j++){         /* すべてのarc を枚挙 */
                if (arc[i,j] == 1){
                    if (tree[i] == tree[j] && mark[i] >= mark[j]){
                        return(CYCLIC);
                    }           /* 後退弧は, 弧の集合からtree_arc を除き */
                }               /* 同一の木に属するnode を結び          */
            }                   /* かつ深さ優先探査番号が大きなnode から */
        }                       /* 小さなnode に向かう弧である          */
        return(ACYCLIC);        /* 自分自身に向かうarc も後退弧である    */
}
```

アルゴリズムは手続き型言語であるC言語により記述されている。トップレベルの制御はmain関数で，深さ優先探査は関数dfs で記述される。有向グラフを深さ優先探査したとき，探査された有向弧（tree arc）全体を深さ優先の極大森という。後退弧は極大森に属さない有向弧であり，かつ極大森に含まれる頂点を子孫の頂点から先祖の頂点に向かっているような有向弧として定義される。この後退弧の有無の検査は関数testerで記述される。

例題１でこの章で述べた処理手順を実行してみる。

【例題１】歯車軸が２軸で変速段数が２段の変速機構の構造を設計すること

【例題１の解】

この例では第１軸，第２軸上にそれぞれ２個の歯車が並ぶ形式となる。第１軸上の歯車をA，Bとし，これらと噛み合う第２軸上の歯車をそれぞれａ，ｂとする。ステップ１の多重ループにより，第１軸と第２軸の歯車の並びの順列の組合せはとして次の４つの組合せが枚挙される。

ｐ（A）＜ｐ（B），ｐ（ａ）＜ｐ（ｂ）.....　（１）
ｐ（A）＜ｐ（B），ｐ（ａ）＞ｐ（ｂ）.....　（２）
ｐ（A）＞ｐ（B），ｐ（ａ）＜ｐ（ｂ）.....　（３）

p（A）＞p（B），p（a）＞p（b）.....　（4）
鏡像関係にある組合せは不等号＜，＞の向きを逆転して一致する対であり，この場合には（1）と（4），（2）と（3）が鏡像関係にある。従って，以下では（1）と（2）を考えることになる。
歯車の大きさを考えて，第1軸の歯車Bと第2軸の歯車aとが衝突の可能性のある対だとする。このとき，どちらの歯車の水平方向の相対的な位置がより左側（右側）になるかにより，次の2つの可能性がある。

p（a）＜p（B）.....　（5）
p（a）＞p（B）.....　（6）

これを（1），（2）と組み合わせて考えると，結局，次の4つの順序関係の集合で記述される位相的構造の候補が枚挙される。

p（A）＜p（B），p（a）＜p（b），p（a）＞p（B）..... 位相的構造1
p（A）＜p（B），p（a）＜p（b），p（a）＜p（B）..... 位相的構造2
p（A）＜p（B），p（a）＞p（b），p（a）＞p（B）..... 位相的構造3
p（A）＜p（B），p（a）＞p（b），p（a）＜p（B）..... 位相的構造4

これらの位相的構造の候補を次の図7に示す。

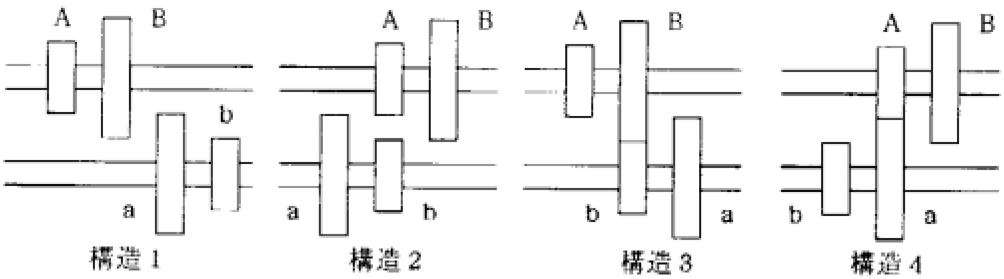


図7　位相的構造の4つの候補

また，この例では歯車Aと歯車a，歯車Bと歯車bとが噛み合うので，次の等式を得る。

p（A）＝p（a）.....　（7）
p（B）＝p（b）.....　（8）

次にステップ4以下の順序関係の集合の矛盾の有無に関する検査を行う。候補の数は4つなので，このプロセスは4個のプロセッサで並列実行される。順序関係を有向グラフに変換する方法は自明である。まず位相的構造1では

{A←B，a←b，a→B，A＝a}.....　（9）
{A←B，a←b，a→B，B＝b}.....　（10）

を得る。各順序関係の集合で等号を代入すると（9）ではa←Bおよびa→B，（10）ではa←bおよびa→bとなり，有向グラフにおいて閉路をもつので，候補1は位相的構造の候補として不適であることがわかる。どちらか一方で閉路が発見されればその時点で候補1の検査は終了される。
