

経路探索における高速な
経路グラフ最適化に関する研究

東京工科大学大学院

バイオ・情報メディア研究科

メディアサイエンス専攻

小坂 大樹

経路探索における高速な
経路グラフ最適化に関する研究

指導教員 渡辺 大地 教授

東京工科大学大学院
バイオ・情報メディア研究科
メディアサイエンス専攻

小坂 大樹

論文の要旨

論文題目	経路探索における高速な 経路グラフ最適化に関する研究
執筆者氏名	小坂 大樹
指導教員	渡辺 大地 教授
キーワード	経路探索、グラフ構造、ゲーム AI 最適化, ダイクストラ法

[要旨]

経路探索を行うには、マップ状態を表すグラフデータである経路グラフが必要である。しかし、マップの大きさなどによって経路グラフのデータが増えると、経路探索の処理時間も増加する。特にゲーム分野では高速な処理を必要とするため、経路探索の処理時間短縮は重要であるといえる。経路探索の処理時間短縮のために、経路探索アルゴリズムであるダイクストラ法や A* アルゴリズムなどの改良が行われてきた。また、経路探索を行う際のグラフのデータを階層化により削減することで探索時間の短縮を図る、上位層グラフなどがある。しかし、マップによっては階層化に適していない経路グラフがある。そこで、本研究では経路探索の処理時間削減のための汎用的な経路グラフのデータ削減方法を提案した。経路グラフのエッジに対して、頻繁に利用するエッジほど重要であると位置付けたエッジ重要度を設定する。そしてエッジ重要度に基づいてエッジを削除することで効果的に経路グラフのデータ削減を図った。まず、総経路長増分指標によりエッジ重要度を求めこれをエッジ重要度の厳密解である厳密重要度とした。しかし、厳密重要度の算出には処理時間が大幅に掛かってしまうため、短い処理時間で厳密重要度に近い結果を得るための手法を提案した。また、提案手法により求めたエッジ重要度を簡易重要度とした。複数の経路グラフで提案手法を適用したところ、いくつかの経路グラフでは厳密重要度に近い結果を得られた。また、厳密重要度と簡易重要度算出に掛かる計算時間の計測も行った。これにより、提案手法はある程度の大きさの経路グラフであれば高速な処理を必要とするゲーム分野に対しての適用が可能であることが分かった。

A b s t r a c t

Title	A Study on Fast Path Graph Optimization in Path Finding
Author	Daiki Kosaka
Advisor	Taichi Watanabe
Key Words	Pathfinding, Graph, GameAI Optimization, Dijkstra

[summary]

In order to perform route search, a route graph, which is graph data representing the map state, is necessary. However, as the size of the map increases, so does the processing time for pathfinding. Especially in the field of games, where high-speed processing is required, it is important to reduce the processing time of pathfinding. In order to reduce the processing time of route finding, improvements have been made to route finding algorithms such as Dijkstra method and A* algorithm. In addition, there are upper layer graphs that reduce the amount of data in the graph for path finding by hierarchizing it, thereby shortening the search time. However, some route graphs are not suitable for hierarchization depending on the map. Therefore, in this study, we proposed a general-purpose data reduction method for pathway graphs to reduce the processing time of pathway search. For each edge in the pathway graph, we set the edge importance, which indicates that the more frequently used edges are more important. Then, the data of the route graph is effectively reduced by deleting the edges based on the edge importance. First, we calculated the edge importance using the total path length increment index, and used this as the exact importance, which is the exact solution of the edge importance. However, the calculation of the exact importance takes a large amount of processing time, so we proposed a method to obtain results close to the exact importance in a short processing time. The edge importance calculated by the proposed method was used as the simplified importance. We applied the proposed method to several path graphs, and obtained results close to the exact importance in some of them. We also measured the computation time required to calculate the exact importance and the simplified importance. The results show that the proposed method can be applied to game fields that require fast processing for pathways of a certain size.

目次

第 1 章	はじめに	1
1.1	研究背景と目的	2
1.2	論文構成	7
第 2 章	提案手法	8
2.1	表記定義	9
2.2	エッジ重要度	10
2.3	総経路長増分指標	11
2.3.1	総経路長増分指標による厳密重要度算出	11
2.3.2	総経路長増分指標の計算時間	12
2.4	提案手法	13
2.4.1	周辺にあるエッジとの類似度	13
2.4.2	経路グラフの重心からの距離	16
2.4.3	重心への方向類似度	17
2.4.4	簡易重要度の算出	18
第 3 章	検証	19
3.1	検証準備	20
3.2	検証:経路グラフ毎の疑似カラーと類似度 H	21
3.2.1	疑似カラーの設定	24
3.2.2	類似度の算出	30
3.3	総経路長増分指標・提案手法による処理時間比較	31
3.4	提案手法による大規模な経路グラフでの計算時間計測	32
第 4 章	まとめ	34
	謝辞	36

参考文献	38
発表実績	42

目 次

1.1	八王子駅から東京工科大学へのルート検索を行った様子	2
1.2	上位層グラフの一例	5
1.3	障害物が点在しているマップの一例	6
2.1	ノードとエッジ	9
2.2	最短経路の一例	10
2.3	エッジ削除による経路グラフのデータ削減を行った様子	10
2.4	削除するエッジにより求まる経路が長くなる場合	11
2.5	経路グラフ G と経路グラフ G_k の一例	12
2.6	エッジ e_i の周辺にあるエッジ	14
2.7	類似度の高いエッジ	15
2.8	重心 G とエッジ e_i との距離を求めた様子	17
2.9	単位ベクトル $\mathbf{V}'_i$ の一例	18
3.1	エッジに疑似カラーを設定した様子	21
3.2	経路グラフ 1	22
3.3	経路グラフ 2	22
3.4	経路グラフ 3	23
3.5	経路グラフ 4	23
3.6	経路グラフ 5	24
3.7	経路グラフ 1:総経路長増分指標	25
3.8	経路グラフ 1:提案手法	25
3.9	経路グラフ 2:総経路長増分指標	26
3.10	経路グラフ 2:提案手法	26
3.11	経路グラフ 3:総経路長増分指標	27
3.12	経路グラフ 3:提案手法	27

3.13 経路グラフ 4:総経路長増分指標	28
3.14 経路グラフ 4:提案手法	28
3.15 経路グラフ 5:総経路長増分指標	29
3.16 経路グラフ 5:提案手法	29

第 1 章

はじめに

1.1 研究背景と目的

経路探索とは、ある地点からある地点への移動経路を算出するものである。カーナビゲーションシステム [1] や、ロボット AI、ゲーム AI[2]、交通シミュレーション [3] や、災害時の経路シミュレーション [4] など多くの分野で用いる。例えば、戸田ら [5] は移動ロボットが地図情報を持つ場合と持たない場合でどのように経路が変化するかシミュレーションを行った。藤本ら [6] は移動ロボットの経路探索時において障害物上のエッジを高速に抽出するための手法について提案した。佐藤ら [7] は Influence Map と経路探索を用いて弾除けシューティングゲームにおける人間らしい AI プレイや実現のための手法を提案した。

図 1.1 は、Google マップで八王子駅から東京工科大学へのルート検索を行った様子を示したものである。



図 1.1 八王子駅から東京工科大学へのルート検索を行った様子

(Google Maps <https://maps.google.co.jp> より参照)

経路探索を行う際のデータとして、グラフを用いる。グラフにおけるノードを各地点、エッジを各地点同士を結ぶ道として扱い、それらのデータを用いて経路探索を行う。そのため、新しい道路ができるといった地形情報に変化がある場合はそれらのデータを追加する必要がある。例えば、ナビタイムジャパンは新しい道路を自動で地図に反映する機能を提供している [8]。

経路探索アルゴリズムについていくつか紹介する。なお、ここで述べるアルゴリズムはごく一部である。例えば、森畑ら [9] の論文ではさらに多くの経路探索アルゴリズムについて紹介している。

経路探索の基本ともいえるアルゴリズムの 1 つに、Dijkstra[10] が提唱したダイクストラ法がある。このアルゴリズムは、エッジが非負のコストを持つグラフにおいてノード間の最適な経路を算出するものである。カーナビゲーションシステムや、電車の乗換案内 [11] にも利用されており、運賃や所要時間をコストとすることで、適切な経路を探索する。

しかし、経路探索を行うマップの拡大やコストなどの情報の追加に伴いグラフデータが大きくなると、ダイクストラ法をそのまま適用しただけでは、探索に多くの時間を要してしまう。カーナビゲーションシステムや電車の乗換案内においても、グラフデータが膨大であるため、実際はダイクストラ法などを改良したアルゴリズムを適用していることが多く、他の分野においても同様である。例えば、James ら [12] や Michael ら [13] はダイクストラ法を拡張した手法を提案している。

経路探索には様々な種類があり、昔から多くの研究 [14] がされている。チェスや将棋のような完全情報ゲームと呼ばれる分野では、ゲーム木探索を行う事で AI を作成し、これを利用した研究 [15][16] が行われている。モンテカルロ法を利用したヒューリスティックな木探索であるモンテカルロ木探索を用いた研究 [17] も行われている。また遺伝的アルゴリズムを用いることで、最短経路だけでなくいくつかの経路候補を求める手法に関する研究 [18][19] も行われている。

ゲーム分野では、キャラクターの移動経路を求める為に経路探索を行う。経路探索などの処理は

キャラクターの位置やマップ状況などが変化していく度に行う必要がある。また、ゲームは1秒間に数十回から数百回程度画面更新を行っているため、極めて高速に計算処理を行う必要があり、経路探索についても同様である。さらに、ゼルダの伝説 ブレス オブ ザ ワイルド [20] や Ghost of Tsushima[21] といった広大なマップ上をキャラクターが自由に移動できるオープンワールドゲームが増えているため、探索時間の短縮が特に重要となっている。

探索時間を短縮するために多くの研究が行われている。2点間最短経路問題を解くために Peter ら [22] は、A*アルゴリズムを提案した。このアルゴリズムは、ダイクストラ法の探索時における冗長な部分を省略するためのアルゴリズムである。ゲーム分野においてキャラクターの移動を行うための経路探索などで用いることが多い。基本的な流れはダイクストラ法と同じである。A*アルゴリズムでは、そこからヒューリスティック関数というものを設定する。ヒューリスティック関数とは、あらかじめ判明している情報を元に追加で設定するコストのことである。多くの場合、目的地となるノードからのユークリッド距離を各ノードに設定し、冗長な探索を省略している。これにより、探索量・時間の短縮を図っている。

また、Stenz[23] が提唱した D*アルゴリズムは、ノードが変化した際に、関連のあるノードのみを更新するアルゴリズムである。もともと存在した道が塞がるなど、マップが変化した際にダイクストラ法や A*アルゴリズムは一から探索をし直す必要があるが、D*アルゴリズムでは、一部のノードのみ更新すれば良いため、マップ変化に応じた再探索の時間を短縮することができる。

一方で、経路探索のアルゴリズムを改良するのではなく、経路探索を行うためのグラフデータを削減することで、探索時間の短縮を図る手法がある。処理時間を短縮するためにデータ削減を図る手法は、3DCG の分野などでも用いる。3D モデルの形状を表現するポリゴンの数を減らすポリゴンリダクション [24][25] により、データ量を削減したりレンダリング時間の短縮を図る。

