人　工　知　能 92－5
（1994．1．19）

時間知識ベースにおける不確定時点の扱いと階層化による効率化

呉　楽水　　石塚　満
東京大学

あらまし　時間推論の主要な課題は時間を含む知識の表現と時間整合性維持の手法である．ＡＩでは時間知識を時間制約として表現し、整合性維持をTCSP（Temporal Constraint Satisfaction Problem）として扱う研究が最近進展している．本稿ではTCSPの表現力の不十分性を解決するために不確定時点の抽象化について考案し，不確定時点の抽象化を利用して複数な事象をグループ化して扱う方法を示す．更に，グループ化による時間知識ベースの階層化による効率的推論法を提案する．

Handling Unfixed Temporal Points in Temporal Knowledge-base and Efficienct Inference based on Hierarchy

WU LeShui　　Mitsuru ISHIZUKA
University of Tokyo

Abstract　The main issues of temporal reasoning are temporal knowledge representation and temporal consistency maintenance. Recently, researches on constraint-based representations for temporal knowledge and temporal CSP(TCSP) for temporal consistency maintenance have been progressing. In this paper, we conside the abstraction of unfixed temporal point to enhance the representation capability of TCSP, and show a way to handle several points as a group. Furthermore we propose an efficient inference method based on the hierarchy of temporal knowledge-base by grouping.

1 まえがき

時間を含む知識の表現と取扱いは，プランニング、スケジューリング、談話理解、予測診断など時間概念を含む推論問題に不可欠である．時間推論の主要課題は時間知識 の表現と時間整合性の維持手法である．時間知識の表現，及び高速推論法については，種々の研究が行われている．主な研究にはAllen's interval algebra, Vilian & Kautz's point algebra, Dean & Mcderm-ott's Time Map Managemaent(TMM), Dechter Meiri & Pearl's Temporal Constraint Problem などがある[6].

その中で，ここでは時間知識を制約として表現され、整合性維持をTCSP（Temporal CSP）という時間制約充足問題とする研究に注目する．制約に基づく時間推論の研究には主として次の二つのモデルがある：時点をベースとするDetcherらの時間制約メトリックネットワーク[2]，及び時区間をベースとするAllenの時区間論理[1]である．時区間制約は時点制約に変換することができるので，定量的な時間情報を取り扱える時点ベースTCSPは最も基本的なモデルといえる（以下明言しない限り，TCSPを時点ベースTCSPの略称とする）．しかし，単純な時点ベースTCSPでは表現力に不十分な点がみられる．このため，時点、時区間を基本変数として多種の制約を扱うインタフェースを用いて問題を記述し，テーブル変換によって内部で時点制約ネットワークを用いて問題を解決する研究（統合モデル）[3][4]は最近の一つの流れとなっている．

時点ベースTCSPでは，制約を時点間の線形不等式によって表す．制約充足によってイベントの可能な発生時間帯、イベント間の時間関係、シナリオの真実性などを推論できる．その解は全ての時点変数間の最小制約からなる最小ネットワーク（Minimal Network）で表される．

ただし、現実世界の時間概念はかなり複雑かつ曖昧であるため、TCSPモデルで複雑な時間知識記述からいかに時点間制約に変換するのかの方法についてまだ幾つかの問題が存在する．時点ベースTCSPでは確定した二つの時点間の制約は簡単に処理できるが，前後関係をはっきり言えない幾つかの時点グループの始点・終点にかかる制約については，制約表現と制約充足はできない．ここではこのような不確定時点に関する制約を確定時点間の制約から求める方法を検討し，その抽象化時点の表現と算出操作について述べる．

更に，TCSPが膨大な計算量とメモリ空間を必要とする問題点に対して，グループ化による時間知識ベースの階層化構築法を提案する．グループ化は不確定時点の抽象化操作によって算出される．階層化を行うことにより整合性維持はグループ間の制約伝搬によって実現される．最後に，階層化による効率性を評価する．

