

# ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ ОЦЕНОК ДЛЯ ТАБЛИЧНОГО АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ ОПТИМАЛЬНОЙ УПАКОВКИ НА ОСНОВЕ ФУНКЦИИ ТРУДОЁМКОСТИ

*М.В. Ульянов,*

*д.т.н, профессор Московского государственного университета печати*

*muljanov@mail.ru*

*О.А. Наумова, И. А. Яковлев,*

*студенты Московского государственного университета приборостроения и информатики*

*В статье излагается подход к прогнозированию временных оценок программных реализаций компьютерных алгоритмов, основанный на использовании теоретической функции трудоёмкости и функции времени выполнения обобщённой базовой операции, получаемой на основе экспериментальных данных с применением аппарата регрессионного анализа. Иллюстративным примером предлагаемого подхода служит табличный алгоритм решения классической задачи одномерной оптимальной упаковки.*

## Введение

При решении класса экономических задач и задач планирования бизнеса, обладающих большой размерностью, возникает дилемма между применением точного алгоритма решения задачи со значительными временными затратами или быстрых эвристических алгоритмов, не гарантирующих получение точного решения. В этом аспекте практический интерес представляет получение прогноза временных затрат для получения точного решения. Ряд задач указанного класса, например, задача об оптимальных расписаниях, задача формирования оптимального пакета акций на фиксированную сумму, задача одномерного раскрытия материала, сводятся к классической задаче одномерной упаковки, которая и служит иллюстративным примером подхода к прогнозированию временных оценок, излагаемого в настоящей статье.

Представим себе, что у нас есть рюкзак с прямоугольным дном и нерастяжимыми стенками определённой высоты. У нас есть также несколько групп коробок, с таким же дном, как у рюкзака. В любой группе очень много коробок, и каждая коробка в группе одинакова по высоте и имеет

определённую стоимость. Наша задача так упаковать рюкзак, чтобы он закрывался, и сумма стоимостей упакованных коробок была бы наибольшей. Хотя мы имеем дело с объёмом, в реальности мы рассматриваем только высоту рюкзака и высоты коробок — в этом смысле задача одномерная. Эта задача, известная, как задача об упаковке рюкзака, более корректно — задача одномерной оптимальной по стоимости упаковки — имеет разнообразные практические применения.

Точное решение этой задачи, существенно сокращающее полный перебор, впервые предложено Р. Беллманом на основе разработанного им метода динамического программирования [1]. В настоящее время предлагается много разнообразных точных и эвристических алгоритмов её решения. Из классических алгоритмов точного решения задачи упаковки наибольший интерес представляет табличный алгоритм, позволяющий получить промежуточные решения для всех дискретов объёма упаковки, что важно в аспекте исследования чувствительности целевой функции. Такого рода исследования чувствительности актуальны для ряда экономических задач, сводимых к задаче одномерной упаковки.

Наиболее простой способ решения задачи прогнозирования – обработка экспериментальных времён выполнения методами математической статистики. Получаемые известными статистическими пакетами [2] функции тренда хорошо работают в области интерполяции значений времени выполнения как функции размерности задачи, но дают значительные ошибки при экстраполяции этой зависимости в область больших размерностей. Очевидная причина этих ошибок – отсутствие знаний о функции трудоёмкости исследуемого алгоритма.

Цель настоящей статьи – изложение подхода к прогнозированию временных оценок программных реализаций компьютерных алгоритмов, основанного на использовании многопараметрической функции трудоёмкости алгоритма и времени выполнения обобщённой базовой операции на примере табличного алгоритма решения задачи одномерной упаковки.

#### Постановка задачи упаковки, уравнение Беллмана и параметризация

Рассмотрим общую постановку задачи одномерной оптимальной по стоимости упаковки [1]. Пусть задано множество типов грузов

$$Y = \{y_i\}, \quad y_i = \{v_i, c_i\}, \quad i = \overline{1, n},$$

где каждый элемент  $y_i$ , соотнесенный с типом груза, обладает целочисленным линейным размером –  $v_i$ , или «объёмом» в общепринятых терминах задачи упаковки, и ценовой характеристикой –  $c_i$ , которая содержательно отражает практически значимые предпочтения для загрузки объектов данного типа. Также целочисленным значением задан объём упаковки  $V$ . В классической постановке элементы  $y_i$  называются типами грузов.

Для описания количества загружаемых в объём  $V$  элементов  $y_i$  введём в рассмотрение характеристический вектор

$$\mathbf{x} = \{x_i\}, \quad x_i = \overline{1, n},$$

где  $x_i$  – неотрицательное целое, т.е.

$$\mathbf{x} \in E_2^n, \quad x_i \geq 0.$$

Значение компонента вектора  $x_i = k$  соответствует загрузке  $k$  элементов типа  $y_i$  в объём  $V$ . Таким образом, описание некоторой упаковки объёма  $V$  представляет собой целочисленную точку в неотрицательной области  $n$ -мерного пространства  $E_2^n$ . Среди всех возможных упаковок существует по крайней

мере одна, максимизирующая суммарную стоимость. Это приводит к следующей постановке задачи упаковки как задачи линейного целочисленного программирования – максимизировать линейный функционал:

$$P_n(\mathbf{x}) = \sum_{i=1}^n C_i(x_i) = \sum_{i=1}^n x_i \cdot c_i \rightarrow \max \quad \text{при} \quad \sum_{i=1}^n x_i \cdot v_i \leq V,$$

где ограничение содержательно означает, что суммарный объём, занимаемый грузами всех типов в количествах, указанных характеристическим вектором  $\mathbf{x}$ , не должен превышать объёма упаковки.

Исходные данные на входе алгоритма:

- ✧ объём упаковки;
- ✧ количество типов грузов, каждый из которых характеризуется объёмом  $v_i$  и стоимостью  $c_i$ ,  $i = \overline{1, n}$ .

