[計測自動制御学会論文集]
Vol.52, No.10, 566/572（2016）

強化学習における線形計画法を用いた効率的解法

泉田　啓*・天野恒佑*

Efficient Algorithms for Reinforcement Learning by Linear Programming

Kei Senda* and Koyu Amano*

Model-based reinforcement learning includes two steps, estimation of a plant and planning. Planning is formulated as dynamic programming (DP) problem, which is solved by a DP method. This DP problem has an equivalent linear programming (LP) problem that can be solved by LP method, but it is generally less efficient than typical DP method. However, numerical examples show linear programming is more efficient than the typical DP method in problems whose self-transition probabilities are large. The reason is clarified by geometrical discussion of each solution of method approaches to optimal solution.

Key Words: reinforcement learning, planning, linear programming, self-transition probability

1. はじめに

強化学習では，状態に J 値と呼ばれる評価値を付け，この値に基づいて行動の選択を行う場合が多い．そこでは，最適方策を与える J 値を獲得することが学習の目的となる[1) 2)]．モデル・ベース強化学習は「プラント推定」と「推定プラントの最適方策の獲得」から成り，後者はプランニングと呼ばれる．プランニングにおいて，価値反復や方策反復などの動的計画 (DP) 法により，効率的に最適方策を得る方法が研究されている[2)]が，それでも膨大な計算が必要となる．そこで本研究では，プランニングにおける計算の効率化を目的とする．

マルコフ決定過程 (MDP) を扱うプランニングの DP 問題は，等価な線形計画 (LP) 問題に変換することができ，LP 法によって解くことができる．しかし，LP 法は DP 法に比べ計算効率が低いと考えられてきた[3)~5)]．文献[3)~8)]等に基づいて計算量を評価すると，次のように要約できる．

離散 MDP を扱うプランニング問題を考え，状態数を N，各状態の行動数を K，LP 形式での問題の入力に必要な総ビット数を L とする．まず，割引問題に LP 法の一解法の単体（シンプレックス）法[9)]を適用すると，DP 法の単純方策反復になる．ただし，単純方策反復では，1 ステップで 1 状態までの方策しか更新しない．一般的な方策反復では 1 ステップで全状態の方策を一度に更新するが，実用的な観点では方策反復は単体法よりも計算効率が良いとされる[3)]．次に，割引率 α が一定 $(0 < \alpha < 1)$ の割引問題において，最悪ケースの計算量で評価する．もし，$L\log(1/(1-\alpha))/(1-\alpha)$ が K^N/N より小さければ価値反復に対し方策反復は N 倍である．また，価値反復に対し標準的な内点法は $NK(1-\alpha)/\log(1/(1-\alpha))$ 倍である．さらに，Ye[5), 8)]により，最負値削減コスト・ピボット則に従う単体法（単純方策反復）は $O(N^4K^2\log N)$，標準的な内点法は $O(N^3K^2L)$，組合せ内点法は $O(N^4K^4\log N)$ であると評価された．なお，L は無限大にまでなり得るので，第 1 と第 3 の解法が強多項式時間アルゴリズムである（最悪ケースの計算量評価に N と K を含むが L を含まない）ことが証明されたことは注目に値する．

このように，計算量の少ない解法同士で比べると，LP 法は DP 法に比べ計算効率が低いと考えられ，LP 法の適用について十分な検討はされてこなかった．しかし，本研究では元の DP 問題と等価な LP 問題を LP 法で解くことにより，効率的に解を得られないか検討する．その結果，自己遷移確率が増大すると，LP 法が DP 法の価値反復と比較して効率的になることが明らかになる．この事実は，後の数値シミュレーションによって確認され，その理由が幾何学的に説明される．

近年，Approximate Linear Programming (ALP) について，多くの研究がなされている[10)~12)]．これらでは，制約条件を緩和したり，変数に重みをつけたりして，元の DP 問題を解き易い LP 問題に近似して LP 法で解いている．他方，本研究は元の DP 問題の解を LP 法で求めるものである．なお，DP 法による効率的な解法も検討されている[13)]．

本論文の構成として，2. で強化学習で扱うプランニング問題を定式化し，3. でその標準的解法である DP 法について説明する．4. でプランニング問題が LP 問題に変換できることを示し，LP 問題の性質についても説明する．5. では数値シミュレーションを行い，提案手法の有効性を示す．6. で数値

* 京都大学大学院工学研究科　京都市西京区京都大学桂
* Graduate School of Engineering, Kyoto University, Nishikyo-ku, Kyoto-Daigaku Katsura
（Received March 4, 2016）
（Revised June 22, 2016）

シミュレーションの結果について考察する．最後に，7. を本研究の結論とする．

2. 強化学習問題

一般的な DP (Dynamic Programming) の定式に従い，離散時間の動的システムを考える [2]．状態 s_i と行動 u_k は，有限集合 $\mathcal{S}$ と $\mathcal{U}$ の元で，離散的な変数とする．状態 $\mathcal{S}$ の元として $s_1, s_2, \ldots, s_N$ の N 個を考え，問題に応じて終端の状態 s_0 を付け加える．行動 $\mathcal{U}$ の元として $u_1, u_2, \ldots, u_K$ の K 個を考える．状態 s_i で行動 u_k を選ぶと状態 s_j に移る場合を遷移 (s_i, u_k, s_j) と書く．このときの状態遷移確率は $p_{ij}(u_k)$ で，コスト $g(s_i, u_k, s_j)$ を生じる．本研究では離散時間の有限 MDP を扱う．すなわち $p_{ij}(u_k)$ は現在の状態と行動のみによって決定される．また，系は時刻に陽には依存しない．状態を行動に写像する方策として，時刻に陽に依存しない定常方策 μ を考え，行動を $u_k = \mu(s_i) \in \mathcal{U}$ のように選択し，行動選択確率 $\pi(s_i,\ u_k)$ は時刻によらない．なお，本論文を通じて使う記号等は付録に示し，定式等の詳細は文献 [14] に従う．

