

移動体トランザクションのデッド ロック問題

濱田 賢、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、列車、飛行機等の移動体の運行計画が集中的に静的に決定されている。運行計画の決定は、考慮すべき要因が多く、複雑で、大きな計算量を必要とする。このために、自然災害、事故等により、計画通りに運行できない場合に、計画の修正、変更は容易ではない。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、距離に応じたロックの強さにより移動経路を確保する方法を示す。また、移動体間でのデッドロック問題を検討する。

Vehicle Transactions

Satoshi Hamada and Makoto Takizawa

Tokyo Denki University

Ishizaka, Hatoyama, Saitama 350-0, Japan

e-mail {hama, taki}@takilab.k.dendai.ac.jp

This paper discusses how to route vehicles to the destinations in a space. In traditional transportation systems like air plane systems, it is decided how to route vehicles before the vehicles depart. According to such a route, each air plane flies. When some accident occurs, it is difficult to reroute the vehicles due to computation complexity. In this paper, a space is structural hierarchically from objects. We present a method by which a route is decided semi-dynamically. A movement of the vehicle is modeled as a transaction which holds objects in the space. We also discuss deadlock problem among vehicles.

1 はじめに

通信ネットワークの経路選択、飛行機等の交通システム、工場等の部品の配送システム等は、ある移動体が空間を移動するシステムとしてモデル化できる。例えば、通信ネットワークの経路選択では、パケット等のデータ単位が移動体となり、ネットワークは移動空間となる。これを、移動体システムとする。こうした移動体システムでは、移動体をどのように有効に目的地に届けるか、即ち移動させるかが問題である。

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^{1,4,13)}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、考慮すべき要因が多く複雑であり、多くの計算量を必要とする。このため、自然災害、事故等により、計画通りに運行できない場合、計画の変更は容易ではない。また、自動車のように移動体自体が、移動を行いながら、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が大きくなる問題点がある。このために、本論文では、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。各オブジェクト毎に管理者が存在し、管理者は領域の状態、例えば、混雑状況、事故状況を把握しているとする。オブジェクト管理者からの情報を基にして、移動体の目的地までの詳細な経路とその通過時間を決めることは、計算量に加えて、オブジェクトの状態が変化することにより、計画通りに移動できるとは限らないという問題がある。

本論文では、まず概略的な経路を決め、移動しながら詳細化していく方法を示す。概略的な経路とは、より上位のオブジェクトについての経路である。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大枠の経路を決める。この後、東海道を通るとしたとき、まず、東京から神奈川までの詳細な経路を決定し、これに従って移動する。東京の移動体に対して、名古屋での詳細な経路を決定しても、名古屋に到着するまでに時間がかかり、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

領域での経路はより概略的なものとし、より近い領域での経路はより詳細なものとしていく。

まず、第2章で、システムのモデルを示す。第3章では、概略的な経路から詳細な経路を求める方法について論じる。第4章では、移動体が移動していくときのオブジェクトの獲得方法とデッドロック問題について述べる。

2 システムモデル

移動体システム 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、 S 内を起点から終点まで移動する。列車、飛行機、自動車は、移動体の例であり、鉄道の線路、航空路、道路が各々の移動空間である。 S は、オブジェクトから構成される。各オブジェクト o は、入力口集合 $IP_o = \{ip_1, \dots, ip_k\} (k \geq 1)$ と出力口集合 $OP_o = \{op_1, \dots, op_h\} (h \geq 1)$ を持つ [図1]。 ip_i から移動体 v を受け付け、 op_j から v を出力することにより、 v の o の移動を示す。 o の ip_i と op_j を、各々、 $ip_i : o$ と $o : op_j$ と書く。また、 $o >$ と $< o$ を、各々 o のある出力口と入力口とする。また、 $< o >$ は、 o のある ip_i から op_j までの経路 $ip_i : o : op_j$ を示し、これを o の基本経路とする。移動体 v が、 $< o$ から $o >$ に移動するためには、 o の要素オブジェクトを通過する必要がある。どの要素オブジェクトを通過するかは、 o がダイクストラ法⁹⁾等により、最適なものを選択する。