グラフデータを削減するための手法の一つとして、上位層グラフ [26] がある。上位層グラフとは、階層構造を持つグラフのことである。上位層グラフでは、グラフデータにおいて位置的に近

いノード群をひとつのエリアとして扱う。その後、各エリアを一つのノードとして扱うことでグラフに対して階層化を行う。図 1.2 は、上位層グラフの一例を示したものである。

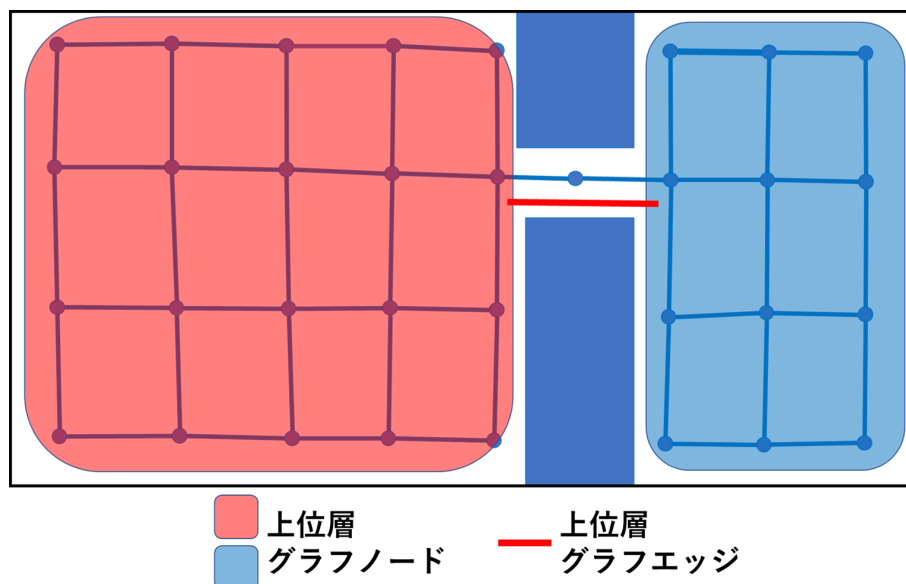


図 1.2 上位層グラフの一例

このような上位層グラフを用いることで、探索時におけるグラフデータの削減を行うことができ、計算時間の削減に繋がる。しかし、上位層グラフでは対応できない場合がある。上位層グラフは、グラフのノードに対してエリア分けを行う必要がある。その為、グランド・セフト・オート [27] における街中や、Ark:Survival Evolved[28] における森の中といった、建物や木などの障害物が点在しているようなマップでは、上位層グラフを作成するためのエリア分けが難しい場合がある。図 1.3 は、障害物が点在しているマップの一例を示したものである。

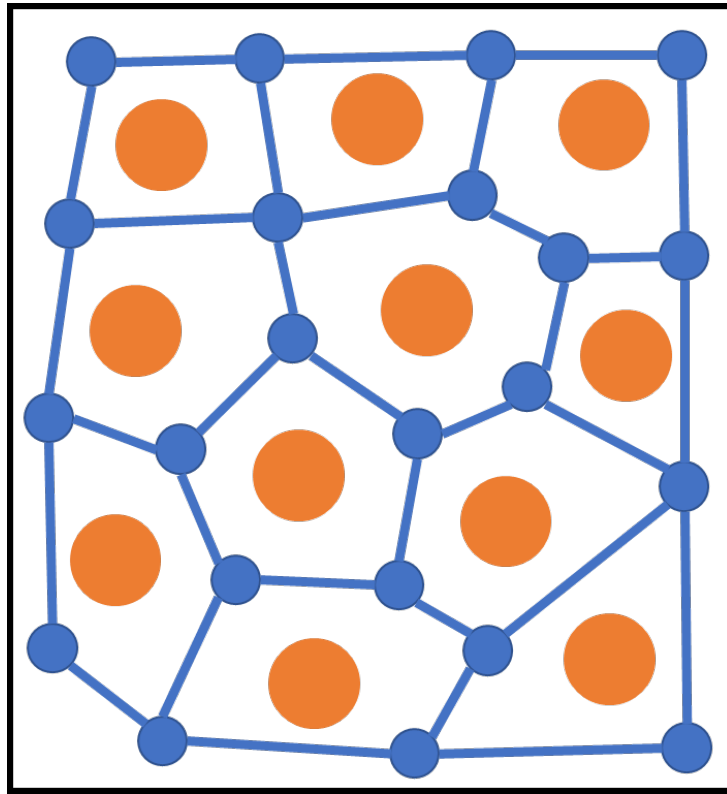


図 1.3 障害物が点在しているマップの一例

障害物の間を縫うようにノードやエッジが配置してあるため、エリア分けが難しくなっている。

そこで本研究では、経路探索の処理時間短縮を目的とした汎用的な経路グラフのデータ削減手法についての提案を行った。経路グラフにおけるエッジは、頻繁に利用するエッジとそうでないエッジが存在する。そして、頻繁に利用するエッジほど重要度の高いものと位置づけ、これを本研究ではエッジ重要度と呼称した。本手法では、エッジ重要度を以下の3つの指標により算出するものとした。

- 周辺にあるエッジとの類似度
- 経路グラフの重心からの距離
- 重心への方向類似度

検証により本手法はある程度の大きさのマップであれば高速な処理を必要とするゲームに対して

の適用が可能であることが分かった。

1.2 論文構成

本論文は全 5 章にて構成する。2 章では本手法について述べ、3 章では本研究の提案手法について検証した結果を述べる。そして、4 章ではまとめを述べる。

第 2 章

提案手法

2.1 表記定義

本研究におけるグラフ各部の数式表現は、基本的には文献 [29] に準拠するものとし、以下のよう
に定める。

- 本論文では、経路探索に用いるグラフのことを経路グラフと呼称する。
- 経路グラフにおける頂点のことをノード、辺のことをエッジと呼称する。図 2.1 で、ノード
及びエッジを示す。
- エッジ e_n がノード P_i とノード P_j を結んでいるとき、エッジ e_n は P_i, P_j 間のユークリッ
ド距離 $|\overrightarrow{P_i P_j}|$ をコストとして持つ。
- P_i を始点, P_j を終点とする経路の内、長さが最短となる経路を最短経路とし、その長さを

$$D(P_i, P_j) \quad (2.1)$$

と表記する。このとき

$$D(P_i, P_i) = 0 \quad (2.2)$$

とする。以下の図 2.2 は図 2.1 において P_0 を始点, P_5 を終点とした際の最短経路を示した
ものである。

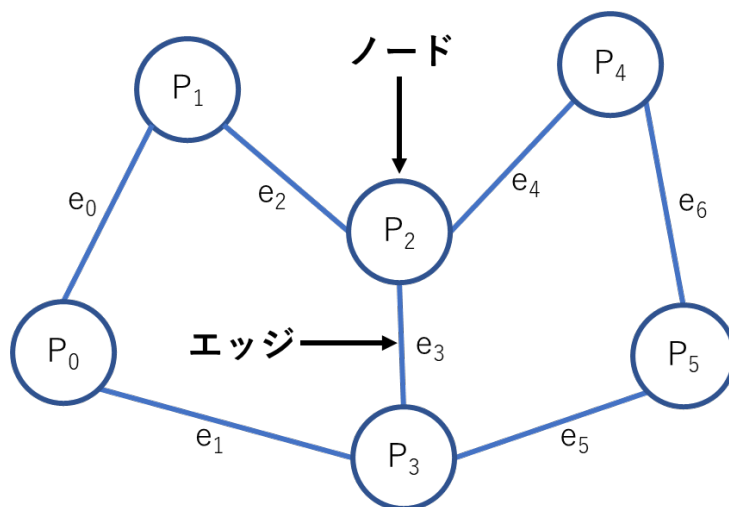


図 2.1 ノードとエッジ

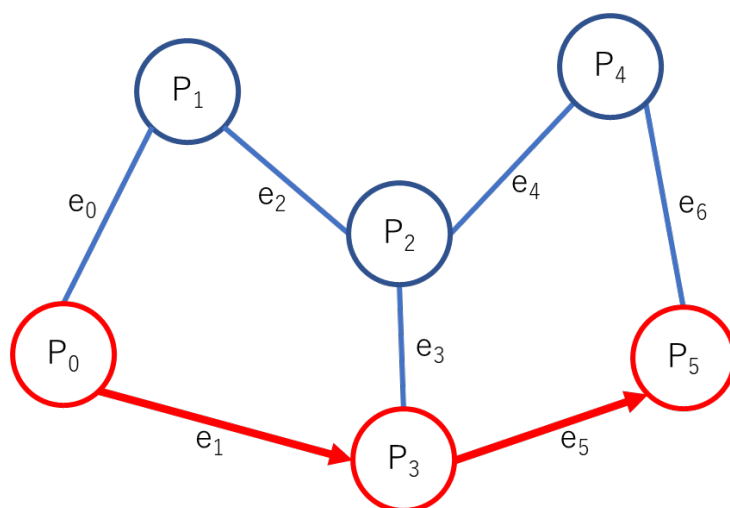


図 2.2 最短経路の一例

2.2 エッジ重要度

本研究では、経路グラフのエッジを削除することで、経路グラフのデータ削減を図る。以下の図 2.3 は、エッジ削除により経路グラフのデータ削減を行った様子を示したものである。

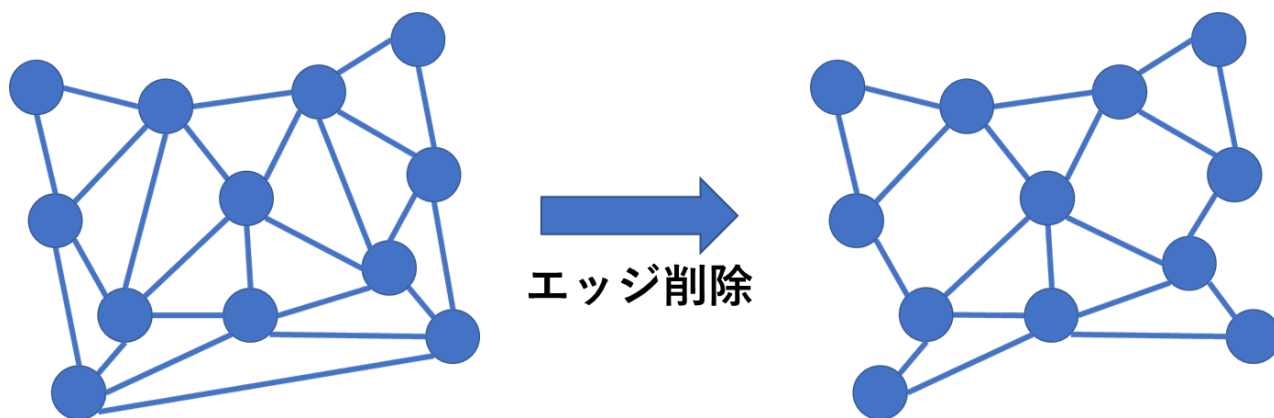


図 2.3 エッジ削除による経路グラフのデータ削減を行った様子

しかし、エッジを削除することで本来求まる経路よりも大きい長さの経路が求まることや、経由するノードの数が増えることがある。長さや経由ノードの増加が少しだけであれば大きな問題はないが、削除するエッジによっては長さや経由ノードの増加が大幅なものとなるため問題であるといえる。以下の図 2.4 は、削除するエッジにより求まる経路が長くなる場合を示したもの