2 TCSPモデル

TCSPモデルでは，時点を基本変数とし，制約を線形不等式によって表す．

時点変数　Xi, Xj

時区間　I = {V1,...,Vn}

= {[a1,b1],...,[an,bn]}

とすると，

Unary Constraint（ある時点変数に関する制約）は

Ti = I : $(a1 \le Xi \le b1) \vee \ldots \vee (an \le Xi \le bn)$ と、

Binary Constraint（二つ時点変数間の制約）は

Tij = I : $(a1 \le Xj\text{-}Xi \le b1) \vee \ldots \vee (an \le Xj\text{-}Xi \le bn)$ と表すことができる．

もし T0 = 0 を導入すると，Ti = T0i ，つまりUnary制約はBinary制約によって統合的に扱える．

時点変数をノードとし，変数間の制約を辺とする有向グラフをTCSPの制約ネットワークグラフという（図1）．

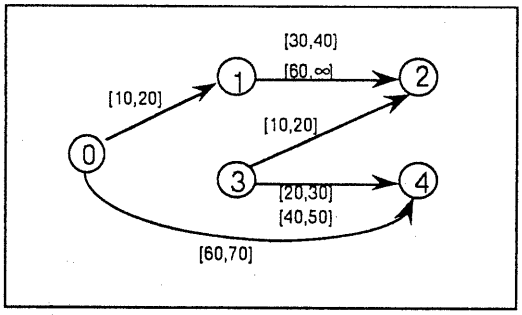

Fig.1 TCSPの制約ネットワークグラフ

TCSPでは制約充足に二つの操作が用意されている．TとSを二つのバイナリ制約とすると，

intersection S ⊕ T はSとTの最小（共有）部分を求める．すなわち

[a,b] ⊕ [c,d] = [Max(a,c), Min(b,d)] となる．

composition S ⊗ T はSとTの制約合成（伝搬）を求める．すなわち

[a,b] ⊗ [c,d] = [a+c, b+d] である．

このような操作を用いて，全ての変数間の最小制約をこの二つの変数間にある全ての経路上の制約伝搬の最小化操作から求められる．つまり

Tij = ⊕∀k (Tik ⊗ Tkj)

として求められる．

TCSPの全ての制約を充足できる時点変数の値の組 X={X1=x1, ..., Xn=xn} はTCSPの一つの解(solution)という．解があるTCSPは整合性がある(consistent)という．Xi = w の時にTCSPに解があれば，w は Xi の実行可能値 (feasible value)という．ある変数に関する全ての実行可能値からなる集合をこの変数の最小領域(minimal domain)とする．全ての変数の最小領域と変数間の最小制約からなるネットワークは，TCSPの最小ネットワーク(minimal network)である．

・ STCSP（Simple TCSP）

辺上にラベルが一個しかないTCSPをSTCSPという．この場合，

制約 Tij = Xj − Xi = [a, b] を

Xj − Xi ≤ b ,

Xi − Xj ≤ − a

と書き直すことができる．もし，b を i->j への距離、− a を j->i への距離とすると，STCSPのグラフは距離グラフに帰着できる（図2）．

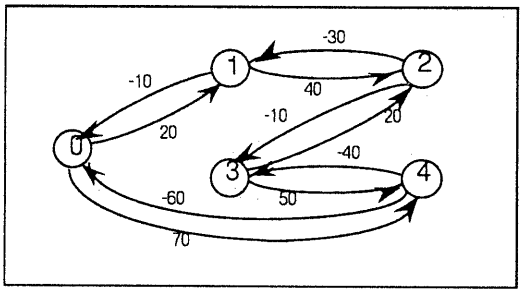

Fig.2 STCSPの距離グラフ

この場合，ノード間の最小制約は最短距離に等しいため，最小制約算出には Warshall-Floyd's All-pairs-shortest-paths algorithm が適している（図3）．負のサイクル（ $d_{ii} < 0$ ）がない場合に限って，STCSPは整合性を持つ．

```
All-pairs-shortest-paths algorithm
1. for i = 1 to n     do dii = 0 ;
2. for i,j = 1 to n   do dij = aij ;
3. for k = 1 to n     do
4.  for i,j = 1 to n   do
5.      dij = min { dij, dik+dkj } ;
```

Fig.3 Warshall-Floyd's algorithm
(dii<0 means minus cycle)