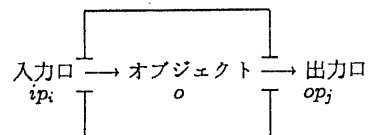


図1: オブジェクト

オブジェクト o は、互いに素な領域を示すオブジェクト $o_1, \dots, o_n$ ($n \geq 0$) から構成される。 o_j を o の要素、 o を o_j の親とする一般的に、 S は非巡回有向グラフ (DAG) として構成できるが、航空システム等の多くの交通システム¹⁾で、移動空間は階層的である。従って、 S は木

構造であり、これを領域木とする。領域木内で、根、葉、先祖、子孫は通常の木と同様に定義される。 $lca(o_i, o_j)$ をオブジェクト o_i と o_j の共通の先祖の中で、一番葉に近いオブジェクトとする。 o のレベル $level(o)$ を、領域木の根から o までの路内のオブジェクト数とする。 $level(o_i) < level(o_j)$ のとき、 o_i は o_j より上位 ($o_i > o_j$) にあるとする。

$level(o_i) = level(o_j)$ のとき、 o_i と o_j は同じレベル ($o_i \equiv o_j$) にあるとする。 $depth(o)$ を、 o から子孫の葉までのオブジェクト数で最大のものとする。各葉が同一のレベルにある領域木を、均衡木とする。本論文では、均衡木を考える。

o_j の出力口 op と o_k の入力口 ip が結合されているとき、 $o_j : op \rightarrow ip : o_k$ 又は $o_j > \rightarrow < o_k$ と書き、 o_j と o_k は接続しているとする。このとき、 o_j から o_k まで、移動体を屈けることができる。ここで、各 $< o$ と $o >$ に対して、 o のある要素 o_i と o_j の1つの $< o_i$ と $o_j >$ が各々対応すると仮定する。あるオブジェクト p に対して、 $p > \rightarrow < o$ のとき $p > \rightarrow < o_i$ 、 $o > \rightarrow < p$ のとき $o_i > \rightarrow < p$ とする。 o_1 から o_n までの経路 $o_1 > \rightarrow < o_2 > \rightarrow \dots \rightarrow < o_n >$ を $< < o_1 >, < o_2 >, \dots, < o_n > >$ と書く。

[例] 図2に関東の一部の領域を示す。四角は市を示し、箱間の辺は接続関係を示す。この領域木を図3に示す。□

[定義] 2つの経路 $p_1 = < < o_1 >, \dots, < o_n > >$ と $p_2 = < < q_1 >, \dots, < q_m > >$ を考える。 $o_1 = q_1 >$ で $< o_n = < q_m >$ のとき、 p_1 と p_2 は、起終点同値 ($p_1 \sim p_2$) である。□

起終点同値な経路では、互いに起点と終点が同一である。ある基本経路 $< o >$ と起終点同値な経路 $p = < < o_1 >, \dots, < o_n > >$ ($p \sim < o >$) を考える。 $o_1, \dots, o_n$ が o の要素であるとき、 p を $< o >^1$ と書く。また、 i レベル拡張 $< o >^i = < < o_1 >^{i-1}, \dots, < o_n >^{i-1} >$ ($i \geq 1$)、 $< o >^0 = < o >$ とする。領域木は有限なので、 $< o >^i = < o >^j$ ($i \leq j$) で、 $< o >^i \neq < o >^j$ ($i > j$) となる固定点 $< o >^i$ が存在し、これを $< o >^*$ と書く。 $< o >^i$ は1つ以上存在するので、 o が移動時間を最小とするものを決定

