РЕГУЛЯРНЫЙ ОБЩЕРОССИЙСКИЙ НАУЧНЫЙ СЕМИНАР

«ТЕОРЕТИЧЕСКАЯ И ПРИКЛАДНАЯ ИНФОРМАТИКА»

КЛАССИФИКАЦИЯ И МЕТОДЫ ПОСТРОЕНИЯ КОМБИНИРОВАННЫХ АЛГОРИТМОВ

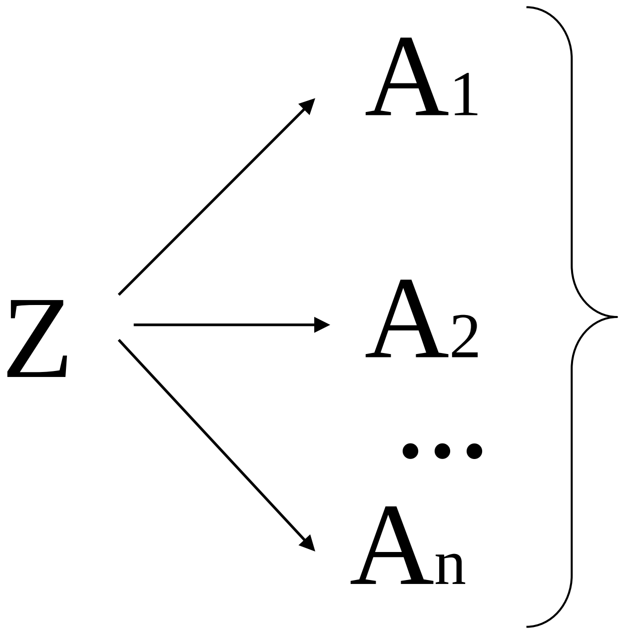
Михаил Васильевич Ульянов

материалы к докладу

12 ноября 2013 г.

ПОСТАНОВКА ЗАДАЧИ
И
ТЕРМИНОЛОГИЯ

Постановка задачи



Алгоритмическое разнообразие создает базу для разработки комбинированных алгоритмов алгоритмических комплексов

Задача: Разработка классификации и методов построения комбинированных алгоритмов на основе анализа ресурсной эффективности с учетом особенностей входов и областей применения

Терминология и обозначения

Пусть D – вход алгоритма, n – длина входа, $n=\mu(|D|)$, **тогда:**

$f_A(D)$ **- трудоёмкость алгоритма** — количество базовых операций в принятой модели вычислений, задаваемых алгоритмом на конкретном входе;

$f_A^{\wedge}(n)=\max_{D\in D_n}\{ f_A(D) \}$ **- худший случай** — наибольшее количество операций, задаваемых алгоритмом A на входах размерности n;

$f_A^{\vee}(n)=\min_{D\in D_n}\{ f_A(D) \}$ **- лучший случай** — наименьшее количество операций, задаваемых алгоритмом A на входах размерности n;

