エネルギー制限つきエージェントによるデータ配送問題

三重野琢也 †， 山内由紀子 ††， 来嶋秀治 ††， 山下雅史 ††
† 九州大学 理学部 物理学科 †† 九州大学大学院 システム情報科学研究院

1 はじめに

バッテリー駆動のロボットやエージェントにアルゴリズムを与えて自律的に動かそうとするとき，いかにエネルギーを節約して効率よくエージェントを動かすかが重要な問題となる．エネルギー制限付きエージェントによる数直線上やグラフ上での情報収集やデータ配送の問題は，これまでに研究されている [1]．本稿ではエネルギー制限のあるエージェントによる直線上でのデータ配送問題を定義し，エージェントの最適なスケジュールを与える．配送するメッセージが 1 つでも，各エージェントの初期座標が任意であれば NP 完全であることが既に示されている [2]．本稿では全エージェントも初期座標が同一である特殊な場合を考え，最適なスケジュールを与える．また，各エージェントが初期状況で同じ座標にいる場合に 1 つのデータを配送できる最大の距離を求める $O(n \log n)$ 時間アルゴリズムを提案する．

2 問題定義

$\mathrm{A} = \{a_1, a_2, ..., a_n\}$ を，数直線上に置かれた n 台のエージェント $(n \geq 1)$ の集合とし，各 a_i に対して初期の座標 l_i とバッテリー量 q_i が与えられているものとする．バッテリー量 q_i は，そのエージェント a_i が移動可能な距離に等しい．L はデータ配送のゴールの座標である．k は L まで運ばなければならないメッセージの数であり，初期状況では全てのメッセージの座標は原点であるとする．また，エージェントはメッセージが存在する座標でしかメッセージの受け渡しができない．

定義 1．データ配送問題
入力：A, L, k
出力：k 個のメッセージをゴールまで運べるなら *YES,* そうでなければ *NO* を返す

データ配送問題においてはメッセージを運ぶ座標の最大値は L であり，L より大きい座標まで運ぶことが出来る場合も L までしか考慮しないものとする．
さらに，データを配送することのできる最大の座標を求める問題を以下のように定義する．最大配送距離問題では，L より大きい座標まで運ぶことが出来る場合も考慮する．

定義 2．最大配送距離問題
入力：A, k
出力：k 個のメッセージ全てを運ぶことのできる最大の座標を返す

本稿では簡単のため，いずれの問題も $k = 1$ の場合のみを考える．

3 $k = 1$ で全エージェントが同じ座標にいる場合の考察

3.1 データ配送問題（全エージェントが L にいるとき）

$k = 1$ の場合，各エージェントが無駄な動きをしないと仮定すると，各エージェントはメッセージを受け取るために 0 以上の距離をメッセージのある方向へ向かって動き，その後はバッテリーが無くなるかゴールに到達するまでゴール方向

へ向かって動くものとすることができる．こうすると，エージェントの動く順番を定めるとそれらのエージェントの動きは一意に定まる．ゆえに $k=1$ の場合のデータ配送のスケジュール X を，動く順番にソートされた n 台のエージェントの列（$X=(a_{X_1},a_{X_2},...,a_{X_n})$）で表すことにする．また一般に，$n$ 台のエージェント全員がメッセージの運搬に寄与するとは限らない．つまり，データ配送が途切れてしまう場合や n 台未満でゴールできる場合もある．「データ配送が途切れる」とは，メッセージが L まで到達せずメッセージが存在する座標にどのエージェントも到達できないことを言う．

補題 1. $k=1, \forall i(1 \le i \le n), l_i = L$ のとき，スケジュール $(a_1,a_2,...,a_n)$ でデータ配送が途切れることがないとすると，n 人のエージェントでメッセージを運ぶことのできる座標 S は，$S = 2^n \sum_{i=1}^{n} 2^{-i} q_i - (2^n-1)L$.

Proof. データ配送が途切れないと仮定すると，n 台が動いた後のメッセージの座標 S_n について，以下の漸化式が成り立つ．

$$S_0 = 0, \qquad S_n = S_{n-1} + \{q_n - (L - S_{n-1})\} = 2S_{n-1} + q_n - L \qquad (n \ge 1).$$

これを用いて，一般項 S_n が，$S_n = 2^n \sum_{i=1}^{n} 2^{-i} q_i - (2^n-1)L$ であることを帰納法で示す．$n=1$ のとき，明らかに成立する．$n = l(l \ge 1)$ で成立すると仮定すると

$$S_l = 2^l \sum_{i=1}^{l} 2^{-i} q_i - (2^l - 1)L.$$

よって，漸化式を用いると

$$S_{l+1} = 2S_l + q_{l+1} - L = 2^{l+1} \sum_{i=1}^{l} 2^{-i} q_i - (2^{l+1}-2)L + q_{l+1} - L = 2^{l+1} \sum_{i=1}^{l} 2^{-i} q_i - (2^{l+1}-1)L.$$

