

# エリアカバレッジタスクにおける回転動作を考慮した移動ロボットの動作計画手法の比較および評価

## 1 はじめに

近年、高い機動性を有するモバイルロボットの活用が検討されてきている。モバイルロボットの活用が期待される例としては、オフィスフロアや工場における室内清掃作業、施設の巡回警備やメンテナンスおよび大規模農場における作物育成補助などが挙げられる。このような活用例では、ロボットは指定された範囲内の全領域を網羅するように移動しながら作業を行うことが求められる。このような作業はエリアカバレッジタスクと呼ばれる。単体ロボットによるエリアカバレッジの場合、作業空間を格子グラフとして考え、そのグラフ上で全域木を構成することで作業空間全体を網羅する巡回経路を得られることが知られている。この巡回経路は経路長という観点では重複なしの最短経路となっているが、必ずしも時間効率の観点で最適とは限らない。そこで本研究では、経路上での回転動作に着目し、格子グラフ上の全域木によって表現される経路の中でタスク完了までの時間が最小となる経路を導出することを目的とする。また、そのために3つの経路計画手法を提案し、その比較および評価を行う

## 2 準備

### 2.1 エリアカバレッジタスク

まず、エリアカバレッジタスクの対象空間について以下の5つの仮定を与える。(1) 対象空間は大きさの等しい正方形セルに分割できる。(2) 正方形セル同士は格子状に整列される。(3) 対象空間は連結であり、任意の2つの地点を結ぶ経路が必ず存在する。(4) 正方形セルは4つの正方形かつ大きさの等しいサブセルに分割できる。(5) サブセルの大きさはロボットが一度にカバー可能な大きさである。これは対象空間がグラフで表現可能であることを表している。また各正方形セルの中心を頂点として考えた全域木から、経路長最短の空間全体を通過する経路を得られることを表している。図1に対象空間の例を示す。

次に、本研究で対象とするロボットの動作について説明する。ロボットが行う動作は、直進移動および移動を

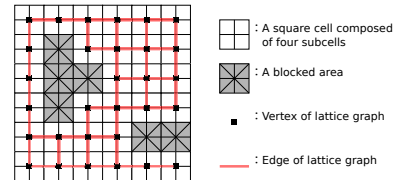


図 1: 対象とする空間の例。

伴わない左右への90度回転のみとする。対象のロボットは一般的な家庭用清掃ロボットを想定し、隣接するサブセル間の直進を直進移動の1単位として考える。また、1単位の直進移動に要する時間に対して1回の90度回転に要する時間が極端に大きいということはないものと仮定する。従って、本研究では最短の走行時間を考えるにあたって、経路長最短かつ回転動作の回数が最小となる経路を考える。

これらの仮定の下で、本研究で取り扱うエリアカバレッジ問題は、エリアカバレッジタスクの対象エリアが与えられた時に、解として全サブセルを被覆するロボットの巡回経路を求める問題として定義される。

本問題における最適解は最小のカバレッジ時間を与えるロボットの巡回経路である。ここでカバレッジ時間とは、エリアカバレッジタスクの完了までに要する時間を表している。ロボットが巡回経路を1周するのに要する時間は、経路上での直進動作の回数および経路上で行った回転動作の回数によって定まる。

### 2.2 問題の定式化

単体ロボットによるエリアカバレッジ問題を、与えられた格子グラフから全域木を得るグラフの問題としてモデル化する。まず利用する定数および変数の記号の定義を表1のように与える。また、定義した記号を利用して問題のモデルを、インスタンス：格子グラフ  $G = (V, E)$ 、解：全域木  $T = (V, E_t)$ 、最適化目標： $\min\{Cost(T)\}$ 、として与える。

問題のインスタンスは、図1に赤線および黒い正方形で示しているように対象エリアの地形データを反映した

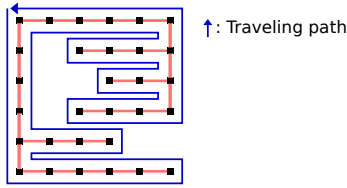


図 2: 図 1 の空間に対する解の例 .

Number of rotation motions					
4	2	2	0	2	4

図 3: 格子グラフ上の頂点の分類 .

