Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической экономики

наименование кафедры

Разработка программного продукта для решения задач

динамического программирования

АВТОРЕФЕРАТ

студентки 3 курса 381 группы

направление 09.04.03 – Прикладная информатика

механико-математического факультета

Прахт Марии Николаевны

фамилия, имя, отчество

Научный руководитель

Зав.кафедрой, доктор физ.-мат. наук ________________ С. П. Сидоров

должность, уч.степень, уч.звание подпись, дата инициалы, фамилия

Зав.кафедрой

профессор, доктор физ.-мат. наук ________________ С. И. Дудов

должность, уч.степень, уч.звание подпись, дата инициалы, фамилия

Саратов 2016

Введение

Фундаментальным инструментом для решения динамических задач экономики является динамическое программирование.

Своё развитие динамическое программирование получило в середине 20-ого века благодаря работам Р. Беллмана, Ф. Бриоши. И актуально по сей день, поскольку позволяет решать сложные задачи путем сведения их к ряду более простых подзадач.

Динамическое программирование применяется не только для решения задач экомоники, но и для решения других прикладных задач.

Модели динамического программирования могут применяться:

- при разработке правил управления запасами, устанавливающими момент пополнения запасов и размер пополняющего заказа;
- при разработке принципов календарного планирования производства и выравнивания занятости в условиях колеблющегося спроса на продукцию;
- при распределении инвестиций между новыми направлениями их использования;
- при составлении календарных планов текущего и капитального ремонта сложного оборудования и его замены;
- при разработке долгосрочных правил замены выбывающих из эксплуатации основных фондов и т.д.

Задачи динамического программирования могут реализовываться на различных языках программирования, например, Pascal, LISP, Mat Lab и других.

Цель работы:

1) Изучить методы решения задач динамического программирования.
2) Применить метод динамического программирования к решению экономических задач.

3) Разработать программный продукт для решения задач динамического программирования с использованием Mat Lab.

В разделе 1 данной работы приводится определение динамического программирования и два основных подхода к решению задач. Также в этом разделе рассмотрен эвристический вывод уравнения Беллмана и инструмент для проверки существования и единственности решения этого уравнения.

Раздел 2 посвящен детерминистическому динамическому программированию. В этом разделе приведено несколько алгоритмов для функции ценности.

В разделе 3 представлен метод формосохраняющего динамического программирования для решения задач принятия решений с дискретным временем и непрерывными состояниями.

В разделе 4 рассказывается о стохастическом динамическом программировании.

Содержание работы

В главе 1 вводится понятие динамического программирования и уравнения Беллмана, как основного инструмента динамического программирования.

Динамическое программирование - метод решения задач путем составления последовательности из подзадач таким образом, что:

1. первый элемент последовательности (возможно несколько элементов) имеет тривиальное решение,
2. последний элемент этой последовательности - это исходная задача,
3. каждая задача этой последовательности может быть решена с использованием решения подзадач с меньшими номерами.

Доказательство работоспособности метода динамического программирования напрямую следует из принципа математической индукции [3].

Вывод уравнения Беллмана. Рассмотрим случай, когда агент, который должен принять решение о выборе набора управляющих переменных $\{y_t\}_{t=1}^{\infty}$в целях максимизации дисконтированной суммы будущих выплат $u(y_t;x_t)$, где x_t является переменным состоянием, которое развивается по следующему закону:

$$x_{t+1} = h(x_t; y_t),$$

где x_0 задан.

Оптимальное значение наш агент может получить путем максимизации будущих выплат:

$$V(x_t) = \max_{\{y_{t+s} \in D(x_{t+s})\}_{s=0}^{\infty}} \sum_{s=0}^{\infty} \beta^s u(y_{t+s}; x_{t+s}).$$

Путем преобразований получаем уравнение Беллмана:

$$V(x_t) = \max_{\{y_t \in D(x_t)\}} u(y_t; x_t) + \beta V(x_{t+1})$$

Проблема состоит в поиске $V(x_t)$. Выполнить данный поиск поможет простая процедура:

1. Делаем начальное предположение о виде $V_0(x_t)$,
2. Согласно уравнению Беллмана строим $V_{i+1}(x_t)$:
$$V_{i+1}(x_t) = \max_{\{y_t \in D(x_t)\}} u(y_t; x_t) + \beta V_i(x_{t+1}),$$
3. Если $V_{i+1}(x_t) = V_i(x_t)$, то неподвижная точка найдена и проблема решена. Если нет, то возвращаемся к пункту 2.