一般的なTCSPは，TCSPをSTCSPに分解して，別々に解いてから結果を統合することによって解決できる．このため，STCSPを基本モデルとして検討することが重要となることから，以下ではSTCSPを取り扱うことにする．

・ TCSPの問題点

現実の時間の概念は極めて曖昧性がありかつ複雑であるため，時間知識をTCSPの制約に落とす時に，TCSPの表現力の不十分さが問題となる．特に，TCSPでは，確定時点に関わる制約しか表せない点で，その表現力に制限がある．

STCSPの整合性維持には時点変数の3乗の計算量を必要とするが，TCSPの全ての組合せを考える場合の計算複雑度は NP-hard であることが証明されている．同様に，整合性のあるTCSPの最小ネットワークを保持するためにも多くなメモリ空間を必要とする．（STCSPなら時点変数ノードの2乗を必要とする）．

3 不確定時点の抽象化

3．1 不確定時点の概念

全節で述べたように，時間知識をTCSPの制約で表す場合，確実な二つの時点変数の制約に書き直さなければいけない．しかし，複数の時点変数からなるグループに関わる制約は，具体的にこのグループ内のどの時点変数に関係するかが明確でない場合もある．例えば，

1）いくつかの出来事からなるイベントに関する制約はこのイベントの始点・終点に関する制約に帰着できるが，始点と終点がはっきりわからない場合．

2）順序がわからない(いろいろな可能性がある)出来事の前後関係に制約がかかる場合．

これを不確定時点（Unfixed Temporal Point）、不確定時点に関する制約と称する[8]．この時，TCSPの表現及び制約伝搬は不可能になる．一つ簡単な例を挙げて説明する．

「例」 AとBの二人はMで待ち合わせしてOに行く場合のシナリオ及びその時間関係を図4に示す．Aが家を出発後一時間以内にOにつくための制約条件？

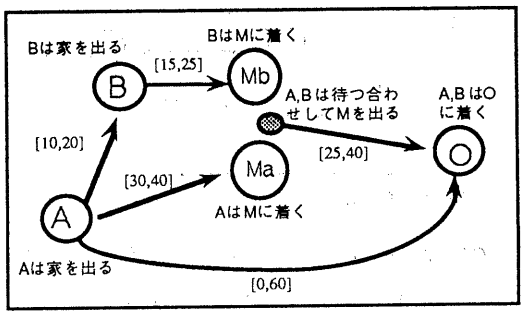

Fig.4 不確定時点の説明例

上記のTCSPでは，MaとMbの前後関係がはっきりわからないので，MからOまで [25,40] の時間制約を（どちらにかけて）表すことができなくなる．この場合，Mから出る時点はMaとMbの終了時点であるため，MaとMbから算出されるはずである．

3．2 不確定時点の抽象化

上記のような不確定時点をTCSPで扱えるようにするため，複数の時点を一つに抽象化した抽象化時点の概念を導入する．

TCSPで取り扱う制約は二つの時点に関するバイナリ制約であるため，不確定時点変数間の前後関係はある他の時点（基準点という）との制約によって比較される．この時，抽象化時点と基準点との制約は不確定時点と基準点間の制約から計算される．基本パターンは図5の二つがある：

