

不十分な情報下での緊急度と  
効率的な網羅を考慮した経路探索

筑波大学

図書館情報メディア研究科

2018年3月

中村 優太

# 目次

第1章	はじめに	1
第2章	問題定義	5
第3章	関連研究	7
第4章	提案手法	9
4.1	概要	9
4.2	目標地点決定方法	10
4.3	経路探索手法	11
4.3.1	A*探索アルゴリズム	12
4.3.2	改変 A*探索アルゴリズム	12
4.3.3	ヒューリスティック関数	12
4.4	アルゴリズム	14
第5章	シミュレーション	15
5.1	シミュレーションの設定	15
5.2	シミュレーション結果	15
第6章	まとめと今後の課題	21
	参考文献	23

# 第 1 章

## はじめに

本論文では、不十分な情報下において、各地点に緊急度が設定されている矩形領域をセル分割して作成されたグラフに対して、網羅経路を探索する問題を対象とする。このような問題は、図 1.1 のような自然災害時における空撮経路探索といったアプリケーションを持つ。

実際に、自然災害が発生した状況下では、次のような条件下での経路探索を行う必要がある。

- 被災した領域全てを空撮する
- 各箇所に緊急度が与えられる
- 与えられた緊急度が動的に変更される

ここで緊急度とは、その地点をどれだけ優先して空撮する必要があるかという度合いである。具体的には、何分以内に空撮を行う必要があるという時間情報が与えられる。

この緊急度は、土砂崩れや津波の影響を受けやすい地形などといったその箇所の危険性や、人口密集地やインフラ設備の有無等の重要度によって事前に与えられる。しかし、この緊急度は後から動的に変更される可能性がある。これは被災地において収集される情報が不十分であることによる。被災地では、予想以上・予想外の事態に陥りやすい。被害規模が予想していたよりも大きい場合や、想定していなかった災害が発生した場合、その情報を入手した時点で緊急度は動的に変更されなければならない。緊急度の動的な変更が、出力結果が使用される前であれば再度計算すればよいが、既に使用されている最中に変更される場合も考えられる。自然災害時における空撮経路探索であれば、実際に経路を飛行しながら空撮を行っている最中に、優先して空撮しなければならない箇所が増えるといった状況にあたる。この時、逐次出力された計算結果と最終的な計算結果に大きな齟齬が生じる可能性が考えられる。為、全体の組み合わせ最適の計算結果が必ずしも最適をとるとは限らない。また、動的な変更が行われる度に全体最適を行うことは、計算時間の問題から現実的ではない。

本論文で対象とする問題では、上記のような制限がある中で、網羅までの時間がより短い経路を選択する必要がある。このとき、緊急度が高い箇所から順番に通過するだけでは、図 1.2 の矢印箇所のように、重複して通過することになることになってしまい、非効率である。しかし、この問題を全体の組合せ最適化問題として解こうとした場合、前述した通り緊急度の動的な変更への対応が困難となる。

そこで、残余領域を可能な限りひとまとまりにする方針を考える。残余領域を可能な限りひとまとまりにする利点は 2 つある。

第一に、網羅が完了するまでの経路を効率化し重複を減らすことができる点である。重複を減らすためには経路の途中で残余領域を分断しないことが重要になる。ここで、残余領域は経路通過途中における、まだ通過していない領域のことを指す。残余領域が分断されると、必ずしも 1 回は既に通った箇所を通らなければならない。例えば、図 1.2 に対して、図 1.3 のように回れば、重複して通過する必要がある。

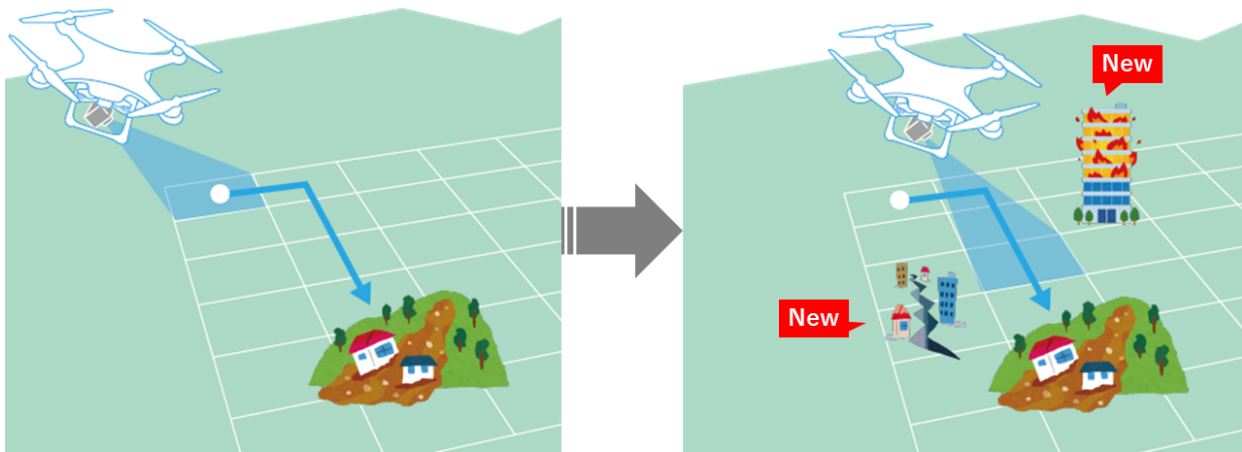


図 1.1 航空機による被災地の空撮

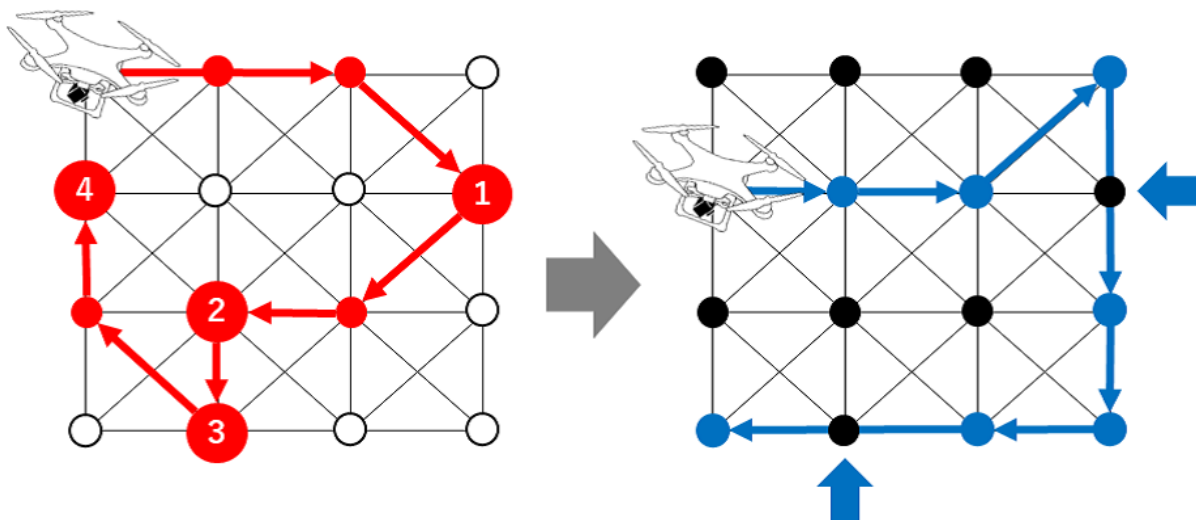


図 1.2 単純に緊急度が高い順番に通過する経路

ないため、効率的だといえる。

第二に、緊急度が動的に変更された場合に経路の効率が極端に悪くなるといった事態を防ぐことができるという点がある。図 1.4 のような緊急度の動的な変更は、場合によっては、極端に非効率な経路を生成してしまう要因になりうる。最悪の場合、度重なる緊急度の動的変更によって、分断された残余領域を往復することが考えられる。残余領域を可能な限りひとまとめにすることにより、緊急度の動的な変更があったとしても、極端に非効率な経路をとることを防ぐことができる。

上記を踏まえて、本研究では残余領域に注目した経路生成手法を提案する。提案手法は、全体最適ではなく、逐次的に目標地点を設定し、改良 A\*探索による経路生成繰り返しを行うことによって網羅経