Другими словами, решение уравнения Беллмана составляет либо нахождение неподвижной точки уравнения Беллмана, либо нахождение неподвижной точки оператора T, такого что:
$$V_{i+1} = TV_i,$$
где через Т обозначен перечень операций в уравнении Беллмана. Следующая проблема заключается в существовании и единственности этой неподвижной точки (решения).

Существование и единственность решения.

Теорема 1. (Теорема о сжимающем операторе). Если (S, ρ) – метрическое пространство и оператор $T: S \to S$ – оператор сжатия с коэффициентом $\beta \in (0; 1)$, то:

1. Т имеет ровно одну неподвижную точку $V \in S$ такую что $V = TV$,
2. Для любого $V \in S, \rho(T^n V_0; V) < \beta^n \rho(V_0, V)$ для $n = 0,\ 1, 2 \dots$

Теорема 2. (Достаточное условие Блеквелла). Пусть $X \subseteq R^l$ и $B(X)$ является множеством, порождаемым отображением $V: X \to R$ с универсальной метрикой. Пусть оператор $T: B(X) \to B(X)$ удовлетворяет следующим условиям:

1) Пусть $V, W \in B(X)$, если $V(x) \leq W(x)$, то для любого $x \in X$
$$TV(x) \leq TW(x)$$
2) Существует константа $\beta \in (0; 1)$ такая, что для любого $V \in B(X)$ и $a \geq 0$ имеем:
$$T(V + a) \leq TV + \beta a.$$

Тогда T является оператором сжатия с коэффициентом β.

В главе 2 рассматриваются алгоритмы для детерминистического случая.

Простой итерационный алгоритм для функции ценности:

1. Выбираем некоторое подмножество χ допустимых значений переменной x

   $$\chi = \{x_1, x_2, \dots, x_N\}$$

   Делаем предположение о виде функции ценности $V_0(x)$ и выбираем параметр $\varepsilon > 0$
2. Для каждого $x_l \in \chi, l = 1, \dots, N$ вычислим:

   $$V_{i+1}(x_l) = \max_{x' \in \chi} u(y(x_l; x'); x_l) + \beta V_i(x')$$

3. Если $\|V_{i+1}(x) - V_i(x)\| < \varepsilon$, переходим к следующему шагу. Иначе возвращаемся к шагу 2.
4. Вычисляем финальное решение как:

   $$y^*(x) = y(x, x')$$

и

$$V^*(x) = \frac{u(y^*(x), x)}{1 - \beta}.$$

Метод интерполяции:

1. Выбираем сетку χ из множества допустимых значений переменной состояний x

   $$\chi = \{x_1, \dots, x_N\}$$

   Выбираем сетку Υ из множества допустимых значений для переменных состояний y

   $$\Upsilon = \{y_1, \dots, y_M\} \text{ с } M \geq N$$

   Выбираем начальное предположение о виде функции ценности $V_0(x)$ и выберем критерий остановки $\varepsilon > 0$.
2. Для каждого $x_l \in \chi, l = 1, \dots, N$ вычислим

$$x'_{lj} = h(y_j, x_l)\ \forall j = 1, \dots, M$$

Вычислим значение интерполированной функции ценности для каждого $x'_{lj} = h(y_j, x_l)$: $\tilde{V}_i(x'_{lj})$

$$V_{i+1}(x_l) = \max_{\{y \in Y\}} u(y, x_l) + \beta \tilde{V}_i(x'_{lj})$$

3. Если $\|V_{i+1}(x) - V_i(x)\| < \varepsilon$, то переходим к следующему шагу, иначе возвращаемся к пункту 2.
4. Вычислим конечное решение

$$V^*(x) = \frac{u(y^*, x)}{1 - \beta}$$

Итерация по стратегиям. Метод Говарда:

1. Установим начальное возможное значение управляющей переменной $y = f_0(x)$ и вычислим значение функции ценности, связанное с этим предположением, предполагая, что это правило применяется на всех шагах принятия решений:

$$V(x_t) = \sum_{s=0}^{\infty} \beta^s u(f_i(x_{i+s}), x_{t+s})$$

При условии, что $x_{t+1} = h(x_t, y_t) = h(x_t, f_i(x_t))$ при $i = 0$. Критерий остановки $\varepsilon > 0$.