Решение методом динамического программирования предполагает получение промежуточных результатов, формирующихся на основе функционального уравнения Беллмана для задачи упаковки [1]:

$$\begin{cases} f_0(v) = 0; \\ f_m(v) = \max_{x_m} \{x_m \cdot c_m + f_{m-1}(v - x_m \cdot v_m)\}, \\ m = \overline{1, n}, \quad v = \overline{0, V}, \quad x_m = 0, 1, \dots, \left[ \frac{v}{v_m} \right], \end{cases} \quad (1)$$

где  $f_m(v)$  – оптимальная стоимость упаковки объёма  $v$  грузами  $m$  типов;  
 $x_m$  – количество грузов типа  $m$  в упаковке объёма  $v$ .

Решением поставленной задачи является значение  $f_n(V)$  и соответствующий вектор

$$\mathbf{x} = \{x_i\}, \quad x_i = \overline{1, n}.$$

Исследуемый табличный алгоритм упаковки реализует функциональное уравнение (1) путём вычисления значений всех строк таблицы для изменяющегося значения дискрета объёма. По типу он – количественно-параметрический алгоритм [3]. Это означает, что число базовых операций, задаваемых алгоритмом, зависит не только от количества данных на входе, но и от их значений. В рассматриваемой задаче значение  $x_m$  зависит от  $v$  и  $v_m$ , следовательно, функция трудоёмкости табличного алгоритма будет зависеть как от числа типов грузов, так и от значений параметров –  $V, v_1, \dots, v_n$ , учёт которых существенно затрудняет анализ.

С целью упрощения анализа алгоритма, следуя [3] введём параметр  $\bar{v}$ , характеризующий, сколько грузов среднего объёма размещается в объёме  $V$ :

$$\bar{v} = \frac{1}{n} \cdot \sum_{i=1}^n v_i, \quad k = \frac{V}{\bar{v}}.$$

Очевидно, что реально количество любых грузов, размещённых в объёме  $V$ , есть целое число, но для оценки вычислительной сложности в среднем, необходимо учитывать, что параметр  $k$  может рассматриваться как действительное число.

**Табличный алгоритм упаковки**

Табличный алгоритм, позволяет получить набор оптимальных решений не только для заданного объёма, но и для всех промежуточных дискретных значений этого объёма. Обозначим вектор оптимальной упаковки для объёма  $v$  элементами  $y_j, j = \overline{1, m}, m \leq n$ , через  $\mathbf{x}_v^m$ , тогда в силу принципа оптимальности [1]

$$f_m(v) = \max_{\mathbf{x}_m} P_m(\mathbf{x}), \quad P_m(\mathbf{x}) = \sum_{i=1}^m x_i \cdot c_i,$$

и порядок предъявления элементов  $y_j$  не является существенным. Решение табличным алгоритмом осуществляется на основе функционального уравнения (1). Результат решения задачи – набор оптимальных значений целевой функции  $f_n(v)$  и соответствующих векторов оптимальной упаковки  $\mathbf{x}_v^m$  для всех объёмов  $v$  от 0 до  $V$  элементами из множества  $Y$ . Отметим, что поскольку табличный алгоритм позволяет получить оптимальные решения для всех промежуточных объёмов упаковки, эту информацию можно использовать для исследования чувствительности целевой функции  $f_n(v)$  по изменению объёма упаковки. Табличная реализация основного функционального уравнения Беллмана приводит к последовательному рассмотрению типов грузов. Для каждого из них мы получаем оптимальную упаковку для всех дискретов объёма. Рекурсивный вызов в функциональном уравнении (1) реализуется табличным алгоритмом как обращение к соответствующей строке предыдущей таблице оптимальной упаковки.

Запись табличного алгоритма решения задачи упаковки в алгоритмическом базисе, введённом в [4], имеет следующий вид (справа в строках указано число базовых операций модели вычислений языка высокого уровня, подробнее см. [4]).

**A(boxC, boxV, n, V; f, x)**  
 (boxV[1...n] – объёмы типов грузов)  
 (boxC[1...n] – стоимости типов грузов)  
 (n – количество типов грузов)

(V – объём упаковки)  
 (f[0...V], f1[0...V] – функция Беллмана – оптимальная стоимость упаковки для текущего и предыдущего типов грузов)  
 (x[0...V, 1...n], x1[0...V, 1...n] – массивы векторов оптимальной упаковки для текущего и предыдущего типов грузов)

**(Фрагмент 1.** Формирование таблицы для грузов типа 1)  
 for v ← 0 to V 1+3(V+1)  
   **begin**  
     x[v, 1] ← v div boxV[1] 5(V+1)  
     x1[v, 1] ← x[v, 1] 5(V+1)  
     f[v] ← x[v, 1] \* boxC[1] 6(V+1)  
     f1[v] ← f[v] 3(V+1)  
   **end**

**(Фрагмент 2.** Цикл по типам грузов)  
 for i ← 2 to n 1+3(n-1)  
   **begin**  
     Vi ← boxV[i] 2(n-1)  
     Ci ← boxC[i] 2(n-1)  
   **end**

**(Фрагмент 3.** Цикл по дискретам объёма упаковки)  
 for v ← 0 to V 1(n-1)+3V(n-1)  
   **begin**  
     maxKol ← 0 1V(n-1)  
     maxC ← f[v] 2V(n-1)  
     k ← v div Vi 2V(n-1)  
   **end**

**(Фрагмент 4.** Цикл нахождения максимума f)  
 for b ← 0 to k V(n-1)+3g  
   **begin**  
     C ← Ci\*b+f1[v-b\*Vi] 6g  
     if maxC < C 1g  
       **then**  
         maxC ← C 1  
         maxKol ← b 1  
   **end** (конец цикла по b)

**(Фрагмент 5.** Сохранение оптимального решения)  
 f[v] ← maxC 2V(n-1)  
 jr ← v-maxKol\*Vi 3V(n-1)

**(Фрагмент 6.** Формирование оптимального вектора упаковки)  
 for j = 1 to i- 2V(n-1)+3V(n-1)\*n/2  
   **begin**  
     x[v, j] = x1[jr, j] 5V(n-1)\*n/2  
   **end**  
   x[v, i] = maxKol 3V(n-1)  
   **end** (конец цикла по v)  
 (Переход к новому типу груза: обмен адресов x, x1 и f, f1)  
   **end** (конец цикла по i)

End.