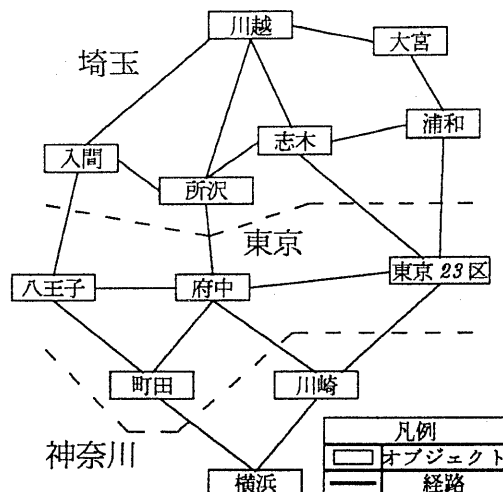


図2: 移動空間 関東

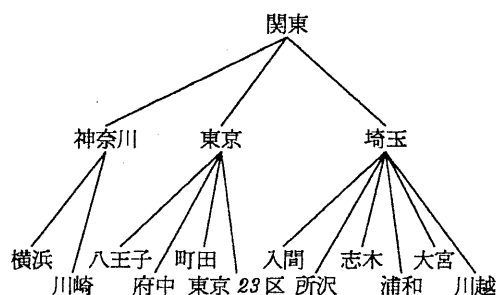


図3: 領域木

する。同じ親を持つオブジェクト $o_1, \dots, o_m$ に対して、 $< < o_1 >, \dots, < o_m > >$ を同位経路とする。

$p_1 (= < < o_1 >, \dots, < o_n > >) \sim p_2$ なる経路 p_1 と p_2 を考える。 $p_2 = < < o_1 >, \dots, < o_{i-1} >, < o_i >^j, < o_{i+1} >, \dots, < o_n > >$ ($j > 1$) ならば、 p_1 は p_2 よりも概略的 (p_2 は p_1 よりも詳細) であり、 $p_1 \ll p_2$ とする。また、 $p_1 \ll p_3 \ll p_2$ なる p_3 が存在するときも、 $p_1 \ll p_2$ とする。経路 p に対して、 $p \ll q$ で $q \ll r$ なる r が存在しない経路 q を p の最概略経路とする。また、 $q \ll p$ で $r \ll q$ なる r の存在しない q を p の最詳細経路とする。最詳細経路は、葉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、この逆を詳細化とする。

p を経路 $\langle\langle o_1 \rangle, \dots, \langle o_m \rangle\rangle$ とする。このとき、 $i > j$ なる各 i と j について、(1) o_i は o_j よりも上位にある($o_i > o_j$)とき、 p を上昇経路とし、(2) $o_i < o_j$ のとき、 p を下降経路とする。ある k と h について、 $o_1 < \dots < o_{h-1} < o_h \equiv \dots \equiv o_k > o_{k+1} > \dots > o_m$ のとき、 p を正規経路とする。ここで、 p の部分系列 $\langle\langle o_1 \rangle, \dots, \langle o_{h-1} \rangle\rangle$ 、 $\langle\langle o_h \rangle, \dots, \langle o_k \rangle\rangle$ 、 $\langle\langle o_{k+1} \rangle, \dots, \langle o_m \rangle\rangle$ を各々 p の上昇部、平坦部、下降部とする。正規経路では、起点から離れるにつれてより概略的となり、終点に近づくにつれてより詳細となる。

[例] 図2に示す移動空間を考える。経路 $p_1 = \langle\langle$ 横浜>, <川崎>, <東京 23 区>, <浦和>>と $p_2 = \langle\langle$ 横浜>, <町田>, <八王子>, <入間>, <川越>>は同位経路である。 $p_3 = \langle\langle$ 横浜>, <東京>, <入間>, <川越>>は正規経路である。ここで、 p_3 は p_2 よりも概略的であり、 p_3 から p_2 を得ることが詳細化である。□

オブジェクト s と d 間の起終点同値な経路 p と q に対して、 $p \ll q$ なる q が存在しない p を s から d への最概略経路とする。ここで、 $\langle\langle o_1 \rangle, \dots, \langle o_{h-1} \rangle, \langle o_h \rangle, \dots, \langle o_k \rangle, \langle o_{k+1} \rangle, \dots, \langle o_m \rangle\rangle$ を、 $mbp(s, d)$ を、 s から d へのある最概略経路とする。最概略経路は、正規経路である。ここで、 $\langle\langle o_h \rangle, \dots, \langle o_k \rangle\rangle$ を平坦部、 o を $lca(s, d)$ とする。このとき、定義から、 $o_{h-1}, \dots, o_k$ は o の要素である。 s と d の先祖で、 o の要素を各々 o_s と o_d とし、最概略起点と最概略終点とする。 $\langle\langle o_s \rangle, \dots, \langle o_d \rangle\rangle$ を s と d 間の最概略同位経路とする。図3で、 $\langle\langle$ 神奈川>, <東京>, <埼玉>>は、川崎と川越間の最概略同位経路である。神奈川は川崎の最概略起点で、埼玉は川越の最概略終点である。

