

# УСТАРЕВШИЕ СТЕРЕОТИПЫ О ДИНАМИЧЕСКОМ ПРОГРАММИРОВАНИИ В ОБРАЗОВАНИИ И НАУКЕ

Козлов А.Н.

*Московский институт радиотехники, электроники и автоматики, Москва, Россия*

Динамическое программирование появилось в сороковых годах прошлого века благодаря американскому математику Ричарду Беллману, который сформулировал принцип оптимальности и на его основе рекуррентное уравнение, получившие его имя. С тех пор прошло более полувека и такие задачи оптимизации, как распределение ресурса до сих пор решаются с использованием принципа Беллмана.

Но проблема даже не в том, что его зачастую применяют в задачах, где его эффективность крайне низка. Гораздо хуже то, что такое нерациональное использование во многих источниках ([1, с.55-57], [2, с. 104-106], [3, с. 209-213], [4, с. 125-129]) позиционируется верным и логичным, что на самом деле не так. И вместо того, чтобы предложить другой метод решения таких «неудобных» задач, оказывается медвежья услуга – при попытке показать мощь и универсальность метода Беллмана он загоняется в крайне неблагоприятные для него условия и обнажается его главный недостаток – крайне низкая его эффективность с использованием регулярной сетки в задачах с большим числом состояний. Это вовсе не означает, что метод плох, он просто несовершенен и его область применения ограничена особенностями конкретных исходных задач.

Такая политика не может не настораживать, особенно, когда неправильное, а иной раз и извращённое применение метода Беллмана печатается в популярных пособиях для студентов. Рассмотрим конкретную задачу распределения ресурса из книги [2, с. 104-106] и её решение с некорректным применением метода Беллмана, а потом предложим более эффективные методы её решения.

**Задача.** Имеется некая грузовая машина, которую нужно заполнить предметами  $\Pi_1, \Pi_2, \dots, \Pi_6$  с весами  $q_1, q_2, \dots, q_6$  и стоимостями  $c_1, c_2, \dots, c_6$  так, чтобы суммарный вес не превышал грузоподъёмность машины  $Q=35$ , а суммарная стоимость была максимальная. Для простоты введено ограничение: нельзя брать большего одного предмета каждого типа. Исходные данные приведены в таблице 1.

Таблица 1

Предмет $\Pi_i$	$\Pi_1$	$\Pi_2$	$\Pi_3$	$\Pi_4$	$\Pi_5$	$\Pi_6$
Вес $q_i$	4	7	11	12	16	20
Стоимость $c_i$	7	10	15	20	27	34

Автор начинает решать задачу с последнего, шестого предмета, рассматривая состояние как величину ресурса грузоподъёмности ещё оставшегося в распоряжении. На

каждом шаге  $i$  для каждого значения оставшегося ресурса  $S$  рассчитывается максимально достижимый выигрыш  $W_i=F_i(S)$ , перебирая варианты распределения ресурса между текущим и всеми предыдущими шагами во всевозможных пропорциях. Также запоминается значение управляющего параметра  $x_i$ , при котором достигается максимальный выигрыш. Сформированные на всех этапах состояния представлены в таблице 2 [2, с. 106]. Для сокращения размеров таблицы и её большей наглядности дублирующие строки пропущены.

Таблица 2

S	I=6		i=5		i=4		i=3		i=2		i=1	
	$x_i$	$W_i$	$x_i$	$W_i$	$x_i$	$W_i$	$x_i$	$W_i$	$x_i$	$W_i$	$x_i$	$W_i$
0	<u>0</u>	0	0	0	0	0	0	0	0	0		
...	...	...	...	...	...	...	...	...	...	...		
7	0	0	0	0	0	0	0	0	1	10		
...	...	...	...	...	...	...	...	...	...	...		
11	0	0	0	0	0	0	1	15	0	15		
12	0	0	0	0	1	20	0	20	0	20		
...	...	...	...	...	...	...	...	...	...	...		
16	0	0	<u>1</u>	27	0	27	0	27	0	27		
...	...	...	...	...	...	...	...	...	...	...		
19	0	0	1	27	0	27	0	27	1	30		
20	1	34	0	34	0	34	0	34	0	34		
...	...	...	...	...	...	...	...	...	...	...		
23	1	34	0	34	0	34	1	35	1	37		
...	...	...	...	...	...	...	...	...	...	...		
27	1	34	0	34	0	34	1	42	1	44		
28	1	34	0	34	<u>1</u>	47	<u>0</u>	47	0	47		
...	...	...	...	...	...	...	...	...	...	...		
31	1	34	0	34	1	47	1	49	0	49		
32	1	34	0	34	1	54	0	54	0	54		
...	...	...	...	...	...	...	...	...	...	...		
35	1	34	0	34	1	54	0	54	<u>1</u>	57	<u>0</u>	<u>57</u>

В процессе решения на каждом шаге при совпадении в ресурсе отбрасывается тот путь (и соответствующее ему состояние), который имеет меньший выигрыш. После последнего шага получается оптимальное решение, опираясь на которое обратным разворотом можно восстановить всю траекторию оптимального движения.

