バイオメディカル・ファジィ・システム学会誌 Vol.22, No.2, pp.21－32（2020）


[Original article]

（2020年10月3日 Accepted）

# 動的計画法を用いた適応的な輪作に関する一考察

前田 康成[1]

1）北見工業大学・地域未来デザイン工学科

**要約：** 日本農業では収入の増大が必要とされている．農業における収入の最大化については従来から数多く研究されている．ある従来研究では輪作における収入を作物の価格が変動しないという仮定のもとで最大化している．しかし，一般的に作物の価格は変動するものである．そこで，本研究では作物の価格がマルコフ連鎖に従って変動するという仮定のもとで輪作における収入を最大化する．収入の最大化には動的計画法を利用する．提案方法の有効性をいくつかのシミュレーション例で示す．提案方法の収入は比較対象よりも大きく，提案方法による変動価格に応じた適応的な作物選択が確認された．提案方法の最適解は理論的限界でもあり，理論的限界は他の輪作方法の評価に利用できる．提案方法はスマート農業における意思決定の自動化への貢献が期待される．

**キーワード：** 輪作，収益最大化，価格変動，適応的，動的計画法，マルコフ連鎖

# A Note on Adaptive Crop Rotation using Dynamic Programming

Yasunari MAEDA[1]

*1) School of Regional Innovation and Social Design Engineering, Kitami Institute of Technology*

***Abstract:*** *Increasing profit is desired in Japanese agriculture. There is many previous research on profit maximization in agriculture. In previous research the profit of crop rotation is maximized. But the previous research is studied under the condition that the price of crop is constant. In General price of crop changes. In this research the profit of crop rotation is maximized under the condition that the price of crop changes depending on Markov chain. The profit is maximized by dynamic programming. The effectiveness of the proposed method is shown by some simulation examples. The profit of the proposed method is greater than that of the comparison target. In the results of the proposed method the adaptive crop selection according to the fluctuating crop price was confirmed. The optimal solution of the proposed method is also a theoretical limit. The theoretical limit can be used to evaluate other crop rotation methods. The proposed method is expected to contribute to the automation of decision making in smart agriculture.*

***Keywords:*** *crop rotation, profit maximization, price fluctuation, adaptive, dynamic programming, Markov chain*

Yasunari MAEDA

165 Koen-cho, Kitami-shi, Hokkaido, 090-8507, Japan

Phone: +81-157-26-9328, Fax: +81-157-26-9344, E-mail: maedaya@mail.kitami-it.ac.jp

## 1. はじめに

近年の情報通信技術（ICT）の発展に伴い，日本の農業でもICTを積極的に利用したスマート農業が注力されている[1]-[3]．スマート農業が注力される背景には日本農業が抱える高齢化・後継者不足・低収入などの問題が関係している．本研究では特に日本農業における低収入に着目する．

数理工学分野では農業収入の最大化を目的とした研究が従来から数多く実施されている[4]-[8]．農業収入の最大化に関する研究には，既に作付けされた（植えられた）特定の作物の栽培管理に関する研究[4][5]や，作付けする（植える）作物の選択（輪作）に関する研究[6]-[8]などがある．従来研究の中にはスマート農業という言葉が使われるようになる以前の研究もあるが，これらの研究はスマート農業における栽培管理や栽培作物選択に関する自動化（意思決定の自動化）技術の基礎研究に相当する．前者の栽培管理では，栽培作物の状況に応じて肥料や農薬の散布，間引きなどの農作業を実施することにより収穫量を制御する．農作業には高コストの作業から低コストの作業まであるので，栽培管理は収穫量と作物価格（作物の収穫時の買取り価格）で決まる売上と農作業コストのトレードオフの問題になる．

本研究では特に後者の作物の選択問題（輪作問題）に着目する．作物の選択問題では，作付けから収穫までの栽培管理に関する意思決定は対象とせずに，作物の選択のみを対象とする．何も制約がなければ，売上が最大になる作物を選択し続ければよい．しかし，作物には同じ場所に同じ作物（あるいは同じ科（イネ科など）の作物）を続けて作付けすると連作障害によって収穫量が減少することがある．よって，作物の選択に関しては直近の1回の選択のみではなく，将来の選択も考慮した計画が重要である．従来研究[6]では有限の数年間における総収入を最大化するための計画的な作物の選択問題を検討している．しかし，作物の価格が固定（価格は変動しないと仮定）されている．現実には価格は変動するものであり，価格固定の場合の最適な作物選択と，価格が変動する場合の最適な作物選択が一致するとは限らない．

他方，従来研究[7][8]では作物の価格が変動する条件のもとでの収入の最大化を検討している．価格が変動するという現実に近い条件を採用しているが，対象とする作物の選択は1回のみである．上記のとおり，作物の選択問題では連作障害を考慮した計画的な作物選択が重要である．従来研究によって直近の1回の選択における連作障害を考慮することはできるが，直近の1回の選択が次回以降の選択に与える影響を考慮することはできない．一般的に，当該回の意思決定と次回以降の意思決定が独立ではない意思決定問題では，直近の1回の決定を対象とした場合の最適解と複数回の決定を対象とした場合の最適解は一致するとは限らない．

そこで，本研究ではより現実に近い問題設定として，作物の価格が変動する条件のもとで有限の数年間の総収益を最大化する作物の選択問題を検討する．具体的には作物価格の変動を表現するために確率モデルのマルコフ連鎖[9]を採用し，総収益の最大化問題を多段的な意思決定問題として動的計画法[9]で解く提案アルゴリズムを提案する．提案アルゴリズムでは，作物価格の変動を考慮した適応的な作物選択により，対象の数年間の総収益の期待値を最大化できる．

本研究の提案アルゴリズムは最適解を算出するので，当該最適解は同問題設定に対する理論的限界でもある．よって，提案アルゴリズムを利用して同問題設定に対する他の方法（近似解法や経験則など）の評価も可能である．本研究の提案アルゴリズムはスマート農業における意思決定の自動化への貢献が期待される．

2章で準備としてマルコフ連鎖，動的計画法，本研究で使用する記号と問題設定について説明し，3章では本研究の提案アルゴリズムの比較対象である固定価格の場合の輪作（作物選択）アルゴリズムを説明する．4章で提案アルゴリズムを提案し，5章で提案アルゴリズムと固定価格の場合の比較対象とのシミュレーションによる比較結果を報告する．6章で考察と今後の改題を述べ，最後に7章でまとめを述べる．

## 2. 準備

### 2.1 動的計画法

本研究では，農業における総収益の期待値を最大化するために，最適化方法の1つである動的計画法を用いる．本節では動的計画法について説明する．動的計画法では多段的な意思決定問題を扱う．多段的の解釈は一意ではないが，本研究の作物の選択問題のように時系列に従って行う複数（有限）回の意思決定を扱う場合が多い．

例えば，有限の$T$期間の意思決定問題を考える．$t$期の状態を示す変数を$X_t$，$t$期に選択する行動を示す変数を$Y_t$，状態集合を$S = \{s_1, s_2, \dots, s_{|S|}\}$，各期で意思決定の対象となる選択可能な行動の集合を$A = \{a_1, a_2, \dots, a_{|A|}\}$とする．$t$期の状態$X_t$で行動$Y_t$を選択すると，確定的に利得$r(X_t, Y_t)$が得られ，$t+1$期の状態$X_{t+1}$に遷移す