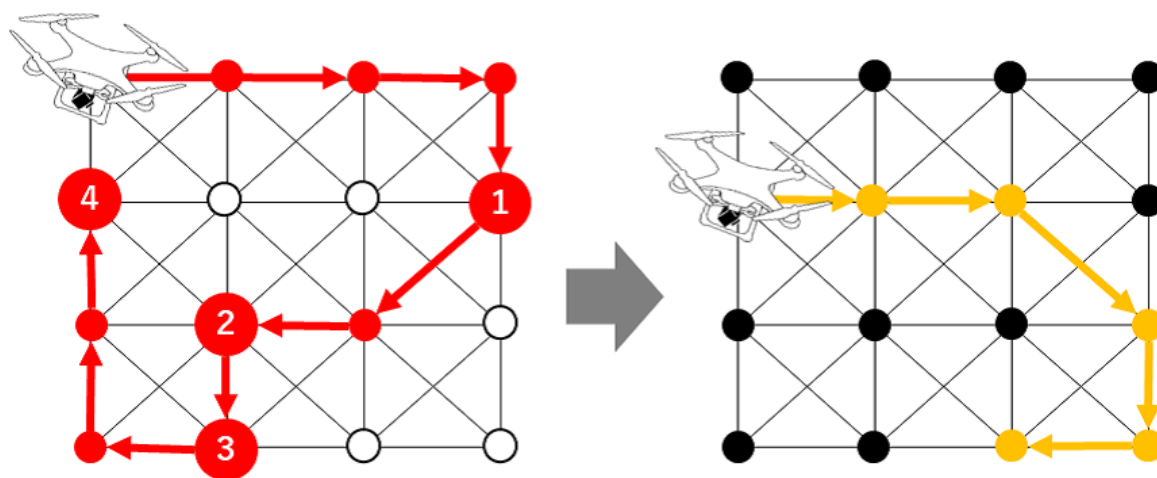


図 1.3 重複を減らすために多少の遠回りをした経路

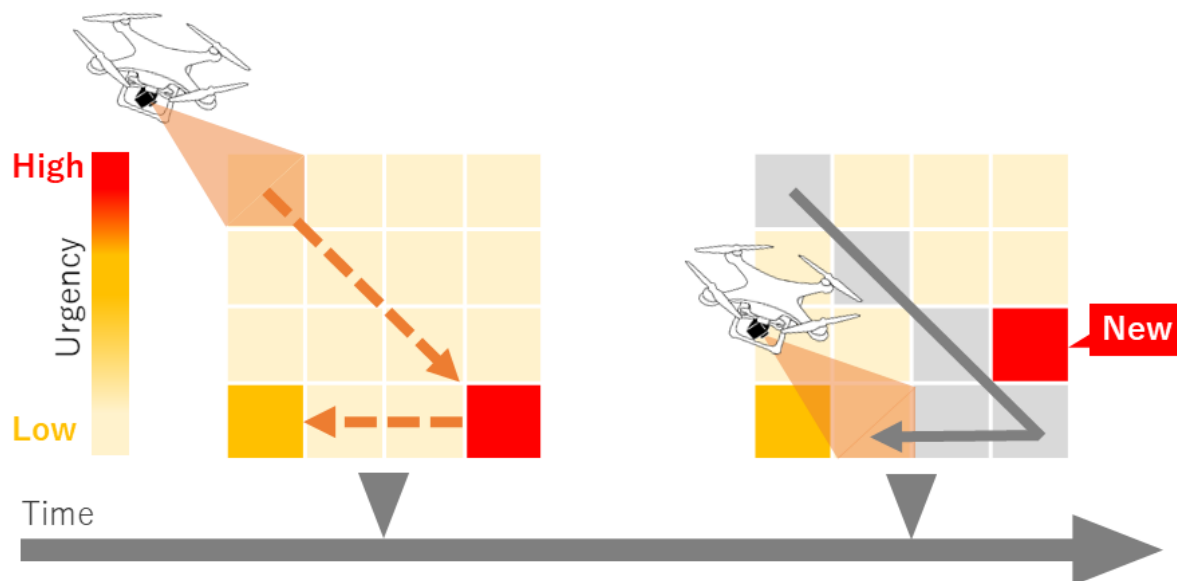


図 1.4 緊急度の動的な変更

路を探索する。この改変 A\*では、残余領域を可能な限り分断しないためのヒューリスティクスを用いて経路探索を行う。このヒューリスティクスによって、重複して通過する回数を減らし、最終的な網羅経路の効率化を図る。同時に、緊急度の動的な変更があった場合、大幅に効率が落ちる可能性を抑える。

本稿では加えて、提案手法を評価するため、単純に締切時間が差し迫っている順番に通過する手法（以下、単純手法）と本手法との比較シミュレーションを行い、実際にどの程度効率化できたのかを確認した。

本研究の貢献は次の2点である。

- 動的な緊急度の変更に対して極端な劣化を防ぎ、効率的な網羅を行う経路探索手法の提案
- 単純手法と提案手法のシミュレーションを行い、締切時間がどれくらい遵守されているかと経路効率を比較

本稿の構成は次の通りである。2節で取り組む問題を定義する。3節で関連研究を整理し、本研究との差異を示す。4節で提案する手法について述べる。5節でシミュレーションとその結果について述べる。6節でまとめを述べる。

## 第 2 章

# 問題定義

本節では、本研究で取り組む問題について定義する。

情報収集を行う領域として矩形セルで構成された多角形領域を想定する。この領域の各セルには、どれくらいまでに撮影しなければならないという締切時刻が秒単位で設定されるものとする。このとき、セルの中央に格子点として頂点がある三角メッシュ状の無向グラフ  $G$  を考える。

本問題で扱うグラフは図 2.1 のようなグラフであり、 $G = (N, E)$  の形で表される。頂点集合  $N = \{n_1, n_2, \dots, n_m\}$  は要素ごとに、セルに対応した締切時刻と高度を持つ。各頂点に対し締切時刻を与える関数を  $Deadline : N \rightarrow Time$  とする。この締切時刻は、後から追加される被災情報によって、動的に変更される可能性がある。頂点に高度を与える関数を  $Height : N \rightarrow \mathbb{R}$ 、座標を与える関数を  $Position : N \rightarrow \mathbb{R}^2$  とする。

辺集合  $E = \{e_1, e_2, \dots, e_k\}$  は要素ごとに長さを持つ。各辺に対し長さを与える関数を  $Length : Edge \rightarrow \mathbb{R}_+$  とする。

被災地を空撮する航空機の飛行経路は、グラフ理論における歩道の形で表す。本問題では、入力としてグラフ  $G = (N, E)$ 、関数  $Deadline$ 、 $Height$ 、 $Length$ 、飛行開始地点  $n_{start}$ 、飛行開始時間  $t_{start}$  が与えられたとき、締切時刻が短い箇所を優先して訪れながらも、早く網羅することのできる長さ  $k$  の歩道  $W_{output} = [n_{start}, n_{i1}, n_{i2}, \dots, n_{k-1}, n_k]$  を求めたい。そこで、締切時刻内に訪れることができたかを評価する関数  $E_u$  と、早く網羅することができたかどうかを評価する関数  $E_c$  の二つの評価関数を用意する。

評価関数  $E_u$  では締切時刻内に訪れることができたかどうかを評価する。もし、締切時刻内に訪れることができなかった場合は、ペナルティが課される。ここで、歩道  $W_{output}$  内において、ある頂点  $n_i$  を初めて訪れた時刻を返す関数を  $FirstVisitTime(n_i, W)$  としたとき、頂点  $n_i$  に対するペナルティ関数  $Penalty(n_i, W)$  を次のように定義する。

$$\begin{aligned} &Penalty(n_i, W) \\ &= \begin{cases} 0 & (FirstVisitTime(n_i, W) \leq Deadline(n_i)) \\ FirstVisitTime(n_i, W) - Deadline(n_i) & (\text{otherwise}) \end{cases} \end{aligned} \quad (2.1)$$

ペナルティ関数  $Penalty(n_i, W)$  を用いて、評価関数  $E_u(W)$  は次のように定義する。

$$E_u(W) = \sum_{i=start}^k Penalty(n_i, W) \quad (2.2)$$

評価関数  $E_c$  では網羅までの時刻を評価する。ここで、飛行経路から航空時間を算出する関数を  $FlightTime(W)$  として、評価関数  $E_c(W)$  は次のように定義する。

$$E_c(W) = FlightTime(W) \quad (2.3)$$

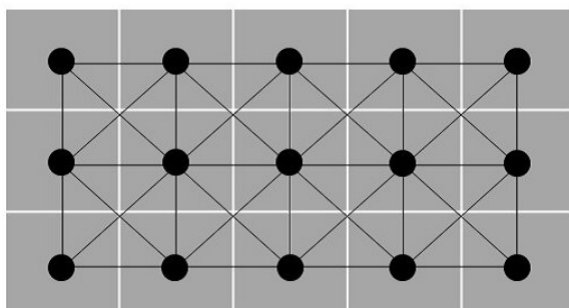


図 2.1 セルに分割された領域へのグラフの配置

よって、本問題は、グラフ  $G = (N, E)$ , 関数  $Deadline, Position, Height, Length$ , 飛行開始地点  $n_{start}$ , 飛行開始時間  $t_{start}$  が入力として与えられたとき,  $V$  中の頂点を全て含み, 第一に評価関数  $E_u$  がより小さく, 第二に評価関数  $E_c$  が小さくなるような長さ  $k$  の歩道  $W_{output} = [n_{start}, n_{i1}, n_{i2}, \dots, n_{ik-1}, n_{ik}]$  を求めるものである.



