

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ РЕШЕНИЯ ЗАДАЧИ ПЛАНИРОВАНИЯ РАБОЧЕЙ СИЛЫ

Д. А. Дворанинович

БрГТУ, Брест

Научный руководитель: Т. Г. Хомицкая

Аннотация: представлена программная реализация применения методов динамического программирования для осуществления планирования производственных процессов на примере задачи распределения рабочей силы.

Ключевые слова: планирование, динамическое программирование

Актуальность. В реальной жизни часто при исследовании производственных процессов встречаются задачи, которые представляют собой совокупность подзадач, рассматриваемых на некоторых шагах или этапах. В частности, такими могут быть задачи планирования и распределения. Структура представления задач данного класса наталкивает на использование при их решении методов динамического программирования.

Постановка задачи. Динамическое программирование определяет оптимальное решение n -мерной задачи путем ее декомпозиции на n этапов, каждый из которых представляет подзадачу относительно одной переменной. [1, с. 343; 2, с.441] Вычислительное преимущество такого подхода состоит в том, что выполняется решение одномерных оптимизационных подзадач вместо большой n -мерной задачи. Фундаментальным принципом динамического программирования, составляющим основу декомпозиции задачи, является оптимальность. Так как природа каждого этапа решения зависит от конкретной оптимизационной задачи, динамическое программирование не предлагает вычислительных алгоритмов непосредственно для каждого этапа. Вычислительные аспекты решения оптимизационных подзадач на каждом этапе проектируются и реализуются по отдельности (что, конечно, не исключает применения единого алгоритма для всех этапов).

Вычисления в динамическом программировании выполняются рекуррентно в том смысле, что оптимальное решение одной подзадачи используется в качестве исходных данных для следующей подзадачи. Решив последнюю подзадачу, будет получено оптимальное решение исходной задачи. Способ выполнения рекуррентных вычислений зависит от того, как производится декомпозиция исходной задачи. В частности, подзадачи обычно связаны между собой некоторыми общими ограничениями. Если осуществляется переход от одной подзадачи к другой, то должны учитываться эти ограничения.

В общем случае задача динамического программирования о распределении рабочей силы формируется следующим образом: При выполнении некоторых проектов число рабочих, необходимых для реализации какого-либо проекта,

регулируется путем их найма и увольнения. Поскольку как найм, так и увольнение рабочих связано с дополнительными затратами, необходимо определить, каким образом должна регулироваться численность рабочих в период реализации проекта [2, с. 455].

Задача планирования рабочей силы является задачей динамического программирования. При рассмотрении задачи можно выделить три основных элемента моделей динамического программирования [2, с. 446].

1. Определение этапов.
2. Определение на каждом этапе вариантов решения (альтернатив).
3. Определение состояний на каждом этапе.

Математическое обоснование. Предположим, что проект будет выполняться в течение n недель и минимальная потребность в рабочей силе на протяжении i -ой недели составит b_i рабочих. При идеальных условиях хотелось бы на протяжении i -ой недели иметь ровно b_i рабочих.

Однако, в зависимости от стоимостных показателей может быть более выгодным отклонение численности рабочей силы как в одну, так и в другую сторону от минимальных потребностей.

Элементы модели динамического программирования определяются следующим образом:

1. *Этап i* представляется порядковым номером недели i , $i = 1, 2, \dots, n$.
2. *Вариантами решения* на i -м этапе являются значения x_i – количество работающих на протяжении i -й недели.
3. *Состоянием* на i -м этапе является x_{i-1} количество работающих на протяжении $(i - 1)$ -й недели (этапа).

В общем случае для подобной задачи можно вычислить общую формулу, которую можно модифицировать под нужды конкретной задачи.

$$f_i(x_{i-1}) = \min_{x_i \geq b_i} \{C_1 * (x_i - b_i) + C_2 * (x_i - x_{i-1}) + f_{i+1}(x_i)\}, i = 1, 2, \dots, n$$

где $f_{n+1}(x_n) \equiv 0$,

C_1 – содержание избытка рабочей силы,

C_2 – сумма найма рабочих на неделю,

b_i – минимум рабочей силы.

Вычисления начинаются с этапа n при $x_n = b_n$ и заканчиваются на этапе 1 [2, с. 455].

К особенностям данной модели можно отнести то, что вычисления начинаются с последнего этапа, передвигаясь к начальному этапу с конца. Для цели упрощения математических операций, в данной задаче конечное число рабочих принимается за необходимое количество рабочих, что даёт максимальную оптимизацию на последнем этапе.

Модификации данного алгоритма могут применяться для разных задач так или иначе связанных с планированием рабочей силы или распределением ее во времени на производственных участках.