である。

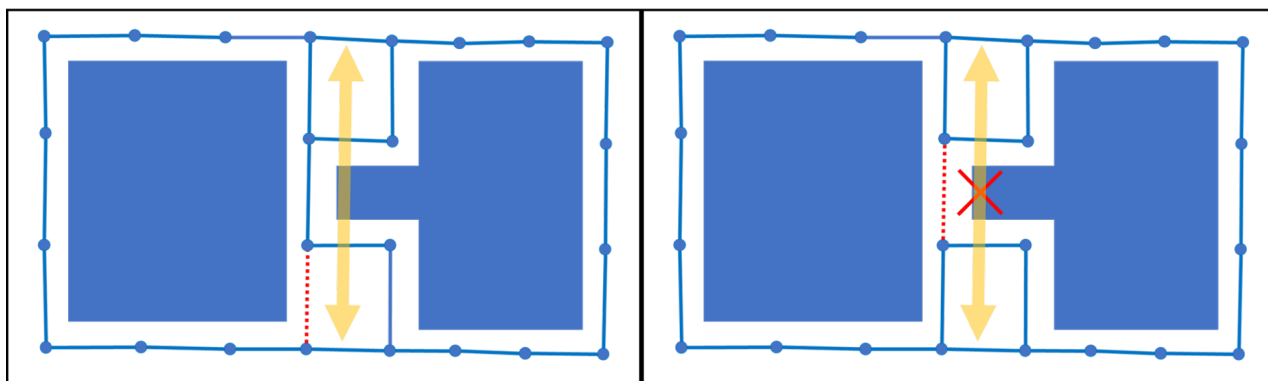


図 2.4 削除するエッジにより求まる経路が長くなる場合

図 2.4 の左側の経路グラフにおける赤いエッジを削除した場合、中央の上下の移動が可能であり求まる経路の長さも大きな変化はない。一方、右側の経路グラフにおける赤いエッジを削除した場合、中央の上下移動ができなくなるため、求まる経路の長さが大幅に増加する。

また実際の道路では、表通りや高速道路などは多くの車が往来するが、路地などは車通りが少ない場合が多い。そのためある道路を封鎖する場合について考えると、表通りより路地を封鎖した方が全体の影響は少ないといえる。

同様に経路グラフに対して経路探索を行う際も、頻繁に利用するエッジとそうでないエッジが存在する。そこで本研究では頻繁に利用するエッジほど重要であるという位置付けを行い、これをエッジ重要度と呼称する。経路グラフのデータ削減のためにエッジ削除を行う際は、エッジ重要度の低いエッジから削除を行っていくことが効果的であるといえる。

2.3 総経路長増分指標

2.3.1 総経路長増分指標による厳密重要度算出

経路グラフ G におけるノード P_i, P_j 間の最短経路の長さ $D(P_i, P_j)$ を求める。すべてのノード

間に対して最短経路を求め、その長さの合計値を算出する。本研究では、経路グラフ G の最短経路の長さの合計値を総経路長 L と呼称する。Sergio ら [30] の論文では、グラフ上のパスを表す正規表現に重みを持たせた手法を提案しており、その中で総経路長と同様の値を算出している。以下の式 (2.3) はノード数を n としたときの総経路長 L を示したものである。

$$L = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} D(P_i, P_j) \quad (2.3)$$

次に、経路グラフ G からエッジ e_k を削除した経路グラフ G_k の総経路長 L_k を求める。以下の図 2.5 は、経路グラフ G と経路グラフ G_k の例を示したものである。

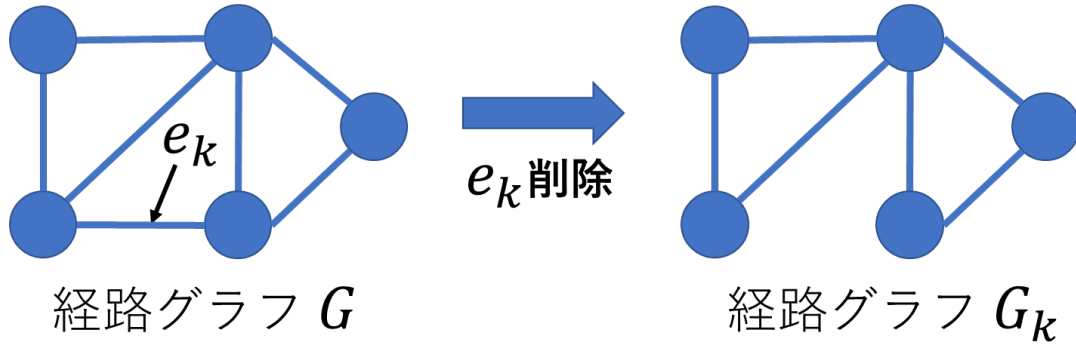


図 2.5 経路グラフ G と経路グラフ G_k の一例

このとき、総経路長の増分 $L_k - L$ の値が大きいほどエッジ削除により求まる経路の長さが大きいといえる。この総経路長の増分によるエッジ重要度算出を総経路長増分指標という。また、総経路長増分指標により求めたエッジ重要度を本研究では厳密重要度と呼称する。増分 $L_k - L$ の値が大きいほどエッジ e_k の厳密重要度は高く、値が小さいほど厳密重要度は低くなる。

2.3.2 総経路長増分指標の計算時間

総経路長増分指標により、実際にエッジを削除することによる総経路長の変化から厳密重要度を求めることができる。しかし、総経路長増分指標により厳密重要度を算出する際、処理時間が大幅に掛かってしまうという問題がある。以下の表 2.1 は、いくつかのマップで総経路長増分指

標により厳密重要度を算出した際の計算時間を示したものである。

表 2.1 総経路長増分指標による厳密重要度算出時の計算時間

ノード数	エッジ数	計算時間 (秒)
100	281	27.665
200	575	263.771
500	1470	5064.112

総経路長増分指標では経路グラフのノード・エッジ数の増加に伴い、計算時間が大幅に増加してしまう。そのため、リアルタイムな処理が必要となるゲームやシミュレーションにおいては更新が止まる大きな要因となる。また、総経路長増分指標ではダイクストラ法による最短経路の算出を総当たりで行っているため、計算量は $O(n^4)$ となる。

2.4 提案手法

本節では、提案手法によるエッジ重要度算出について述べる。提案手法を用いることで総経路長増分指標より短い処理時間で、かつ厳密重要度に近い結果を目指す。また、提案手法により求めたエッジ重要度のことを簡易重要度と呼称する。

提案手法では、以下の 3 つを用いて簡易重要度算出を行う。

- 周辺にあるエッジとの類似度
- 経路グラフの重心からの距離
- 重心への方向類似度

2.4.1 周辺にあるエッジとの類似度

まず、周辺にあるエッジについて説明する。エッジ e_i, e_j の中点 M_i, M_j が、 $|\overrightarrow{M_i M_j}| \leq T_1$ を満たすときエッジ e_j はエッジ e_i の周辺にあるエッジとする。このとき、 T_1 は閾値であり、経路グ

ラフの状態によって調整を行う。以下の図 2.6 は、エッジ e_i の周辺にあるエッジを示したものである。

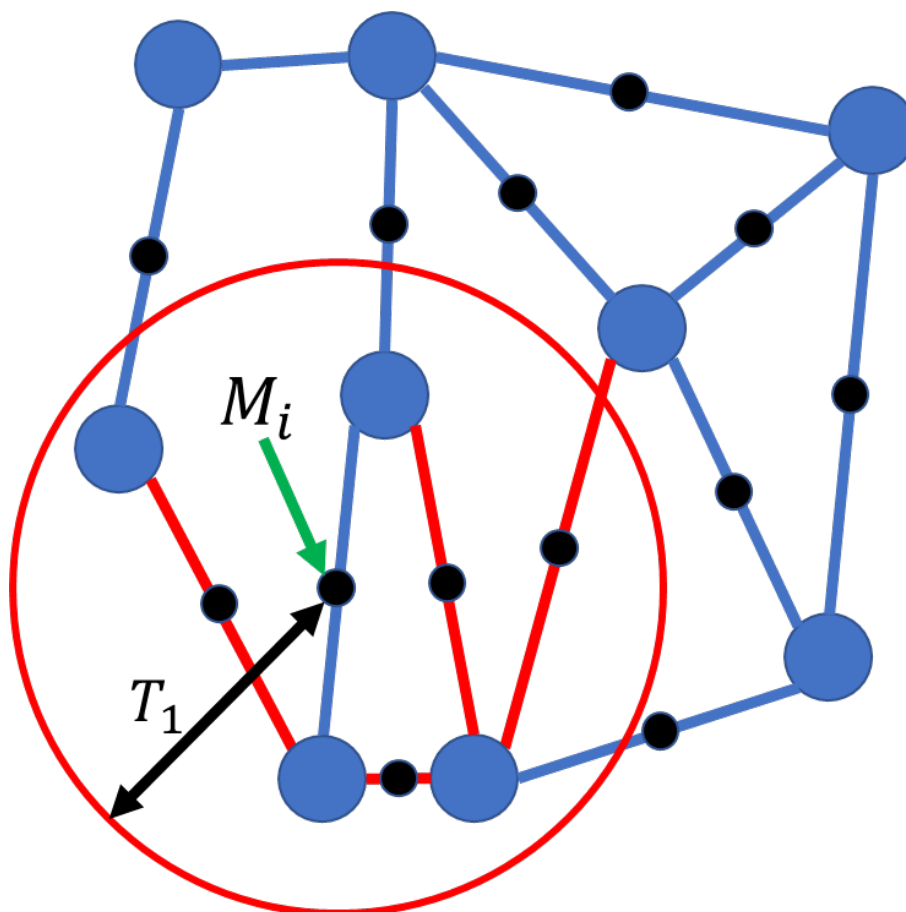


図 2.6 エッジ e_i の周辺にあるエッジ

図 2.6 における赤い円の中心がエッジ e_i の中点 M_i であり、赤いエッジがエッジ e_i の周辺にあるエッジである。この周辺のエッジはグラフの状態に限らずある程度の定数となるため、周辺のエッジ算出における計算量は $O(1)$ となる。次に、エッジの類似度について説明する。ある 2 つのエッジ e_i, e_j をみたとき、2 つのエッジの距離が近く、同じ方向を向いているほどエッジ e_i, e_j の類似度は高いとする。以下の図 2.7 は、類似度の高いエッジ同士を示したものである。

