УДК 004.0- 519.854.001

Рекурсивные реализации задач динамического программирования

А.И. Андрюхин, к.т.н, В.А. Полетаев, студент.
Донецкий национальный технический университет, г. Донецк,
alexandruckin@ramber.ru

Андрюхин А.И. Полетаев В.А. Рекурсивные реализации задач динамического программирования. *В работе рассмотрены рекурсивные реализации известных задач, решаемых с помощью метода динамического программирования. Представлены задача оптимального управления, задача оптимального распределения ресурсов, задача оптимальной замены оборудования. Приведен общий алгоритм решения задач динамического программирования. Указаны достоинства и недостатки применения принципа Беллмана и рекурсии в программных реализациях. Приведены результаты компьютерных расчетов. В расчетах использовался пакет Mathematica.*

Ключевые слова*: динамическое программирование, принцип Беллмана, рекурсия.*

Введение

Исследование операций занимается изучением экстремальных задач управления, планированием и разработкой методов их решения. Динамическое программирование, как один из разделов исследования операций, предполагает , что процесс принятия решения и управления может быть разбит на отдельные части или этапы, т.е. является многоэтапным. Одним из основных методов динамического программирования (МДП) является метод функциональных уравнений Р.Беллмана [1–5], который базируется на использовании его же принципа оптимальности.

Рекурсия и ее применение

Ответ на вопрос о нужности использования рекурсии в программировании непрост: с одной стороны, обычно стараются не использовать рекурсию в программах вследствие понижения эффективности и стараются ее искоренять[6]. С другой стороны, если исследуемая проблема является по своей сути рекурсивной, то программная реализация может быть структурно проще и нагляднее. Также можно добавить, что окружающий нас мир является примером глобального использования рекурсии согласно работам[7-8] и др., а рекурсия в современных функциональных языках типа Haskell является единственным способом организации цикла.

Рекурсия применяется там, где она естественно требуется вследствие рекурсивности самой задачи. При этом скорость работы программы имеет малое значение и не нужно решать проблему параллельного функционирования программных модулей реализуемой системы.

Примеры реализации функции Аккермана в [9] показательны в этом плане.

Принцип оптимальности Беллмана

Известный принцип Беллмана состоит в том, что, при любом начальном состоянии на любом шаге и любом управлении, выбранном на этом шаге, последующие управляющие воздействия должны выбираться оптимальными относительно состояния, в котором будет система в конце данного шага, т.е. не они не зависят от предыстории.

Более детально, рассмотрим рис.1, на котором представлена траектория движения управляемого объекта в пространстве переменных состояний (x_1,..,x_i..,x_n). Точка $\boldsymbol{x_S}$ делит оптимальную траекторию $\boldsymbol{x_0, x_T}$ на две части, которые обозначены 1 и 2 на рис.1. Принцип Беллмана гласит, что если вся траектория $\boldsymbol{x_0, x_T}$ оптимальна, то оптимален и участок 2.

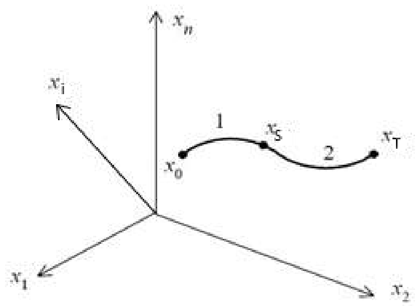

Рисунок 1-Траектория оптимального движения объекта управления

Применение принципа Беллмана зачастую показывает, что выбранное на данном шаге управление не является локально лучшим, но будет оптимальным по глобальному критерию всего многоэтапного процесса управления. Непрерывный управляемый процесс можно рассматривать как дискретный, условно разбив его на временные отрезки. К сожалению, нет формального метода разбиения всего процесса на этапы и здесь все зависит от умений исследователя.

Алгоритм решения задач МДП

Характерным признаком МДП является небольшое количество подзадач, которые решаются многократно.

Ввиду применения принципа Беллмана следующим отличительным свойством решения задач МДП является оптимальность для подзадач.