DP 問題は，有限のステージ数にわたってコストが蓄積する finite hoziron 問題と，期間不定でコストが蓄積する infinite horizon 問題に区別できる．本研究では後者を扱うが，時刻を特別な状態量とみなすことにより，前者は後者に変換できるので，一般性は失われない．また，infinite horizon 問題で，求める方策は一般的に定常方策になる [2]．

初期状態 $s^{(0)} = s_i$ から開始し，第 t 回目の遷移で状態 $s^{(t)}$ を訪れ，加法的に蓄積するコストを生じるとき，方策 μ の全コストの期待値 (J 値) は

$$J^{\mu}(s_i) = E\left[\sum_{t=0}^{\infty} \alpha^t g\left(s^{(t)}, \mu(s^{(t)}), s^{(t+1)}\right) \;\middle|\; s^{(0)} = s_i\right]$$

ここで，α は $0 < \alpha \leq 1$ で割引率と呼ぶ．最適 J 値は $J^*(s_i) = \min_{\mu} J^{\mu}(s_i)$ と表され，すべての状態に対し $J^{\mu}(s_i) = J^*(s_i)$ なら方策 μ は最適である．最適 J 値とそれによって導かれる最適方策 μ^* は次式を満たす．

$$J^*(s_i) = \min_{u_k} \sum_{j=0}^{N} p_{ij}(u_k)\{g(s_i, u_k, s_j) + J^*(s_j)\} \quad (1)$$

$$\mu^*(s_i) = \arg\min_{u_k} \sum_{j=0}^{N} p_{ij}(u_k)\{g(s_i, u_k, s_j) + J^*(s_j)\} \quad (2)$$

(1) 式は Bellman 方程式と呼ばれる．本研究では，infinite horizon 問題の一種である，以下の確率的最短経路問題を考える．割引なし ($\alpha = 1$) とし，コストを生じない終端 (ゴール) 状態 s_0 が存在し，すべての行動 u_k に対して $p_{00}(u_k) = 1,\ g(s_0, u_k, s_0) = 0,\ J(s_0) = 0$ とする．これらの条件下で，コストの総和の期待値が最小で終端状態に達すること，すなわち，$J^{\mu}(s_i)$ を最小化する最適方策を見つけることが，この問題の目的である．この問題の解法は，非常に多くの強化学習問題に適用できる一般的解法であることが知られている [2]．

3. DP 問題の標準的解法

正確にプラントが推定され，状態遷移確率 $p_{ij}(u_k)$ とコスト $g(s_i, u_k, s_j)$ が既知とする．このとき，(1) 式の解 $J^*(s_i)$ を求める問題を DP 問題，その解法を DP 法と呼ぶ．DP 法は Bellman の最適性の原理 [15] に基づき，効率的に最適解を求めるものである．N 次元 J 値ベクトル $\boldsymbol{J}$ から新たな J 値ベクトルを求める写像 T を次のように定義する．

$$(TJ)(s_i) = \min_{u_k} \sum_{j=0}^{N} p_{ij}(u_k)\{g(s_i, u_k, s_j) + J(s_j)\} \quad (3)$$

任意の初期ベクトル $\boldsymbol{J}^{(0)}$ からはじめ，順次 $\boldsymbol{J}^{(m)} = T^m \boldsymbol{J}_0$ $(m = 0, 1, \ldots)$ を求め，(1) 式を満たす最適 J 値ベクトル $\boldsymbol{J}^*$ を次式で求めることができる．

$$\boldsymbol{J}^* = \lim_{m \to \infty} T^m \boldsymbol{J}^{(0)}$$

このようにして，ある J 値ベクトルから開始して $T^m \boldsymbol{J}$ を計算する DP 反復法のことを価値反復法と呼ぶ．

4. DP 問題と等価な LP 問題

4.1 DP 問題から LP 問題への変換

以下のように，DP 問題は等価な LP 問題に変換できる [2]．

価値反復法の写像 T は単調性を有し，$\boldsymbol{J} \leq \bar{\boldsymbol{J}}$ を満たすベクトル $\boldsymbol{J}$，$\bar{\boldsymbol{J}}$ に関して，$T\boldsymbol{J} \leq T\bar{\boldsymbol{J}}$ が成り立つ [2]．ある N 次元ベクトル $\boldsymbol{J}$ が $\boldsymbol{J} \leq T\boldsymbol{J}$ を満たせば T の単調性より $\boldsymbol{J} \leq \boldsymbol{J}^* = T\boldsymbol{J}^*$ が成り立つ．すなわち，最適解 $\boldsymbol{J}^*$ は，拘束 $\boldsymbol{J} \leq T\boldsymbol{J}$ を満たす最大の $\boldsymbol{J}$ である．この拘束は

$$J(s_i) \leq \sum_{j=0}^{N} p_{ij}(u_k)\{g(s_i, u_k, s_j) + J(s_j)\},\ \forall (s_i, u_k) \quad (4)$$

となる．よって，DP 問題と等価な LP 問題を

$$\max \quad \sum_{i=1}^{N} J(s_i)$$

$$\text{s.t.} \quad J(s_i) \leq \sum_{j=0}^{N} p_{ij}(u_k)\{g(s_i, u_k, s_j) + J(s_j)\},\ \forall (s_i, u_k)$$

と定義できる．また，1 ステップコストは $g(s_i, u_k, s_j) \geq 0$ なので $J(s_i) \geq 0$, $\forall s_i$ となる．付録のように，状態遷移確率 $\boldsymbol{p}$ に基づき $\boldsymbol{P}$，$\boldsymbol{p}$ とコスト $\boldsymbol{g}$ から $\bar{\boldsymbol{g}}$ を与えれば，この問題は行列表記で次のように定義できる．

$$\max \qquad \boldsymbol{e}^T \boldsymbol{J} \quad (5)$$

$$\text{s.t.} \qquad (\boldsymbol{E} - \boldsymbol{P})\boldsymbol{J} \leq \bar{\boldsymbol{g}} \quad (6)$$

$$\boldsymbol{J} \geq \boldsymbol{0} \quad (7)$$

4.2 双対原理

LP 問題には双対問題と呼ばれる等価な LP 問題が存在する．これらの 2 つの問題には密接な関係があり，元の問題の

Table 1 Relation between primal problem and dual problem of DP problem

	primal problem	dual problem
type of problem	maximize	minimize
type of variable	$\boldsymbol{J}$	$\boldsymbol{L}$（$\Longleftrightarrow$ policy）
number of variable	N	NK
number of constraint	NK	N
type of constraint	inequality	equality