ゆえに、任意の n で成立する．従って，$S = S_n = 2^n \sum_{i=1}^{n} 2^{-i} q_i - (2^n-1)L$. □

以下，スケジュール X に対する補題 1 の S を S_X と書くことにする．

定理 1. $k=1, \forall i(1 \le i \le n) l_i = L$ とする．メッセージを運ぶことができる距離が最大となるスケジュールは，エージェントのバッテリー量が非増加順となるスケジュールである．

Proof. まず，データ配送が途中で途切れない場合を背理法で示す．バッテリー量の非増加順で動くときがメッセージを運べる距離が最大ではない，つまり，あるスケジュール $X = (a_{X_1}, a_{X_2}, \dots, a_{X_i}, \dots, a_{X_j}, \dots, a_{X_n})$ (ただし、$q_{X_i} < q_{X_j}$) が存在し，S_X がメッセージを運ぶことができる距離の最大であると仮定する．このとき、スケジュール中の a_{X_i} と a_{X_j} の位置を入れ替えたスケジュールを X' とすると，補題 1 より，

$$S_{X'} - S_X = 2^n(2^{-X_i} q_{X_j} + 2^{-X_j} q_{X_i} - 2^{-X_i} q_{X_i} - 2^{-X_j} q_{X_j}) = 2^n(2^{-X_i} - 2^{-X_j})(q_{X_i} - q_{X_j}) > 0$$

となり，S_X が最大であることに矛盾する．したがって，バッテリー量の非増加順で動くときが最大である．また，スケジュール X でデータ配送が途中で途切れる場合は，途切れる直前までのエージェントについて上と同じ議論を行うことで示すことができる． □

以下では，a_i は q_i について非増加順にソートされている (つまり，$q_1 \ge q_2 \ge ... \ge q_n$ となっている) ものとする．また，補題 1 の式に $L = S = S_n$ を代入すると，このときの S_n は，データ配送が途切れない時にちょうどメッセージを持って戻って来ることができるスタートの座標を表し，$S_n = \sum_{i=1}^{n} 2^{-i} q_i$ となる．従って，データ配送が成功するときは必ず S_n は L 以上となっている．

定理 2. $k=1, \forall i(1 \le i \le n) l_i = L, q_1 \ge q_2 \ge ... \ge q_n$ とする．このときデータ配送が成功するための必要十分条件は，$L \le \sum_{i=1}^{n} 2^{-i} q_i$ である．

Proof. データ配送が成功するとき，$L \le \sum_{i=1}^{n} 2^{-i} q_i$ であることは上の議論から明らか．次に，$L \le \sum_{i=1}^{n} 2^{-i} q_i$ のときにデータ配送が成功することを示す．上の議論より，データ配送が途切れないことが示せれば証明は完了である．$n \ge 2$

のとき，$L \leq \sum_{i=1}^{n} 2^{-i} q_i$ ならばデータ配送は途切れないことを，帰納法で示す ($n=1$ のときの成立は明らか)．証明の為，ダミーのエージェント a_0 ($q_0 = 0$, $l_0 = 0$) が存在し，初期状況でメッセージを持っているものとする．最初に，a_1 が必ず a_0 からメッセージを受信できることを示す．

$$\sum_{i=1}^{n} \frac{q_1}{2^i} \geq \sum_{i=1}^{n} \frac{q_i}{2^i} \geq L \quad \text{より，} \quad q_1 \sum_{i=1}^{n} \frac{1}{2^i} \geq L. \quad \text{ゆえに，} \quad q_1 \geq \frac{2^n}{2^n - 1} L \geq L.$$

よって，a_1 は必ず a_0 からメッセージを受信できる．次に，a_{l-1} が a_{l-2} からメッセージを受信できると仮定したとき，a_l は a_{l-1} からメッセージを受信できることを示す ($l \geq 2$).

$$\sum_{i=1}^{l-1} \frac{q_i}{2^i} + \sum_{i=l}^{n} \frac{q_l}{2^i} \geq \sum_{i=1}^{n} \frac{q_i}{2^i} \geq L \quad \text{より，} \quad q_l \sum_{i=l}^{n} \frac{1}{2^i} \geq L - \sum_{i=1}^{l-1} \frac{q_i}{2^i}.$$

即ち，

$$q_l \geq \frac{2^{n-l+1}}{2^{n-l+1} - 1} \left(L - \sum_{i=1}^{l-1} \frac{q_i}{2^i} \right) \geq L - \sum_{i=1}^{l-1} \frac{q_i}{2^i}.$$