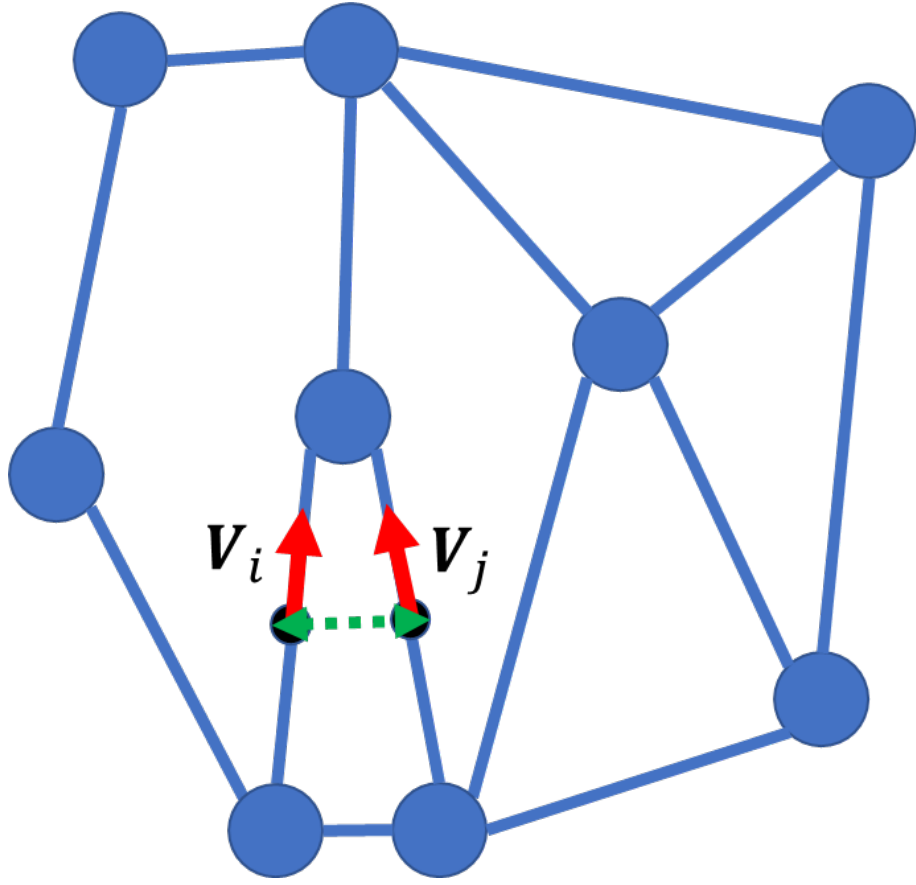


図 2.7 類似度の高いエッジ

赤いエッジ同士が類似度の高いエッジである。

エッジ e_i, e_j の向いている方向については、エッジ e_i, e_j の単位方向ベクトル $\mathbf{V}_i, \mathbf{V}_j$ の内積値の絶対値を用いて求める。絶対値をとることで方向ベクトル同士が反対方向を向いている場合でも同じ方向を向いているとみなす。

最後にエッジ e_i とその周辺のエッジ e_j に対して以下の計算を行うことで、エッジ e_i と周辺のエッジとの類似度を求める。

$$I_i = \begin{cases} \frac{|\mathbf{V}_i \cdot \mathbf{V}_j|}{|\overrightarrow{M_i M_j}|} & |\overrightarrow{M_i M_j}| \geq 1 \text{ かつ } |\mathbf{V}_i \cdot \mathbf{V}_j| \geq T_2 \text{ のとき} \\ |\mathbf{V}_i \cdot \mathbf{V}_j| & |\overrightarrow{M_i M_j}| < 1 \text{ かつ } |\mathbf{V}_i \cdot \mathbf{V}_j| \geq T_2 \text{ のとき} \\ 0 & |\mathbf{V}_i \cdot \mathbf{V}_j| < T_2 \text{ のとき} \end{cases} \quad (2.4)$$

T_2 は閾値であり、経路グラフの状態によって調整を行う。また、周辺のエッジ e_j が複数存在

する場合、そのすべてに対して I_i を求めて合計値を算出する。 I_i の値が大きいほど、周辺のエッジのとの類似度が高い。この類似度が高いほど、簡易重要度は低くなる。周辺にあるエッジとの類似度は各エッジに対して周辺にあるエッジを求めており、周辺のエッジを求める際の計算量は $O(1)$ である。そのため、周辺にあるエッジとの類似度算出時の計算量は $O(n)$ となる。

2.4.2 経路グラフの重心からの距離

まず、経路グラフの重心を求め G とする。そして、重心 G とエッジ e_i の中点 M_i との距離 $|\overrightarrow{GM_i}|$ を求める。この距離 $|\overrightarrow{GM_i}|$ をエッジ e_i の経路グラフの重心からの距離とする。また、距離 $|\overrightarrow{GM_i}|$ の値が大きい程、簡易重要度は低くなる傾向にある。この経路グラフの重心からの距離は各エッジに対して求めるため、計算量は $O(n)$ となる。以下の図 2.8 は、エッジ e_i と重心 G との距離を求めた様子を示したものである。

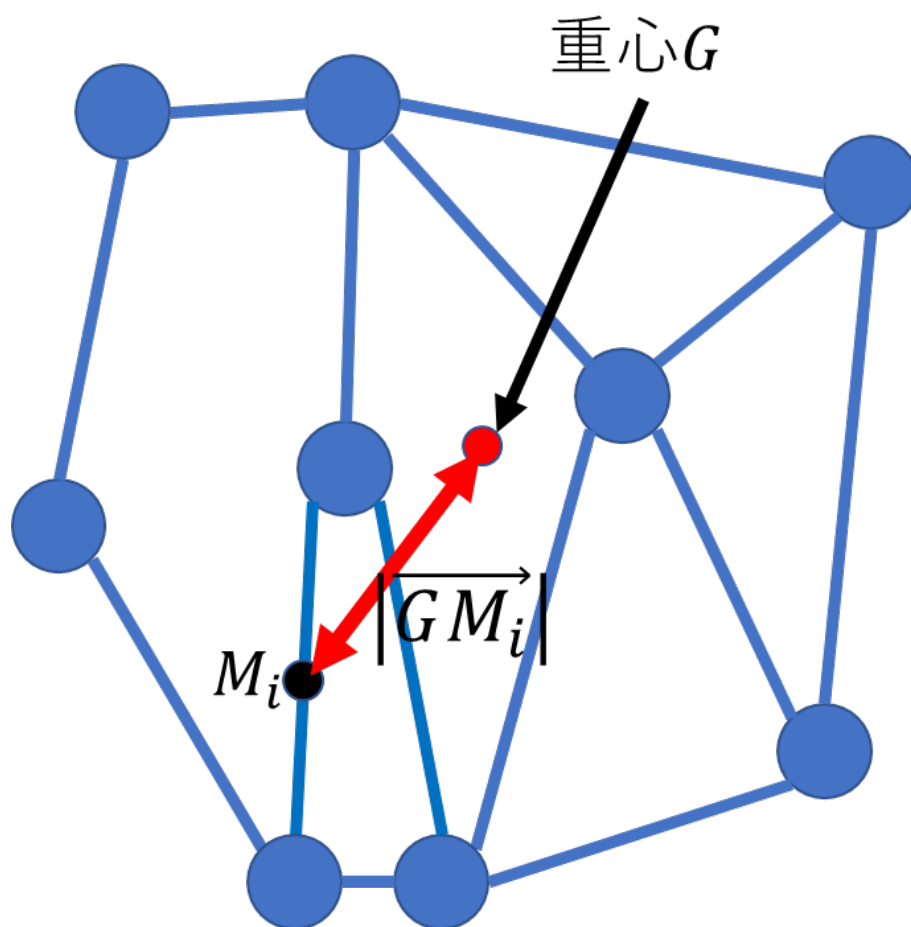


図 2.8 重心 G とエッジ e_i との距離を求めた様子

2.4.3 重心への方向類似度

まず、エッジ e_i の中点 M_i から重心 G へと向かう単位ベクトル V'_i を求める。以下の図 2.9 は、単位ベクトル V'_i の一例を示したものである。

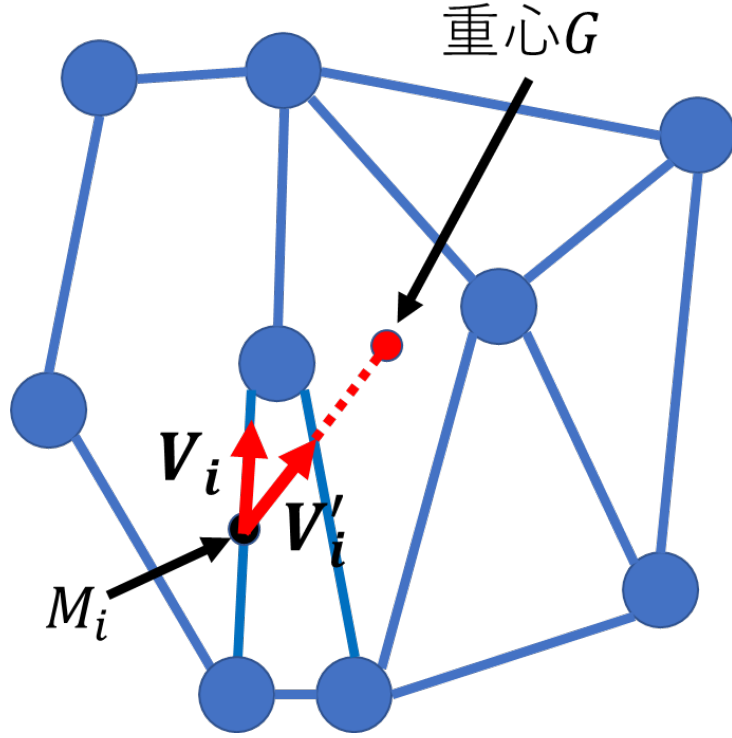


図 2.9 単位ベクトル V'_i の一例

次に、エッジ e_i の単位方向ベクトル V_i と単位ベクトル V'_i の内積値の絶対値 $|V_i \cdot V'_i|$ を求める。この内積値の絶対値 $|V_i \cdot V'_i|$ の値が大きいほど、重心への方向類似度は大きく、簡易重要度は高くなる。この重心への方向類似度は各エッジに対して求めるため、計算量は $O(n)$ となる。

2.4.4 簡易重要度の算出

最後に、エッジ e_i の簡易重要度 P_i を式 (2.5) により求める。

$$P_i = \alpha I_i + \beta |\overrightarrow{GM_i}| + \gamma |V_i \cdot V'_i| \quad (2.5)$$

α, β, γ は調整係数であり、閾値 T_1, T_2 と同様に経路グラフの状況によって調整を行うことで、簡易重要度を厳密重要度へと近づける。また簡易重要度算出に必要な周辺にあるエッジとの類似度、経路グラフの重心からの距離、重心への方向類似度の計算量はすべて $O(n)$ であるため、簡易重要度の算出時の計算量も $O(n)$ となる。

第 3 章

検証

3.1 検証準備

エッジ数 n の経路グラフ G のエッジ e_i において、総経路長増分指標を用いて求めた厳密重要度を L_i 、提案手法を用いて求めた簡易重要度を P_i とする。このとき、経路グラフ G における厳密重要度と簡易重要度の類似度 H を以下の式 (3.1) により求める。

$$H = 1 - \left(\frac{\sum_{i=0}^{n-1} |L_i - P_i|}{n} \right) \quad (3.1)$$

この類似度 H の値が 1 に近くなるほど、提案手法による簡易重要度が厳密重要度に近い結果を得ることができたといえる。

そこで、複数の経路グラフにおいて厳密重要度と簡易重要度を求め、そこから類似度 H を求めることで、提案手法がどのような経路グラフにおいて有効であるかを検証する。なお、提案手法による簡易重要度は閾値、調整係数を調整し厳密重要度に近づけたものである。

