人　工　知　能　80－1
記　号　処　理　63－1
（1992．1．16）

定量的制約とリレーションによる時区間の時間的性質の決定

佐々木寛　山田誠二　豊田順一
大阪大学産業科学研究所

人工知能の問題の一つに現実世界の時間概念をどのように表現し，どのように推論を行うかという問題がある．本研究では対象とする環境に適した時間順序に従った複数エージェントの時区間の定量的属性を決定するために，時区間についての定量的制約とAllenによって提案された定性的制約であるrelationを扱う枠組みを提案する．具体的には、relationにみられる定性的制約と，時区間幅，時刻（時点），方向の３種類の定量的制約を用いて時区間の決定を行う．一つのrelationを満足する時区間は限りなく存在し，かつ周囲の時区間との競合を伴う為，一意に決定する事は困難である．そこで，定性的制約，定量的制約，最適値の反映を統一的に扱え，かつ一意に解を決定できることから，数理計画法を用いる．また，本研究で扱うクラスが線形計画問題の範疇に入ることを示す．

Determining temporal properties of intervals by quantitative constraints and relations

Hiroshi Sasaki　Seiji Yamada　Junichi Toyoda
I.S.I.R. Osaka University

This paper describes new representation and a method of determining temporal properties of intervals based on Allen's Interval Logic and Kowalski's Event Calculus. In this paper, to determine "time points" of intervals, which is suitable for the environment around agents, a method with mathematical programing, in which qualitative and quantitative constraints are integrated, is proposed. First, we input quantitative constraints, qualitative constraints, optimal values into system. The quantitative constraints are "duration" of intervals, and "time points" of intervals. The qualitative constraint is temporal relations proposed by Allen. The optimal value is duration of intervals as an optimal duration of interval. Finally, the system makes out an appropriate list of intervals that is suitable for environment. Furthermore, for maintaining the consistency of whole relations, we use mathematical programing.

1．はじめに

人工知能の課題の一つに，現実世界ではごく日常的に扱われている時間概念をどのように表現し，推論を行うかという問題がある．本研究では，既に設定された時区間の関係を基に時区間の時間幅の定量値を，デフォルト値とAllenによるリレーションから数理計画法を用いて扱う枠組みを提案する．

Allenのリレーションは，前後関係等の時区間の定性的関係のみを記述しているため，定量的制約から実際にはありえないような時区間の関係をも無矛盾とみなす危険がある．そこで，本研究ではAllenのリレーションにおいて無矛盾な時区間とリレーションの集合に対し，定量的制約を付加して時区間の時間幅を決定する．一つのリレーションを満足する時区間の集合は無限に存在し，一意に決定する事はできないが，時間幅に関する最適値等の制約により，数理計画法を用いて決定できる．この問題は，線形計画法で扱え決定的手続きで一意に解が得られるクラスである．

2．Allenの区間論理とデフォルト推論

Allenによって提案された区間論理[1]では，時区間表現の基本として時点ではなく，時区間を基本とする．そして，これらの相対的リレーション（二項関係）で時間が表現される．この二項関係は７種類定義されており（Fig. 1），この二項関係が時区間すべての間に与えられていない場合には，既知の関係から伝播則(推移則)（Table 1）をつかって新しい二項関係が推論される．

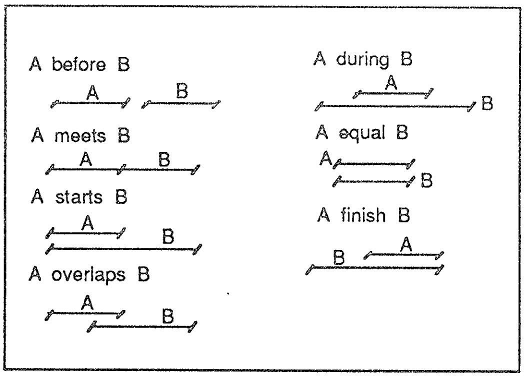


Fig.1 Allenのrelation

Table1　ALLEN'S TRANSIVITY TABLE

B r2 C / A r1 B	<	>	d	di	o	
"before" <	<	no info	< o m d s	<	<	
"after" >	no info	>	> oi mi d f	>	> oi mi d f	
"during" d	<	>	d	no info	< o m d s	