## 第3章

# 関連研究

本節では、まず本研究で扱う問題に関連のある既存の問題を取り上げ、その差異について述べる。その後、情報収集のための航空機の航空経路生成という点において関連した研究について整理し、それぞれの差異について述べる。

本研究で扱う問題に近いものとして、配送計画問題 (Vehicle Routing Problem : VRP)[1] の中でも時間枠付き運搬経路問題 (Vehicle Routing Problem with Time Windows : VRPTW), および制約条件を緩めた Vehicle Routing Problem with Soft Time Windows (以下, VRPSTW), グラフが動的に変更される Dynamic Vehicle Routing Problem (以下, DVRP) がある。VRP は、特定の施設に待機する運搬車が全ての顧客を巡回し、戻ってくる上で距離の総和が最短のものを求める問題である。VRPTW は、加えて各顧客に Time Window (訪れることが可能な時間帯) が設定されており、その時間帯に必ず訪れなければならない。VRPSTW は、設定された Time Window に遅れてもよいが、その分のペナルティを課すように VRPTW の制約条件を緩めた問題である。これらの問題は NP 困難であること [2] が知られており、これまでにヒューリスティクスによる解法の研究 [3, 4, 5] が多くなされている。これらの解法は全体最適を図っており、計算時間のオーダーは  $O(n^2)$  である。

DVRP では、最初に顧客が設定されて経路が決定されるものの、運搬車が経路を実行し始めると、新しい顧客が追加されるという状況で、新しい顧客に対応して最短経路を探索する問題である [6]。この問題では、複雑な経路探索問題を複数回行うことになってしまうことが多いため、静的な VRP と比較してとることのできる解法に限られる [7]。解法としては、蟻コロニー最適化を動的に拡張したアルゴリズム等 [8, 9] がある。これらのアルゴリズムでは、以前の解を再利用し、動的な変更の予測を立てることで、アルゴリズムの効率化を行っている。

これらの問題に対して、本研究で扱う問題は、動的な時間枠の変更が存在する経路最適化問題である。動的なグラフの変更があるという点で、DVRP と同じであるが、本研究では災害時を想定しているため、発生頻度が乏しく予測も難しい。よって、以前の解を再利用することで動的な変更の予測を立てるといった効率化は行えないため、全体最適によって現実的な時間内に解を求めることは難しい。本研究では、残余領域に注目して逐次的に経路を算出することにより、この問題を解決する。

情報収集のための航空機の航空経路生成については多くの研究がおこなわれているものの、緊急度の高い箇所を優先して回りつつ、網羅性も考慮した経路生成に関する研究は、我々の知る限り初めてである。他の各手法について、次から述べる。

航空機の航空経路生成の代表的な手法としてまず、HAWK[10] が挙げられる。この手法では、要救助者を発見するための経路生成を行うが、このとき経路生成に空間充填曲線であるムーア曲線を用いることで、位置特定の精度への対応と網羅性の確保を行っている。生成される経路は、要救助者がいる可能性の低い箇所は大雑把に、高い箇所は細かく回るような経路となる。対して、本研究では重要な箇所を

細かく回るのではなく、先に回るような経路を生成する。

Liu らの研究 [11] では、情報収集を行う領域を 2D-tree で分割し、分割された場所の中で重要度が高い箇所を多く通る経路を生成する手法を提案している。この手法は、情報収集を行う領域に対してあらかじめ特定のパターンで重要度が設定されているという前提がある。この前提の上で、重要度の高い頂点を多く持つ区域を複数回通過するような経路を生成する手法を提案している。対して、本研究では、重要な場所を何回も通るような経路ではなく、重要な場所を早く通るような経路の生成を行う。また、本手法では情報収集を行う領域に対しての上記のような前提は持たない。

Suijit らの研究 [12] では、情報収集を行う領域に対し優先度を考慮した経路設定を行う方法を提案している。この手法においても、あらかじめ情報収集を行う領域に重み付けを行う。この重み付けされた場所同士の隣接関係をグラフとみなして経路制御を行う。

この手法では、優先度の高い箇所を多く含む経路を生成しているが、情報収集を行う領域を網羅的に回るわけではない。対して、本研究では情報収集を行う領域の優先度のみでなく網羅性も考慮した経路生成を行う。

## 第 4 章

# 提案手法

本節ではまず、提案手法の概要を述べる。次に、利用する改変 A\*探索アルゴリズムの説明を行う。その後、本手法におけるヒューリスティクスについて述べる。最後に、提案手法のアルゴリズムの詳細について述べる。

### 4.1 概要

本手法では、緊急度として与えられた締切時刻を可能な限り遵守しつつ、網羅するまでの時間が短くなるような経路を生成する。本手法の主な流れは次の通りである。まず、図 4.1 のように最も締切時刻に近い箇所を目標地点として、現在地点からの経路探索を改変 A\*探索を用いて行う。この時点で、現在地点と目標地点の間の経路は固定される。その後、図 4.2 のように次に締切時刻に近い箇所を目標地点として、先ほどの目標地点からの経路探索を行う。この 2 つの手順を通過していない箇所がなくなるまで繰り返す。

本手法の経路探索の指針として、次の 3 つがある。

1. 締切時刻までに目標地点を訪れることができる
2. 既に訪れた地点を重複して訪れない
3. 残余領域が分断されない

本手法では、実際に航空機が飛行中に締切時刻の変更が行われたとしても、残りの領域を可能な限りまとめて訪れるようになっているため、動的な締切時刻の変更に対し比較的強いといえる。そのため、本手法では全体最適ではなく、目標地点の設定とそこまでの経路探索を繰り返すことによって経路の生成を行う。

上記のような指針を満たす経路を探索する手法として、A\*探索アルゴリズムのコスト計算に改変を加えた手法として改変 A\*探索アルゴリズムを用いる。改変 A\*探索アルゴリズムは次のような特徴を持つ。

- コスト計算を時間で行い、目標地点に到着するまでの見積もり時間を算出する
- 残余領域が分断されないよう、その形状を評価するヒューリスティック関数を持つ

この手法では、経路のコスト計算の際に、目標地点までの見積もり時間と残余領域の形状を評価する。求めたい経路は、目標地点までの最短経路ではなく、締切時間を遵守しつつも最終的に網羅した時に効率的になる経路である。そのため、改変 A\*探索では必ずしも最短経路をとらない。

