知識処理に基づいた2次元メッシュの自動生成システムの開発

高田 修[†] 中西広吉[†]
堀之内成明[†] 永岡真[†]

計算力学の分野では，メッシュの質の良否が結果の精度や計算時間に大きく影響を及ぼすため，質の高いメッシュを生成することが重要な課題である．メッシュ生成には分野固有の経験や知識を必要とするため，知識処理によるアプローチが有効である．我々は，流体解析や塑性変形解析などの計算力学の分野に適用できる構造格子による汎用的な2次元メッシュの自動生成システムを開発した．本論文では，専門家のメッシュ生成過程のモデル化，その実現手法である推論方式，探索手法について述べる．また，システムの評価のために，「自動車エンジンの吸気流解析」と「鍛造変形解析」の問題に適用し，モデル化の妥当性および推論方式，探索手法の有用性を確認した．システムの特徴は以下のとおりである．(1) 大域的な依存関係と局所的な依存関係を組み合わせた探索手法により，同時に複数の解を効率よく求めることができる，(2) 分野に依存しない標準的な手順（メタ・ルール）と分野固有の知識（ルール）を分離することにより，異なる分野への適用が容易である，(3) 複数の競合解消戦略を組み合わせた推論方式を用いることにより，複雑な処理の流れを的確に表現できる．

A Knowledge-Based System for Automatic Generation of Two Dimensional Meshes

OSAMU TAKATA,[†] KOUKICHI NAKANISHI,[†] NARIAKI HORINOUCHI[†] and MAKOTO NAGAOKA[†]

The mesh generation is one of major tasks confronted in computational dynamics. The analysis accuracy and the computation cost depend directly on the generated mesh. Since various kinds of domain knowledge are needed in order to generate an acceptable mesh structure, the knowledge-based approach is effective. We have developed an expert system called GENMAI (Artificial Intelligence Mesh GENerator) for automatically generating two-dimensional meshes, which is easily applicable to various fields of computational dynamics with structured meshes. We analyzed some experienced mesh generation processes in computational fluid dynamics and plastic deformation analysis, and then could formulate these processes. From a viewpoint of this formulation, GENMAI was designed so as to be widely applicable to a large class of two-dimensional geometries. To check the validity of this formulation, we applied GENMAI to simulations of automotive engine intake flow analysis and plastic deformation analysis, and made the functionality of GENMAI clear. The characteristics of GENMAI are as follows: (1) Plural solutions can be efficiently obtained at the same time using global dependency and local dependency; and (2) The meta-level inference method and its knowledge representation are very applicable to various fields of analyses.

1. はじめに

数値計算による構造解析，塑性変形解析，衝突解析，そして流体解析などは総称して計算力学と呼ばれる．近年の計算機の性能向上にともない，その適用範囲は急激に拡大している．計算力学の手法は，メッシュ生成（前処理），解析（本計算），解析結果の評価（後処理）の3つの過程からなるが，メッシュ（あるいは格子とも呼ぶ）の質は解析結果の精度や計算時間に大きな影響を及ぼすため，質の高いメッシュを生成することが重要である．現状ではメッシュ生成のために，分野ごとに専用プログラムを開発するか，あるいは市販のメッシュ生成ツールを利用する必要がある．

分野ごとに専用プログラムを開発する場合，分野固有の知識を持った専門家が，Fortranなどの手続き型言語を用いて適切なメッシュを計算するプログラムを記述してきた．このようなプログラムは，処理対象の形状，解析手法，適用分野に固有で，専用的であることが多い．しかも，新しい処理対象に適用するにはそ

† 株式会社豊田中央研究所
Toyota Central Res. & Develop. Labs., Inc.

のたびにプログラムを修正する必要があり，多大な時間を要する．たとえば，3次元実形状の自動車の車体周りの流れを解析する場合，スーパーコンピュータを用いると本計算が5日程度で済むのに対して，適切なメッシュを得るためにはプログラムの記述を含めて1～3カ月を要している．

一方，市販のメッシュ生成ツールを利用する場合，ユーザがプログラムを開発する必要はないが，適切なメッシュを得るには分野固有の知識を必要とし，しかもその分野の専門家であっても多くの試行錯誤を繰り返さなければ目的を達成できない．

これらの問題を解決するためにいくつかのメッシュ生成システムが提案されている．提案されているものは，メッシュの体系により2つに大別される．その1つは，構造解析など有限要素法を用いた計算で主に利用される三角形の非構造格子を対象としたメッシュ生成システムである．これらに関しては，いくつかの自動化アルゴリズムが提案されている[1),2)]．もう1つは，流体解析など差分法を用いた計算で主に利用される四角形の構造格子を対象としたメッシュ生成システムである．これらは複雑な形状への対応が難しく，しかも自動生成のアルゴリズム化が困難である．

また，両者の中間的なものとして四角形の非構造格子を対象としたメッシュ自動生成システムが提案されている．たとえば，アドバンシングフロント法を四角形に改良したものや，境界から四角形メッシュを並べていく Paving 法などがある[3),4)]．これらの方法で適切なメッシュを生成するためには，それらのアルゴリズムに応じたノウハウを必要とするといった問題点がある．

以上のように，四角形を対象としたメッシュ生成に関しては様々な問題点があり，自動化が困難である．したがって，現状では三角形の非構造格子を対象としたメッシュ生成システムが主流となっている[5)]．しかし，同じ計算精度を得るためには，計算時間，メモリ容量の面で四角形の構造格子は三角形の非構造格子より優れている．また，差分法を用いた解析プログラムは四角形の構造格子しか利用することができない．なお，いったん四角形のメッシュを得ればそれから三角形のメッシュを生成することは容易であり，三角形のメッシュを用いる解析手法にも適用できる．

このように四角形の構造格子は解析手法からみた面では有用性が高いが，それを生成するためにメッシュ生成や分野固有の知識を多く必要とする．そのため，知識処理に基づいたいくつかのシステムが提案されている．しかし，これらは適用性や保守性の点で，汎用的なものとして実現されていない．なお，現在までに提案されている知識処理に基づいたメッシュ生成システムの特徴や問題点については，2.1 節で述べる．

本論文では，経験の浅いユーザでも適切な四角形の構造格子を生成でき，しかも様々な計算力学の分野に容易に適用できる汎用的な2次元メッシュ自動生成システム（GENMAI: Artificial Intelligence Mesh GENerator）について述べる．我々は，自動車の空力解析，自動車用エンジン内の流れ解析，塑性変形解析における専門家のメッシュ生成過程を分析し，そのモデルを作成した．このモデルに基づきシステムを設計し，プログラムを開発した．例として，自動車用エンジンの吸気流解析モデルおよび鍛造変形解析モデルに適用して，システムのモデル化の妥当性，推論方式および探索手法の有用性を確認した．

2. システムのアプローチ

前述のように質の良い四角形メッシュを生成するには分野固有の経験や知識を必要とするため，知識処理が有効とされている．知識処理に基づいた既存のメッシュ生成システムの特徴および問題点について検討し，我々が開発した知識処理に基づいた GENMAI についてそのねらいと考え方について述べる．

