数理解析研究所講究録 1375 巻 2004 年 260-266

移動系における最大個数巡回アルゴリズム

九州工業大学大学院情報工学研究科情報科学専攻　下入佐 真一 (Shinichi Shimoirisa)
Department of Systems Innovation and Infomatics,
Kyushu Institute of Technology
九州産業大学情報科学部社会情報システム学科　朝廣 雄一 (Yuichi Asahiro)
Departmant of Social Information Systems,
Kyushu Sangyo University
九州工業大学情報工学部システム創成情報工学化　宮野 英次 (Eiji Miyano)
Department of Systems Innovation and Infomatics,
Kyushu Institute of Technology

1　はじめに

本稿では，平面上を移動する n 個の物体を巡回する経路選択問題に対して，次のような条件を付けた問題を考える．(i) n 個の物体の平面上での初期座標はあらかじめ与えられいる．(ii) 移動物体は一定の方向と一定の速さで等速直線移動しており，これらについてもあらかじめ与えられている．(iii) 巡回者の数は 1 とする．(iv) 巡回者は一定の速さで直線的に移動可能で，時間 0 で移動方向を換えることができる．例えば，一定の速度で生産ライン上を流れて来る部品を，ロボットアームにより加工する際の最適スケジュールなど，制御ロボットへの応用が考えられる．

移動物体の速さが，巡回者の速さより遅い場合には，全ての移動物体を巡回することが可能であり，できるだけ短い時間ですべての移動物体を巡回するような経路を選択する最小化問題になる．このような問題は，Moving-Target TSP [HRZ03] または Kinetic TSP [HN99] として知られている．この問題は巡回セールスマン問題の自然な拡張問題であり，一般の場合には NP 困難となる．しかし，すべての移動物体が一直線上を移動するような制限をつけた場合には，$O(n^2)$ 時間で最適経路を見つけるアルゴリズムが存在する [HRZ03]．また，すべての移動物体が同じ速さおよび方向で移動しているような場合には，PTAS が存在するが示されている [HN99]．一度に巡回できる移動体の個数を制限した問題の計算時間も報告されている [CM99]．

巡回者よりも速く移動する，または等しい速さで移動する物体がある場合には，移動物体の初期位置，移動方向，経路，もしくは巡回者の可動範囲 (例えば，直線上や半平面上を移動可能など) によっては，訪問できない移動物体が存在する．このような場合には，全ての移動体を巡回可能か否かを判定する問題や，訪問する移動物体の個数をできるだけ多くするような最大化問題，すなわち Prize-Collecting TSP の拡張問題を考えることができる．文献 [AHM2004] では，巡回者の可動方向を 2 方向とした場合についての計算時間が報告されている．

本稿では，巡回者の可動方向を 3 方向 (上下左) および 4 方向 (上下左右) とした場合に対する移動物体巡回問題を考える (図 1 参照)．また，巡回者と n 個の移動物体の速さはすべて等しく，移動物体の可動範囲は平行な 2 直線上に限られているとする．第 2 節では問題の定義と簡単な例を示し，第 3 節では巡回者が 4 方向に移動可能であるときに，巡回個数を最大とするような経路を求める $O(n+T_s(n))$ 時間アルゴリズムを示す．また，第 4 節では巡回者の可動方向が 3 に制限した場合に n 個の移動物体を全て巡回可能か否かを判定する $O(n^3)$ 時間アルゴリズムを示す．