2. Найдем новое стратегическое правило $y = f_{i+1}(x)$ такое что

$$f_{i+1}(x) \in \operatorname*{Argmax}_{y} u(y, x) + \beta V(x')$$

с $x' = h(x, f_i(x))$.

3. Если $\|f_{i+1}(x) - f_i(x)\| < \varepsilon$, то останавливаем выполнение алгоритма, иначе возвращаемся к пункту 2.

В главе 3 рассмотрены формосохраняющие алгоритмы.

- Шаг 1 (Оптимизация). Вычисляем для каждого $1 \le i \le m_t$

$$v_i = \max_{y_i \in D(x_i, t)} \left(u_t(x_i, y_i) + \beta \tilde{V}_{t+1}(x_{i'}; \mathbf{c}^{t+1}) \right), \text{ где } x_{i'} = h(x_i, y_i).$$

- Шаг 2 (Аппроксимация). Используем некоторый метод приближения для вычисления $\mathbf{c}^t$ такого, чтобы $\widetilde{V}(x;\mathbf{c}^t)$ аппроксимировала данные $\{(x_i, v_i\}$.

В главе 4 рассмотрены алгоритмы нахождения решения для стохастическог случая.

Простой итерационный алгоритм для стохастической задачи:

1. Выбираем сетку χ допустимых значений для переменных состояний x

$$\chi = \{x_1, \dots, x_N\}$$

и шоков, s

$$S = \{s_1, \dots, s_M\}$$

вместе с матрицей переходов $\Pi = (\pi_{ij})$

Выбираем начальное предположение о виде функции ценности $V_0(x)$ и выберем критерий остановки $\varepsilon > 0$.

2. Для каждого $x_l \in \chi, l = 1, \dots, N,$ и $s_k \in S, k = 1, \dots, M$ считаем

$$V_{i+1}(x_l, s_k) = \max_{\{x' \in \chi\}} u(y(x_l, s_k, x'), x_l, s_k) + \beta \sum_{j=1}^{M} \pi_{ij} V_i(x', s'_j)$$

3. Если $\|V_{i+1}(x,s) - V_i(x,s)\| < \varepsilon$, то переходим к следующему шагу, иначе возвращаемся к пункту 2.
4. Вычислим конечное решение:

$$y^*(x,s) = y(x, x'(s,s), s)$$

Итерация по стратегиям:

1. Установим начальное допустимое множество решающих правил для управляющей переменной $y = f_0(x, s_k), k = 1, \dots, M$ и рассчитаем значение, полагая, что $x_{t+1} = h(x_t, y_t, s_t) = h(x_t, f_i(x_t, s_t), s_t)$ с $i = 0$. Установим критерий остановки $\varepsilon > 0$.
2. Найдем новое правило $y = f_{i+1}(x, s_k), k = 1, \dots, M$, такой что

$$f_{i+1}(x, s_k) \in \operatorname*{Argmax}_y u(y, x, s_k) + \beta \sum_{j=1}^{M} \pi_{ij} V\left(x', s'_j\right)$$

с $x' = h(x, f_i(x, s_k), s_k)$.

3. Проверим выполнение условия $\|f_{i+1}(x, s) - f_i(x, s)\| < \varepsilon$. Если условие выполняется, то останавливаемся, если нет, то возвращаемся к пункту 2.

Заключение

Таким образом, в работе рассмотрены методы решения задач динамического программирования и разработаны программные продукты для решения задачи на приближении функции ценности с дискретным временем и непрерывными состояниями.

В разделе 1 данной работы был рассмотрен эвристический вывод уравнения Беллмана, и были доказаны теоремы о сжимающем операторе, которые являются «хорошим» инструментом для проверки (или доказательства) существования и единственности решения уравнения Беллмана.

В разделе 2 было приведено несколько алгоритмов приближения функции ценности для детерминистического ДП: простой итерационный метод, метод интерполяции, метод итерации по стратегиям, метод Говарда. Каждый из этих алгоритмов был реализован в программных продуктах, код создания которых размещен в приложениях A, B, C.

В разделе 3 представлены алгоритмы формосохраняющего динамического программирования для решения задач принятия решений с дискретным временем и непрерывными состояниями на основе применения двух формосохраняющих методов приближения. Алгоритмы решения в данном разделе рассматриваются на примере задачи оптимально роста.