また、求めたエッジ重要度を用いてエッジに疑似カラーを設定することで、経路グラフ毎の出力結果の違いについても検証する。以下の図 3.1 は、エッジ重要度を用いてエッジに疑似カラーを設定した様子を示したものである。

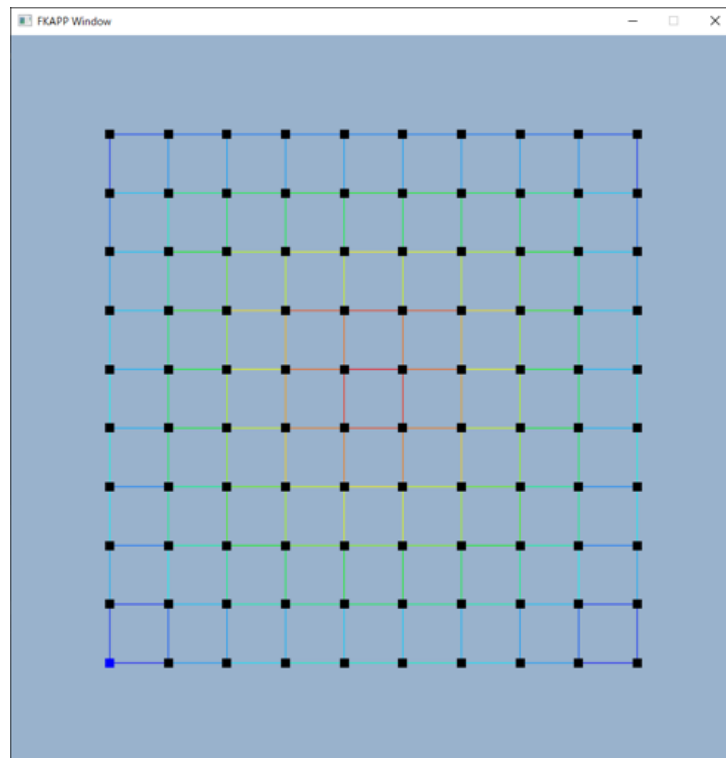


図 3.1 エッジに疑似カラーを設定した様子

3.2 検証:経路グラフ毎の疑似カラーと類似度 H

今回は 5 つの経路グラフにて厳密重要度と簡易重要度を求め、その後それらをもとに疑似カラーの設定と類似度 H の算出を行った。以下の図 3.2～3.6 は、検証用の経路グラフ 1～5 を示したものである。

