

協調的交通システム

濱田 賢、長谷川 正裕、滝沢 誠
東京電機大学理工学部

現在の鉄道、飛行機等の交通システムでは、集中的に静的に列車、飛行機等の移動体の運行計画が決定されている。運行計画の決定は、複雑であり、頻繁に行えるものではない。また、自然災害、事故等により、運行計画通りに運行できない場合に、運行計画の修正、変更は容易ではない。このために、本論文では、運行計画を分散的に、かつ動的に決定する方式について考察する。また、移動体の移動を、移動空間内のオブジェクトをロックするトランザクションとしてモデル化し、デッドロック問題を検討する。

Cooperating Transportation System

Satoshi Hamada, Masahiro Hasegawa, and Makoto Takizawa
Tokyo Denki University
Ishizaka, Hatoyama, Saitama 350-03, Japan
e-mail {hama, hase, taki}@takilab.k.dendai.ac.jp

This paper discusses how to make a plan of a vehicle to move to a destination in a space. In traditional transportation systems like air plane systems, a plan of whole planes is decided statically. According to the plan, each air plane flies. In this paper, we present a method that a plan is decided semi-dynamically. A moving of end vehicle is modeled as a transaction which holds the space. We also discuss dead lock problem among vehicles.

1. はじめに

現在の鉄道、飛行機等の交通システムでは、集中的に静的に列車、飛行機 [DEEN91, FRA190, POZE89]等の移動体の運行計画が決定されている。運行計画の決定は、複雑であり、頻繁に行えるものではない。また、自然災害、事故等により、運行計画通りに運行できない場合、運行計画の修正、変更は時間がかかり、容易ではない。このために、本論文では、運行計画を分散的、かつ動的に決定する方式について考察する。移動体自体が、移動を行うたびに、移動経路を詳細に決定することは、大規模で複雑な移動空間を持つシステムでは計算量の点から現実的ではない。このために、まず、移動空間を、複数の領域オブジェクトに階層的に分割する。例えば、日本を例にとると、まず、関東、東北、...という領域に分割し、さらに、関東を東京、埼玉、...と分割する。日本、関東、東京といった具合に領域オブジェクトが階層的に構成される。各領域オブジェクト毎に管理者が存在し、管理者は領域内の移動体の状況、例えば、混雑状況、事故状況を把握しているとする。移動体は、はじめに、管理者からの情報を基にして、目的地までの経路の決定を行う。このときに、目的地までの詳細な計画を立てることは、次の点から問題がある。

- (1) 時間の経過とともに、各領域の状況が変化するために、計画通りに移動できるとは限らない。
- (2) 大規模で、複雑な移動空間を持つ場合には、詳細な計画を立てるための計算量が大きい。

このために、本論文では、まず概略的な計画を立てる。概略的な計画とは、より上位の領域オブジェクトについての計画である。例えば、東京から大阪まで移動する場合に、東海道を通るか、中央道を通るかといった計画である。この後、東海道を通るとしたとき、東京から神奈川までの詳細な概略的な経路を決定し、これに従って移動を行う。概略的な計画とは、より上位の領域オブジェクトについての計画である。移動するにつれて、計画をより詳細化していく。詳細な計画とは、より下位の領域オブジェクト

についての計画である。移動体が東京にいるときに、名古屋での詳細な経路を決定しても、その通り移動できるとは限らない。このために、より遠方の領域での計画はより概略的なものとし、より近い領域での計画はより詳細なものとしていく。

まず、第2章では、システムのモデルについて述べる。第3章では、概略的な経路から詳細な経路を求めていくための戦略について述べる。第4章では、計画をすすめていくときの領域オブジェクトの確保方法について述べる。第5章では、領域オブジェクトでの経路決定方式について考える。

2. システム・モデル

交通システム T は、移動体集合 V と移動空間 S から構成される ($T = \langle V, S \rangle$)。移動体は、システム内の移動空間 S を移動し、目的地に達する。列車、飛行機、自動車は、移動体の例である。鉄道の線路、航空路、道路が移動空間である。移動空間 S は、基本要素としての領域オブジェクトから階層的に構成される。

領域オブジェクト o は、入力口と出力口を持つ [図1]。領域オブジェクトは、ある入力口から移動体 v を受け付け、目標とする出力口から移動体 v を出力する。移動体 v が o の入力口から出力口に出力されることにより、 v の移動をモデル化する。

