



〈書評〉Eric V. Denardo, Dynamic Programming : Models and Applications

末廣, 英生

(Citation)

国民経済雑誌, 148(1):105-108

(Issue Date)

1983-07

(Resource Type)

departmental bulletin paper

(Version)

Version of Record

(JaLCD0I)

<https://doi.org/10.24546/00172827>

(URL)

<https://hdl.handle.net/20.500.14094/00172827>



Eric V. Denardo, *Dynamic Programming:*

Models and Applications,

Prentice-Hall, Englewood Cliffs, 1982, xii+227 pp.

末 廣 英 生

I

多段決定過程問題を解く手法として、Dynamic Programming (以下 DP) は、関数方程式

$$(*) \cdots f_N(s) = \max_P [R_N(S, P) + f_{N-1}(\hat{S}_N(S, P))]$$

を用いる点に特徴がある。ここに、 $f_N(s)$ = 状態 S から出発し最適政策を用いた時の N 期間の利得合計、 $R_N(S, P)$ = 状態 S の下で決定 P を行なった時の今期の利得、 $\hat{S}_N(S, P)$ = 状態 S の下で決定 P を行なった時の状態の変換、である。(*) は DP のメリットの 2 つの側面を同時に表わしている。第 1 は、多段決定過程の理論化の側面である。決定者は、決定の各段階において、来期以後最適に行動する (すなわち、 $f_{N-1}(\hat{S}_N(S, P))$ を得る) ことを前提として今期の行動を最適化すれば、全体が最適になる (= 理論としての再帰固定)。第 2 は、最適解の計算手法としての側面である。あらかじめ $f_{N-1}(\hat{S}_N(S, P))$ を計算しておくことによって、再帰的に $f_N(s)$ が計算できる (= 計算手法としての再帰固定法)。

しかし、DP には計算手法として欠点がある。すなわち、(*) の $f_N(s)$ の関数形を陽表的に求めうるのは少数のケースに限られる (Aoki [1])。したがって、多くの場合、数値計算に頼らねばならない。その際、(*) の $f_N(s)$ を知るためには、すべての可能な $\hat{S}_N(S, P)$ に対して f_{N-1} を計算しておく必要がある。しかし、この作業はしばしば容易にコンピューターの計算・記憶容量を超える (Dreyfus [2])。

II

本書は、計算手法の側面を中心とした DP の解説書と言える。特に、実際の解計算における計算効率化に関して、著者自身を含むこれまでの研究成果が丁寧に解説されている。全体は、sequential decision process への導入的解説の第 1 章、確実性下の決定を

扱う第2～5章、不確実性下の決定を扱う第6～8章、効率的データ処理および凸関数の性質を解説する2つの付録から成る。本論の概要は次の通りである。

第2章。有限・非循環的ネットワークの最短経路問題が取り上げられる。これは、次の2つの意味で、sequential decision process のプロトタイプとすることができる。第1に、この問題の heuristic な定式化を通じて、DP の基本的概念 i) 埋め込み (embedding), ii) 関数方程式 (functional equation), iii) 再帰固定 (recursive fixing), iv) 最適性の原理 (principle of optimality), が導入される。第2に、実際に解を計算するための基本的な数値計算手法 i) 再帰固定法 (recursive fixing), ii) 逐次到達法 (reaching), が示される。そして、逐次到達法のメリットとして、次の2点が主張される。第1に、ネットワークの特殊性や最適政策の特徴について事前情報がある場合等は、逐次到達法の方が計算効率が高い。第2に、ネットワークが循環を持つ場合には、再帰固定法では解が得られないが、逐次到達法では解くことができる。その際、計算の高速化の諸技法 i) 配列・リスト法 (heaps and lists), ii) ラベル修正法 (label correction), iii) 一括法 (buckets), が利用できる。

第3章。ネットワーク問題の応用として資源配分問題が取り上げられ、この応用手続きとして DP における状態概念 (state) が解説される。解計算は、再帰固定法・逐次到達法のいずれでも可能である。しかし、逐次到達法の場合、計算の高速化が可能である。資源の種類が増えるにしたがい、計算量が飛躍的に増大するという困難に対しては、ラグランジュ乗数法による次数低下処理法が紹介される。最適ラグランジュ乗数ベクトルの計算手法として i) 2等分法 (bisection), ii) 傾斜改定法 (updating the slope), iii) Linear Programming 傾斜改定法, がある。また、目的関数が凹、あるいは最小凹優関数 (least concave majorant) で近似できる場合には、限界分析による逐次交換法 (trading step) が、再帰固定法による DP 解法より効率的である。

第4章。決定集合が非可算の場合を数値計算による DP 解法にのせる手法として、格子点法 (grids) が解説される。計算の高速化のためには、補間法 (interpolation) を用いた再帰固定法と逐次到達法の組み合わせが望ましい。例として化学反応装置の最適制御問題が検討される。この例は段階 (stage) を持つ sequential decision process であって、DP における段階概念の役割が解説される。