次に構造2では

{A←B，a←b，a←B，A＝a}.....　（11）
{A←B，a←b，a←B，B＝b}.....　（12）

となり，等号を代入してもいずれの集合でも有向グラフは閉路はもたない。
構造3では

{A←B，a→b，a→B，A＝a}.....　（13）

でａ→ｂおよびａ←ｂの閉路が存在するので，ともに位相的構造の候補としては不適である。結局，構造２だけが歯車変速機構の位相的構造の候補として残される。

――――――――（例題１終わり）――――――――

５．計算量の考察

次に設計仕様が与えられたときに枚挙される歯車軸上の歯車の並びの候補の数を調べてみる。一般的な２軸ｎ段変速の変速機構について考える。この場合は第１軸，第２軸上にそれぞれｎ個の歯車が並び，ｎ個の変速状態を実現するｎ組の歯車の対がが最初から設計仕様として与えられる。速度比が大きいものから小さいもの順に第１速から第ｎ速までをとり，第１軸を入力軸，第２軸を出力軸とする。このとき第１軸の歯車はその径が第１速から第ｎ速の順に大きくなり，第２軸の歯車はその径が第１速から第ｎ速の順に小さくなるので，衝突する可能性がある第１軸と第２軸の歯車の対は歯車径の数値とは無関係に決まってしまう。例えば，第１軸上の第ｉ速を実現する歯車は第２軸上の第ｊ速（１≦ｊ≦ｉ－１，ｉ≧２）を実現する歯車と衝突する可能性があることがわかる。図８に２軸３段変速歯車減速機構の例を示す。

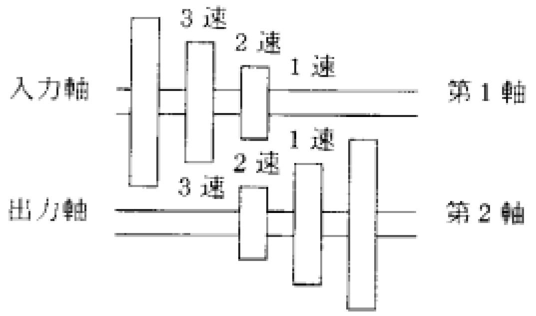


図８　２軸３段変速歯車減速機構

まず第１軸上の歯車の並びは順列を求めてｎ！通りの候補がある。第２軸も同様である。ここでは構造の左右を逆にして互いに移りあうような鏡像関係にある２つの位相的構造は同一であるとみなすので，ここまでで候補の数はｎ！ｎ！／２通りとなる。つぎに第１軸と第２軸との組み合わせ方を考慮する必要がある。第１軸の第ｉ速歯車は第２軸上にあるｉ－１個の歯車と衝突する可能性があるので，第１軸の第ｉ速歯車の第２軸に対する相対的な配置場所としてはｉ－１個の歯車の両端と各歯車の間の計ｉ個の場所が考えうる。これを第１軸上のすべての歯車（１≦ｉ≦ｎ）について考えると，全部でｎ！通りの相対的な位置の組み合わせが可能である。従って，２軸ｎ段変速の歯車変速機構の位相的構造の候補としてｎ！ｎ！ｎ！／２通りが可能であることがわかる。これに対して例えば，ｎ＝３，４，５に対する値は１０８，６９１２，８６４０００となる。このすべての候補の一つ一つについて有向グラフの閉路の存在の有無を検証する必要がある。ここでは組合せの爆発が生じており，大きなｎについては手に負えない問題となるのでなんらかの対策が必要となる。

そこでｎ段変速の場合の歯車の並びの順列を考えてみると，実はｎ－１段変速の場合の歯車の並びの順列から構成できることがわかる。ｎ段変速の場合の歯車の並びの順列を作るにはｎ－１段変速の歯車の並びの順列で，両端あるいは歯車と歯車の間の計ｉ箇所のどれか一つに新たに歯車を挿入すればよい。従って，ｎ－１段変速で有向グラフの閉路があるとわかった位相的構造はそこで棄却し，それからｎ段変速の場合の位相的構

造を構成しないようにすれば，枚挙する候補の数をかなり減らすことができる。例題１で，２軸２段変速の歯車変速機構の位相的構造の候補のうち，解として適切な候補は鏡像関係を考えると２つある。インクリメンタルに構造を作る場合は鏡像関係にある２つの位相的構造を区別して考える必要がある。２軸２段変速の変速機構の構造から２軸３段変速の歯車変速機構の位相的構造を構成すると，第１軸上の新しい歯車の置き方が３通り，第２軸上の置き方も３通り，新たに構成した２つの軸の相対的な位置の組合せの仕方として３通り，計２×３×３×３＝５４通りの構造が枚挙される。閉路の検査については，これらの候補のうち鏡像関係にある２つの位相的構造は同一とみなして実行すればよい。図９に２段変速機構から構成した３通りの例を示す。

鏡像関係にある構造を区別すると，一般に，ｎ－１段変速機構の位相的構造の候補で閉路の検査を通過した解の候補の個数をｓ（ｎ－１）とすると，これから枚挙されるｎ段変速機構の位相的構造の候補の数はn^3×ｓ（ｎ－１）となり，これから鏡像関係にある候補を除いたものについて閉路の検査を実行すればよいことになる。

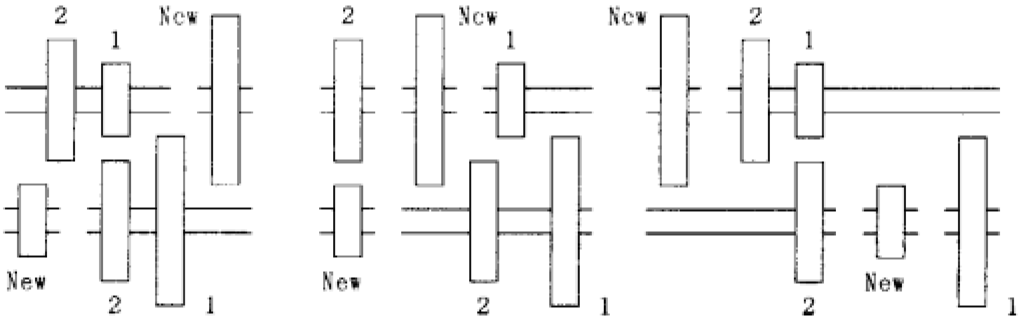


図９　２段変速機構から３段変速機構へ