**Анализ трудоёмкости табличного алгоритма**

Трудоёмкость трёх первых фрагментов табличного алгоритма может быть легко получена на основе базовых операций, указанных в строках записи, и составляет:

$$\begin{cases} f_1(V) = 22V + 23, \\ f_2(n) = 7n - 6, \\ f_3(n, V) = 8Vn - 8V + n - 1. \end{cases}$$

Оценим вычислительную сложность основной части алгоритма (фрагмент 4) в условиях принятой параметризации. Заметим, что число операций в одном

проходе цикла по  $b$  имеет порядок  $\Theta(1)$ . Цикл по нахождению максимума целевого функционала зависит от цикла по дискретам объёма, и их совместное рассмотрение приводит к следующим результатам (табл. 1).

Таблица 1

Значения счётчика цикла по дискретам объёма	Количество проходов цикла вычисления максимума
от 0 до $\bar{v}-1$	0
от $\bar{v}$ до $2\cdot\bar{v}-1$	1
...	...
от $(k-1)\cdot\bar{v}$ до $k\cdot\bar{v}-1$	$k-1$
при $k\cdot\bar{v} = V$	$k$

Таким образом, совокупное количество проходов внешнего (по  $v$ ) и внутреннего (по  $b$ ) циклов равно:

$$f_4(V, k) = V \cdot \Theta(1) + (1 \cdot \bar{v} + 2 \cdot \bar{v} + \dots + (k-1) \cdot \bar{v} + k) \cdot \Theta(1) = \Theta(1) \cdot \left( V + k + \bar{v} \cdot \sum_{i=1}^{k-1} i \right) = \Theta(1) \cdot \left( V + k + \bar{v} \cdot \frac{k \cdot (k-1)}{2} \right).$$

Подстановка  $k = V/\bar{v}$ , отражающая введённую параметризацию, даёт следующую оценку для двух внутренних циклов:

$$f_4(V, k) = \left( \frac{V^2}{2\bar{v}} + \frac{V}{2} + \frac{V}{\bar{v}} \right) \cdot \Theta(1),$$

и учитывая, что цикл по типам грузов выполняется  $n-1$  раз, окончательно получаем вычислительную сложность фрагмента 4:

$$f_4(n, V, k) = g(n, V, k) \cdot \Theta(1),$$

$$g(n, V, k) = (n-1) \cdot \left( \frac{V^2}{2\bar{v}} + \frac{V}{2} + k \right).$$

Трудоёмкость этого фрагмента может быть получена на основе подсчёта базовых операций в строках и полученной выше вычислительной сложности и составляет:

$$f_4(n, V, k) = 5nV^2k + 6V^2n + 10kn - 5V^2k - 6V - 10k.$$

Трудоёмкость блока *then* связана с оценкой среднего числа переписываний в блоке поиска максимума. Уделим этому блоку особое внимание. При входе в блок выполняется 2 операции. Остаётся понять, сколько раз будет осуществлён вход в блок переписывания максимальной стоимости. Математическое ожидание числа переписываний

определяется на основании анализа алгоритма поиска максимума [4], полученные результаты показаны в табл. 2.

Таблица 2

Значения счётчика цикла по дискретам объёма	Число проходов цикла вычисления максимума	Математическое ожидание числа переписываний
от 1 до $\bar{v}$	1	1
от $\bar{v} + 1$ до $2\cdot\bar{v}$	2	$H_2$
...	...	...
от $(k-1)\cdot\bar{v}+1$ до $k\bar{v}$	$k$	$H_k$
от $k\bar{v} + 1$ до $V$	$k+1$	$H_{k+1}$

Следует объяснить, почему мы здесь различаем значения  $k\cdot\bar{v}$  и  $V$ . Мы считаем, что все грузы имеют такой объём, что не более  $k$  грузов каждого типа может быть размещено в объёме  $V$ . Но фиксированное значение  $k$  обеспечивает не только значение среднего объёма, равное  $V/k$ , а целый сегмент значений

$$\bar{v} \in [1 + V/(k+1); V/k].$$

При таком рассмотрении средний объём грузов будет совпадать с серединой указанного сегмента:

$$\bar{v} = \frac{\left( \frac{V}{k+1} + 1 \right) + \frac{V}{k}}{2} = \frac{V(2k+1)}{2k(k+1)} + \frac{1}{2}.$$

Очевидно, что значения  $k\cdot\bar{v}$  и  $V$  будут отличаться на некоторое значение  $\Delta V$ , а именно:

$$\Delta V = V - k\bar{v} = \frac{1}{2} \left( \frac{V}{k+1} - k \right). \quad (2)$$

На основе результатов анализа, указанных в табл. 2, получим математическое ожидание числа выполнений блока *then* на полном проходе цикла по дискретам объёма – оно представляет собой сумму гармонических чисел:

$$\Delta V \cdot H_{k+1} + \sum_{l=1}^k \bar{v} \cdot H_l, \quad H_l = \sum_{i=1}^l \frac{1}{i}.$$

Поскольку при входе в блок выполняется две операции, и с учётом цикла по типам грузов, получим трудоёмкость блока *then* в следующем виде:

$$f_{then}(n, V, k) = 2(n-1) \cdot \left( \Delta V \cdot H_{k+1} + \sum_{l=1}^k \bar{v} \cdot H_l \right),$$

и, используя формулу для суммы гармонических чисел [5], введённую параметризацию, и подставляя выражение для  $\Delta V$  – формула (2), окончательно имеем:

$$f_{then}(n, V, k) = (n-1) \cdot \left[ \left( \frac{V}{k+1} - k \right) \cdot H_{k+1} + V(2k+1) \left( \frac{H_k}{k} - \frac{1}{k+1} \right) + H_k \cdot (k+1) - k \right].$$

