1H1-4

区間コストの公平性を考慮する経路最適化の検討

A Study on Route Optimization Considering Fairness Among Partial Paths

松井 俊浩　　松尾 啓志
Toshihiro Matsui　　Hiroshi Matsuo

名古屋工業大学
Nagoya Institute of Technology

In route optimization tasks, shortest path problems are generally considered. The problems are solved using several optimization algorithms such as A* algorithm, which is based on tree search and dynamic programming. In several practical cases, cost values should be minimized for individual parts of paths simultaneously. Such problems can be considered as multi-objective problems for the partial paths. To represent this class of problems, we employ leximax based criterion, which consider the worst cases and unfairness among the cost values of partial paths. The influence of the proposed criterion is experimentally evaluated.

1. はじめに

エージェントの行動経路の最適化は一般に，経路上の区間の合計コストを最小化することを目的とする．基本的な経路最適化手法に，分枝限定法と動的計画法に基づく A*アルゴリズムがある．その一方で，実際的な問題においては，各区間のコストやリスクを平準化することが求められる場合があると考えられる．このような問題は各区間のコストを目的とする多目的最適化問題と考えられる．個々の効用を最大化する多目的最適化問題において，公平性を考慮する最適化の指標に lexmin がある．leximin は効用値を昇順に整列した目的ベクトルについての辞書順序に基づいて定義され，多目的問題の目的ベクトルをスカラ化して比較する社会厚生の指標に相当する．leximin についての最大化は，最小の利得を最大化し，かつ，公平性をある程度改善する．この leximin の最適化は，動的計画法に分解できる．経路最適化手法である A*アルゴリズムは動的計画法に基づくため，leximin と同様の指標を経路の評価に導入できると考えられる．そこで本研究では，このような指標にもとづき，各区間の公平性を改善しつつコストを最適化する経路探索手法について検討する．

2. 準備

2.1 経路最適化問題

経路最適化問題は最短経路問題に基づく．問題はグラフ $G = \langle V, E \rangle$ 上で定義される．V は頂点の集合，E は辺の集合である．各辺 $e_j \in E$ には，正のコスト値 w_j が対応付けられる．

ある始点 $v_s \in V$ から，ある終点 $v_g \in E$ までのパスのうち，そのパスに含まれる各辺のコスト値を結合した値が最小となるパスを最適経路とする．一般には，辺のコスト値の結合は加算であり，パスに含まれるコスト値の合計が経路のコスト値となる．経路コストが最小となる最適経路を求めることが目的である．

本研究では，コスト値は例えば数個から十個などの十分に少ない数の離散値で表現されるものとする．また，次節の A*アルゴリズムにおける発見的距離関数を簡単にするために，図 1 に示されるような 4 近傍のグリッド状のグラフを対象とする．

連絡先: 松井 俊浩，名古屋工業大学，〒 466-8555 愛知県名古屋市昭和区御器所町，matsui.t@nitech.ac.jp

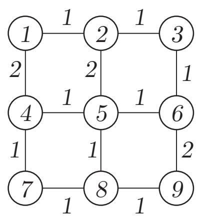

図 1: グリッド状のグラフ

図の頂点の数字はその名前を表し，辺の数字はそのコスト値を表す．

2.2 A*アルゴリズム

A*アルゴリズムは分枝限定法と動的計画法に基づく経路探索アルゴリズムである．経路を始点から終点まで探索しつつ，各頂点についての経路コストの推定値を動的計画法により更新し，終点が発見されればそれまでの情報から経路コストが最小となる最適経路を得る．探索木においては，各頂点についての経路コストの推定値をもとに，展開する頂点を選択する．各頂点 $v_i \in V$ についての経路コストの推定値は，始点からその頂点までの経路コスト $g(v_i)$，発見的距離関数に基づくその頂点から終点までの経路コスト $h(v_i)$ の合計として計算される．グリッド状のグラフの場合は，辺のコストの下限値とマンハッタン距離に基づいて発見的距離関数を定義できる．アルゴリズムの詳細については文献 [Russell 03] などを参照されたい．