ここでｎ－１段変速機構からｎ段変速機構を構成する際に，第１軸に新たに付加する歯車の径がすでに第１軸にあった歯車の径より大きくなるように構成しても一般性は失わない。すると第１軸に追加される新しい歯車は第２軸上にすでにあるすべての歯車と衝突する可能性があり，第２軸に追加される新しい歯車は第１軸上にすでにある歯車とは衝突しないことになる。この事実に着目すると，枚挙すべき候補の数をさらに減らすことができる。ｎ－１段変速機構の構造からｎ段変速機構の構造の候補を構成する際に，第１軸の上の新しい歯車の置き方にはｎ通りあるが，ここで第２軸上の新しい歯車の置き方を，第１軸に追加した歯車と第２軸上にすでにある歯車との相対的な位置関係に合わせて決めることを考える。例えば，第１軸に新たに追加する歯車の第２軸に対する相対的な位置が第２軸の歯車Ａの右側であれば，第２軸に新たに追加する歯車は歯車Ａの右側に置くことにする。あるいは第１軸に追加する歯車の第２軸に対する相対的な位置が第２軸上の歯車Ａと歯車Ｂの間であれば，第２軸に追加する歯車も歯車Ａと歯車Ｂの間に置く，というようにする。

図１０に歯車軸上に新しい歯車を付加する場合の３つの例を示す。この方法で２軸２段の機構の位相的構造から２軸３段の機構の位相的構造の候補を構成すると，枚挙される候補の数は２×３×３＝１８個まで減る。ｎ－１段変速機構の構造として閉路の有無の検査を通過した解の候補の個数をｓ（ｎ－１）とすると，この方法でこれから枚挙されるｎ段変速機構の構造の候補の数はn^2×ｓ（ｎ－１）となる。この方法は生成部に検査部の機能の一部を組み込むことにより，枚挙する候補の数を減らして生成検査法の効率化を図るという一つの典型的な例である。

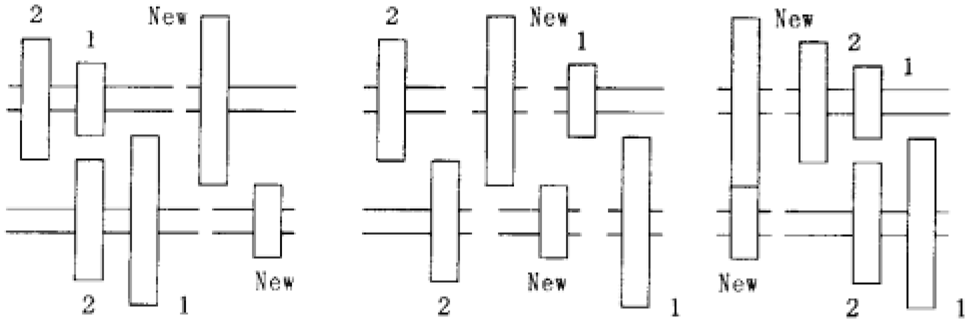


図１０　新しい歯車の挿入場所

６．幾何学的構造の決定

６．１　問題の記述

ここでは与えられた位相的構造に基づき，歯車減速機構の幾何学的構造の候補を枚挙し，検査する具体的なアルゴリズムを調べる。

生成部では位相的構造で位置の順序関係だけが決められた歯車軸上の歯車と歯車の間に適当なスペースを挿入した幾何学的構造を枚挙することが要求される。検査部では枚挙された幾何学的構造について，それぞれすべり軸に指定された歯車軸を移動させるという操作を行い，設計仕様から噛み合いが指定された歯車の対を噛み合わせ，同時噛み合い禁止の条件のもとですべての噛み合いが実現できるかどうかを検査することが要求される。最後に，検査を通過した候補のうちで減速機構の全体幅を最小にするという条件を満たす幾何学的構造の候補が解として出力される。

最適性の条件を考えるためには，歯車減速機構の幾何学的構造の候補すべてを枚挙することが必要である。そこで位相的構造の場合と同様に，枚挙すべき候補の数を考察してみる。全体幅を最小にするという目標があるので，位相的構造に挿入すべきスペースの総数は最初は１から始めることにし，最適解が得られるまでスペースの総数を一つづつ増加させて幾何学的構造を順次，枚挙することにする。図１１は２軸３段変速の歯車減速機構のある位相的構造

{p(C)＜p(B)＜p(A), p(c)＜p(b)＜p(a), p(C)＜p(b), p(B)＜p(a)}

が与えられたときに，各歯車間に挿入すべきスペースの総数が２である場合に枚挙された４通りの幾何学的構造の例である。

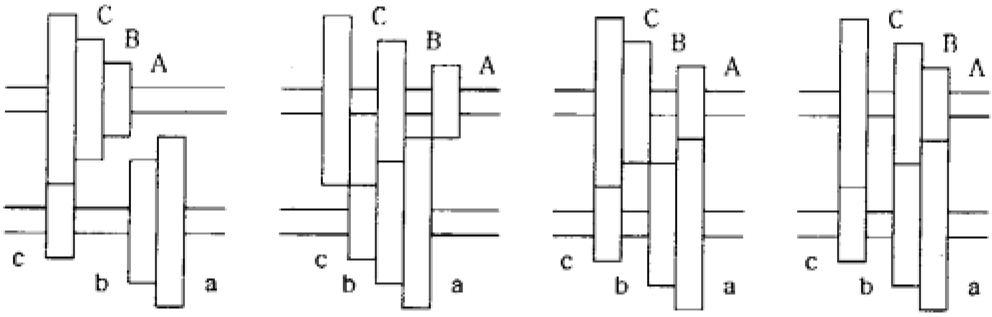


図１１　幾何学的構造

６．２　計算量の考察

一般的な２軸ｎ段変速の歯車減速機構について考える。ある一つの位相的構造が与えられたとして，各歯車間に挿入すべきスペースの総数がｍとする。第１軸，第２軸それぞれの上にｎ個の歯車があるので，スペースが挿入できる場所は２（ｎ－１）個ある。各スペースで２（ｎ－１）通りの挿入の仕方があるので，枚挙される幾何学的構造の数は全部で$\{2^m(n-1)\}$あることになる。設計仕様が複雑になり挿入すべきスペースの総数が増加すると，枚挙される候補の数は指数関数的に増加するので，ここでも位相的構造の場合と同様，組合せ爆発が生じることがわかる。ｍ＝４，ｎ＝４のとき１３５６通り，ｍ＝５，ｎ＝５のとき３２７６２通りとなる。