るとする．利得が示すものは対象となる実問題によって異なるが，本研究のように収益最大化の場合にはお金などが相当する．利得や次の状態は確定的に生起する場合と確率的に生起する場合が考えられるが，ここでは簡便のため確定的な場合で説明する．

ここでの目的は総利得$\sum_{i=1}^{T} r(X_i, Y_i)$の最大化である．この最大化を再帰的な処理で実施するのが動的計画法である．実際の意思決定は時系列に従って1期から$T$期に向けて順番に行うが，最適化問題を動的計画法で解く際には$T$期から1期に向けて遡りながら解く．動的計画法では，$t$期（$1 \leq t < T$）の状態$X_t$における意思決定を次式で行う．

$$V(X_t, t) = \max_{a_i \in A(X_t)} r(X_t, a_i) + V(X_{t+1}, t+1), \quad (1)$$

ただし，$A(X_t)$は状態$X_t$で選択可能な行動の集合，$V(X_t, t)$は$t$期以降の総利得の最大値である．$t$期の状態$X_t$における最適な意思決定は式(1)の右辺を最大化する行動である．

式(1)では次の期の$V(X_{t+1}, t+1)$を再帰的に呼び出している．動的計画法の計算量は式(1)の実施回数に依存する．よって，1度計算した$V(X_{t+1}, t+1)$の値を記憶しておいて，再帰呼び出しの際に記憶しておいた値を利用することによって，計算量の膨大化を回避することが多い．

## 2.2 マルコフ連鎖

本研究では，作物の価格変動を表現するために時間の経過に伴って確率的に状態が変化する確率モデルであるマルコフ連鎖を使用する．本節ではマルコフ連鎖について簡単に説明する．マルコフ連鎖は状態$s_i \in S = \{s_1, s_2, \ldots, s_{|S|}\}$と，状態$s_i$から次の期に状態$s_j$へ遷移する遷移確率$\Pr(s_j|s_i)$で構成される．遷移確率は現在の状態のみに依存して，それ以前の状態とは独立である．この性質がマルコフ性である．

遷移確率を用いて数期先にある状態にいる確率を計算できる．また，無限回の遷移を繰返した場合に平均的にある状態にいる確率も計算できる．この確率を定常確率$\Pr(s_i)$と呼ぶ．例えば，状態数$|S| = 2$の場合の定常確率は以下の式(2)，式(3)，式(4)による連立方程式を解くことで求められる．

$$\Pr(s_1|s_1)\Pr(s_1) + \Pr(s_1|s_2)\Pr(s_2) = \Pr(s_1). \quad (2)$$

$$\Pr(s_2|s_1)\Pr(s_1) + \Pr(s_2|s_2)\Pr(s_2) = \Pr(s_2). \quad (3)$$

$$\Pr(s_1) + \Pr(s_2) = 1. \quad (4)$$

式(2)，式(3)，式(4)の連立方程式を解くと，定常確率は式(5)，式(6)のようになる．

$$\Pr(s_1) = \frac{\Pr(s_1|s_2)}{1+\Pr(s_1|s_2)-\Pr(s_1|s_1)}. \quad (5)$$

$$\Pr(s_2) = \frac{1-\Pr(s_1|s_1)}{1+\Pr(s_1|s_2)-\Pr(s_1|s_1)}. \quad (6)$$

## 2.3 記号と問題設定（主に本研究と比較対象で共通）

本研究で使用する記号や問題設定について説明する．本研究では作物の選択問題を扱うが，同じ科の作物を続けて栽培したことによる連作障害を確実に回避するために，連作障害が起きる可能性のある年数（輪作年限）の間は同じ科の作物を栽培しないようにする輪作[10]を対象とする．また，農業には1年間に2回異なる作物を作付けする二毛作もあるが，本研究では作付けは1年間に1回とする．

本研究では作物の価格が変動する条件のもとで有限の数年間の総収益を最大化する輪作アルゴリズムを提案するが，比較対象として従来研究[6]と同様に価格が固定のもとで総収益を最大化する輪作アルゴリズムを想定する．なお，本研究と従来研究[6]とでは，価格以外にも連作障害のモデル化の仕方，作物ごとの栽培コストの導入（本研究）／未導入（従来研究），使用する最適化手法なども異なる．従来研究と本研究の直接的な比較は難しいため，本研究の問題設定のうち作物の価格に関してのみ変動から固定に変更した問題設定のもとで総収益を最大化する輪作アルゴリズムを比較対象とする．

作物の価格固定の場合の方がわかりやすい（簡便な）問題設定となるため，最初に価格固定の場合の記号について説明する．$a_i$は$i$番目の作物を示し，$A = \{a_1, a_2, \ldots, a_{|A|}\}$は作物集合である．$c_i$は作物の科を示し，$C = \{c_1, c_2, \ldots, c_{|C|}\}$は科の集合である．$n(a_i)$は作物$a_i$の輪作年限，$c(a_i) \in C$は作物$a_i$の属する科である．作物$a_i$を栽培した場合，同じ場所で$n(a_i)$年以内に同じ科$c(a_i)$に属する作物（作物$a_i$も含む）を栽培すると連作障害が起きる．実際には必ず起きるわけではないが，本研究では議論を簡便にするために輪作年限の間隔を空けないと連作障害が起きると仮定する．$b(a_i)$は作物$a_i$を栽培するのに必要なコスト（万円），$e(a_i)$は作物$a_i$の収穫量（トン），$f(a_i)$は作物$a_i$の価格（万円／トン）を示す．

連作障害が該当する作物は当該年の選択肢から排除する．有限の$T$年間の作物選択問題を考える．価格固定の

場合の作物選択問題における状態は過去$N$年間の栽培作物の情報で構成される．栽培作物の履歴情報は連作障害に該当するかどうかの判断のために利用され，$N$は次式で算出される．

$$N = \max_{a_i \in A} n(a_i). \tag{7}$$

価格固定の場合の$t$年目の状態を示す変数を$X_t$，栽培作物を$Y_t$とすると，

$$X_t = \left(Y_{t-N}, \dots, Y_{t-2}, Y_{t-1}\right), \tag{8}$$

である．

### 2.4 記号と問題設定（本研究）

次に本研究の本来の問題設定である価格が変動する場合の記号について説明する．各種記号のほとんどは価格固定の場合と同様であり，ここでは異なる部分のみ説明する．作物の価格は各作物の市場における相場の状態に依存すると仮定する．$s_i$は$i$番目の相場状態を示し，$S = \{s_1, s_2, \dots, s_{|S|}\}$は相場状態集合である．$f(s_j, a_i)$は相場状態$s_j$における作物$a_i$の価格（万円／トン）を示す．なお，相場状態の定義は全作物で共通とするが，年単位で変動する相場状態は作物ごとの相場状態とする．相場状態の変動はマルコフ連鎖に従う．作物ごとに異なるパラメータ（遷移確率）のマルコフ連鎖によるモデルも考えられるが，議論を簡便にするため本研究ではマルコフ連鎖の遷移確率も全作物で共通とする．$\Pr(s_j|s_i)$は相場状態$s_i$が1年後に相場状態$s_j$に遷移（変動）するマルコフ連鎖の遷移確率である．