代わりに双対問題を解くことや，両方の問題をまとめて考察することが有効な場合がある．本節では，双対問題の定式を示し，双対理論の基盤となる双対定理について説明する．

4.2.1 双対問題

双対問題が定められるとき，元の問題のことを主問題と呼ぶ．主問題 (5)–(7) の双対問題は以下で定義される．

$$\min \qquad \bar{\boldsymbol{g}}^T\boldsymbol{L} \tag{8}$$

$$\text{s.t.} \qquad (\boldsymbol{E}-\boldsymbol{P})^T\boldsymbol{L}=\boldsymbol{e} \tag{9}$$

$$\boldsymbol{L}\geq\boldsymbol{0} \tag{10}$$

主問題 (5)–(7) と双対問題 (8)–(10) からわかるように，主問題が不等式制約の最大化問題であれば，双対問題は等式制約の最小化問題になる．また，主問題で変数の数が N 個，制約式の数が NK 個であれば，双対問題では逆転に変数が NK 個，制約式が N 個となる．なお，双対問題の双対問題はもとの主問題となることが知られている．DP 問題における主問題と双対問題の関係を Table 1 にまとめる．

4.2.2 双対定理

主問題と双対問題には以下の双対定理が成り立つ．

《**定理 1**》 主問題あるいは双対問題の一方が最適解を持てば他方も最適解を持ち，それぞれの最適解に対する目的関数の値は等しくなる (双対定理)．

この定理により次式が成り立つ．

$$\boldsymbol{e}^T\boldsymbol{J}\leq\boldsymbol{e}^T\boldsymbol{J}^*=\bar{\boldsymbol{g}}^T\boldsymbol{L}^*\leq\bar{\boldsymbol{g}}^T\boldsymbol{L}$$

4.3 双対問題の考察

付録に従って，最適方策 μ^* により $\boldsymbol{\Pi}^*$ を与えるとき，Bellman 方程式は行列表記を用いて

$$\boldsymbol{J}^*=(\boldsymbol{I}-\boldsymbol{\Pi}^*\boldsymbol{P})^{-1}\boldsymbol{\Pi}^*\bar{\boldsymbol{g}}$$

となる．ここで，$a_{ij}^{\mu^*}\triangleq\left[(\boldsymbol{I}-\boldsymbol{\Pi}^*\boldsymbol{P})^{-1}\right]_{ij}$ とすると

$$J^*(s_i)=\sum_{j=1}^{N}a_{ij}^{\mu^*}\sum_{\ell=1}^{K}\pi^*(s_j,u_\ell)\bar{g}(s_j,u_\ell)$$

が成立する．これと双対定理により，$\sum_{i=1}^{N}J^*(s_i)$ は

$$\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{N}a_{ji}^{\mu^*}\pi^*(s_i,u_k)\bar{g}(s_i,u_k)$$

$$=\sum_{i=1}^{N}\sum_{k=1}^{K}\bar{g}(s_i,u_k)L^*(s_i,u_k)$$

となり，かつ主問題と双対問題の最適解がそれぞれ一意であることから，両辺の比較により

$$L^*(s_i,u_k)=\pi^*(s_i,u_k)\sum_{j=1}^{N}a_{ji}^{\mu^*}$$

が成立する．さらに，$\pi^*(s_i,u_k)$ は行動選択確率なので

$$\pi^*(s_i,u_k)=\frac{L^*(s_i,u_k)}{\sum_{\ell=1}^{K}L^*(s_i,u_\ell)}$$

となる．すなわち，双対問題の最適解 $\boldsymbol{L}^*$ から最適方策 μ^* を導くことができることがわかる．

4.4 LP 法

本節では，LP 法の中でも効率的な解法として知られる単体（シンプレックス）法と内点法を説明する [16),17)]．

4.4.1 単体法

単体法では一般に標準形の問題を扱う．ここで標準形とは，(a) 最小化問題（目的関数を最小化する），(b) 制約条件がすべて等式，(c) 全ての変数に非負条件が課せられる，の 3 つの条件を満たすものを指す．双対問題 (8)–(10) は標準形である．

双対問題 (8)–(10) において，変数の数は NK，制約条件の数は N である．そのため，変数の定め方には $N(K-1)$ 個の自由度がある．任意の $N(K-1)$ 個の変数を 0 とすると，残りの N 個の変数を定めることができる．このとき，定められる N 個の変数を基底変数，0 とする $N(K-1)$ 個の変数を非基底変数と呼ぶ．ここで，N 個の基底変数からなる N 次元ベクトルを $\boldsymbol{L}_B$，$N(K-1)$ 個の非基底変数からなる $N(K-1)$ 次元ベクトルを $\boldsymbol{L}_N$ で表す．また，$N\times NK$ 行列である $(\boldsymbol{E}-\boldsymbol{P})^T$ を $N\times N$ の基底行列 $\boldsymbol{B}$ および $N\times N(K-1)$ の非基底行列 $\boldsymbol{N}$ に分割すると (9) 式は

$$\boldsymbol{B}\boldsymbol{L}_B+\boldsymbol{N}\boldsymbol{L}_N=\boldsymbol{e}\quad\Leftrightarrow\quad\boldsymbol{L}_B=\boldsymbol{B}^{-1}\boldsymbol{e}-\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{L}_N$$

よって目的関数 (8) は

$$\bar{\boldsymbol{g}}^T\boldsymbol{L}=\bar{\boldsymbol{g}}_B^T\boldsymbol{L}_B+\bar{\boldsymbol{g}}_N^T\boldsymbol{L}_N=\bar{\boldsymbol{g}}_B^T\boldsymbol{B}^{-1}\boldsymbol{e}+\tilde{\boldsymbol{g}}_N^T\boldsymbol{L}_N$$