Трудоёмкости фрагментов 5 и 6 легко вычисляются на основе числа базовых операций языка высокого уровня, приведённых в строках записи алгоритма:

$$f_5(n, V) = 5Vn - 5V, \quad f_6(n, V) = 4Vn^2 + Vn - 5V.$$

Для получения формулы трудоёмкости табличного алгоритма в среднем остаётся просуммировать трудоёмкости всех последовательно выполняющихся фрагментов. В результате получаем:

$$\begin{aligned} \overline{f_A}(n, V, k) = & 4n^2V + 5nVk + 20nV + 10nk + \\ & + (n-1) \cdot \left[ \left( \frac{V}{k+1} - k \right) \cdot H_{k+1} + V(2k+1) \cdot \left( \frac{H_k}{k} - \frac{1}{k+1} \right) + \right. \\ & \left. + H_k \cdot (k+1) - k \right] - 5Vk + 8n - 2V - 10k + 16. \end{aligned} \quad (3)$$

**Экспериментальное исследование программной реализации табличного алгоритма**

Для подтверждения теоретически полученного результата проведено экспериментальное исследование программной реализации табличного алгоритма. Эксперимент проводился в диапазоне значений

$$k = \overline{3, 10} \text{ и } n = 3, 6, 9 \text{ при } V = 1000.$$

Для каждой пары  $n, k$  проводилось 1000 экспериментов с генерацией различных значений объёмов грузов, обеспечивающих заданное  $k$  с последующим усреднением полученных значений трудоёмкости. Результаты проведённых экспериментов представлены в табл. 3 и на рис. 1–3.

Результаты анализа полученных экспериментальных данных в рассмотренном диапазоне значений  $k$  при фиксированных значениях  $V$  и  $n$  показывают незначительное расхождение между полученной теоретической функцией трудоёмкости, определяемой по формуле (3), и экспериментальными результатами. Так, на всём исследованном диапазоне значений параметров входа, максимальная разность не превышает 2%. Полученная теоретическая формула трудоёмкости может быть использована для прогнозирования времени выполнения табличного алгоритма и решения других задач, например, задачи определения области рационального использования табличного алгоритма в сравнении с другими точными алгоритмами решения задачи одномерной оптимальной по стоимости упаковки.

Таблица 3

$n$	$k$	Экспериментальная трудоёмкость в среднем	Теоретическая трудоёмкость	$\Delta\%$	
3	3	131 273,5	130 193,4	0,822786	
	4	141 928,8	140 802,9	0,793285	
	5	153 076,3	151 329,6	1,141065	
	6	163 999,6	161 794,4	1,344637	
	7	174 687,7	172 211,1	1,417730	
	8	184 924,0	182 589,4	1,262465	
	9	195 525,5	192 936,2	1,324277	
	10	207 361,7	203 256,6	1,979681	
	6	3	353 595,0	352 447,5	0,324524
		4	379 572,3	378 971,3	0,158336
5		406 148,6	405 287,9	0,211918	
6		433 038,3	431 450,0	0,366780	
7		459 196,9	457 491,9	0,371300	
8		485 252,9	483 437,6	0,374094	
9		510 992,5	509 304,5	0,330338	
10		538 431,1	535 105,5	0,617646	
9		3	647 176,0	646 701,6	0,073303
		4	688 704,9	689 139,6	0,063118
	5	730 495,2	731 246,3	0,102821	
	6	772 981,6	773 105,5	0,016029	
	7	814 236,6	814 772,6	0,065829	
	8	854 831,7	856 285,7	0,170092	
	9	896 490,6	897 672,7	0,131859	
	10	939 732,7	938 954,4	0,082821	

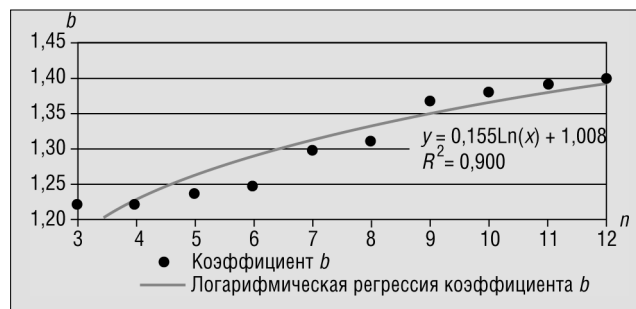


Рис. 1. Теоретическая и экспериментальная трудоёмкость табличного алгоритма при значениях  $V = 1000$  и  $n = 3$

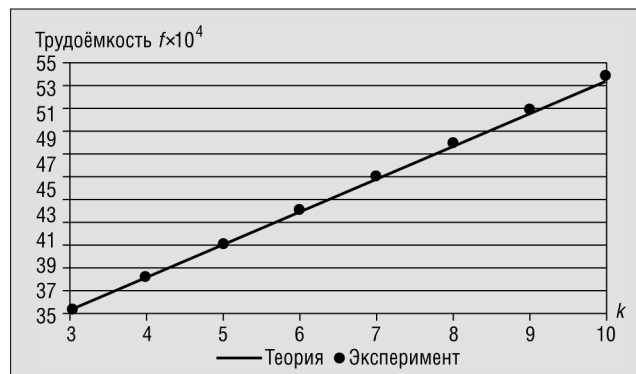


Рис. 2. Теоретическая и экспериментальная трудоёмкость табличного алгоритма при значениях  $V = 1000$  и  $n = 3$

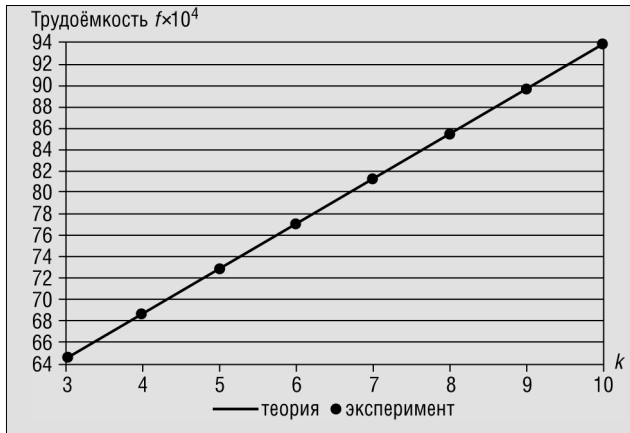


Рис. 3. Теоретическая и экспериментальная трудоёмкость табличного алгоритма при значениях  $V = 1000$  и  $n = 9$

