

制約論理プログラミング言語による  
定性シミュレーション

6C-5

○ 白水 隆司 大和田 勇人 溝口 文雄  
(東京理科大学 理工学部)

## 1. はじめに

定性シミュレーション法は、対象系に関する定性的な制約や情報をもとにシミュレーションを行う有効な方法である。しかし一方では、定量的な情報や2つの系を比較して解析する比較解析、さらには時間の遅れなどを考慮する定性微分解析[1]を導入しようとした場合、その実現が難しくなる。

そこで本稿では、定性シミュレーションシステムを制約論理プログラミング言語の枠組みを用いて行うことによって、定性微分解析[1]や定量的な処理を容易にシミュレーションシステムに導入することができることを示す。これは、制約論理プログラミング言語を用いることによって、比較解析や定性微分解析に用いる時間の管理と量の管理、さらに制約やパラメータの定量的情報や定性的情報に関する処理を宣言的に記述することができるからである。そして、これは制約論理プログラミング言語であるCS-Prolog[4]の論理変数と遅延実行メカニズムの利用によって実現している。

## 2. 制約論理プログラミング言語

制約論理プログラミング言語は人工知能システムにおける新しいパラダイムを実行可能にする言語であり、近年その処理能力が注目されている。そのなかでも制約論理プログラミング言語CS-Prolog[4]はPrologの単一化を拡張したプログラミング言語であり、線形結合の方程式および不等式が解けることからその記述性が高いと言える。

図1は、同言語処理系の実行例である。Prologにおける論理変数によって直接的に数式を宣言的に表現することができる。

```
?- csp X>=0, Y>=0, Z>=0, X+Y=Z, X*Y>0.
1433msec.
X = _576
Y = _618
Z = _576+_618
```

図1 CS-Prologの実行例

このように記述性の高いCS-Prologを用いれば、定性的な情報に定量的情報を融合した処理や定性微分解析を容易に実現することができる。以下にその実現方法について述べる。

## 3. 定量的情報の処理

制約論理プログラミング言語上での数式処理は図1のようにPrologの論理変数によって自然に表現できるので、従来の定性推論におけるパラメータXにおける変化傾向inc, std, decをそれぞれDX>0, DX=0, DX<0として記述すればよい。

また、従来のMULやADDなど乗算や加算の制約処理は、図2のように定性値の場合の制約条件を記述することによって行っていた。しかし、CS-Prologを用いれば図3のように制約を直接的に、また宣言的に書くことが

できる。そのためADDの各引数に定量値が単一化されれば、ADDやMULの制約は定量的に処理される。図3は加算ADDの制約に関する制約記述である。これによって定性情報および定量情報の両方を扱うことができる。

```
add([], [], [], _, _, _) :- !.
add(X, Y, Z, Hx, Hy, Hz) :-
  X = [qs(_, _, _, Vx, Dx) | Xs],
  Y = [qs(_, _, _, Vy, Dy) | Ys],
  Z = [qs(_, _, _, Vz, Dz) | Zs],
  add_direction(Dx, Dy, Dz),
  add_consistency(Vx, Vy, Vz, Hx, Hy, Hz),
  add_cs_value(Vx, Vy, Vz, Hx, Hy, Hz, Cx, Cy, Cz),
  add(Xs?, Ys?, Zs?, Cx, Cy, Cz).
```

図2 従来の加算の制約記述(抜粋)

```
add([], [], []) :- !.
add(X, Y, Z) :-
  X = [qs(_, _, _, Vx, Dx) | Xs],
  Y = [qs(_, _, _, Vy, Dy) | Ys],
  Z = [qs(_, _, _, Vz, Dz) | Zs],
  Dx+Dy=Dz, Vx+Vy = Vz, (←直接的に記述できる)
  add(Xs?, Ys?, Zs?).
```

図3 制約言語による加算の制約記述

## 4. 比較解析と定性微分解析の導入

定性微分解析[1]は、モデルが異なる系に関する挙動を比較する比較解析と、定性的な微係数を導入することによって、系の振舞いにどのように影響を与えるかを予測する方法である。