2.1 現状システムの問題点

現在提案されている知識処理に基づいたシステムは，大きく2つのクラスに分類される．1つのクラスは適用対象を限定し，浅い知識を用いて推論する解析型のシステムである[6)]．もう1つのクラスはあらかじめシステム内に登録されているプリミティブを組み合わせて解を得る合成型のシステムである．

前者の例として，NASA で開発されている EZGRID や Fannenhoffer らのシステムなどがある[7),8)]．これらは，2次元の翼周りの流れを解析対象とし，差分法向けの構造格子専用のメッシュ生成システムである．これらは解析手法や適用分野を限定しているため，複雑な形状の取扱いも比較的容易である．しかし，ユーザは形状に関する情報のほかに，解析に関するノウハウを入力する必要がある．

後者の例として，Sandia 国立研究所で開発されている AMEKS や Dabke らのシステムなどがある[9),10)]．これらは非構造格子を対象としている．あらかじめシステム内に登録されているプリミティブを組み合わせて形状を合成するため，複雑な形状への対応が困難である．以上のように，既存の知識処理に基づいたメッシュ生成システムは，異なる解析手法への拡張性や複雑な形状への適用性などに問題がある．

2.2 システムのねらい

メッシュを生成するためには，形状，境界条件，解析手法の特徴，格子体系，物理特性など，分野固有の知識や経験を多く必要とする．また，質の良いメッシュを生成するには，熟練した専門家でも多くの試行錯誤を繰り返し，多大な時間を要する．

本研究では，以下の点をねらいとしてシステムを開発する．

(1) ノウハウを知識ベース化し，経験の浅いユーザでも専門家と同程度に質の良いメッシュを生成できる．

(2) 知識の記述性，保守性の良い表現および柔軟な推論方式を用いることにより，様々な分野に容易に適用できる．

(1)に対しては，専門家の試行錯誤の過程をモデル化し，それを知識ベース・システムの枠組みで実現する．(2)に対しては，メッシュ生成における標準的な分野に依存しない手順と分野固有の知識とを分類し，手順的な知識をメタ・ルールで，分野固有の知識をルールで表現する．また，複数の競合解消戦略を導入し，それらをメタ・ルールで制御することにより，分野固有の知識を容易に記述可能とし，システムの適応性，柔軟性を高める．

2.3 システムの考え方

自動車の空力解析，自動車用エンジン内の流れ解析，塑性変形解析における専門家のメッシュ生成過程を分析した結果，以下に示すプロセスにより，メッシュを生成していることが明らかとなった．特に，質の良いメッシュを生成するためには，(1)のプロセスで多くの試行錯誤を繰り返していた．

(1) 与えられた領域に関して，全体領域を単純な形状からなる複数の小領域（副領域と呼ぶ）に分割する（**図1** の(a)）．

(2) 各副領域について，格子生成手法（トランスファイナイト内挿法など）を適用し，格子を生成する（図1の(b)）．

(3) 隣接する副領域間の整合性を図り，次に全体に対して調整する．

(1) の副領域分割法について以下に説明する．**図2**(a) のように比較的単純な四辺形領域に関しては，領域全体に単純な格子生成手法（たとえば，トランスファイナイト内挿法など）を用いても質の良いメッシュを生成することができる．しかし，図2(b) のようにメッシュを生成する領域に著しいくぼみがある場合には，領域をはみ出してメッシュを生成することがある．このようなときに，図2(c) のように分割線を入れて領域を2つに分割してから各領域についてメッシュを生成する方法を副領域分割法という．

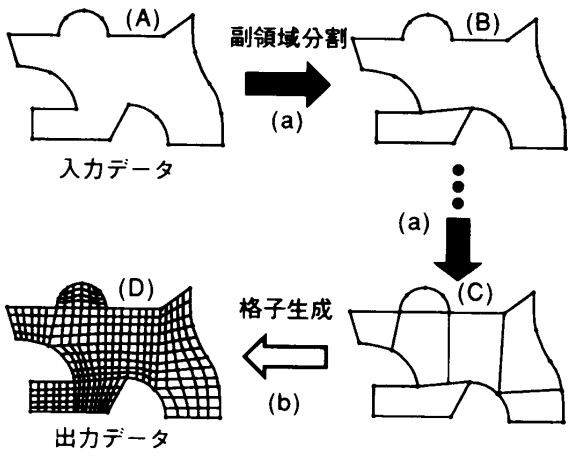

図1 メッシュ生成プロセス
Fig. 1 Mesh generation process.

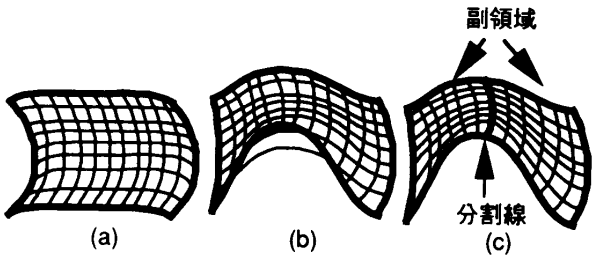

図2 副領域分割
Fig. 2 Subdivision.

副領域分割法は，様々な計算力学の分野で用いられている．また，副領域分割法では，異なる分野でも全体の処理フローがほぼ同じで，各処理においては分野ごとに利用する知識の内容は異なるが，それらの知識の適用方法が類似している．したがって，分野に依存しない共通のメッシュ生成システムを構築可能である．

3. システム概要

本章では実現手法の概要，すなわちシステム構成，データ表現および処理手順について述べる．

3.1 システム構成

本システムは推論部および探索制御部から構成される．推論部は，メタ知識ベースとその推論機構（メタ推論機構），知識ベースとその推論機構，およびワーキング・メモリからなる．探索制御部は推論部で決定されたワーキング・メモリの依存関係を保持し，効率良く推論を進めるものである．推論部および探索制御部はPrologにより記述され，数値計算およびインタフェース部はC言語により記述されている．

3.2 データ表現

システムでは，入出力データおよびシステム内で扱うデータをオブジェクト☆として実現している．以下に，システムで用いるオブジェクト・クラスを示す．太文字はオブジェクト・クラスの名称を表す．

- 領域に関するもの
  - **基本領域** 副領域分割の対象となる領域（入力形状（図1(A)），分割線により分けられた2つの領域（図1(B)の2つの領域））．
  - **副領域** 基本領域の中で単純な形状をしていて分割する必要のない小領域（図1(C)の各々の領域）．
- 線分に関するもの
  - **セグメント** 形状を表す線分．線種として直線，円弧，スプライン曲線があり，各線分は壁，流入，流出，内部・外部物体，自由表面，分割線などの属性を持っている．
- 点に関するもの
  - **形状点** セグメントの端点および内部点

次にデータの保存法について述べる．3.3節で示すように1つの基本領域に関して複数の分割パターンが生じる．本システムでは，複数の解を同時に求めるため，分割パターンごとにデータを管理している．また，同じセグメントを構成要素として持つ基本領域の属するデータを同一世界（ATMSにおける文脈）として管理する[11),12)]．

