プロダクション規則と局所評価関数による制約充足問題の解法*

金田　泰
新情報処理開発機構

廣川　真男
日立製作所基礎研究所

著者は自己組織的計算のための計算モデル CCM (化学的キャスティング・モデル) を提案している．CCM においては，“プログラム” は少数のプロダクション規則と，局所的情報だけを参照する評価関数 (局所秩序度) とで構成される．この報告では，このモデルにもとづいてある種の制約充足問題をとく方法をしめす．この方法では決定的かつ手続き的な制約伝搬にはよらずに確率的な手法によって，また非常に単純な “プログラム” によって N クウィーン問題や地図の彩色問題などの制約充足問題を平均すると多項式時間でとくことができる．この報告ではまた，この方法による計算の特性すなわち実行時間やその分布などを実測値にもとづいて論じる．

A Method of Solving Constraint Satisfaction Problems using Production Rules and Local Evaluation Functions†

Yasusi Kanada
Real-World Computing Partnership

Masao Hirokawa
Advanced Research Laboratory,
Hitachi Ltd.

The authors have proposed the Chemical Casting Model (CCM), which is a computation model for self-organizing computation. In this model, “programs” consist of a few production rules and evaluation functions (or local order degrees), both of which only refer to local information. A method of solving certain constraint-satisfaction problems, based on this model, is presented in this paper. This method enables to solve constraint-satisfaction problems, such as the N-queens problems or map coloring problems, in a polynomial order time, without using deterministic and procedural constraint propagation, but using a stochastic method, and by a very simple “program.” This paper also mentions to the characteristics of the computation by on this method, based on several measurements.

---

* この研究の一部は第 1 著者が日立製作所中央研究所においておこなったものである．
† Part of this research was done at Central Research Laboratory, Hitachi Ltd.

1. はじめに

金田 [Kan 92a, Kan 92b] は開放系 (環境に対してひらかれたシステム) における仕様も明確でない現実世界の問題をとくための自己組織的計算をめざして化学的キャスティング・モデル (Chemical Casting Model, CCM) という計算モデルを提案している[注1]．CCM の重要な特徴は，単純かつ汎用的な “プログラム” によって局所的な情報だけにもとづいて “大域的な秩序” をつくりだすような計算 (自己組織化) をおこなうという点にある[注2]．CCM には，自己組織的計算のために重要だとかんがえられる局所秩序度 (一種の評価関数) や非決定性 (確率的制御) などの機構がとりいれられている．またこのモデルによって，遺伝的アルゴリズムなどと同様に記号的計算とパタン的計算との中間領域の開拓をめざしている．

CCM は上記のように開放系における自己組織的計算をめざして開発されたモデルだが，現在のところは古典的な制約充足問題や最適化問題への適用をこころみている段階である．制約充足問題への適用においては，決定的かつ手続き的な制約伝搬によらず確率的な方法によって，非常に単純な “プログラム” によって平均すると多項式時間でいくつかの問題がとけることを，CCM にもとづく計算言語処理系 SOOC (Self-Organization-Oriented Computing) 上で確認している [Kan 92a, Kan 92b]．これまでにといた問題は N クウィーン問題や地図の彩色問題などである．

この報告では，ある種の制約充足問題を制約伝搬によらず CCM にもとづいてとくための方法をしめす．第 2 章では，準備のためにかんたんに CCM について説明する．第 3 章では制約充足問題の解法をしめすとともに，N クウィーン問題を例としてこの解法についてさらに説明する．

2. 計算モデル CCM

この章では，化学的キャスティング・モデルについて説明する．CCM は化学反応系とのアナロジにもとづく計算モデルであり [Kan 92a]，不完全な情報，あるいは非決定的な計画にもとづく計算のためのモデルである．

CCM の構成要素についてかんたんに説明する．CCM は Chemical Abstract Machine [Ber 90] と同様にプロダクション・システムにもとづくモデルである．プロダクション・システムにおける作業記憶 (短期記憶) は CCM においても作業記憶とよぶ．すなわち，CCM が作用するデータは作業記憶にふくまれる．そして，プロダクション・システムにおける規則ベースすなわちプログラムに相当するものをキャスタとよぶ．CCM は不完全な計画にもとづく計算のためのモデルなので，完全な計画を意味するプログラムということばのかわりにキャスタということばを使用する．いまのところ，キャスタはユーザによって記述され，そのままのかたちでつかわれる．