ただし，$\tilde{\boldsymbol{g}}_N^T=\bar{\boldsymbol{g}}_N^T-\bar{\boldsymbol{g}}_B^T\boldsymbol{B}^{-1}\boldsymbol{N}$ である．

単体法の手順を示す．

ステップ 1 初期実行可能基底解 $(\boldsymbol{L}_B,\boldsymbol{L}_N)=(\boldsymbol{B}^{-1}\boldsymbol{e},\boldsymbol{0})$ を選ぶ．

ステップ 2 $\tilde{\boldsymbol{g}}_N\geq\boldsymbol{0}$ であれば，すでに最適解が得られているので計算を終了する．そうでなければ負の要素 $\tilde{g}_j$ に対応する非基底変数 L_j を 1 つ選び，基底変数の非負制約を破らない範囲 (ある基底変数が 0 になるまで) で増加させる．(基底変数と非基底変数の入れ替え)

ステップ 3 目的関数を調べ，$\tilde{\boldsymbol{g}}_N\geq\boldsymbol{0}$ を満たせば計算を終了する．そうでなければステップ 2 に戻る．

ステップ 2 では，基底変数と非基底変数が入れ替わるため，制約条件の境界に沿って隣接した頂点に移動することとなる．つまり，単体法では実行可能領域の頂点を移動することで最適解に到達する解法と解釈できる．

主問題 (5)–(7) は，以下の標準形の問題に変換できる．

$$\begin{aligned} \min \quad & -\boldsymbol{e}^T\boldsymbol{J} \\ \text{s.t.} \quad & (\boldsymbol{E}-\boldsymbol{P})\boldsymbol{J}+\boldsymbol{z}=\bar{\boldsymbol{g}} \\ & \boldsymbol{J}\geq\boldsymbol{0},\ \boldsymbol{z}\geq\boldsymbol{0} \end{aligned}$$

ここで，$\boldsymbol{z}$ は不等式を等式にするためのスラック変数である．このような標準形は，単体法によって解くことができる．一般に，主問題を解くものを主単体法，双対問題を解くものを双対単体法と呼ぶ．

4.4.2　内点法

内点法は実行可能領域の内部を移動して最適解に近づいていく反復解法である．内点法にはいくつか種類があり，その中で，効率的とされる主双対内点法について説明する．

主双対内点法は主双対問題という主問題と双対問題の両方を考慮した問題を解くものである．ここでは，主問題 (5)–(7) と双対問題 (8)–(10) に対して主双対問題を導く．まず，主問題と双対問題の目的関数の差は以下のようになる．

$$\begin{aligned} \bar{\boldsymbol{g}}^T\boldsymbol{L}-\boldsymbol{e}^T\boldsymbol{J} =& \{(\boldsymbol{E}-\boldsymbol{P})\boldsymbol{J}+\boldsymbol{z}\}^T\boldsymbol{L}-\boldsymbol{e}^T\boldsymbol{J} \\ =& \boldsymbol{J}^T\boldsymbol{e}+\boldsymbol{z}^T\boldsymbol{L}-\boldsymbol{e}^T\boldsymbol{J}=\boldsymbol{z}^T\boldsymbol{L} \end{aligned}$$

ここで，$\boldsymbol{z}^T\boldsymbol{L}$ は双対ギャップと呼ばれる．双対定理より，$\boldsymbol{e}^T\boldsymbol{J}^*=\bar{\boldsymbol{g}}^T\boldsymbol{L}^*$ なので，最適解において $\boldsymbol{z}^{*T}\boldsymbol{L}^*=0$ が成り立つ．これより，主問題の (5)–(7) と双対問題 (8)–(10) の制約条件を満たした上で，$\boldsymbol{z}^T\boldsymbol{L}=0$ となる解が最適解となる．すなわち，以下を満たす解 $(\boldsymbol{J},\boldsymbol{z},\boldsymbol{L})$ は最適解となる．

$$\begin{aligned} (\boldsymbol{E}-\boldsymbol{P})\boldsymbol{J}+\boldsymbol{z}&=\bar{\boldsymbol{g}} \\ (\boldsymbol{E}-\boldsymbol{P})^T\boldsymbol{L}&=\boldsymbol{e} \\ \boldsymbol{z}^T\boldsymbol{L}&=0 \\ \boldsymbol{J}\geq\boldsymbol{0},\ \boldsymbol{z}\geq&\boldsymbol{0},\ \boldsymbol{L}\geq\boldsymbol{0} \end{aligned}$$

このような解を求める問題を主双対問題という．主双対内点法は，主双対問題をアフィンスケーリング法やパス追跡法によって解くものである．本研究では，現在主流となっている Mehrotra の予測子・修正子法[17]を扱う．

4.5　計算効率の評価方法

本研究では，最適解を得るまでに必要となる計算回数で計算効率を評価する．まず，変数 $\boldsymbol{J}$ あるいは $\boldsymbol{L}$ の更新を 1 回の反復とみなし，1 反復に必要な計算量を見積もる．その後，最適解を得るまでに必要となる反復回数を求め，それらの積によって総計算量が求められる．なお，本研究のいずれの問題においても価値反復の方が方策反復より効率的なため，DP 法の計算コストは価値反復法で評価する．価値反復法，主単体法，双対単体法，主双対内点法の 4 つの解法について 1 反復あたりの計算量を求め，Table 2 に示す．なお，最適解を得るまでに必要となる反復回数は，数値シミュレーションで実際に問題を解いて求める．

5.　数　値　例

迷路問題と質点制御問題の数値例で評価する．

Table 2　Calculation cost for updating vector $\boldsymbol{J}$

value iteration	N^2K
primal simplex	$N^2K^2+N^2K+2NK$
dual simplex	N^2K+2N
primal-dual interior point	$N^3K+N^3/3+10N^2K$ $+2N^2+15NK+N$