格子グラフ  $G$  である . 格子グラフの頂点は対象エリア上の正方形セルに対応している . 辺は正方形セル同士の接続状況を表しており , 通行可能な正方形セル間のみ辺が存在する .

本問題の解はグラフ上の全域木である . 図 2 に示されているように , 全域木からはその木の外周に沿ったロボットの巡回経路を導出することができる . そのためグラフ上で全域木を得ることができれば , 対象エリアの全体をカバーする巡回経路を導出することが可能である .

本問題における最適化目標は , 解として得られた全域木のコストの最小化である . 全域木  $T$  のコストは ,  $T$  から得られる経路上での走行時間に対応している . ここで ,  $T$  は全域木であり , 得られる経路は常に経路長最短の巡回経路である . そのため , 複数の全域木を比較するための指標を考える場合 , 常に一定である経路長ではなく , 走行時間の差を生み出す経路上での回転動作の回数をコストとして用いる . 回転動作の回数はロボットの 90 度回転を 1 単位として算出する . 頂点の次数と辺の接続パターンにより各頂点の周りでの回転の回数が定まり , 全域木全体の回転に起因するコストを得ることができる . 図 3 に

表 1: 定数および変数記号の定義 .

記号	定義
$G$	入力グラフ . $G = (V, E)$
$V$	頂点集合 .
$E$	辺集合 .
$T$	全域木 . $T = (V, E_t), E_t \subseteq E$
$Cost(T)$	$Cost(T) = R_T$
$R_T$	木 $T$ に含まれる頂点を持つ 90 度回転の回数の総和 .

-4/+4	-2/+2	0/0	0/0	0/0	+2/-2	+4/-4

図 4: 辺の追加/除去とコスト変化の関係 .

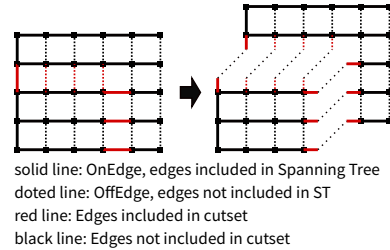


図 5: カットセットの例 .

格子グラフ上の各頂点に置ける回転の回数を示す . 点線が頂点の周囲でのロボットの経路 , 赤色で示している部分が 90 度回転である . また , 図 3 に従って格子グラフ上の一般のグラフについてもグラフのコストを定めることが可能である .

### 2.3 問題の性質

まずはじめに , 辺とコストの関係について説明する . グラフ上の辺を , その辺の追加/除去によるコストの変化によって分類すると図 4 のようになる . この分類によって , 辺の追加/除去によるコストの変化は , 対象の辺と垂直に接続されている辺の本数のみによって決定されていることが分かる .

次に , グラフのカットによるコストの独立性について説明する . 図 5 のように , ある辺の集合によってグラフを切断することを考える . この辺の集合をカットセットと呼ぶこととする . カットセットに含まれる辺の有無を固定すると , 切断されたグラフ間ではコストは独立である .

次に , ロボットの移動方向に基づいた領域分割について説明する . 本研究で対象としている問題では , ロボットの移動方向は , 2 次元平面上での垂直方向または水平方向のいずれかである . 従って , ロボットの移動方向によって図 6 のように 2 種類の領域を考えることが可能であり , また , 方向の違いによって領域を分割することが可能である .

最後に , 領域の分割において , その境界線の候補は , グラフの外周において内角 270 度の点から引かれる直線に限定出来るという性質について説明する . まず , ある方向の領域が , 異なる方向の領域の内部に完全に含まれて

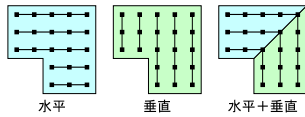


図 6: 領域の水平/垂直分割の例 .

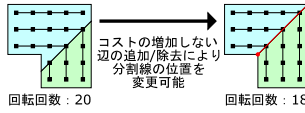


図 7: コストの増加しない変形による境界線の移動 .

いる場合を考える . この場合 , 内部に含まれている領域を外側の領域と同じ方向の領域に置換えるとコストは必ず減少する . 次に , 2 つの領域が単純に隣接している場合を考える . 図 7 のように , 2 つの領域が単純に隣接しており , その境界線が内角 270 度の点から引かれた直線ではない場合 , コストの増加しない辺の追加/除去によって境界線を内角 270 度の点から引かれた直線的位置へと移動させることが可能である .