Даже при первом знакомстве с таблицей очевидно, что суммарное число состояний больше состояний при полном переборе – их  $36*5+1=181$  против  $2*(2^6-1)/(2-1)=126$ , что связано с обработкой большого количества недостижимых состояний. А если для точности решения понадобится брать дискрет ещё меньше – допустим 0,001 вместо 1 при  $q_1=3,995$  – тогда необходимо рассматривать  $(35/0,001+1)*5+1=175006$  (!) состояний, львиная доля которых попросту недостижима. Причина данного явления заключается в неверном подходе и некорректном использовании метода Беллмана. Автор опирается на тот факт, что не имеет

принципиального значения направление решения задачи – с начала в конец, то есть формировать состояния динамически, шаг за шагом, начиная с нулевого; или с конца в начало, зная при этом конечное оптимальное состояние. Однако проблема в том, что его наперёд невозможно вычислить, поэтому приходится вводить такой минимальный дискрет, при котором оптимальное решение в целом гарантированно не потеряется. Возникает дилемма: введение малого дискрета приведёт к значительному росту числа состояний и времени расчёта, а его увеличение приведёт к погрешностям в решении.

Попытаемся теперь решить ту же задачу с использованием динамического формирования состояний – теперь состояния будут формироваться не относительно выбранного дискрета по ресурсу, а посредством последовательного перебора предметов, но только совместно с правилами отбраковки неконкурентоспособных состояний.

Перед первым шагом имеется единственное состояние со значением затраченного ресурса 0 и выигрышем 0, то есть 0(0). На первом шаге имеется возможность не брать предмет – остаётся состояние 0(0) с управляющей траекторией [0xxxxx] и взять предмет – появится состояние 4(7) с управляющей траекторией [1xxxxx]. На втором шаге процедура повторяется с одним лишь различием – начальным состоянием уже считается не 0(0), а пара состояний – 0(0) и 4(7). Для каждого из них опять делается выбор – взять второй предмет или не взять его. При этом затраченный ресурс и выигрыш предыдущего состояния суммируется с соответствующими значениями на текущем, втором шаге. В итоге, образуются четыре состояния: 0(0) [00xxxx], 4(7) [10xxxx], 7(10) [01xxxx], 11(17) [11xxxx]. На третьем шаге – 0(0), 4(7), 7(10), 11(15), 11(17), 15(22), 18(25), 22(32). Вот здесь начинается самое интересное – имеются два совпадающих по ресурсу состояния – 11(15) и 11(17). Согласно принципу оптимальности Беллмана [1, с. 36-37] – на дальнейший выбор предметов не оказывают влияние обстоятельства достижения этих состояний – можно отбросить состояние 11(15) и оставить 11(17) [5, с. 88]. Следовательно, итоговое множество состояний на третьем шаге имеет вид: 0(0), 4(7), 7(10), 11(17), 15(22), 18(25), 22(32).

А теперь следует обратить внимание вот на что: сработал тот же принцип Беллмана отбраковки сходящихся путей, что и в предыдущем методе решения, однако в данном случае не происходит рассмотрение избыточных состояний, а только реально достижимых. А стоило только правильно применить метод Беллмана, и уже вместо 108 состояний на первых трёх этапах получаем лишь 13.

Однако оказывается, что в реальных задачах распределения ресурса такое пересечение путей довольно редкое событие. Поэтому целесообразно ввести дополнительную отбраковку, но уже не путей, а неконкурентоспособных состояний. Продолжим решение задачи с того места, на котором остановились. Итак, применяя всё то же правило

формирования состояний, на четвёртом шаге получаем следующие состояния: 0(0), 4(7), 7(10), 11(17), 12(20), 15(22), 16(27), 18(25), 19(30), 22(32), 23(37), 27(42), 30(45), 34(52). В этом ряду везде выигрыши от достижения этих состояний расположены в порядке возрастания, за исключением состояния 18(25), которое менее выгодно по сравнению с 16(27) – затраты больше, а выигрыш меньше. Руководствуясь здравой логикой и принципом «Зачем платить больше», можно без последствий отбраковать состояние 18(25) и оставить 16(27) [5, с. 89]. Таким образом, остаются следующие состояния: 0(0), 4(7), 7(10), 11(17), 12(20), 15(22), 16(27), 19(30), 22(32), 23(37), 27(42), 30(45), 34(52).

Сформировав состояния для всех шести шагов и зная, что максимум достигается при максимальном вложении ресурса, выбираем последнее состояние на последнем шаге как оптимальное. Так как мы сохраняли путь достижения всех состояний, то легко вычислить всю оптимальную траекторию. Для данной задачи она – [010110], а выигрыш – 57.

Итак, применение противоположного правила формирования состояний – динамического вместо регулярной сетки – позволило более эффективно использовать метод Беллмана. Таким образом, при корректном использовании метода Беллмана его эффективность значительно выше, чем при некорректном. И дело даже не в выигранном времени или точности решения задач оптимального распределения ресурса, а в принципиальной ненаучности подхода. Хотя как вариант, метод Беллмана можно использовать в такой, не свойственной для него форме, только в качестве демонстрации его универсальности и примера того, как его не нужно применять.

#### СПИСОК ЛИТЕРАТУРЫ:

1. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. – М.: Наука, 1965. – 460 с.
2. Вентцель Е.С. Исследование операций: задачи, принципы, методология. – 3-е изд., стереотип. – М.: Дрофа, 2004. – 208 с.
3. Косоруков О.А., Мищенко А.В. Исследование операций: Учебник / Косоруков О.А., Мищенко А.В. // Под. общ. ред. д.э.н., проф. Н.П. Тихомирова. – М.: Издательство «Экзамен», 2003. – 448.
4. Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы. – Изд. 2-е, испр. – М.: Физматлит, 2003. – 240 с.
5. Струченков В.И. Методы оптимизации в прикладных задачах. – М.: СОЛОН-ПРЕСС, 2009. – 320 с.