作業記憶にふくまれるべきオブジェクトあるいはデータとしては，つぎのようなものがある (図 1 参照)．原子はデータの単位であり，内部状態をもつ．原子にはデータ型があり，それを元素ともよぶ．原子どうしをリンクによって結合することができ，結合された全体を分子とよぶ．リンクは無向でも有向でもよい．リンクの存在は通常のプロダクション・システムにない CCM の特徴のひとつである．無向のリンクは化学結合に似ているが，化学結合には有向のリンクに相当するものはない．また，リンクにはラベルをつけることもできる．

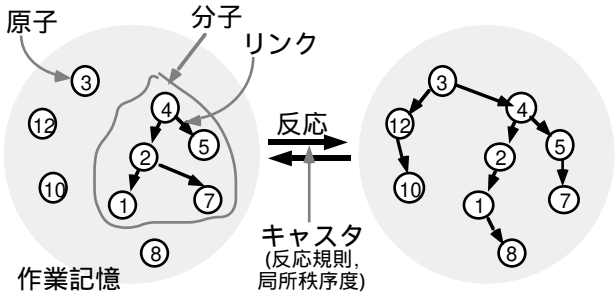


図 1　化学的キャスティング・モデルの構成要素

キャスタは反応規則と局所秩序度とで構成される．反応規則はシステムの局所的な (ミクロな) 変化のしかたをきめる規則であり，ユーザにより定義される．ここで「局所的」ということばは，その反応規則によって参照される原子数がすくないということを意味する[注3]．反応規則は前向き推論によるプロダクション規則として記述される．したがって，つぎのようなかたちをしている．

LHS → RHS.

反応規則は化学反応式に相当するものだといえる．後述する N クウィーン問題やグラフ彩色問題 [Kan 92b] などをはじめとするおおくの単純なシステムにおいては反応規則は 1 個だけ存在するが，複数の変

[注1] このモデルは金田 [Kan 92a] においては「化学的プログラミング・モデル」とよばれていた．

[注2] 局所的情報にもとづく計算をめざすひとつの理由は，いわゆる創発的計算 [For 91] を実現することである．

[注3] CCM においては，化学反応系のように (物理的な意味での) 距離の概念が導入されていないから，局所的ということばは距離がちかいということを意味しない．

化のしかたをみとめるより複雑なシステムにおいては複数個の反応規則が存在する．

局所秩序度は局所的な “組織化” あるいは “秩序化” の程度をあらわす一種の評価関数であり，作業記憶の局所的な状態が “のぞましい” ほどおおきな値をとるように，ユーザにより定義される．局所秩序度の存在は，通常のプロダクション・システムにくらべたときの CCM のもっともおおきな特徴である．局所秩序度はつぎの 2 つのうちのいずれかのかたちで定義される．

(1) 自己秩序度 *o*(*e*)
1 個の原子 *e* に対して定義される．

(2) 相互秩序度 *o*(*e1*, *e2*)
2 個の原子からなる対 <*e1*, *e2*> に対して定義される．

後述の *N* クウィーン問題のキャスタは相互秩序度を使用しているが，以下の説明においては，かんたんのため自己秩序度だけをかんがえる．自己秩序度は規則の適用時に原子ごとに計算されるが，その値は当該原子の内部状態だけでなく，そこからでるリンクがつながったさきの原子の状態にも依存しうる．

反応はつぎの 2 つの条件をみたすときにおこる．反応規則の左辺 LHS および右辺 RHS には原子とマッチする 1 個または複数個のパタンがあらわれるが，第 1 の条件は左辺にあらわれるすべてのパタンのそれぞれにマッチする原子が存在することである．

反応がおこるとこれらの原子は消滅して，そのかわりに右辺にあらわれる原子が生成される．ただし，左辺と右辺とに対応する原子があらわれるばあいは，その原子は生成・消滅するかわりにかきかえられる．このような規則とそれにあらわれる (左辺および右辺の) パタンにマッチするすべての原子との組をインスタンスとよぶ．ひとつのインスタンスがふくむ原子のうち，反応前に存在するものすなわち左辺にあらわれるものの局所秩序度の総和を “反応前のインスタンス秩序度”，反応後に存在するものすなわち右辺にあらわれるものの局所秩序度の総和を “反応後のインスタンス秩序度” とよぶ．反応後のインスタンス秩序度をあらかじめ計算したものが反応前のインスタンス秩序度よりおおきいとき，すなわち反応によって局所秩序度の和が増加する時だけ反応がおこるというのが第 2 の条件である．

そして，いずれかのインスタンスについて上記の 2 条件がみたされているかぎり，反応はくりかえしおこる．これらの条件をみたすインスタンスが存在しなくなると実行は中断される．