**Методика прогнозирования времени выполнения**

Подход к прогнозированию времени выполнения программной реализации алгоритма на основе функции трудоёмкости как функции длины входа предложен одним из авторов в [6]. Для прогнозирования времени выполнения необходимо провести серию экспериментов на различных длинах входа алгоритма с целью определения среднего времени выполнения программы и на основе известной функции трудоёмкости определить среднее время на одну базовую операцию:

$$\bar{t}_{оп} = \bar{t}_э(n) / \bar{f}_A(n). \tag{4}$$

Однако полученное таким образом среднее время на базовую операцию не учитывает частотную встречаемость операций фрагментов алгоритма, определяющих главный асимптотический порядок функции трудоёмкости, что, как показано в [6] влияет на  $\bar{t}_{оп}$ . Варианты характерного поведения  $\bar{t}_{оп}$ , как функции длины входа, показаны на рис. 4 (подробнее см. [6]).

Если зависимость  $\bar{t}_{оп}(n)$  слабо выражена, то прогнозирование времени для интересующей размерности входа  $n^*$  может быть выполнено по формуле:

$$\bar{t}_э(n^*) = \bar{t}_{оп} \cdot \bar{f}_A(n^*). \tag{5}$$

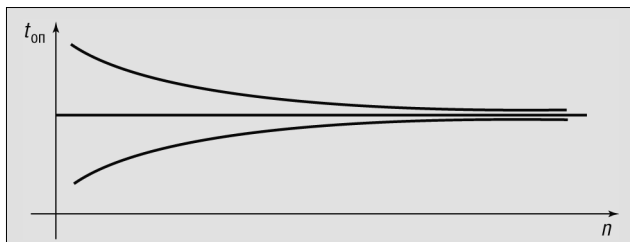


Рис. 4. Зависимость среднего времени на операцию от длины входа

в противном случае на основе экспериментальных данных методом, изложенным в [6], может быть построена функциональная зависимость  $\bar{t}_{оп}(n)$ , и прогноз времени выполнения рассчитывается по формуле:

$$\bar{t}_э(n^*) = \bar{t}_{оп}(n^*) \cdot \bar{f}_A(n^*). \tag{6}$$

Для алгоритма, рассматриваемого в настоящей статье, задача прогнозирования усложняется в связи с параметризацией, приводящей к функции трудоёмкости табличного алгоритма (3) как функции от трёх аргументов. Расширению подхода, предложенного в [6] на программные реализации алгоритмов, обладающих параметрическими аргументами в функции трудоёмкости, и посвящена данная часть статьи.

Первый этап предлагаемой методики – проведение серии экспериментов с программной реализацией алгоритма для получения экспериментальной зависимости среднего времени на базовую операцию в виде однопараметрических зависимостей при других фиксированных аргументах. Программная реализация алгоритма выполнена в среде Borland Delphi 7 на компьютере AMD Athlon 64 X2 4600+ (2,4), ASUS A8N-VM CSM, 2 Гб ОЗУ под управлением Windows XP SP2. Время выполнения программной реализации алгоритма определялось при помощи тактового счётчика и усреднялось по 10000 экспериментам. Среднее время выполнения обобщённой базовой операции определялось по формуле

$$\bar{t}_{оп}(n, V, k) = \bar{t}_э(n, V, k) / \bar{f}_A(n, V, k),$$

где  $\bar{f}_A(n, V, k)$  вычислялось по формуле (3).

При проведении эксперимента задействовалось лишь одно ядро процессора. Поэтому при переводе тактовых значений времени в наносекунды правильнее будет считать частоту процессора как частоту одного ядра. В табл. 4 приведены полученные экспериментальные результаты для значений

$$k = \sqrt{9,25} \text{ при } V = 500 \text{ и } n = 3.$$

Графически результаты этого эксперимента представлены на рис. 5.

Полученные данные показывают, что среднее время на операцию уменьшается с ростом  $k$ . Эту зависимость можно объяснить ростом частотной встречаемости более быстрых операций при увеличении значений  $k$  во фрагменте алгоритма, определяющего главный асимптотический порядок функции трудоёмкости (фрагмент 4).

Таблица 4

$k$	$\bar{t}_{оп}$	$k$	$\bar{t}_{оп}$
9	2,063	18	1,692
10	2,017	19	1,671
11	1,964	20	1,664
12	1,913	21	1,612
13	1,875	22	1,592
14	1,829	23	1,570
15	1,799	24	1,570
16	1,767	25	1,550
17	1,738		

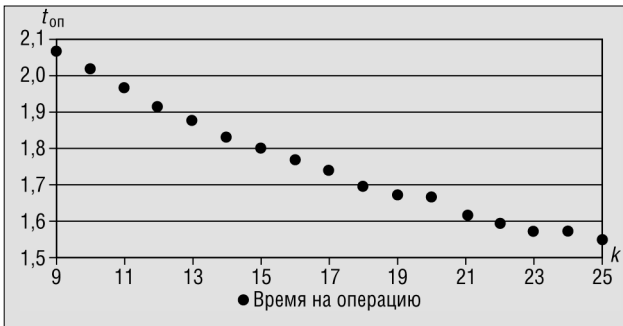


Рис. 5. Зависимость среднего времени на операцию от значения  $k$  при  $V = 500, n = 3$

Проведена серия экспериментов для выявления зависимости  $\bar{t}_{оп}$  от параметра  $n$ . Экспериментальные данные при  $V = 500$  и  $k = 14$  приведены в табл. 5, и показаны на рис. 6. Зависимость от параметра  $n$  более слабая и имеет ступенчатую форму. Можно предположить, что такая зависимость — следствие возрастающего объёма используемой памяти одновременно с ростом  $n$ , и особенностями использования кэш-памяти.