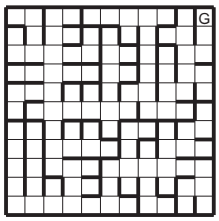

Fig. 1　11×11 maze

5.1　迷路問題

Fig. 1 に示す迷路問題について考える．エージェントは状態として現在の位置を観測し，行動として上下左右への移動をとる．壁のない方向への行動を選択した際は，確率 $1-p_{ii}$ で選んだ行動の方向へ遷移し，確率 p_{ii} で現在の状態に留まる．壁のある方向への行動を選択した場合，確率 1 で現在の状態に留まるものとする．また，いずれの遷移も 1 ステップコストとして 1 を生じる．また，迷路の右上の角のマスをゴールとして設定し，これは終端状態に対応している．以上により，状態数 $N=120$，行動数 $K=4$ の問題となる．

この問題を価値反復法，主単体法，双対単体法，主双対内点法の 4 つの解法を用いて解き，最適解を得るまでに必要とした反復回数を求める．その際，初期状態として，価値反復法と単体法では全ての未知変数を 0 とし，主双対内点法ではランダム方策に対応した初期点を選択する．また，以下のように解の収束を判定する．単体法は有限回の反復で収束する．価値反復と主双対内点法は有限回の反復で収束しないため，予め求めた J^* に対して $||J-J^*||^2/||J^*||^2<10^{-6}$ を収束判定条件とする．このことから，得られる解の質は，価値反復と主双対内点法は同程度で，単体法が最も優れている．

シミュレーションは $p_{ii}=0.0,\ 0.5,\ 0.9$ の 3 通りで行った．その結果として，各解法で最適解を求めるまでに必要とした反復回数を Table 3 に，1 反復あたりの計算回数を Table 2 として総計算回数を求めたものを Table 4 に示す．

Table 4 より，価値反復法では，自己遷移確率 p_{ii} を大きくすると計算回数が増加している．一方，LP 法である他の 3 つの解法では，自己遷移確率 p_{ii} が大きくなっても計算回数はほとんど変わらない．その結果，$p_{ii}=0.0$ のときは，価値反復法の計算回数が最も少ないが，$p_{ii}=0.9$ のときは，双対単体法の計算回数が最も少なくなる．

5.2　質点制御問題

質点制御問題は，1 次元空間上に存在する質点を終端状態に制御する問題である．このとき，終端状態に到るコストが最小

Table 3 Iteration number to obtain the optimal vector $\boldsymbol{J}^*$ (maze)

	11×11 maze		
	$p_{ii}=0.0$	$p_{ii}=0.5$	$p_{ii}=0.9$
value iteration	32	102	581
primal simplex	216	234	264
dual simplex	121	121	121
primal-dual interior point	7	7	7

Table 4 Calculation cost to obtain the optimal vector $\boldsymbol{J}^*$ for maze problem ($\times 10^6$)

	11×11 maze		
	$p_{ii}=0.0$	$p_{ii}=0.5$	$p_{ii}=0.9$
value iteration	1.84	5.88	33.5
primal simplex	62.4	67.6	76.3
dual simplex	7.0	7.0	7.0
primal-dual interior point	56.7	56.7	56.7

になるように制御入力 u を学習する．質点の位置を x [m]，速度を $\dot{x}=v$ [m/s]，制御入力を u [N]，質点の質量を m [kg]，粘性係数を c [Ns/m] とすると運動方程式は，$\ddot{x}+\frac{c}{m}\dot{x}=\frac{1}{m}u$ となる．状態を $\boldsymbol{x}=[x\ \dot{x}]^T$ とし，サンプリングタイムを T [s]，制御入力をゼロ次ホールドとすると以下の離散時間の状態方程式で表すことができる．

$$\boldsymbol{x}^{(t+1)}=\boldsymbol{A}_d\boldsymbol{x}^{(t)}+\boldsymbol{B}_d u^{(t)}$$

$$\boldsymbol{A}_d=\begin{bmatrix}1 & \frac{m}{c}\left\{1-\exp\left(-\frac{m}{c}T\right)\right\}\\ 0 & \exp\left(-\frac{c}{m}T\right)\end{bmatrix}$$

$$\boldsymbol{B}_d=\begin{bmatrix}\frac{m}{c^2}\left\{\exp\left(-\frac{c}{m}T\right)-1\right\}+\frac{T}{c}\\ \frac{1}{c}\left\{1-\exp\left(-\frac{c}{m}T\right)\right\}\end{bmatrix}$$

ただし，状態遷移により x あるいは $\dot{x}$ が状態空間の境界の外側へ遷移する場合は，確率 1 で，その境界に遷移するものとする．これにより，状態遷移確率 $\boldsymbol{P}$ が規定される．1 ステップコストを，現在の状態 $[x_t, v_t]$，行動 u_t に対して

$$g=\frac{1}{2}\left\{(x_t)^2+(v_t)^2+(u_t)^2\right\} \qquad (11)$$

とし，位置 $x=0.0$，速度 $\dot{x}=0.0$ となる状態を終端状態とし，このときの最適方策を求める．

状態空間は，位置 x に関して区間 $[-0.25, 1.25]$ を 15 に，速度 $\dot{x}$ に関しては区間 $[-0.525, 0.525]$ を 21 に，行動空間は，u に関して区間 $[-1.25, 1.25]$ を 10 に離散化する．よって，扱う問題は状態数 $N=315$，行動数 $K=10$ になる．解法としては価値反復法，主単体法，双対単体法，主双対内点法の 4 種類を用いる．各解法の初期状態と収束判定条件は，前節と同じとする．シミュレーションの結果として，各解法で最適解を得るまでに必要とした反復回数を Table 5 に，1 反復あたりの計算回数として Table 2 の値を用いて，総計算回数を求めたものを Table 6 に示す．

Table 6 より，価値反復法では，質量 m を大きくすると計算回数が増加している．一方，LP 法である他の 3 つの解法では，m を変化させても，価値反復法ほど計算回数に変化は見られない．結果として，$m=0.50$ のとき最も計算回数の

Table 5 Iteration number to obtain the optimal vector $\boldsymbol{J}^*$ for mass control problem

	$N=315,\ K=10$		
	$m=0.50$	$m=5.0$	$m=50$
value iteration	537	1462	14832
primal simplex	2691	1365	674
dual simplex	756	726	445
primal-dual interior point	20	18	23