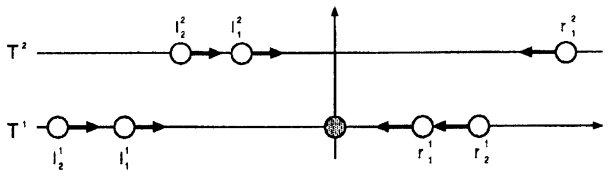

図 1: 今回の問題設定

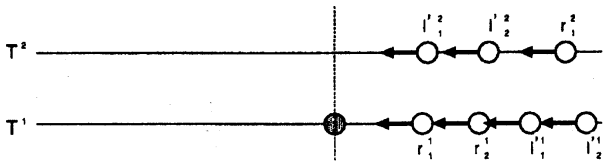

図 2: ロボットの移動方向が縦の動きのみの場合

2 可動方向を限定した場合の移動物体巡回問題

以下では，巡回者をロボット，移動物体をボールとして議論する．移動物体巡回問題は，ロボットとボールの可動方向や速さを限定することにより，様々な問題を定義することができる．今回の問題は以下のように定義される (図 1 参照)．ここで，x の正方向 (負方向) を $E(W)$ 方向，y の正方向 (負方向) を $N(S)$ 方向と呼ぶことにし，例えば，N 方向と E 方向のみに移動する場合，可動方向 D を集合 $D = \{N, E\}$ と表す．

問題の定義： 初期入力として，xy 平面上の $y = 0$ 上または $y = d$ 上に n 個のボール (図 1 で○) が与えられ，すべてのボールは等速 v で E または W 方向のみに移動する．すなわち，各ボール b の可動方向 D_b は，$D_b = \{E, W\}$ である．ロボットは原点を出発して，全てのボールを訪問 (回収) またはできるだけ多くのボールを回収するような経路に従って移動する．ただし，ロボットの可動方向 D_r は，$D_r = \{N, E, S, W\}$ またはその部分集合とし，等速 v で移動するかまたは停止したままボールの到達を待つものとし，時間 0 で移動方向を換えることができるとする．

ロボットとボールの速さが等しいため，初期 x 座標が負のボールは右方向へ，正のボールは左方向へ移動すると仮定でき，回収するボールの集合が決まれば，回収経路も一意に決まる．以下では，ボールが移動する $y = 0$ および $y = d$ の直線をそれぞれ T^1 および T^2 とする．また T^1 上を移動するボールについて，左から来るボールを x 座標の降順に $L^1 = \{l_1^1, l_2^1, l_3^1, \cdots\}$，右から来るボールを x 座標の昇順に $R^1 = \{r_1^1, r_2^1, r_3^1, \cdots\}$ とする．同様に，T^2 上を移動するボールについて，左から来るボールを $L^2 = \{l_1^2, l_2^2, l_3^2, \cdots\}$，右から来るボールを $R^2 = \{r_1^2, r_2^2, r_3^2, \cdots\}$ とする．また，ボールの総数は $n = |L^1 \cup L^2 \cup R^1 \cup R^2|$ とする．それぞれのボールの初期位置の x 座標を $x(l_1^1)$，y 座標を $y(l_1^1)$ などと表す．

簡単な例として，ロボットが 2 方向のみに可動，すなわち $D_r = \{N, S\}$ のときに最大個数を回収するときの回収順序を求めるアルゴリズムを考える (これは第 3 節以降で利用される)．ロボットの可動方向が縦のみの場合には，ボールとロボットとの相対距離だけが重要であるので，図 2 のように，L^* のボールを R^* のボールのある方へ置き換えても同じ問題とみなすことができ，$L^1 \cup L^2 = \emptyset$ としてよい．

定理 1 [AHM2004]. ロボットの可動方向が $D_r = \{N, S\}$ のときに．最大巡回経路を求める $O(n + T_s(n))$ のアルゴリズムが存在する．(ただし $T_s(n)$ はボールの x 座標のソートにかかる時間とする．)