Общий подход построения алгоритмов решения задач МДП определим следующими пунктами:

1. Четкая формулировка проблемы.
2. Выполнить разбиение оптимизируемого процесса на этапы (это искусство, а не наука, что указывалось ранее).
3. Написать рекуррентное соотношение, т.е определить соотношение между глобальным критерием оптимальности всего процесса или задачи и локальными критериями оптимальности этапов или подзадач. В этом пункте необходимо также обеспечить возможность получения характеристик найденного глобального оптимального решения. В чисто практическом плане при программной рекурсивной реализации необходимо явно указывать условие выхода из рекурсии.
4. Найти оптимальное глобальное решение для проблемной задачи или оптимизируемого процесса. При рекурсивном решении используем стратегию решения сверху вниз т.е. берем глобальную задачу, в которой решаем необходимые подзадачи. В противном случае используем стратегию снизу вверх, т.е. решаем сначала элементарные подзадачи, потом на следующей итерации подзадачи, которые требуют результатов решенных подзадач на предыдущей итерации и т.д., пока не будет решена глобальная проблема. Обычно последний подход является более быстродействующим .
5. Решить задачу обратного хода, т.е. получить не только значения критерия оптимальности решения, но и найти само решение,

Задача оптимального управления

Рассмотрим подход к оптимальному управлению дискретной системы согласно [1,10]. Пусть дискретная система описывается с помощью рекуррентных уравнений

$$x(k+1) = f^{k+1}\left[x(k), u(k)\right]$$

где $k = 0, 1, \ldots, N-1$; $x(0)$ задано.

Целью управления является минимизация функционала

$$J = \sum_{k=0}^{N-1} L\left[x(k), u(k)\right],$$

где L – скалярная функция.

Минимальное значение функционала J зависит от начального состояния $x(0)$ (оно задается, т.е. неуправляемо). Обозначим этот минимум через $j_N[x(0)]$.

Справедливы следующие соотношения:

$$j_N[x(0)] = \min_{u(0)} \min_{u(1)} \ldots \min_{u(N-1)} \{L[x(0), u(0)] +$$
$$L[x(1), u(1)] + \ldots + L[x(N-1), u(N-1)]\} =$$
$$= \min_{u(0)} \{L[x(0), u(0)]\} + \min_{u(1)} \ldots \min_{u(N-1)} \{L[x(1), u(1)] + \ldots$$
$$+ L[x(N-1), u(N-1)]\} =$$
$$= \min_{u(0)} \{L[x(0), u(0)] + j_{N-2}[x(1)]\}$$

Здесь $j_{N-1}[x(1)]$ – минимальное значение критерия качества для процесса длительностью в $N-1$ шагов и имеющее начальное состояние $x(1)$:

$$j_{N-1}[x(1)] = \min_{u(1)} \{L[x(1), u(1)] + j_{N-q-1}[x(2)]\}.$$

Для процесса управления с $N-q$ шагами, имеющего в качестве начального состояния $x(q+1)$ $(1 \le q \le N-1)$:

$$j_{N-1}[x(q)] = \min_{u(q)} \{L[x(q), u(q)] + j_{N-q-1}[x(q+1)]\}$$

Последнее соотношение является частным вариантом уравнения Беллмана.

Применим эти базовые соотношения к решению стандартной непрерывной задачи.

Пример расчета для задачи оптимального управления.

Пусть траектория объекта управления x(t) удовлетворяет уравнению $x = -\dot{x} + bx$.

Необходимо определить минимальное значение функционала

$$\int_{t0}^{t1} (nx^2 + x)dt$$

при условиях t0=5, t1=7, x(t0)=0, x(t1)=1, n=6, b=3.

Число шагов дискретизации возьмем равным 8.

Прямая и рекурсивные программные реализации представлены на рис.2.

Подчеркнем, что достоинством рекурсивной реализации является неизменяемый программный код при изменении числа шагов дискретизации.

Задача оптимального распределения ресурсов

Пусть имеется некоторое количество ресурсов x, которое необходимо распределить между N различными мероприятиями, фирмами, объектами, и т. д. таким образом, чтобы получить максимальную эффективность от выбранного способа распределения.