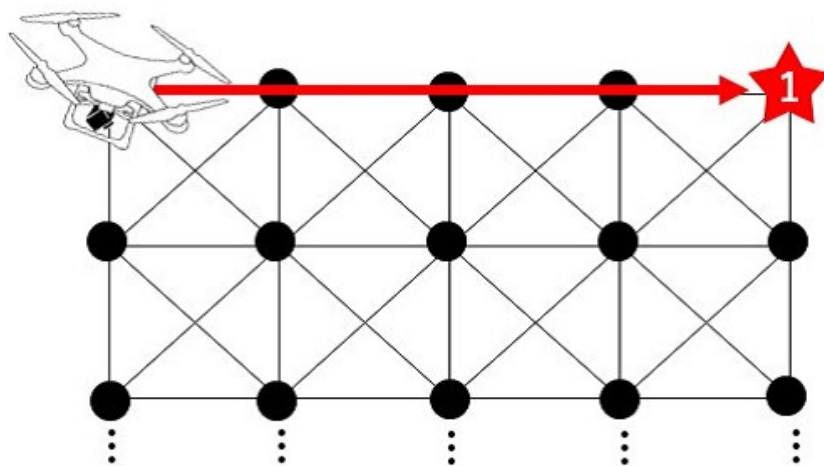


図 4.1 最も締切時間が差し迫っている地点までの経路

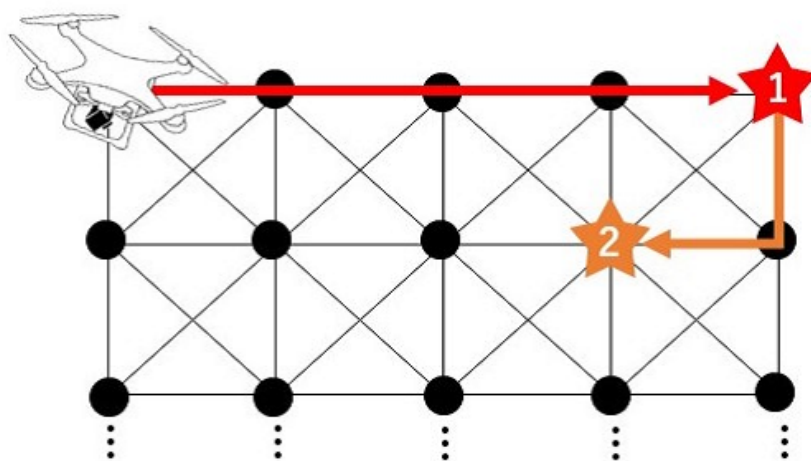


図 4.2 2 番目に締切時間が差し迫っている地点までの経路

## 4.2 目標地点決定方法

本手法では、緊急度が高く、現在地点から早く訪れることができ、可能であれば他の緊急度が高い箇所を通過できる経路が存在する地点を次の目標地点に決定する。目標地点決定のための手順は次のようになる。

1. 締切時刻が最も近い地点を目標地点とする。
2. 同じ締切時刻の地点が複数存在した場合、現在地点からの最短距離をとるときの飛行時間が短い箇所を目標地点とする。
3. 最短距離をとるときの飛行時間が同じだった場合は、まずそれぞれの歩道  $\Delta W$  を計算する。ここで、開始ノードを  $n_c$ 、終了ノードを  $n_{target}$  としたとき、目標地点決定に用いる関数として余

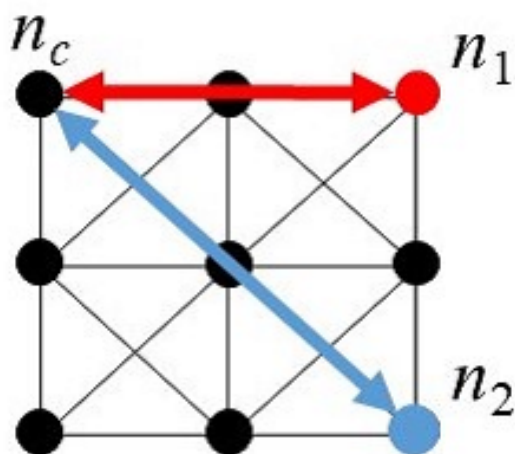


図 4.3 締切時刻が同じ地点が複数存在する場合

裕度  $Q(\Delta W)$  を次のように定義する.

$$Q(\Delta W) = \sum_{i=c}^{target} \{Deadline(n_i) - FirstTime(n_i, \Delta W)\} \quad (4.1)$$

余裕度  $Q(\Delta W)$  が返すの値が小さい方を目標地点とする.

4. ランダムに目標地点を決定する.

例えば, 現在地点の頂点  $n_c$  と次のような条件を満たす頂点  $n_1, n_2, n_3$  があるとする. ここで,  $\Delta W_c^1, \Delta W_c^2$  は現在地点  $n_c$  から  $n_1, n_2$  へのそれぞれの経路,  $ShortestTime(n_c, n_{target})$  は頂点  $n_c, n_{target}$  間の最短経路の飛行時間を返す関数とする. このとき, それぞれの最短距離をとるときの飛行時間として  $ShortestTime(n_c, n_1)$  と  $ShortestTime(n_c, n_2)$  を比較し, 小さい方を優先する. 図 4.3 のような分布の場合は,  $n_1$  の方がより距離が近く, 最短距離をとるときの飛行時間は短くなるため,  $n_1$  を次の目標地点として設定する.

もし, 最短距離をとるときの飛行時間が同じであった場合 ( $ShortestTime(n_c, n_1) = ShortestTime(n_c, n_2)$  である場合) には,  $\Delta W_c^1, \Delta W_c^2$  を求めた後,  $Q(\Delta W_c^1)$  と  $Q(\Delta W_c^2)$  を比較し, 大きい方を目標地点に設定する. 余裕度  $Q$  により実際に訪れる時間と締切時刻の差分の総和を求め, その値が小さい方を訪れるまでに余裕がないものとみなし, 優先する.

もし  $Q(\Delta W_c^1)$  と  $Q(\Delta W_c^2)$  も同じだった場合は, ランダムにて決定する.

### 4.3 経路探索手法

本手法では, A\*探索アルゴリズムを改変した改変 A\*探索アルゴリズムを用いる. まず, A\*探索アルゴリズムについて概要を述べた後, 本手法で用いる改変 A\*探索アルゴリズムの概要について言及する. その後, 改変 A\*探索アルゴリズムで用いるヒューリスティック関数について述べる.

### 4.3.1 A\*探索アルゴリズム

A\*探索アルゴリズムは、グラフにおける開始地点から目標地点までの最短経路を見つける問題において、スタートからの距離とゴールまでの距離の和の比較を繰り返すことで最短経路を探索するアルゴリズムである。まず、地点  $n_c$  に対して開始地点  $n_s$  からの距離を  $g(n_c)$ 、目標地点までの距離を  $h(n_c)$  とし、これらの和を総距離  $f(n_c)$  とする。

$$f(n_c) = g(n_c) + h(n_c) \quad (4.2)$$

ここで距離をコストとして扱えば、最短経路を求める際には総コスト  $f(n_c)$  が最小となり続けるような経路を選択すればよい。

しかし、開始地点からのコスト  $g(n_c)$  とは違い、目標地点までのコスト  $h(n_c)$  は既知ではない。そのため、目標地点までのコストの推定値  $h^*(n_c)$  を与えるものとする。この推定値  $h^*(n_c)$  はヒューリスティック関数と呼ばれる。総コストは  $f^*(n_c)$  として次のように与えられる。

$$f^*(n_c) = g(n_c) + h^*(n_c) \quad (4.3)$$

$h^*(n_c)$  には、ある程度妥当性を持つように設計した推定値が与えられる。

### 4.3.2 改変 A\*探索アルゴリズム

本手法で用いる改変 A\*探索アルゴリズムは、残余領域ができるだけひとまとまりになるように、目標地点に向かう経路探索を行う。そのため、このアルゴリズムによって生成される経路は、必ずしも最短経路をとるとは限らない。このような経路を生成するために、改変 A\*探索アルゴリズムは残余領域の形を評価するヒューリスティクスを持つ。このヒューリスティクスは、残余領域ができるだけひとまとまりになるような経路をとるために関数  $h^*(n_c)$  に  $c_{group}$  として組み込まれている。

改変 A\*探索アルゴリズムの考え方は、次の通りである。まず、地点  $n_c$  に対して開始地点  $n_s$  からの時間を  $g(n_c)$ 、目標地点までの時間を  $h(n_c)$  とし、これらの和を総時間  $f(n_c)$  とする。

$$f(n_c) = g(n_c) + h(n_c) \quad (4.4)$$

ここで時間をコストとして扱えば、最短経路を求める際には総コスト  $f(n_c)$  が最小となり続けるような経路を選択すればよい。

しかし、開始地点からのコスト  $g(n_c)$  とは違い、目標地点までのコスト  $h(n_c)$  は既知ではない。そのため、目標地点までのコストの推定値  $h^*(n_c)$  を与えるものとする。この推定値  $h^*(n_c)$  はヒューリスティック関数と呼ばれる。総コストは  $f^*(n_c)$  として次のように与えられる。

$$f^*(n_c) = g(n_c) + h^*(n_c) \quad (4.5)$$

この  $f^*(n_c)$  の比較を繰り返すことでコストが最小となるような経路を探索するアルゴリズムが改変 A\*探索アルゴリズムである。

### 4.3.3 ヒューリスティック関数

本手法で用いるヒューリスティック関数  $h^*(n_c)$  は、次のように定義する。

$$h^*(n_c) = t_{target} + t_{covering} \quad (4.6)$$

ここで,  $t_{target}$  は現在地点から目標地点までの見積もり時間,  $t_{covering}$  は残余領域を網羅する際の見積もり時間とする.