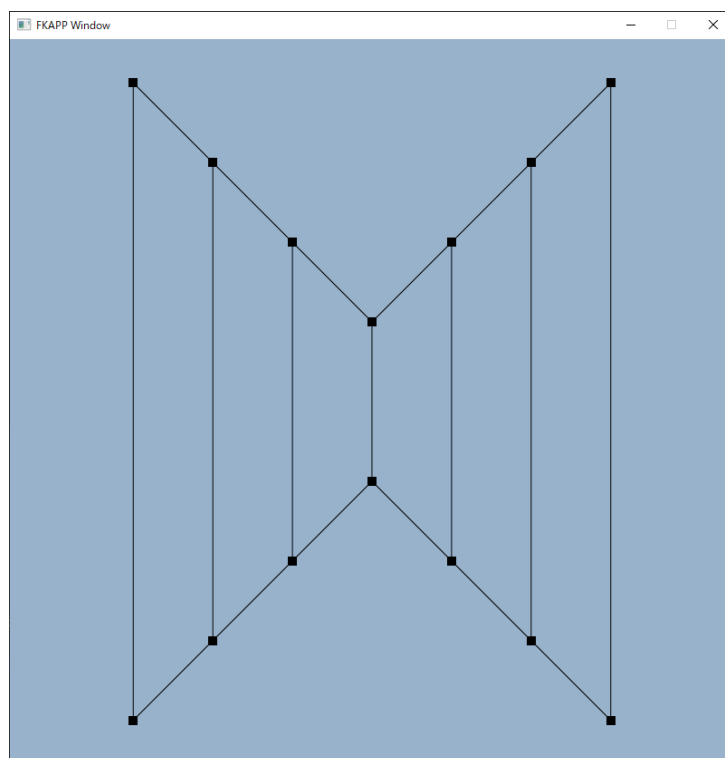


図 3.2 経路グラフ 1

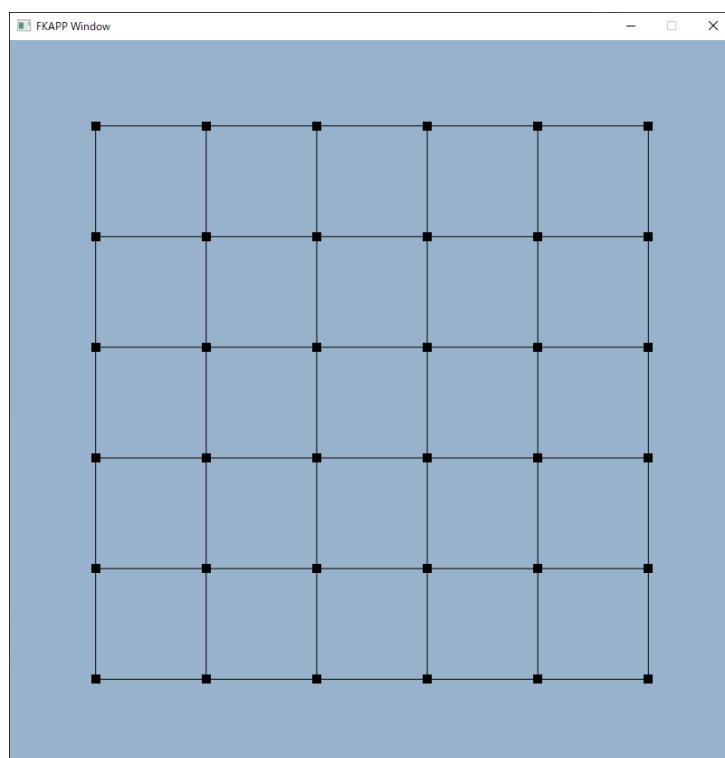


図 3.3 経路グラフ 2

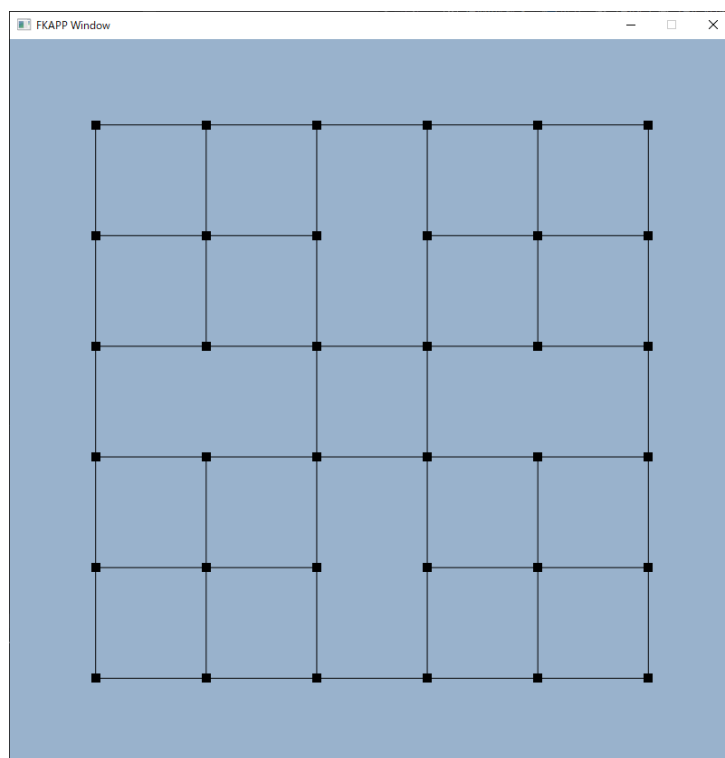


図 3.4 経路グラフ 3

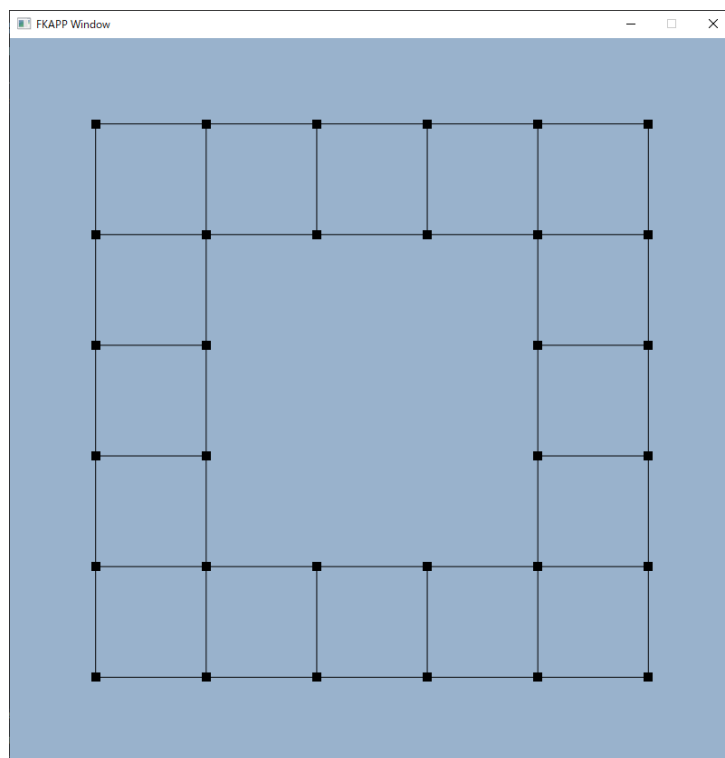


図 3.5 経路グラフ 4

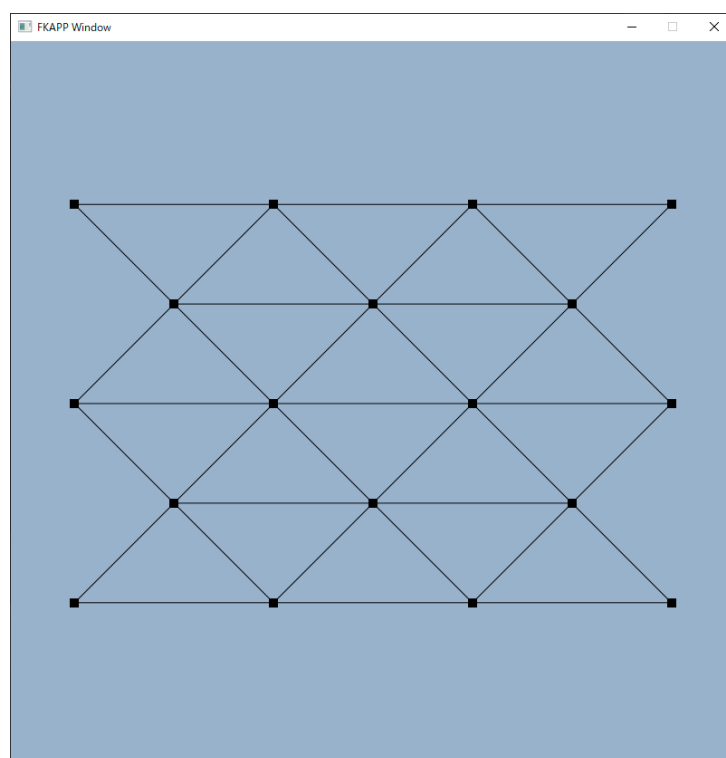


図 3.6 経路グラフ 5

3.2.1 疑似カラーの設定

以下の図 3.7～3.16 は、検証用の経路グラフ 1～5 においてエッジに疑似カラーを設定した様子
を示したものである。

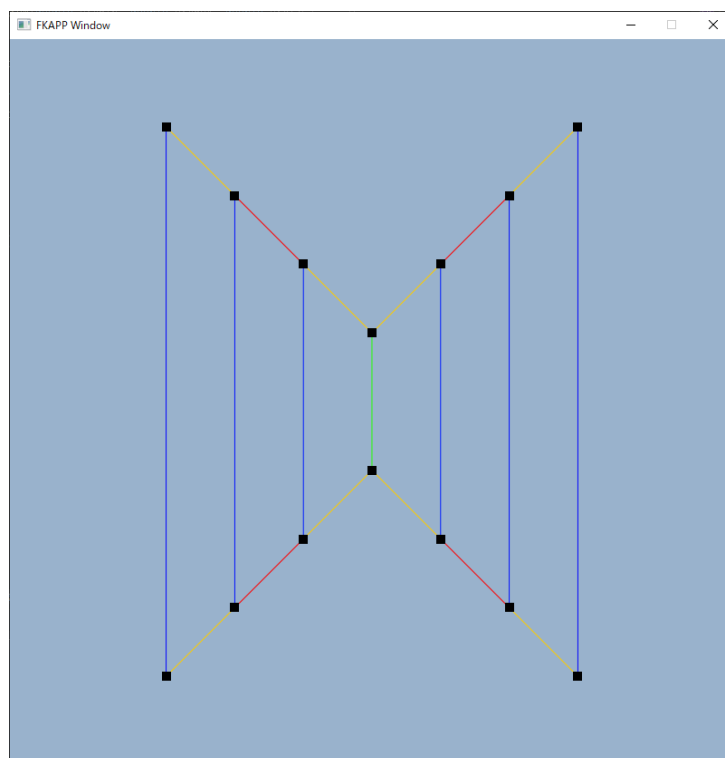


図 3.7 経路グラフ 1:総経路長増分指標

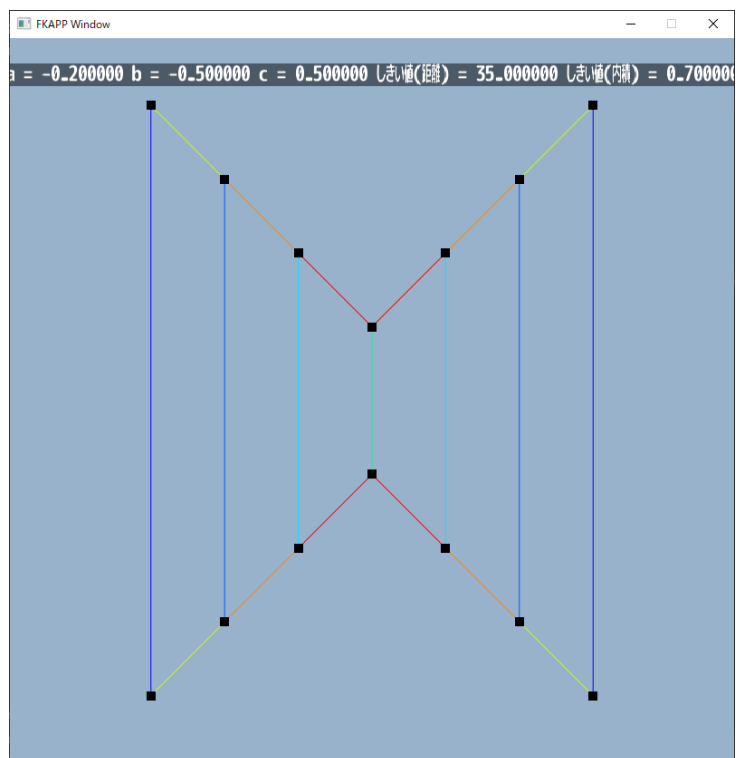


図 3.8 経路グラフ 1:提案手法

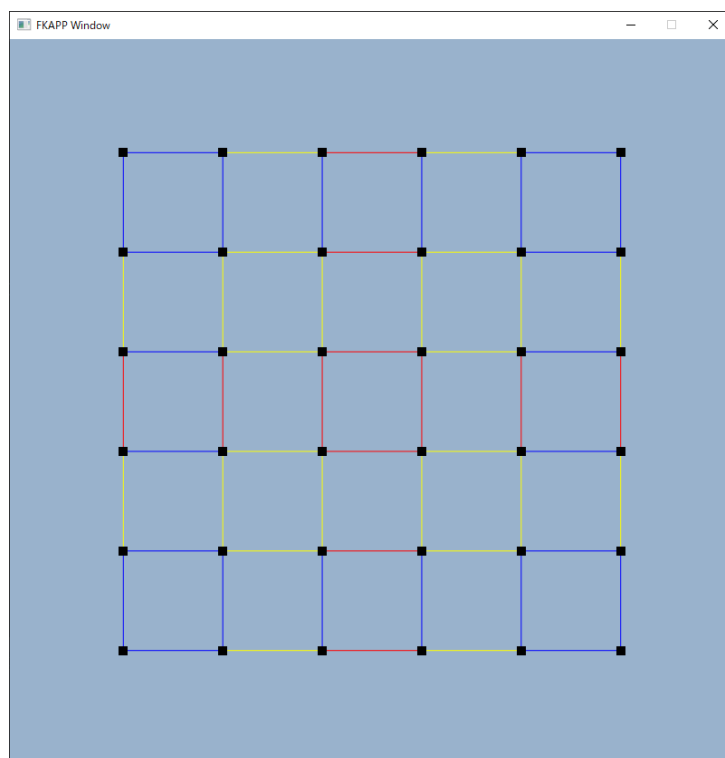


図 3.9 経路グラフ 2:総経路長増分指標

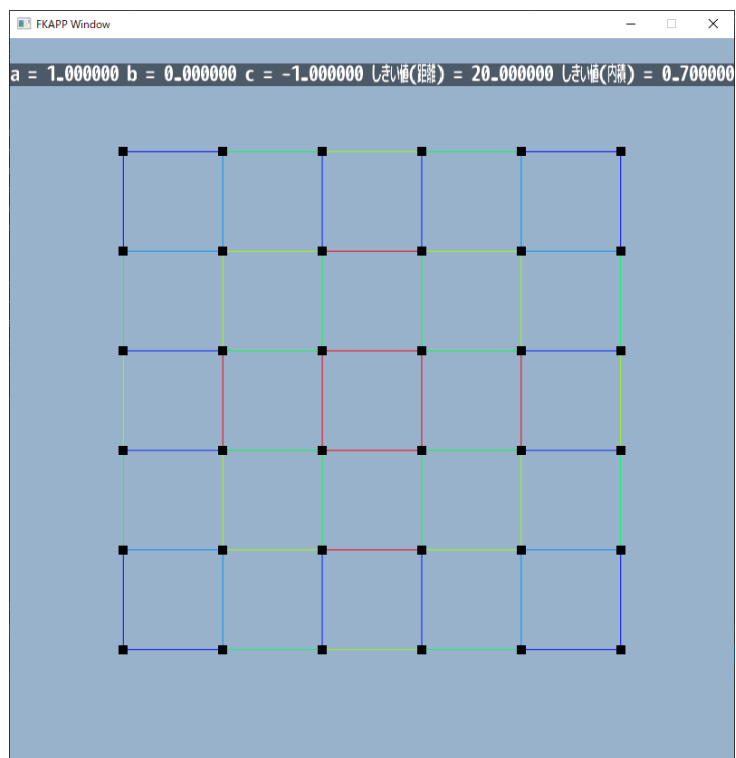


図 3.10 経路グラフ 2:提案手法

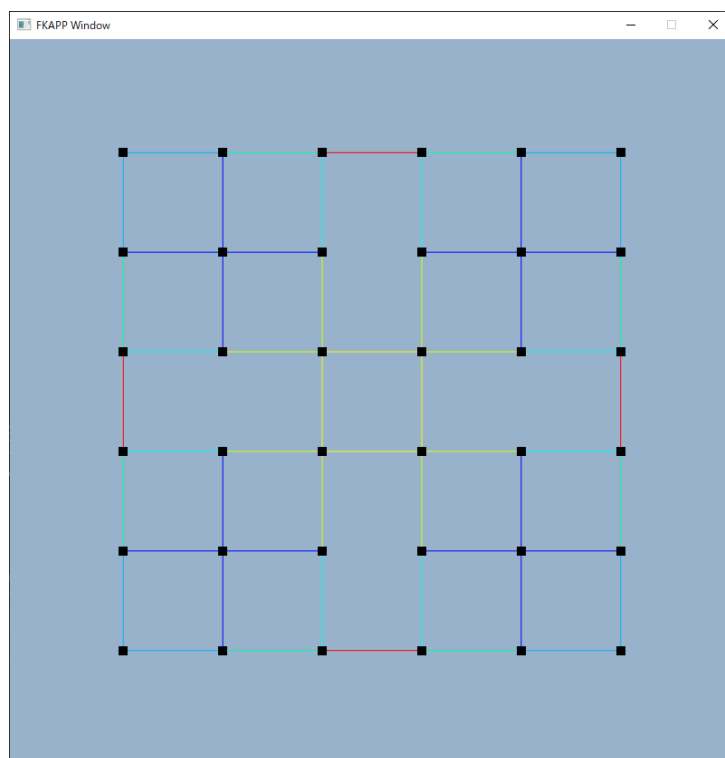


図 3.11 経路グラフ 3:総経路長増分指標

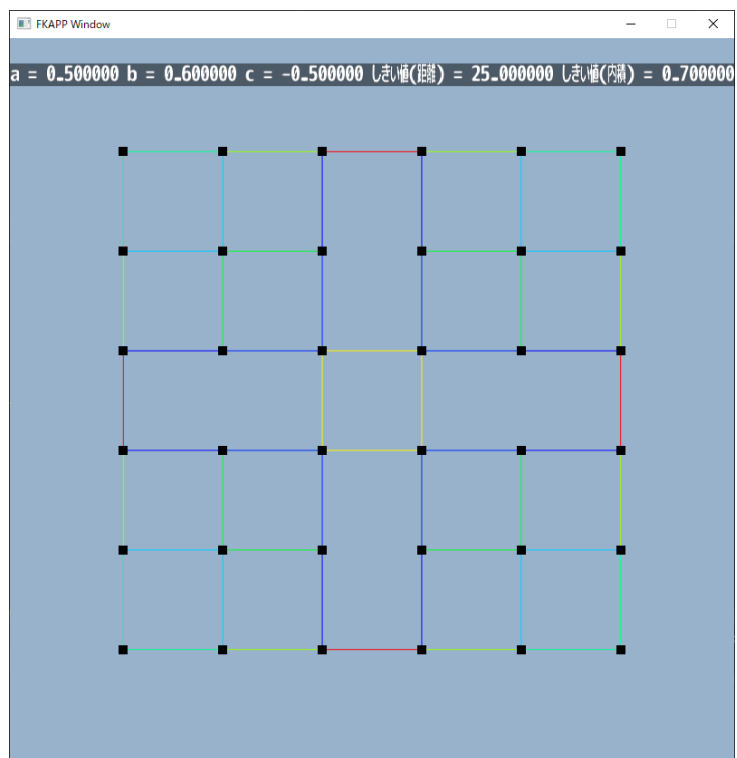


図 3.12 経路グラフ 3:提案手法

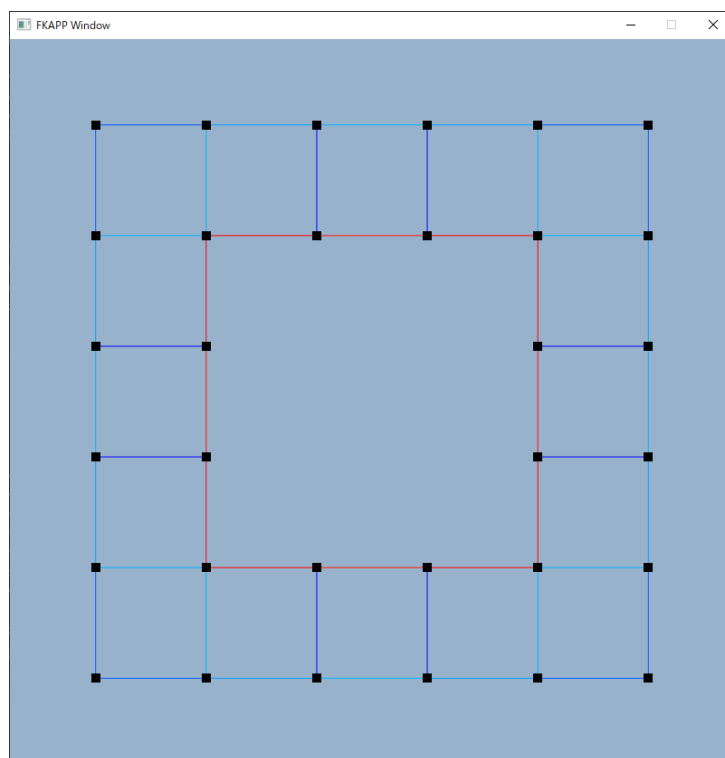


図 3.13 経路グラフ 4:総経路長増分指標

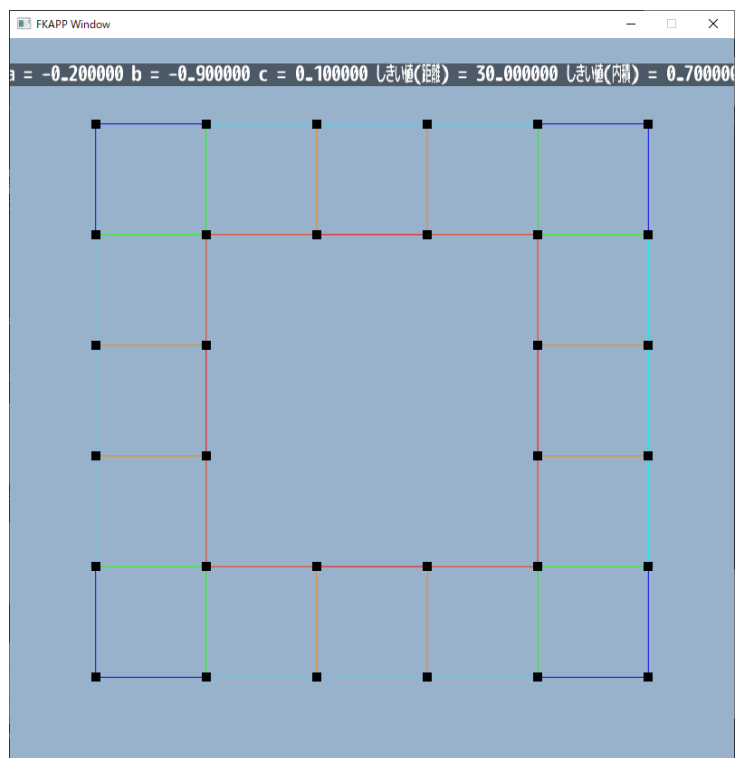


図 3.14 経路グラフ 4:提案手法

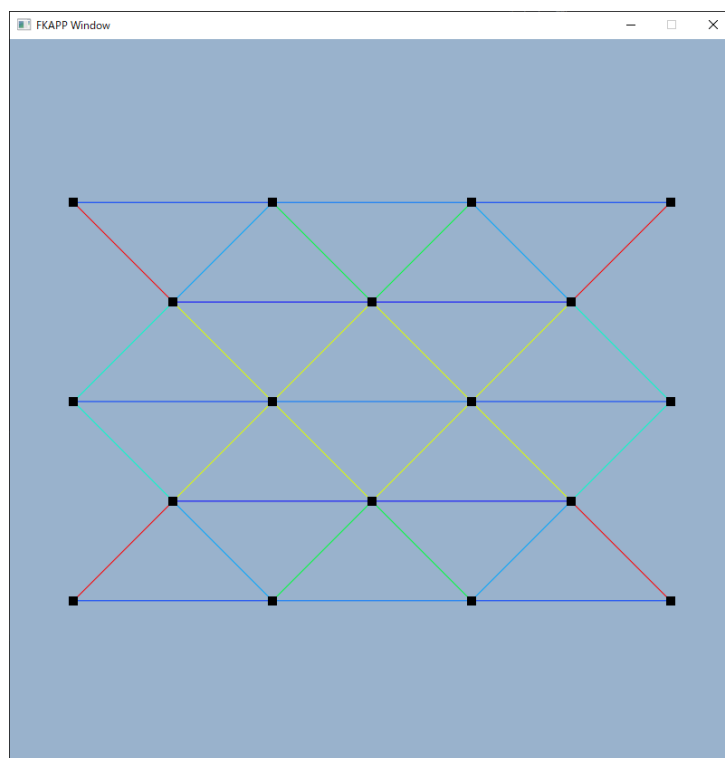


図 3.15 経路グラフ 5:総経路長増分指標

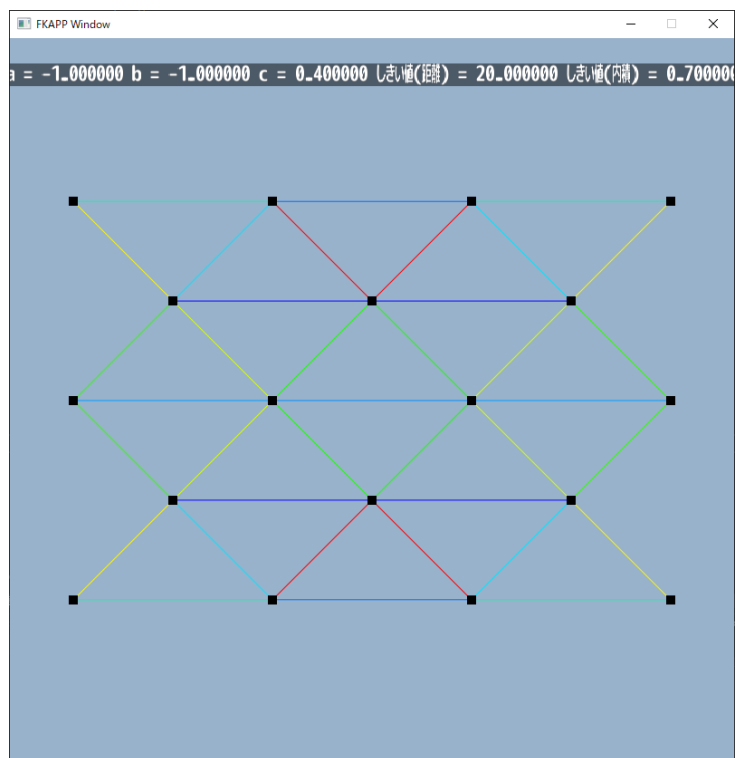


図 3.16 経路グラフ 5:提案手法

経路グラフ 1 においては、総経路長増分指標では上部分のノード同士と、下部分のノード同士を繋ぐエッジを削除した場合、必ず上下の移動を行う必要があるため、総経路長の増分が多くなる。一方で、上下のノード間を繋ぐエッジを削除しても、総経路長の増分は少ない。そのため、上下のノードを結ぶエッジの厳密重要度が低く、それ以外が高くなるという特徴があり、提案手法ではこの特徴を再現することができた。