3 経路決定戦略

起点から終点までの最適な詳細経路の決定は、計算量の点から困難な場合がある。また、各オブジェクト o の状況は、経路を決定した時点と、移動体が o に到着した時点では、事故等の予測できない事象により、異なっている場合がある。従っ

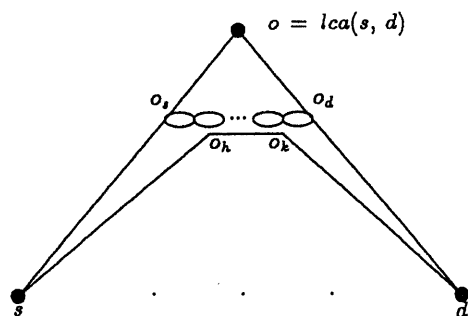


図4: 正規経路

て、出発前に、終点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決定し、移動するに従って、経路を逐次詳細化する戦略を用いる。

移動体 v が、起点 s から終点 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を決定する。

- (1) s と d の $o = lca(s, d)$ を見つけ、最概略起点 o_s と最概略終点 o_d を求める。
- (2) o に、 o_s から o_d への同位経路の決定を依頼する。 o は、最概略同位経路 $p = \langle\langle o_s \rangle, \langle p_1 \rangle, \dots, \langle p_m \rangle, \langle o_d \rangle\rangle$ を決定する。
- (3) s から p_1 までの上昇経路の決定を o_s に、 p_m から d までの下降経路の決定を o_d に、依頼する。これにより $b = \langle\langle o_1 \rangle, \dots, \langle o_n \rangle\rangle$ が得られる。

(1), (2), (3)によって得られた経路 b が最概略経路 $mbp(s, d)$ である[図5]。

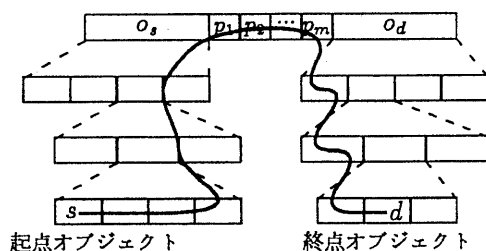


図5: 最概略経路

s から d までの最概略経路 b の詳細化を行う。

[詳細化] b から, 正規経路 $\langle o_1 \rangle^{i_1}, \langle o_2 \rangle^{i_2}, \dots, \langle o_h \rangle^{i_h}, \dots, \langle o_k \rangle^{i_k}, \dots, \langle o_{n-1} \rangle^{i_{n-1}}, \langle o_n \rangle^{i_n}$ (ここで, $i_1 \geq i_2 \geq \dots \geq i_{h-1}, \geq i_h = \dots = i_k < i_{k+1} \leq \dots \leq i_n$) が得られる. 起終点以外を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 $p = \langle p_1, p_2, \dots, p_r \rangle$ と書き直す. ここで, p_1 は移動体 v が現在いる葉オブジェクトで, p_r は終点である. v を p_1 から p_2 に移動させる. この結果, 経路 $\langle p_2, \dots, p_r \rangle$ が得られる. 次々と経路を詳細化しながら, v が移動していく.

概略的経路 $b = \langle o_1, \dots, o_n \rangle$ の詳細化レベル $i_2, \dots, i_{n-1}$ をどのように決めるかを考える. 本論文では, 以下の方法を用いる.

- (1) $h = n/2$ とする. ここで, b 内の各オブジェクト o_j について, $h_j = \text{depth}(o_j)$ とする.
- (2) 各 o_j の詳細化のレベル i_j を以下の式により与える. 図6に, j に対する i_j を示す.

$$i_j = h_j * (1 - (j/n)^\alpha) \quad (\text{for } i \leq h).$$

$$i_j = ((h_j - 1) * (1 - (j/n)^\beta) - 1 + h_j) \quad (\text{for } n > i > h).$$

$$i_n = h_n.$$

本論文では, $\alpha = 2, \beta = 20$ とする. o_h よりも起点に近いオブジェクトをより詳細化し, 遠いオブジェクトはあまり詳細化しない. v が, 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には, 現在のオブジェクトから終点までの経路決定を改めて行う.