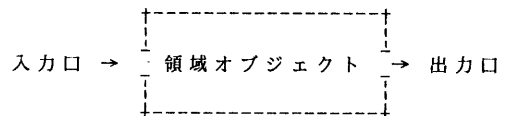


図1. 領域オブジェクト

領域オブジェクト o は、複数の領域オブジェクト $o_1, \dots, o_n$ から構成される [図2]。 o_j を o の下位の領域オブジェクト、 o を o_j の上位の領域オブジェクトとする ($j = 1, \dots, n$)。ここでは、領域オブジェクトは、階層的に構成されていると仮定する。一般的に非巡回有向グラフ (DAG) として構成できるが、問題を簡単にするためと、航空システム等の多くの交通システムでは移動空間は階層的

である[DEEN91]ことから、この仮定を設ける。階層の中で、全ての領域オブジェクトの共通の先祖となるものを根とする。また、下位の領域オブジェクトを持たない領域オブジェクトを葉とする。 o_1 と o_2 を移動空間 S 内の領域オブジェクトとする。 $lca(o_1, o_2)$ を o_1 と o_2 の共通の先祖の領域オブジェクトの中で、一番葉に近い領域オブジェクトとする。

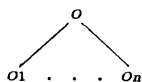


図2. 階層構造

o の下位の領域オブジェクト $o_1, \dots, o_n$ は、互いに結合されている。任意の二つの領域オブジェクト o_j と o_k に対して、 o_j のある出力口 op と o_k の入力口 ip が結合されているとき、 o_j から o_k に結合されているといい、 $o_j \rightarrow o_k$ と書く。入出力口を明確にするためには、 $o_j:op \rightarrow ip:o_k$ とも書く。 $o_j \rightarrow o_k$ ならば、 o_j の出力口から o_k の入力口まで、移動体を届けることができる。 ip と op を、オブジェクト o の入力口と出力口とする。これらは、 o の下位の領域オブジェクト o_j の入力口 ip_j と、 o_k の出力口 op_k に対応するとする。移動体 v が、 ip から op に移動するためには、更に下位層の領域オブジェクトを通過する必要がある。 o 内でどのような下位の領域オブジェクトを移動体が通過するかを決めるのは、 o の役目である。本論文では、上位領域オブジェクトの入出力口は、下位領域オブジェクトの一つの入出力口に対応していると仮定する。

[例] 図3は、領域オブジェクト O が、三つの下位領域オブジェクト A, B, C から構成されている場合を示す。 O の入力口 ip は、下位領域オブジェクト A の入力口 ipa に、出力口 op は、 C の出力口 opc に対応している。移動体 v は、 O の入力口 ip から出力口 op に移動しようとする、 v は O にこれを要求する。 O は、どのような領域オブジェクトを通過するかを決定する。例えば、 A から B に行き、次に C に行く。□

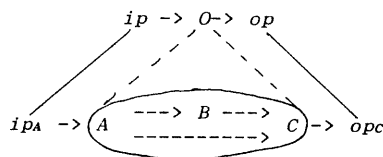


図3. オブジェクトの構成

[定義] 以下の条件を満足する領域オブジェクトの系列 $\langle o_1, \dots, o_m \rangle$ を、同位経路とする。 $o_1, \dots, o_m$ は、同一の領域オブジェクトの一つ下位の領域オブジェクトであり、 $o_1 \rightarrow \dots \rightarrow o_m$ である。□