第5章。古典的生産平滑化問題の DP 解法が取り上げられる。この問題は、段階を持つ DP 問題として、直接的に解くことができる。しかし、ネットワークフロー問題の特殊ケースとみなすことによって、より効率的に解ける場合がある。すなわち、規模の経済が存在する場合には、循環フローを持たないネットワークフロー問題として、計算のより容易な DP 問題に再定式化できる。他方、規模の不経済が存在する場合には、ある

種の限界原理によってより効率的に解が得られる。この場合には、パラメーターの変化に対する最適解の摂動原理が美しい形で得られる。

第6章以降、マルコフ決定問題が扱われる。まず、マルコフ決定モデルのDPによる定式化と確率モデルにおける政策・政策空間 (policy・policy space) 概念が解説される。そして、状態空間・決定集合の有限性の下で、最適政策 (optimal policy) の存在定理が証明される。具体例として、有限期間在庫モデルが検討される。マルコフ決定モデルの場合、逐次到達法では解を得ることはできないが、再帰固定法では解くことができる。しかし、この例のような典型的な多段階決定問題でも、段階概念を用いないDP解法の方が計算効率が高い。他方、古典的な解析的分析も行なわれ、単一水準政策 (order-up-to level) が導かれる。また、確率分布の型として、ポアソン過程が紹介されている。

第7章。第6章の例が固定的注文費用のケースに拡張され、古典的な解析的分析が紹介される。適当な十分条件の下で、いわゆる (s-S) 政策が最適政策となる。モデルは種々のバリエーションを包摂するほどに十分一般的であるが、費用関数の凸・擬凸性が中心的な役割を果たす。

第8章。定常性、有限性、時間割引き、線型効用関数で特徴づけられる離散型無限期間マルコフ決定モデルが取り扱われる。DPの関数方程式 ((*) で $N = \infty$) を満足する最適政策が存在する。しかし、実際に解を計算するためには、関数空間上で最適化を含む方程式を解かねばならないので、第7章までの手法は適用できない。数値解法として、政策反復法 (policy iteration) と Linear Programming 法が紹介され、両者が本質的に同一のアルゴリズムであることが示される。近似解で満足できる場合、逐次近似法 (successive approximation) がある。この方法は状態変数の数が多い場合に有力であり、逐次計算の3つの高速化手法が挙げられている。最後に、有限期間問題・無限期間問題のそれぞれの最適政策の間に一種のターンパイク定理 (Dorfman, Samuelson and Solow [3]) が成り立つことが証明されている。

III

本書の特徴は2点ある。第1に、DPによる定式化手続きにおいて、段階概念 (stage model) よりも状態概念 (state model) を重視する。したがって、(*)で N が段階を表わす場合でも、(N, S) を新たな状態ベクトルとみて

$$(**) \dots f(N, S) = \max_P [R(N, S, P) + f(\hat{N}(N, S, P), \hat{S}(N, S, P))]$$

なる関数方程式を解く。このことのメリットは2つある。1) 解に関する事前情報がある場合、(**)の右辺の最適化が容易になる。2) 組み合わせ的問題においては、初めから段階概念が適用できないことが多い。

第2に、DPにおける数値計算手続きにおいて、逐次到達法が提唱される。つまり、DPの基本概念である再帰固定の考え方が、実際の解計算には用いられない。そのかわりに、(**)の $f(N, S)$ の暫定値が、状態空間上で繰り返し計算されることになる。このことのメリットは2つある。1)ネットワーク構造や解の特性に関する事前情報がある場合、計算が高速化できる。2)循環を持つネットワーク問題は、再帰固定法では解くことができない。ただし、逐次到達法の適用は、現実性モデルに限定される。

以上の2つの特徴は、多段決定問題におけるDPの理論化の側面と計算手法としての側面を分離し、後者の効率性を追求したものである。実際に複雑なシステムを解析する者にとって、このことは不可欠でありかつ有用なのである。この点からすれば、本書はDPの1つの健全な発展の軌跡を示していると言える。しかし、上記の2特徴における計算効率化の考え方は、次のようなものである。すなわち、(*)では、結局、状態空間上のすべての点に対して $f_N(s)$ を計算してしまう。他方、(**)では、事前情報に基づいて、不必要な状態に対する $f(N, S)$ の計算をスキップできる、ということである。この際、再帰固定法によるstage modelの解計算(*)から逐次到達法によるstate modelの解計算(**)への移行によって、多段決定過程の理論化としての側面が、DPから失われる。しかるに、(**)において、(*)の再帰固定の考え方に代わりうる理論が提示されているわけではない。したがって、多段決定過程の解析的分析にとって、再帰固定の考え方は依然重要である。この点からすれば、本書が示す方向とは別に、多段決定過程の理論化の側面が、他の動的最適化手法（たとえば、変分法、最大値原理）との比較において、一層明らかにされねばならない（伊賀 [4]）。

参 考 文 献

- [1] M. Aoki, 'Dynamic Resource Allocation Problem' mimeo, 1982.
- [2] S. E. Dreyfus, 'Computational Aspects of Dynamic Programming' Operations Research, June 1957.
- [3] R. Dorfman, P. A. Samuelson and R. M. Solow, *Linear Programming and Economic Analysis*, McGraw-Hill, 1958.
- [4] 伊賀隆「ダイナミック・プログラミングとその応用」国民経済雑誌, 1973年1月。