4 同期方式

移動体は, 経路の決定時と移動時に, 移動空間 S 内のオブジェクトをロックして獲得する. より概略的と詳細な経路内のオブジェクトに対しては, より弱いと強いロックを各々行う.

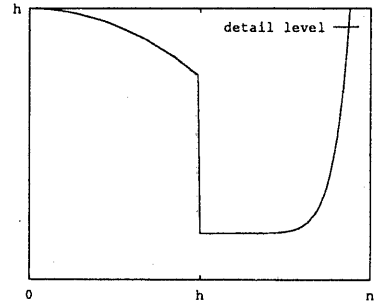


図 6: 詳細化のレベル

4.1 オブジェクトの状態

移動体 v は, オブジェクト o に対して, どの入力口から出力口に移動したいかの要求を行なう. o は, 以下に述べるオブジェクトの状態情報を基にして, 経路の決定を行なう. 本論文では, 各移動体は, 運行速度を自身で制御できないものとする.

各基本経路 t は, 以下の属性値を持つ.

$cap(t)$ = t 内を同時に通過できる移動体の総数.

$hold(t)$ = t の通過を要求 (通過中も含む) している移動体の総数.

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

$cap(t)$ を t の容量, $time(t)$ を t の移動時間とする. このとき, t の混雑度 $cong(t)$ と移動時間 $etime(t)$ を, 以下で与える.

$$cong(t) = hold(t) / cap(t).$$

$$etime(t) = time(t) / (1 - cong(t)).$$

$etime(t)$ は, 移動体の t の通過時間の目安を与える実効移動時間である. $cong(t) = 1$ のとき, t は移動体をこれ以上受け付けられず, t を通過しようとする移動体は待つことになる. ここで, 経路 p に対して, $cap(p)$ と $time(p)$ を各々 p 内の基本経路の容量と移動時間の総和とする.

ここで、 $t_1 \cup t_2$ を、基本経路 t_1 と t_2 から構成される領域とする。ここで、 $cap(t_i \cup t_j) = cap(t_i) + cap(t_j) - cap(t_i \cap t_j)$ である。 o が n 個の基本経路 $t_1, \dots, t_n$ を持つならば、 o を $t_1 \cup \dots \cup t_n$ と書く。また、 $t_1 \cap t_2$ は、 t_1 と t_2 の共通部分を示す。 $cap(t_i \cup t_j) = cap(t_i) + cap(t_j)$ のとき、 t_i と t_j は独立とする。本論文では、各基本経路は独立と仮定する。

o と接続する o_i と o_j に対して、 o_i から o_j に移動するための $o_i \rightarrow \langle o \rangle \rightarrow o_j$ なる基本経路 $\langle o \rangle$ の集合を考える。この集合内の基本経路 t で、 $cap(t) / time(t)$ が最大のものを、 o_i から o_j への最代表基本経路とする。最代表基本経路は、高速道路のような幹線を示している。

o と $o_1, \dots, o_n$ の状態変数の関係について考える。 o の基本経路 t の詳細経路のうち、最代表基本経路によるものの集合を P_t とする。 t の容量と移動時間は要素オブジェクトの状態から以下のように計算される。

$$cap(t) = \sum_{p \in P_t} cap(p) \cdot$$

$$time(t) = (\sum_{p \in P_t} (cap(p) \times time(p))) / \sum_{p \in P_t} cap(p)$$

これらの値は、各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

移動体 v の移動を、オブジェクトのロックを行なうトランザクション³⁾と考える。 $\langle o \rangle^1 = \langle o_1 \rangle, \dots, \langle o_n \rangle$ とする。 $\langle o \rangle$ を移動する v は、 $\langle o_i \rangle$ を移動する部分トランザクション v_i から構成されると考えるこの意味で移動体トランザクションは階層型^{11,12)}である。 v の同期方式を以下に示す。これを半開方式とする。

[同期方式]

- (1) v は $\langle o \rangle$ をロックする。 v が根の時は、最概略同位経路を含む親の基本経路が $\langle o \rangle$ である。
- (2) $v_i, \dots, v_n$ を実行する。即ち、 $\langle o \rangle$ が詳細化される。