2.3 各区間のコストの公平性

各区間の最大コストを平準化しつつ削減することを考える．図 1 の例において，始点 v_s を 1 の頂点，終点 v_g を 9 の頂点とするとき，従来の合計距離の最小化では，最適経路の一つは頂点 $1, 2, 3, 6, 9$ の順であり，その合計コストは 5 である．最大コストを改善しつつ経路コストを削減する場合は，最適経路は頂点 $1, 2, 3, 6, 5, 8, 9$ の順であり，その合計コストは 6 であるが，この経路にはコスト値 2 が含まれない．ここでは，単に一つの区間の最大コストを改善するのではなく，全体として各区間のコストを平準化しつつ削減することを目的とする．このような経路は，たとえば各区間に対応する周辺住民や設備の不満や消耗が度々尖鋭化する状況を抑制する場合に考慮されうる．

2.4 多目的最適化問題

多目的最適化問題は複数の目的関数の値を同時に最適化することを目的とする．ここでは次のような多目的制約最適化問題 MOCOP を考える．

定義 1 (MOCOP) *MOCOP* は $\langle X, D, F\rangle$ により定義される．X は変数の集合，D は変数の値域の集合，F は目的関数の集合である．変数 $x_i \in X$ は有限離散集合 $D_i \in D$ の要素である変数値をとる．変数の集合 $X_i \subseteq X$ について，関数 $f_i \in F$ は $f_i(x_{i,1}, \cdots, x_{i,k}) : D_{i,1} \times \cdots \times D_{i,k} \to \mathbb{N}$ のように定義される．ただし $x_{i,1}, \cdots, x_{i,k} \in X_i$ である．$f_i(x_{i,1}, \cdots, x_{i,k})$ を簡単に $f_i(X_i)$ と表す. 各目的関数の値をある指標のもとで最適化することが目的である．

各目的関数の値の組合わせは目的ベクトルにより表される．

定義 2 (目的ベクトル) 目的ベクトル $\mathbf{v}$ は $[v_1, \cdots, v_K]$ のように定義される．ただしある割り当て $\mathcal{A}$ についての v_j は $v_j = f_j(\mathcal{A}_{\downarrow X_j})$ である．

ここでは，理想的には全ての目的関数値についての最大化を目的とする．しかし，一般に目的の間にはトレードオフがあるため，同時には最大化できない．そのため，目的ベクトル間のパレート従属性にもとづくパレート最適解を選択する [Sen 97, Marler 04].

2.5 leximin

一般にパレート最適解は複数あり，それらの選択の指標として社会厚生の指標やスカラ化関数が用いられる [Sen 97, Marler 04]．従来の社会厚生である合計 $\sum_{j=1}^{K} f_j(X_j)$ では効率性が考慮される．合計の最大化はパレート最適であるが，公平性は考慮されない．*Maximin* は最小の目的の値 $\min_{j=1}^{K} f_j(X_j)$ を最大化する．これにより最悪の場合の目的を改善するが，パレート最適性は保証されない．パレート最適性を保証するためには合計などによるタイブレークが必要である．また最悪値の改善のみが考慮され，その他の目的の値は区別されない．

Leximin は昇順に整列された目的ベクトルについての辞書順にもとづく順序関係である [Bouveret 09, Matsui 14, Matsui 15].

定義 3 (整列された目的ベクトル) 整列された目的ベクトル $\mathbf{v}$ は値が昇順に整列された目的ベクトルである．

定義 4 (Leximin) $\mathbf{v} = [v_1, \cdots, v_K]$ と $\mathbf{v}' = [v'_1, \cdots, v'_K]$ を長さ K の整列された目的ベクトルとするとき，$\prec_{leximin}$ による順序は次のように定義される．$\exists t, \forall t' < t, v_{t'} = v'_{t'} \wedge v_t < v'_t$ であり，かつそのときのみ $\mathbf{v} \prec_{leximin} \mathbf{v}'$ である．

すなわち lexmin は最小値による比較を順に繰り返す指標である．Leximin についての最大化は maximin のサブセットであるため最悪の場合を改善する．さらに，ある程度の公平性が考慮され，パレート最適である．