ただし，一般には上記の 2 つの条件をみたすインスタンスは複数個存在する．条件をみたすインスタンスが複数個生成される原因としては，ひとつの規則の条件部をみたす原子の組が複数個存在するばあいと，複数の規則についてその条件部をみたす原子の組が存在するばあいとがある．いずれのばあいでも，これらのインスタンスのうちのいずれがどのような順序で，あるいは並列に反応するかは非決定的であり，反応の順序はシステムが自発的にきめる．反応の順序によらず，のぞむ計算をおこなわせるはたらきをする (すべき) のが局所秩序度である．

上記のような自発性あるいは非決定性を CCM にあたえているひとつの理由は，非決定性のないアルゴリズミックな計算においては，プログラマがあたえた “よけいな制御” によってプログラムの動作が制約され，自己組織的な計算を不可能にしているばあいがあるとかんがえられるからである [Kan 92b]．

しかし，反応の順序をある程度はユーザが制御することができないと，のぞんだ計算を実現できないばあいがある．ユーザはスケジュリング戦略というものを指定することによってインスタンスの選択順序を制御し，反応の順序を部分的に制御することができる．スケジュリング戦略は，従来のプロダクション・システムにおける競合解消戦略に対応するものである．スケジュリング戦略にはインスタンスを系統的に選択する系統的戦略と，ランダムに選択するランダム戦略とがある．スケジュリング戦略について，くわしくは金田 [Kan 92a] を参照されたい．

3. 制約充足問題の解法

この章では，CCM によってある種の制約充足問題をとくための解法をしめす．3.1 節ではその概略手順をしめし，3.2 節では *N* クウィーン問題を例としてよりくわしい説明をする．

3.1 概略手順

CCM によってある種の制約充足問題をとくための概略手順をしめす．ただし，この手順はすべての制約充足問題に適用できるわけではない．また，この手順は発見的な要素をふくんでいるし，つくられたキャスタが十分高速に動作する保証はあたえられていないから，これにしたがえば問題がかならずとけるというわけではない．

(1) 反応規則の仮決定

制約充足問題をとく計算は，解探索の一種とみることができる．このようにみたとき，このステップでは探索空間を移動するための適当な操作 (1 個ま

たは複数個のオペレータ) すなわち反応規則となるべきものをみつける．ただし，これらの操作は局所秩序度の決定後に修正する必要があるかもしれないので，かならずしもまだ実際に記述する必要はない．

これらの操作は，くりかえし適用することによって探索空間をおおいつくせるようにきめる必要がある．ただし，逆にこれらの操作によってあきらかに解ではない多数の状態に到達できるようになっていると (すなわち探索空間が不要な状態を多数ふくんでいると) 探索効率が低下する．これらの操作によってプログラムの性能が左右されるので，慎重に選択する必要がある．

(2) データ構造の記述

(1) にもとづいてなにを原子として表現し，それらをどのようにリンクするか，あるいはリンクを使用しないかをきめる．そして，決定したデータ構造を (SOOC 言語で) 記述する．

(3) 局所秩序度の記述

原子または原子対が解にふくまれるための必要条件 C をもとめて，それにもとづいて局所秩序度を記述する．C は，解において原子がみたすべき制約だということもできる．すなわち，つぎのいずれかのようにする．

(i) 自己秩序度の記述

「解にふくまれるすべての原子 a について C(a) がなりたち，かつ探索空間内の任意の状態 S について作業記憶がふくむ任意の原子 a について C(a) がなりたてば S が解である」という命題がなりたつような条件 C(a) がみつかれば，

if C(a) **then** $o(a) = 1$ **else** $o(a) = 0$

のように自己秩序度 o を記述する (実際には SOOC によって記述する)．

(ii) 相互秩序度の記述

「解にふくまれることなる原子からなるすべての対 $<a1, a2>$ $(a1 \neq a2)$ について C($<a1, a2>$) がなりたち，かつ探索空間内の任意の状態 S について作業記憶がふくむ任意の原子の対 p について C(p) がなりたてば S が解である」という命題がなりたつような条件 C(p) がみつかれば，

if C($<a1, a2>$) **then** $o(a1, a2) = 1$ **else** $o(a1, a2) = 0$

のように相互秩序度 o を記述する．

上記の命題の意味については 3.2 節で説明する．ところで，局所秩序度o の値を上記では 0 または 1 としたが，1 のかわりにそれ以外の正の実数値をとるようにすれば，局所的な制約をその重要度によっておもみづけすることが可能である[注4]．また，o はある集合のメンバシップ関数とみなすことができる．この集合を So とすると $S \subset So$ がなりたつ[注5]．