終了パターン：いくつかの時点変数の終了時点を

求めるパターン（例えば，待ち合わせ）

開始パターン：いくつかの時点変数の開始時点を求めるパターン（例えば，発生）．

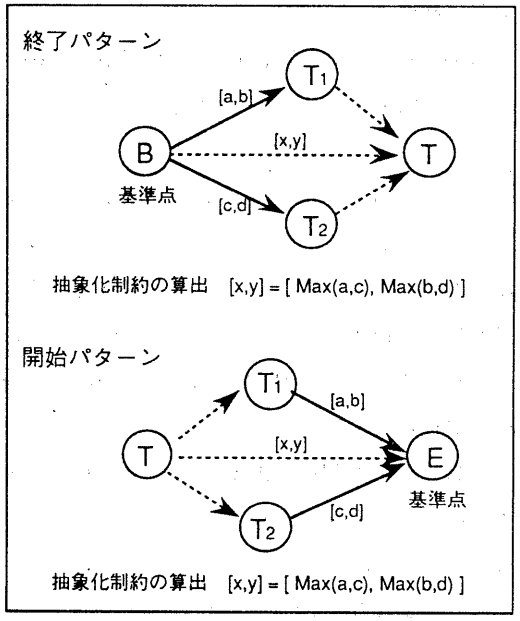

Fig.5 抽象化の二つ基本パターン

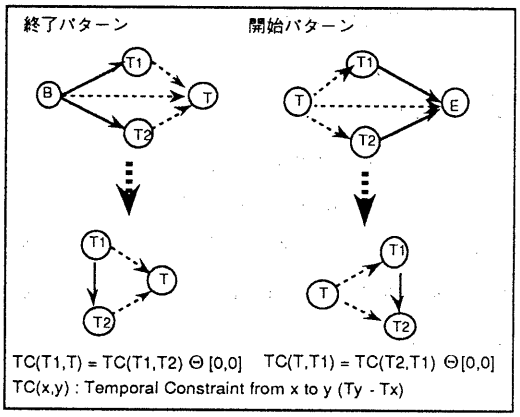

Fig.6 抽象化パターンの変形

図５では，基準時点（B,E）と抽象化時点（T）間の制約は，抽象化される時点（T1、T2）と基準点間の制約から求められる．もし，基準点を介在した制約を直接的に抽象化される時点変数間の制約に変換すれば，図６のようになる．つまり，抽象化に用いる多数時点間の制約からも抽象化制約を求めることができる．

図６の抽象化操作から以下のことがわかる．

１）二つの抽象化パターンを同一の操作に帰着できる．つまり

Θ：[a,b]Θ[c,d]=[Max(a,c),Max(b,d)]

２）互いに制約関係のない時点間でこのような抽象化はできない．

３）多数時点間の抽象化操作はこの二つ基本パターンの組み合わせ及び繰り返しによって算出される．

このΘ操作によって図４の問題は図７のように解決される．

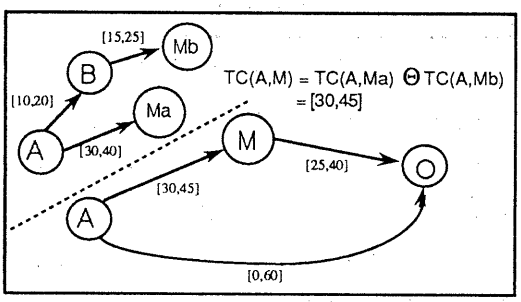

Fig.7 抽象化時点の導入でFig.4の問題の扱い

図７でA－M－Oは単純なSTCSPであるため，Warshall-Floyd の最短経路アルゴリズムによって最小制約ネットワークを算出できる．

4 事象のグループ化

問題領域が異なると，時間知識の形式も違ってくるが，時間推論は主に事象間の同期性と順序性を対象とする場合，事象をまとめて取り扱うのが自然である（図８）．

事象のグループ化によりいくつかの関連事象を一つの事象にまとめて取り扱うことができる．まとめられた事象は始点・終点だけが外部事象

との時間制約を持つことで表すことができる，つまり，外部から内部時点を見る必要がない．このような操作はグループ化と呼び，グループを構成する事象をグループのメンバーという．グループは始点・終点及び始点終点間の制約によって表す．

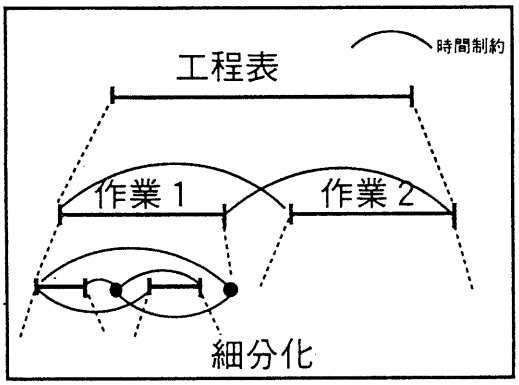

Fig.8 事象のグループの例

ここで，事象は，時点、時区間のほか，グループでもよい．つまり，あるグループは他グループをメンバーにすることができる．

ここで，グループのすべてのメンバーが制約的に連結的であるとき関連しているという．そうでないと，グループの始点終点間の制約は [0,∞) になってしまう．

4．1　グループの記述

グループは以下のようにして記述する．

グループの構成記述：

```
Group :- {all members},
         all constraints between members ;
```