- (3) (2) が完了したとき、 $\langle o_1 \rangle, \dots, \langle o_n \rangle$ のロックを解放する。□

$t = \langle o \rangle$ を、以下によりロックする。

[ロック手続き] $hold(t) < cap(t)$ ならば、 $hold(t) = hold(t) + 1$; v が根で、 t が最代表基本経路の場合には、 t を含む上位の基本経路 u に対して、 $hold(u) = hold(u) + 1$ とする。□ ここで、初期時には、 $hold(t) = 0$ である。

概略的な経路決定時には、より上位のオブジェクトに対してのみロックがなされる。経路が詳細化されるにつれて、より下位のオブジェクトがロックされる。

(3) での、 $t = \langle o_i \rangle$ のロックの解放は以下のように行なう。

[ロックの解放] $hold(t) = hold(t) - 1$; t が根の場合には、 t を含む上位の基本経路 u に対して、 $hold(u) = hold(u) - 1$ とする。□

本方式では、 v が $\langle o \rangle$ を通過した後も、 $\langle o_1 \rangle, \dots, \langle o_n \rangle$ のロックは解放されても、 $\langle o \rangle$ のロックは保持される。

[例] 図 7 を考える。 $\langle o \rangle$ の詳細化した経路として、 $\langle a \rangle, \langle b \rangle, \langle c \rangle$ の 3 つがあるとすると。さらに、 $\langle a \rangle$ には $\langle d \rangle$ と $\langle e \rangle$ が、 $\langle c \rangle$ には $\langle g \rangle$ と $\langle h \rangle$ の詳細経路があるとすると。 $\langle d \rangle, \langle e \rangle, \langle f \rangle, \langle g \rangle, \langle h \rangle$ は、各々容量として、2, 3, 5, 2, 1 を持つ。例えば、 $\langle a \rangle$ の容量は、下位の $\langle d \rangle$ と $\langle e \rangle$ のうち最代表基本経路 $\langle e \rangle$ とすると、容量 3 となる。図 11 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、 $hold(\langle o \rangle) = 1$ となる。次に、詳細化により、 $\langle a \rangle$ が定まると、 $hold(\langle a \rangle) = 1$ となる。他の移動体が $\langle o \rangle$ をロックすると、 $hold(\langle o \rangle) = 2$ となる。□

4.3 デッドロック

[例] v_1, v_2, v_3, v_4 が、各々オブジェクト a, b, c, d をロックして、各々 b, c, d, a をロックしようとする、デッドロックとなる。この状態を示す待ちグラフを図 8 に示す。□

移動体がオブジェクトをロックできないときオ

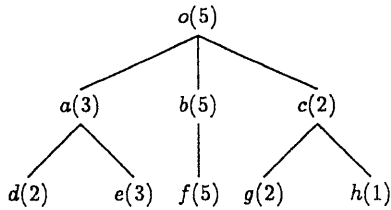


図 7: ロック

プロジェクトを待ち合ってデッドロックとなる場合がある。システムの状態は、移動体を節点とした待ちグラフ^{9,13)}により示せる。移動体 v が、他の移動体 w によりロックされているオブジェクトを待つ時、 v から w に有向辺を設ける。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。

デッドロック閉路内の移動体を選択し、これがロックしているオブジェクトを解放させることにより、デッドロックを解除できる。従来のトランザクション³⁾と異なり、移動体を消去(アボート)できない。このために、 v を $\langle o \rangle$ とは別の経路に移動させて、 $\langle o \rangle$ のロックを解放する必要がある。この移動方法としては、以下の二つがある。

- (1) v がこれまで通過してきた経路を、後戻りさせる。
- (2) v は、新しい経路を見つけて、その経路に従って移動する。

これらを移動体の補償とする。移動体トランザクションが、二相ロック形式でない場合には、(1)の後戻りを行なえない場合がある。 v は、オブジェクト o を通過すると、 o のロックを解放している。従って、 v の通過後、 o は他の移動体によりロックされているかもしれない。(2)の問題点は、デッドロックを解除するために、一つ以上の移動体が、次々に連鎖的に補償され得ることである。