3.3 処理手順

モデル化された専門家のメッシュ生成過程について述べる（**図3**）．(2)～(4)の各プロセスは，すべての基本領域が副領域になるまで再帰的に繰り返され，(1)～(4)の各プロセス（図3の中で太線で囲まれたもの）は分野固有の知識を用いて処理され，(1)～(8)の処理の流れはメタ・ルールにより記述される．

(1) **入力データの内部表現への変換**：入力データを，システムで用いる内部データ（基本領域，セグメント，点）に変換する．

(2) **基本領域の初期化**：基本領域に関して，分割線の始点候補，分割線を出すべき方向などを決定する．

(3) **基本領域の分割**：本システムの主要部分であり，

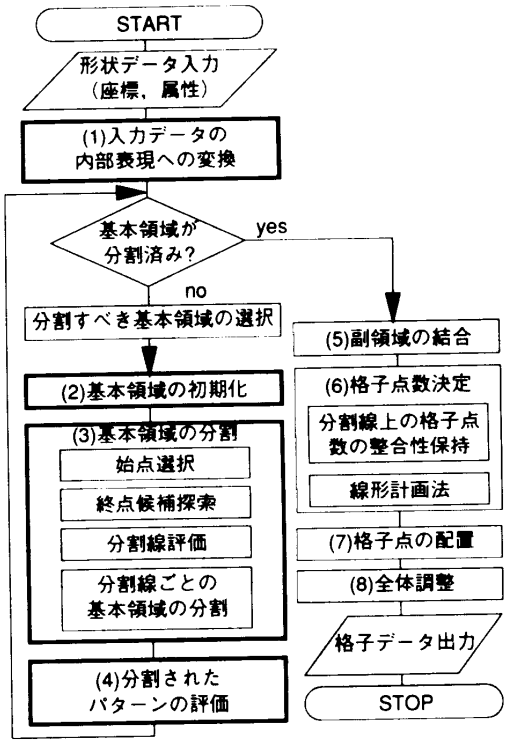

図3 GENMAIのメッシュ生成プロセス
Fig. 3 Mesh generation process in GENMAI.

与えられた基本領域を分割線により，2つの基本領域に分割する．**図4**に示すように，1つの基本領域に関して複数個の分割始点（図中の●）が存在し，各始点に関して複数個の終点（図中の○）が存在する．そのため，1つの基本領域から複数の分割パターンが生じる．このうち，領域や隣接するセグメントなどに関する制約を満たしたもののみ分割パターンの候補となる．制約を満足するパターンが1つも存在しない場合は，1つ前に戻って，(4)の“次に評価値の高い”ものについて処理を進める．

(4) **分割されたパターンの評価**：(3)で得られた制約を満足した複数の分割パターンから最適なものを選び出すために，各分割パターンを評価し，評価値の低いものを枝刈りする．ただし，評価値の高いものが後の分割に失敗した場合，“次に評価値の高い”分割パターン（枝刈されたもの）について処理を進める．

(5) **分割された副領域の結合**：(2)～(4)のプロセスにより再帰的に分割された基本領域（最終的には副領域）を集めて，入力の基本領域を得られた副領域で再構成する．

(6) **セグメント上の格子点数の決定**：各副領域の対辺の格子点数が等しくなるように，各セグメント上

☆ 本システムでは，形状データ（セグメント，点，基本領域など）をオブジェクトとして実現する．扱うデータはフレーム構造と類似しているが，次の点からフレームではなく，オブジェクトとして表現した．各データは抽象化されたデータ・タイプ（クラス）とその実体であるデータ（インスタンス）から構成される．各データの操作（参照，登録，削除など）はクラスごとに共通な関数（メッセージ）で行われ，異なるクラスへの操作も同じ関数で行われる．

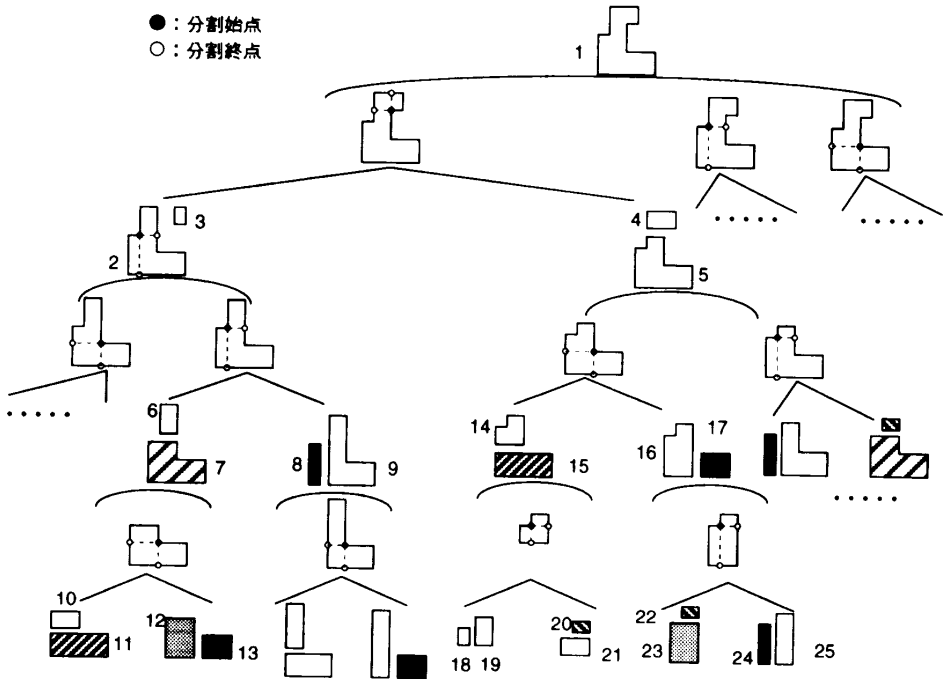

図4 生成分割パターン
Fig. 4 Examples of subdivision patterns.

に配置する格子点数を決定する☆.

(7) **格子点の配置**：セグメント上に配置する格子点の位置を決定し，トランスファイナイト内挿法などにより，内部に分布させるべき格子点の位置を計算する.

(8) **全体調整**：隣接する副領域の格子間隔が急変しないように格子点の配置を調整し，最終的なメッシュを生成する.

4. 探索手法

前述のとおり，本システムの処理手順に従えば，1つの基本領域から複数の分割パターンが生じる．たとえば，図4において，基本領域1に関して，3つの分割始点（図中の●）が存在し，各始点に関して2つの分割終点（図中の○）が存在する．したがって，[基本領域2, 基本領域3]，[基本領域4, 基本領域5] など複数の分割パターンが生成される．また，分割線を出す順番により，同じ形状の基本領域（図4で同じハッチングが付いている領域）が生成される．そのため，単純に処理を進めると組合せ爆発が生じる.