価格が変動する場合の作物選択問題における状態は過去$N$年間の栽培作物と前年（過去1年間）の全作物の相場状態の情報で構成される．栽培作物の履歴情報は連作障害に該当するかどうかの判断のために利用され，$N$は式(7)による．前年の作物の相場状態の情報は価格の予測に利用される．価格が変動する場合の$t$年目の状態を示す変数を$X_t$，栽培作物を$Y_t$，作物の相場状態を$W_t$とすると状態は次式のようになる．

$$X_t = \left(Y_{t-N}, \dots, Y_{t-2}, Y_{t-1}, W_{t-1}\right), \tag{9}$$

ただし，

$$W_{t-1} = \left(W_{t-1,1}, W_{t-1,2}, \dots, W_{t-1,|A|}\right), \tag{10}$$

$W_{t-1,i} \in S$はt − 1年目の作物$a_i$の相場状態である．

## 3. 価格固定の輪作アルゴリズム（提案アルゴリズムの比較対象）

本研究の提案アルゴリズムに対する比較対象である，価格固定の場合の輪作アルゴリズムを以下に示す．動的計画法での処理は$T$年目から1年目まで遡りながら実施する．処理は最後の$T$年目と$t$年目（$1 \le t < T$）で異なる．$T$年目の処理を以下に示す．式(11)では$T$年目の状態が$X_T$のもとで，連作障害に該当する作物を選択肢から排除して収益の最大値を算出している．式(11)の右辺の最大値に対応する作物が$T$年目の状態が$X_T$である条件のもとで栽培する最適な作物$Y_T$である．

$$V(X_T, T) = \max_{a_i \in A(X_T)} e(a_i) f(a_i) - b(a_i), \tag{11}$$

ただし，

$$A(X_T) = \left\{a_i \middle| \forall j, 1 \le j \le N, c(a_i) \ne c\left(Y_{T-j}\right) \vee n\left(Y_{T-j}\right) < j\right\}. \tag{12}$$

次に$t$年目（$1 \le t < T$）の処理を以下に示す．式(13)では$t$年目の状態が$X_t$のもとで，連作障害に該当する作物を選択肢から排除して$t$年目以降の総収益の最大値を算出している．式(13)の右辺の最大値に対応する作物が$t$年目の状態が$X_t$である条件のもとで栽培する最適な作物$Y_t$である．

$$V(X_t, t) = \max_{a_i \in A(X_t)} e(a_i) f(a_i) - b(a_i) + V\left(X_{t+1}, t+1\right), \tag{13}$$

ただし，

$$A(X_t) = \left\{a_i \middle| \forall j, 1 \le j \le N, c(a_i) \ne c\left(Y_{t-j}\right) \vee n\left(Y_{t-j}\right) < j\right\}. \tag{14}$$

式(11)および式(13)を用いて$T$年目から1年目まで遡りながら各年のすべての状態候補について処理を実施することによって，作物の価格固定のもとで$T$年間の総収益を最大化する栽培作物を選択できる．価格固定の場合は確定的なモデルのため，総収益そのものの最大化である．

## 4. 価格が変動する場合の輪作アルゴリズム（本研究の提案アルゴリズム）

本研究の提案アルゴリズムである，作物価格がマルコフ連鎖に従って変動する場合の輪作アルゴリズムを以下に示す．価格が変動する場合も動的計画法での処理は$T$年目から1年目まで遡りながら実施する．処理は最後の$T$年目と$t$年目（$1 \le t < T$）で異なる．$T$年目の処理を以

下に示す．式(15)は$T$年目の状態が$X_T$のもとで，連作障害に該当する作物を選択肢から排除して収益の期待値の最大値を算出している．式(15)の右辺の最大値に対応する作物が$T$年目の状態が$X_T$である条件のもとで栽培する最適な作物$Y_T$である．

$$V(X_T,T)=\max_{a_i\in A(X_T)}$$

$$\sum_{s_k\in S}\Pr\left(s_k\middle|W_{T-1,i}\right)\left(e(a_i)f(s_k,a_i)-b(a_i)\right),\quad(15)$$

ただし，

$$A(X_T)=$$
$$\left\{a_i\middle|\forall j,1\le j\le N,c(a_i)\ne c\left(Y_{T-j}\right)\vee n\left(Y_{T-j}\right)<j\right\}.\quad(16)$$

次に$t$年目（$1\le t<T$）の処理を以下に示す．式(17)は$t$年目の状態が$X_t$のもとで，連作障害に該当する作物を選択肢から排除して$t$年目以降の総収益の期待値の最大値を算出している．式(17)の右辺の最大値に対応する作物が$t$年目の状態が$X_t$である条件のもとで栽培する最適な作物$Y_t$である．

$$V(X_t,t)=\max_{a_i\in A(X_t)}\sum_{W_t\in S^{|A|}}\prod_{k=1}^{|A|}\Pr\left(W_{t,k}\middle|W_{t-1,k}\right)$$

$$\left(e(a_i)f\left(W_{t,i},a_i\right)-b(a_i)+V\left(X_{t+1},t+1\right)\right),\quad(17)$$

ただし，

$$A(X_t)=$$
$$\left\{a_i\middle|\forall j,1\le j\le N,c(a_i)\ne c\left(Y_{t-j}\right)\vee n\left(Y_{t-j}\right)<j\right\}.\quad(18)$$

式(15)および式(17)を用いて$T$年目から1年目まで遡りながら各年のすべての状態候補について処理を実施することによって，作物の価格が変動するもとで$T$年間の期待総収益を最大化する栽培作物を選択できる．価格が変動する場合は確率モデルである相場状態のマルコフ連鎖を含むため，総収益の期待値である期待総収益の最大化である．

## 5. シミュレーション例

提案アルゴリズムの有効性を検証するためにシミュレーション例を紹介する．

### 5.1 設定

シミュレーション例の設定を以下に示す．作物数は$|A|=6$，科の数は$|C|=3$，相場状態数は$|S|=2$，期間は$T=5$年間とする．

提案アルゴリズムとの比較対象は価格固定の場合だが，価格固定の場合の価格は提案モデルのマルコフ連鎖による相場状態の定常確率で価格の期待値（平均値）を算出したものを使用する．シミュレーション例では，同じ相場状態に遷移する（同じ状態を維持する）確率について$\Pr\left(s_1\middle|s_1\right)=\Pr\left(s_2\middle|s_2\right)$の条件の場合を紹介する．この条件のもとでの定常確率は$\Pr\left(s_1\right)=\Pr\left(s_2\right)=0.5$である．

各種設定（輪作年限$n(a_i)$，科$c(a_i)$，コスト$b(a_i)$，収穫量$e(a_i)$，変動価格$f\left(s_j,a_i\right)$，固定価格$f(a_i)$）について2つのパターン（パターン1とパターン2）を表1から表6に示す．表1から表3がパターン1で表4から表6がパターン2である．

表1.　作物$a_i$の輪作年限$n(a_i)$，科$c(a_i)$，コスト$b(a_i)$，収穫量$e(a_i)$（パターン1）

| 作物 | $n(a_i)$ | $c(a_i)$ | $b(a_i)$ | $e(a_i)$ |
|---|---|---|---|---|
| $a_1$ | 1 | $c_1$ | 500 | 100 |
| $a_2$ | 2 | $c_1$ | 400 | 100 |
| $a_3$ | 1 | $c_2$ | 450 | 100 |
| $a_4$ | 2 | $c_2$ | 350 | 100 |
| $a_5$ | 1 | $c_3$ | 480 | 100 |
| $a_6$ | 2 | $c_3$ | 380 | 100 |