目標地点までの見積もり時間  $t_{target}$

$t_{target}$  は, 目標地点  $n_t$  を通過するまでの見積もりの時間である. 航空機の最高速度を  $v_{max}$  とすれば,  $t_{target}$  は現在地点から目標地点までの時間として, 次のように定義する.

$$t_{target} = \frac{\|Position(n_t) - Position(n_c)\|}{v_{max}} \quad (4.7)$$

残余領域の見積もり時間  $t_{covering}$

$t_{covering}$  は, 残余領域を網羅することを考えたときの見積もり時間である. 網羅を考えたとき, 残余領域は可能な限りまとまって存在することが望ましい. これは残余領域が分断されている場合, 網羅するときに必ず1度は既に通った箇所をもう一度通ることになり, 非効率であるためである.

ここで, グラフ  $G$  の頂点集合の内, 経路  $W$  に含まれていない頂点  $n'$  の部分集合を  $N' \subset N$  とする. また, 経路  $W$  に含まれる頂点を要素にもつ辺を除いた辺の集合  $E' \subset E$  とする. このとき, 残余領域は  $G' = (N', E')$  の形で表すことができる. 評価のために, 残余領域  $G'$  から  $l$  個の要素を持つ完全グラフ  $K_l = (N_k, E_k)$  を作成する. 頂点集合  $N_k \subset N$  は,  $N'$  中の1-連結している頂点集合  $C$  を1つの要素とする集合である. 辺集合  $E_k$  の各要素はコストとして距離を持つ. 各辺が持つ距離には, 1-連結グラフ  $C$  同士を結ぶ最短距離を近似値として用いる.

グラフ  $K$  を利用して, 残余領域に関して次の3つの指標を評価する.

#### 1. 残余領域の連結成分数

残余領域が分断されているかどうかを評価するため, 残余領域の連結成分数を用いる. 残余領域の連結成分とは, 残余領域に含まれる頂点の集合  $U \subset N$  の中で, 1-連結している頂点の集合  $C$  を指す. 連結成分の個数が増えた場合, 少なくとも1つは既に通った頂点を通らなければ網羅的に探索することができない. 残余領域の連結成分数は, グラフ  $K$  において頂点の個数  $|N_k|$  で表すことができる.

#### 2. 残余領域同士の距離

複数の残余領域が存在する場合, 残余領域同士の距離が遠ければ, その分重複して通過しなければならない頂点が増えてしまい, 経路の効率が悪化する. そこで, 複数の残余領域が存在する場合の網羅の効率性を評価するために, 残余領域同士の距離を評価する. 残余領域同士の距離の総和は, グラフ  $K_l$  の頂点集合  $N_k$  を網羅する際の辺のコストの総和  $total\_distance$  である. この時の網羅の順番は, 頂点集合  $N_k$  の要素に含まれる頂点  $n$  の締切時間から貪欲法を用いて決定する.

#### 3. 残余領域の分断の可能性

残余領域が複雑な形状であった場合, 動的な緊急度の変更によって極端に効率が低下してしまう可能性がある. そのため, 残余領域グラフ  $G'$  は, 分断の可能性が低くなるような, 単純な形状の密なグラフであることが望ましい. そこで, 分断するのに必要な最小辺数である辺連結度  $\lambda(G')$  を評価する. 残余領域グラフ  $G'$  において, 辺に重みがあるので辺連結度  $\lambda(G')$  は辺カットの重み和の最小値である. 残余領域グラフ  $G'$  の辺連結度  $\lambda(G')$  が小さい場合は, 分断しやすい複雑な形状のグラフをであるといえる.

上記の指標を用いて, 残余領域の見積もり時間を算出する. 隣り合う頂点間の最短距離を  $n\_distance$

**Algorithm 1** Efficient Route Search Algorithm**Input:**  $v_s, G$ **Output:**  $W$ 


---

```

1:  $v_c \leftarrow v_s$ 
2:  $U \leftarrow G.node$ 
3:  $R \leftarrow \phi$ 
4: while  $U \neq empty$  do
5:    $target \leftarrow U.priority.max$ 
6:    $path \leftarrow Mod - A\_star(v_c, target, R)$ 
7:    $R \leftarrow R + path$ 
8:    $Move(R)$ 
9:    $v_c \leftarrow target$ 
10:   $U \leftarrow U - R.all$ 
11: end while

```

---

として、残余領域の見積もり時間  $t_{covering}$  は次のように定義する.

$$t_{covering} = \frac{total\_distance + n\_distance \times |N_k| + \lambda(G')}{v_{max}} \quad (4.8)$$

#### 4.4 アルゴリズム

提案手法の疑似コードを Algorithm1 に示す.

まず、入力として開始地点  $v_s$  と、領域のグラフ  $G$  を与える. 1, 2 行目では初期化として、現在地点  $v_c$  に  $v_s$  を、未探索ノードの集合  $U$  に  $G.node$  を代入する. 5 行目では、 $target$  に優先順位が最大のセルを代入する. 6 行目において、現在地点  $v_c$  から  $target$  までの経路探索を A\*探索アルゴリズムを用いて行い、経路  $path$  を得る. 7 行目で、 $R$  に  $path$  を追加する. 8, 9 行目で、経路にもとづいて移動を行い、 $v_c$  を  $target$  に更新する. 10 行目で、未探索ノードの集合  $U$  から通過したノードを削除する. 未探索ノードがなくなるまで、この 5~10 行目までの処理を繰り返す.



## 第5章

# シミュレーション

本節では、まずシミュレーションを行う準備として使用するデータやパラメータなどの設定について述べる。その後、単純な手法と提案手法それぞれのシミュレーション結果を示し、評価を行う。

### 5.1 シミュレーションの設定

シミュレーションを行うにあたり、約  $2.6\text{km}^2$  の面積を持つ筑波大学筑波キャンパスが包含できる程度の約  $3\text{km}^2$  の矩形領域を考える。この領域に対し、表 5.1 のような航空機で空撮することを想定する。

表 5.1 より、撮影視野は約  $172.05\text{m} \times$  約  $68.80\text{m}$  の矩形領域をとる。今回のシミュレーションでは  $3\text{km}^2$  の矩形領域を撮影視野で分割し、 $16 \times 16$  のノードを持つグラフを対象とする。それぞれランダムに締切時間を設定したグラフ 100 個を用意する。締切時間のことを考えずに、グラフを端から順番に通過した時に網羅が完了するまでの時間を  $min\_flight\_time$  とする。締切時間の分布には、 $\mu = 1.0 \times min\_flight\_time$ ,  $\delta = 0.25 \times min\_flight\_time$  の値を用いる。これらのグラフ 1 つ 1 つに対して、提案手法を用いた時の経路  $W_{proposal}$ , 単純手法を用いた時の経路  $W_{simple}$  を求める。ここで、単純手法は締切時間が短いノードを次の目標地点に設定し、そのノードまでの最短経路探索を繰り返す手法とする。

### 5.2 シミュレーション結果

単純な手法と提案手法のシミュレーション結果を示し、その性能を比較する。比較には、2章で定義した、締切時間をどれだけ遵守できたかを評価するための評価関数  $E_u(W)$  と、どれだけ短い航空時間で網羅できたかを評価する関数  $E_c(W)$  を用いる。

まず、グラフ毎の経路  $W_{proposal}$ , 経路  $W_{simple}$  から  $E_u(W_{proposal})$ ,  $E_u(W_{simple})$  を算出し、どちらがより締切時間を遵守する経路を作成できたかを比較する。グラフ毎に  $E_u(W)$  を比較したものを図 5.1 に示す。ここで図 5.1 においては、可視性を考えて、 $E_u(W_{simple})$  の値でソートした上で横軸のグラフ番号の割り振りを行っている。図 5.1 から、ほとんどのグラフにおいて  $E_u(W_{proposal})$  の方が低い値をとっていることが分かる。これは、提案手法の経路  $W_{proposal}$  の方が、締切時間を大きく超過して到着したノードが少ないことを示す。しかし、グラフ番号 48, 61, 96, 100 の 4 つのグラフにおいては、単純手法の  $E_u(W_{simple})$  が提案手法の  $E_u(W_{proposal})$  よりも低い値をとる。

次に、グラフ毎の経路  $W_{proposal}$ , 経路  $W_{simple}$  から  $E_c(W_{proposal})$ ,  $E_c(W_{simple})$  を算出し、どちらがより短い航空時間で網羅できたかを比較する。グラフ毎に  $E_c(W)$  を比較したものを図 5.1 に示す。図 5.2 から、シミュレーション対象の全てのグラフにおいて単純手法と比べて提案手法の  $E_c(W_{proposal})$  の方が低い値をとる。これは、提案手法によって網羅までの航空時間が短縮されたことを示す。