1つの世界（分割前の基本領域の属する世界）から複数の世界（分割パターン）が生成されるときに，どの世界について探索（"分割すべき基本領域の選択"）を進めるかという世界に関する情報（依存関係）を大域的に管理し，探索が失敗して，後戻りするときの情報を世界ごとに管理する．本システムでは，簡易的なATMSを構築し，これらの依存関係を用いて，複数の解を同時に効率良く求める．これは，巡回型のATMSに相当する[13),14)].

4.1 局所的な依存関係

本システムでは，ある世界から複数の世界が生成された場合，すべての世界について探索を進めるのではなく，その中で評価値の高い上位 n 個の世界について処理を進める（5.2節の分割パターンの評価を参照）．上位 n 個の世界がすべて副領域に分割された場合探索は終了する．一方，上位 n 個の世界の中で片方あるいは両方の世界の分割が継続できなくなれば，その世界をNGとし，後戻りを行い，"次に評価値の高い"世界について探索を進める.

制約条件を満足した世界の中で，評価値を計算し，上位 n 個のものをin-worldに，その他のものをout-worldに登録する．この依存関係は各世界ごとに管理される☆☆.

☆ 全体の面積とユーザの設定した格子点数から基準となる格子点間隔を求め，各セグメント上に配置すべき点の数の下限を決める．次に，副領域の対辺の格子点数が等しく，かつ格子点数が最小になるようにセグメント上に置く格子点数を線形計画法で決める.

☆☆ 各世界は，TMSで用いられるin, outのほかにNGという状態を持つ[15)]．いったんNGになったものは永久的にNGのままである．これはNG-worldとして大域的に管理される（4.2節参照）.

in-world 制約条件を満足して，現在の状況では真（分割すべきノード）であると信じられている世界

out-world 制約条件を満足しているが，評価値が低いため，現在の状況では真であると信じられていない世界．in-world のものがNGとなったときに，その中で評価値が高いものが in-world に移る．

たとえば，図4の世界1（基本領域1の属する世界）における局所的な依存関係は，$n = 1$ のとき，in-world={{2,3}}, out-world={{4,5}} となる．

4.2 大域的な依存関係

本システムでは，同時に複数個の解を求める．どの世界について探索を進めるかという情報を大域的に管理する．新しく世界を生成した（分割により基本領域が生成された）ときに，過去に生成された世界の中で同一のもの（同じセグメントを構成要素として持つ基本領域の属する世界）があるかを調べ，同一のものがあればそれらをまとめて管理する．

依存関係として以下の3つのものがあり，これらは大域的に保持される．

expanded-world 過去に分割した世界

expanding-world これから分割する世界

NG-world これまでの分割で失敗した（制約条件を満足しなかった）世界

たとえば，図4で基本領域1から [基本領域2, 基本領域3], [基本領域4, 基本領域5] などを生成し，基本領域2, 3について探索を進める場合，expanded-world={1}, expanding-world={2,3}, NG-world={} として大域的に管理される．

4.3 2つの依存関係を組み合わせた探索手法

以上の局所的な依存関係と大域的な依存関係を組み合わせた探索手法について述べる（**図5**）．(1)～(4)が図3の“分割すべき基本領域の選択”に対応する．

(1) expanding-world の先頭の世界（W）を1つ取り出す．

(2) W が副領域のときは(1)に戻り，expanding-world の次の世界について探索を進める．

(3) expanded-world の中に W と同じものがあれば，その世界にリンクを付け，その結果を利用し，W については探索しないで，(1)に戻る．

(4) NG-world の中に W と同じものがあれば，“NG の処理”(9)へ進む．

(5) 上記以外の場合は，W について探索を進める．

(6) W から新しく m 個の世界（W1,...,Wm）が生成されるとする．

(7) 新しく世界が生成されないとき（$m = 0$），W および W のペアとなる世界を NG とし，“NG の処理”(9)へ進む．

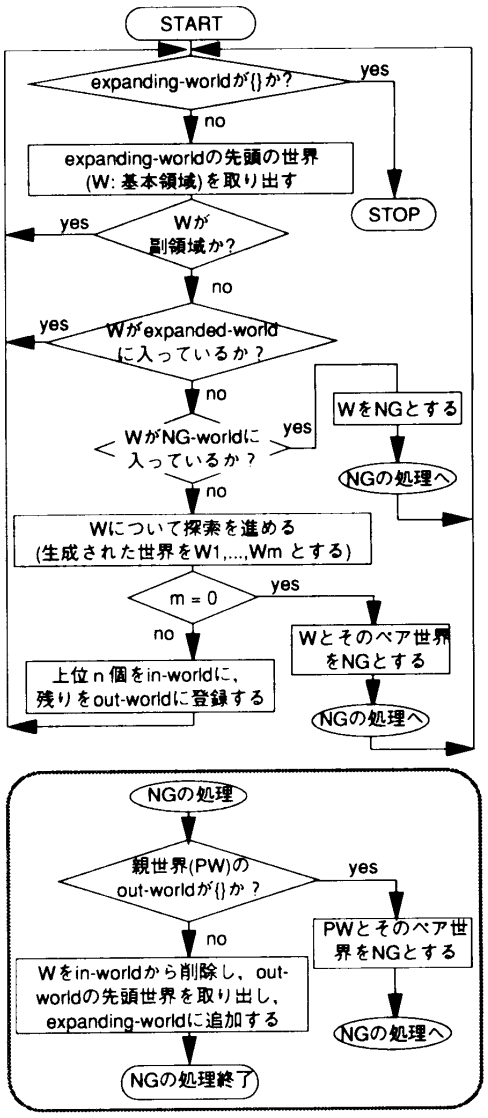

図5 探索手法
Fig. 5 Search method.

(8) 各分割パターンについて評価値を計算し，上位 n 個を in-world に登録するとともに，expanding-world に追加する．残りのものは out-world は {} である．(1)に戻る．ただし，$m \leq n$ のときは，out-world には登録しない．

(9) “NG の処理”

(a) W を生成した親の世界（PW）の out-world が {} かを調べる．

(b) out-world が {} であれば，PW および PW とペアとなる世界を NG とし，それぞれについて“NG の処理”を行う．

(c) out-world が {} でなければ，PW の in-world から W を削除し，out-world の中で“次に評価値の高い”世界を out-world から削除し，in-world

表 1 依存関係に基づく探索例
Table 1 Example of search method.