表2.　作物$a_i$の変動価格$f\left(s_j,a_i\right)$と固定価格$f(a_i)$（パターン1）

| 作物 | $f(s_1,a_i)$ | $f(s_2,a_i)$ | $f(a_i)$ |
|---|---|---|---|
| $a_1$ | 12 | 8 | 10 |
| $a_2$ | 13 | 9 | 11 |
| $a_3$ | 13 | 9 | 11 |
| $a_4$ | 15 | 11 | 13 |
| $a_5$ | 11 | 7 | 9 |
| $a_6$ | 13 | 9 | 11 |

解釈が容易なように収穫量は一定に設定した．相場状態$s_1$は価格が高い状態，相場状態$s_2$は価格が低い状態である．固定価格$f(a_i)$は相場状態の定常確率$\Pr(s_i)$で相場状態に依存した変動価格$f\left(s_j,a_i\right)$の期待値を算出したものである．表3と表6は価格が変動する場合の動的計画法の式(15)および式(17)における各パターンの各作物$a_i$に対応した各相場状態$s_j$のもとでの収益$e(a_i)f\left(a_i,s_j\right)-b(a_i)$と，固定価格の場合の動的計画法の式(11)および

式(13)における各パターンの各作物$a_i$に対する収益$e(a_i)f(a_i)-b(a_i)$である．上記の設定は市場データなどを参考にした著者による架空の設定である．提案アルゴリズムに関するより現実的な検証および改善のために必要な実データに基づく検証は今後の課題である．

表3.　変動価格の収益$(s_j)=e(a_i)f(a_i,s_j)-b(a_i)$と固定価格の収益(固)$=e(a_i)f(a_i)-b(a_i)$
(パターン1)

| 作物 | 収益$(s_1)$ | 収益$(s_2)$ | 収益(固) |
|---|---|---|---|
| $a_1$ | 700 | 300 | 500 |
| $a_2$ | 900 | 500 | 700 |
| $a_3$ | 850 | 450 | 650 |
| $a_4$ | 1150 | 750 | 950 |
| $a_5$ | 620 | 220 | 420 |
| $a_6$ | 920 | 520 | 720 |

表4.　作物$a_i$の輪作年限$n(a_i)$，科$c(a_i)$，コスト$b(a_i)$，収穫量$e(a_i)$　(パターン2)

| 作物 | $n(a_i)$ | $c(a_i)$ | $b(a_i)$ | $e(a_i)$ |
|---|---|---|---|---|
| $a_1$ | 1 | $c_1$ | 400 | 100 |
| $a_2$ | 2 | $c_1$ | 500 | 100 |
| $a_3$ | 1 | $c_2$ | 350 | 100 |
| $a_4$ | 2 | $c_2$ | 450 | 100 |
| $a_5$ | 1 | $c_3$ | 380 | 100 |
| $a_6$ | 2 | $c_3$ | 480 | 100 |

表5.　作物$a_i$の変動価格$f(s_j,a_i)$と固定価格$f(a_i)$
(パターン2)

| 作物 | $f(s_1,a_i)$ | $f(s_2,a_i)$ | $f(a_i)$ |
|---|---|---|---|
| $a_1$ | 13 | 9 | 11 |
| $a_2$ | 12 | 8 | 10 |
| $a_3$ | 15 | 11 | 13 |
| $a_4$ | 13 | 9 | 11 |
| $a_5$ | 13 | 9 | 11 |
| $a_6$ | 11 | 7 | 9 |

ここで，パターン1とパターン2の設定の特徴について説明する．まず表1と表4では共通して，作物$a_1$と$a_2$が科$c_1$に属し，作物$a_3$と$a_4$が科$c_2$に属し，作物$a_5$と$a_6$が科$c_3$に属し，各科の作物のうち番号が小さい作物（$a_1$，$a_3$，$a_5$）の輪作年限が1年，番号が大きい作物（$a_2$，$a_4$，$a_6$）の輪作年限が2年である．

表6.　変動価格の収益$(s_j)=e(a_i)f(a_i,s_j)-b(a_i)$と固定価格の収益(固)$=e(a_i)f(a_i)-b(a_i)$
(パターン2)

| 作物 | 収益$(s_1)$ | 収益$(s_2)$ | 収益(固) |
|---|---|---|---|
| $a_1$ | 900 | 500 | 700 |
| $a_2$ | 700 | 300 | 500 |
| $a_3$ | 1150 | 750 | 950 |
| $a_4$ | 850 | 450 | 650 |
| $a_5$ | 920 | 520 | 720 |
| $a_6$ | 620 | 220 | 420 |

表1と表4の残りの情報及び表2と表5の情報は，表3と表6に集約されている．パターン1の表3では，すべての科において輪作年限が2年の作物の方が収益が大きい．よって，収益が表3の縦の1つの列の値のみ（例えば収益（固）のみ）であれば，作物$a_2$，$a_4$，$a_6$の3年間のローテーションでの輪作が良さそうに思われるパターンである．

次にパターン2の表6では，すべての科において輪作年限が1年の作物の方が収益が大きい．輪作年限が1年の作物の中での収益の大きさの順番は1位が$a_3$，2位が$a_5$，3位が$a_1$である．収益が大きいのは輪作年限が1年の作物なので，収益が表6の縦の1つの列の値のみ（例えば収益（固）のみ）であれば，作物$a_3$，$a_5$の2年間のローテーションでの輪作が良さそうに思われるパターンである．

2つのパターンは表1～表6を一見すると違いが見えにくいが，上記のように単純なローテーションの範囲で考えた場合に異なる輪作が想定される2つのパターンである．

### 5.2　結果

実施したシミュレーションのおおまかな流れを図1にフローチャートで示す．

図1では，最初に表1などの各種情報を設定し，相場状態の定常確率，固定価格を計算する．次に従来研究相当の価格固定の場合のアルゴリズムと本研究の提案である価格変動の場合のアルゴリズムでT = 5年間のすべての状態候補に関して栽培作物を算出し記憶する．次に相場状態がマルコフ連鎖で変化するシミュレーション環境下で両アルゴリズムの算出済の選択作物を栽培した場合の収益を比較するシミュレーションを10万回繰返す．

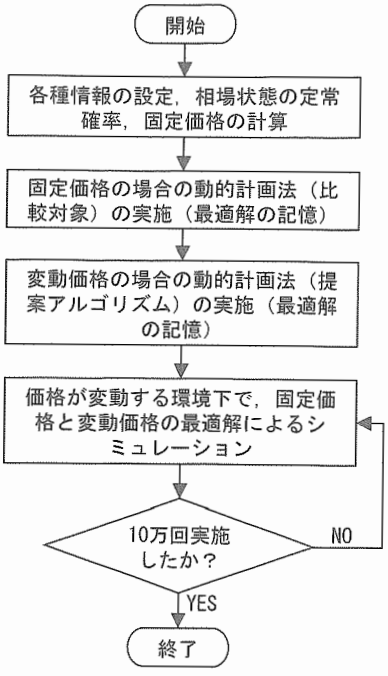



図1.　シミュレーションの流れ

シミュレーション結果のうち，T = 5年間の収益の10万回の試行の平均値（万円）を図2（パターン1）と図3（パターン2）に示す．収益（変）が変動価格の場合の提案アルゴリズムでの収益の平均値，収益（固）が比較対象の固定価格の場合の収益の平均値である．なお，シミュレーション中の初期状態は，状態を構成する過去2年分の栽培履歴の科は等確率で発生させ，前年の6作物の相場状態はマルコフ連鎖の定常確率に従って発生させた．