表 5.1 航空機設定

重量	1380g
水平視野角	60°
垂直視野角	27°
最大飛行速度	72km/h
地上からの撮影高度	149m

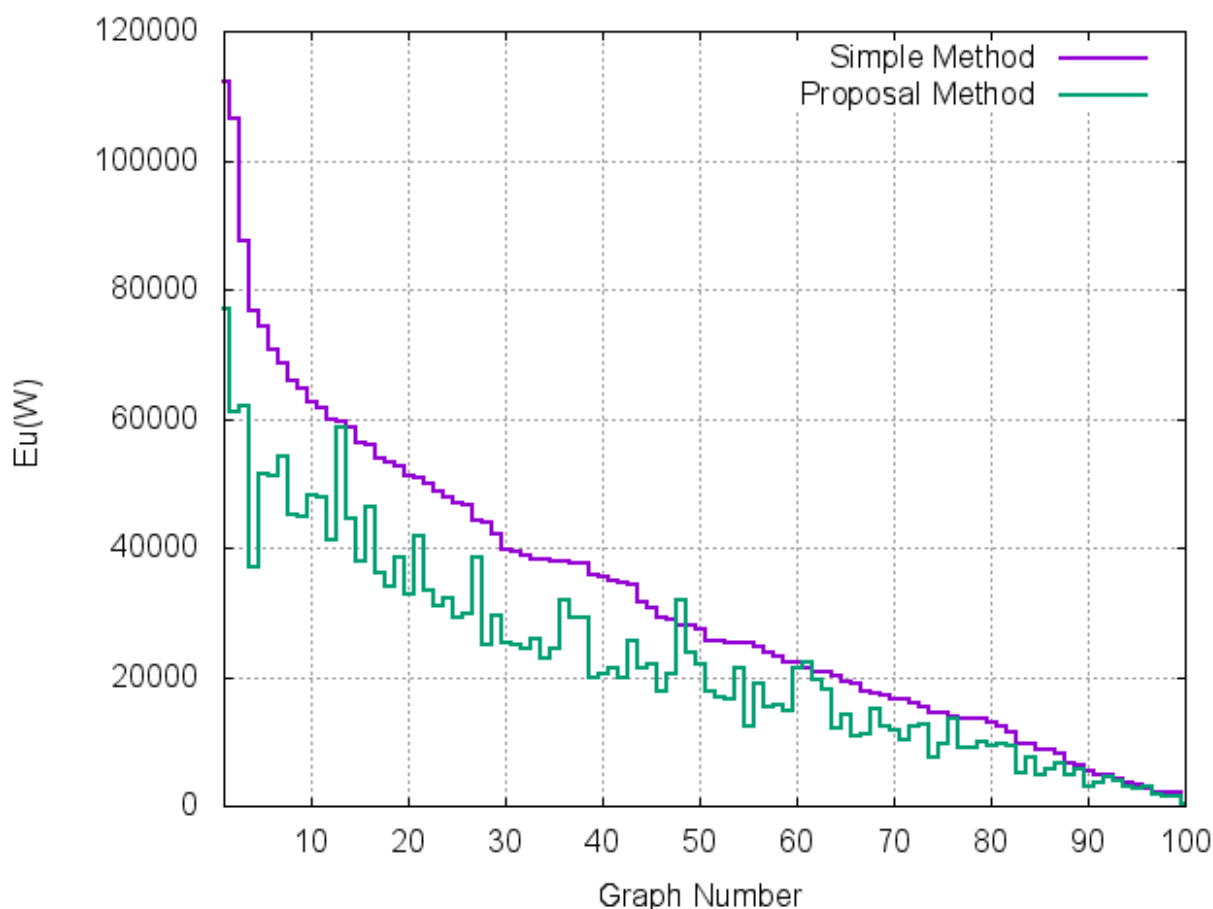


図 5.1 各グラフに対する  $E_u(W_{proposal})$  と  $E_u(W_{simple})$  の比較

さらに、 $E_c(W)$  に対する箱ひげ図を用いて両手法の  $E_u(W)$  と  $E_c(W)$  それぞれの分布を評価する。作成した箱ひげ図を図 5.3, 図 5.4 に示す。図 5.3 から、提案手法によって分散が抑えられていることがわかる。図 5.4 から、外れ値を除けば最大値、最小値共に提案手法が勝っていることが確認できた。これらことから、提案手法がどのようなグラフに対しても比較的安定した効率化が可能であることがわかる。

続いて、シミュレーション結果の統計量を比較する。各グラフの  $E_u(W)$  と  $E_c(W)$  の値の平均と標準偏差を表 5.2 に示す。表 5.2 から、単純な手法に比べて、提案手法では  $E_c(W)$  に対して平均 29.4275351%,  $E_u(W)$  に対して平均 36.52036839% の効率化が確認できた。

最後に、提案手法と単純手法それぞれの  $E_u(W)$ ,  $E_c(W)$  の値から散布図を作成した。作成した散布図を図 5.5 に示す。図 5.1, 図 5.2, 図 5.5 から、ほとんどのグラフに対して  $E_u(W)$ ,  $E_c(W)$  共に提案手法の方が優勢であることがわかる。上記のことから、単純手法に比べ、提案手法は不十分な情報下に