グループを表す関数：

MEM(Group) : Group's All Members

BTP(Group) : Beginning Temporal Point

ETP(Group) : Ending Temporal Point

LEN(Group) : Length of Group

簡単にわかるが、

LEN(Group) = ETP(Group)−BTP(Group)

つまり，LENはグループの始点終点間制約に等しい．

グループのメンバーがグループの場合，子グループ（メンバー）と称する．そうでない場合は一般メンバーという．子グループの始点終点間の制約を簡単化して子グループ制約と称し，グループの始点終点間制約LEN(Group)をグループ制約と言う．

前節で用いた例の問題をグループ化して記述すれば，次のようになる．

```
gAB :- {A,B,Ma,Mb},
       TC(A,B)=[10,15],
       TC(A,Ma)=[30,40],
       TC(B,Mb)=[15,25],
       BTP(gAB)=A;
gAll :- {gAB,C,O},
       TC(BTP(gAB),O)=[0,60],
       TC(ETP(gAB),O)=[25,40] ;
```

ここで TC(x,y) は時点変数xとyとの間のバイナリ時間制約（Temporal Constraint）を表している．

4．2　グループ始点終点算出アルゴリズム

グループに対しては，一般の最小化法でその内部時点間の制約を求めるほかにも，グループの始点終点及び始点終点間の制約を求める必要もある．もし，始点終点が不明確な場合，不確定時点の抽象化操作

を利用してグループのメンバーから始点・終点を導くことができる．具体的なアルゴリズムは図９で与えている．このアルゴリズムの計算量はｎの２乗である．

このようなグループ化操作によって不確定時点を気にせずに扱うことができるようになる．

```
TC(B,E) = [0, 0]
for Ti = T1 to Tn
   TC(B,Ti) = [0, 0]
   TC(Ti,E) = [0, 0]
   for Tj = T1 to Tn
      TC(B,Ti) = TC(B,Ti) ⊖ TC(Tj,Ti)
      TC(Ti,E) = TC(Ti,E) ⊖ TC(Ti,Tj)
      TC(B,E) = TC(B,E) ⊖ TC(Ti,Tj)
   end for
end for

* T1,...,Tn : all points in the group
* TC(x,y) : Temporal Constraint from x to y
* B is BTP(group) and E is ETP(group)
* ⊖ : [a,b] ⊖ [c,d] = [Max(a,c),Max(b,d)]
```

Fig.9 グループ化アルゴリズム

5　時間知識ベースの階層化

１節で述べたように，TCSPには表現力の不十分性と計算量、メモリ量の増大などの問題点がある．時間知識ベースの規模が大きくなればなるほど時空量増大の問題は顕著になってくる．そのため，大規模な時間知識ベースを維持する時に，システムの構築法をよく考慮する必要がある．

グループ化によって階層化表現することは時間知識ベースの効率によい効果があると考えられる[5]．いま，階層化の構築法に以下のような制限を与える．

グループは他グループのメンバーになることができるが，直接属している親グループは一つ以下である．

このように事象からグループを形成し，グループにより時間知識を階層化して構築することができる（図８）．この階層化においては不確定時点に特別な注意を払わなくても良い．ただし，構築制限のため，グループのメンバーは外部から直接には見えなくなる．一旦グループにまとめたら，このグループの始点終点しか参照できなくなる．