なお，3 個以上の原子からなる組に対して局所秩序度を定義することも原理的には可能だが，2 個以下のばあいととくにかわらないので，それについては省略する．

(4) 反応規則の記述

(1) で仮決定した操作を反応規則として (SOOC によって) 記述する．ただし，それらの操作をそのまま反応規則とすると，反応がおこるべきばあいにも (3) で記述した局所秩序度から計算されるインスタンス秩序度が反応によって増加しない，したがって反応がおこらないばあいがある．このようなばあいには規則の両辺に適当なパタンを追加して，上記のようなばあいにインスタンス秩序度が増加するようにする．ただし，反応がおこるべきばあいとインスタンス秩序度が増加するばあいとがかならずしも一致していなくてもよい．

ところで，規則は原子にマッチするパタンをすくなくとも 1 個ふくんでいなければならないが，(3) において相互秩序度を記述するばあいにはこの規則は原子にマッチするパタンをすくなくとも 2 個ふくんでいなければならない．

(5) 初期状態の設定

探索空間内の適当な状態を初期状態として設定する (設定するためのプログラムを記述する[注6])．

(6) 計算

上記の初期状態のもとで，上記の反応規則と局所秩序度とを SOOC 処理系にあたえて動作させる．

■

上記の方法は制約充足問題を最適化問題に変換し

[注4] 後述する N クウィーン問題のようにすべての局所的な制約を同時にみたすことが容易なばあいは，これらにおもみづけすることはあまり意味がないとかんがえられるが，LSI の配線のように制約をみたせない部分がのこっても (すべての線を自動配線できなくても) 意味があるばあいには，おもみづけに意味があるとかんがえられる．

[注5] ここで o がとりうる値として 0, 1 という離散値ではなく 0 から 1 までの連続値をとるようにすれば，一種のファジィ制約充足 (?) が実現されるが，それはこの報告の主題からはずれるのでこれ以上のべない．

[注6] 現在の SOOC 言語は初期状態の記述のための十分な機能をもっていない．そのため，SOOC 処理系のベースになっている Lisp によって初期設定のためのプログラムを記述する．

てといているということができる．すなわち，上記の計算においては局所秩序度の作業記憶全体にわたる和である大域秩序度 (より正確な定義は金田 [Kan 92b] 参照) を最大化 (または極大化) するように処理系が動作する．そして，大域秩序度すなわち最適化関数が最大となる点が制約充足問題の解である[注7]．

ところが，上記の方法においては解に到達しないまま停止したり，いつまでたっても停止しなかったりするばあいがある．解に到達せずに停止するのは大域秩序度が局所最大点にとらわれ，かつ反応によりインスタンス秩序度が増加するようなインスタンスが存在しないばあいである[注8]．これらの現象がおこるばあいには，適当な処置をとる．

もっともかんたんな処置は同一のキャスタ，初期状態のもとで再実行させることである．非系統的なスケジュリング戦略を使用しているばあい (たとえば乱数によって原子や規則を選択しているばあい) には，単純に再実行するだけで解がもとめられるばあいがある．それではだめならば，初期状態を変更してから再実行させる．それでも解決しないばあいには，キャスタの一部または局所秩序度を変更する必要がある．

CCM にもとづく計算は確率的なので，停止性は確率的にしか保証できないであろう (どうすれば確率的に保証することができるかも，まだわかっていない)．しかし，問題によっては正当性すなわち停止したときにただしい解がえられることがしめされる．たとえば，次節でしめす N クウィーン問題においては正当性を証明することができる．

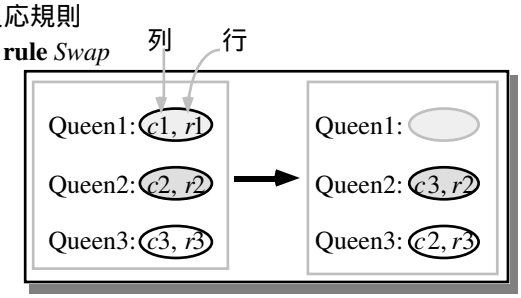
3.2 例題 : N クウィーン問題

N クウィーン問題は，N 行 N 列の “チェス・ボード” に，たがいにとりあわないように N 個のクウィーンを配置する制約充足問題である．以下，前節の手順にしたがって N クウィーン問題のひとつの解をもとめるキャスタをつくっていくが，つくられるキャスタをあらかじめ図 2 にしめす．このキャスタじたいについては金田 [Kan 92a] がくわしく説明しているので，この報告では最低限の説明だけをする．