局所的依存関係			大域的依存関係			備考
world	in -world	out -world	expanded-world	expanding -world	NG -world	
1	{{2,3}}	{{4,5}}	{1}	{2,3}	{}	(a)
2	{{6,7}}	{{8,9}}	{1,2}	{6,7,3}	{}	
6	{{6}}	{}	{1,2,6}	{7,3}	{}	副領域 (b)
7	{{10,11}}	{{12,13}}	{1,2,6,7}	{10,11,3}	{}	
10	{{10}}	{}	{1,2,6,7,10}	{11,3}	{}	副領域
11	{}	{}			{11}	NG(c)
10	{}	{}			{10,11}	NG(d)
7	{{12,13}}	{}	{1,2,6,7,10}	{12,13,3}	{10,11}	
12	{{12}}	{}	{1,2,6,7,10,12}	{13,3}	{10,11}	副領域
13	{{13}}	{}	{1,2,6,7,10,12,13}	{3}	{10,11}	副領域
3	{}	{}			{3,10,11}	NG(e)
2	{}	{}			{2,3,10,11}	NG
1	{{4,5}}	{}	{1,2,6,7,10,12,13}	{4,5}	{2,3,10,11}	
4	{{4}}	{}	{1,2,6,7,10,12,13,4}	{5}	{2,3,10,11}	副領域
5	{{14,15}}	{{16,17}}	{1,2,6,7,10,12,13,4,5}	{14,15}	{2,3,10,11}	
14	{{18,19}}	{{20,21}}	{1,2,6,7,10,12,13,4,5,14}	{18,19,15}	{2,3,10,11}	
18	{{18}}	{}	{1,2,6,7,10,12,13,4,5,14,18}	{19,15}	{2,3,10,11}	副領域
19	{{19}}	{}	{1,2,6,7,10,12,13,4,5,14,18,19}	{15}	{2,3,10,11}	副領域
15				{}	{2,3,10,11}	11 と同じ NG(f)
14	{}	{}			{14,2,3,10,11}	
5	{{16,17}}	{}	{1,2,6,7,10,12,13,4,5,14,18,19}	{16,17}	{14,2,3,10,11}	
16	{{22,23}}	{{24,25}}	{1,2,6,7,10,12,13,4,5,14,18,19,16}	{22,23,17}	{14,2,3,10,11}	
22	{{22}}	{}	{1,2,6,7,10,12,13,4,5,14,18,19,16,22}	{23,17}	{14,2,3,10,11}	副領域
23			{1,2,6,7,10,12,13,4,5,14,18,19,16,22}	{17}	{14,2,3,10,11}	12 と同じ
17			{1,2,6,7,10,12,13,4,5,14,18,19,16,22}	{}	{14,2,3,10,11}	13 と同じ

と expanding-world に移し，(1) に戻る．

次に，上記探索手法を図 4 の例を用いて簡単に説明する．図 4 において領域の横の数字は，生成された基本領域およびそれが属する世界の番号を示す．そのときの探索過程で更新される各世界の局所的な依存関係と大域的な依存関係を**表 1** に示す．ただし，説明を簡単化するために，同時に展開する個数 (n) を 1 とする．

初期状態として，expanding-world={1} となり，その他は {} である．expanding-world の先頭の世界を 1 つ取り出す（世界 1）．世界 1 が副領域か，あるいは expanded-world, NG-world の中に世界 1 と同じものがあるかを調べる．上記以外であるため，世界 1 について探索を進める．図 4 に示すように [基本領域 2,3], [4,5] などの分割パターンが生成される．2 行目の右 2 つの領域からも分割パターンが生成されるが，説明を簡単にするためにこれらについては省略する．$n = 1$ であるため，世界 1 の局所的な依存関係は，評価の結果，in-world = {{2,3}}, out-world = {{4,5}} となる．このとき，大域的な依存関係は，expanded-world = {1}, expanding-world = {2,3}, NG-world = {} となる（表 1 備考 (a)）．

同様に探索を進め，世界 6 の分割を考える．世界 6 の属する領域は副領域であるため，in-world = {{6}}, out-world = {} となる（表 1 備考 (b)）．

さらに探索を進め，世界 11 の分割を考える．世界 11 は制約条件を満足しないため，NG となり，NG-world に登録される．このとき，in-world, out-world はそ

れぞれ{}, {}となる（表1備考(c)）．世界11がNGと分かったためにそのペアである世界10をNGとし，NG-worldに登録し，“NGの処理”に進む（表1備考(d)）．ただし，expanded-worldから世界10は消去しない．

“NGの処理”としては，世界10，11の親世界7に後戻りを行い，out-worldの先頭の要素{12,13}をin-world，expanding-worldに移す．

次に，世界12，13について探索を進める．ともに副領域であるので，世界7についての探索はいったん終了する．同様に，世界2についての探索もいったん終了する．

次に，世界3について探索を進める．世界3が制約条件を満足しないため，“NGの処理”を行う（表1備考(e)）．

さらに探索を進め，世界15の分割を考える．このとき，世界15とNG-worldの中の世界11が同じであるため，“NGの処理”を行う（表1備考(f)）．さらに探索を進め，世界17の分割を考える．世界17と世界13が同一世界であるため，世界17について探索は進めない．ここで，expanding-worldが{}となり，探索は終了する．その結果，[4, [[22, 12],13]]という分割パターンを得る．

以上のように大域的な依存関係と局所的な依存関係を組み合わせることにより，同一世界に関して重複した計算（分割）を行わないため探索効率が良い．

また，ペアの世界の片側が最終的に分割できないと分かったとき（NGのとき），もう一方の世界をNGとすることにより探索の効率化を図ることができる．これは，ATMSのhyper resolutionに相当する[16]．たとえば，[A, B]という分割の場合，Aが[A1, A2], [A3, A4]という2つの分割パターンを持つとき，[A, B]は，[[A1, A2], B], [[A3, A4], B]となる．このとき，[A1, A2], [A3, A4]がともに展開できないときにBをNGにすることである．

上記探索戦略では，失敗したノードに一番近いものに戻って探索を進めるため，動作としてはPrologの年代順バックトラッキングに似ているが，過去に計算した世界に関して再計算（分割）しないため，効率が良い．

5. 推論方式と知識表現

自動車の空力解析，自動車用エンジン内の流れ解析，塑性変形解析の分野における知識およびその知識の使われ方を分析し，メッシュ生成に関して分野に依存しない基本的な手順と分野固有の知識に分類した．本章では，推論方式および知識ベースについて述べる．

5.1 推論方式

本システムでは，基本領域，副領域などのデータをインスタンスで表現している．メッシュ生成過程では，知識ベースを用いてこれらのインスタンスの属性値を操作（参照，追加，削除および修正）し，インスタンスの状態を変更することにより，与えられた基本領域を副領域に分割する．

メッシュ生成の処理の流れは，異なる分野でも全体の処理フローがほぼ同じで，各処理においては分野ごとに利用する知識の内容は異なるが，それらの知識の適用方法が類似している．すなわち，どのインスタンス群についてどのルール群をどのように適用するかあらかじめ決まっている．

そこで，次に示すメタ・ルールによる2段階の推論方式を用いて，メッシュ生成を実現している（**図6**, **図7**のメタ・ルールの(1), (2)参照）．

(1) **インスタンスの選出**：あるオブジェクト・クラスのインスタンスに関して，制約条件（分野に依存しない知識）を満たすものを集める．

(2) **分野固有の知識の適用**：(1)で集められた各インスタンスについて，制約条件（分野固有の知識）を満たすものの属性値を操作（参照，追加，削除および修正）する．