例えば，

before(A B)&before(B C)

=> before(A C)

meet(A B)&during(B C)

=> (overlap(A C),during(A C),or meet(A C))

が伝播則により推論される．表現される世界は時間表現の静的要素としてproperty（属性）を用意し，動的な要素としてevent（出来事）とprocess（過程）を用意している．ある属性Pが区間Aで成立することを，述語Holdsを用いて，

Holds(P A)

と表現する．Holds(P A)が成立するとき，Aの任意の部分区間A'においてHolds(P A')が成立する．属性に対し，行為や時区間等のように属性を引き起すものを，生起(occurence)という．生起には，process（過程）とevent（出来事）がある．processは変化を生じない行為（出現回数を数えることの不可能な行為）であり，「太郎は歩いている」はprocessに相当する．一方，eventとはprocessとは逆に，変化を生ずる行為である．例

えば「太郎は学校まで歩いた」はこのeventに相当する．

Allenによって提案された推論は，時区間の関係についてのみ推論なので，順序関係を取り扱うことに適している．しかし，時区間の定量的な長さは考慮しておらず，絶対的な時間軸が存在しない為，時区間に存在する定量的な関係を扱うことはできない．さらに，局所的な伝播則により推論が行なわれるため，大域的に無矛盾な無矛盾姓を保証できないという問題もある[1]．また，Allenのリレーションで大域的に無矛盾でも，定量的な制約によりありえない結果になる場合もある．

一方，時区間の時間幅を定量的に扱ったものとして，KowalskiのEvent Calculus[2]がある．これは，不完全な情報から時区間の時間幅，開始終了時点を推論するもので，時間データベースの管理と談話理解を主目的とする．Event Calculusでは，時区間をその発生時区間と終了時区間だけで表現する．例えば，「五日前にAがBに本を渡した」という時区間E1の効果は，時区間A「花子が本を所有している」を終了し，時区間B「太郎が本を所有している」を開始することである．その後「きのう太郎が本を次郎に与えた」という時区間E2が追加されると，時区間BはE1とE2の間に成立していたことが推論される．また，ある２つの時点における時区間が異なれば，E1とE2の間に別の時区間があったことが推論される．ここで，「時区間は発生すると，中断されたことが明示されない限り継続する」というデフォルト規則を用いているため，「三郎は９時からミーティングである」という知識からは，「（９時から）未来永劫ミーティングである」という滑稽な結果が推論される．これは，デフォルト規則が適切でないたことによる．また，Event Calculusでは時区間の始点と終点しか記述できない為，我々が日常扱う開始・終了・継続といった時区間に関する様々な形の時間概念に対応する事が困難である．

本研究では，Allenの時間推論の解，つまり大域的に無矛盾な時区間とリレーションの集合に対し，時区間の時間幅に関する定量的制約(最適，最大，最小値)を付加し，それらの制約をすべて満たすように，絶対時間軸上での時区間の開始時点と終了時点の決定を行なう．この制約充足問題は，線形計画法により決定的手続きにより，一意に解を求めることが可能なクラスである．

結果として本研究は，Allenの区間論理に定量的制約を与えており，さらにEvent Calculusにより適切なデフォルト規則とAllenのリレーションを付加したことになり，時間推論における定性的制約と定量的制約の融合になっている．

３．時間に関する制約とその表現

３．１　時区間と時点

本研究で扱う時区間とは，絶対時間軸上におけるある大きさをもつ領域を意味する．また，時点とは，絶対時間軸上における一点である．つまり，時点は時間幅が０である．

本研究で扱う時点は，以下のものに限定される．

・開始時点：時区間の開始を表す時点.