(1) 反応規則の仮決定

図 3 にしめすように 2 個のクウィーンの列を交換する操作をくりかえすことによって解をもとめるという方針をたてる．初期状態において各行各列にただひとつのクウィーンが存在するように配置すれば，規則を何回適用してもこの条件がくずれることはない．また，この操作 (すなわち互換) をくりかえすことによって任意の置換を実現することができるから，適当なクウィーンを選択してこの操作をくりかえすことによって，上記の条件をみたすすべてのクウィーン配置を生成することができる．すなわち，探索すべきすべての状態が，上記の操作によってさだめられる探索空間にちょうどふくまれている．

■ 反応規則

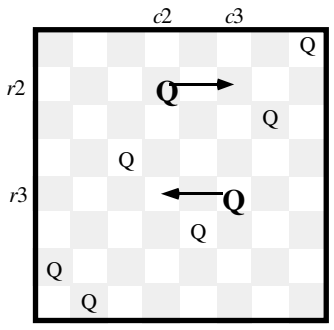


■ 局所秩序度 (相互秩序度)

2 個のクウィーンのあいだで秩序度を定義する。

$$o(x, y) = 0 \quad \textbf{if } x.column - y.column = x.row - y.row \textbf{ or } x.column - y.column = y.row - x.row,$$
$$1 \quad \textbf{otherwise}.$$

図 2 CCM による N クウィーン問題のキャスタ



2 個のクウィーンをえらんでその欄を交換するという操作をくりかえすことによって解に到達することをめざす。

図 3 N クウィーン問題のプログラムの規則の意味

ここで，仮に上記の規則のかわりに 3 個のクウィーンの列を回転させる ($c1 \leftarrow c2 \leftarrow c3 \leftarrow c1$) 規則をつかうことをかんがえる．すると，この置換は偶置換であるから，それをくみあわせても 2 個のクウィーンの列交換のような奇置換を生成することはできず，探索空間は探索すべきすべての状態をふくんでいない．したがって，このような規則は適当でない (ただし，このばあいはこの探索空間にも解が存在

[注7] なお，CCM において大域秩序度 o は一般には任意の実数値をとりうる．しかし制約充足問題においては，後述する N クウィーン問題におけるように，局所秩序度が定義されていれば $0 \le o \le M$ (M は作業記憶にふくまれる原子 (対) の個数) の範囲の整数値だけをとる．

[注8] 局所最大点にとらわれるばあいがあるという点では，CCM は遺伝的アルゴリズムにくらべてよわい．しかし，大域秩序度の局所最大点にとらわれずに解がもとめられる問題もある N クウィーン問題のばあいがそうである [Kan 92a]．

するため，この規則をつかっても解をもとめることができる)．

また，仮に上記の規則のかわりに 1 個のクウィーンの座標を任意にかえる規則をつかうことをかんがえると，この規則によってきめられる探索空間はクウィーンが各行各列にただ 1 個存在するという制約をみたさない状態を多数ふくむから，探索効率が低下する．

なお，もし最初に記述した規則にもとづいて局所秩序度の記述や計算をするとうまくいかないことがわかれば，このステップにもどって反応規則をかきかえる．

(2) データ構造の記述

クウィーンを原子として表現することにきめる．各クウィーンはその行および列を内部状態 (整数値) としてもつことにする．これらをべつの原子として表現してクウィーンとそれらとのあいだにリンクをはることもできるが，ここではそうしない．このようにきめると，(1) できめた操作は原子の内部状態のかきかえによって実現されることになる[注9]．

SOOC-93[注10] によるデータ構造の記述はつぎのようになる (ここで，defelement マクロの構文は Common Lisp の defstruct マクロにちかい)．

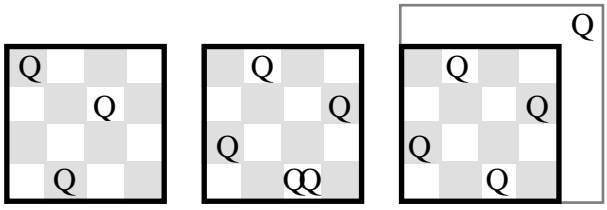
```
(defelement queen      ; 元素 queen (queen 型) の宣言
    row
    col)
```

(3) 局所秩序度の記述

このステップでは，2 個のクウィーンの関係として，すなわち相互秩序度として局所秩序度を定義する．自己秩序度ではなくて相互秩序度によって定義するのは，各クウィーンはリンクでつながれておらず，単独ではクウィーンが解にふくまれるべき状態かどうかを判定することができないからである．