Примем следующие обозначения: x_i- количество ресурсов, выделенных i-му предприятию,

$R_j(x_i)$ - это функция эффективности использования ресурса x_i j-м мероприятием;

$f_j(x)$- критерий эффективности использования x-единиц ресурсов первыми j различными мероприятиями.

Построим математическую модель этой оптимизационной задачи:

$F_n(x)=\max\sum R_j(x_j)$
при условии , что $\sum x_j=x$, $x_j>0$, $j=1,n$

Можно показать, что
$F_1(x)=R_1(x)$,
$F_j(x)=\max (R_j(z\backslash)+F_{j-1}(x-z)),j=2,n$

Тем самым мы получили рекуррентные функциональные уравнения Беллмана .

Рассмотрим конкретную задачу по распределению средств между мероприятиями.

Пример расчета для задачи оптимального распределения ресурсов.

Совет директоров рассматривает предложения по 4 мероприятиям для увеличения производительности труда.

Для увеличения производительности труда совет директоров выделяет средства в объеме 150 у.е. с дискретностью 25 у.е. Повышение производительности труда зависит от выделенной суммы на каждое мероприятие. Эти характеристики представлены в табл. 1.

Найти распределение средств между мероприятиями, обеспечивающее максимальный прирост производительности труда, причем на одно мероприятие можно выделять не более одного вложения. Рекурсивное программная реализация представлена на рис.3

Таблица 1.Характеристики распределений

Мероприятия	Объем вложений z в у.е.					
	25	50	75	100	125	150
Мероприятие 1 $R_1(x)$	5	9	18	22	29	39
Мероприятие 2 $R_2(x)$	7	12	15	20	34	37
Мероприятие 3 $R_3(x)$	11	20	23	26	32	37
Мероприятие 4 $R_4(x)$	6	15	24	30	36	38

Задача оптимальной стратегии замены оборудования и ее математическая модель

Проблема заключается в построении оптимальной стратегии при замене старых устройств технологического оборудования на новые. Старое оборудование вызывает увеличение затрат на его ремонт и обслуживание, что обуславливает понижение его производительности. Естественно возникает необходимость динамически отслеживать временной момент , когда старое оборудование нужно заменить новым с возможностью продажи первого.

Определение оптимальных временных моментов замены оборудования составляют суть проблемы оптимальной стратегии замены оборудования.

В роли критерия оптимальности при решении этой проблемы в течение определенного временного интервала может выступать

1)максимум прибыли от эксплуатации оборудования,

2) минимум общих затрат на эксплуатацию, подлежащие минимизации.

Введем следующие обозначения :

t-дискретный момент времени,

r(t)- стоимость продукции, производимой за один год на единице оборудования возраста t лет,

u(t)- ежегодные затраты на обслуживание и

s(t) -остаточная стоимость оборудования возраста лет;

p - покупная цена оборудования.

Рассмотрим период n лет, в пределах которого требуется определить оптимальный цикл замены оборудования.

Обозначим через $F_n(t)$ максимальный доход, получаемый от оборудования возраста t лет за оставшиеся n лет цикла использования оборудования при условии оптимальной

```
NMinimize[{{Δ/2 (b u0^2 + (n x1^2 + u1^2) + (n x2^2 + u2^2) + (n x3^2 + u3^2) + (n x4^2 + u4^2) + (n x5^2 + u5^2) + (n x6^2 + u6^2) + (n x7^2 + u7^2)),
численная минимизация
  x1 - Δ b u0 == 0,
  x2 - (1 - Δ) x1 - Δ b u1 == 0,
  x3 - (1 - Δ) x2 - Δ b u2 == 0,
  x4 - (1 - Δ) x3 - Δ b u3 == 0,
  x5 - (1 - Δ) x4 - Δ b u4 == 0,
  x6 - (1 - Δ) x5 - Δ b u5 == 0,
  x7 - (1 - Δ) x6 - Δ b u6 == 0,
  1 - (1 - Δ) x7 - Δ b u7 == 0
 }, {x1, x2, x3, x4, x5, x6, x7, u0, u1, u2, u3, u4, u5, u6, u7}]

{0.141675, {x1 → 6.63382×10^-9, x2 → 1.16589×10^-7, x3 → 1.66934×10^-6, x4 → 0.0000238801, x5 → 0.000341608,
  x6 → 0.00488673, x7 → 0.0699052, u0 → 5.89673×10^-9, u1 → 9.99495×10^-8, u2 → 1.41908×10^-6, u3 → 0.0000202994, u4 → 0.0002
  u5 → 0.00415398, u6 → 0.0594231, u7 → 0.850053}}
```