整列された目的ベクトルの加算はベクトルの連結と再整列により定義される．

定義 5 (整列された目的ベクトルの加算) $\mathbf{v}$ と $\mathbf{v}'$ をベクトル $[v_1, \cdots, v_K]$ と $[v'_1, \cdots, v'_{K'}]$ とするとき，二つのベクトルの合計 $\mathbf{v} \oplus \mathbf{v}'$ は $\mathbf{v}'' = [v''_1, \cdots v''_{K+K'}]$ である．ただし $\mathbf{v}''$ の値は $\mathbf{v}$ と $\mathbf{v}'$ に含まれる全ての値からなる. $\mathbf{v}''$ の値は昇順に整列される．

整列された目的ベクトルの加算について次の不変性が成り立つ [Matsui 14].

命題 1 (leximin の不変性) $\mathbf{v}$ と $\mathbf{v}'$ を同じ長さの整列された目的ベクトルとする．また，$\mathbf{v}''$ を別の整列された目的ベクトルとする．$\mathbf{v} \prec_{leximin} \mathbf{v}'$ であるとき，$\mathbf{v} \oplus \mathbf{v}'' \prec_{leximin} \mathbf{v}' \oplus \mathbf{v}''$ である．

この不変性により，最適化問題のための動的計画法を構成できる [Matsui 14, Matsui 15]．ただし，組み合わせ最適化問題を同じ長さのベクトルについての部分問題に分解することが前提である．

また，整列された目的ベクトルは，目的の値とその数のペアを整列したベクトルとして表現できる [Matsui 14]．これは，連長圧縮あるいは整列されたヒストグラムに相当する．この表現において leximin の比較およびベクトルの加算を直接的に行なうことは容易である．

3. 各区間の公平性を考慮する経路最適化

各区間の公平性として，コスト値の平準化を考慮する最適化問題の解法に取り組む．先述の lexmin に基づく最適化問題を動的計画法に基づいて分解できることから，A*アルゴリズムにおけるコスト値の結合を，合計から整列ベクトルの結合に置き換えることを試みる．

組み合わせ最適化の最大化問題のための手法を，最短経路問題に適用するために，1) leximin についての最大化を leximax についての最小化に置き換える，2) 異なる長さのベクトルを比較する，の 2 点が必要である．

3.1 leximax についての最小化

leximin についての最大化は，最小化問題においては，大小関係が逆である leximax についての最小化に置き換えられる．leximax は leximin と同様に定義されるが，目的ベクトルの値が降順に整列されることが異なる．

定義 6 (降順に整列された目的ベクトル) 降順に整列された目的ベクトル $\mathbf{v}$ は値が降順に整列された目的ベクトルである．

定義 7 (Leximax) $\mathbf{v} = [v_1, \cdots, v_K]$ と $\mathbf{v}' = [v'_1, \cdots, v'_K]$ を長さ K の降順に整列された目的ベクトルとするとき，$\prec_{leximax}$ による順序は次のように定義される．$\exists t, \forall t' < t, v_{t'} = v'_{t'} \wedge v_t < v'_t$ であり，かつそのときのみ $\mathbf{v} \prec_{leximax} \mathbf{v}'$ である．

これにより，各目的の最悪の場合の値が最大コスト値となり，最適化問題の目的は利得の最大化からコストの最小化になる．降順に整列された目的ベクトルの加算は leximin と同様であるが，その値が降順に整列される．

3.2 異なる経路長の比較

経路最適化問題では，経路長が異なれば目的ベクトル長も異なる．そこで，leximax の定義を経路長の比較に拡張した可変ベクトル長の leximax (vleximax) を用いる．

定義 8 (vleximax) $\mathbf{v} = [v_1, \cdots, v_K]$ と $\mathbf{v}' = [v'_1, \cdots, v'_{K'}]$ を長さ K と K' の降順に整列された目的ベクトルとする．$K = K'$ の場合は，$\prec_{vleximax}$ は $\prec_{leximax}$ と同様である．その他の場合は，$K < K' \wedge [v_1, \cdots, v_K] = [v'_1, \cdots, v'_K]$ であり，かつそのときのみ $\mathbf{v} \prec_{vleximax} \mathbf{v}'$ である．