Процесс реализации алгоритма. Для реализации данного алгоритма в его классическом виде использовался язык C++. Была создана структура, которая представляет собой таблицу с реализацией расчета на каждом этапе. При этом заполнение таблиц и получение результата вынесено за пределы данной структуры и предоставляет собой отдельный модуль, в который включен массив из структур и проведение вычислений по алгоритму.

Изначально пользователь с клавиатуры вводит значения для системы, которые будут применяться для расчета. Этот этап является общим для всей программы. Вариант реализации представлен на рисунке 1.

```
Период (количество периодов): 5
Минимальная потребность в рабочей силе по периодам:
Минимальная потребность в рабочей силе по периоду №1: 8
Минимальная потребность в рабочей силе по периоду №2: 7
Минимальная потребность в рабочей силе по периоду №3: 4
Минимальная потребность в рабочей силе по периоду №4: 8
Минимальная потребность в рабочей силе по периоду №5: 7
Стоимость содержания избытка одного рабочего: 4
Стоимость найма одного рабочего: 4
Стоимость содержания одного рабочего: 5
```

Рисунок 1 – Пример заполнения изначальных параметров системы.

В программе предусмотрено два варианта работы: автоматизированный и автоматический.

В автоматизированном режиме пользователь получает совокупность таблиц (рисунок 2), с помощью которых пользователь сам принимает решение по распределению персонала.

№5	7	F(x)	x*					
8	0	0	7					
№4	8	F(x)	x*					
4	24	24	8					
5	19	19	8					
6	14	14	8					
7	9	9	8					
8	0	0	8					
№3	4	5	6	7	8	F(x)	x*	
7	24	23	22	21	25	21	7	
8	24	23	22	21	16	16	8	
№2	7	8	F(x)	x*				
8	21	20	20	8				
№1	8	F(x)	x*					
0	64	64	8					

Рисунок 2 – Пример работы автоматизированного режима программы.

В автоматическом случае работы программы, пользователь получает готовое распределение в виде таблицы, в котором указаны: номер периода, минимальная потребность в рабочей силе и оптимальное количество работников. Пример вывода указан на рисунке 3.

№Пер.	мин	раб
1	8	8
2	7	8
3	4	8
4	8	8
5	7	7

Рисунок 3 – Пример работы автоматического режима программы.

Практическое применение. Представленная программа может использоваться для обучения студентов методам динамического программирования по предмету «Системный анализ и исследование операций», а так же для планирования распределения персонала по производственным участкам. Разработанная программа может служить основой для реализации более сложных алгоритмов планирования.

Вывод. Модели динамического программирования являются эффективным инструментом решения прикладных задач. Посредством моделей динамического

программирования и их модификаций, возможно решение комплексных и изменяемых во времени задач, которые решить стандартными линейными математическими методами практически невозможно.

Список литературы:

1. Жолобов, Д.А. Введение в математическое программирование: Учебное пособие. – М.:МИФИ, 2008. – 376 с.
2. Таха, Хэмди А. Введение в исследование операций, 7-е издание.: Пер. с англ. – М. : Издательский дом “Вильямс”, 2007. – 912 с.

УДК 004.021:032.26

ОБУЧЕНИЕ ДВУХКОМПОНЕНТНЫХ НЕЙРОННЫХ СЕТЕЙ

В. М. Жежеров, Т. А. Лузько

Брестский государственный технический университет, г. Брест

Научные руководители: Л. П. Махнист, канд. техн. наук, доцент,

Т. И. Каримова, канд. физ.-мат. наук, доцент

В работе рассматривается обучение искусственных нейронных сетей прямого распространения с использованием метода обратного распространения ошибки.

Предполагалось, что все слои сети содержат два нейронных элемента.

Обучение такой нейронной сети с использованием метода обратного распространения ошибки состояло в нахождении весовых коэффициентов w_{ij} и порогов T_j нейронной сети, которые минимизируют функцию ошибки сети, построенную по методу наименьших квадратов

$$E(w_{11}, w_{21}, T_1, w_{12}, w_{22}, T_2) = \frac{1}{2} \left((y_1 - t_1)^2 + (y_2 - t_2)^2 \right),$$

где $y_j = F(S_j)$ – значение гладкой функции активации j -го выходного нейрона сети, $S_j = w_{1j}x_1 + w_{2j}x_2 - T_j$, x_i – выходное значение i -го нейрона предыдущего слоя,

t_j – ожидаемый выход j -го выходного нейрона ($i = \overline{1, 2}$, $j = \overline{1, 2}$).

Используем обозначения:

$\overline{W} = (w_{11}, w_{21}, T_1, w_{12}, w_{22}, T_2)^T$ – вектор-столбец весовых коэффициентов w_{ij} и

порогов T_j нейронной сети, а $\overline{W}_j = (w_{1j}, w_{2j}, T_j)^T$ – вектор-столбец весовых коэффициентов w_{ij} и порога T_j , связанных с j -ым выходным нейроном сети,