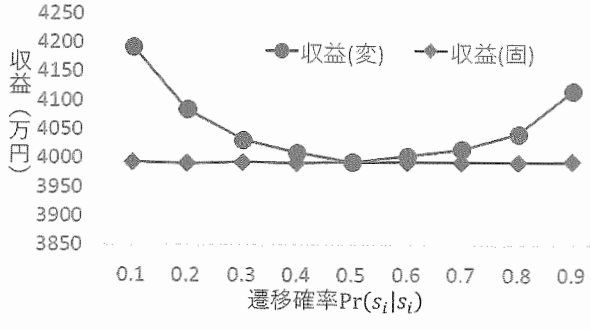



図2.　変動価格と固定価格の比較（パターン1）

図2，図3より，総収益についてはパターン1とパターン2でほぼ同様の結果である．図2，図3ともに同じ相場状態に遷移する（同じ状態を維持する）遷移確率Pr($s_i|s_i$)が0.9寄りの大きい場合または0.1寄りの小さい場合には提案アルゴリズムによる価格変動に対応した場合の収益が価格固定を想定した場合の収益よりも大きく，差が大きいところで1割程度である．他方，遷移確率Pr($s_i|s_i$)が0.5付近では差は無い．これは，遷移確率Pr($s_i|s_i$)が0.9寄りまたは0.1寄りの場合には提案アルゴリズム中で相場状態の遷移先を確率的に予測して期待値を算出する部分で従来研究相当の価格固定の場合よりもより適切に作物を選択しているからである．他方，遷移確率Pr($s_i|s_i$)が0.5付近では提案アルゴリズムで遷移先を予測しても遷移先が$s_1$と$s_2$のどちらになるかは五分五分のために従来研究相当の価格固定の場合と同様の作物選択になっているからである．

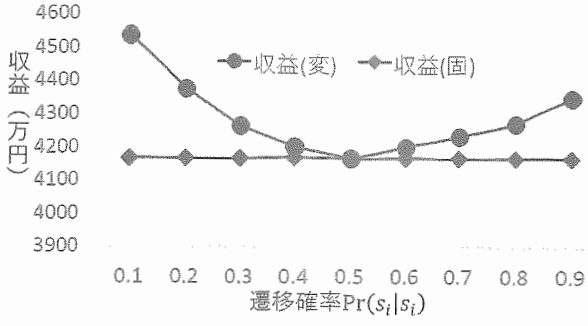



図3.　変動価格と固定価格の比較（パターン2）

次に価格変動を考慮した提案アルゴリズムと価格固定の場合の作物選択の具体的な差について説明する．価格固定の場合のアルゴリズムによる作物選択は，5.1節で2つのパターンの特徴を説明した際に紹介した単純なローテーションによる輪作になる．作物選択結果の例を表7（パターン1）と表8（パターン2）に示す．

表7.　価格固定の場合のパターン1での選択作物

| | 例1 | 例2 | 例3 |
|---|---|---|---|
| $t$ | 1 | 2 | 3 |
| $Y_{t-2}$ | $a_2$ | $a_4$ | $a_6$ |
| $Y_{t-1}$ | $a_4$ | $a_6$ | $a_2$ |
| $Y_t$ | $a_6$ | $a_2$ | $a_4$ |

パターン1ではすべての科で輪作年限が2年の作物の収益が1年の作物よりも大きいので，表7のように輪作年限が2年の作物$a_2$，$a_4$，$a_6$の3年間のローテーションになる．パターン2ではすべての科で輪作年限が1年の作物の収益が2年の作物よりも大きいので，基本的に表8のように輪作年限が1年の作物のうち収益が大きい上位2つの作物$a_3$，$a_5$の2年間のローテーションになる．なお，パターン2については初期状態次第で，1年目または2年目の

作物が$a_1$になることもあるが，基本は作物$a_3$，$a_5$のローテーションである．例えば，2年前が作物$a_4$で1年前が作物$a_6$の初期状態に対しては輪作年限の制約によって連作障害を回避するために科$c_1$の作物しか選択できないため，作物$a_1$を選択する．

表8.　価格固定の場合のパターン2での選択作物

| | 例1 | 例2 |
|---|---|---|
| $t$ | 1 | 2 |
| $Y_{t-2}$ | $a_3$ | $a_5$ |
| $Y_{t-1}$ | $a_5$ | $a_3$ |
| $Y_t$ | $a_3$ | $a_5$ |

価格変動を考慮した提案アルゴリズムの作物選択結果の例を表9（パターン1）と表10（パターン2）に示す．表9，表10ともに$\Pr(s_i|s_i)=0.1$の場合であり，相場状態が前年と同じ確率が0.1，異なる確率が0.9である．

表9.　提案アルゴリズムのパターン1での選択作物

$(\Pr(s_i|s_i)=0.1)$

| | 例1 | 例2 | 例3 | 例4 | 例5 | 例6 |
|---|---|---|---|---|---|---|
| $t$ | 1 | 1 | 2 | 2 | 3 | 3 |
| $Y_{t-2}$ | $a_2$ | $a_2$ | $a_4$ | $a_4$ | $a_6$ | $a_6$ |
| $Y_{t-1}$ | $a_4$ | $a_4$ | $a_6$ | $a_6$ | $a_2$ | $a_2$ |
| $W_{t-1,1}$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ |
| $W_{t-1,2}$ | $s_2$ | $s_2$ | $s_2$ | $s_1$ | $s_2$ | $s_2$ |
| $W_{t-1,3}$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ |
| $W_{t-1,4}$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ | $s_1$ |
| $W_{t-1,5}$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ |
| $W_{t-1,6}$ | $s_2$ | $s_1$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ |
| $Y_t$ | $a_6$ | $a_5$ | $a_2$ | $a_1$ | $a_4$ | $a_3$ |

表9の例1，例2は表7の例1と対応（年$t$，2年前の作物$Y_{t-2}$，1年前の作物$Y_{t-1}$が同じ）している．表9の例1では前年のすべての作物の相場状態が低価格の$s_2$なので，次はすべての作物の相場状態が高価格の$s_1$になる確率が高い状況である．この状況では表7の例1と同じ作物$a_6$が選択されている．他方，表9の例2では，作物$a_6$の次の相場状態は価格が低い$s_2$になる確率が高く，作物$a_5$の次の相場状態は価格が高い$s_1$になる確率が高いため，表7の例1とは異なる作物$a_5$が選択されている．なお，表9の例1，例2では2年前の作物$Y_{t-2}$が科$c_1$の輪作年限2年の$a_2$，1年前の作物$Y_{t-1}$が科$c_2$の輪作年限2年の$a_4$なので，連作障害を回避するために作物の候補は$a_5$と$a_6$のみである．表9の例2では，前年の相場状態に応じて適応的に単純なローテーションとは異なる作物を選択している．同様に，表9の例3，例4は表7の例2と対応し，表9の例5，例6は表7の例3と対応している．表9の例3，例5ではそれぞれ単純なローテーションの表7の例2，例3と同じ作物を選択し，表9の例4，例6ではそれぞれ前年の相場状態に応じて適応的に単純なローテーションの表7の例2，例3とは異なる作物を選択している．