そこで，ここではすべての幾何学的構造の候補を枚挙することはやめて，真の最適な幾何学的構造を求めることをあきらめる代りに，あるヒューリスティック探査を用いて，同時噛み合いの禁止や，歯車の衝突を回避する条件を満たすために新たにスペースを挿入すべき場所を限定することにより，枚挙する候補の数を減らすことを考える。うまいヒューリスティクスが利用できるならば最適に近い幾何学的構造の少数の候補を枚挙して調べることにより，許容できる時間の範囲で望みの結果を得ることができる。現実の機械設計を調べると，すべり軸よりも固定軸である歯車軸上の歯車の間にスペースをより多く取ることが多いことがわかる。これは動く歯車軸の幅をより小さくし，固定された歯車軸の幅をより大きくすることであり，一般的にみて機構の全体幅をより小さくすることに貢献すると考えてよい。そこで次のヒューリスティクスを利用する。

- 固定軸に指定された歯車軸上の歯車の間に優先的にスペースを挿入する。それだけで同時噛み合いの禁止や歯車が衝突しない条件が満足されないような場合に限り，すべり軸に指定された歯車軸上の歯車の間にスペースを挿入する。

位相的構造で指定された歯車軸上の歯車の並びの間に適当なスペースを生成する手続きは，一つのバックトラックの機構を用いて実現する。その一般的な枠組みを述べる前に，例題２で具体的な処理の手順を説明する。

【例題２】次に示す位相的構造が与えられた２軸３段変速の歯車減速機構の幾何学的構造を求めることを考える。設計仕様としては，図１２に示すように第１軸と第２軸の歯車の対，Ａとａ，Ｂとｂ，Ｃとｃが噛み合うものとし，それぞれ第１速から第３速までを実現するようにする。

$\{p(C)<p(B)<p(A),\ p(c)<p(b)<p(a),\ p(C)<p(b),\ p(B)<p(a)\}$ .....　（１３）

この例題の構造は２軸なので衝突する可能性がある歯車の対は与えなくてもわかる。また第１軸→すべり軸／第２軸→固定軸，第１軸→固定軸／第２軸→すべり軸の２つの場合を考える必要がある。

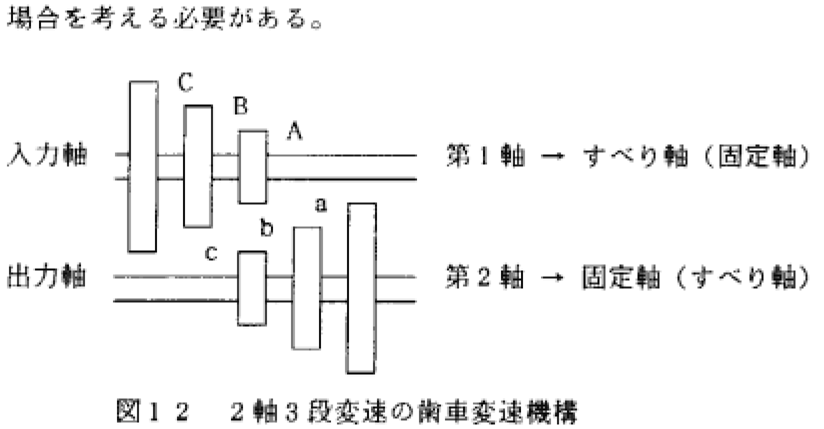


図１２　２軸３段変速の歯車変速機構

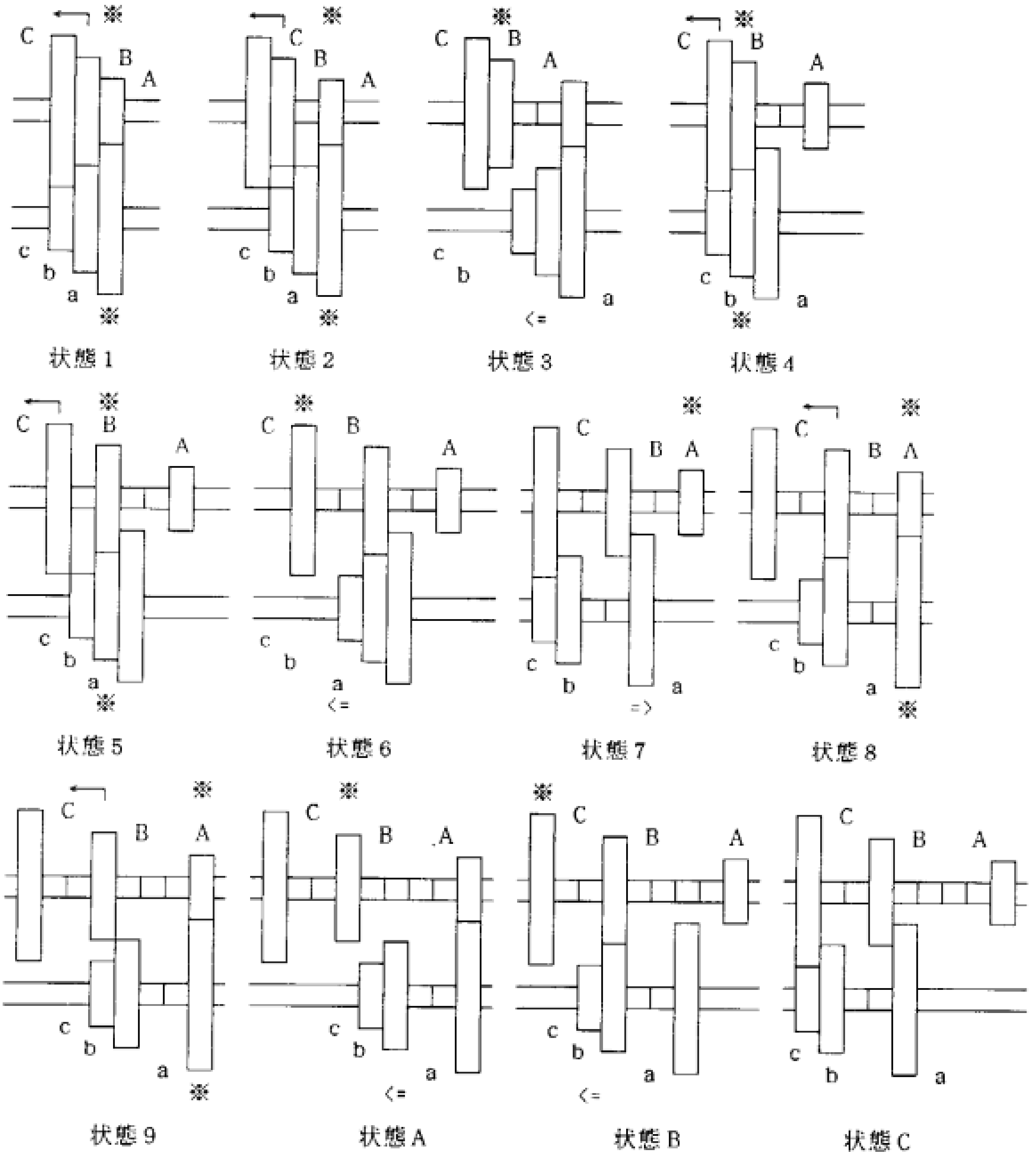


図１３　例題の途中結果

【例題２の解】

まず第１軸が固定軸，第２軸がすべり軸の場合を調べる。

（状態１→状態２）

第１速を実現するために歯車Ａと歯車ａを噛み合わせた状態１から解の探査を始める。この状態ではすべての歯車は密着して置かれている。状態１では歯車Ｂと歯車ｂ，歯車Ｃと歯車ｃの２ヶ所で同時噛み合いが生じているので，固定軸である第１軸上の歯車の並びの間にスペースを生成することによりこれらの同時噛み合いを解消する。スペースを生成する場所の候補として，歯車Ａと歯車Ｂの間，歯車Ｂと歯車Ｃの間の２ヶ所あるが，まず歯車Ａと歯車Ｂの間を選んでスペースを１個生成する。

なお，以下の処理手順では，歯車軸上の歯車は算盤玉にたとえることができる。記号

※を付けた歯車はその絶対的な位置が空間的に固定されているとみなし，それ以外の歯車は軸上を左右に自由に動きうるものとする。また一度生成されたスペースは取り除くことができず，以降はあたかも透明の歯車が存在するかのように扱うことにする。また図１３で，例えば状態４で歯車ａと歯車Ｂ，歯車ｂと歯車Ｃの側面が接触したり，状態９で歯車Ｂと歯車ｂの角の部分が接触しているように見えるが，このような箇所にはすでに適当なクリアランスがとられていると考えて議論する。

歯車Ａと歯車Ｂの間にスペースを生成する操作は次のように行う。まず，噛み合うべき歯車Ａと歯車ａの対を歯車軸上に固定し，それ以外の歯車はすべて歯車軸上を自由に動きうるようにする。歯車Ｂは固定された歯車Ａの左側にあるので，スペースを生成するには歯車Ｂを歯車Ａとは反対の左方向に１つだけ移動することが必要である。このとき歯Ｂに左側に隣接している車歯車Ｃは自由に動けるので，歯車Ｂとともに左方向に１だけ移動する。また，このスペース生成操作では第１軸と第２軸の歯車の衝突を考慮する必要があるが，この場合には衝突は発生しないので第２軸上の歯車の位置には変化は生じない。スペース生成操作の結果，状態２が得られる。スペース生成操作は記号┌→，←┐などで表す。

（状態２→状態３）

状態２では歯車Ｂと歯車ｂ，歯車Ｃと歯車ｃの対の同時噛み合いは解消されている。しかし，歯車Ｂと歯車ｂ，歯車Ｃと歯車ｃ対は，その位置が１だけずれているだけなので，状態２ですべり軸である第２軸をすべらせた場合には，まだ歯車Ａと歯車ａの対が噛み合いが完全に離脱しない間に歯車Ｂと歯車ｂ，歯車Ｃと歯車ｃの対で噛み合いが生じてしまうことは明らかである。従って，歯車Ａと歯車Ｂの間に生成すべきスペースの個数は１個ではなく２個であったことがわかる。歯車Ａと歯車Ｂの間にスペースをさらに１個生成したもの状態３である。状態３は状態１でスペースを２個生成して得られたと考えてもよい。