Таблица 5

$n$	$\bar{t}_{оп}$	$n$	$\bar{t}_{оп}$
3	1,829	8	1,860
4	1,796	9	1,978
5	1,791	10	1,954
6	1,770	11	1,962
7	1,866	12	1,967

По данным табл. 6 и 7 вычислено усреднённое время на обобщённую базовую операцию в диапазоне изменения

$$n = \overline{3,12} \text{ и } k = \overline{9,25},$$

которое составило  $\bar{t}_{оп} = 1,816$  такта. Этот результат может быть использован для упрощённого прогноза на основе формулы (5).

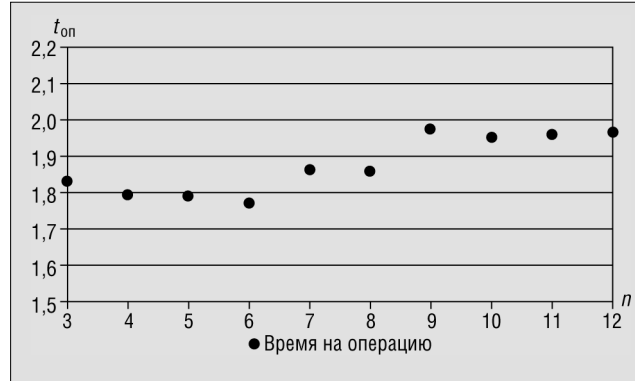


Рис. 6. Зависимость среднего времени на операцию от значения  $n$  при  $V = 500, k = 14$

Для получения более точных прогнозов на основе формулы (6) необходимо получить функциональную зависимость  $\bar{t}_{оп}$  от параметров функции трудоёмкости — это второй этап предлагаемой методики.

Для получения оценки зависимости среднего времени на обобщённую операцию от параметра  $k$  применим метод, предложенный в [6]. Значения  $n$  и  $V$  будем считать фиксированными. В этом случае, каждая из аддитивных компонент функции трудоёмкости по аргументу  $k$  может быть упорядочена по асимптотической иерархии. Рассмотрим два основных компонента:

$$\bar{f}_A(k) = \bar{f}_2(k) + \bar{f}_1(k) = 5nVk + 2VnH_k,$$

при этом время на операцию как функция  $k$  представимо в виде:

$$\bar{t}_{оп}(k) \approx \frac{\bar{t}_2 \cdot \bar{f}_2(k) + \bar{t}_1 \cdot \bar{f}_1(k)}{\bar{f}_2(k) + \bar{f}_1(k)} = \frac{\bar{t}_2 \cdot 5nVk + \bar{t}_1 \cdot 2VnH_k}{5nVk + 2VnH_k}.$$

Исходя из рассматриваемого диапазона значений  $k$ , будем считать значение гармонического числа  $H_k$  слабо меняющимся. Зафиксируем значение  $H_k \approx 3,39$ , и обозначим через:

$$\alpha = \bar{t}_1 / \bar{t}_2, \quad \bar{t}_1 = \alpha \cdot \bar{t}_2, \quad \beta = 6,78 \cdot nV / 5nV = 1,356,$$

В результате получим:

$$\begin{aligned} \bar{t}_{оп}(k) &= \bar{t}_2 \frac{k + \alpha \cdot \beta}{k + \beta} = \bar{t}_2 \left( \frac{k + \beta - \beta + \alpha \cdot \beta}{k + \beta} \right) = \\ &= \bar{t}_2 \left( 1 + \frac{\beta \cdot (\alpha - 1)}{k + \beta} \right) = \bar{t}_2 + \bar{t}_2 \cdot \beta \cdot (\alpha - 1) \frac{1}{k + \beta}. \end{aligned} \quad (7)$$

Выражение (7) можно преобразовать в линейную форму:

$$y = ax + b,$$

сделав замену переменных:

$$y = \bar{t}_{on}(k), \quad b = \bar{t}_2, \quad x = 1/(k + \beta), \quad a = \bar{t}_2 \cdot \beta \cdot (\alpha - 1).$$

Теперь, на основе экспериментальных данных методом наименьших квадратов определяются коэффициенты  $a$  и  $b$  уравнения линейной регрессии. Поскольку наблюдается зависимость значения среднего времени на операцию и от значения  $n$ , это не могло не отразиться на полученных методом наименьших квадратов коэффициентах  $a$  и  $b$ . На рис. 7 показаны экспериментально полученные значения  $\bar{t}_{on}(k)$  для исследуемого диапазона значений  $k$  при значениях  $n = 3$  и  $n = 6$ , а также полученные для этих значений  $n$  уравнения регрессии при введённой замене переменных.

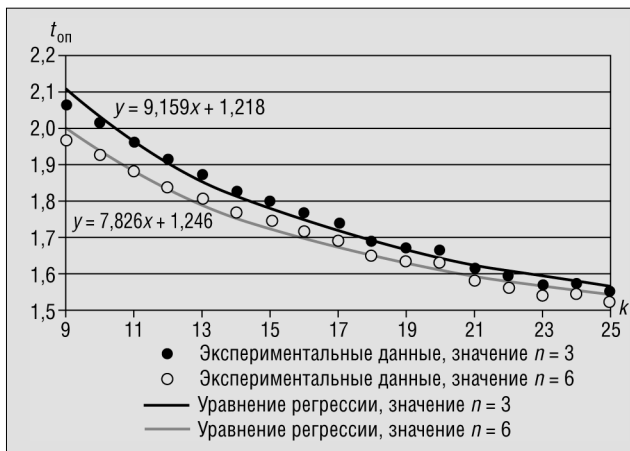


Рис. 7. Зависимость экспериментальных значений  $\bar{t}_{on}(k)$  и уравнения регрессии при  $n = 3$  (сверху) и  $n = 6$  (снизу)

Серия экспериментов для диапазона значений  $n = 3, \dots, 12$  и  $k = 9, \dots, 25$  позволила определить коэффициенты уравнения линейной регрессии  $a$  и  $b$  на всём диапазоне  $n = 3, \dots, 12$ , приведённые в табл. 6.