以下，階層化に関する制約伝搬法及び整合性維持法について述べる．最後にその効率を評価する．

5．1　階層化のいくつかの概念

親子関係：あるグループにおいてグループはすべての内部メンバーの親であり，すべての内部メンバーはグループの子となる．

トップグループ（Top Group）：親のないグループはトップグループである．トップグループによって無関係な事象列は区別される．

レベル：トップグループからの階層のレベルである．トップグループのレベルは０である．

親リスト：トップグループまでのすべてのグループからなる集合である．

5．2　整合性維持法

階層化時間知識ベースの整合性維持法はグループ追加，グループ内追加，制約変更などの場合の処理を含んでいる．

グループ追加は時間知識ベースに新たなグループを追加することなので，知識ベースの階層化を構築するための最も基本的な操作である．グループのメンバー MEM(Group) に時点変数と子グループを含む場合の追加操作は次の通りである．

[１] グループメンバーの始点終点間制約をグループ内に取り入れる．

[２] グループ内の制約充足及びグループ制約の算出

を行う.

[3] 親子関係、レベルなどを整理する.

[4] 制約が変更された場合の変更処理を行う.

グループ内制約（またはメンバー）が追加された場合，次のように追加前の制約関係を更新する必要がある.

[1] 追加メンバーがグループである場合，そのグループ制約を追加制約とする.

[2] TCSPの制約追加操作を行う.

[3] 親子関係、レベルなどを整理する.

[4] 制約が変更された場合の変更処理を行う.

あるグループ内の二つ時点変数間の制約が変更された場合，制約の種類に従って以下のように変更処理は異なる.

[1] 非子グループ制約の変更なら処理しない.

[2] 子グループ制約の変更なら子グループ制約を子グループに追加する.

[3] グループ制約の変更ならグループ制約をその親グループに追加する.

ここでは，具体的なアルゴリズムは示していないが，Warshall-Floyd最短経路アルゴリズムの他にいくつかの高速アルゴリズムも利用できる．例えば，湯上（富士通）のグラフ二重分割高速アルゴリズム[7]など.

5．3　制約の取り出し

階層化した場合，グループの内部時点間の制約は保持されているが，異なるグループ間の時点間の制約は直接には保持ていないため，必要な時にグループ間の制約伝搬によって計算する．隣接していないグループ間の制約伝搬はグループをつなぐ層間の伝搬によって求められる.

・　制約伝搬経路

二つの親リストを比較して，最も近い共通親までの経路を制約伝搬経路という（図１０）．経路が存在していない場合は，制約がないことを表し，つまり，(-∞,∞) である.

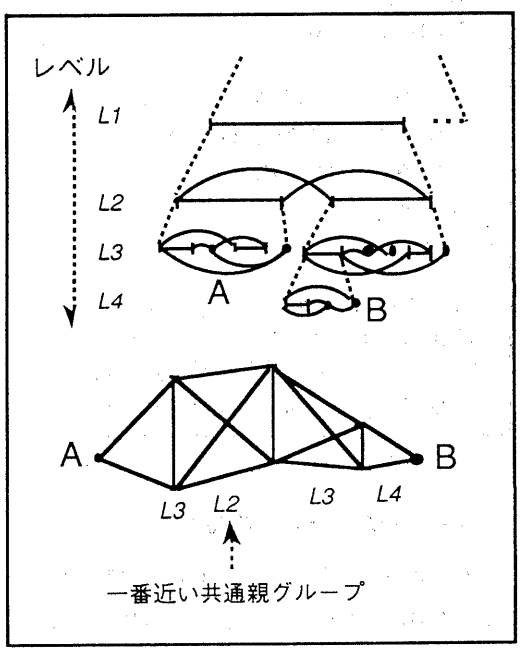

Fig.10 階層化した場合の制約伝搬経路

・　層間の制約伝搬

制約伝搬の基本は層（グループ）間の伝搬である．図１１は時点変数OとA層間の制約からOとB層間の制約を求める例であり，図示した制約はすべて計算済みの最小制約である．図から分かるように，n層までに制約を伝搬するには１２ｎの計算量が必要となる．時点Oをグループに置き換えた場合の制約伝搬法も同じように計算できる.

