協調的交通システム

1E−1

濱田 賢、長谷川 正裕、滝沢 誠
東京電機大学

1. はじめに

現在の交通システムでは、集中的に静的に列車、飛行機[DEEN90]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、変更は、複雑であり、頻繁に行えない。このために、本論文では、運行計画を分散的に動的に決定する方式について考察する。移動体が、移動する度に、目的地までの詳細な計画を立てることは、以下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、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。移動体は、まず概略的な上位の領域オブジェクトについての計画を立て、移動するにつれて、より下位の領域オブジェクトについての詳細化した計画をたてる。

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、移動体の経路決定のための戦略について。第4章では、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 = <V, S>)。移動体は、システム内の移動空間Sを移動する。移動空間Sは、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。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。領域オブジェクトo_jをoの下位の領域オブジェクトとする(j = 1, ...,n)。全ての領域オブジェクトの共通の先祖となるものを根、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。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、$o_j \rightarrow o_k$と書き、o_jからo_kまで、移動体を届けることができる。

[例] 図1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。□

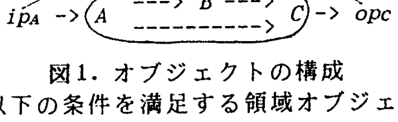

図1. オブジェクトの構成

[定義] 以下の条件を満足する領域オブジェクトの系列<o_1, ..., o_m>を、同位経路とする。o_1, ..., 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、$o_1 \rightarrow \ldots \rightarrow o_m$である。□

同位経路p = <o_1, ..., o_m>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<o_{j1}, ..., o_{jm}>により表せる。ここで、$(o_j)^1$ = <o_{j1}, ..., o_{jm}>とする。$(o_j)^i$を、o_jのi層詳細化した経路とする。$(o_j)^*$を、最下位の領域オブジェクトのみの経路とする。$i > k$ならば、$(o_j)^i$は$(o_j)^k$よりも、詳細であり、$(o_j)^k$は$(o_j)^i$より概略的であるとする。p = <o_1, ..., o_m>に対して、q = <$(o_1)^{h1}$, ..., $(o_m)^{hm}$>(ここで、各$hj \geq 1$)を経路とする。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し、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。

[定義] 経路p = <o_1, ..., o_m>を同位経路とする。より詳細な経路<$(o_1)^{h1}$, ..., $(o_m)^{hm}$>で、$h1 \geq \ldots \geq hm \geq 1$であるものを、正規経路とする。□

[定義] 移動空間Sにおいて、領域オブジェクトsとdの$lca(s, d)$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$mbp(s, d)$とする。□

3. 経路決定戦略

移動体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、動的に変化する。このため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$mbp(s, d)$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$lca(s, d)$をみつけ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2]。
(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、mbp = <o_s, p_1, ..., p_m, o_d>を決定する。

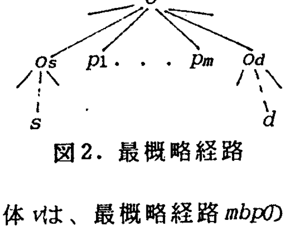

図2. 最概略経路

次に、移動体vは、最概略経路mbpの詳細化を行う。

Cooperating Transportation System
Satoshi HAMADA, Masahiro HASEGAWA,
and Makoto TAKIZAWA
Tokyo Denki University

[詳細化]

(1) mbp 内の、o_1の詳細化を行い、正規経路 $bp = \langle (o_1)^*, o_2, ..., o_m \rangle$が得られる。

(2) bpの後置の$\langle o_2, ..., o_m \rangle$の詳細化も行う。この結果、正規経路$\langle (o_1)^*, (o_2)^{i2}, ..., (o_m)^{im} \rangle (i2 \geqq ... \geqq im \geqq 1)$を得る。□

4. 同期方式

移動体は、経路計画をたてる段階で、領域オブジェクトに対してロックを行う。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、入力口集合$is = \{ip_1, ..., ip_i\}$と、出力口集合$os = \{op_1, ..., op_o\}$を持つ。領域オブジェクトoの状態は、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。

入出力組tの混雑度cong(t)は、以下で与えられる。

congt(t) = holdt(t) / maxt(t)。

ptimet(t) = timet(t) / (1 - congt(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である。oに対して、以下を定義する。

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

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

領域オブジェクトoは、階層的に構成されている。このとき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。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、time(p)をp内の入出力組のtimet(t)の和とする。

maxt(t) = P_t内の各経路の容量の総和。

timet(t)= P_t内の各経路pについてmax(p)×time(p)の和を、max(p)の和で割った加重平均。

maxとtimeは、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$req = \langle ip, op \rangle$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 hold(o) < max(o) and hold(t) < maxt(t), then hold(o) = hold(o) +1, and hold(t) = hold(t) +1; oの全ての上位の領域オブジェクトo'に対して、 hold(o') = hold(o') +1 and hold(t') = hold(t') +1; else failed.□

ここで、初期時には、hold(o) = 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。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。

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。

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。したがって通過してきた経路を後戻りできるとは限らない。問題は、デッドロックである。デッドロックを解除する方法としては、以下がある。

(1) ある移動体を選択して、アボートする。

(2) ある移動体の経路変更を行う。

(1)は、後戻りできない場合があるため、(2)のみが可能となる。(2)の問題点は、一つ以上の移動体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3で、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。ここで、動体v_1、v_2、v_3は他の可能な経路がなく、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、一つでも空き領域オブジェクトを持つ場合には、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。

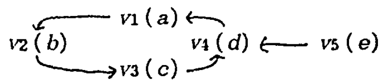

図3. デッドロック

以上より各移動体は、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、次の二つのアボ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ボートされる移動体の数を減少できる。

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$req = \langle ip, op \rangle$ 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、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選択する。

6. まとめ

移動体が各領域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と、ロックによるデッドロックについて述べた。

参考文献

[DEEN90] Deen, S. M., "Cooperating Agents - A Database Perspective," *Proc. of International Working Conf. on Cooperating Knowledge Base Systems*, 1990.