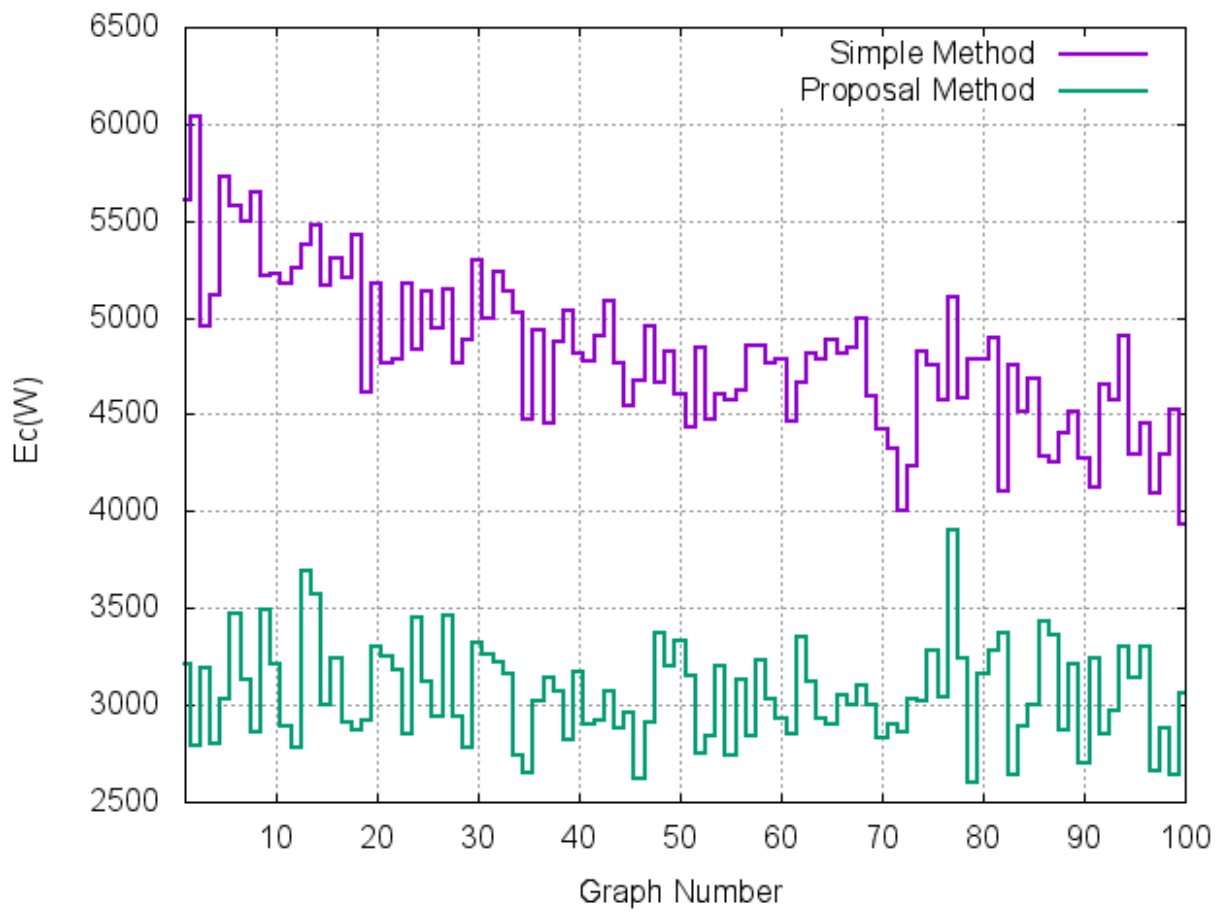
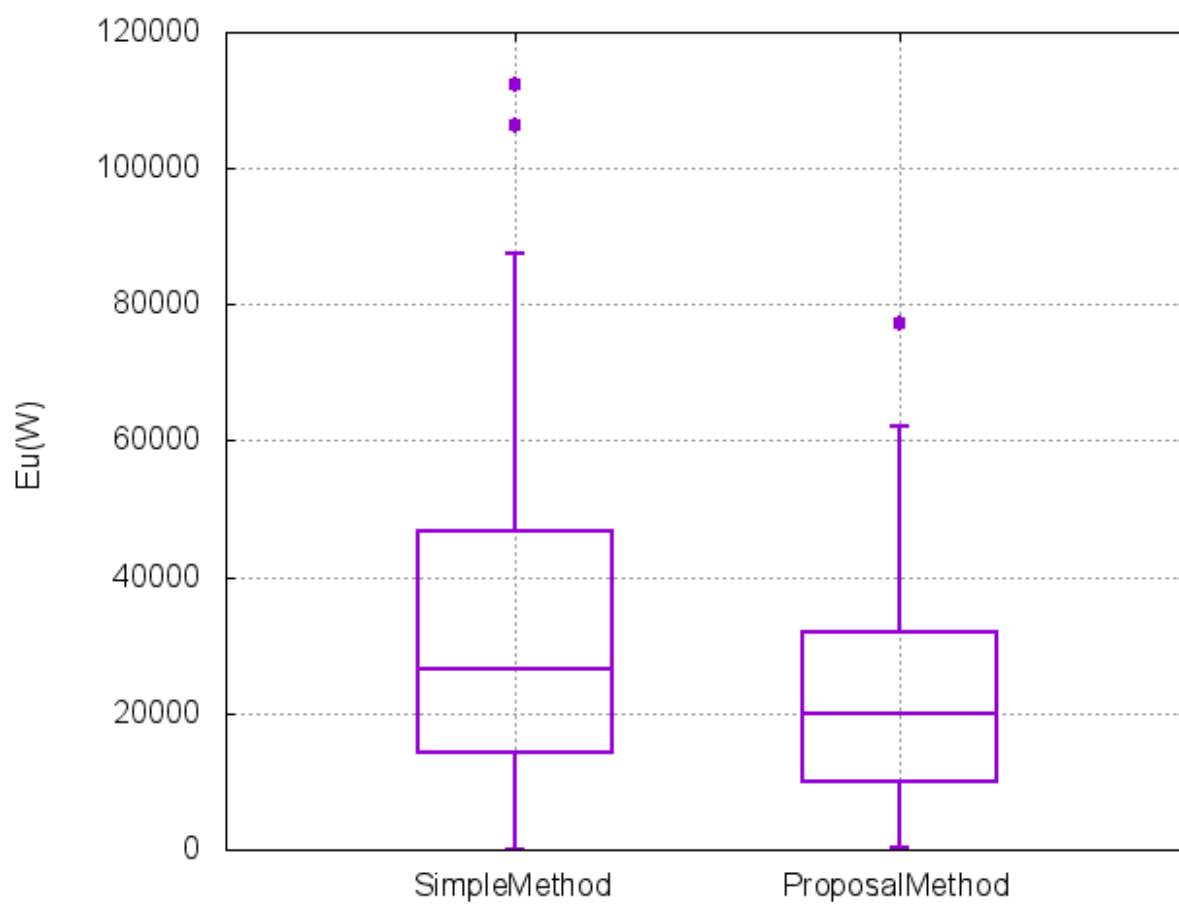


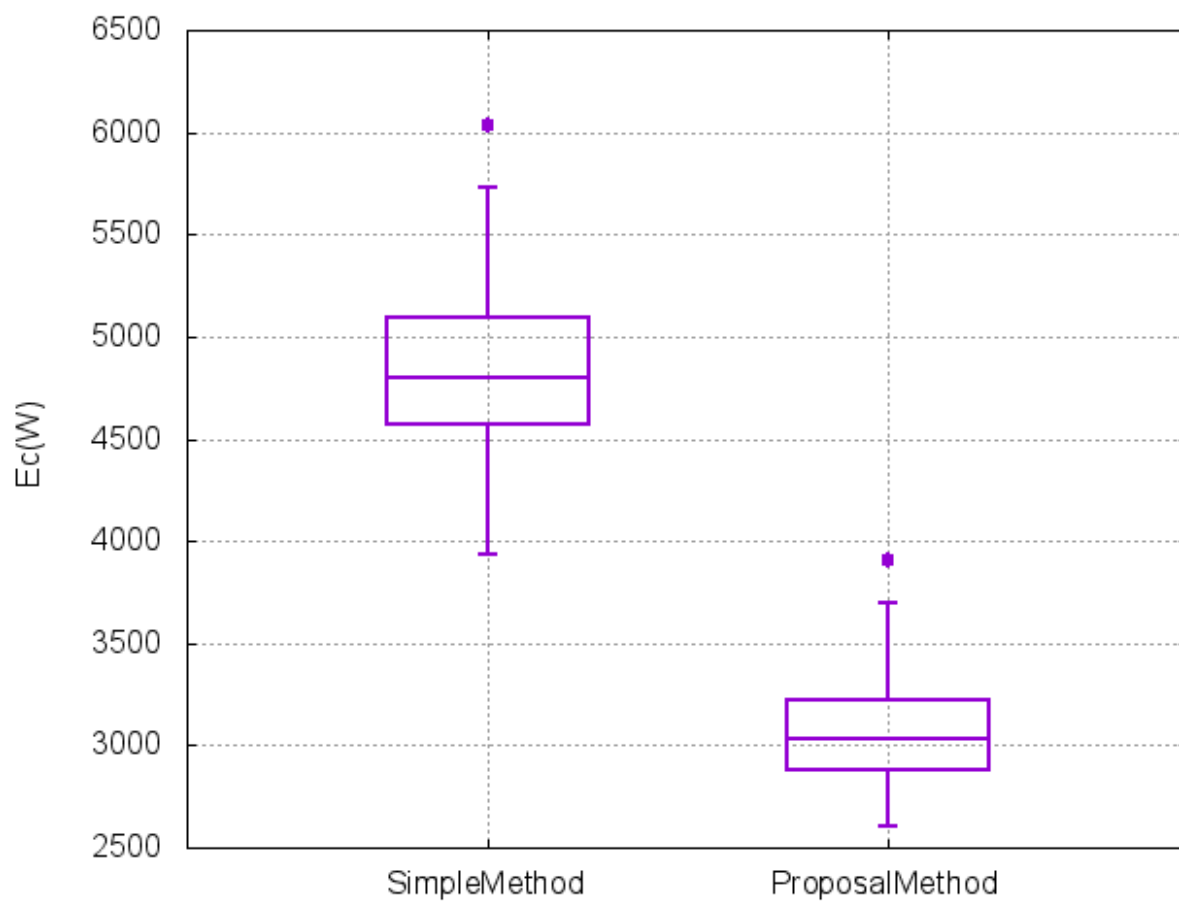
図 5.2 各グラフに対する  $E_c(W_{proposal})$  と  $E_c(W_{simple})$  の比較

表 5.2 各グラフに対する  $E_u(W)$  と  $E_c(W)$  の平均と標準偏差の比較

	単純手法	提案手法
$E_u(W)$ の平均	31928.77	22532.92
$E_u(W)$ の標準偏差	22960.94046	15980.40222
$E_c(W)$ の平均	4820.213159	3059.853556
$E_c(W)$ の標準偏差	393.215807	246.8548394