（状態３→状態４）

状態３では第１速を実現する歯車Ａと歯車ａの噛み合いには支障がないので，次に第２速を実現する歯車Ｂと歯車ｂの噛み合いを調べる。そのためにすべり軸である第２軸を実際に動かして，歯車ｂを歯車Ｂと噛み合える位置に移動するすべり操作が必要になる。すべり操作では噛み合いを作る歯車で固定軸上にある歯車Ｂを固定し，それ以外の歯車はすべて自由に動けるようにする。そして，すべり軸を実際にすべらせることにより噛み合いを作る歯車ですべり軸上にある歯車ｂを歯車Ｂと同じ位置まで移動させる。歯車ｂは歯車Ｂから右に２つ離れているので，第２軸を左方向に２だけすべらせる操作が必要である。

ここで注意しておくが，すべり軸上にある歯車が自由であるというのは，すべり軸上の歯車はすべり操作においてはすべり軸とともに移動するが，すべり操作の途中で固定軸やすべり軸の上にある固定された歯車と接触する場合にはその移動が妨げられるということを意味すると約束する。

この際に歯車ｂを所定の位置まで移動できることは，歯車の並びの順列を考えた位相的構造を生成した段階ですでに保証されている。状態３で第２軸をすべらせた結果が状態４である。この場合も第１軸と第２軸の歯車の間には衝突はなく，第２軸には変更はない。すべり操作をスペース操作と区別するために記号＜＝，＝＞で表す。

（状態４→状態５→状態６）

状態４では歯車Ｃと歯車ｃの対で同時噛み合いが発生している。この同時噛み合いは固定軸である第１軸上の歯車Ｂと歯車Ｃの間にスペースを生成して解消する。状態１から状態３を得たのと同様，噛み合いを作る歯車Ｂと歯車ｂの対を固定し，他の歯車は自由にする。固定された歯車Ｂが歯車Ｃの右側にあるので，歯車Ｃを左方向に２だけ移動

して歯車Bと歯車Cの間にスペースを2個生成する。このスペース生成操作の結果が状態6である。このスペース生成操作では第1軸と第2軸とで歯車の接触はなく，第2軸には変更はない。

（状態6→状態7）

状態6で第2速の噛み合いが達成されたので，次に第3速を実現する歯車Cと歯車cの噛み合いを調べる。ここでも状態3から状態4を得たのと同様，固定軸上の歯車Cを固定し，他のすべての歯車を自由にする。そして，すべり軸である第2軸をすべらせることにより歯車cを歯車Cと噛み合える位置に移動させる。歯車cは歯車Cからスペース2個分右方向に離れているので，第2軸を左方向に2だけ移動するすべり操作が必要である。

このすべり操作で歯車bと歯車cはすべり軸とともに移動できるが，歯車aは固定軸上の歯車Bと接触する。歯車aはすべり軸とともに左方向に移動しようとして歯車Bを左方向に押しつける。しかし固定軸上の歯車Cは固定されており，歯車Bと歯車Cの間にはスペース2個が存在するので歯車Bの左方向の移動は束縛されている。従って，歯車aはすべり操作から取り残されて歯車Bの右側の位置にそのまま留まることになる。この状態が状態7である。

（状態7→状態8）

状態7では第3速の噛み合いが実現されている。もし状態7の幾何学的構造にスペースを追加せず変更を加えずに，第1速，第2速の噛み合いが実現できれば，この幾何学的構造は設計仕様を満足する構造となるので問題の解として出力される。そこで第1速の噛み合いを実現するために歯車Aを固定して第2軸をすべらせて歯車aを歯車Aと同じ位置に移動させる。歯車aは歯車Aの左側に2だけ離れているので，第2軸を2だけ右方向にすべらせる。そのすべり操作の結果，状態8が得られる。

（状態8→状態9→状態A）

状態8では歯車Bと歯車bの対で同時噛み合いが生じている。そこで前と同様，歯車Aと歯車aを固定して，歯車Aと歯車Bの間にスペースを2個生成した結果が状態Aである。

（状態A→状態B→状態C）

状態Aでは歯車Aと歯車Bの間にスペース4個，歯車Bと歯車Cの間にスペース2個，歯車aと歯車bの間にスペース2個が生成された幾何学的構造を表す。状態Aでは第1速の噛み合いが実現されている。状態Aから第2速の噛み合いを実現したものが状態B，状態Bから第3速の噛み合いを実現したものが状態Cである。状態A，B，Cより一つの幾何学的構造のもとですべての速度比が得られるので，この幾何学的構造が問題の解である。