2 個のクウィーンが対角線方向にないことがそれらが解にふくまれるための必要条件であるから，この条件を C とする．前記の初期状態のもとでは，盤面 (作業記憶) にふくまれる任意のクウィーンの対について C がなりたてばその状態は解であるから，C は (3)(ii) でのべた命題をみたしている．したがって，これを局所秩序度として記述する．

ここで，(3)(ii) の命題の意味について説明しておく．命題の前半は，制約 C がすべての原子対についてなりたつべきであることをいっている[注11]．後半は，システムが停止したときすなわち反応がおこりうるインスタンスがなくなったとき解に到達していること，すなわち正当性を保証している．この条件がないと，たとえば N クウィーン問題においては図 4 にしめすようにクウィーンが N 個未満しか配置されていない状態や盤面上の同一の位置に 2 個以上のクウィーンが配置されている状態など (が探索空間にふくまれているとすると，そこ) でシステムが不正に停止する可能性がある．なお，(3)(i) の命題に関してもほぼおなじことがいえるので，その説明は省略する．



クウィーン数がすくなすぎる　同位置に複数のクウィーンがある　盤面をはみだすクウィーンがある

図 4　(3)(ii) の命題後半をみたさない例
(4 クウィーン問題の例)

SOOC-93 による局所秩序度の記述はつぎのようになる (ここで，deforder マクロの構文は CLOS の defmethod マクロにちかい)．

```
(deforder ((x queen) (y queen))
    ; queen 型の原子 x, y 間の相互秩序度の宣言
    (if (or (eql (– (queen-col x) (queen-col y))
                 (– (queen-row x) (queen-row y)))
            (eql (– (queen-col x) (queen-col y))
                 (– (queen-row y) (queen-row x))))
        0
        1))
```

(4) 反応規則の記述

(1) で仮決定した規則にあらわれる 2 個のクウィーンのあいだの局所秩序度は，反応させても変化しない．したがって，インスタンス秩序度も変化しない．したがって，このままではうまく動作しない．そこで，第 3 のクウィーンを規則の両辺にくわえる．このクウィーンは反応によって変化しないが，インスタンス秩序度の計算にだけ参加する．このようなパタンを触媒とよぶ．触媒をくわえることにより，第 3 のクウィーンともともと規則にふくまれていた

[注9] リンクで表現すれば，(1) できめた操作はリンクのかきかえによって実現されることになる．

[注10] 現在使用可能な SOOC 処理系である SOOC-92 とは一部構文がことなっている．

[注11] なお，一般には C(p) は原子対 p が解にふくまれるための必要十分条件とすることはできない．なぜなら，C(p) をみたすような原子対 p は，解にふくまれるもの以外にも存在するからである．たとえば，N クウィーン問題において対角線方向にないクウィーンの対が解にふくまれるもの以外にも多数存在しうることはあきらかであろう．

クウィーンとの関係は反応によって変化するようになるので，うまく動作する可能性がでてくる．すなわち，インスタンス秩序度が増加するばあいにかならずしも大域秩序度が増加するとはかぎらないが，もし平均的には大域秩序度が増加するならば，この規則をつかって確率的に解をもとめるこができるとかんがえられる．そこで，とりあえずこのような規則を記述する．

もしこれでうまく動作しない (計算時間がかかりすぎる) と仮定すると，第 4，第 5 のクウィーンを追加することによって，うまく動作するようにできるかもしれない．そうやって反応にあたってより非局所的な範囲まで参照するようにすれば，局所最大点におちいる確率はたかくなるが，計算時間は短縮させることができる (あるいは，事実上解をもとめられなかったものが，もとめられるようになる)．ただし N クウィーン問題のばあいには，実際には触媒は 1 個だけで十分である[注12]．

SOOC-93 による反応規則の記述はつぎのようになる．

```
(defrule swap-queen
    (var C1 R1 C2 R2 C3 R3)      ; 変数宣言
    (reaction
      (exist queen Q1 (:col C1) (:row R1))
      (exist queen Q2 (:col C2) (:row R2))
      (exist queen Q3)           ; ここまでが左辺
      <->                        ; (この規則は可逆)
      (exist queen Q1 (:col C2) (:row R1))
      (exist queen Q2 (:col C1) (:row R2))
      (exist queen Q3)))         ; ここまでが右辺
```