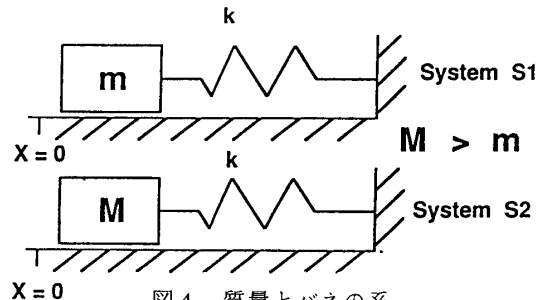


図4 質量とバネの系

従来のQSIM[5]を基本とするシミュレーション法では、図4のような質量とバネの系における第1の系S1(質量がmの系)と第2の系S2(質量がMの系:  $M > m$ )のシミュレーションをそれぞれ行っていた。しかし、図1のS1とS2をそれぞれ定性的にシミュレーションしても、すべての定性値(値の符号)と変化傾向の遷移組は等しくなる。

このようにS1とS2を個別にシミュレーションする方法では、次のような質問に答えることができない。  
① 図4の系のおもりの質量が増加したら、位置  $X = 0$  を通過するときの速度はどうなるか?

②図4の質量が増加したら、位置  $X = 0$  に到達する時間はどうなるか？

QSIM[5]を基本とする定性シミュレーション法においては、対象とする系の正常値と初期値を与えることによって、その系の平衡状態からどのように振舞うかに関するシミュレーションが行える。しかし①のように、系の平衡状態自体が異なる場合は従来の枠組みでは処理することができない。

また②においては、到達する時区間についての処理が必要である。しかし、従来の定性シミュレーション法の場合は時区間を瞬間と区間の交互の遷移によって実現しているため、S1の時区間とS2の時区間を比較しなければ②の問いには答えられない。

ここで、定性微分解析[1]を定性シミュレーション法に導入する。定性微分解析に関する規則は数多くあるが、ここではCS-Prologを用いると遅れ規則がどのように実現されるかについて示す。

[遅れ規則]

$\text{deriv}(X, V, T)$  という関係 (位置  $X$  の  $T$  に関する微分が  $V$  であるという意味) において、S1のパラメータ  $X$  の状態遷移が  $X(T(\tau i))$  から  $X(T(\tau i+1))$  に、S2におけるパラメータ  $X$  の状態遷移が  $X'(T(\tau i'))$  から  $X'(T(\tau i+1'))$  に変化するとき、もし  $V(T(\tau i')) > V(T(\tau i)) > 0$ 、かつ  $X(\tau i) = X(\tau i')$ 、 $X(\tau i+1) = X(\tau i+1')$  ならば、 $T(\tau i+1') - T(\tau i') < T(\tau i+1) - T(\tau i)$  である。

```
qs([], [], [], [], [], [], []) :- !.
qs(Time1, Time2, P, q(L0), q(L0), D1, D2, LS, S0) :-
    Time1 = [i(T0, T1) | TL1],
    Time2 = [i(T2, T3) | TL2],
    D1 > D2, D1 > 0, D2 > 0,
    TD1 = T1 - T0, TD2 = T3 - T2, TD1 > TD2,
    upper_landmark(L0, L1, LS),
    NT1 = i(T0, T1), NT2 = i(T2, T3),
    NV1 = i(L0, L1), NV2 = i(L0, L1),
    observation1(NT1, P, NV1, ND1), ND1 > 0,
    observation2(NT2, P, NV2, ND2), ND2 > 0,
    S0 = [qs2(NT1, NT2, P, NV1, NV2, ND1, ND2) | S1],
    qs(TL1?, TL2?, P, NV1, NV2, D11, D22, LS, S1).
```

図5 CS-Prologによる遅れ規則の記述

この「遅れ規則」を制約論理プログラミング言語CS-Prologによって記述したのが図5である。図中のS0の中の述語qs2は各パラメータの属性であり、第1引数と第2引数には、それぞれ第1の系S1と第2の系S2の定性的時間が格納され、内部的には  $i(T0, T1)$ 、または  $i(T0)$  という形のリストで表現されている。  $i(T0, T1)$  とは時刻  $T0$  と  $T1$  の間の時刻であり、  $i(T0)$  は時刻  $T0$  を示す。また第4引数から第7引数は、当該パラメータの値とその変化傾向の値を示す。これらのパラメータの値は定量値；あるいはランドマーク値に対応する論理変数のいずれかである。そのため、第3節で示したように定量的な値がその論理変数と単一化が行われていればその値で計算される。