・終了時点：時区間の終了を表す時点.

・継続時点：開始/終了時点以外の時区間中の任意の時点.

さらにこれらの時点は，確定時点と不確定時点に分類される．例えば，「一時に会議が始まるのを見た」という情報からは確定-開始時点が得られ，「一時に会議が終わるのが常だ」という知識からは，不確定-終了時点が得られる．以上の時点の分類をまとめると，Fig.2のようになる．ここで確定時点の値は不変であるのに対し，不確定時点はあくまで望ましい最適値として扱われる．

- 時点
  - 開始時点
    - 確定-開始時点
    - 不確定-開始時点
  - 継続時点
    - 確定-継続時点
    - 不確定-継続時点
  - 終了時点
    - 確定-終了時点
    - 不確定-終了時点

Fig.2 時点の分類

３．２　定量的制約

時区間には種々の属性があるが，本研究では絶対時間軸上での時区間の開始時点と終了時点の決定を目

指す．当然のことながらそのような定量的属性を決定するためには，定量的制約が必要になる．以下に本研究で用いる定量的制約を示す．

(A) 時区間の時間幅の最適値，最大値，最小値

(B) 時点

定量的制約(A)：時間幅の最適値とは経験的にわかっている，あるいは望ましい時区間の長さであり，最大最小値は時間幅の取りえる値の範囲を表現している．時間幅の最適値と最大値を無限大，最小値を無限小としたとき，前述のデフォルト推論によるEvent Calculusに等価になると考えられるので，定量的制約(A)はデフォルト推論の拡張になっている．原らの研究[3]もデフォルトの拡張を行なっているが，本研究の方が扱える範囲が広いと考える．

３．３　定性的制約

本研究では，定性的制約としてAllenのリレーションと時区間のタイプを扱っている．リレーションはAllenの提案した７種類を用いる(Fig.1)．本研究では，定量的制約と定性的制約を数理計画法で統一的に扱うため，すべてのリレーションは線形等式・不等式群による表現に変換される．

以上述べた本研究で取り扱う時区間に関する情報から得られる制約を定性/定量及び数理計画法の制約式/評価関数という視点でまとめるとFig.3のようになる．

定量的情報	区間幅（最適値）	評価値
	区間幅（最大／最小）	制約
	確定時点	制約
	非確定時点	評価値
定性的情報	Allenのrelation	制約

Fig.3 本研究の扱う制約と評価値

３．４　制約の記述形式

本節では，制約の記述形式について述べる．以下の４種類の記述形式ですべての制約入力を行う．ただし，書式はLispの関数に準拠している．

(1) 時区間の表現

書式：(duration INTERVAL OPTIMAL MIN MAX)

最適，最小，最大値は，すべて時区間のそれである．最大/最小値がない場合はNILを入力する．例えば，会議の最適値が２時間，最小値が１時間であるというのは，(duration 'meeting 2 nil nil)と記述する．

(2) リレーションの表現

書式：(relation (INTERVAL1 AGENT1) (INTERVAL2 INTERVAL2) RELATION)

時区間は，時区間名とエージェントとのリストで表現する．これは，異なるエージェントにより実行される同一名時区間があるためである．例えば，太郎は会議１を会議２の前に開きたいというのであれば，次のように表現となる．

(relation '(Meeting1 Taro) '(Meeting2 Taro) ' before)

(3) 時点の表現

3.1で述べたように，本研究で扱う時点は，時区間の開始，継続，終了時点の３種である．これらの時点を独立に入力する必要はなく，人間が扱いやすい組み合わせで記述すればよい．この組み合わせをタイプと呼ぶ．ここで用いたタイプとそれに含まれる時点をTable 2に示す．表中のTIME-LISTが，それぞれのタイプによって与えられる時点を表す．タイプのうち，周期は少し特殊であり，開始時点とその時点が周期T1で繰り返されることを表す．