Table 6 Calculation cost to obtain the optimal vector $\boldsymbol{J}^*$ for mass control problrem ($\times 10^8$)

	$N=315,\ K=10$		
	$m=0.50$	$m=5.0$	$m=50$
value iteration	5.33	14.5	147
primal simplex	294	149	73.6
dual simplex	7.51	7.21	4.42
primal-dual interior point	66.6	60.0	76.6

少なかった価値反復法は，$m=50$ のとき最も計算回数が多くなる．このとき，価値反復法の計算量は，双対単体法の 30 倍以上である．

6. シミュレーションの考察

6.1 自己遷移確率の変化が各解法に与える影響

自己遷移確率 p_{ii} を変化させた際に，価値反復法と LP 法で異なる影響が見られたことを説明するために，2 状態 2 行動の単純な最短経路問題を扱う．この問題において主問題 (5)–(7) を定義すると，実行可能領域は Fig. 2 の左図のように示すことができる．ここで，図内の $l(s_i, u_k)$ は状態 s_i，行動 u_k に対しての不等式制約の境界線を表しており，左辺が $J(s_2)$ となるように式変形すると，それぞれ

$$l(s_1,u_1): J(s_2)\geq\frac{1}{p_{12}(u_1)}\{(1-p_{11}(u_1))J(s_1)-\bar{g}(s_1,u_1)\}$$

$$l(s_1,u_2): J(s_2)\geq\frac{1}{p_{12}(u_2)}\{(1-p_{11}(u_2))J(s_1)-\bar{g}(s_1,u_2)\}$$

$$l(s_2,u_1): J(s_2)\leq\frac{1}{1-p_{22}(u_1)}\{p_{21}(u_1))J(s_1)+\bar{g}(s_2,u_1)\}$$

$$l(s_2,u_2): J(s_2)\leq\frac{1}{1-p_{22}(u_2)}\{p_{21}(u_2))J(s_1)+\bar{g}(s_2,u_2)\}$$

と表すことができる．ここで，自己遷移確率，つまり $p_{11}(u_1)$, $p_{11}(u_2)$, $p_{22}(u_1)$, $p_{22}(u_2)$ の値が大きくなると，$l(s_1,u_1)$, $l(s_1,u_2)$ は傾きが小さくなり $l(s_2,u_1)$, $l(s_2,u_2)$ は傾きが大きくなる．このとき，Fig. 2 の左図で表されていた実行可能領域は，右図に示されるような細長い形に変化する．

ここで，各解法の解の更新の様子を幾何学的に考える．

価値反復法は，Fig. 3 のように，実行可能領域内に超直方体 (2 次元の場合は長方形) を生成し，超直方体の頂点から頂点へ移動することで解を更新していく．これについては，次節で詳しく説明する．そのため，自己遷移確率が大きくなり実行可能領域が細長くなると，最適解に対して 1 回の反復で進む距離が小さくなり，その結果，反復回数が増大する．

一方，Fig. 4 のように，単体法は実行可能領域の端点を移動することで，内点法は実行可能領域内部を移動することで解を更新する [17]．そのため，価値反復のように実行可能領域

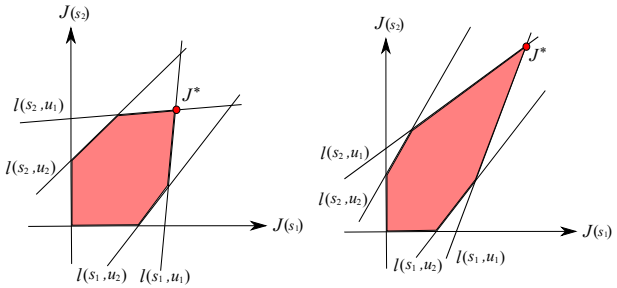

Fig. 2　Feasible area of the problem including two state and two action (the self-transition probabilities of right figure are larger than left one)

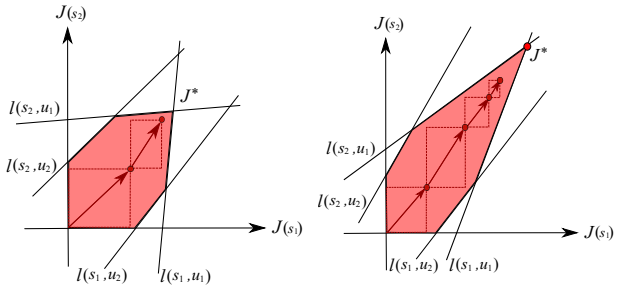

Fig. 3　How to approach the optimal solution by value iteration (the self-transition probabilities of right figure are larger than left one)

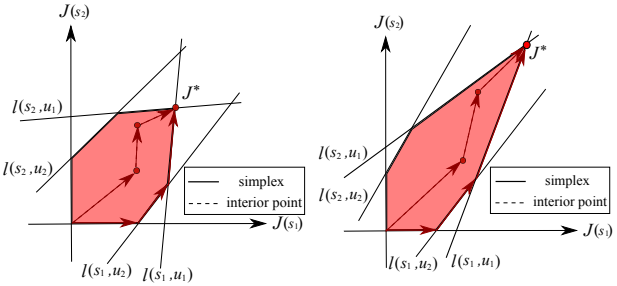

Fig. 4　How to approach the optimal solution by simplex and interior point (the self-transition probabilities of right figure are larger than left one)

の形状変化や大きさの影響を直接的に受けて反復回数が増加する働きは無く，反復回数は殆ど増加しない．

以上の理由により，自己遷移確率が大きくなると，価値反復法では反復回数が増大し，単体法や内点法では反復回数に大きな変化は現れない．このことは，数値計算でも確かめられている．

また，質点制御問題において質量 m を変化させた際にも，同様の様子が見られた．これは m が大きくなると，同じ制御に対して質点の移動量が減少し，自己遷移確率が増えるためである．よって，自己遷移確率が大きくなると，標準的な DP 法の価値反復法では反復回数が増大するため，双対単体法などの LP 法の方が効率的になる．このことは，これまで知られていなかった．