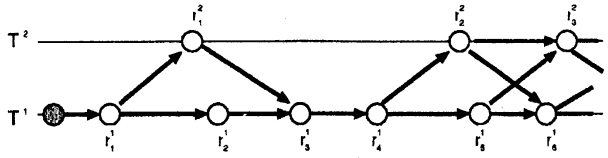

図 3: DAG の生成

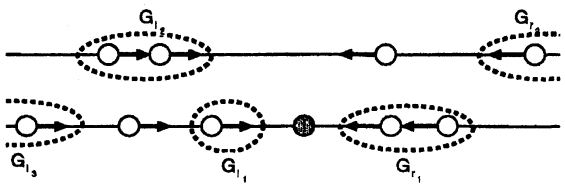

図 4: グループ分け

証明. まずアルゴリズムは準備として，ボールの初期座標から図 3 のような非巡回有向グラフ (DAG) を生成する．このときボールを頂点とし，有向辺はあるボールを取った後，次にどのボールを取るかを示している．ロボットが現在いる直線を T^k，そうでない直線を $T^{k'}$ とする．有向辺は，あるボール r_i^k から $R^1 \cup R^2$ のボールの中で，r_i^k を取った後まだ取れるボールで，最も左 (x 座標の最小) のボールへ出すものとする．よって，入次数および出次数は高々2 となる．例えば，図 3 の場合は，(r_1^1, r_2^1) と (r_1^1, r_1^2) の 2 辺となる．このとき，$|x(r_{i+1}^k) - x(r_j^{k'})| \geq d$ の場合は，r_i^k から $MIN(x(r_{i+1}^k), x(r_j^{k'}))$ の方へ 1 本のみ有向辺が出るものとする．これは，$MIN(x(r_{i+1}^k), x(r_j^{k'}))$ を取った後，$MAX(x(r_{i+1}^k), x(r_j^{k'}))$ を取ることが出来るからである．求めた DAG について，最長経路が回収個数を最大にする回収順序に対応し，そのような経路は $O(n)$ 時間で求めることができる．初期 x 座標をキーとして，ソートをする必要があるため，ソートにかかる時間を $T_s(n)$ とすると，アルゴリズムの計算は $O(n + T_s(n))$ 時間となる．□

3 巡回者の可動方向が 4 方向の場合

本節では，ロボットの可動方向が上下左右，$D_r = \{N, E, S, W\}$ の 4 方向である場合の最大巡回問題を考える．

定理 2. ロボットの可動方向が $D_r = \{N, E, S, W\}$ のときに，最大巡回経路を求める $O(n + T_s(n))$ のアルゴリズムが存在する．(ただし $T_s(n)$ はボールの x 座標のソートにかかる時間とする．)

証明. まず，$L^1 \cup L^2$，$R^1 \cup R^2$ の集合それぞれで最適な取り方 (最大回収のボール集合)OPT_L, OPT_R を求める．これは，定理 1 のアルゴリズムで求めることができる．OPT_L のボールを x 座標の降順に OPT_{l_1}, $OPT_{l_2}, \cdots$, OPT_R のボールを x 座標の昇順に $OPT_{r_1}, OPT_{r_2}, \cdots$ とし，図 4 のようにグループ分けを考える．例えば，$x(l_1^1) > x(l_1^2)$ のとき，左側のボール $L^1 \cup L^2$ は次のようにグループ分けされる ($R^1 \cup R^2$ も同様)．

$G_{l_1} = \{OPT_{l_1}, \cdots, OPT_{l_{i_1}}\}$．ただし，$i_1 = \min\{i \mid y(OPT_{l_i}) \neq y(OPT_{l_{i+1}}), i > 0\}$

$G_{l_2} = \{OPT_{l_{i_1+1}}, \cdots, OPT_{l_{i_2}}\}$．ただし，$i_2 = \min\{i \mid y(OPT_{l_i}) \neq y(OPT_{l_{i+1}}), i > i_1\}$

$\cdots$

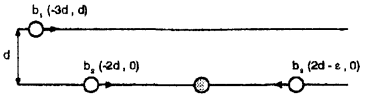

図 5: 複雑になる例 (a)

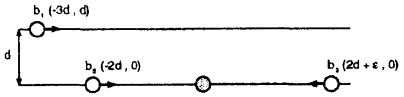

図 6: 複雑になる例 (b)

ロボットとボールの速さが等しいことから次の事実が成り立つ.

事実 1. ロボットが左へ移動してボールを取っているときには $R^1 \cup R^2$ の未回収の部分集合とのロボットの距離は一定であり，ロボットが右へ移動してボールを取っているときには $L^1 \cup L^2$ の未回収の部分集合との距離は一定である.

以下のアルゴリズムにより，$OPT_L \cup OPT_R$ を回収できることを事実 1 より証明することができ，$|OPT| \leq |OPT_L \cup OPT_R|$ より，最大回収経路を求めることができる：

アルゴリズム Max-Four-Directions
ステップ 1： ボールを x 座標をキーとして昇順にソートする.
ステップ 2： $L^1 \cup L^2$，$R^1 \cup R^2$ のそれぞれについて DAG を求める．同時に，グループ分けを行う.
ステップ 3： 回収するボールのグループが無くなるまで以下を実行する.
(3-1) ロボットがいる直線 (T^1 または T^2) の左側に，現在の左側のボールの先頭グループがいる場合は，そのグループを左へ移動しながら取る.
(3-2) ロボットがいる直線の右側に，現在の右側のボールの先頭グループがいる場合は，そのグループを右へ移動しながら取る.
(3-3) 別の直線へ移る.

ステップ 1 のソートにかかる時間を $T_s(n)$ とすると，ロボットが上下左右に可動である場合の最大回収問題は $O(n + T_s(n))$ で計算可能である. □

4 巡回者の可動方向が 3 方向の場合

ロボットの可動方向が $D_r = \{N, S, W\}$ の 3 方向に制限をした場合を考える．直感的には，ロボットの可動方向が少ないほど問題は簡単になるが，前節の 4 方向に可動である場合に比べ，3 方向に制限を強めた方が問題が複雑になる．次のような例を考える (図 5，6 参照). (a) 3 つのボール b_1，b_2，b_3 の初期座標がそれぞれ $(-3d, d)$，$(-2d, 0)$，$(2d - \epsilon)$ $(0 \leq \epsilon \leq 2d)$ の場合には，b_3，b_2，b_1 の順序ですべてのボールを回収可能であるが，b_2 を取るために左に移動すると，b_1 または b_3 のいずれか一方しか回収できなくなる. (b) b_3 の初期座標が $(2d + \epsilon)$ の場合は，b_2，b_1，b_3 の順序で回収したときのみ 3 つのボールを回収可能であり，初期座標の微小な違いにより回収順序が大きく異なる可能性がある.

本節では，ロボットの可動方向が 3 方向の場合に，与えられた n 個のボールすべてを回収可能か否かを判定するアルゴリズムを考える．もし，T^1 上のあるボールと x 座標の値の差が d 未満であるような T^2 上のボールがあるとすると，これらの 2 個両方を回収することは不可能である．x 座標をキーとしてソートした後，任意の i，j に対して，$|x(l_i^1) - x(l_j^2)| < d$，$|x(r_i^1) - x(r_j^2)| < d$ となるような組があるか否かの判定は $O(n)$ 時間でできるので，以下では入力にそのような組は無いとする.

アルゴリズムを設計する前に，いくつかの特別な場合について観察を行う．L^1, L^2, R^1, R^2 のうち，例えば，$L^1, R^1 \neq \emptyset$，$L^2, R^2 = \emptyset$ であるような入力の場合，状況 $C(L^1, R^1)$ などと表す．一般の場合は $C(L^1, L^2, R^1, R^2)$ である.

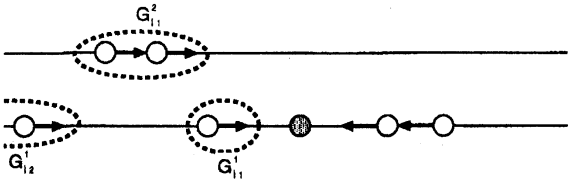

図 7: 状況 $C(L^1, L^2, R^1)$ のときのグループ分け

まず，状況 $C(L^2, R^1)$ に対する全回収可能判定アルゴリズムを考える．r^1_1 の初期位置によって 2 通りの取り方がある：(i) $x(r^1_1) \geq 2d$. 直線 T^2 に移動し左へ移動しながら L^2 のボールを全部取り，その後，T^1 に戻り R^1 のボールを全部取ることができるため，必ず全てのボールを回収可能である．(ii) $x(r^1_1) < 2d$ のとき．動作直後に直線 T^2 へ移ると r^1_1 は取れないので，T^2 に移動する前にロボットは原点で停止して r^1_1 を取らねばならない．ここで，r^1_1 を取った瞬間は，また状況 $C(L^2, R^1)$ となり，同じ事が言える．よって，結局 $x(r^1_{i+1}) - x(r^1_i) \geq 2d$ となる最小の i を s とすると，軸 T^2 に移動する前に，$r^1_1, r^1_2, \cdots, r^1_s$ を取る必要がある．このとき，ボールの初期座標について $x(l^1_1) \leq -x(r^1_i)$ であることを判定することにより全てのボールを回収可能であるか判定できる．これ以降は (i) と同様の場合を考えれば良い．

別の例として，状況 $C(L^2, R^1, R^2)$ に対するアルゴリズムを考える．r^1_1, r^2_1 の初期位置によって取り方が違って来る．(i) $x(r^1_1) > x(r^2_1)$. 直線 T^2 に移動し，左に移動しながら L^2 のボールを全部取ると，状況 $C(R^1, R^2)$ に帰着できる．このとき，第 2 節で述べたロボットの可動方向を $\{N, S\}$ としたときのアルゴリズムを適用することが可能である．(ii-a) $x(r^1_1) < x(r^2_1)$. $r^1_1 \geq 2d$ ならば，直線 T^2 に移動し，左へ移動しながら L^2 のボールを全部取ると，状況 $C(R^1, R^2)$ に帰着できる．(ii-b) $r^1_1 < 2d$ ならば，状況 $C(L^2, R^1)$ と同様に，$x(r^1_{i+1}) - x(r^1_i) \geq 2d$ となる最小の i を s とすると，連続して $r^1_1, \cdots, r^1_s$ を取る必要がある．r^1_s まで取ると，(ii-a) と同じ．ここで，$r^1_1, \cdots, r^1_s$ を取るときは停止していれば良いことに注意する．

上のように，幾つかの特別な場合は，全てのボールを回収可能かを簡単に判定することができる．ここで，全てのボールを回収可能か判定するアルゴリズムを設計する際にキーとなる状況 $C(L^1, L^2, R^1)$ を考える：

状況 $C(L^1, L^2, R^1)$ の場合． ボールを図 7 のようにグループ分けする．例えば，$x(l^1_1) > x(l^2_1)$ の場合，左側のボール L^1, L^2 は次のようにグループ分けされる ($x(l^1_1) < x(l^2_1)$ の場合も同様)．

$G^1_{l_1} = \{l^1_1, \cdots, l^1_{i_1}\}$. ただし, $i_1 = \min\{i \mid x(l^1_i) > x(l^2_1)\}$
$G^2_{l_1} = \{l^2_1, \cdots, l^2_{j_1}\}$. ただし, $j_1 = \min\{j \mid x(l^2_j) > x(l^1_{i_1+1})\}$
$G^1_{l_2} = \{l^1_{i_1+1}, \cdots, l^1_{i_2}\}$. ただし, $i_2 = \min\{i \mid x(l^1_i) > x(l^2_{j_1+1})\}$
$G^2_{l_2} = \{l^2_{j_1+1}, \cdots, l^1_{j_2}\}$. ただし, $j_2 = \min\{j \mid x(l^2_j) > x(l^1_{i_2+1})\}$

以降同様に，$G^1_{l_3}, G^2_{l_3}, \cdots, G^1_{l_m}, G^2_{l_m}$ を考える．このとき，$G^1_{l_m} = \emptyset$ の場合も考えられるが，簡単のため $G^1_{l_m} \neq \emptyset$ として議論する．

$L^1(L^2)$ のボールについては，$T^1(T^2)$ 上を左方向に v で移動して回収，もしくは，$T^1(T^2)$ 上で停止をして $L^1(L^2)$ のボールの到達を待って回収の 2 つの場合が可能である．しかし，ロボットとボールは同じ速さ v で移動し，ロボットは右方向に移動できないため，R^1 のボールを取るためには，直線 T^1 上で停止してボールの到達を待たなければならない．また，次の事実 2 を証明することができるため，直線 T^2 上では常に左方向にロボットは移動するとみなすことができる．

事実 2. 直線 T^2 に止まってボールを取る動作は，直線 T^1 に同じ時間停止後，T^2 に移り，左へ動き続けながらボールを取る動作で模倣できる．

表 1: 往復回数と右側のボールを取る最短時間

T^1-T^2 往復回数 \ R_1	r_1^1	r_2^1	$\cdots$	r_j^1	$\cdots$	
0						
1						
2						
$\cdots$						
m						
$m+1$						

$C(L^1, L^2, R^1)$ に対する回収アルゴリズムの基本戦略は，できるだけ T^1 上に滞在して L^1 および R^1 を取りながら，必要なときにのみ T^2 に移るということを繰り返すというものである．ここで，以下のような値 t_i を定義する．すなわち，$G_{l_i}^k$ を取っている間 (取り始める前，ならびに取り終わった後も含む) に直線 T^1 に停止して R^1 のボールの到達を待つことが可能な時間の最大値である．これは初期 x 座標のみで決まる．

$$t_1 = \mid x(l_1^2) + d \mid,\ t_2 = \mid (x(l_{j_1}^2) + d) - (x(l_{j_1+1}^2) + d) \mid = \mid x(l_{j_1}^2) - x(l_{j_1+1}^2) \mid,$$
$$t_i = \mid x(l_{j_{i-1}}^2) - x(l_{j_{i-1}+1}^2) \mid \quad (2 \le i \le m),\ t_{m+1} = \infty$$

T^1 から T^2 に移動する時刻は，動的計画法に基づく方法で求める．変数 $z[i,j]$ を定義し ($0 \le i \le m+1$, m は L^1 に関するグループの数，$0 \le j \le |R^1|$)，以下の表を考える．$z[i,j]$ には直線 T^1 と T^2 の間を i 回往復して r_j^1 を取る時刻の最小値を格納し，取れない場合は ∞ とする．r_j^1 列のすべての値が ∞ であれば，r_j^1 を回収不可能であることを意味する．表の中で，往復回数と $z[i,j]$ に格納されている時刻が決まれば，ロボットやすべてのボールの位置は一意に定まる．例え，往復回数が定まったとしても r_j^1 の回収可能な時刻には，ある程度の範囲があるが，その範囲中での遅い時刻の取り方は，最速で取った後に左に動いたものと本質的に変わらないため，r_j^1 を取る時刻の最小値のみを考えれば良い．

ここでは全てのボールを回収することを考えているので，r_j^1 の列に関して，往復回数 i というのは，$G_{l_i}^1, G_{l_i}^2$ を取った後で，$G_{l_{i+1}}^1$ を取っている間に一時停止をして r_j^1 を取るということを意味していることに注意する．各セルの値を計算するために，R^1 の連続する 2 個のボール間の相対距離 w_i を考え，その値から a_i および b_i の 2 つ値を定義する．

$$w_1 = x(r_1^1) = 2da_1 + b_1$$
$$w_i = x(r_i^1) - x(r_{i-1}^1) = 2da_i + b_i$$

a_i は上下の往復回数，b_i は停止する時刻を表しており，r_{i-1}^1 の回収時刻から r_i^1 を取るまでに，ロボットは

- a_i 回の軸を往復して，時刻 b_i だけ停止する．
- $a_i - 1$ 回の軸を往復して，時刻 $b_i + 2d$ だけ停止する．
- $a_i - 2$ 回の軸を往復して，時刻 $b_i + 4d$ だけ停止する．
- $\cdots$