例えば、図3で、 $\langle A, B, C \rangle$ 、 $\langle B, C \rangle$ は、同位経路の例である。

同位経路 $p = \langle o_1, \dots, o_m \rangle$ を考える。このとき、 p 内の領域オブジェクト o_j は、さらに1つ下位の領域オブジェクトの同位経路 $\langle o_{j1}, \dots, o_{jm} \rangle$ により表せる。ここで、 $(o_j)^1 = \langle o_{j1}, \dots, o_{jm} \rangle$ とする。 $(o_j)^2$ 、 $(o_j)^3, \dots$ も同様に定義できる。 $(o_j)^i$ を、 o_j の i 層詳細化した経路とする。移動空間 S は、有限であるので、 $(o_j)^i = (o_j)^{i+1} = \dots$ なる固定点が存在し、これを $(o_j)^*$ と書く。 $(o_j)^*$ は、最下位の領域オブジェクトのみの経路である。 $i > k$ ならば、 $(o_j)^i$ は $(o_j)^k$ よりも、より下位の領域オブジェクトを含んでおり、より詳細であり、 $(o_j)^k$ は $(o_j)^i$ より概略的であるとする。同位経路 $p = \langle o_1, \dots, o_m \rangle$ に対して、 $q = \langle (o_1)^{h_1}, \dots, (o_m)^{h_m} \rangle$ (ここで、各 $h_j \geq 1$)を経路とする。 p は q より概略的、 q は p より詳細であるという。最詳細な経路は、最下位の領域オブジェクトのみから構成される。移動体は、最終的には、最詳細経路を通過する。ある経路から、より概略的な経路を求めることを概略化とする。逆に、経路から、より詳細な経路を求めることを詳細化とする。

経路 p は q より概略的とする。このとき、 p は、 q よりもより上位の領域オブジェクトを含み、 q は p よりもより下位の領域オブジェクトから構成される。移動空間が階層的に構成されていることから、経路 p の最概略な経路は、一意に定まる。しかし、 p の詳細経路は、一般に複数存在する。即ち、ある地点から他

の地点に起動するための経路が複数ある。従って、詳細化では、複数の詳細経路から、ある条件を満足する一つを決める必要がある。

[定義] 経路 $p = \langle o_1, \dots, o_m \rangle$ を最概略な同位経路とする。より詳細な経路 $\langle (o_1)^{h_1}, \dots, (o_m)^{h_m} \rangle$ で、 $h_1 \geq \dots \geq h_m \geq 1$ であるものを、正規経路とする。□

正規経路は、出発領域オブジェクトから目的領域オブジェクトに近くなるにつれてより概略的となり、出発領域オブジェクトに近いところでは、より詳細である経路である。

[例] 図4に示す移動空間 S を考える。 S は根であり、 A, B, C が S の下位の領域オブジェクトである。ここで、経路 $p_1 = \langle a, B, C \rangle$ は、正規経路である。また、 $p_2 = \langle a_1, a_2, d, e, C \rangle$ も正規経路である。ここで、 $(a)^* = \langle a_1, a_2 \rangle$ で、 $(B)^1 = \langle d, e \rangle$ ならば、 p_1 は p_2 よりも概略的である。 p_1 から p_2 を得ることが詳細化である。□

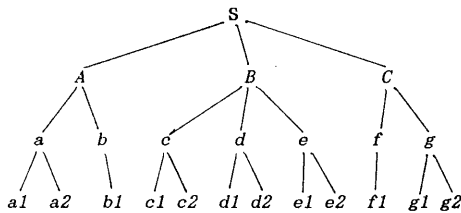


図4. 正規経路

[定義] 移動空間 S において、領域オブジェクト s と d の $\text{lca}(s, d)$ を o とする。 o の下位の領域オブジェクトの中で、 s の先祖を o_s とし、 d の先祖を o_d とする。このとき、 o_s から o_d までの同位経路を、 s から d までの最概略経路 $\text{mbp}(s, d)$ とする。□

s の o_s と d の o_d は、一意に定まるが、 $\text{mbp}(s, d)$ は、一意に定まるとは限らない。 o_s から o_d までに、複数の同位経路が存在する場合があるからである。

3. 経路決定戦略

次に、移動体の目的値までの経路決定について考える。一般に、出発地から目的地までの可能な経路の数は複数あり、この中から最適なものを選択するのは、計算量の点から困難である。また、各領域オブジェクトの状況は、経路を決定した時点と、移動体が移動している時点では、異なっている。従って、出発前に、目的地の領域オブジェクトまでの詳細な経路を決定するかわりに、概略的な正規経路をまず決定し、移動しながら、より詳細な経路を決定する戦略を用いる。

[移動経路決定戦略]

- (1) 最概略経路の決定を行う。
- (2) 概略的な経路を詳細化した正規経路を求める。□

移動体 v が、ある領域オブジェクト s から d に移動したいとする。このとき、 v は、以下の手順により最概略経路 mbp を決定する。

[最概略経路決定]