図8の状況で、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 d と隣接する e にいととする。デッドロック状態を解消するためには、まず、 v_5 を補償する。つぎに、 v_4 を e に移

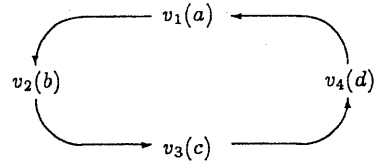


図 8: デッドロック

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。

デッドロックした移動体 v をどのように移動させるかについて考える。前述したように、 v が二相ロック形式ならば、通過してきた全オブジェクトを最下位のレベルでロックしている。従って、必ず後戻りできる。このときは、起点までではなく、どこまで後戻りすべきかが問題となる²¹⁾。しかし、同時に移動できる移動体が減少し、空間の利用効率が低下してしまう。しかし、移動体 v の通過後も下位のオブジェクトのロックは解放するが、上位のオブジェクトのロックは保持している。 v が遠ざかる程、より上位のオブジェクトがロックされる。このため、通過した上位のオブジェクトに対して、戻るための経路決定を依頼できる。

5 おわりに

本論文では、目的地までの経路を動的に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、オブジェクトについての経路を最概略経路として決定する。これは、ちょうど、大雑把に経路を決定することに相当する。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詳細化を行い、より詳細な経路の決定を行う。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階層型トランザクションとしてモデル化し、その同期方式として半開方式を示した。

参考文献

- [1] Deen, S. M.: *Cooperating Agents - A Database Perspective*, *Proc. of Inter-*

- national Working Conf. on Cooperating Knowledge Base Systems*, (1990).
- [2] Deen, S. M., Hamada, S., and Takizawa, M.: *Broad Path Decision in Vehicle System*, *Prof. of DEXA'92*, pp. 8-13 (1992).
- [3] Eswaren, K.P., Gray, J., Lorie, R. A., and Traiger, I. L.: *The Notion of Consistency and Predicate Locks in Database Systems*, *CACM*, Vol.19, No.11, pp. 624-637 (1976).
- [4] Fraichard, T., and Demazeau, Y.: *Motion Planning in a Multi-Agent World, Decentralized A.I.*, North-Holland pp. 137-153 (1990).
- [5] Gasser, L.: *Conceptual Modeling in Distributed Artificial Intelligence*, 人工知能学会誌, Vol.5, No.4, pp. 401-410 (1990).
- [6] 濱田 賢, 長谷川 正裕, 滝沢 誠 : 協調の交通システム, 情報処理学会, マルチメディア通信と分散処理研究会 DPS-52, p. 77-84. (1992).
- [7] 長谷川 豊 : トラフィック制御システムにおける分散処理, 計測と制御, Vol.26, No.1, p. 57-61. (1987).
- [8] 石畑 清 : アルゴリズムとデータ構造, 岩波講座ソフトウェア科学, 岩波書店, Vol.3, (1990).
- [9] 加藤 誠巳 : 都市交通における最適経路情報提供システム, 情報処理 Vol.28, No.3, pp.307-314 (1987).
- [10] 岸本 了造 : 分散型協調マルチエージェント型知の通信網, 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, pp. 37-49 (1991).
- [11] Lynch, N. and Merritt, M.: *Introduction to the Theory of Nested Transactions*, MIT/LCS/TR 367, 1986.
- [12] Moss, J. E.: *Nested Transactions: An Approach to Reliable Distributed Computing*, *The MIT Press Series in Information Systems*, 1985.
- [13] Pozesky, T. M. and Mann, K. M.: *The US Air Traffic Control System Architecture*, *Proc. of the IEEE*, No.11, pp. 1605-1617 (1989).
- [14] 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*, pp. 90-101 (1987).
- [15] Takizawa, M., Hasegawa, M., and Deen, S. M.: *Interoperability of Distributed Information System*, *Proc. of the International Conf. on Interoperable Database Systems (IMS'91)*, pp. 239-242 (1991).
- [16] Takizawa, M. and Deen, S. M.: *Lock Mode Based Resolution of Uncompensatable Deadlock in Compensating Nested Transactions*, *Proc. of the 2nd Far-East Workshop on Future Database Systems, Kyoto*, pp. 168-175 (1992).