A*アルゴリズムにおいて vleximax により経路長を合理的に比較するためには，降順に整列された目的ベクトルが適切でなければならない．

3.3 発見的距離関数

A*アルゴリズムにおける発見的距離関数が返す経路コストの推定値 $h(v_i)$ は，真の経路コスト以下の下界値なければならない．グリッド状のグラフについて，辺のコストの下限値とマンハッタン距離に基づいて発見的距離関数を定義できる．合計コストを最小化する場合は，マンハッタン距離と辺のコストの下限値の積を経路コストの下界値とすることができる．

vleximax についての最小化の場合は，マンハッタン距離の個数分の，辺のコストの下限値からなるベクトルを，経路コストの下界値とすることができる

3.4 解法の正しさと複雑さ

A*アルゴリズムによる経路が最適であるためには，発見的距離関数が返す経路コストの推定値 $h(v_i)$ が真の経路コスト以下であれば良い．グリッド状のグラフにおける vleximax にもとづく経路の最小化では，ある頂点から終点までのマンハッタン距離は残りの可能な最小ベクトル長であるから，その数の目的の下限値からなるベクトルは，残りの経路についての下界値である．vleximin による異なる長さのベクトル比較は，真の経路コストのベクトルの比較において，同じ長さの部分までが同一の場合に，ベクトル長すなわち区間数が小さいものを選ぶタイブレークの効果をもたらす．経路コストの推定値は下界値であるから，ベクトルの長さが異なっても誤った経路は最適経路とはならない．したがって，解法は vleximax についての最適な経路を返す．また，合計コストの最小化の場合に推定値 $h(v_i)$ を 0 とすることができるように，vleximax についての最小化でも単に空のベクトルを $h(v_i)$ として用いても最適経路は得られる．

vleximax を用いる場合の計算コストのオーバヘッドは，単にスカラ値の合計と比較に基づく場合よりも顕著に大きいが，多項式的である．整列された目的ベクトルを連長圧縮表現する場合の各ベクトルの比較および格納のための時空間複雑度は，目的の離散値の種類が n 個であれば $O(n)$ である．ソートされた n 個の値の列への値の追加は，赤黒木のようなデータ構造を用いれば，$O(\log n)$ の時間複雑度で操作できる．その一方で，発見的距離関数の設計が難しい場合は，木探索における近傍の展開が，非効率的となる可能性がある．

4. 評価

提案手法を実験により評価した．例題は 10×10 頂点および 100×100 頂点のグリッド状のグラフとし，左上の頂点を始点 v_s，右下の頂点を終点 v_g とした．各辺のコスト値として，$[1, 2]$, $[1, 5]$ および $[1, 10]$ の整数値を，一様分布に基づきランダムに選んだ．それぞれのパラーメータについて 10 個の例題の結果を平均した．従来の合計に基づく解法 ‘sum.’ および leximax (vleximax) に基づく解法 ‘lxm.’ を比較した．実験には Core i7-3930K CPU (3.20GHz), 16GB memory, Linux 2.6.32, g++ (GCC) 4.4.7 の環境を用いた．

表 1 と 2 に解品質を示す．解品質はコストの合計，最小値，最大値，経路に含まれる辺の数 (len.)，タイル尺度 (theil) により評価する．タイル尺度は不公平性の尺度であり，小さい値のほうが不公平さが少ないことを表す．解法 ‘lxm.’ はコストの合計とのトレードオフにより，不公平さを削減し，解法 ‘sum.‘の場合よりも小さいタイル尺度となった．また，可能な場合に各区間のコストの最大値も抑制された．

表 3 に計算コストを示す．計算コストは各頂点の近傍を操作した回数 (iter.)，展開された頂点の数 (num. of opn. nodes), 実行時間により評価する．グリッド状のグラフとマンハッタン距離に基づく発見的距離関数はさほど効率的ではないため，大半の頂点が展開されたが，解法 ‘lxm.’ の場合もある程度は発見的距離関数と枝刈りの効果が見られた．解法 ‘lxm.’ の実行時間は整列されたベクトルと vleximax の計算コストの影響のため，解法 ‘sum.’ よりも顕著に大きい．その一方で，比較的小規模な問題であれば許容できる可能性があると考えられる．