ここで，$\sum_{i=1}^{l-1} 2^{-i} q_i$ は、$l-1$ 番目までのエージェントがメッセージを運んだ座標であり，$L - \sum_{i=1}^{l-1} 2^{-i} q_i$ は a_{l-1} が動き終わった時点でのメッセージとゴールの間の距離を表す．つまり，a_l は必ず a_{l-1} からメッセージを受信することができる．従って，データ配送が途切れることは無い． □

3.2 最大配送距離問題（全エージェントが s にいるとき）

次に，各 l_i が同じ値 s の場合の最大配送距離問題について考える．ここでも a_i は q_i について非増加順にソートされているものとする．この問題は以下の 3 つに場合分けができる．

(i) $s = 0$

(ii) $s \neq 0$ かつ 最適な動きでも s より遠くには運べない

(iii) $s \neq 0$ かつ s より遠くに持っていくことができる

(i) の場合は明らかに最大のバッテリーを持つエージェントが 1 人で運ぶときが最適であり，(ii) の場合は補題 1 より、バッテリー量の非増加順に動かすときが最適である．よって以下では、(iii) の場合についてのみ考察する．

ここで，座標 s 以降でメッセージの運搬に寄与できるエージェントは高々 1 台であるので，座標 s に 2 台以上を残す場合は考えなくて良い．結局，1 台のエージェントを s に残しておき残りの $n-1$ 台でメッセージを s まで (できるだけ s の近くまで) 運ぶという戦略を考えればよいことになる．定理 1 の証明の考え方を用いると，以下の補題 2 が得られる．この補題は，後に示す **Algorithm 1** の正当性を保証する．

補題 2. $k = 1, \forall i (1 \leq i \leq n) l_i = s, q_1 \geq q_2 \geq ... \geq q_n$ とする．$A \backslash \{a_{m-1}\}$ はメッセージを s まで運べず，$A \backslash \{a_m\}$ はメッセージを s まで運べるような $m (\geq 2)$ が存在すると仮定すると，以下の *4* つの命題は真である．

1. $A \backslash \{a_i\}$ がメッセージを s まで運べないとき，$j \leq i$ なる任意の j に対して，$A \backslash \{a_j\}$ はメッセージを s まで運べない．

2. $A \backslash \{a_i\}$ がメッセージを s まで運べるとき，$j \geq i$ なる任意の j に対して，$A \backslash \{a_j\}$ はメッセージを s まで運べる．

3. $i \leq m-1$ なる任意の i に対して，a_i が最後に動くようなスケジュールでメッセージを運べる距離は a_{m-1} が最後に動くようなスケジュールでメッセージを運べる距離以下である．

4. $i \geq m$ なる任意の i に対して，a_i が最後に動くようなスケジュールでメッセージを運べる距離は a_m が最後に動くようなスケジュールでメッセージを運べる距離以下である．

Proof. a_i 以外の $n-1$ 台のエージェントをバッテリー量非増加順で動かした時にメッセージを運べる距離の最大値を L_i と書くことにする．

(1. の証明) $q_i \leq q_j$ であるので，合計バッテリー量は $A\backslash\{a_j\}$ の方が大きくなることはない．よって，$L_j \leq L_i$ となる．

(2. の証明) $q_i \geq q_j$ であるので，合計バッテリー量は $A\backslash\{a_j\}$ の方が小さくなることはない．よって，$L_j \geq L_i$ となる．

(3. の証明) $2 \leq i \leq m-1$ なる任意の i について，a_{i-1} を最後に動かすスケジュールを X，a_i を最後に動かすスケジュールを Y としたとき，$S_X \leq S_Y$ であることを示す．補題 1 より，

$$S_Y - S_X = 2^n(2^{-i+1}q_{i-1} + 2^{-n}q_i) - 2^n(2^{-i+1}q_i + 2^{-n}q_{i-1}) = (2^{n-i+1}-1)(q_{i-1}-q_i) \geq 0.$$

従って，$2 \leq i \leq m-1$ においては，$i = m-1$ のスケジュール Y' のときに運べる距離は最大となる．

(4. の証明) a_i が最後に動くようなスケジュールで n 台でメッセージを運べる距離を $L_{\max}(i)$ とすると，

$$L_{\max}(m) = \max\{s + q_m, L_m\}$$

である．よって，どちらが最大になるかによって場合分けをして考える．

$s + q_m \geq L_m$ の場合，$q_m \geq q_i$ より，

$$L_{\max}(m) = s + q_m \geq s + q_i.$$

また，2. の証明より $L_m \geq L_i$ であるので，

$$L_{\max}(m) = s + q_m \geq L_m \geq L_i.$$

よって，$L_{\max}(m) \geq L_{\max}(i)$.