6.2　価値反復法の解更新の様子

前節の最短経路問題における価値反復の解更新を幾何学的に解釈する（Fig. 5 参照）．(3) 式の価値反復は

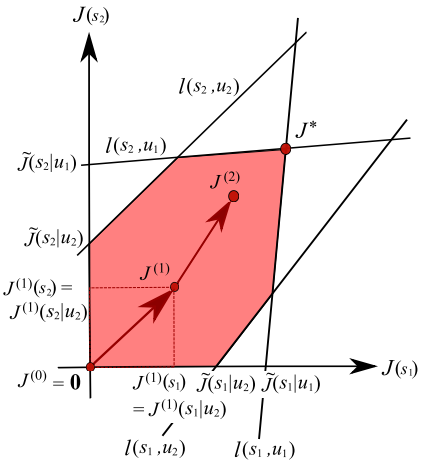

Fig. 5　Geometric understanding of value iteration how to approach to the optimal solution

$$J^{(m+1)}(s_i)=\min_{u_k}\left[\bar{g}(s_i,u_k)+\sum_{j=1}^{N}p_{ij}(u_k)J^{(m)}(s_j)\right],\ \forall s_i$$

ここで，$\bar{g}(s_i,u_k)=\sum_{j=1}^{N}p_{ij}(u_k)g(s_i,u_k,s_j)$, $m=0,1,\ldots$ は第 m 反復を表す．また，実行可能領域を示す (4) 式は

$$J(s_i)\le\bar{g}(s_i,u_k)+\sum_{j=1}^{N}p_{ij}(u_k)J(s_j)\ ,\ \forall(s_i,u_k)$$

になり，$l(s_1,u_1)$〜$l(s_2,u_2)$ を表す．任意の初期 J 値 $\boldsymbol{J}^{(0)}$ から価値反復は収束するが，数値例では $\boldsymbol{J}^{(0)}=\mathbf{0}$ とした．$\bar{g}(s_i,u_k)\ge 0,\forall(s_i,u_k)$ より $\boldsymbol{J}^{(0)}$ は実行可能領域内にある．

価値反復により $J^{(1)}(s_1)$ を求める操作は

$$\begin{aligned}&J^{(1)}(s_1)\\&=\min_{u_k}\left[\bar{g}(s_1,u_k)+p_{11}(u_k)J^{(0)}(s_1)+p_{12}(u_k)J^{(0)}(s_2)\right]\\&=\min\left(J^{(1)}(s_1|u_1),J^{(1)}(s_1|u_2)\right)\end{aligned}$$

ただし，$J^{(1)}(s_1|u_1)=\bar{g}(s_1,u_1)+p_{11}(u_1)J^{(0)}(s_1)+p_{12}(u_1)J^{(0)}(s_2)$, $J^{(1)}(s_1|u_2)=\bar{g}(s_1,u_2)+p_{11}(u_2)J^{(0)}(s_1)+p_{12}(u_2)J^{(0)}(s_2)$ である．すると，$J^{(1)}(s_1|u_1)$ において

$$\begin{aligned}&J^{(k+1)}(s_1|u_1)\\&=\bar{g}(s_1,u_1)+p_{11}(u_1)J^{(0)}(s_1)+p_{12}(u_1)J^{(0)}(s_2)\\&=(1-p_{11}(u_1))\,\tilde{J}(s_1|u_1)+p_{11}(u_1)J^{(0)}(s_1)\end{aligned}$$

ここで，$\tilde{J}(s_1|u_1)$ は次式を満たす値である．

$$\tilde{J}(s_1|u_1)=\bar{g}(s_1,u_1)+p_{11}(u_1)\tilde{J}(s_1|u_1)+p_{12}(u_1)J^{(0)}(s_2)$$

これは，$\boldsymbol{J}^{(0)}$ から右に半直線を伸ばし，$l(s_1,u_1)$ の境界線

$$J(s_1)=\bar{g}(s_1,u_1)+p_{11}(u_1)J(s_1)+p_{12}(u_1)(s_2)$$

との交点の $J(s_1)$ 座標である．更新される値 $J^{(1)}(s_1|u_1)$ は現在の座標 $J^{(0)}(s_1)$ と境界座標 $\tilde{J}(s_1|u_1)$ の内分である．同様に $J^{(1)}(s_1|u_2)$ は $J^{(0)}(s_1)$ と $\tilde{J}(s_1|u_2)$ の内分で

$$\tilde{J}(s_1|u_2) = \bar{g}(s_1, u_2) + p_{11}(u_2)\tilde{J}(s_1|u_2) + p_{12}(u_1)J^{(0)}(s_2)$$

である．両者を比較して，値の小さい方が $J^{(1)}(s_1)$ となる．同様に $J^{(1)}(s_2)$ も更新され，新しい $\boldsymbol{J}^{(1)}$ を得る．これが，価値反復で $\boldsymbol{J}^{(0)}$ から $\boldsymbol{J}^{(1)}$ を得る 1 回の反復で，結果的に実行可能領域内に超直方体 (この場合は長方形) を生成し，超直方体の頂点から頂点へ移動することになる．

これを反復し，$\boldsymbol{J}^{(0)}$ から $\boldsymbol{J}^{(m)}$ $(m = 1, 2, \ldots)$ を得ることにより，Fig. 3 のように $\boldsymbol{J}^*$ に収束する．

7. 結　　論

本論文では，プランニング問題を効率的に解くことを目的とし，LP 法について検討した．数値シミュレーションを行い，自己遷移確率が大きくなると，LP 法が DP 法よりも効率的になるという，新たな知見が得られた．その要因を，問題と各解法の幾何学的性質に基づいて説明した．本研究は科学研究費補助金に関連してなされたことを付記する．