### 3 提案手法

#### 3.1 動的計画法をベースにした手法

前述したように , カットセットおよびカットセット中の各辺の on/off を決定すると , カットセットによって分割された各グラフのコストはそれぞれ独立である . 従って , あるカットセットによって切り出される 2 つの頂点集合の片方について , コスト最小となる木をもう片方のグラフの形状にかかわらず独立に考えることが可能である .

そこで , 任意の 1 頂点から開始し , 木を維持するように次の頂点を選択して木を成長させていくことを考える . この時 , 成長途中の木と残りのグラフとの境界にカットセットを考え , 各カットセットについて最小コストの木のみを保持するようにする . カットセットの取り方およびカットセット内の辺の有無の組み合わせ数に応じた計算量が必要となるが , 最終的にグラフ全体をカバーするコスト最小の木を得ることができる .

#### 3.2 グラフの分割および領域の種類を用いた手法

2 次元平面上でのロボットの移動方向 , すなわちグラフの辺の接続方向によって頂点を 2 色に塗り分けることを

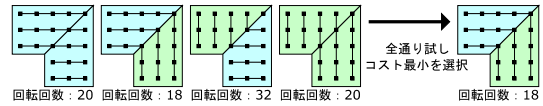


図 8: 領域を分割し , 各領域の方向の組み合わせを全通りチェックする手法の例 .

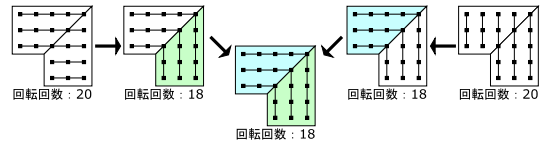


図 9: 最適解における分割との差分を計算する手法の例 .

考える . ただし , 各色の領域の境界線に含まれる頂点は両方の色に塗られている . この時 , 領域の境界線の候補は内角が 270 度の角から斜めに引かれる直線の集合である . これらの直線によって分割された各領域内はそれぞれ単一の色によって塗られる . 従って , これらの各領域について 2 色の色のいずれに塗るかの組み合わせを考え , コストが最小となるものを選択することで最適な塗り分けを得ることができる . 計算量は , 2 の領域数乗である . 本手法によって領域を決定する例を図 8 に示す .

#### 3.3 最適なグラフ分割との差分を計算することにより高速に近似解を得る手法

辺の方向を統一したグラフから , 図 4 に従ってコストが減少するように辺の追加除去を繰り返すことで領域の色を塗り替えていくことを考える . この時 , コストの減少量が最大の塗り替えの中で , 塗り替え面積最小の塗り替え方を考えると , この塗り替えた部分の色は最適に塗り分けられたグラフと同じ色となっている . また , 垂直統一および水平統一の両方の場合からの塗り替えを行った時 , 残りの色が確定していない各領域の色はそれぞれ独立に決定可能であり , さらに各領域内は単一の色で塗られると考えられる . 従って , 垂直統一および水平統一の両方の場合からの塗り替えを行った後 , 未確定の領域についてそれぞれ独立に各色への塗り分けを考え , コストが小さくなる方を選択することでコストの小さい塗り分け方を決定することができる . 本手法によって領域を決定する例を図 9 に示す .

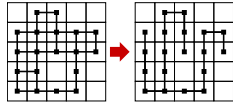


図 10: 小規模な作業空間および最適解 .

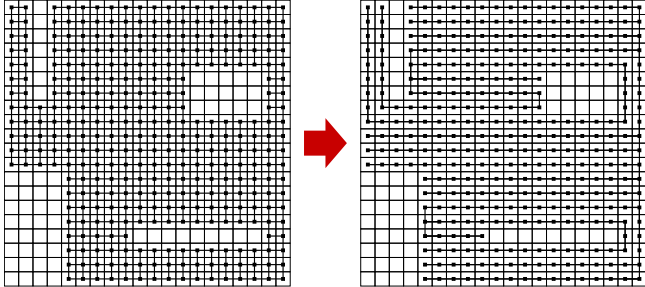


図 11: 大規模かつ単純な構造の作業空間および最適解 .