という動作を行うことにより，正確に w_i だけの距離を縮められる必要があることになる．

往復回数を $u (0 \le u \le a_1)$ とすると，u 回の往復で r_1^1 が取れるかどうかは，$\Sigma_{i=1}^{u} t_i \ge 2da_1 + b_1 - 2du$ が成立しているかどうかを調べれば分かる．$u > a_1$ のときは，r_1^1 を取ることが出来ない．表の第 $m+1$ 行には，$a_1 > m$ の場合，すなわち $L^1 \cup L^2$ を全て回収後に R^1 を取るときの時刻が格納される．

$z[i,j]$ は，$z[1,j-1],\cdots,z[i,j-1]$ より求まる．上記の r_1^1 を考えたときと同じようにして，各 h について $z[h,j-1]$ の値から，さらに $i-h$ 回の往復で，$2da_i+b_i$ の距離を正確に縮めることが出来るかを，上のように停止時間の和で求めていく．表の 1 つのセルの値は $O(n)$ 個の値から求められ，表は $O(n^2)$ 個あるので，全体としては $O(n^3)$ 時間で表を埋めることが出来る．よって，入力が $C(L^1,L^2,R^1)$ であった場合には，$O(n^3)$ 時間ですべてのボールを回収可能か否かを判定することができる．このことを利用して，以下のような定理が導かれる．

定理 3. ロボットの可動方向が $D_r=\{N,S,W\}$ のときに，全巡回経路を求める $O(n^3)$ のアルゴリズムが存在する．

証明. (L^1 や L^2 のボールを回収しながら) R^1 の幾つかの初期 x 座標が小さいボールを順に回収して，直線 T^2 に移動した状況を考える．これは，T^1 と T^2 を置き換えることで，状況 $C(L^1,L^2,R^1)$ と同じになる．よって，$C(L^1,L^2,R^1)$ に対して考えたアルゴリズムにより，残りの全てのボールを回収することが可能か判定することができる．同様の表の作成，各変数の計算時間も全く同様であり，$O(n^3)$ 時間で動作可能である．□

5 おわりに

移動物体に対する最大個数巡回問題については，これまで可動範囲を 2 方向 (直線) とした場合について議論されてきた．今回は，3 方向，4 方向まで可動である問題を考えた．一般的には，ロボットがボールよりも速く移動可能である場合には，最遠のボールから回収するという貪欲法が効率良く動作し，ロボットがボールよりも遅い場合には，最近傍のボールから回収するというアルゴリズムが効率良く動作する．しかし，3 方向の場合には，可動 3 方向には最速 v，非可動 1 方向には速さ 0 という 2 つの速さを持っていると見なすことができるため，2 つの貪欲法の適用が難しくなっている．今後の課題として，3 方向の場合の最大個数回収アルゴリズムの設計やボールが移動する方向や直線の数をより一般的にした場合の計算時間を考えていきたい．

参考文献

[AHM2004] Y. Asahiro, T. Horiyama, K. Makino, H. Ono, T. Sakuma, and M. Yamashita How to Collect Balls Moving in the Euclidean Plane Proc. Computing: The Australasian Theory Symposium(CATS), pp.1-16 (2004)

[CM99] P.Chalasani and R.Motwani. Approximating capacitated routing and delivery problems. *SIAM J Computing* 28, 2133–2149 (1999)

[HN99] M. Hammar and B. J. Nilsson. Approximation results for kinetic variants of TSP, In *Proc of 26th International Colloquium, Automata, Languages and Programming*, LNCS 1644, 392–401 (1999)

[HRZ03] C.S.Helvig, G.Robins, and A.Zelikovsky. Moving Target TSP and Related Problems, *J of Algorithms*, to appear (2003)