参 考 文 献

1）R. S. Sutton and A. G. Barto: *Reinforcement Learning: An Introduction*, MIT Press (1998)
2）D. P. Bertsekas and J. N. Tsitsiklis: *Neuro-Dynamic Programming*, Athena Scientific (1996)
3）M. L. Puterman: *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, A Wiley-Interscience Publication (1994)
4）C. Guestrin, et al.: Efficient Solution Algorithms for Factored MDPs, *Journal of Artificial Intelligence Research*, **19**, 399/468 (2003)
5）Y. Ye: The Simplex and Policy-Iteration Methods Are Strongly Polynomial for the Markov Decision Problem with a Fixed Discount Rate, *Mathematics of Operations Research*, **36**-4, 593/603 (2011)
6）M. L. Littman, T. L. Dean, L. P. Kaelbling: On the Complexity of Solving Markov Decision Problems, *11th Annual Conf. Uncertainty Artificial Intelligence (UAI-95)*, 394/402 (1994)
7）Y. Mansour and S. Singh: On the complexity of policy iteration, *15th International Conf. Uncertainty Artificial Intelligence*, Morgan Kaufmann, San Francisco, 401/408 (1999)
8）Y. Ye: A New Complexity Result on Solving the Markov Decision Problem, *Mathematics of Operations Research*, **30**-3, 733/749 (2005)
9）G. B. Dantzig: *Linear Programming and Extensions*, Princeton Univ. Press, NJ (1963)
10）D. P. de Farias and B. V. Roy: Approximate Linear Programming for Average-cost Dynamic Programming, *Advances in Neural Information Processing Systems*, 1587/1594 (2002)
11）M. Petrik and S. Zilberstein: Constraint Relaxation in Approximate Linear Programs, *International Conference on Machine Learning*, Montreal, Canada (2009)
12）V. V. Desai, V. F. Farias, C. C. Moallemi: Approximate Dynamic Programming via a Smoothed Linear Program, *Operations Research*, **60**-3, 655/674 (2012)
13）泉田, 服部, 幸田: 強化学習における方策評価の効率化による学習の加速, 計測自動制御学会論文集, **49**-7, 696/702 (2013)
14）藤井, 泉田, 真野: 状態遷移モデルを逐次推定する強化学習の学習加速 計測自動制御学会論文集, **42**-1, 47/53 (2006)
15）R. E. Bellman: *Dynamic Programming*, Princeton Univ. Press, NJ (1957)
16）玉置 久: システム最適化, オーム社 (2005)
17）小島, 土谷, 水野, 矢部: 内点法, 朝倉書店 (2001)

《付　　録》

本論文中で用いた記号を定義する．$\boldsymbol{e}$ はすべての成分が 1 のベクトル，$\boldsymbol{I}$ は単位行列を表す．また，$\boldsymbol{e}_i$ は i 番目成分が 1 で残りが 0 のベクトルであり，$\boldsymbol{e}^K$ はすべての成分が 1 の K 次ベクトルを表す．

$$\begin{aligned}
\boldsymbol{\pi}(s_i) &\triangleq [\ \pi(s_i, u_1), \pi(s_i, u_2), \ldots, \pi(s_i, u_K)\ ]^T \\
\boldsymbol{\Pi}(s_i) &\triangleq \boldsymbol{e}_i \boldsymbol{\pi}^T(s_i) \\
\boldsymbol{\Pi} &\triangleq [\ \boldsymbol{\Pi}(s_1), \boldsymbol{\Pi}(s_2), \ldots, \boldsymbol{\Pi}(s_N)\ ] \\
\boldsymbol{p}_{i|j\neq 0}(u_k) &\triangleq [\ p_{i1}(u_k), p_{i2}(u_k), \ldots, p_{iN}(u_k)\ ]^T \\
\boldsymbol{P}_i &\triangleq [\ \boldsymbol{p}_{i|j\neq 0}(u_1), \boldsymbol{p}_{i|j\neq 0}(u_2), \ldots, \boldsymbol{p}_{i|j\neq 0}(u_K)\ ]^T \\
\boldsymbol{P} &\triangleq [\ \boldsymbol{P}_1^T, \boldsymbol{P}_2^T, \ldots, \boldsymbol{P}_N^T\ ]^T \\
\boldsymbol{p}_i(u_k) &\triangleq [\ p_{i0}(u_k), p_{i1}(u_k), \ldots, p_{iN}(u_k)\ ]^T \\
\boldsymbol{p}_i &\triangleq [\ \boldsymbol{p}_i^T(u_1), \boldsymbol{p}_i^T(u_2), \ldots, \boldsymbol{p}_i^T(u_K)\ ]^T \\
\boldsymbol{p} &\triangleq [\ \boldsymbol{p}_1^T, \boldsymbol{p}_2^T, \ldots, \boldsymbol{p}_N^T\ ]^T \\
\boldsymbol{g}(s_i, u_k) &\triangleq [g(s_i, u_k, s_0), \ldots, g(s_i, u_k, s_N)]^T \\
\bar{g}(s_i, u_k) &\triangleq \boldsymbol{p}_i^T(u_k)\boldsymbol{g}(s_i, u_k) \\
\bar{\boldsymbol{g}}(s_i) &\triangleq [\bar{g}(s_i, u_1), \bar{g}(s_i, u_2), \ldots, \bar{g}(s_i, u_K)]^T \\
\bar{\boldsymbol{g}} &\triangleq \left[\bar{\boldsymbol{g}}^T(s_1), \bar{\boldsymbol{g}}^T(s_2), \ldots, \bar{\boldsymbol{g}}^T(s_N)\right]^T \\
\boldsymbol{J} &\triangleq [J(s_1), J(s_2), \ldots, J(s_N)]^T \\
\boldsymbol{E} &\triangleq \mathrm{diag}[\boldsymbol{e}^K \ldots \boldsymbol{e}^K]
\end{aligned}$$

［著　者　紹　介］

泉　田　　啓（正会員）

1988 年大阪府立大学大学院博士前期課程修了．同年同大学工学部助手，94 年同助教授，2002 年金沢大学助教授，2007 年同教授，2008 年京都大学工学研究科教授，現在に至る．96〜97 年ミシガン州立大学客員教授など兼任．92 年 AIAA GNC 最優秀発表論文賞，94 年システム制御情報学会賞，2002 年とやま賞 (学術研究部門) など受賞．ロボットの知能化，動物の運動知能などの研究に従事．日本ロボット学会などの会員．博士 (工学)．

天　野　恒　佑

京都大学大学院工学研究科修士課程（航空宇宙工学専攻）．ロボットの知能化や強化学習などの研究に従事．