知識ベースシステムでは，同時に実行可能なルールや実行可能なインスタンスが存在する場合，これらの競合集合の中から実行すべきものを決定する必要がある．本システムでは同時に複数の解を得るため，1つのインスタンスについて複数のルールを適用したり，あるいはすべてのインスタンスについて複数のルールを適用することが必要となる．また，4章で述べたように上位n個のインスタンスについて探索を進めるために，優先順位の高いインスタンス，あるいは優先順位の高いルールを適用する戦略が必要となる．したがって，MEA，LEXなどの単純な競合解消戦略ではこれらを実現することは困難である[17]．

本システムでは，**表2**に示す複数の競合解消戦略を用い，メタ・ルールの中のinference_engine述語で，競合解消戦略，適用すべきルール群，インスタンス群，および対象とする世界を指定する．なお，照合処理，競合解消処理およびルールの発火は実行時になされる．

- **inference_engine（戦略，ルール群，インスタンス群，世界）**

次に，メタ・ルールおよびルールの例を示す．図6に示したメタ・ルールは戦略3（wms-rule）の例である．これは，(1)まず，与えられた基本領域（BD）か

○メタ・ルールの例（分割始点の決定）
(1)基本領域(BD)の構成セグメント(SEGS)の始点(PTS)を集める
(2)PTSについて，ルール群select_start_pointを戦略wms-ruleにより推論する
○ルール群の例（select_start_point：隣接するセグメントに関する制約）
rule1: 隣接するセグメントが直線と直線の制約
rule2: 隣接するセグメントが直線と円弧の制約
rule3: 隣接するセグメントが直線とスプラインの制約
・・・・
○ルールの例（rule1：直線と直線の場合）
点PTに隣接する2つのセグメントを集める(Pseg, Nseg)
Pseg, Nsegが直線であるかを調べる
PsegとNsegの交点角（なす角）を計算する
データベースからのしきい値と交点角を比較する
条件を満たせばPTに分割始点属性を付加する

○分割始点の決定メタ・ルール［内部表現］
```
meta_rule(select_start_point, BD, W):-
  getData(basic_domain, BD, W, segment, SEGS),
  getDataAll(segment, SEGS, W, start_point, PTS),
  -->
  inference_engine(wms_rule, select_start_point, PTS, W).
```

○rule1：直線と直線の場合［内部表現］
```
rule(select_start_point, rule1, W, PT):-
  getData(point, PT, W, previous_segment, Pseg),
  getData(point, PT, W, next_segment, Nseg),
  getData(segment, Pseg, W, line_type,straight),
  getData(segment, Nseg, W, line_type,straight),
  intersect_angle(in, straight, straight, Pseg, Nseg, W, ANGLE),
  data_base([straight], test_intersect_angle, INT_ANGLE),
  INT_ANGLE < ANGLE,
  -->
  putData(point, PT,W, division_point, yes).
```

図6 知識の例（分割始点の決定）
Fig. 6 Example of Knowledge (Division start point).

○メタ・ルールの例（分割パターンの評価）
(1)基本領域(BD)の分割線(DLS)を集める
(2)DLSについて，ルール群evaluate_dlineを戦略wms-rulesにより推論する
○ルール群の例（evaluate_dline：分割パターン評価値の設定）
rule1: 四辺の対辺比に関する評価
rule2: 四頂点の角度に関する評価
rule3: 分割線の傾きに関する評価
・・・・
○ルールの例（rule1：四辺の対辺比に関する評価）
分割線DLで分割される基本領域を集める(BD1, BD2)
BD1の四辺の構成セグメントを集める
構成セグメント長さおよび各辺の長さを求める
対辺比を計算する e1= f(四辺の長さ)
f(四辺の長さ): 対辺比に関する評価関数
BD1にe1を登録する
BD2について同じ処理を行なう

図7 知識の例（分割パターンの評価）
Fig. 7 Example of Knowledge (Evaluation of regions).

らセグメントの始点であるすべての点インスタンス（PTS）を集める．(2)次に，集められたすべての点インスタンスに対して隣接するセグメントの制約のルール群（select_start_point）を適用し，ルールにより分割始点属性を付加する．この場合，先頭から順にルールの条件部を調べて，最初に満足した1つのルールについて結論部を実行する．

また，図7に示したメタ・ルールは戦略4（wms-rules）の例である．これは，すべての分割線セグメント・インスタンス（DLS）に関して，各評価項目を計算するルール群evaluated_line（分割パターンの評価値の決定）を調べ，条件部を満足したすべてのルールについて結論部を実行する．

戦略3と戦略4はすべてのインスタンスについてルールを適用する競合解消戦略である．これに対して，1つのインスタンスについてルールを適用するものが，戦略1（wm-rule），戦略2（wm-rules）である．

本システムでは，処理手順をメタ・ルールで記述しているため，分野によって処理手順が異なる場合，メタ・ルールの一部を修正することにより対応することができる．また，分野固有の知識は，適用すべき処理手順ごとにグルーピングされているため，知識の保守が容易である．なお，ここで提案した競合解消戦略はこの問題に固有なものではなく，複雑な処理手順を記述する問題向きで，汎用的な推論機構として利用できる[18]．

5.2 知識ベース

3.3節で述べたように(1)～(4)の処理は知識に基づいて進められる．その中で再帰的に繰り返し呼び出される(2)～(4)で使う知識について簡単に述べる．なお，これらの知識は，表3のオブジェクト操作述語およびPrologの述語を用いて記述される（図6の内部表現を参照．ただし，大文字で始まるものは変数を表す）．

基本領域の初期化 分割された基本領域の属性を新たに定義する知識である．
- 分割線の始点候補の決定
- 分割線を出すべき方向の決定

基本領域の分割 基本領域を分割線により2つの基本領域に分割する知識および分割された領域の良否を決定する知識である．
- 分割線の始点決定（図6参照）
- 分割線を出す順番の決定
- 分割線の終点の決定
- 分割線の良否の決定
- 分割された領域の良否の決定

表2 競合解消戦略
Table 2 Conflict resolution strategy.

戦略名	競合解消戦略
戦略1 (wm-rule)	先頭のインスタンスについて，先頭から順にルールを調べ，条件部を満足した最初の1つのルールについて結論部を実行する．もし，条件を満足するルールが1つもない場合には，次のインスタンスに移る．
戦略2 (wm-rules)	先頭のインスタンスについて，すべてのルールを調べ，条件部を満足したすべてのルールについて結論部を実行する．もし，条件を満足するルールが1つもない場合には，次のインスタンスに移る．
戦略3 (wms-rule)	すべてのインスタンスについて，先頭から順にルールを調べ，条件部を満足した最初の1つのルールについて結論部を実行する．この操作をすべてのインスタンスに適用する．
戦略4 (wms-rules)	すべてのインスタンスについて，すべてのルールを調べ，条件部を満足したすべてのルールについて結論部を実行する．この操作をすべてのインスタンスに適用する．

表3 オブジェクト操作述語
Table 3 Object manipulation predicates.