第1軸をすべり軸，第2軸を固定軸とした場合は，この解の第1軸と第2軸とを入れ替えた幾何学的構造が得られ全体幅は等しくなる。これは例題2が2軸の変速機構を扱ったからである。この幾何学的構造では，すべり軸である第2軸の歯車はすべり操作の途中では常に固定軸である第1軸の両端の歯車の間に位置するので，全体幅は固定軸の幅と同じ9となる。厳密にいえば，歯車間のクリアランスを考慮して機構の全体幅は9より大きくしなければならない。ここで得られた幾何学的構造は，問題で与えられた位相的構造に基づいて設計した場合に，実際の機械設計で知られている全体幅最小の歯車変速機構と一致する。

――――――（例題2終わり）――――――

6．3 処理手順

この例は偶然にも真の最適解を見つけることができて処理が停止した例であるが，一般的にいえば，最適解を得るにはすべての場合を尽くす必要があり，ヒューリスティクスに頼る方法では解の最適性は保証されない。また処理手順が停止するという保証もないので，場合によっては無限大の幅の幾何学的構造を次々に作り続けることもありうる。このことに注意して，例題に示した具体的な処理手順を考慮すると，問題解決の一般的な枠組みとして図１４に示すものを考えることができる。

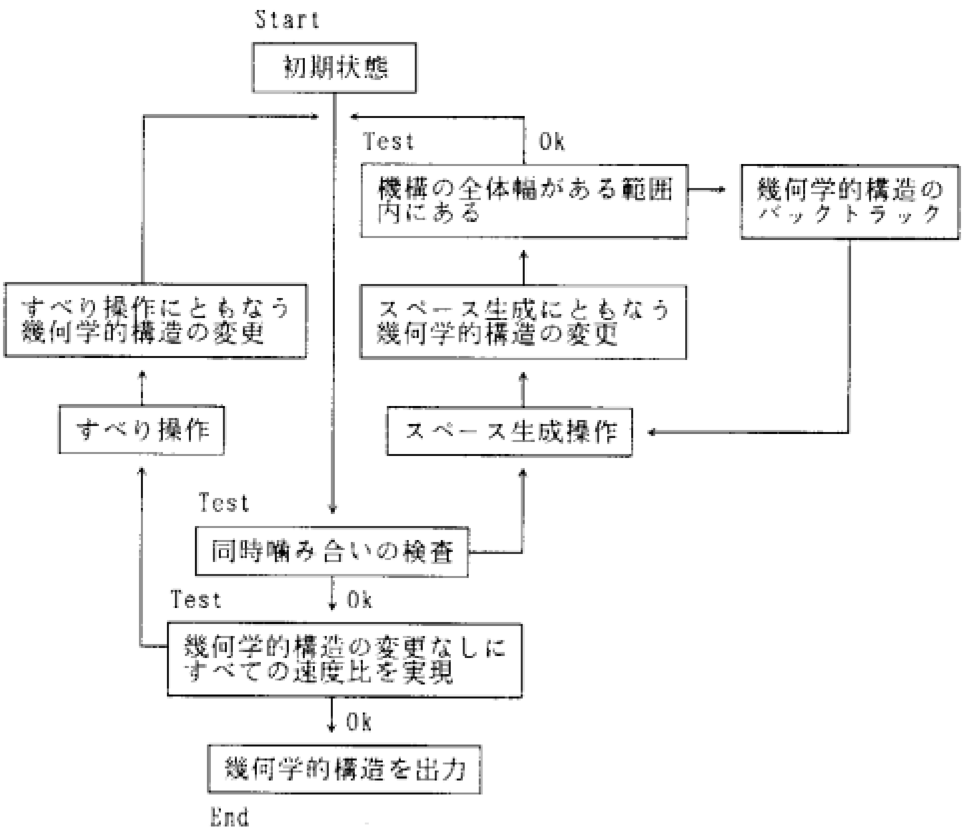


図１４　幾何学的構造の決定の処理手順

６．３．１　処理手順に関する考え方

この枠組みの処理の流れに２つのループがある。一つは幾何学的構造の候補について同時噛み合いが生じているかどうかを検査し，同時噛み合いがある場合に適切な箇所にスペースを生成して幾何学的構造を修正して同時噛み合いを解消する内側のループ，もう一つは幾何学的構造の候補において実際に設計仕様で決められた噛み合いを実現できることを検査し，必要に応じて幾何学的構造に修正を加える外側のループである。また処理手続きが停止せずに無限に構造を作り続けるような場合を考慮して，歯車変速機構の全体幅についてある上限を設けることにする。幾何学的構造の候補がこの制限を越えたときには，これ以上のスペース生成による変更は停止して，構造を以前に得られたある時点の構造までバックトラックさせて，別の場所に新たにスペースを生成するような機構を設けることにする。幾何学的構造の候補はこれらの検査，修正が行なわれ，ある幾何学的構造ですべての速度比が実現できる構造が得られれば，それが問題に対する解として最後に出力される。

並列処理環境では図１４に示した処理手順はすべて一つのプロセッサで実行されることになる。位相的構造を決める部分で並列処理を行っているので，それに続いて幾何学

的構造を決める処理も同じプロセッサで処理されることになる。次にこの枠組みの３つの基本部分について述べる。

6．3．2　スペース生成操作

スペース生成が必要になるのは同時噛み合いが生じる場合である。この操作はすべり操作とは異なり，幾何学的構造に修正を加えるための操作である。同時噛み合いは，ある速度比を実現するために噛み合うべき歯車の対が噛み合っているときに，(a)別の歯車対の水平方向の相対位置が等しい場合および，(b)別の歯車対の水平方向の相対位置が１しか違っていない場合に発生する。同時噛み合いを解消するためには(a)の場合にはスペースを２個生成，(b)の場合にはスペースを１個生成して，別の歯車対の水平方向の相対位置が２以上ずれるように幾何学的構造を修正する必要がある。

スペース生成操作では，まず固定軸およびすべり軸上にある噛み合うべき歯車の対を空間的に固定し，それ以外の歯車はすべて歯車軸上を左右方向に自由に動き得るようにする。ここでは機構の全体幅をより小さくするために，すべり軸よりも固定軸のほうにより多くのスペースを生成する，というヒューリスティクスを利用する。従って，固定軸上にあって同時噛み合いを起している歯車を移動することによりスペースを生成することになる。移動の対象となる歯車をどちらの方向に移動させるかは，その歯車が，同じ軸上にあって固定された歯車の左右どちら側にあるかに依存する。移動すべき歯車が固定された歯車の左側（右側）にあれば，その歯車は左方向（右方向）に移動されることになる。

スペース生成操作では常に異なる軸上にある歯車の間での接触の検出を行う必要がある。衝突する可能性がある歯車の対は既に与えられているので，対をなす歯車の水平方向の相対的な位置の差が１になったかどうかで衝突の有無が判断できる。