Рисунок 2а-Прямое решение задачи оптимального управления, как задачи математического программирования.

```
Δ = (t1 - t0)/K;
x[(t1 - t0)/Δ] := 1
x[0] := 0
x[i_] := (1 - Δ) x[i - 1] + Δ b u[i - 1]
u[K - 1] = (x[K] - (1 - Δ) x[K - 1])/(Δ b);
S[K] = 0;
S[k_] := Δ/2 (n (x[k])^2 + (u[k])^2) + S[k + 1]

For[i = K - 2, i ≥ 0, i--, u[i] = Minimize[S[i], {u[i]}][[2, 1, 2]]]
цикл ДЛЯ                          минимизировать

N[S[0]]
численное приближение
Grid[{Table[u_i, {i, 0, K - 1}], Table[N[u[i]], {i, 0, K - 1}]}]
табл···  таблица значений         табл···  численное приближение

0.141675
```

u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7
7.21581×10^{-9}	9.92173×10^{-8}	1.41903×10^{-6}	0.0000202994	0.000290385	0.00415398	0.0594231	0.850053

Рисунок 2б-Рекурсивное решение задачи оптимального управления, как задачи математического программирования.

стратегии. Возраст оборудования отсчитывается в направлении течения процесса. Считаем, что t=0 соответствует варианту использования нового оборудования.

Временные этапы процесса нумеруются в обратном направлении по отношению к ходу процесса и нумерация k этапов и t-возраст оборудования представлена в таб.2.

Таблица 2.Нумерация этапов и возраста оборудования

Этапы- k	n	n-1	n-2	…	1	0
Возраст-t оборудования	0	1	2	…	t-1	t

На каждом этапе k=1,.. n необходимо принять решение по замене оборудования.

Критерием для решения является максимизация прибыли.

Функциональные уравнения, основанные на принципе оптимальности, имеют вид:

$$f_1(t)=max(r(t)-u(t),s(t)-p+r(0)-u(0)) \quad (1)$$
$$f_k(t)=max(r(t)-u(t)+f(t+1),s(t)-p+r(0)-u(0)) \quad (2)$$
при $к=2,..,n$

Уравнение (1) описывает одноэтапный процесс, а уравнение (2) процесс с n- этапами.

Оба уравнения состоят из двух частей: первая часть определяет доход, получаемый при сохранении оборудования; вторая часть — доход, получаемый при замене оборудования на новое.

В уравнении (2) функция r(t)-u(t) есть разность между стоимостью произведенной продукции и эксплуатационными издержками на k –м этапе процесса.

Функция $F_{k-1}(t+1)$ характеризует суммарную прибыль от оставшихся k-1 этапов для оборудования, возраст которого в начале реализации этих этапов составляет t+1 лет. Вторая часть (2) эксплицируется следующим образом: она определяет чистые издержки по замене оборудования, возраст которого t лет. Функция r(0) определяет доход, получаемый от нового оборудования возраста 0 лет. Предполагаем мгновенный переход от работы на оборудовании возраста t лет к работе на новом оборудовании.

Функция $F_{n-1}(t+1)$ представляет собой доход от оставшихся k-1 этапов ($F_0(t)=0$) , до начала выполнения которых возраст оборудования составляет один год. Аналогичная экспликация для уравнения для одноэтапного процесса, но в этом случае нет слагаемого вида $F_0(t+1)$, так как k принимает значение 1, 2,…, n.