経路グラフ 2 においては、総経路長増分指標では同じ行・列のエッジをみたとき、中央ほど厳密重要度が高く、外側にいくにつれ低くなっている。これは上下または左右のノード間の移動を考えたとき、中央に近いエッジのほど通過する回数が多くなるためである。提案手法によりこれらの特徴を再現することができた。

経路グラフ 3 においては、経路グラフ中心から上下左右にある赤いエッジや中心付近にある黄色いエッジについては、提案手法により近い結果を得られたがそれ以外のエッジについては近い結果を得ることはほとんどできなかった。

経路グラフ 4 においては、内側のエッジについては提案手法により近い結果を得られたが、外側のエッジや内側と外側を繋いでいるエッジについてはあまり近い結果を得られなかった。

経路グラフ 5 においては、総経路長増分指標では横方向のエッジの厳密重要度が低く、斜め方向のエッジは高くなるという特徴があり、提案手法ではこの特徴を再現することができた。

3.2.2 類似度の算出

以下の表 3.2 は各経路グラフにおける類似度 H を示したものである。

表 3.1 各経路グラフにおける類似度 H

経路グラフ	類似度 H
経路グラフ 1	0.86953
経路グラフ 2	0.85288
経路グラフ 3	0.70221
経路グラフ 4	0.75213
経路グラフ 5	0.82110

経路グラフ 1・2・5 は類似度が 0.8 以上となっており、簡易重要度が厳密重要度に近い結果を得ることが概ねできたといえる。しかし、経路グラフ 3・4 は類似度が 0.8 未満となっており、経路グラフ 1・2・5 と比べるとあまり近い結果が得られなかった。この経路グラフ毎による類似度 H の出力結果の違いについての要因は現状不明である。また、提案手法における閾値・係数の調整は手探りでを行っているためより高い類似度が得られる可能性があり、これらの適切な設定については課題である。

3.3 総経路長増分指標・提案手法による処理時間比較

本提案手法の有用性を評価するために、総経路長増分指標と提案手法を用いてエッジ重要度を算出し、計算時間の計測を行った。以下の表 3.2 は検証に用いた PC のスペックを示したものである。

表 3.2 検証用 PC

OS	Windows 10
CPU	Intel(R) Core(TM)i7-7700K CPU @ 4.20GHz
メモリ	16 GB
GPU	GeForce GTX 1080 Ti