## 4 シミュレーション・評価

シミュレーションは、頂点数やグラフ構造の異なる 3 つの作業領域に対して行った。3 つの作業領域は、頂点数 19 の小規模な作業領域、頂点数 337 の大規模かつ単純な構造の作業領域、そして頂点数 342 の大規模かつ複雑な構造の作業領域である。

図 10, 図 11, 図 12 の左側に 3 つの作業領域を示す。

まずはじめに、小規模な作業領域に提案した 3 つの手法を適用した。1 つ目の手法では約 69 分で図 10 の右側のように最適解を得ることができた。一方、グラフの分割を用いた 2 つ目および 3 つ目の手法ではそれぞれ約 0.75 秒および 0.58 秒で最適解を得ることができた。

次に、大規模かつ単純な構造の作業領域に対して提案手法を適用した。1 つ目の手法では、現実的な時間内で解を得ることができなかった。2 つ目の手法では、約 66 秒で図 11 の右側のように最適解を得ることができた。3 つ

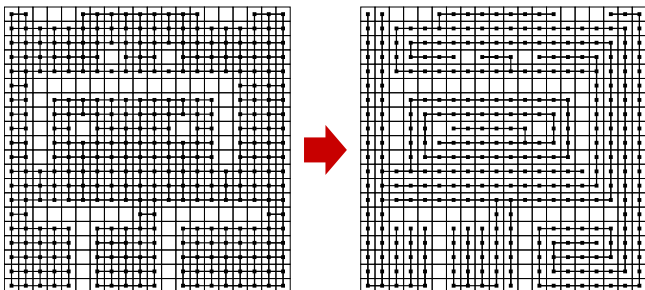


図 12: 大規模かつ複雑な構造の作業空間および効率の良い解の例 .

目の手法では、約 0.69 秒で 2 つ目の手法で得られた解とコストの等しい解を得ることができた。

最後に、大規模かつ複雑な構造の作業領域に対して提案手法を適用した。1 つ目および 2 つ目の手法については、現実的な時間内で最適解を得ることができなかった。3 つ目の手法については、単純な構造の作業領域の場合とほぼ同等の時間で図 12 の右側のように解を得ることができた。

これらの結果より、動的計画法に基づいた手法は、回転動作を考慮したエリアカバレッジ問題に対しては全く実用的ではないことが分かる。ロボットの直径を 35cm と仮定した時、図 10 の作業領域は 3.5m × 3.5m の領域内に収まっているが、これは非常に狭い領域である。1 つ目の手法では、このような非常に狭い領域に対しても最適解を得るために 1 時間以上の時間を要している。

また、2 つ目の手法については、グラフ構造が単純な場合に限定することで高速に最適解を得られることが分かる。ロボットの直径が 35cm の場合、図 11 と図 12 はそれぞれ 14m × 14m の領域に収まっている。また、これらのグラフの頂点数はそれぞれ 337 と 342 とほぼ同等であり、領域の広さについて大きな差は存在しないと考えられる。領域の広さについて大きな差がないにもかかわらず、計算時間については非常に大きな差が存在している。この計算時間の差は、2 つ目の手法の計算時間が作業領域の分割数の指数であるためである。図 11 の作業領域では、領域は 16 個に分割されており、調べるべき領域の種類組み合わせは 65534 通りである。一方、図 12 の作業領域では、領域は 32 個に分割されており、調べるべき領域の種類組み合わせは 4294967 通りである。つまり、2 つの作業領域間で計算時間には 60000 倍の差が存在している。以上のことより、2 つ目の手法は複雑な構造のグラフには向いていないことが分かる。しかしながら、グラフの構造が単純であれば、上記の例より更に大規模な作業空間に対しても高速に最適解を得ることが可能である。

最後に、3 つ目の手法については、異なるサイズや構造のグラフに対して高速に解を得られることが分かる。図 10 および図 11 の領域に対しては、他の手法によって得られた最適解と等しいコストの解を高速に得ることができている。また図 12 の領域に対して得られた解も、領域の形状に対して適したグラフ構造を与えているように観察される。以上のことより、3 つ目の手法は様々なサイズや構造のグラフに対して非常に高速に効率の良い解を得ることが出来るといえる。