この枠組みでは，同時噛み合いが発生したとき，その都度ごとに必要最小限のスペースを生成するという戦略をとっている。この戦略はすべての幾何学的構造の候補を枚挙して真の最適解を求めようとした場合に組合せ爆発が生じて問題が手に負えなくなる，という事情を考慮して採用した方法である。この処理手続きが停止するならば，この戦略では一つの位相的構造に対して一つの幾何学的構造が決まることになるので，調べるべき解の候補の数はかなり減らすことができる。しかしこの戦略では最適解の探査を局所的に行うことになるので真の最適解が得られる保証がないこと，さらに困難な問題として処理手続き自体が停止するかどうかの保証がないことがある。前者の問題について幾何学的構造にスペースを生成するための適切なヒューリスティクスを利用することで対処できる。後者の問題については，幾何学的構造の全体幅についてある許容範囲を設けて，この制限を越えるような候補にこれ以上スペースを生成しても最適解は得られないとみなして解の探査を打ち切るという対処が考えられる。この機構は幾何学的構造のバックトラック機構のところで述べる。

6．3．3　すべり操作

すべり操作が必要になるのは，幾何学的構造の途中解である速度比を実現する歯車対が実際に噛みあわせられるかどうかを調べる場合である。すべり操作はスペース生成操作とは異なり，実際にすべり軸をすべらせる操作を表すが，その結果として幾何学的構造にスペースが生成されることもある。

すべり操作では噛み合うべき歯車対をなす歯車で固定軸上にある歯車を空間的に固定する。それ以外の歯車で固定軸上にある歯車は自由に動けるようにし，すべり軸上の歯

車はすべり軸が移動する場合にはすべり軸とともに移動できるようにする。ただしすべり軸上の歯車は，外部の障害物（固定軸上の歯車）と接触，衝突する状況になり，かつその障害物がすべり軸の移動を妨げる状況にある場合，その歯車は障害物と接触したまますべり運動から取り残されるようにしておく。

スペース生成操作，すべり操作のいずれも固定軸上の歯車の間に優先的にスペースを生成することを意図しているので，すべり軸上の歯車の間にスペースが生成されるのは，スペース生成操作やすべり操作において歯車どうしの衝突が発生する場合に限られている。

6．3．4　幾何学的構造のバックトラック機構

この機構で最も重要なことは，ある幾何学的構造で処理手続きが停止せず解の探査に失敗した場合に，どの途中解までバックトラックして解の探査を再開するかという点である。このような問題に対する一般的な有効な対策を見つけることは困難であり，ここでは問題特有の性質に着目したヒューリスティクスに頼る他に道はなさそうである。そこで，提案する枠組みでは必要に応じてスペースを順次生成するという局所的な解の探査を行っていること，バックトラックが起動されたときは無限ループが生じて処理手続きが停止しなくなっている可能性が大きいことを考慮すると，できるだけ深いバックトラックを行うことが有効であるという一つのヒューリスティクスが得られる。

スペース生成は固定軸上の歯車の間に優先的に行われるが，場合によってはこれが原因で処理手続きが止まらなくなることもありうる。そのためにはバックトラックが起動されたとき固定軸だけでなくすべり軸上の適当な場所にもスペースを生成する機構が必要となる。しかし，どのような状況のときにすべり軸上な歯車の間にスペースを生成すればよいかという点については，経験に従うという以外に決め手はない。

7．　衝突する可能性がある歯車の対が不明な場合

本報告では問題設定において，相異なる隣り合う軸上にあって互いに衝突する可能性がある歯車の対がすべてまえもって与えられると仮定した。しかし，そのためにはすべての歯車の径の大小関係が，歯車変速機構の構造を設計する前の段階で既知であるか，あるいはある程度の予測がついていることが必要である。従って，現実には衝突する可能性がある歯車の対が不明な場合も存在する。この場合には本文で述べた方法はそのままでは利用できない。ただし２軸の歯車変速機構では各歯車に対する速度比の割り当てを決めた段階で歯車の径の大小関係を陽に与えなくとも自動的に衝突する歯車の対がわかってしまうので，ここでは３軸以上の歯車変速機構を考えることにする。

ここで述べた制限のもとで物理的に正しい変速機構の構造を設計する方法としては，多軸の変速機構の構造を２軸の変速機構の構造の組み合わせとして構築する手法がある。いま，変速段数がＭ段のある変速機構の構造を設計する問題が与えられたとする。ここで次のようにＭを素因数分解する。