本項では、3 種類の異なるノード・エッジ数の経路グラフに対して検証を行った。以下の表 3.3 は総経路長増分指標と提案手法を用いてエッジ重要度を算出した際の計算時間を示したものである。

る。なお、時間単位はミリ秒である。

表 3.3 総経路長増分指標・提案手法によるエッジ重要度算出の計算時間

ノード数	エッジ数	総経路長増分指標	提案手法
100	281	27665	5
200	575	263771	18
500	1470	5064112	80

総経路長増分指標の場合、ノード・エッジ数の増加に伴い計算時間が大幅に増加している。これは、厳密重要度算出の為に必要な総経路長の算出がノード・エッジ数に大きく影響しているためである。一方提案手法では、ノード・エッジ数の増加に伴う計算時間の増加が、総経路長増分指標と比べるとかなり少なくなっている。これは、提案手法による簡易重要度の算出にノードを用いないためであると考えられる。また、提案手法においてノード・エッジ数の増加に伴う計算時間の増加量がやや大きくなっている。これは、ある空間内のノード・エッジの密度を増加させることで、ノード・エッジ数の増加を図ったことにより、周辺にあるエッジとの類似度を求める際の計算量が増加したためだと考えられる。

3.4 提案手法による大規模な経路グラフでの計算時間計測

本項では、6種類の異なるノード・エッジ数の経路グラフに対して、提案手法により簡易重要度を算出しその計算時間について計測する。3.3節とは違い、より大規模な経路グラフに対して検証を行う。それにより、経路グラフのデータ削減がより必要である大規模なマップでの利用や、高速な処理を必要とするゲームでの利用における提案手法の有用性を評価する。以下の表 3.4 は大規模な経路グラフでの提案手法による簡易重要度算出の計算時間を示したものである。なお、時間単位はミリ秒である。

表 3.4 提案手法による大規模な経路グラフでの計算時間計測

ノード数	エッジ数	提案手法
100	281	5
200	575	18
500	1470	80
1000	2952	299
2000	5938	1208
5000	14862	7151

提案手法を用いることで、ある程度の大きさの経路グラフであれば高速な処理を必要とするゲーム分野に対しても適用ができるといえる。また、ノード数 1000 以上の大規模な経路グラフにおいても、ある程度の計算時間が確保できるのであれば提案手法は有効である。

第 4 章

まとめ

経路探索の処理時間の削減は、高速に処理を必要とするゲーム分野において重要である。経路探索の処理時間削減のために、経路探索アルゴリズムを改良する方法と経路探索に用いるグラフデータを削減する方法がある。本論文では、経路探索の処理時間削減のために汎用的な経路グラフのデータ削減手法について提案した。経路グラフにおけるエッジに対して、頻繁に利用するエッジほど重要であると位置付けたエッジ重要度というものを設定し、エッジ重要度に基づいてエッジを削除することで効果的に経路グラフのデータ削減を図った。まず、総経路長増分指標によりエッジ重要度を求めこれを厳密解である厳密重要度とした。しかし、総経路長増分指標では処理時間が大幅に掛かってしまうため、短い処理時間で総経路長増分指標で求めた厳密重要度に近い結果を得るための手法を提案した。複数の経路グラフにおいて厳密重要度と簡易重要度を求めたところ、いくつかの経路グラフでは厳密重要度に近い結果を得ることができた。しかし、経路グラフによる出力結果の違いの要因や簡易重要度算出に必要となる係数・閾値等の適切な調整方法については課題である。また、提案手法に対してエッジ重要度算出に掛かる計算時間の計測を行った。ある程度の大きさの経路グラフであれば高速な処理を必要とするゲーム分野に対しての適用が可能である。また、大規模なマップにおいてもある程度の計算時間が確保できる状況であれば提案手法は有効であると考えられる。

謝辭

本論文を執筆するにあたり、多くのアドバイスや指導をしてくださった先生方、手伝ってくださった研究室の先輩方や友人に感謝を述べたいと思います。

渡辺先生には多くの助力をいただきました。研究の指導や相談、研究に必要なプログラムの手伝い、学会投稿への添削や準備など、多くのご指導やご助力のおかげでここまで研究を進めることができました。本当にありがとうございました。

阿部先生には多くの助力をいただきました。研究室のミーティングでは研究を進めるにあたって重要なことを数多く教えていただき、研究を進めるのにとっても参考になりました。本当にありがとうございました。

柿本先生と三上先生には多くの助言をいただきました。予備審査や修士1年の時の合同ミーティングなどでは多くの助言やご指摘をいただきました。本当にありがとうございました。

飯沼先生には、予備審査で普段とは異なる視点からの助言やご指摘をいただきました。本当にありがとうございました。

研究室の先輩方や友人には、研究の相談や論文の添削だけでなく学生生活においても多くの助力をいただきました。本当にありがとうございました。

最後に改めて、ご指導いただきました先生方、手伝ってくださった研究室の先輩方や友人に心から感謝致します。

本当にありがとうございました。

参考文献

- [1] 独立行政法人工業所有権情報・研修館. 平成 16 年度 特許流通支援チャート カーナビ経路探索技術. <https://www.inpit.go.jp/blob/katsuyo/pdf/chart/fdenki22.pdf>. 参照: 2021.11.20.
- [2] 三宅陽一郎. デジタルゲームにおける人工知能技術の応用の現在. 人工知能, Vol. 30, No. 1, pp. 45–64, 2015.
- [3] 均狩野, 小塚英城. CA 法による広域道路交通シミュレータを用いた経路案内方式の評価. Technical Report 83(2002-ITS-010), 筑波大学 電子情報工学系, 筑波大学 電子情報工学系, sep 2002.
- [4] Shin nyeong Heo, Sheng yu Lu, Ji-sun Shin, and Hee-hyol Lee. Multi-robot-multi-target path planning and position estimation for disaster area. In *2018 International Conference on Information and Communication Technology Robotics (ICT-ROBOT)*, pp. 1–4, 2018.
- [5] 戸田肇, 渡部慶二. 移動ロボットの経路探索と空間認識に関する研究. 計測自動制御学会東北支部 第 170 回研究集会, Vol. 170, No. 5, pp. 1–9, 1997.
- [6] 藤本敬介, 守屋俊夫, 中山泰一. 車輪移動ロボットの経路探索の高速化のためのマップリンク法. 日本ロボット学会誌, Vol. 33, No. 8, pp. 630–641, 2015.
- [7] 直之佐藤, Temsiririrkkul Sila, Huu Phuc Luong, 心池田. Influence map を用いた経路探索による人間らしい弾避けのシューティングゲーム AI プレイヤ. ゲームプログラミングワークショップ 2016 論文集, 第 2016 巻, pp. 57–64, oct 2016.
- [8] NAVITIME JAPAN Co. 国内初！新しい道路を自動で地図に反映する「地図自動更新機能」を提供開始 | プレスリリース/お知らせ | ナビタイムジャパン. https://corporate.navitime.co.jp/topics/pr/201612/13_3985.html. 参照: 2021.11.20.
- [9] 森畑明昌, 松崎公紀, 武市正人. 領域限定言語に基づく最適経路問合せ. 情報処理学会論文誌プログラミング, Vol. 4, No. 2, pp. 116–133, 2011.

- [10] E.W.Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, Vol. 1, pp. 269–271, 1959.
- [11] 岩通ソフトシステム株式会社. ダイクストラ法アルゴリズム. <https://www.iwass.co.jp/column/column-03.html>. 参照: 2021.11.20.
- [12] James R. Driscoll, Harold N. Gabow, Ruth Shrairman, and Robert E. Tarjan. Relaxed heaps: An alternative to fibonacci heaps with applications to parallel computation. *Commun. ACM*, Vol. 31, No. 11, p. 1343–1354, nov 1988.
- [13] Michael L. Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, Vol. 34, No. 3, p. 596–615, jul 1987.
- [14] RICHARD BELLMAN. On a routing problem. *Quarterly of Applied Mathematics*, Vol. 16, No. 1, pp. 87–90, 1958.
- [15] 横山秀, 金子知適. ゲーム木に基づく並列探索での下位曲面の分担. *The 21st Game Programming Workshop 2016*, 2016.
- [16] K. Hoki and T. Kaneko. Large-scale optimization for evaluation functions with minimax search. *Journal of Artificial Intelligence Research*, Vol. 49, pp. 527–568, 2014.
- [17] M. Winands et al. Monte-carlo tree search solver. *CG, Springer-Verlag*, pp. 25–36, 2008.
- [18] 稲垣潤, 長谷山美紀, 北島秀夫. 遺伝的アルゴリズムを用いた複数経由点を伴う経路探索法. 電子情報通信学会論文誌 (D-I), Vol. J83-D-I, No. 5, pp. 504–507, 2000.
- [19] 健太原, 荘一塚原, 均狩野, 敏弘曾田, 久黒河. 多目的遺伝的アルゴリズムを用いたカーナビゲーションのための予測経路探索. Technical Report 22(2006-ITS-024), 筑波大学, 筑波大学, 筑波大学, 松下電器産業株式会社, 松下電器産業株式会社, mar 2006.
- [20] Nintendo. ゼルダの伝説 ブレス オブ ザ ワイルド — nintendo switch / wii u — nintendo. <https://www.nintendo.co.jp/zelda/index.html>. 参照: 2022.02.12.

- [21] Sony Interactive Entertainment LLC. Ghost of tsushima - about tsushima. https://www.nagasaki-tabinet.com/static/ghost_of_tsushima/. 参照:2022.02.15.
- [22] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, pp. 100–107, 1968.
- [23] Anthony Stentz. Optimal and efficient path planning for partially known environments. *IEEE International Conference on Robotics and Automation*, 1994.
- [24] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, p. 19–26, New York, NY, USA, 1993. Association for Computing Machinery.
- [25] Paul S. Heckbert and Michael Garland. Optimal triangulation and quadric-based surface simplification. *Comput. Geom.*, Vol. 14, pp. 49–65, 1999.
- [26] 長谷川誠. 『LEFT ALIVE』における地形表現とナビゲーション ai. https://cedil.cesa.or.jp/cedil_sessions/view/2065. 参照: 2022.1.20.
- [27] Inc. Rockstar Games. Grand theft auto v. <https://www.rockstargames.com/V/jp>. 参照: 2021.11.23.
- [28] SPIKE CHUNSOFT. Ark: Survival evolved — スパイク・チュンソフト. <https://www.spike-chunsoft.co.jp/ark/>. 参照:2021.11.23.
- [29] 石村園子. やさしく学べる離散数学. 共立出版株式会社, 2007.
- [30] Sergio Flesca, Filippo Furfaro, and Sergio Greco. Weighted path queries on semistructured databases. *Inf. Comput.*, Vol. 204, No. 5, p. 679–696, may 2006.

発表実績

ポスター発表

1. 小坂大樹, 阿部雅樹, 渡辺大地, 経路探索における削減位相操作を用いた経路グラフ最適化に関する研究, 映像表現・芸術科学フォーラム, 2021.

口頭発表

1. 小坂大樹, 阿部雅樹, 渡辺大地, 経路探索における部分グラフを用いた経路グラフ最適化に関する研究, 第 30 回デジタルコンテンツクリエーション合同研究発表会, 2022.