表 1: 解品質 (10×10 頂点のグリッド)

cost	alg.	solution quality				
		sum.	min.	max.	len.	theil
$[1, 2]$	sum.	20.7	1	2	18	0.039
	lxm.	21.9	1	2	19.6	0.032
$[1, 5]$	sum.	34.3	1	4	18	0.135
	lxm.	41.3	1	3.4	22.4	0.095
$[1, 10]$	sum.	58.6	1	7.7	18.4	0.223
	lxm.	74.4	1	6.6	23.2	0.147

表 2: 解品質 (100×100 頂点のグリッド)

cost	alg.	solution quality				
		sum.	min.	max.	len.	theil
$[1, 2]$	sum.	213.2	1	2	198	0.025
	lxm.	286.9	1	2	282.8	0.006
$[1, 5]$	sum.	346.1	1	5	199.6	0.132
	lxm.	446.7	1	3.6	275.6	0.085
$[1, 10]$	sum.	580.8	1	9.5	202.8	0.215
	lxm.	960.3	1	6.8	350	0.128

表 3: 計算コスト (100×100 頂点のグリッド)

cost	alg.	iter.	num. of opn. nodes	exec. time [s]
$[1, 2]$	sum.	7217	7537	0.022
	lxm.	8359	8883	0.246
$[1, 5]$	sum.	9989	9996	0.021
	lxm.	9497	9923	0.223
$[1, 10]$	sum.	9996	9999	0.019
	lxm.	8468	9182	0.337

5. 議論

leximax についての最大化は，最悪の場合のコスト値の改善を優先するため，そのトレードオフによりコスト値の合計は増加する．コスト値の範囲が比較的広い場合は，そのトレードオフは拡大し，合計の増加の程度も大きくなる．また，ベクトル長すなわち，経路に含まれる辺の数も増加する．このようなコストの増加が極端な場合には，コスト値の範囲の調整や，経路長とのトレードオフを制御する手法などの検討が必要と考えられる．その一方で，提案手法により得られる経路はある種の公平性にもとづく解を分析するものと考えられる．

本研究では，グリッド状のグラフを前提とし，マンハッタン距離に基づく発見的距離関数を用いた．これは，ある頂点から終点までのマンハッタン距離が残りの可能な最小ベクトル長であることを前提としている．他のトポロジやより効率的な推定コストのためには異なる考察が必要であると考えられる．

6. おわりに

本研究では，leximax の最大化にもとづき，各区間の公平性を考慮しつつコストを最適化する経路探索手法について検討した．提案手法により，最大コストとなる区間を抑制しつつ公平性を改善する経路が得られた．解法の効率化，合計コストや経路長との実際的なトレードオフなどが今後の課題として挙げられる．

参考文献

[Bouveret 09] Bouveret, S. and Lemaître, M.: Computing Leximin-optimal Solutions in Constraint Networks, *Artificial Intelligence*, Vol. 173, No. 2, pp. 343–364 (2009)

[Marler 04] Marler, R. T. and Arora, J. S.: Survey of multi-objective optimization methods for engineering, *Structural and Multidisciplinary Optimization*, Vol. 26, pp. 369–395 (2004)

[Matsui 14] Matsui, T., Silaghi, M., Hirayama, K., Yokoo, M., and Matsuo, H.: Leximin Multiple Objective Optimization for Preferences of Agents, in *17th International Conference on Principles and Practice of Multi-Agent Systems*, pp. 423–438 (2014)

[Matsui 15] Matsui, T., Silaghi, M., Okimoto, T., Hirayama, K., Yokoo, M., and Matsuo, H.: Leximin Asymmetric Multiple Objective DCOP on Factor Graph, in *18th International Conference on Principles and Practice of Multi-Agent Systems*, pp. 134–151 (2015)

[Russell 03] Russell, S. and Norvig, P.: *Artificial Intelligence: A Modern Approach (2nd Edition)*, Prentice Hall (2003)

[Sen 97] Sen, A. K.: *Choice, Welfare and Measurement*, Harvard University Press (1997)