Рассматриваемые уравнения (1) и (2) являются рекуррентными функциональными уравнениями, которые позволяют определить величину $F_k(t)$ в зависимости от $F_{k-1}(t+1)$.

Пример расчета задачи оптимальной замены оборудования

Определить оптимальный цикл замены оборудования при следующих исходных данных p=0,s(t)=0, f(t)= r(t)-u(t). Остальные данные, необходимые для расчета определены в рекурсивной программной реализация, которая представлена на рис.4.

Выводы

Необходимо подчеркнуть, что метод динамического программирования предполагает свое применение в условиях неопределенности для исследования и решения задач оптимального управления процессами.

Если считать, что условия неопределенности имеют случайную природу, то необходимо использовать методы теории вероятности и случайных процессов для оптимизации процессов управления. Если же неопределенность связана с активным целенаправленной реакцией внешней среды, то необходимо применять методы теории игр. Последние мы используем и в случае, когда необходимо обеспечить гарантированный результат либо отсечь варианты реализации худших неуправляемых воздействий.

Методы динамического программирования также часто применяют для построения оптимальных алгоритмов определения экстремумов и корней функций.

Однако наряду с универсальностью и малым объемом программного кода (хотя плохо понимаемого и трудно разрабатываемого во многих случаях) есть определенные недостатки.

В первую очередь можно указать, что метод динамического программирования является в сущности простым перебором всевозможных решений, хотя и экономным ввиду использования численных результатов предыдущих этапа на текущем. Поэтому быстрый рост вариантов решения обуславливает важность нейтрализации проклятия размерности, что подчеркивал Р.Беллман.

Для этого возможны следующие основные пути:

1) существенно жертвовать точностью вычислений;

2) определять приближенные управляющие воздействия и соответствующие траектории объекта управления, которые будут являться глобально оптимальными по отношению к малым (локальным) вариациям находимых приближений.

Многочисленные примеры программных реализаций образцов применения МДП на языках C, Java, а также в пакетах Mathlab, Maple представлены в [10-13].

```
x = 150;
d = 25;
n = 4;

dom = Table[d i, {i, 1, x/d}];
         таблица значений

    ( 5   9  18  22  29  39 )
V = ( 7  12  15  20  34  37 ) ;
    (11  20  23  26  32  37 )
    ( 6  15  24  30  36  38 )

Do[Do[R[i, d z] = V[[i, z]], {z, 1, Length[V[[1]]]}], {i, 1, n}]
о··· оператор цикла                      длина

F[j_, 0] := {0, 0}
F[1, x_] := {R[1, x], x}
F[1, 0] := {0, 0}
F[j_, x_] := SortBy[Table[Join[{R[j, z] + F[j - 1, x - z][[1]]}, F[j - 1, x - z][[2 ;;]],
                  сортир··· табли··· соединить
{z}], {z, Select[dom, # <= x &]}], -#[[1]] &][[1]]

res = F[n, x];
Print["Эффективность: ", res[[1]]];
печатать
For[i = 2, i ≤ Length[res], i++, Print["Мероприятие ", i - 1, ": ", res[[i]]]];
цикл ДЛЯ     длина                  печатать

Эффективность: 51
Мероприятие 1: 0
Мероприятие 2: 25
Мероприятие 3: 50
Мероприятие 4: 75
```