表10.　提案アルゴリズムのパターン2での選択作物

$(\Pr(s_i|s_i)=0.1)$

| | 例1 | 例2 | 例3 | 例4 |
|---|---|---|---|---|
| $t$ | 1 | 1 | 2 | 2 |
| $Y_{t-2}$ | $a_3$ | $a_3$ | $a_5$ | $a_5$ |
| $Y_{t-1}$ | $a_5$ | $a_5$ | $a_3$ | $a_3$ |
| $W_{t-1,1}$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ |
| $W_{t-1,2}$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ |
| $W_{t-1,3}$ | $s_2$ | $s_1$ | $s_2$ | $s_2$ |
| $W_{t-1,4}$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ |
| $W_{t-1,5}$ | $s_2$ | $s_2$ | $s_2$ | $s_1$ |
| $W_{t-1,6}$ | $s_2$ | $s_2$ | $s_2$ | $s_2$ |
| $Y_t$ | $a_3$ | $a_1$ | $a_5$ | $a_1$ |

表10の例1，例2は表8の例1と対応（年$t$，2年前の作物$Y_{t-2}$，1年前の作物$Y_{t-1}$が同じ）している．表10の例1では前年のすべての作物の相場状態が低価格の$s_2$なので，次はすべての作物の相場状態が高価格の$s_1$になる確率が高い状況である．この状況では表8の例1と同じ作物$a_3$が選択されている．他方，表10の例2では，作物$a_3$の次の相場状態は価格が低い$s_2$になる確率が高く，作物$a_1$の次の相場状態は価格が高い$s_1$になる確率が高いため，表8の例1とは異なる作物$a_1$が選択されている．なお，表10の例1，例2では2年前の作物$Y_{t-2}$が科$c_2$の輪作年限1年の$a_3$，1年前の作物$Y_{t-1}$が科$c_3$の輪作年限1年の$a_5$なので，連作障害に該当しない候補は$a_1$，$a_2$，$a_3$，$a_4$である．そのため，表10の例2では作物$a_3$の次に価格が高い作物$a_1$を選択している．表10の例2では，前年の相場状態に応じて適応的に単純なローテーションとは異なる科の作物を選択している．同様に，表10の例3，例4は表8の例2と対応している．表10の例3では単純なローテーションの表8の例2と同じ作物を選択し，表10の例4では前年の相場状態に応じて適応的に単純なローテーションの表8の例2とは

異なる科の作物を選択している．

このように提案アルゴリズムでは，価格（相場状態）の変動を考慮して適応的に作物を選択することによって，価格固定の場合よりもより高い収益が期待できる適応的な輪作が可能になる．上記のような提案アルゴリズムによる適応的な輪作の収益と価格固定の場合の輪作の収益の差が大きいのが，今回のシミュレーション例では遷移確率$\Pr(s_i|s_i)$が0.9寄りまたは0.1寄りの場合である．

## 6. 考察と今後の課題

### 6.1 考察

本研究の提案アルゴリズムは価格変動する問題設定に対する最適解（最適な選択作物と総収益の期待値の最大値）を算出するが，当該最適解は同問題設定に対する理論的限界でもある．そのため，提案アルゴリズムを利用して同問題設定に対する他の方法（近似解法や経験則など）の評価も可能である．例えば，5章のシミュレーションは価格変動する問題設定に対する近似解法として価格固定の場合のアルゴリズム（本研究の比較対象）を採用し，理論的限界（本研究の提案アルゴリズムによる最適解）と比較したという解釈もできる．シミュレーション結果より，相場状態が同じ状態に遷移する確率$\Pr(s_i|s_i)$が0.5付近では近似解法が理論的限界を達成していることが確認できる．

本研究は基礎研究の初期検討であり，今後，より現実に近い問題設定に拡張していくことが想定される．拡張研究に関しても最適なアルゴリズムを提案することができれば，提案アルゴリズムそのものが最適な作物選択に利用できるとともに，他の方法（最適な提案アルゴリズムよりも実装などが容易な経験則や近似解法）の評価にも利用可能である．よって，本研究やそのような拡張研究の提案アルゴリズムはスマート農業における意思決定の自動化への貢献が期待される．

### 6.2 今後の課題

本研究では，価格変動を考慮した適応的な輪作アルゴリズムの初期検討を行ったが，相場状態の変動をモデル化したマルコフ連鎖の遷移確率や各相場状態に対応する価格が既知であるという仮定を置いている．現実にはこれらの真の値は未知であり，過去の履歴データなどから何らかの方法で推定する必要がある．例えば，遷移確率については，ベイズ統計学に基づいて履歴データから未知の遷移確率を学習することによる，ベイズ最適なアルゴリズムの提案などが考えられる．今回は連作に該当する作物を選択候補から排除したが，従来研究[6]のように連作障害に該当する場合の少ない収穫量を本研究のモデルに組み込むことも可能である．例えば，連作に該当する場合に確率的に連作障害が発生するモデルなどが考えられる．

本研究では，栽培作物の選択問題を検討対象としたが，農家の収入（収益）増のためには既に作付け（選択）された栽培作物に対する日々の栽培管理（肥料や農薬の散布，間引きなどの農作業の選択）問題も重要である．将来的には，栽培作物の選択問題と栽培管理問題を統合した意思決定問題に関して総収益の最大化を検討すべきである．例えば，栽培管理問題も本研究の栽培作物の選択問題と同様に多段的な意思決定問題として定式化可能なので，統合された問題も同様に多段的な意思決定問題として定式化可能と考える．

## 7. まとめ

本研究では，作物価格の相場状態の変動を考慮したもとで，対象とする有限の数年間の期待総収益を最大化する適応的な輪作アルゴリズムを検討した．具体的には作物の相場状態の変動をマルコフ連鎖でモデル化し，動的計画法によって期待総収益を最大化する適応的な作物の選択を可能にした．提案アルゴリズムの有効性を検証するために，シミュレーションによって価格変動を考慮しない場合（価格固定の場合）の動的計画法による作物選択と提案アルゴリズムによる適応的な作物選択を比較した．その結果，価格変動を考慮した提案アルゴリズムによる収益は価格変動を考慮しない場合の収益と同等かそれ以上であることが確認できた．特に，相場状態の次の状態が高確率で予測可能な状況において提案アルゴリズムの収益がより大きくなる傾向を確認できた．

本研究は基礎研究の初期検討であるため，簡便な問題設定としたが，今後は．6.2節で今後の課題として挙げた拡張研究など，より現実に近い問題設定を検討する．

## 参考文献

[1] 平林健史，清水博幸，進藤卓也，木許雅則，大田健紘：スマート農業に向けた取組み，電子情報通信学会誌，Vol.103，No.6，pp.591-599，2020.

[2] 神成淳司：農業におけるAI活用，第21回人工知能学会全国大会，2E5-9，2007.