- (1) 移動空間 S において、 s と d の $\text{lca}(s, d)$ をみつける。これを、 o とする。 o の下位の領域オブジェクトの中で、 s の先祖を o_s とし、 d の先祖を o_d とする [図5]。
- (2) 領域オブジェクト o に、 o_s から o_d への同位経路決定を依頼する。 o は、概略的な同位経路 $\text{mbp} = \langle o_s, p_1, \dots, p_m, o_d \rangle$ を決定する。ここで、 $o_s, o_d, p_1, \dots, p_m$ は、 o の下位の同位領域オブジェクトである。 mbp は、 s から d への最概略経路 $\text{mbp}(s, d)$ である。□

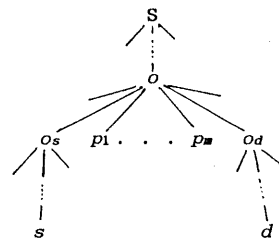


図5. 最概略経路

最概略経路は、出発オブジェクト s から目的オブジェクト d までの経路を示し、上位領域オブジェクトから構成されている。次に、移動体 v は、得られた最概略経路 $mbp = \langle o_1, o_2, \dots, o_m \rangle$ の詳細化を行い、より詳細な経路を得る。

[詳細化]

- (1) $mbp = \langle o_1, \dots, o_m \rangle$ に対して、 o_1 の詳細化 $(o_1)^*$ を行う。これは、 o_1 に依頼する。これにより、 mbp よりも詳細な正規経路 $bp = \langle (o_1)^*, o_2, \dots, o_m \rangle$ が得られる。
- (2) bp の後置の $\langle o_2, \dots, o_m \rangle$ の詳細化も行う。この結果、正規経路 $\langle (o_1)^*, (o_2)^{i_2}, \dots, (o_m)^{i_m} \rangle$ (ここで、 $i_2 \geq \dots \geq i_m \geq 1$) が得られる。どの程度詳細化するかは、以降に述べる。□

この結果得られた経路を $p = \langle o_1, o_2, \dots, o_m \rangle$ とする。ここで、 o_1 は、移動体 v が現在いる最下位の領域オブジェクトである。また、 o_2 は、 v が次に移動する最下位の領域オブジェクトである。移動体 v を、 o_1 から o_2 に移動させる。この結果、経路 $\langle o_2, \dots, o_m \rangle$ が得られる。

移動体 v が、決定された経路により移動しているときに、事故等により予定した経路に従って、移動できないと判断される場合がある。このときには、現在の領域オブジェクトから目的地の領域オブジェクトまでの経路決定を改めて行う。

詳細化の (2) で、概略的経路 $\langle o_1, \dots, o_m \rangle$ の詳細化により、 $\langle (o_1)^*, (o_2)^{i_2}, \dots, (o_m)^{i_m} \rangle$ が得られる。ここで、 $o_2, \dots, o_m$ をどの程度詳細化するかを考える。ここでは以下の方法を用いる。

- (1) 各 o_i について ($i = 2, \dots, m$)、 $(o_j)^* = (o_j)^k$ で、 $(o_j)^* \neq (o_j)^h$ ($h < k$) とする。
- (2) $ij = k * 2^{-(i-1)}$ とする。□

4. 同期方式

移動体は、経路計画をたてる段階で、移動空間の領域オブジェクトに対してロック

を行う。概略的な経路計画に対しては、より弱いロックを行い、より詳細な経路計画に対しては、より強いロックを行う。最終的には、移動体が通過する領域オブジェクトはロックが行われている必要があり、移動体の通過によりロックは解除される。

4.1 領域オブジェクトの状態

各領域オブジェクト o は、前述したように、入力口集合 $is = \{ip_1, \dots, ip_l\}$ と、出力口集合 $os = \{op_1, \dots, op_o\}$ を持つ。移動体 v は、領域オブジェクト o に対して、どの入力口 ip_j から出力口 op_k に移動したいかを要求する。領域オブジェクト o は、要求 $req = \langle ip_j, op_k \rangle$ を受け付ける。

領域オブジェクト o の状態は、各入力口と出力口の組 $\langle ip_j, op_k \rangle$ に対して、与えられる。各入出力組 $t = \langle ip_j, op_k \rangle \in is \times os$ に対して、以下が与えられる。

$maxt(t) = t$ として o 内に存在できる移動体の総数。

$holdt(t) = t$ を通過することを要求している移動体の総数。

$timet(t) = t$ を移動体が通過するときの最小移動時間。これは、 o 内が空であるときの移動時間である。

$maxt(t)$ を、入出力組 t の容量とする。また、 $timet(t)$ を t の移動時間とする。このとき、入出力組 t の混雑度 $cong(t)$ は、以下で与えられる。

$cong(t) = holdt(t) / maxt(t)$ 。

$ptimet(t) = timet(t) / (1 - cong(t))$ 。

$ptimet(t)$ は、移動体が入出力組 t を取った時の移動時間の目安を与える値である。即ち、混雑してれば、時間がかかる。 $cong(t) = 1$ の時は、もう移動体を受け付けられない状態である。すなわち、要求した移動体は待ちになる。このとき、 o に対して、以下を定義する。

$max(o) =$ 各入出力組 $t \in is \times os$ の $maxt(t)$ の総和。

$hold(o) =$ 各入出力組 $t \in is \times os$ の $holdt(t)$ の総和。

$cong(o) = hold(o) / max(o)$

領域オブジェクト o の入出力組の中で、以下のものを最代表的であるとする。

[定義] o の入出力組 t のなかで、 $\max t(t) / \text{time} t(t)$ が最大のものを最代表とする。□

最代表の入出力組は、単位時間内の移動体数、即ちスループットが最大のものである。例えば、いくつかの道路の中で、高速道路のような幹線に対応するものである。

領域オブジェクト o は、下位の領域オブジェクト $o_1, \dots, o_n$ から構成されているとする。このとき、 o と下位の $o_1, \dots, o_n$ の状態変数の関係について考える。 o の各入出力組 t に対して、可能な詳細経路の集合を P_t とする。ここで、経路 p に対して、 $\max(p)$ は、経路内の領域オブジェクトの入出力組の容量の総和とし、これを経路の容量とする。また、 $\text{time}(p)$ を p 内の入出力組の $\text{time} t(t)$ の和とする。これは、経路 p を通過できる最小時間となる。

$\max t(t) = P_t$ 内の各経路の容量の総和。また、 t の移動時間 $\text{time} t(t)$ も、 P_t から決定される。

$\text{time} t(t) = P_t$ 内の各経路 p について $\max(p) \times \text{time}(p)$ の和を、 $\max(p)$ の和で割った加重平均。

これらの値は、 $\max$ と time は、各領域オブジェクト毎に固定であるのでシステムの設計時に計算される。

4.2 ロック方式

領域オブジェクト o が移動体 v からの要求 $\text{req} = \langle ip, op \rangle$ を受け付けたときに、 o は以下の手続きによりロックされる。

[ロック手続き]

- (1) $t = \langle ip, op \rangle$;
- (2) if $\text{hold}(o) < \max(o)$ and $\text{hold}(t) < \max t(t)$, then $\text{hold}(o) = \text{hold}(o) + 1$, and $\text{hold}(t) = \text{hold}(t) + 1$; o の全ての上位の領域オブジェクト o' に対して、 $\text{hold}(o') = \text{hold}(o') + 1$ and $\text{hold}(t') = \text{hold}(t') + 1$;
- else failed.□

ここで、初期時には、 $\text{hold}(o) = \text{hold}(t) = 0$ である。経路 $\langle o \rangle$ が詳細化されるときには、

詳細化された $(o)^1$ 内の下位の領域オブジェクトがロックされる。概略的な経路決定時には、より上位の領域オブジェクトに対してのみロックがなされている。経路が詳細化されるにつれて、より下位の領域オブジェクトがロックされる。

[例] 図6を考える。経路 $\langle o \rangle$ の詳細化した経路として、 $\langle a \rangle$ 、 $\langle b \rangle$ 、 $\langle c \rangle$ の3つがあるとする。さらに、 $\langle a \rangle$ には、 $\langle d \rangle$ と $\langle e \rangle$ の二つの詳細経路があり、 $\langle c \rangle$ には $\langle g \rangle$ と $\langle h \rangle$ の二つがあるとする。最下位の経路 $\langle d \rangle$ 、 $\langle e \rangle$ 、 $\langle f \rangle$ 、 $\langle g \rangle$ 、 $\langle h \rangle$ は、各々容量として、2、3、5、2、1を持つ。次に上位の領域オブジェクトの経路 $\langle a \rangle$ は、下位の $\langle d \rangle$ と $\langle e \rangle$ の容量の和で、 $2 + 3 = 5$ となる。図6で、括弧内の値は、経路の容量を示す。ある移動体 v_1 が概略的経路 $\langle o \rangle$ を決定すると、 $\text{hold}(\langle o \rangle) = 1$ となる。次に、詳細化により、 $\langle a \rangle$ が定まると、 $\text{hold}(\langle a \rangle) = 1$ となる。他の移動体 v_2 が同じく、 $\langle o \rangle$ をとると、 $\text{hold}(\langle o \rangle) = 2$ となる。□

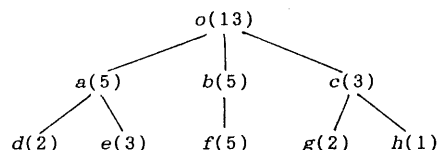


図6. ロック

詳細化を行っている時、ある領域オブジェクトをロックできなかった場合には、以下の二つの方法がある。

- (1) 待つ。
 - (2) 他の経路を見つけるために詳細化をやり直す。
- (1)を行う場合は、現在地から近い領域の詳細化ができなかった場合である。遠い場合には、(2)を行う。(2)では o の上位の領域オブジェクトで詳細化をやり直す。

移動体 v が領域オブジェクト o の入出力組 t を通過した場合には、 o のロックを解放する。

[ロックの解放]

- (1) $\text{hold}(o) = \text{hold}(o) - 1$ and $\text{hold}(t) = \text{hold}(t) - 1$;

- (2) o の上位の領域オブジェクト o' に対して、 $\text{hold}(o') = \text{hold}(o') - 1$ and $\text{hold}(t') = \text{hold}(t') - 1$; □

4.3 デッドロック

移動体の移動を、領域オブジェクトをロックすることとして、一つのトランザクションと考える。移動体が通過するとロックを解放するので、これは二相ロック形式ではない。デッドロック等によりトランザクションをアボートするとき、既に通過した領域オブジェクトは解放される。これにより、他の移動体が領域オブジェクトを利用できる。移動体の移動をアボートすることは、以下のように考えられる。

- (1) これまでに移動してきた経路を、後戻りする。
- (2) 目的地に達するように、他の経路を選択する。

もし、トランザクションを二相ロック形式として、これまで通過してきた領域オブジェクトのロックを解放しないならば、これまで通過してきた経路を後戻りできる。しかし、ロックを解放してしまっているの、通過してきた経路を後戻りできるとは限らない。本論文では、(2)の方法を考える。

問題は、デッドロックである。図7は、デッドロックを示している。移動体 v_1 、 v_2 、 v_3 、 v_4 が、各々領域オブジェクト a 、 b 、 c 、 d をロックしていて、次に、 b 、 c 、 d 、 a をロックしようとしているとする。このとき、デッドロックとなる。システムの状態は、移動体を節点とした有効グラフにより示せる。移動体 v が他の移動体 w がロックしている領域オブジェクトを待つ時、 v から w に有効辺を設ける。これは、データベースシステムにおける待ちグラフと同じである。この待ちグラフ内に巡回閉路が存在する場合に、デッドロックが生じている。デッドロックを解除する方法としては、以下がある。

- (1) ある移動体、例えば、 v_1 を選択して、アボートする。
 - (2) ある移動体、例えば、 v_2 が、 c ではなく、他の領域オブジェクトをロックするように経路変更ができる。
- (1)は、移動体の移動トランザクションが、

二相ロック形式でない場合には、先に述べたように、後戻りできない場合があるために、(2)のみが可能となる。

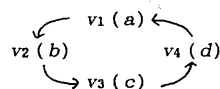


図7. デッドロック

(2)の問題点は、デッドロックを解除するために、一つ以上の移動体の移動がアボートされねばならない場合があることである。

[例]図8を考える。これは、図7の状態で、移動体 v_1 、 v_2 、 v_3 は他の可能な経路がないとする。 v_4 は、可能な経路として、 e を持つとする。ここで、 v_5 が領域オブジェクト d の隣の e のところにいるとする。このとき、デッドロック状態を解消するためには、まず、 v_5 をアボートする。つぎに、 v_4 を e に移動させ、 d のロックを解放させることによりデッドロックを解消できる。

移動空間 S が、移動体により、完全に満たされていれば、このデッドロックは、解決できない。しかし、一つでも空き領域オブジェクトがある場合には、複数の移動体のアボートにより、デッドロックの解消が試みられる。

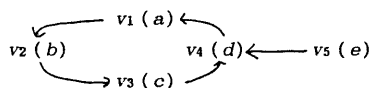


図8. デッドロック

以上みてきたように、デッドロックの解消は、複数の移動体のアボートを行う必要があることが問題となる。このために、デッドロックが生じた場合に、待ちグラフの巡回閉路内の移動体だけをアボートすることにより、解決することを試みる。二相ロック形式のように、これまでに移動してきた全領域オブジェクトのロックを保持していると、移動空間内で移動できる移動体の数を減少させてしまう。このために、各移動体は、最低一つ以前の領域オブジェクト

のロックを保持することにする。これにより、移動体は、現在いる所から、次の二つのアポート方法が可能となる。

- (1) 他の可能な経路を見つける。
- (2) 一つ前に戻る。

この方法により、デッドロックを解消するためにアポートされる移動体の数を減少できる。一般に、以下がいえる。

[n ロック方式] 各移動体は、これまでに通過してきた領域オブジェクトの中で、最近の n 個のロックを解放しないで保持する。□

n が増大するにつれて、デッドロック時にアポートされる移動体数を減少できる。一方、移動空間内の移動体の数が減少してしまう。

5. 領域オブジェクトでの経路決定

領域オブジェクト o が要求 $r = \langle ip, op \rangle$ を受け付けた時を考える。一般に経路は複数存在するので、その中から一番適当な経路を選択しなければならない。その方法は、各領域オブジェクト内の $ptimet$ を重みとしてダイクストラ法を用い、複数の経路のうち $ptimet$ の和が最小な経路を領域オブジェクトにおける経路決定とする。事故等の異常は、最下位の領域オブジェクトにより検出され、上位層に通知される。

6. まとめ

本論文では、移動体が各領域の管理者と協調を行いながら、目的地までの経路を分散して決定していく方式について述べた。本方式では、まずより上位の領域オブジェクトについての経路を概略的経路として決定する。これは、ちょうど、大雑把な経路決定に相当する。次に、移動体の移動とともに、移動体により近い部分についてより詳細な決定を行う。経路の決定は、移動空間内の領域オブジェクトのロックとして、実現される。本論文では、ロックによるデッドロックについての考察も行った。本方式により、予定していた経路をたどることができなくなる緊急事態に対しても、概略

的な経路を詳細化する段階において新しい経路を見つけることにより対処できる。

参考文献

- [DEEN90] Deen, S. M., "Cooperating Agents - A Database Perspective," *Proc. of International Working Conf. on Cooperating Knowledge Base Systems*, 1990.
- [ELLI91] Ellis, C. A., Gibbs, S. J., and Rein, G. L., "Groupware," *Communication of the ACM*, Vol.34, No.1, 1991, pp.38-58.
- [KISH91] 岸本、"分散型協調マルチエージェント型知的通信網," 情報処理学会「B-ISDN時代におけるマルチメディア通信と分散処理シンポジウム」論文集, 1991, pp.37-49.
- [TAKI90] Takizawa, M. and Deen, S. M., "Synchronization Problems of Compensate Operations in the Object-Model," *Proc. of International Working Conf. on Cooperating Knowledge Based Systems*, 1990.
- [TAKI91] Takizawa, M., Hasegawa, M., and Deen, S. M., "Interoperability of Distributed Information System," *Proc. of the International Conf. on Interoperable Database Systems (IMS'91)*, 1991, pp.239-242.
- [STEE87] Steeb, R., Cammarata, S., Hayes-Roth, A. F., Thorndyke, W. P., and Wesson, B. R., "Distributed Intelligence for Air Fleet Control," *Readings in Distributed Artificial Intelligence*, 1987, pp.90-101.
- [FRAI90] Fraichard, T., and Demazeau, Y., "Motion Planning in a Multi-Agent World," *Decentralized A.I.*, 1990, pp.137-153.
- [POZE89] Pozesky, T. M. and Mann, K. M., "The US Air Traffic Control System Architecture," *Proc. of the IEEE*, No.11, 1989, pp.1605-1617.