Рисунок 3-Рекурсивная реализация задачи оптимального управления ресурсами.

```
In[12]:= p = 11;
         n = 7;

In[14]:= v = (0   1   2  3  4  5  6
              12  10  8  6  5  4  3
              0   0   1  1  1  2  2
              12  10  6  4  3  2  1); (* (t
                                           r[t]
                                           u[t]
                                           s[t]) *)
         f[t_] = 0;
         For[i = 1, i < Length[v[[1]]], i++,
         |цикл ДЛЯ        |длина
          f[v[[1, i]]] = v[[2, i]] - v[[3, i]];
          s[v[[1, i]]] = v[[4, i]];
         ]

In[17]:= f[k_, t_] := Max[f[t] + f[k - 1, t + 1], -p + f[0] + f[k - 1, 1]]
                        |максимум
         f[1, t_] := Max[f[t], -p + f[0]]
                     |максимум

In[19]:= result = Table[Table[f[k, t], {t, 0, n}], {k, 1, n}];
                  |табли··· |таблица значений
         Print[Grid[result]]
         |печат··· |таблица
         12 10  7  5  4  2  1  1
         22 17 12 11 11 11 11 11
         29 22 18 18 18 18 18 18
         34 28 25 23 23 23 23 23
         40 35 30 29 29 29 29 29
         47 40 36 36 36 36 36 36
         52 46 43 41 41 41 41 41

In[21]:= For[i = 1, i ≤ n && result[[n, i]] > -p + f[0] + f[n - 1, 1], i++, 0]
         |цикл ДЛЯ
         Print ["Оптимальный год замены оборудования - ", i - 1]
         Оптимальный год замены оборудования - 3
```

Рисунок 4-Рекурсивная реализация задачи оптимальной замены оборудования.

Литература

1.Ф.Л.Черноусько. Динамическое программирование//Соросовский образовательный журнал, 1998,№ 2, с.139-144.
2. Черноусько Ф.Л., Баничук Н.В. Вариационные задачи механики и управления: Численные методы. М.: Наука, 1973.
3. Беллман Р. Динамическое программирование. .М.:Изд-во иностранная литература., 1960.
4. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М.: Наука, 1965.
5. Беллман Р., Калаба Р. Динамическое программирование и современная теория управления. М.: Наука, 1969.
6.Шалыто А., Туккель Н., Шамгунов Н. Ханойские башни и автоматы // Программирование. 2002. №8, С.82-90.
7. Д.Хофштадтер. Гедель, Эшер, Бах: эта бесконечная гирлянда.Самара,2001,752 с.
8.Анисимов А. В. Рекурсивные преобразователи информации К.:Вища школа,1987,225 с.
9.В.Любченко. О борьбе с рекурсией. Мир ПК, 2002, № 11, С.150-152.
10.A. Lew, H. Mauch.Dynamic Programming.A Computational Tool. Springer-Verlag Berlin Heidelberg 2007
11. Applied dynamic programming for optimization of dynamical systems / Rush D. Robinett III ... [et al.].,SIAM,Philadelphia,2005.
12.Moshe Sniedovich, Dynamic Programming: Foundations and Principles.// CRC Press Taylor & Francis Group Second Edition. 2011.
13. Huaguang Zhang, Derong Liu, Yanhong Luo, Ding Wang. Adaptive Dynamic Programming for Control.Algorithms and Stability// Springer-Verlag London, 2013.

Андрюхин А.І. Полєтаєв В.А. Рекурсивні реалізації завдань динамічного програмування. *В роботі розглянуті рекурсивні реалізації відомих задач вирішуються за допомогою методу динамічного програмування. Представлені завдання оптимального управління, завдання оптимального розподілу ресурсів, завдання оптимальної заміни обладнання. Наведено загальний алгоритм вирішення задач динамічного програмування. Вказані переваги і недоліки застосування принципу Беллмана і рекурсії в програмних реалізаціях. Наведено результати комп'ютерних расчетов. В розрахунках використовувався пакет Mathematica.*

Ключові слова: *динамічне програмування, принцип Беллмана, рекурсія.*

Andruckin A.I., Poletaev V.A. Recursive implementation of dynamic programming problems. *The paper discusses the recursive implementation of the known problems that can be solved using dynamic programming method. The optimal control problem, the problem of optimal allocation of resources, the problem of optimal replacement equipment are presented. General algorithm for solving dynamic programming problems presented. Advantages and disadvantages of the application of the principle of Bellman and recursion are discussed in software implementations. Computing results are shown. Mathematica package was used in the calculations.*

Key words*: dynamic programming, Bellman principle, recursion.*

Статья поступила в редакцию 20.05.2016
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом