5K-08

モバイルロボット群のランダムウォークに基づく分散配置アルゴリズムのシミュレーション評価 *

尾崎大誠 †　　首藤裕一 ‡

法政大学 情報科学部

1　はじめに

グラフ上の頂点間を自律的に移動する移動ロボット（以下、ロボット）がいかなる問題（タスク）を解くことができるか、あるいは、様々なタスクを解くにためにどの程度の時間・空間計算量が必要なのか、といった問いは理論的・工学的に重要であり、多くの研究者によって探求されてきた。各ロボットは共通のアルゴリズムにしたがって自律的に動作し目標を達成する。グラフの分散配置問題は Augustine と Moses Jr[1] によって導入された。グラフの分散配置問題は以下のように定義される。

(i) 実行開始時に k 体のロボットが n 個の頂点をもつグラフ上の任意の頂点に存在する。

(ii) 実行を開始してから有限時間以内にすべてのロボットが相異なる頂点上に存在し、以後、どのロボットも移動しないような状況に到達する。

実用上の応用としては、分散配置問題は、複数のロボットが同じリソースを共有している場合に大きなコストが生じるような問題に応用可能である。たとえば、電気自動車の充電が考えられる。電気自動車は充電するのに数時間かかり、他の自動車の充電を待つ時間は利用者にとって大きなコストとなる。このような場合、自動車を待たせるのではなく、すべての自動車を空いている充電ステーションに移動させる、すなわち分散配置させることが利用者の利益に叶うかもしれない。

2　Molla と Moses Jr のアルゴリズム [2]

分散配置問題に関する研究は主に、決定性アルゴリズムに限定して時間・空間計算量を最小化することを目的にしていた。一方で、Molla と Moses Jr[2] は、乱数を用いることで主に空間計算量を削減することを試みた。彼らが提案した乱択アルゴリズムのうち、特に単純ランダムウォーク（以降、単にランダムウォークと呼ぶ）にもとづくアルゴリズムは、各ロボットは（永続）メモリを 1 ビットしか用いず、空間計算量が小さい。このアルゴリズムでは、各ロボット r は二値変数 $r.\texttt{settled} \in \{0,1\}$ のみを維持しする。$r.\texttt{settled} = 1$ のときロボット r は**配置済み**であるといい、そうでないとき r は**未配置**であるという。また、配置済みのロボットが存在しない頂点を**空**であるという。実行開始時にはすべてのロボットは未配置である。このアルゴリズムは、おおまかにいうと、各ロボットが空の頂点を見つけてそこに配置されるまで独立した単純ランダムウォークを行い、配置された後は二度と移動しないアルゴリズムである。具体的には、アルゴリズムの実行は各時刻 $t = 0, 1, \ldots$ においてロボット群が以下の動作を行うことで進行していく。

1. 各頂点 v について、v に配置済みのロボットが存在しないとき、v にいるロボットのなかからひとつのロボットを等確率に選び、そのロボットを配置済みにする。
2. 各未配置ロボット r は、他のロボットとは独立に、現在地 v の隣接頂点を等確率で選び、選択した隣接頂点に移動する。

本稿では、この Molla と Moses Jr のアルゴリズム [2] を RW1 と呼ぶ。

3　本研究の問い

分散配置アルゴリズム RW1 は各ロボットが配置されるまで独立に単純ランダムウォークを行うものであった。一方で、独立に単純ランダムウォークを行うのではなく、同一頂点に存在するロボットが一緒に単純ランダ

---

* Simulation Evaluation of Randomized Dispersion of Mobile Robots

† Taisei Ozaki, Hosei University

‡ Yuichi Sudo, Hosei University

ムウォークを行うアルゴリズムも自然に発想される。具体的には、各時刻 $t = 0, 1, \ldots$ においてロボット群が以下の動作を行うようなアルゴリズム RW2 を考える。

1. 各頂点 v について、v に配置済みのロボットが存在しないとき、v にいるロボットのなかからひとつのロボットを等確率に選び、そのロボットを配置済みにする。
2. 各頂点 v に滞在する未配置ロボットの集合を $R_{v,t}$ とする。$R_{v,t}$ に属すロボットは v の隣接頂点 u を等確率で選び、全員が v から u に移動する。

RW1 と RW2 を実行したとき、分散配置が達成されるまでの時間の期待値が小さくなるのはどちらだろうか。これらの期待時間はグラフの形状にも依存するし、ロボットの初期配置にも依存する。グラフの形状や初期配置に依らず一方が他方よりも小さくなるということはいえなさそうである。簡単のため、初期配置ですべてのロボットがひとつの頂点 s に位置していると仮定する。まず、極端な例として、完全グラフ K_n 上での実行を考える。このときは RW1 の期待実行時間の方が小さい。RW2 の期待実行時間はクーポンコレクタ問題と同様の解析で $\Theta(n \log n)$ となるが、RW1 の期待実行時間は、簡単な解析により $O(n \log \log n)$ で抑えられるからである。一方で、一般のグラフ G に対して RW2 の期待実行時間はあきらかにランダムウォークの被覆時間 $T(G, s)$ に一致するが、RW1 の期待実行時間の解析は複雑である。RW1 の期待実行時間が $O(T(G, s) \log n)$ で抑えられるということはマルコフの不等式とブールの不等式から簡単に導けるが、実際に $\Omega(T(G, s) \log n)$ となるグラフ G と頂点 s が存在するのかどうかは分からない。

本稿では、この問いを明らかにする端緒として、ランダムウォークの被覆時間が最大となることで知られるロリポップグラフ上で RW1 と RW2 を動作させたときの期待実行時間をシミュレーション実験で比較評価する。

4 実験結果

頂点数 $n/2$ のクリークと頂点数 $n/2$ のラインングラフを連結したロリポップグラフを L_n とする。ライン部分の頂点を $v_1, v_2, \ldots, v_{n/2}$（ただし v_i と v_{i+1} が隣接）で表し、クリーク部分の頂点 v_0 とライン部分の頂点 v_1 が隣接しているものとする。また、v_{-1} を、クリーク部分の v_0 でない任意の頂点であるとする。本シミュレーション実験では、$n = 100, 200, 300, \ldots, 1000$ に対して

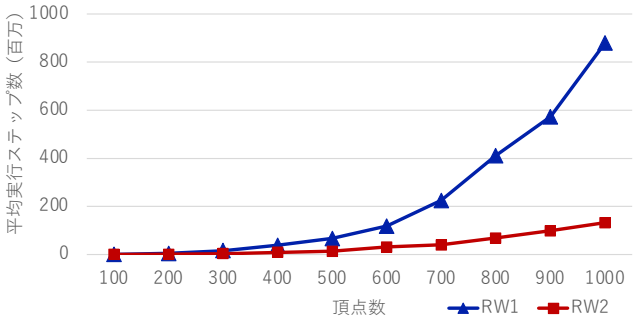

図 1　L_n 上での RW1 と RW2 の平均実行ステップ数

L_n 上で RW1 と RW2 を 100 回ずつ動作させ、その平均実行ステップ数を比較する。ただし、$k = n$ とし、実行開始時には全ロボットが v_{-1} に位置しているものとする。

図 1に実験結果を示す。$n \geq 100$ においては、常に RW1 の実行ステップ数の方が大きい。RW1 と RW2 の平均実行ステップ数をそれぞれ T_1, T_2 とおくと、その比率 T_1/T_2 は n が増えるに従い増加しており、$n = 100$ のとき約 3.77、$n = 1000$ のとき約 6.63 である。$T_2 = O(n^3)$ はよく知られた事実であり、前節で述べた上界 T_1 は $O(n^3 \log n)$ となる。この実験結果は、T_1 がこの上界と一致する可能性を示唆しているように見える。

5 おわりに

本研究では、ランダムウォークに基づく 2 種類の乱択分散配置アルゴリズム RW1 と RW2 をロリポップ上で動作させたときの実行時間をシミュレーションで比較評価した。今後の課題としては、ロリポップグラフ以外の様々なグラフ上での実行時間を比較することや、実行開始時にロボット群が複数の頂点に位置している場合や $k < n$ の場合の実行時間を評価することが挙げられる。

参考文献

[1] J. Augustine and W. K. M. Jr. Dispersion of mobile robots: A study of memory-time trade-offs. In *19th International Conference on Distributed Computing and Networking (ICDCN)*, pages 1–10, 2018.

[2] A. R. Molla and W. K. M. Jr. Dispersion of mobile robots: the power of randomness. In *International Conference on Theory and Applications of Models of Computation*, pages 481–500, 2019.