[3] 農林水産省：ICT農業の現状とこれから（AI農業を中

心に），2015.
https://www.maff.go.jp/j/shokusan/sosyutu/sosyutu/aisystem/pdf/ict_ai.pdf, 参照（2020.6.2）
[4] 玉木浩二：作物の栽培管理システム（第1報）除草作業のモデル定式化，農業機械学会誌，Vol.34, No.3, pp.262-268, 1972.
[5] 玉木浩二：作物の栽培管理システム（第2報）シミュレーション，農業機械学会誌，Vol.35, No.1, pp.45-51, 1973.
[6] T. Itoh: Innovative Models for Crop Planning Problem to Improve Production Efficiency in Agricultural Management under Uncertainty, Innovation and Supply Chain Management, Vol.8, No.4, pp.169-173, 2014.
[7] T. Itoh, H. Ishii and T. Nanseki: A model of crop planning under uncertainty in agricultural management, International Journal of Production Economics, Vol.81-82, pp.555-558, 2003.
[8] T. Toyonaga, T. Itoh and H. Ishii: A Crop Problem with Fuzzy Random Profit Coefficients, Fuzzy Optimization and Decision Making, Vol.4, pp.51-69, 2005.
[9] 森村英典，高橋幸雄：マルコフ解析，日科技連，東京，1979.
[10] タキイ種苗株式会社：輪作の実践プラン，2018.
https://www.takii.co.jp/flower/bn/pdf/201811_19.pdf，参照（2020.6.2）