$L_m \geq s + q_m$ の場合，

$$L_{\max}(m) = L_m \geq s + q_m \geq s + q_i \quad \text{かつ} \quad L_{\max}(m) = L_m \geq L_i.$$

よって，$L_{\max}(m) \geq L_{\max}(i)$. □

補題 2 から，(iii) の場合において最も遠くへ運ぶことができるスケジュールは，a_m が最後に動くようなスケジュールか、a_{m-1} が最後に動くようなスケジュールかのどちらかであることがわかる．これを用いて，最大配送距離問題を $O(n \log n)$ 時間で計算するアルゴリズムを提案する．

3.3 最大配送距離問題（全員が s にいるとき）を計算する $O(n \log n)$ 時間アルゴリズム

全員が同じ座標にいるときにどこまで遠くに運べるかを $O(n \log n)$ 時間で計算できるアルゴリズムを提案する．まず，エージェントのバッテリー量非増加順のソートは $O(n \log n)$ 時間でできる．また補題 2 より，もし $A\backslash\{a_i\}$ がメッセージを s まで運べるならば，i より大きな各 j に対して a_j を最後に動かすスケジュールで運べる距離は必ず a_i を最後に動かすスケジュールで運べる距離以下であることが言える．また，もし $A\backslash\{a_i\}$ がメッセージを s まで運べないならば，i 未満の各 j に対して a_j を最後に動かすスケジュールで運べる距離は必ず a_i を最後に動かすスケジュールで運べる距離以下であることが言える．つまり s まで戻れるか戻れないかを一回比較する度に探索範囲を半分にできるため，二分探索を行える．そうすると最後の添字は必ず $m-1$ か m のどちらかになっているため，それらの比較を行えばよい．また，$q_0 = q_{n+1} = 0$ とする．

以下のアルゴリズムは，$O(n \log n)$ 時間で配送距離の最大値を求められる．よって，以下の定理が言える．

定理 3. $k = 1$ で全てのエージェントの初期座標が同じとき，メッセージを運ぶことができる最大の座標を $O(n \log n)$ 時間で求めることができる．

Algorithm 1 最大配送距離問題を解くアルゴリズム：$DataDel2(A, s)$

$b \leftarrow 1$
$e \leftarrow n$
while $b \leq e$ **do**
  $m \leftarrow \lfloor \frac{b+e}{2} \rfloor$
  if $f(A \backslash \{a_m\}, s) =$ "YES" **then**
    $e \leftarrow m - 1$
  else
    $b \leftarrow m + 1$
  end if
end while
return $\max\{g(A \backslash \{a_{m-1}\}, s, q_{m-1}), g(A \backslash \{a_m\}, s, q_m), g(A \backslash \{a_{m+1}\}, s, q_{m+1})\}$

Algorithm 2 A が s までメッセージを運べるかを判定するアルゴリズム：$f(A, s)$

$n \leftarrow |A|$
$S \leftarrow \sum_{i=1}^{n} 2^{-k} q_k$
if $S \geq s$ **then**
  return "YES"
else
  return "NO"
end if

Algorithm 3 バッテリー q を持つエージェントが座標 s に残るスケジュールで運べる距離を求めるアルゴリズム：$g(A, s, q)$

$n \leftarrow |A|$
$S_0 = 0$
for $j \leftarrow 1$ **to** n **do**
  $S_j = 2S_{j-1} + q_j - s$
  if $S_j < S_{j-1}$ **then**
    $u \leftarrow 2S_{j-1} + q - s$
    return $\max\{u, S_{j-1}\}$
  else if $S_j \geq s$ **then**
    return $\max\{S_j, s + q\}$
  end if
end for
$u \leftarrow 2S_{n-1} + q - s$
return $\max\{u, S_{n-1}\}$

4 まとめ

$k = 1$ で全てのエージェントがゴールにいるときの最適スケジュールと，$k = 1$ で全てのエージェントが同じ座標にいるときにどこまで遠くに運べるかを $O(n \log n)$ で判定するアルゴリズムを示した．今後は $k = 2$，$k = 3$ および一般の k の場合での最適なスケジュールの計算を課題としたい．

参考文献

[1] Julian Anaya, Jérémie Chalopin, Jurek Czyzowicz, Arnaud Labourel, Andrzej, and Yann Vaxès. "Collecting Information by Power-Aware Mobile Agents," In Proceedings of the 26th International Symposium on Distributed Computing 2012, pp.46–60, 2012.

[2] Jérémie Chalopin, Riko Jacob, Matúš Mihalák, and Peter Widmayer. "Data Delivery by Energy-Constrained Mobile Agents on a Line," In Proceedings of the 41st International Colloquium on Automata, Languages, and Programming, pp. 423–434 2014.