述語名	使い方
get	インスタンス属性の参照 get(class, instance, world, slot, Value)
getAll	複数のインスタンス属性の参照 getAll(class, instance, world, slot, Value)
put	インスタンス属性の登録 put(class, instance, world, slot, value)
putAll	複数のインスタンス属性の登録（同一値を登録） putAll(class, instance, world, slot, value)
putsAll	複数のインスタンス属性の登録（個別値を登録） putsAll(class,instance,world,slot,values)
collect	インスタンス属性の検査：インスタンス属性がvalueであるインスタンスを集める collect(class,instances,world,slot,value, Instances)
create	インスタンスの作成 create(class, world, Instance)

分割パターンの評価 生成された複数個の分割パターンに評価値を付けるルールである．各分割パターンの評価値は，i 番目の評価項目の評価値 e_i とその重み w_i との積和 $\sum_i w_i e_i$ により計算される．各分割パターンの i 番目の評価値 e_i はルールの中で計算される（図7参照）．どの評価項目を選択するか，各評価値の計算はどのようにするか，あるいはそのときの重みをどのようにするかは，専門家によって経験的に決められている．たとえば，評価項目として，以下のものなどがある．

- 四辺の対辺比（図7参照）
- 四頂点の角度
- 分割線と交わるセグメントのなす角度

6. 実 行 例

3.3節のメッシュ生成過程のモデル化に基づいてシステムを試作した．なお，適用分野として，鍛造変形解析および自動車用エンジンの吸気流解析を取り上げ，それぞれの知識ベースを構築した．

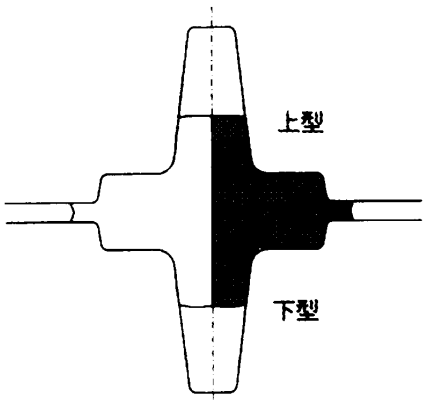

図8 鍛造変形解析モデルの入力形状
Fig. 8 Input form of a plastic deformation.

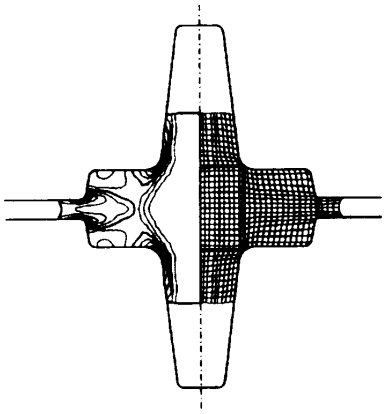

図9 鍛造変形解析の出力例
Fig. 9 Output for a plastic deformation.

図8に示す鍛造変形解析モデルのハッチング部分についての出力例を図9に示す．図9の中心軸より右側は副領域分割パターンと計算格子を表している．ただし，太線で区切られている領域が得られた副領域である．また，左側は得られたメッシュを用いた計算結果で，変形の厳しさの程度を意味する相当ひずみ分布を示している．得られた副領域分割パターンは専門家が考案するものと同等で，得られた計算結果も経験的な推測と一致する．

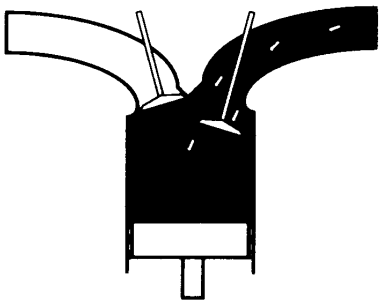

図10 エンジン吸気流解析モデルの入力形状
Fig. 10 Input form of an automotive engine intake.

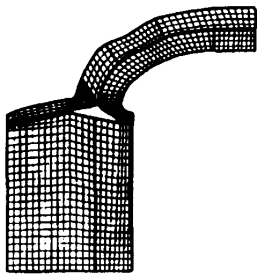

図11 エンジン吸気流解析モデルの格子出力例
Fig. 11 Output for an automotive engine intake.

図10に示す自動車用エンジンの吸気流解析モデルのハッチング部分について，格子生成の出力例を図11に示す．図11の計算格子を用いて流れを計算した結果は，速度，圧力などの物理量の分布に関して実験値や他の計算結果と比べて良い結果を示している[19]．

次に，依存関係を用いた探索の例について述べる．図10について探索したときの流れを図12に示す．ただし，同時に複数展開する世界の数 n を2とした．細枠の四角は基本領域を，太枠の四角は副領域を示す．四角の中の数字は探索した順番（生成された基本領域の属する世界の番号に対応している）を示している．また，細線の四角（基本領域）の横のアルファベットは基本領域において選択された始点の名前である．この例の場合，入力の基本領域1においてa, b, c, d, eという5つの分割線の始点がある．たとえば，基本領域1について始点aから[2,3]と[4,5]という2つの分割パターンが生成される．×印のついた基本領域は分割に失敗し，バックトラッキングが起こったことを表している．ハッチングの基本領域はバックトラッキング時に使用される候補（out-world）である．‖の付いている世界は同一世界が存在するもので，再計算しない．

この例題の場合，生成する世界の数は，本手法では145個であるのに対して，単純な手法では180個であった．1つの世界を探索するのに20秒から40秒程度かかり，依存関係を計算するのは1秒弱であった．実際の実行時間はSun SPARC Station2を用いて，本手法では14分，単純な手法では19分かかった．また，同様な結果を他の例でも得ており，これにより探索手法の有効性が明らかとなった．

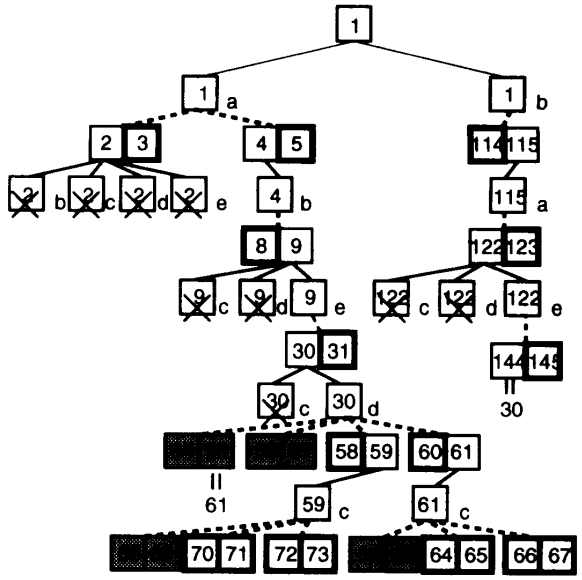

図12 探索の流れの例
Fig. 12 Example of search flow.