Таблица 6

$n$	$a$	$b$	$n$	$a$	$b$
3	9,160	1,218	8	8,163	1,312
4	8,637	1,220	9	9,074	1,368
5	8,328	1,235	10	8,513	1,382
6	7,826	1,247	11	8,415	1,393
7	8,473	1,299	12	8,314	1,401

Полученные результаты в виде графиков зависимости коэффициентов  $a$  и  $b$  от параметра показаны на рис. 8 и 9.

Эти результаты показывают: существует зависимость коэффициентов  $a$  и  $b$  от значения параметра  $n$ . Поэтому вполне логично (с целью повышения точности прогноза) построить уравнение регрессии для каждого из этих коэффициентов – это третий

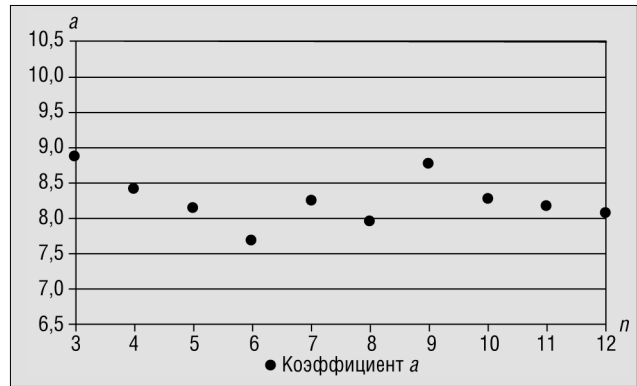


Рис. 8. Зависимость коэффициента регрессии  $a$  от параметра  $n$

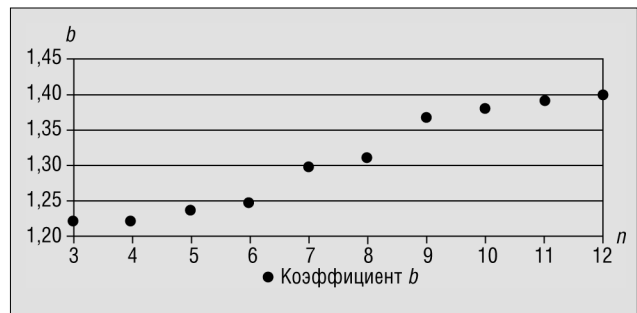


Рис. 9. Зависимость коэффициента регрессии  $b$  от параметра  $n$

этап предлагаемой методики – многопараметрическая зависимость среднего времени на операцию учитывается функцией регрессии коэффициентов.

Линейная регрессия для коэффициента  $a$  имеет следующий вид:

$$a(n) = -0,029n + 8,708,$$

что позволяет сделать вывод о незначительным влиянии  $n$  на коэффициент  $a$ , и коэффициент  $a$  в этом случае можно считать константой, равной среднему значению 8,49. Для коэффициента  $b$  на основе максимума достоверности было выбрано уравнение регрессии вида  $y = c \ln(n) + d$ , с достоверностью аппроксимации  $R^2 = 0,9$ . Для этого уравнения получены коэффициенты  $c = 0,155$  и  $d = 1,008$ . Полученное уравнение регрессии и экспериментальные точки показаны на рис. 10.

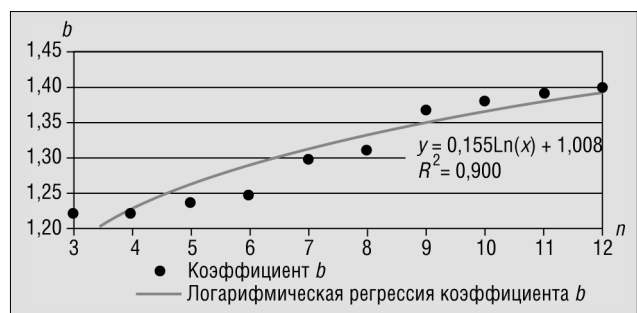


Рис. 10. Зависимость коэффициента  $b$  от параметра  $n$  и уравнение регрессии



В результате применения предложенной методики на основе обработки экспериментальных данных получено уравнение, связывающее среднее время выполнения обобщённой операции и параметры  $n$  и  $k$ :

$$t_{on}(n, k) = 8,49 \frac{1}{k + 1,356} + 0,155 \cdot \ln(n) + 1,008. \quad (8)$$

Результат позволяет прогнозировать время выполнения программной реализации табличного алгоритма упаковки, имеющего многопараметрическую функцию трудоёмкости по формуле, являющейся расширением формулы (6):

$$t_{вып}(V, n, k) = t_{on}(n, k) \cdot f_A(V, n, k). \quad (9)$$

### Прогнозирование и обсуждение экспериментальных результатов

С целью проверки полученных результатов на основе формулы (8) вычислены значения  $t_{оп}$  в диапазоне прогнозирования

$$n = \overline{13, 25}, \quad V = 500, \quad k = \overline{26, 40}$$

и проведено прогнозирование времени выполнения табличного алгоритма упаковки на основе полученных результатов по формуле (9). В сравнении аналогичное прогнозирование проведено с использованием усреднённого времени на базовую операцию —  $t_{оп} = 1,816$  такта.