В разделе 4 рассматривается дискретизация шоков (квадратурный подход) и несколько алгоритмов приближения функции ценности для непрерывных состояний: простой итерационный алгоритм для стохастической задачи и итерация по стратегиям. Каждый из этих алгоритмов был реализован в программных продуктах, код создания которых размещен в приложениях D, F, G.

В настоящее время вопросы, связанные с применением динамического программирования в различных областях экономики и менеджмента, разрабатываются многими отечественными и зарубежными исследователями, математиками и экономистами. Разработан ряд методов и моделей, предназначенных для предприятий и различного характера. Это говорит о том, что этот метод является востребованным и необходимым для решения многих управленческих задач.

Список использованных источников

1. Бойцов Д. И., Сидоров, С. П. Алгоритмы формосохраняющего динамического программирования для решения задачи оптимального роста / Д. И. Бойцов, С. П. Сидоров // Математическое моделирование в экономике и управлении риска: материалы III Междунар. Молодежной науч.-практ. конф. Саратов: Изд-во Сарат. ун-та, 2014. С. 44-49.
2. Беллман, Р. Динамическое программирование / Р. Беллман. М.: Изд-во иностранной литературы, 1960.
3. Акулич, И. Л. Математическое программирование в примерах и задачах: Учеб. пособие для студентов эконом. спец. вузов / И. Л. Акулич. М.: Высшая школа, 1986. 319 с.
4. Ануфриев, И. Е. Самоучитель MatLab 5.3/6.x. / И. Е. Ануфриев. СПб.: БХВ-Петербург, 2002. 736 с.
5. Потемкин, В. Г. Система MatLab. Справочное пособие / В. Г. Потемкин. М.: ДИАЛОГ–МИФИ, 1997. 350 с.
6. Дьяконов, В. П. MATLAB. Полный самоучитель / В. П. Дьяконов. М.: ДМК Пресс, 2012. 768 с.
7. Мартынов, Н. Н. Matlab 7. Элементарное введение / Н. Н. Мартынов. М.: «Кудиц – Образ», 2005г. 416с.
8. Курбатова, Е. А. MATLAB 7. Самоучитель / Е. А. Курбатова. И: Вильямс, 2005, 256с.
9. Tauchen, G. Quadrature Based Methods for Obtaining Approximate Solutions to Nonlinear Asset Pricing Models / G. Tauchen, R. Hussey. Econometrica, 1991. 371-396 p.
10. Bertsekas, D. Dynamic Programming and Stochastic Control / D. Bertsekas. New York: Academic Press, 1976.
11. Judd, K.L. Numerical methods in economics / K. L. Judd. Cambridge, Massachussets: MIT Press, 1998.

12. Judd, K., Solnick, A. Numerical Dynamic Programming with Shape-preserving Splines / K. Judd, A. Solnick. Manuscript, Hoover Institution 1994.
13. Lucas, R., Stokey N., Prescott, E. Recursive Methods in Economic Dynamics / R. Lucas, N. Stokey, E. Prescott. Cambridge (MA): Harvard University Press, 1989.
14. Cai, Y., Judd, K. L. Stable and efficient computational methods for dynamic programming // J. Eur. Econ. Assoc. 2010. P. 626-634.
15. Cai, Y., Judd, K. L. Shape-preserving dynamic programming // Math. Meth. Oper. Res. 2013. P. 407-421.
16. Cai, Y. Dynamic programming and its application in economics and finance / Y. Cai. PhD thesis, Stanford University. 2009.
17. Schumaker, L. On shape-preserving quadratic spline interpolation // SIAM J. Numer. Anal. 1983. P. 854-864.
18. Wang, S.P., Judd, K.L. Solving a savings allocation problem by numerical dynamic programming with shape-preserving interpolation // Comput. Oper. Res. 2000. P. 399-408.
19. Коровкин, П. П. О порядке приближения функций линейными положительными операторами // ДАН СССР. 1957. С. 1158-1161.
20. Popoviciu, T. About the Best Polynomial Approximation of Continuous Functions / T. Popoviciu. Mathematical Monography. Sect. Mat. Univ. Cluj, 1937.
21. Pál J. Approksimation of konvekse Funktioner ved konvekse Polynomier // Mat. Tidsskrift. 1925. P. 60-65.