(5) 初期状態の設定

(1) ですでに決定したように，各行各列にただひとつのクウィーンが存在するような配置を初期状態とする．もちろん対角線方向には 2 個以上のクウィーンがあってもよい．このような条件をみたす配置を乱数などをつかって生成してもよいが，もっともかんたんな初期状態はすべてのクウィーンをひとつの対角線上に配置した状態である．

(6) 計算

上記の反応規則と局所秩序度とを SOOC 処理系[注13]にあたえて動作させると，(ランダム戦略のばあい) そこそこの計算時間でほとんど 100 % 解をもとめることができる．(4) でのべた第 4 のクウィーンを追加すると解がもとめられるまでの反応回数は減少するが，規則条件部のマッチングにかかる時間が増加するため計算時間は増大する．したがって，第 4 のクウィーンを追加する必要はないとかんがえられる．

ところで，この計算によって解がうまくもとめられるばあいには，各原子または原子対の局所秩序度は図 5 にしめすように変化する．はじめは局所秩序度が 0 の原子 (対) と 1 の原子 (対) とがまざりあっている．反応によって 0 から 1 になるものもあり，1 から 0 になるものもある．しかし，反応がくりかえされると，やがてすべてが 1 となって停止する[注14]．エイト・クウィーン問題のばあいにはつねに 28 (= ${}_8C_2$) 個の原子対が存在し，したがって大域秩序度の最大値すなわち解における大域秩序度は 28 である．■

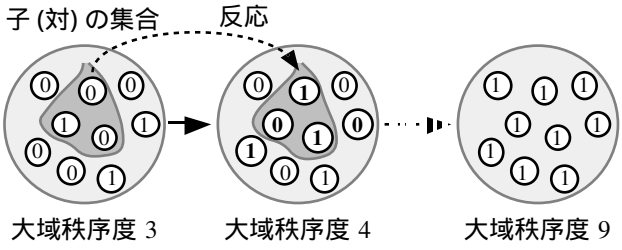


図 5　局所秩序度と大域秩序度の変化の例

最後に，金田 [Kan 92a] がしめさなかった N クウィーン問題のキャスタの実行特性の測定結果をいくつかしめす．

図 6 はランダム戦略を使用して計算したばあいの，解がもとめられるまでの反応回数 (Actions) とマッチング回数 (Tests) すなわち規則左辺の実行回数の測定例である．計算時間はしめしていないが，それはマッチング回数にほぼ比例する．測定は SUN4 の KCL 上の SOOC-92 処理系によっておこない，5 回の測定の平均値をしめしている．平均値を使用したのは，ばらつきをすくなくするためである．点線でしめしているのは N^4 のかたむきをしめす直線である．マッチング回数は $O(N^{4.6})$ である．

また，図 7 は対角線配置を初期状態とする計算における大域秩序度の変化を例示している．大域秩序度が増減をくりかえしながら，最後には解すなわち大域秩序度が 28 の状態に達していることがわかる．このふるまいを説明するための大域秩序度の変

[注12] たとえばランダム戦略のもとで反応回数と計算時間とをくらべると，触媒数が 1 個で 68 回と 1.3 秒，2 個で 42 回と 1.8 秒，3 個で 23 回と 2.9 秒となった．ただしこの測定では，しらべたインスタンスが 100 回つづけて反応しないときに停止させ，それまでの反応回数と計算時間の 20 回の平均値をとった．条件をかえると触媒数が 2 のほうが計算時間がみじかいばあいもあるが，大差はない．

[注13] 前ページの脚注を参照．

[注14] 図 5 は CCM にもとづく制約充足と物理学における相転移問題との関係を示唆している．

化の確率モデルおよびエイト・クウィーン問題の求解過程に関するより詳細な分析に関しては金田 [Kan 93, Kan 92b] を参照されたい．

図 8 には，*N* クウィーン問題において解がもとめられるまでの反応回数の頻度分布の例をしめす．反応回数の測定は 200 回おこなった．実線でしめしたのが確率モデルからの予測値であり，ほぼ指数関数になっている．