На рис. 11 показаны результаты прогнозирования времени выполнения в диапазоне

$$n = \overline{13, 25}, \quad \text{при } k = 40, \quad V = 5$$



Рис. 11. Функции прогнозов времени выполнения в зависимости от значения параметра  $k$

Таким образом, при прогнозировании, основанном на усреднённом времени на операцию, полученное время выполнения с ростом  $k$  всё более расходитсся с экспериментальными данными. Прогноз,

основанный на уравнении регрессии, точнее. Это даёт основания предположить, что при бóльших, чем в рассматриваемом диапазоне, значениях  $k$ , точность прогнозирования при помощи аппарата регрессии будет выше относительно точности прогнозирования по усреднённому времени. На рис. 12 показаны результаты прогнозирования времени в диапазоне по

$$n = \overline{13, 25}, \quad \text{при } k = 40, \quad V = 500.$$

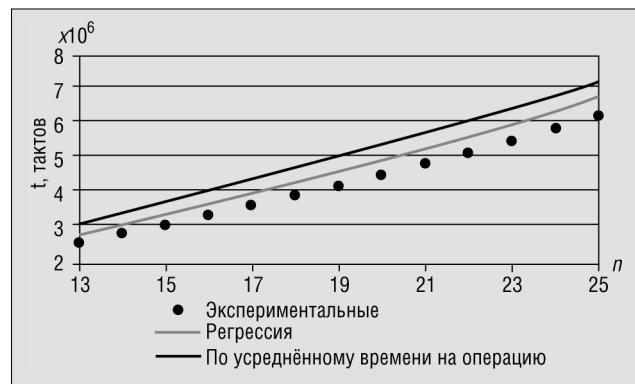


Рис. 12. Прогнозирование времени выполнения алгоритма в зависимости от параметра  $n$

Как видно из рис. 12, результаты прогнозирования, основанные на усреднённом времени на операцию, и при помощи функции среднего времени на операцию, полученной на основе аппарата регрессии идут практически параллельно, и растут практически с такой же скоростью, как и экспериментальные данные. Однако прогноз, сделанный на основе (9), ближе к экспериментальным данным.

В целом при использовании предложенной методики средняя погрешность прогнозирования составила 5,8%. Средняя погрешность прогноза на основе усреднённого времени на операцию — 12,3%. Таким образом, применение данной методики при прогнозировании времени выполнения программной реализации табличного алгоритма упаковки позволило увеличить точность прогнозирования более чем в 2 раза на исследованном диапазоне значений параметров.

Предварительные эксперименты по определению зависимости среднего времени на операцию от параметра показали, что эта зависимость имеет сложный характер, определяемый временем обращения процессора к иерархическим структурам памяти. Исследование и обоснование этой зависимости авторы рассматривают как самостоятельную задачу, представляющую собой развитие данной работы.

**Заключение**

На основе теоретического анализа получена функция трудоёмкости для табличного алгоритма решения классической задачи одномерной упаковки. Проведённые экспериментальные исследования показали, что полученная теоретически функция трудоёмкости даёт вполне приемлемые оценки и может быть использована для решения задачи прогнозирования времени выполнения.

В статье предложена методика прогнозирования времени выполнения для программных реализаций алгоритмов, имеющих многопараметрическую функцию трудоёмкости. На основе экспериментальных данных для программной реализации табличного алгоритма упаковки с использованием аппарата регрессии, в том числе и регрессии коэффициентов, получена функциональная зависимость среднего времени на обобщённую базовую операцию от параметров входа алгоритма. С использованием полученной функции трудоёмкости и функциональной зависимости среднего времени на базовую операцию получен прогноз времени выполнения, имеющий более чем в два раза меньшую погрешность, чем прогноз, основанный на усреднённом времени на базовую операцию. Полученные результаты могут

быть использованы при прогнозировании времени выполнения программных реализаций алгоритмов, для которых теоретически получены функции трудоёмкости в обобщённых базовых операциях.

В области экономики и бизнес-информатики полученные результаты могут найти применение для прогнозирования времени получения точного решения оптимизационных задач большой размерности, сводимых к задаче одномерной оптимальной по стоимости упаковки. В более широком смысле результаты могут быть использованы для группы задач оптимального планирования, допускающих решение методом динамического программирования и требующих исследования целевой функции по чувствительности к объёму распределяемого ресурса. Примером может служить задача составления оптимальных расписаний, как элемент общей задачи планирования бизнеса при значительном объёме исходных данных. В этом аспекте предложенный в статье подход к прогнозированию временных затрат на получение точного решения даёт возможность обоснования выбора точного алгоритма решения по сравнению с эвристическими алгоритмами на основе экономической оценки в параметрах «точность — временные затраты». ■

**Литература**

1. Беллман Р., Дрейфус Р. Прикладные задачи динамического программирования: Пер. с англ. — М.: Наука, 1965, — 457 с.
2. Тюрин Ю.И., Макаров А.А. Анализ данных на компьютере / Под ред. В.Э. Фигурнова — М.: ИНФРА-М, 2003. — 544 с.
3. Головешкин В. А., Ульянов М. В. Теория рекурсии для программистов. — М.: Издательство «Наука ФИЗМАТЛИТ», 2006. — 296 с.
4. Ульянов М. В. Классификация и методы сравнительного анализа вычислительных алгоритмов. Научное издание. — М.: Издательство физико-математической литературы, 2004. — 212 с.
5. Грэхем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики: Пер. с англ. — М.: Мир, 1998. — 703 с.
6. Ульянов М. В. Метод прогнозирования временных оценок программных реализаций алгоритмов на основе функции трудоёмкости // Информационные технологии. 2004. № 5. С. 54–62.



**ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ – ВЫСШАЯ ШКОЛА ЭКОНОМИКИ**

представляет свои периодические издания

**ВЕСТНИК МЕЖДУНАРОДНЫХ  
ОРГАНИЗАЦИЙ:**  
образование, наука, новая экономика  
**ИНФОРМАЦИОННО-АНАЛИТИЧЕСКИЙ ЖУРНАЛ**

Издается с 2006 г.

Главный редактор –  
Марина Владимировна Ларионова

Освещает деятельность ведущих международных организаций и объединений в области образования, науки, новой экономики, а также в области международной и социально-экономической политики, решения вопросов глобального развития. Содержит информацию о международных конференциях, форумах и семинарах, проектах и новых публикациях.

Каталог Агентства «Роспечать» – индекс 20054

Координаты редакции:  
101000 Москва, ул. Мясницкая, 20  
Тел.: (495) 621-4464, факс: (495) 621-8711  
E-mail: iori@hse.ru