$$M = 2^{M1} \times 3^{M2} \times 5^{M3} \times \ldots$$

$$N = M1 + M2 + M3 + \ldots.$$

ＭがＮ個の素因数に分解されたとすると，Ｍ段の変速機構はＮ個の２軸変速機構の組合せで実現できる。各素因数に対応して，２軸２段変速機構，２軸３段変速機構，２軸５段変速機構，．．．などの構造が用意できれば，あとはこれらを順次，連結するだけで任意の段数の歯車変速機構の構造が構成できる。また固定軸／すべり軸は交互に並ぶように決めてやればよい。通常の設計では２段変速，３段変速の機構だけで構成できるよう

な構造を設計することが多いようである。現実の機械設計では機構の全体幅が最小となる２軸２段変速機構，２軸３段変速機構が既に知られている。これらを図１５に示す。

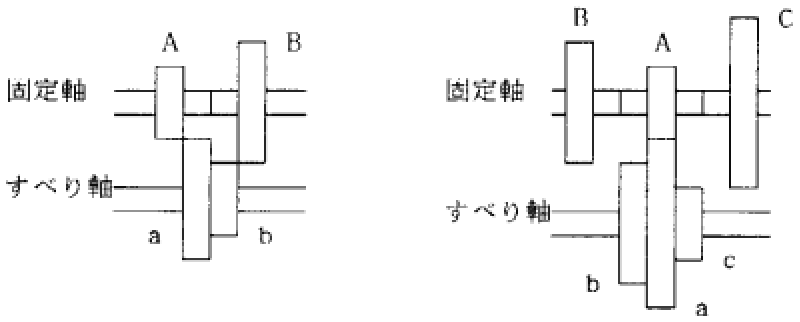


図１５　　２軸２段変速機構と２軸３段変速機構

結局，構造を設計する段階で歯車径の大小関係が未知であるという制限を受ける場合であっても，変速段数を適当に素因数に分解して，図１５に示すような２軸の変速機構を機械的につないでいくだけで構造を決定できる。ただし，この方法では機構の全体幅を最小にすることは考慮されていないので，最後に得られた多軸の変速機構の構造を修正して機構全体の幅を小さくするという手続きが必要となる。修正するためには歯車の並び替え，歯車間のスペースの圧縮，一つの歯車に複数の機能を共有させて使用する歯車数を減らす，など複雑な処理が必要となることが予想される。これらを実行することは極めて困難な問題であり，現実的ではない。

８．おわりに

本報告では機械設計向け制約指向知識コンパイラであるＭＥＣＨＡＮＩＣＯＴの機能の拡張を目的として，工作機械主軸用歯車減速機構の構造を設計する問題を対象として，問題の設定，問題の表現方法，そして問題解決に対する一つのアプローチを論じた。

歯車減速機構の構造設計問題は普通の機械設計問題とは異なり，いわゆる組合せ問題の範疇に入る問題として扱った。組合せ問題を扱う場合に避けてとおれないのは，組合せ爆発の問題である。ここではこの問題に対する解決として，構造設計問題を２つの部分問題に分割して前半は生成検査法を用い，後半はヒューリスティクスに基づき解を探索する，という問題解決戦略を採用した。これらの戦略で組合せ爆発をある程度は避けることができるものの，複雑な設計仕様が与えられた場合には多少の並列処理を導入したとしても焼け石に水で手に負えない問題になってしまうこと，解の探索を効率化した結果，真の最適解を見つけることは断念せざるを得なくなったこと，などの課題が残された。以上の問題解決に対する検討において，現実の設計問題の難しさがあらためて認識されるという結果になった。

ここで述べた枠組みに対する今後の研究課題は，例題として扱う構造設計問題に固有の性質に着目することにより，よりよいヒューリスティクスの発見，より効果的な並列処理の応用技術の開発，などを通じて組合せ問題を手に負える問題に徐々に近付けていくことであるといえる。

謝辞

本研究を行うにあたり，検討会議を通じて有意義な議論に参加し貴重な助言をいただ

いた機械設計グループの寺崎智研究員，滝寛和　主任研究員，坂根清和研究員，その他の第５研究室の研究員の皆様に厚く御礼申し上げます。また本研究の機会を与えて下さり常日頃，御指導いただいたＩＣＯＴの渕一博所長ならびに第５研究室の生駒憲治室長に深く感謝致します。

【参考文献】

［１］寺崎，永井，横山，　井上，堀内，滝：機械設計支援システム構築ツール
－ MECHANICOT－，人工知能学会，知識ベースシステム研究会資料，
SIG-KBS-8803,(1988).

［２］井上，永井，藤井，今村，小島：工作機械の設計手法の解析－旋盤の回転機能部品の設計,ICOT Technical Memorandum,TM-494, ICOT, (1988)

［３］永井，寺崎：設計向け制約指向知識コンパイラにおける制約解析および手順生成について，第３回人工知能学会全国大会，(1989) pp.693-696

［４］岡本：増補版 工作機械の構成 第４章（内田老鶴圃）(1977)

［５］須藤：機械設計(9) 歯車減速機の設計製図（パワー社）(1968)

［６］Ａ.Ｖ.エイホ，Ｊ.Ｅ.ホップクロフト，Ｊ.Ｄ.ウルマン（大野訳）：データ構造とアルゴリズム 第６章（培風館）(1987)