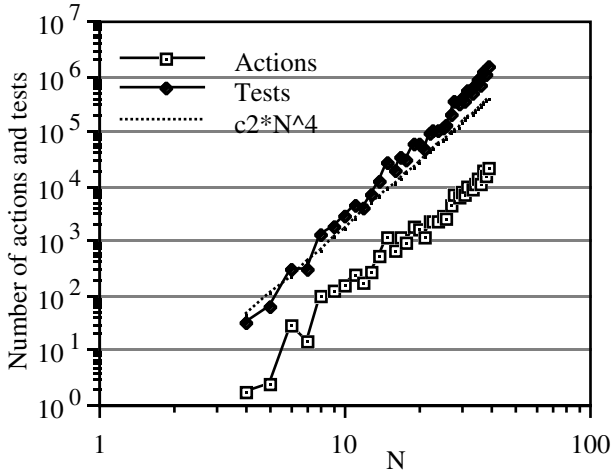


図 6　*N* クウィーン問題における反応回数とマッチング回数の例

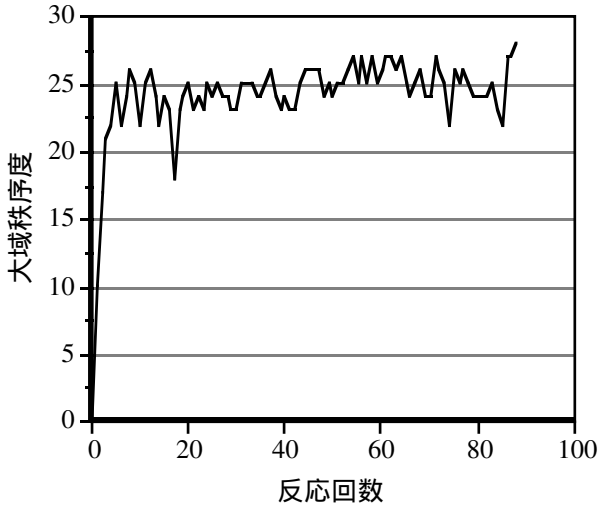


図 7　*N* クウィーン問題における大域秩序度変化の実測値の例

4. 結論

この報告では，ある種の制約充足問題を CCM にもとづいてとくための方法をしめした．この方法では制約伝搬によらず確率的な方法によって，また非常に単純な “プログラム” によってある種の問題を多項式時間でとくことができる．

この方法はまだ一部の制約充足問題に適用できるだけであり，また発見的な部分があるために機械的に実行できるわけでもない．しかし，今後，よりひろい範囲の制約充足問題に適用できるように改善したり，問題の仕様じたいが時間とともに変化するような，従来の手法では解決しにくい問題に適用範囲をひろげることによって，現実世界の問題に適用できるようになるとかんがえられる．

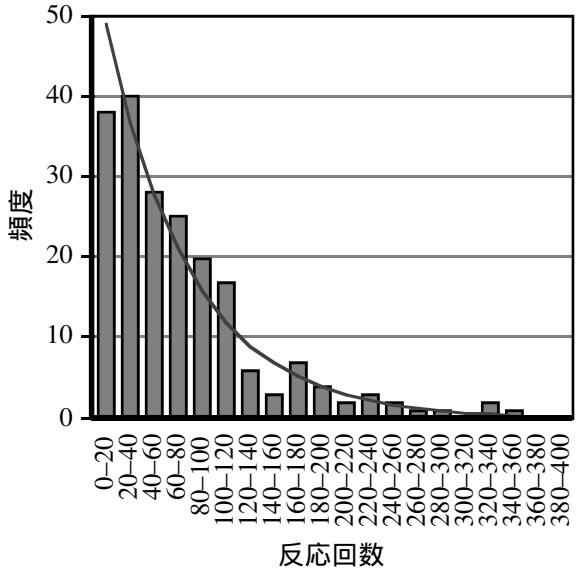


図 8　*N* クウィーン問題における解がもとめられるまでの反応回数の頻度分布

謝辞

この研究をはじめるきっかけをつくっていただいた日立製作所日立研究所の 坂東 忠秋 部長，研究の継続をゆるしていただいている新情報処理開発機構の 岡 隆一 部長，および第 33 回プログラミング・シンポジウムやほかの機会に議論にくわわっていただいた，おおくの方々に感謝する．

参考文献

[Ber 90]　Berry, G., and Boudol, G.: The Chemical Abstract Machine, *Proc. 17th Annual ACM Symposium on Principles of Programming Languages*, pp. 81–94, 1990.

[Kan 92a] 金田　泰: コンピュータによる自己組織系のモデルをめざして, *第 33 回プログラミング・シンポジウム報告集*, 1992.

[Kan 92b] 金田　泰: 自己組織系としての計算システム — ソフトウェア研究への 2 つの提案 —, *夏のプログラミング・シンポジウム報告集*, 1992.

[Kan 93]　金田　泰: 確率過程としての計算 — 計算過程のマクロ・モデルの必要性とその例 —, *情報処理学会プログラミング研究会*, 1993.3.

[For 91]　Forrest, Stephanie, ed.: *Emergent Computation*, MIT Press, 1991.