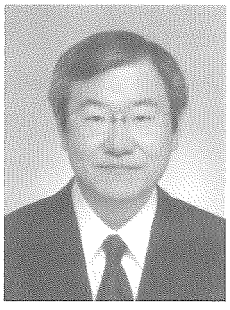
## 付録

パターン1およびパターン2のシミュレーションにはPythonで作成したプログラム，Windows10 Pro 64ビットOS，CPU2.70GHz，メモリ12.0GBの計算機を使用し，シミュレーション全体で約76分を要した．参考例として，パターン1の$\Pr(s_i|s_i) = 0.1$の場合のシミュレーションで使用したソースコードをに示す．

```
import numpy as np
num_A = 6 # 作物数
num_C = 3 # 科の数
n = np.array([1, 2, 1, 2, 1, 2]) # 各作物の輪作年限n
c = np.array([1, 1, 2, 2, 3, 3]) #各作物の科c
b = np.array([500, 400, 450, 350, 480, 380]) # 各作物のコストb
e = np.array([100, 100, 100, 100, 100, 100]) # 各作物の収穫量e
num_S = 2 # 相場状態数num_S
# 相場状態s_kのもとでの作物a_iの価格fm[k][i]
fm = np.array([[12, 13, 13, 15, 11, 13], [8, 9, 9, 11, 7, 9]])
# 相場状態遷移確率（状態s_iから状態s_jへの遷移確率pp[i][j]）
pp = np.array([[0.1, 0.9], [0.9, 0.1]])
# 各作物の固定価格fの計算（マルコフ連鎖の定常確率で期待値）
f = np.zeros(num_A)
teijou = np.zeros(num_S)
teijou[0] = pp[1][0]/(1+pp[1][0]-pp[0][0])
teijou[1] = (1-pp[0][0])/(1+pp[1][0]-pp[0][0])
print("定常確率s1 ", teijou[0]," 定常確率s2 ", teijou[1])
for i in range(num_A):
    f[i] = teijou[0]*fm[0][i]+teijou[1]*fm[1][i]
    print("固定価格f(作物", i+1,")", f[i])
T = 5 # 計画年数T
# 価格固定用
v_pre = np.zeros((num_A, num_A, T))
d_pre = np.zeros((num_A, num_A, T))
check_pre = np.zeros((num_A, num_A, T))
# 価格変動用
vm_pre =
 np.zeros((num_A, num_A, num_S, num_S, num_S, num_S, num_S, num_S, T))
dm_pre =
 np.zeros((num_A, num_A, num_S, num_S, num_S, num_S, num_S, num_S, T))
checkm_pre =
 np.zeros((num_A, num_A, num_S, num_S, num_S, num_S, num_S, num_S, T))

# 価格固定用の動的計画法
def v(c1, c2, t):
    if check_pre[c1-1][c2-1][t-1]==1:
        return  np.array([v_pre[c1-1][c2-1][t-1], d_pre[c1-1][c2-
1][t-1]])
    else:
        if t==T:
            v_temp = np.zeros(num_A)
            for i in range(num_A):
                # 1年前の作物との連作障害チェック
                if c[i]==c[c2-1] and n[c2-1]>=1:
                    v_temp[i] = -1000 # 連作を選択しない設定
                # 2年前の作物との連作障害チェック
                elif c[i]==c[c1-1] and n[c1-1]>=2:
                    v_temp[i] = -1000 # 連作を選択しない設定
                else:
                    v_temp[i] = e[i]*f[i]-b[i]
            sol_v = np.amax(v_temp)
            sol_d = np.argmax(v_temp)+1
        else:
            v_temp = np.zeros(num_A)
            for i in range(num_A):
                # 1年前の作物との連作障害チェック
                if c[i]==c[c2-1] and n[c2-1]>=1:
                    v_temp[i] = -1000 # 連作を選択しない設定
                # 2年前の作物との連作障害チェック
                elif c[i]==c[c1-1] and n[c1-1]>=2:
                    v_temp[i] = -1000 # 連作を選択しない設定
                else:
                    v_temp[i] = e[i]*f[i]-b[i]+v(c2, i+1, t+1)[0]
            sol_v = np.amax(v_temp)
            sol_d = np.argmax(v_temp)+1
        check_pre[c1-1][c2-1][t-1] = 1
        v_pre[c1-1][c2-1][t-1] = sol_v
        d_pre[c1-1][c2-1][t-1] = sol_d
        return np.array([sol_v, sol_d])
print("価格固定の場合の全初期状態について")
for i in range(num_A):
    for j in range(num_A):
        if c[i]!=c[j]: # 連作障害該当の初期状態は排除
            print("v(", i+1, ",", j+1, ", 1) ", v(i+1, j+1, 1))
            Y1 = int(v(i+1, j+1, 1)[1])
            print("v(", j+1, ",", Y1, ", 2) ", v(j+1, Y1, 2))
            Y2 = int(v(j+1, Y1, 2)[1])
            print("v(", Y1, ",", Y2, ", 3) ", v(Y1, Y2, 3))
            Y3 = int(v(Y1, Y2, 3)[1])
            print("v(", Y2, ",", Y3, ", 4) ", v(Y2, Y3, 4))
            Y4 = int(v(Y2, Y3, 4)[1])
            print("v(", Y3, ",", Y4, ", 5) ", v(Y3, Y4, 5))
```

```
            print("")
print("")

# 価格変動用の動的計画法
def vm(c1,c2,s1,s2,s3,s4,s5,s6,t):
    if      checkm_pre[c1-1][c2-1][int(s1)-1][int(s2)-1][int(s3)-
1][int(s4)-1][int(s5)-1][int(s6)-1][t-1]==1:
        return   np.array([vm_pre[c1-1][c2-1][int(s1)-1][int(s2)-
1][int(s3)-1][int(s4)-1][int(s5)-1][int(s6)-1][t-1],dm_pre[c1-
1][c2-1][int(s1)-1][int(s2)-1][int(s3)-1][int(s4)-1][int(s5)-
1][int(s6)-1][t-1]])
    else:
        Wpre = np.zeros(num_A) # 前年の各作物の相場状態
        Wpre[0] = int(s1)
        Wpre[1] = int(s2)
        Wpre[2] = int(s3)
        Wpre[3] = int(s4)
        Wpre[4] = int(s5)
        Wpre[5] = int(s6)
        if t==T:
            v_temp = np.zeros(num_A)
            for i in range(num_A):
                # 1年前の作物との連作障害チェック
                if c[i]==c[c2-1] and n[c2-1]>=1:
                    v_temp[i] = -1000 # 連作を選択しない設定
                # 2年前の作物との連作障害チェック
                elif c[i]==c[c1-1] and n[c1-1]>=2:
                    v_temp[i] = -1000 # 連作を選択しない設定
                else:
                    for j in range(num_S):
                        v_temp[i] +=
                      pp[int(Wpre[i])-1][j]*(e[i]*fm[j][i]-b[i])
            sol_v = np.amax(v_temp)
            sol_d = np.argmax(v_temp)+1
        else:
            v_temp = np.zeros(num_A)
            for i in range(num_A):
                # 1年前の作物との連作障害チェック
                if c[i]==c[c2-1] and n[c2-1]>=1:
                    v_temp[i] = -1000 # 連作を選択しない設定
                # 2年前の作物との連作障害チェック
                elif c[i]==c[c1-1] and n[c1-1]>=2:
                    v_temp[i] = -1000 # 連作を選択しない設定
                else:
                    Wt = np.zeros(num_A) # 次（遷移後）の相場状態
                    for j1 in range(num_S):
                        for j2 in range(num_S):
                            for j3 in range(num_S):
                                for j4 in range(num_S):
                                    for j5 in range(num_S):
                                        for j6 in range(num_S):
                                            Wt[0] = j1
                                            Wt[1] = j2
                                            Wt[2] = j3
                                            Wt[3] = j4
                                            Wt[4] = j5
                                            Wt[5] = j6
                                            p1 = 1.0
                                            for j7 in
                                             range(num_A):
                                                p1 *=
                                pp[int(Wpre[j7])-1][int(Wt[j7])]
                                            v_temp[i] +=
                                        p1*(e[i]*fm[int(Wt[i])][i]-
                        b[i]+vm(c2,i+1,Wt[0]+1,Wt[1]+1,Wt[2]+1,
                                Wt[3]+1,Wt[4]+1,Wt[5]+1,t+1)[0])
            sol_v = np.amax(v_temp)
            sol_d = np.argmax(v_temp)+1
            checkm_pre[c1-1][c2-1][int(s1)-1][int(s2)-1][int(s3)-1]
              [int(s4)-1][int(s5)-1][int(s6)-1][t-1] = 1
            vm_pre[c1-1][c2-1][int(s1)-1][int(s2)-1][int(s3)-1]
              [int(s4)-1][int(s5)-1][int(s6)-1][t-1] = sol_v
            dm_pre[c1-1][c2-1][int(s1)-1][int(s2)-1][int(s3)-1]
              [int(s4)-1][int(s5)-1][int(s6)-1][t-1] = sol_d
            return np.array([sol_v,sol_d])
print("動的な価格の場合の全状態について")
for i1 in range(T):
    for i2 in range(num_A):
        for i3 in range(num_A):
            if c[i2]!=c[i3]: # 連作障害該当の初期状態は排除
                for i4 in range(num_S):
                    for i5 in range(num_S):
                        for i6 in range(num_S):
                            for i7 in range(num_S):
                                for i8 in range(num_S):
                                    for i9 in range(num_S):
                print("vm(",i2+1,",",i3+1,",",i4+1,",",i5+1,",",
                i6+1,",",i7+1,",",i8+1,",",i9+1,",",i1+1,") ",
                vm(i2+1,i3+1,i4+1,i5+1,i6+1,i7+1,i8+1,i9+1,i1+1))
print("")

# 価格変動環境下での比較シミュレーション
# 価格変動vmと価格固定vと行動選択による収益の比較
num_Simu=100000 # 繰返し回数
total_income_simu = 0
total_income_m_simu = 0
for i2 in range(num_Simu):
    # 価格固定vでの行動選択のシミュ
    X = np.zeros(2+num_A)
    X_next = np.zeros(2+num_A)
    check1=0
    while check1==0:
        X[0] = int(np.random.choice(np.arange(num_A),1,
         p=[1/num_A,1/num_A,1/num_A,1/num_A,1/num_A,1/num_A]))+1
        X[1] = int(np.random.choice(np.arange(num_A),1,
         p=[1/num_A,1/num_A,1/num_A,1/num_A,1/num_A,1/num_A]))+1
        if c[int(X[0])-1]!=c[int(X[1])-1]:
            check1=1
    for i3 in range(num_A):
        X[i3+2] = int(np.random.choice(np.arange(num_S),1,
         p=[teijou[0],1-teijou[0]]))+1
    for i3 in range(T):
        Y = int(v(int(X[0]),int(X[1]),i3+1)[1])
        for i4 in range(num_A):
            X_next[i4+2] = int(np.random.choice(np.arange(num_S),
             1,p=pp[int(X[i4+2])-1]))+1
        total_income_simu +=
         e[Y-1]*fm[int(X_next[Y-1+2])-1][Y-1]-b[Y-1]
        X[0] = X[1]
        X[1] = Y
        for i4 in range(num_A):
            X[i4+2] = X_next[i4+2]
    # 価格変動vmでの行動選択のシミュ
    Xm = np.zeros(2+num_A)
    Xm_next = np.zeros(2+num_A)
    check1=0
    while check1==0:
        Xm[0] = int(np.random.choice(np.arange(num_A),1,
         p=[1/num_A,1/num_A,1/num_A,1/num_A,1/num_A,1/num_A]))+1
        Xm[1] = int(np.random.choice(np.arange(num_A),1,
         p=[1/num_A,1/num_A,1/num_A,1/num_A,1/num_A,1/num_A]))+1
        if c[int(Xm[0])-1]!=c[int(Xm[1])-1]:
            check1=1
    for i3 in range(num_A):
        Xm[i3+2] = int(np.random.choice(np.arange(num_S),1,
         p=[teijou[0],1-teijou[0]]))+1
```

```
    for i3 in range(T):
        Y = int(vm(int(Xm[0]), int(Xm[1]), int(Xm[2]), int(Xm[3]),
          int(Xm[4]), int(Xm[5]), int(Xm[6]), int(Xm[7]), i3+1)[1])
        for i4 in range(num_A):
            Xm_next[i4+2] = int(np.random.choice(np.arange(num_S)
              , 1, p=pp[int(Xm[i4+2])-1]))+1
        total_income_m_simu +=
          e[Y-1]*fm[int(Xm_next[Y-1+2])-1][Y-1]-b[Y-1]
        Xm[0] = Xm[1]
        Xm[1] = Y
        for i4 in range(num_A):
            Xm[i4+2] = Xm_next[i4+2]
print("シミュレーション結果")
print(num_Simu,"試行での価格変動vmの平均総収益 ",
  total_income_m_simu/num_Simu)
print(num_Simu,"試行での価格固定vの平均総収益 ",
  total_income_simu/num_Simu)
```



**前田康成（まえだやすなり）**

平成7年早大・理工卒．平成9年同大学院理工学研究科修士課程修了．日本電信電話（株），東日本電信電話（株），北見工大助手，助教，准教授を経て平成28年同大学教授，現在に至る．博士（工学）．統計的決定理論の学習問題への応用に関する研究に従事．電子情報通信学会等各会員．