以上，塑性変形解析および自動車用エンジン内の流れ解析の分野に関して，それぞれの知識ベースを構築し，メッシュを生成することにより，異なる分野への適用が可能であることを確認した．

また，従来経験の浅いユーザでは困難であった問題に対しても，専門家と同等な案を得ることができた．しかも，専門家でも1つの案を作成するのに半日から一日程度かかっていたものを短時間で妥当な複数個の案を生成することが可能となった．

7. まとめ

我々は，自動車の空力解析，自動車用エンジン内の流れ解析および塑性変形解析における専門家のメッシュ生成過程を分析し，そのモデルを構築した．また，このモデルに基づきシステムを設計し，プログラムを開発した．自動車用エンジンの吸気流解析モデルおよび鍛造変形解析モデルに対してこのメッシュ生成システムを適用し，システムのモデル化の妥当性および推論方式，探索手法の有効性を確認できた．

上記2つの分野に関して，それぞれの知識ベースを構築し適用することにより，様々な分野へ容易に適用できることが確認された．また，経験の浅いユーザでも専門家と同等な案を短時間に得ることが可能となった．

本システムの特徴は，以下のとおりである．

- 分野に依存しない標準的な手順をメタ・ルールで，分野固有の知識をルールで表現し，ルールを適用すべき処理手順ごとにグルーピングしているため，ルールの保守が容易である．
- メタ・ルールで適用すべきデータ，知識ベースおよび競合解消戦略を指定することにより，複雑な処理の流れを的確に記述できる．
- 大域的な依存関係と局所的な依存関係を組み合わせた探索手法により，同時に複数の案を効率良く求めることができる．

今後の課題として，新しい形状などに対応するために知識ベースを自動的にチューニングする学習手法などを導入する必要がある．現状では新しい形状に対応する場合，システム開発者が過去の知識の整合性などを考慮して，新しい知識を導入している．このような状況では，帰納的な学習手法などを用いて自動的に知識を獲得する手法が有効である[20]．

謝辞 本論文をまとめるにあたり，ご指導をいただいた名古屋大学大学院工学研究科情報工学専攻渡邉豊英教授に深く感謝いたします．

参考文献

1) Canvendish, J.C.: Automatic Triangulation of Arbitrary Planar Domains for the Finite Element Method, *Int'l J. Numer. Methods Eng.*, Vol.8, pp.679–697 (1974).
2) Probert, E.J., Hassan, O., Morgan, K. and Peraire, J.: An Adaptive Finite Element Method for Transient Compressible Flows with Moving Boundaries, *J. Comp. Phys.*, Vol.32, pp.751–765 (1991).
3) Zhu, J.Z., Zienkiewicz, O.C., Hinton, E. and Wu, J.: A New Approach to the Development of Automatic Quadrilateral Mesh Generation, *Int'l J. for Numerical Methods in Engineering*, Vol.32, pp.849–866 (1991).
4) Blacker, T.D. and Stephenson, M.B.: PAVING: A New Approach to Automated Quadrilateral Mesh Generation, *Int'l J. for Numerical Methods in Engineering*, Vol.32, pp.811–847 (1991).
5) (社) 日本機械学会：RC112 次世代計算力学システムに関する研究分科会研究成果報告書 (1994).
6) 石塚 満，小林重信：エキスパートシステム開発，丸善 (1991).
7) Andrews, A.E.: Knowledge-Based Zonal Grid Generation for Computational Fluid Dynamics, *NASA Conf. Publ.*, Vol.NASA-CP-3019, pp.73–80 (1988).
8) Dannenhoffer, J.f.: Computer-Aided Block Structuring Through the Use of Optimization and Expert System Techniques, *AIAA-91-1585-CP*, pp.654–661 (1991).
9) Blacker, T.D., Mitchiner, J.L., Phillips, L.R. and Lin, Y.T.: Knowledge System Approach to Automated Two-Dimensional Quadrilateral Mesh Generation, *ASME Int'l Comput. Eng. Conf.*, pp.153–162 (1988).
10) Dabke, P.D., Haque, I., Srikrishna, M. and Jackson, J.E.: A Knowledge-Based System for Generation and Control of Finite-Element Meshes in Forging Simulation, *Materials Engineering and Performance*, Vol.1, No.3, pp.415–428 (1992).
11) de Kleer J.: An Assumption-base TMS, *Artificial Intelligence*, Vol.28, pp.127–162 (1986).
12) 白井良明：人工知能の理論，コロナ社 (1992).
13) 溝口理一郎：エキスパートシステム II，朝倉書店 (1993).
14) de Kleer J.: Back to Backtracking: Controlling the ATMS, *Proc. AAAI-86*, pp.910–917 (1986).
15) Doyle, J.: A Truth Maintenance System, *Artificial Intelligence*, No.12, pp.231–272 (1979).
16) de Kleer J.: Extending the ATMS, *Artificial Intelligence*, No.28, pp.163–196 (1986).
17) 溝口理一郎：エキスパートシステム I，朝倉書店 (1993).
18) 高田 修，中西広吉ほか：鍛造工程設計支援システム，第2回 FAN シンポジウム講演論文集，No.920–87，日本機械学会，pp.191–196 (1992).
19) 杉浦繁貴ほか：吸気系内流れの数値解析，自技会学術講演会前刷集，No.882141，pp.563–566 (1988).
20) Dolsak, B., Jezernik, A. and Flasker, J.: A Contribution to Development of an Expert System for Finite Element Mesh Generation, *Proc. World Congress on Expert Systems*, pp.2047–2057 (1991).

(平成7年4月21日受付)
(平成8年6月 6 日採録)

高田 修（正会員）

1958年生．1983年名古屋大学大学院工学研究科情報工学専攻修士課程修了．同年（株）豊田中央研究所入社．現在，ソフトウェア研究室研究員．設計・生産分野における知識処理の応用に関する研究に従事．平成6年度日本塑性加工学会会田技術奨励賞．人工知能学会会員．

中西　広吉

1960年生．1985年名古屋大学大学院工学系研究科金属・鉄鋼工学専攻修士課程修了．同年（株）豊田中央研究所入社．現在，材料加工研究室研究員．塑性加工のうち，特に鍛造加工の工程設計における知識処理や塑性変形解析，および解析パラメータ測定などの計算機支援技術に関する研究に従事．平成6年度日本塑性加工学会会田技術奨励賞．日本塑性加工学会会員．

堀之内成明（正会員）

1961年生．1985年名古屋大学大学院工学研究科情報工学専攻修士課程修了．同年（株）豊田中央研究所入社．現在，数理・情報研究室研究員．自動車の空力解析，および非圧縮性流体解析に関わる数値計算手法の研究に従事．日本応用数理学会，自動車技術会各会員．

永岡　真

1962年生．1986年大阪府立大学大学院工学研究科航空工学専攻修士課程修了．同年（株）豊田中央研究所入社．1993～1994年米国George Mason大学客員研究員．現在，豊田中央研究所燃焼研究室研究員．エンジン内の流れ・燃焼の数値解析とモデリング，非構造格子による圧縮性流体計算法の研究に従事．1992年自動車技術会論文賞，1996年SAE-Arch T. Colwell Merit Award．機械学会会員．