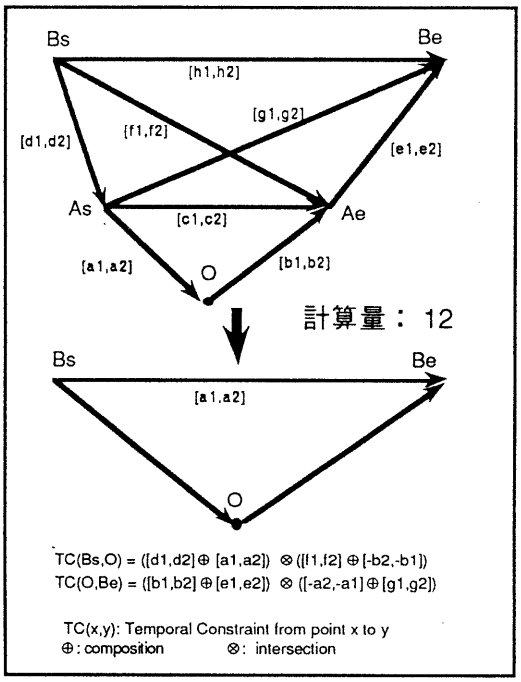

Fig.11 層間の制約伝搬の計算

5．4 矛盾処理

全体の整合性を維持する途中で矛盾が検出された場合，直ちに操作を中止し，エラー報告またはエラー回復に処理を移す．

5．5 階層化による推論効率の評価効

n 時点変数のSTCSPに対して，

計算量：n の3乗

メモリ：n の2乗

として階層化による推論効率を評価する．グループ化と階層化によってnは小さくなり，計算とメモリ量の効率は高くなる．ただし，制約を取り出す場合，層間伝搬によるダイナミックな計算が必要となる．

表1は100 時点を 10 グループ 2 層にする場合の効率評価例を示している．

	階層化無し	階層化あり
制約充足計算量	100 x100 x100= 1,000,000	10x1000+20x20x20= 18,000
最小制約メモリ	100x100= 10,000	10x100+20x20= 1,400
制約取出計算量	0	グループ内 0 グループ間 12x2+6*= 30

[*] 最後に共通親層での制約計算は6である．

Table1 階層化による推論効率の評価例

5．6 階層化制限条件の緩和

階層化の制限条件は時間知識ベースの構成に制限を与えるが，時間知識ベースの管理上の複雑さの軽減と不確定時点の扱いに有効である．制限外の制約がある場合，グループ分割と合成、親グループ間の制約への近似変換などの手法は利用できる可能性がある．実用的には，外部からグループの始点・終点だけに制約をかける場合が極めて多い．

6 むすび

時間概念の曖昧性と複雑性などに対処するため，不確定時点の抽象化操作を考案し，抽象化操作を利用していくつかの事象のまとまりをグループ化して扱う方法についての検討を示した．その上で，時間推論を効率化するため，グループ化による時間知識ベースの階層化による構築法を示した．今後，より多くの機能を導入するつもりである．また既存TCSPネットを階層化に変換する方法についても検討してみる予定である．更に，時間知識ベースにおける与えられるゴールを満たすのに必要な無矛盾な仮説の集合を見いだす仮説推論について検討する予定である．

参考文献

[1] J.F. Allen : Maintaining Knowledge about Temporal intervals, *Communication of ACM*, vol.26, No.11, pp. 882-843(1983).

[2] R. Detcher, I. Meiri, J. Pearl : Temporal Constraint Network, *Knowledge Representation*, pp. 61-96(1992).

[3] I.Meiri : Combining Qualitative and Quantitative Constraints in Temporal Reasoning, *AAAI-91*, pp. 260-267(1991).

[4] H. A. Kautz, P. B. Ladkin : Integrating Metric and Qualitative Temporal Reasoning, *AAAI-91*, pp. 241-246(1991).

[5] Johannes. A. G.M. Koomen : Localizing Temporal Constraint Propagation, *AAAI-89*, pp198-202(1989).

[6] 新田克巳：時間を含む知識の表現と推論，*人工知能学会誌* Vol.3 No.5, pp.563-571(1988).

[7] 湯上伸弘：時間制約充足問題の解法，*1992年度人工知能学会全国大会（第6回）No. 11-3.*

[8] 呉楽水，石塚満：不確定時点に関する時間制約を含む時間推論方式，*1993年電子情報通信学会秋季大会No. D-127.*