また，与えられる時点が，確定時点であるか不確定時点であるかで書式が異なる．残念ながら現状では，同一時区間について，確定時点と不確定時点の一方しか与えられないという制限がある．

・確定時点の記述

書式： (certain-event INTERVAL (AGENT1 ･･･ AGENTn) TIME-LIST TYPE)

・不確定時点の記述

書式： (uncertain-event INTERVAL (AGENT1 ･･･ AGENTn) TIME-LIST TYPE)

両者とも時区間(EVENT)がエージェント(AGENT)により，いつ(TIME-LIST)実行されるかを，時区間の状態(TYPE)を付加して記述する．ここで，TIME-LISTとTYPEの関係については，Table 2参照．

例として，「会議1は2時間行なうのが望ましい」，「1時から太郎は会議1に出席するのが常である」という知識は以下のようになる．

(duration 'meeting1 2 nil nil)

(certain-event 'meeting1 '(Taro) '(1) 'forward)

さらに，「太郎はその後（会議1の後）花子と会議2を開きたい」のであれば，

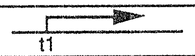
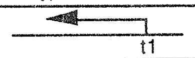
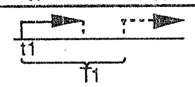
(duration 'meeting2 2 nil nil)

(uncertain-event 'meetng2 '(Taro Hanako) '() 'forward)

(relation '(meeting1 Taro) '(meeting2 Taro) 'before)

とすればよい．

Table2 TIME-LIST and TYPE

TYPE	TIME-LIST	意味	状態図
開始(forward)	(t1)	t1から開始	
終了(backward)	(t1)	t1に終了	
継続(hold-at)	(t1)	t1で真	
区間(interval)	(t1 t1)	t1からt2まで真	
周期(cycle)	(t1 T1)	t1から真で,周期T1	

4．数理計画法の適用

4．1　制約の変換

本節では，これまで述べてきた定量的制約と定性的制約とがどのように統合的に処理されるのかを述べる．これまでに述べた時区間に関する制約をまとめると，以下のようになる．

(1)時区間のリレーション，時間幅の最大／最小値，確定時点

(2)時間幅の最適値，不確定時点

ここで重要なことは，すべての時区間の時区間の端点を変数とすると，Allenのリレーションを含む上記の制約はすべて，線形等式・不等式群と評価関数に変換可能であることである．(1)では，時区間のリレーションは時間的な先行後続関係を，時区間幅の最大／最小値は時区間の開始時点と終了時点との差の不等関係を示しており，また確定時点は等式であるから，(1)はすべて線形方程式不等式に変換可能であることがわかる．(2)は，時間幅の最適値，不確定時点ともに各変数のとり得る望ましい最適値を表している．つまり，誤差最小の関数に変換可能である．したがって，この線形方程式不等式と2乗誤差の評価関数は，数理計画法において解法が知られている凸2次計画法と呼ばれるクラスの問題であり，決定論的に解決可能である．このように，定性的制約と定量的制約を統一的に扱うために，凸2次計画法の制約式と評価関数にすべての入力制約を変換するための変換表をTable3.1,3.2,3.3,3.4に示す．評価関数は，時区間の時間幅と不確定時点についてそれぞれの最適値との二乗誤差の総和である．

制約から数理計画法への変換の具体的な例を以下に挙げる．

Table3.1 translate into equations from "duration"

TRANSLATED EQUATIONS
$\Phi = (INTERVAL_{end} - INTERVAL_{st} - OPTIMAL)^2$ $min \le INTERVAL_{end} - INTERVAL_{st} \le max$

Table3.3 translate into equations from "uncertain-event"

TYPES	TRANSLATED EQUATIONS
FORWARD	$\Phi = (INTERVAL_{st} - t1)^2$ $INTERVAL_{st} \le INTERVAL_{end}$
BACKWARD	$\Phi = (INTERVAL_{end} - t1)^2$ $INTERVAL_{st} \le INTERVAL_{end}$
HOLD-AT	$\Phi = (INTERVAL_{mid} - t1)^2$ $INTERVAL_{st} \le INTERVAL_{mid}$ $INTERVAL_{mid} \le INTERVAL_{end}$
INTERVAL	$\Phi = (INTERVAL_{st} - t1)^2 + (INTERVAL_{end} - t2)^2$ $INTERVAL_{st} \le INTERVAL_{end}$
CYCLE	$\Phi = (INTERVAL_{st} - t1)^2 + (INTERVAL'_{ST} - t1 - T1)^2$ $INTERVAL_{st} \le INTERVAL_{end}$ $INTERVAL_{end} \le INTERVAL'_{st}$ $INTERVAL'_{st} \le INTERVAL'_{end}$

Table3.2 translate into equations from "certain-event"

TYPES	TRANSLATED EQUATIONS
FORWARD	$INTERVAL_{st} = t1$ $INTERVAL_{end} \ge t1$
BACKWARD	$INTERVAL_{st} \le t1$ $INTERVAL_{end} = t1$
HOLD-AT	$INTERVAL_{st} \le t1$ $INTERVAL_{mid} = t1$ $INTERVAL_{end} \ge t1$
INTERVAL	$INTERVAL_{st} = t1$ $INTERVAL_{end} = t2$
CYCLE	$INTERVAL_{st} = t1$ $INTERVAL_{end} \ge t1$ $INTERVAL'_{st} = t1+T1$ $INTERVAL'_{end} \ge t1+T1$

Table3.4 translate into equations from "relation"

RELATIONS	TRANSLATED EQUATIONS
EQUAL	$INTERVAL1_{st} = INTERVAL2_{st}$ $INTERVAL1_{end} = INTERVAL2_{end}$
BEFORE	$INTERVAL1_{end} \le INTERVAL2_{st}$
MEET	$INTERVAL1_{end} = INTERVAL2_{st}$
START	$INTERVAL1_{st} = INTERVAL2_{st}$ $INTERVAL1_{end} \le INTERVAL2_{end}$
OVERLAP	$INTERVAL1_{st} \le INTERVAL2_{st}$ $INTERVAL2_{st} \le INTERVAL1_{end}$ $INTERVAL1_{end} \le INTERVAL2_{end}$
DURING	$INTERVAL2_{st} \le INTERVAL1_{st}$ $INTERVAL1_{end} \le INTERVAL2_{end}$
FINISH	$INTERVAL2_{st} \le INTERVAL1_{st}$ $INTERVAL1_{end} = INTERVAL2_{end}$

(duration 'task1 3 1 4)

(duration 'task2 4 nil nil)

(certain-event 'task1 '(agent1) '(10) 'forward)

(uncertain-event 'task2 '(agent2) '(12) 'hold-at)

(relation '(task1 agent1) '(task2 agent2) 'meet)

この例は，２つの時区間task1, task2が，２つのエージェントにより実行されるスケジュールである．

いま各時区間の開始時点，中間時点，終了時点を，-st, -mid, -endで各々示すことにすると，変換表を用いて以下の線形等式・不等式群と２次関数に容易に変換することができる．

(duration 'task1 3 1 4)より

評価関数：$\Phi = (task1_{end} - task1_{st} - 3)^2$

線形不等式：$1 \le task1_{end} - task1_{st} \le 4$

(duration 'task2 4 nil nil)より

評価関数：$\Phi = (task2_{end} - task2_{st} - 4)^2$

線形不等式：$0 \le task1_{end} - task1_{st}$

(observative-event 'task1 '(agent1) '(10) 'forward)より

線形不等式：$task1_{st} = 10$,

$task1_{st} \le task1_{end}$

(unobservative-event 'task2 '(agent2) '(12) 'hold-at) より

評価関数：$\Phi = (task2_{mid} - 12)^2$

線形不等式：$task2_{st} \le task2_{mid}$,

$task2_{mid} \le task2_{end}$

(relation '(task1 agent1) '(task2 agent2) 'meet)より

線形不等式：$task1_{end} = task2_{st}$

以上の線形等式・不等式群を満足する範囲で評価関数の和を最小化する数理計画法に帰着できる．このクラスの問題は前述したように凸2次計画法と呼ばれるクラスに含まれる．最終的な制約式と評価関数は以下のようになる．

評価関数：$\Phi = (task1_{end} - task1_{st} - 3)^2 + (task2_{end} - task2_{st} - 4)^2 + (task2_{mid} - 12)^2$

制約式：$1 \leq task1_{end} - task1_{st} \leq 4,$

$0 \leq task1_{end} - task1_{st},$

$task1_{st} = 10,$

$task1_{st} \leq task1_{end},$

$task2_{st} \leq task2_{mid},$

$task2_{mid} \leq task2_{end},$

$task1_{end} = task2_{st}$

4．2　凸2次計画法

本節では，制約式と評価関数が与えられたときの凸2次計画法について述べる．このクラスの問題は，クーンタッカー条件を満足すれば，それが最適解であることが保証されている[4]．以下にその実際を示す．

一般に，制約不等式及び評価関数は次のように書くことができる．

$$F(x) = \frac{1}{2}x^t Dx + c^t x \Rightarrow min$$

$$Ax \geq b$$

$$x \geq 0$$

先ほども述べた通りクーンタッカー条件により，凸2次計画法の最適値を求めることは，与えられた次のM，qをもとに作られる線形方程式$w = q + Mz + w$ を満足するw，zを求めることと同値である．

$$w = \begin{bmatrix} \lambda \\ \mu \end{bmatrix}, z = \begin{bmatrix} x \\ u \end{bmatrix}, q = \begin{bmatrix} c \\ -b \end{bmatrix}, M = \begin{bmatrix} D & -A^t \\ A & 0 \end{bmatrix}$$

$$w = q + Mz + w\ ,\ z^t w = 0$$

$$w \geq 0\ ,\ z \geq 0$$

この問題は線形相補性問題と呼ばれており，システムではこれを相補性条件を加えたシンプレックス法で解いている．つまり，凸2次計画法に帰着された問題は，次の相補性条件を考えた線形計画問題にさらに帰着されることになり，これを解けばよいことになる．

$$\sum \omega \Rightarrow 0$$

$$w = q + Mz + \omega\ ,\ z^t w = 0$$

$$w \geq 0\ ,\ z \geq 0\ ,\ \omega \geq 0$$

もし，求められた解の人為変数ωが0以外の値をもつならば，凸2次計画問題の最適解は存在しない．なぜなら，人為変数とは，値が0であることを前提として導入された変数だからである．なお，制約充足の計算量は，時区間数をnとしたとき，$O(n^3)$である．これはシンプレックス法にかかる計算量から求められた．

5．　例題

具体例として，システムに次のスケジュール問題を解かせた．

（問題）

・太郎は１０時から１８時まで会社にいると思われる．
・太郎は昼食を１１時から１３時の間に３０分から４５分くらいかけて食べたいと思っている．
・花子は９時から１７時まで会社にいると思われる．
・花子は昼食を１１時から１４時の間に１時間から２時間くらいかけて食べたいと思っている．
・次郎は１１時から１９時までの会社にいると思われる．
・次郎は昼食を１２時から１４時の間に４５分から１時間くらいかけて食べたいと思っている．
・太郎は次郎と会議１を２時間から３時間開きたい．
・太郎は花子と会議２を会議１の後に１時間から２時間開きたい．

ただし，ここでは時区間の最適値は与えられてない為，最大値と最小値の平均値を最適値とし，時区間が重ならないことをリレーションに追加したことを断っておく．結果をFig.4.1，Fig.4.2に示す．

```
(setting)
    (duration 'at_company 8 nil nil)
    (duration 'lunch1 5/8 1/2 3/4)
    (duration 'lunch2 3/2 1 2)
    (duration 'lunch3 7/8 3/4 1)
    (duration 'meeting1 2 nil nil)
    (duration 'meeting2 2 nil nil)
    (uncertain-event 'at_company '(taro) '(10 18) 'interval)
    (uncertain-event 'lunch1 '(taro) '(11 13) 'interval)
    (uncertain-event 'at_company '(hanako) '(9 17) 'interval)
    (uncertain-event 'lunch2 '(hanako) '(11 14) 'interval)
    (uncertain-event 'at_company '(jiro) '(11 19) 'interval)
    (uncertain-event 'lunch3 '(jiro) '(12 14) 'interval)
    (uncertain-event 'meeting1 '(taro hanako) '() 'interval)
    (uncertain-event 'meeting2 '(taro jiro) '() 'forward)
    (relation '(lunch1 taro) '(at_company taro) 'during)
    (relation '(lunch2 hanako) '(at_company hanako) 'during)
    (relation '(lunch2 hanako) '(at_company hanako) 'during)
    (relation '(lunch3 jiro) '(at_company jiro) 'during)
    (relation '(meeting1 taro) '(at_company taro) 'during)
    (relation '(meeting2 taro) '(at_company taro) 'during)
    (relation '(meeting1 hanako) '(at_company hanako) 'during)
    (relation '(meeting2 jiro) '(at_company jiro) 'during)
    (relation '(meeting2 taro) '(meeting1 taro) 'after)
    (relation '(meeting2 jiro) '(lunch3 jiro) 'after)
    (relation '(meeting1 hanako) '(lunch2 hanako) 'after)
    (relation '(meeting1 taro) '(lunch1 taro) 'after)
(*eof)
```

Fig.4.1 program for problem

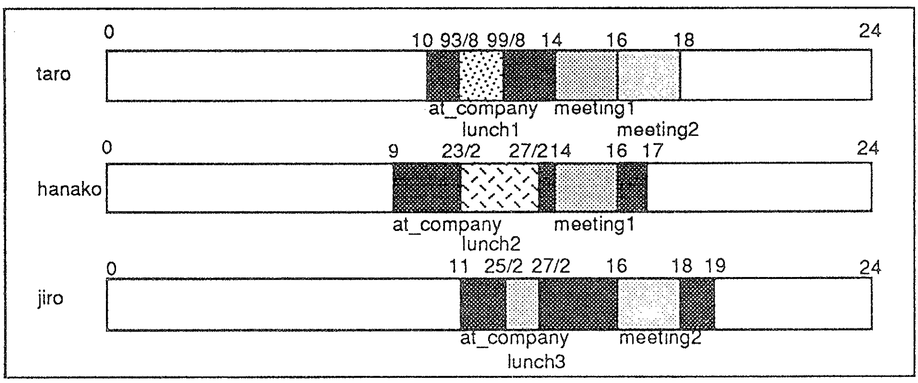


Fig.4.2 solution of problem

６．問題点

本研究における問題点を挙げる．

・時間軸の下限

制約不足による結果をFig.5に示す．与えた制約は開集合である．このような場合，時区間が0時から開始される解となる．この理由として，次の２点が考えられる．

```
(setting)
(duration `task1 2 nil nil)
(duration 'task2 2 nil nil)
(certain-event 'task1 '(agent1) '(12)  'forward)
(uncertain-event 'task2 '(agent1) '() 'forward)
(relation '(task2 agent1) '(task1 agent1)
'before)
(*eof)
```

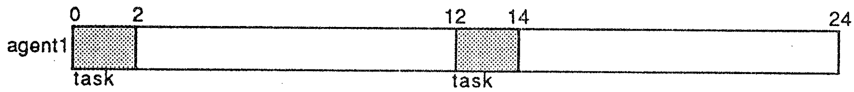


Fig.5

(1)線形計画法で解いているため，変数は正値である．したがって制約範囲が自動的に閉集合となり凸２次計画法により解が得られる．さらにシンプレックス法にて解いているため制約集合の頂点が解として選択されてしまう．

(2)対象とする不確定時点の値は，nilである．これは，評価値が0とみなすことができる．つまり開始時刻の最適値が0であることになる．

以上の２点から0時から開始される結果が得られたと考える．

(2)の解決法としては，現実的に起こり得ない領域にはなるべく時区間が入らないように評価関数あるいは制約を与えることで，現実的な解が得られる．しかし，(1)は数理計画法で扱うことの限界を示している．したがって，(1)に対する本質的な解決法はないと思われる．

・リレーションの厳密さ

数理計画法に帰着させることで一意に解を算出する事ができる反面，「Allenのリレーションの線形等式・不等式群への変換が完全でない」という問題がある．変換が現密に行えていないリレーションは，次の５つである．

before,overlap,during,start,finish

これは次の理由による．

「数理計画法で扱える制約不等式は閉集合でなければならない」

つまり，リレーションでは等号を含まないにもかかわらず，数理計画法で扱う為に等号を付加し閉集合としなければならないのため，場合によっては，beforeで記述された時区間が結果としてmeetとなることもありうる．本研究では，これを次のように解釈する．結果がmeetとなるのは他の影響（他の時区間との競合や制約式あるいは評価値の影響）を受けた結果であり，入力されたリレーション（この文中では，overlapを示す）は満足しているので，制約を満足する出力として十分である．

７．まとめと今後

本研究では，従来の区間論理とEvent Calculusを基礎として，既に得られている定性的制約に定量的制約を付加し，双方を満たすような時区間のパラメータを数理計画法による時区間の演算手法を提案した．

また，本研究の応用として，定性的制約であるAllenのリレーションのみを充足するようなプランニング[5]から得られたプラン(Fig.6)の検証を行なうことを考えている．このプランは，リレーションに関しては大域的に無矛盾であるが，オペレーションに対応する時区間に対する定量的制約を加えて，数理計画法での解の有無を調べることにより，実現可能なプランであるか否か検証できる．具体的には，Allenのアルゴリズムより求められた解に定量的制約を加え，本システムに入力する．その結果，Allenのアルゴリズムでは複数個ある解集合を，さらに絞り込むことが可能であると思われる．

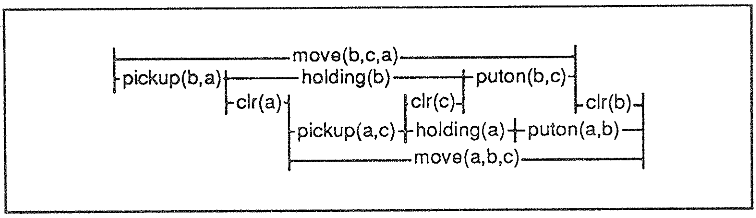


Fig.6 planning result using a temporal world model

＜謝辞＞

本研究を進めるにあたり，数々の有益な助言を与えてくださった辻三郎教授と豊田研及び辻研の皆さんに感謝します．また，数理計画法に関する有益な助言を頂いた大阪大学基礎工学部石井恵一教授に感謝する次第です．

＜参考文献＞

[1]Allen, J. F. : Maintaining Knowledge about Temporal Interval, Communication of the ACM, Vol.26, No.11, pp832-843 (1983)

[2]Kowalski, R. and Sergot, M. : A Logic-based Calculus of Events, New Generation Computing 4, pp. 67-95 (1986)

[3]原, 北上, 中島 : 時間概念の表現とデフォルト推論，人工知能学会誌，vol.3, No.2, pp. 86-93 (1988)

[4]刀根薫: 数理計画, 朝倉書店 (1983)

[5]J. F. Allen : Planning using a Temporal World Model, IJCAI-83, pp741-pp747 (1983)