において緊急度と網羅時間を共に考慮した経路が出力できた。

図 5.3  $E_u(W)$  に対する箱ひげ図

図 5.4  $E_c(W)$  に対する箱ひげ図

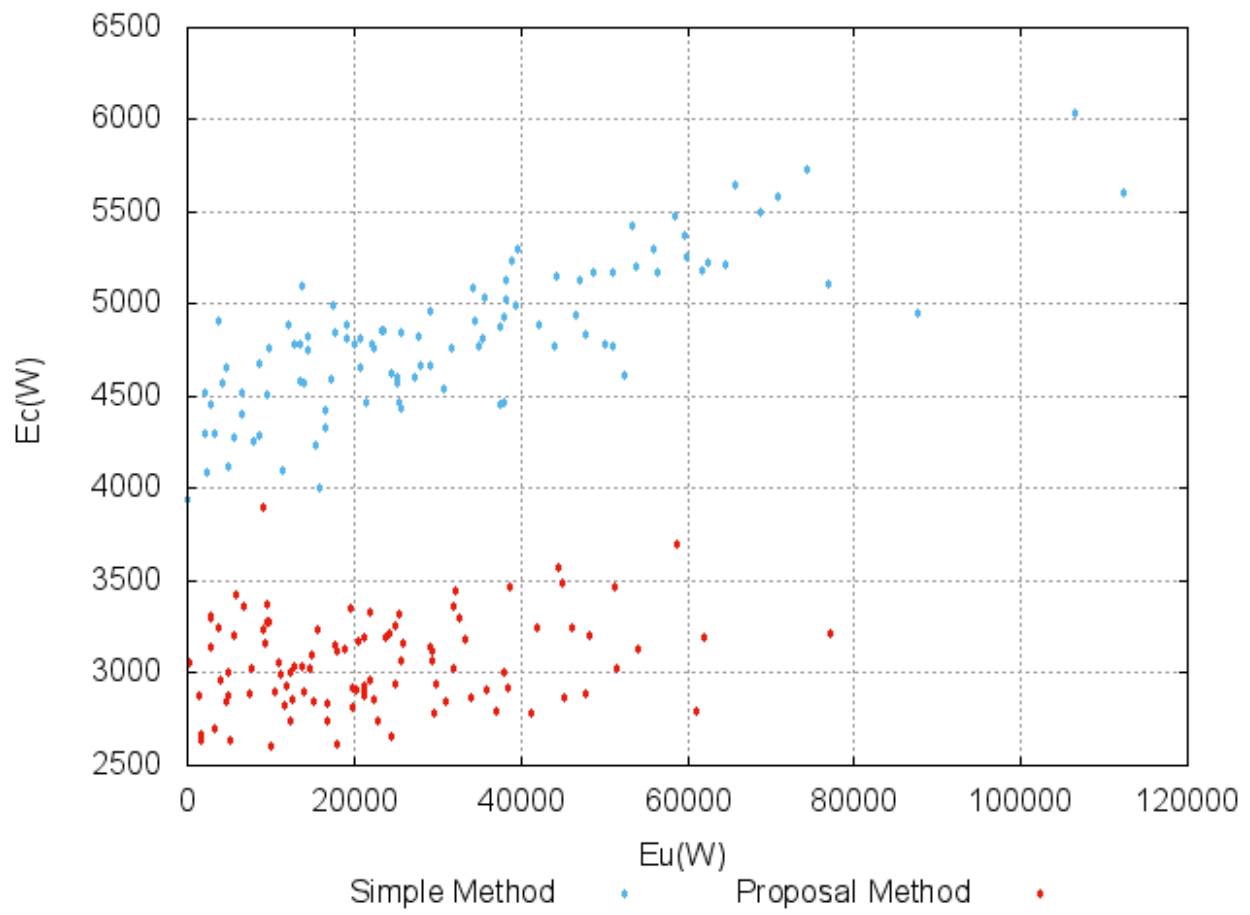


図 5.5  $E_u(W)$ ,  $E_c(W)$  の散布図による手法の比較

## 第 6 章

# まとめと今後の課題

本論文では、不十分な情報下での緊急度に基づき、効率的に網羅経路を探索する手法の提案を行った。不十分な情報下では、緊急度が動的に変更されることが想定される。このとき、緊急度の高い箇所を優先しつつ網羅を行うような経路を、全体最適によって求めることは計算量の観点から現実的ではない。だからといって、単純に緊急度の高い箇所から順番に回る方法では、多くの重複が発生してしまい、非効率である。

これらに対し提案手法では、グラフのノードを網羅的に訪れる際に、締め切り時間に関して当初は不十分な情報であり動的に更新されるような状況下でも、効率的な経路を計算する。経路の計算には、A\*探索アルゴリズムを改変した改変 A\*探索アルゴリズムを用いる。本手法は、目標地点決定と改変 A\*探索アルゴリズムによる経路探索を繰り返すことによって、逐次的に経路を探索する。本手法の特徴は、残余領域が可能な限りひとかたまりの状態を維持できるようなヒューリスティクスを探索アルゴリズムに組み込むことで、情報の動的な変更に対しても極端な性能劣化を防ぐことにある。

また、本論文では提案手法の評価のために、単純に緊急度の高い箇所から順番に通過する場合との比較シミュレーションを行った。締切時間をランダムに設定した 100 パターンのグラフを用いて実験を行った結果、本手法は単純に緊急度の高いノードから回る手法と比べて、締切時間をどれだけ遵守できたかの評価に関して平均 29.4275351% の効率化を確認した。加えて、どれだけ短い航空時間で網羅できたかの評価に関して、航空時間を平均 36.52036839% 削減することができた。

今後の課題としては、第 1 にアルゴリズムの効率化の必要性が挙げられる。逐次処理である上、近似計算を用いているため、組み合わせ最適化の厳密解法と比べると、計算量は大幅に削減できたものの、ノード数が 10000 を超えるようなサイズの大きいグラフに対しては、現段階で実用に耐える時間内での計算ができていない。そのため、サイズの大きなグラフに対して提案手法が適用できるよう、アルゴリズムを効率化する必要がある。

第 2 に、実際の被災地での実用性の検証が必要である。今回は、締切時間をランダムに設定したが、提案手法で求める経路の効率性は与えられるグラフに依存する。そのため、実際の被災地のデータを用いてグラフを作成し、提案手法の有用性を検証する必要がある。

## 謝辞

本研究を進めるにあたり、いつも丁寧かつ熱心にご指導頂いた図書館情報メディア研究科の森嶋厚行先生と松原正樹先生に深く感謝致します。ご多忙である中、何度も研究の相談に応じて頂きました。また、日常の議論を通して研究に関する様々な知識や示唆を頂きました。

また、同研究科の鈴木伸崇先生には、これまでの知見に基づいて研究内容に関する多様なアドバイスを頂きました。研究を進めるにあたり、研究打ち合わせでの鈴木先生のご指摘に何度も助けて頂きましたことに関しまして、ここに心からの感謝の意を表します。

同期の皆様、および杉本永森研究室、阪口研究室の皆様には、ゼミや研究の相談に何度も乗って頂きました。研究に関しての様々な協力に関して、皆様へ感謝の気持ちと御礼を申し上げたく、謝辞にかえさせていただきます。



## 参考文献

- [1] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, Vol. 6, No. 1, pp. 80–91, 1959.
- [2] Jan Karel Lenstra and AHG Kan. Complexity of vehicle routing and scheduling problems. *Networks*, Vol. 11, No. 2, pp. 221–227, 1981.
- [3] Roberto Cordone and Roberto Wolfler Calvo. A heuristic for the vehicle routing problem with time windows. *Journal of Heuristics*, Vol. 7, No. 2, pp. 107–129, 2001.
- [4] Martin Desrochers, Jacques Desrosiers, and Marius Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, Vol. 40, No. 2, pp. 342–354, 1992.
- [5] Éric Taillard, Philippe Badeau, Michel Gendreau, François Guertin, and Jean-Yves Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, Vol. 31, No. 2, pp. 170–186, 1997.
- [6] Nagraj Balakrishnan. Simple heuristics for the vehicle routing problem with soft time windows. *Journal of the Operational Research Society*, pp. 279–287, 1993.
- [7] Bruce L Golden, Subramanian Raghavan, and Edward A Wasil. *The vehicle routing problem: latest advances and new challenges*, Vol. 43. Springer Science & Business Media, 2008.
- [8] Roberto Montemanni, Luca Maria Gambardella, Andrea Emilio Rizzoli, and Alberto V Donati. Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*, Vol. 10, No. 4, pp. 327–343, 2005.
- [9] Marco Dorigo and Thomas Stützle. Ant colony optimization: overview and recent advances. In *Handbook of metaheuristics*, pp. 227–263. Springer, 2010.
- [10] Zhongli Liu, Yinjie Chen, Benyuan Liu, Chengyu Cao, and Xinwen Fu. Hawk: an unmanned mini-helicopter-based aerial wireless kit for localization. *IEEE Transactions on Mobile Computing*, Vol. 13, No. 2, pp. 287–298, 2014.
- [11] Qilie Liu, Xiang Pang, Yong Wang, Yingjun Pan, and Leilei Li. An improved path management policy for the ferry in opportunistic networks. *JNW*, Vol. 7, No. 10, pp. 1568–1575, 2012.
- [12] PB Sujit and D Ghose. Search using multiple uavs with flight time constraints. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 40, No. 2, pp. 491–509, 2004.