## 5. 本方法によるシミュレーション

本方法の例題として、質量とバネの系を対象として述べる。質量とバネの系は第2節で示した図4のような2つの系である。このとき、質量  $M$  が増加した場合に位置  $X = 0$  を通過する時間はどうなるかという問いに対して、図6のようなモデルと問いをCS-Prologのゴールとして呼び出すことによって、定性的な遷移とそのときの時区間が求められる。

図6はバネに関する定量的な値を処理する場合のゴールを示している。T1とT2は瞬間と区間によって処理されるが、図5のような遅れ規則などによって2つの時区間は比較される。図中の?はfreezeによる遅延実行メカニズムが用いられていることを表している。述語hold1, hold2は当該パラメータの時間と量に関する制約であり、inc, dec, stdなどの制約を記述している。

```
spring(Pm) :-
    Pm = [X, V, A, M, K, F1, F2],
    deriv(X?, V?, [T1?, T2?]),
    deriv(V?, A?, [T1?, T2?]),
    mul(M?, A?, F2?, [M0, M1], [A0, A0], [NF2, NF2]),
    mul(K?, X?, F1?, [K0, K0], [X0, X0], [NF1, NF1]),
    mon_decrease(F1?, F2?, [NF1, NF1], [NF2, NF2]),
    par(T1?, T2?, x, q(X0), D1, q(X0), D2, [0, X0], X),
    par(T1?, T2?, m, q(M0), 0, q(M1), 0, [0, M0, M1], M),
    par(T1?, T2?, f2, q(NF2), _, q(NF2), _, [NF2, 0], F2),
    D1 < 0, D2 < 0, D3 < 0, D4 < 0, D11 > 0, D12 > 0,
    simulate(T1, T2, Pm).
```

```
hold1(_, x, _, _) :- !.
hold1(_, a, _, _) :- !.
hold1(_, m, _, 0) :- !. % m is std

hold2(_, x, _, _) :- !.
hold2(_, a, _, _) :- !.
hold2(_, m, _, 0) :- !. % m is std
```

図6 質量とバネの系のゴール例 (抜粋)

これによって、?- csp spring(X).のゴールを呼び出すことによって、定性微分解析と定量的な情報の処理が制約ベースで行なわれ、対象系の挙動を得ることができる。

## 6. まとめ

定性微分解析と定量性を導入したシミュレーションを制約論理プログラミング言語の枠組みを用いることによって容易に実現する方法を示した。これは、制約論理プログラミング言語における論理変数と遅延実行によって、定量的な情報の処理と定性的な情報の処理、定性微分解析をシミュレーションと統一的に行なえるからである。定性シミュレーションにおける定性微分解析と定量性の導入が、従来のQSIM[5]を基本とする定性シミュレーション法を拡張することによって実現できる。

今後の課題としてこれらの枠組みによる大規模工学システム[2]による例、モデル化[3]との融合、偏微分の導入、制約処理系の高速化などがある。

## 7. 参考文献

- [1] Daniel S. Weld : Comparative Analysis, Artif. Intell., Vol. 36, pp. 333-373 (1988).
- [2] 白水, 溝口 : 定性推論における構造モデル修正法, 情報処理学会第38回全国大会予稿集, pp. 450-452 (1989).
- [3] 溝口, 白水, 大和田 : 定性推論のためのモデル化支援システムの開発, 人工知能学会第2回全国大会論文集, pp. 65-68 (1988).
- [4] 川村, 大和田, 溝口 : CS-Prolog: 拡張単一化に基づくCONSTRAINT SOLVER, Proc. of LPC'87, pp. 21-28 (1987).
- [5] Kuipers, B. J.. Qualitative Simulation, Artif. Intell., Vol. 29, pp. 289-338 (1986).