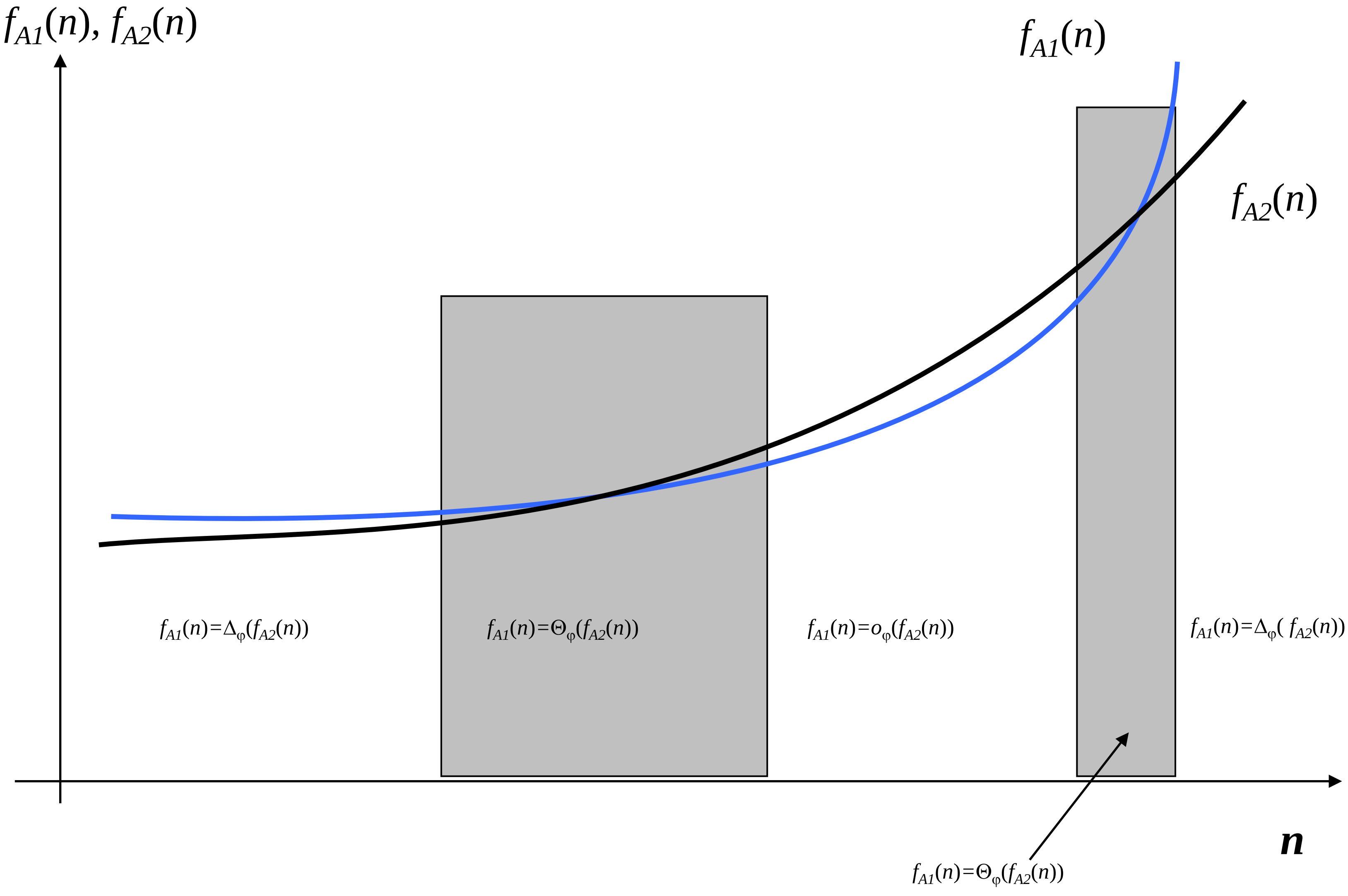
$\overline{f_A}(n)=\sum_{D\in D_n} p(D)\cdot f_A(D)$ **- средний случай** — среднее количество операций, задаваемых алгоритмом A на входах размерности n;

$V_A(D)$ - **функция объема памяти -** максимальное число ячеек памяти модели вычислений, задействованных в ходе выполнения алгоритма**;**

Оценка качества алгоритма

- Оценки ресурсных характеристик – функции трудоемкости и функции объема памяти.
- Оценки алгоритма в рамках специальных классификаций.
- Оценки по информационной чувствительности и временной устойчивости.
- Оценки по дополнительным критериям – точность, сходимость и т.д.

Возможный вид функций ресурсных характеристик алгоритмов



Комбинированные алгоритмические решения (алгоритмические комплексы)

Выбор алгоритма, рационального по выбранному критерию качества, на основе анализа входа в зависимости от:

- размерности решаемой задачи – длины входа;
- значения характерных параметров задачи

В этом случае задача решается ***одним алгоритмом***, выбранным на основе анализа текущего входа (адаптация к входу путем выбора наиболее рационального алгоритма).

Комбинированные алгоритмы

Создание алгоритма решения задачи **как комбинации нескольких алгоритмов** с различными ресурсными характеристиками, которые являются рациональными по выбранному критерию оценки для различных сегментов размерности задачи или других характерных параметров входа.

Общий пример – повышение ресурсной эффективности рекурсивных алгоритмов за счет сокращения дерева рекурсии путем замены нижних уровней дерева рекурсии предвычислениями, которые выполняются алгоритмом, более эффективным для малых длин входа.

ПОДХОДЫ К РАЗРАБОТКЕ КОМБИНИРОВАННЫХ АЛГОРИТМОВ

Стационарный подход

В процессе разработки комбинированного алгоритма, при теоретической оптимизации функции качества, получаемый оптимальный порог переключения между комбинируемыми алгоритмами **не зависит** от особенностей входа и является **стационарным** для всей расчетной области применения.

Комбинированный алгоритм не анализирует вход, т.е. является стационарно оптимальным на допустимом множестве входов.

Статически-адаптивный подход

В процессе разработки комбинированного алгоритма, при теоретической оптимизации функции качества, получаемый оптимальный порог переключения комбинируемых алгоритмов **зависит** от особенностей входа.

Оптимальный порог переключения **рассчитывается** по теоретической формуле для каждого входа **перед запуском** собственно комбинированного алгоритма, т.е. выполняется **статическая адаптация** к входу алгоритма.

Динамически-адаптивный подход

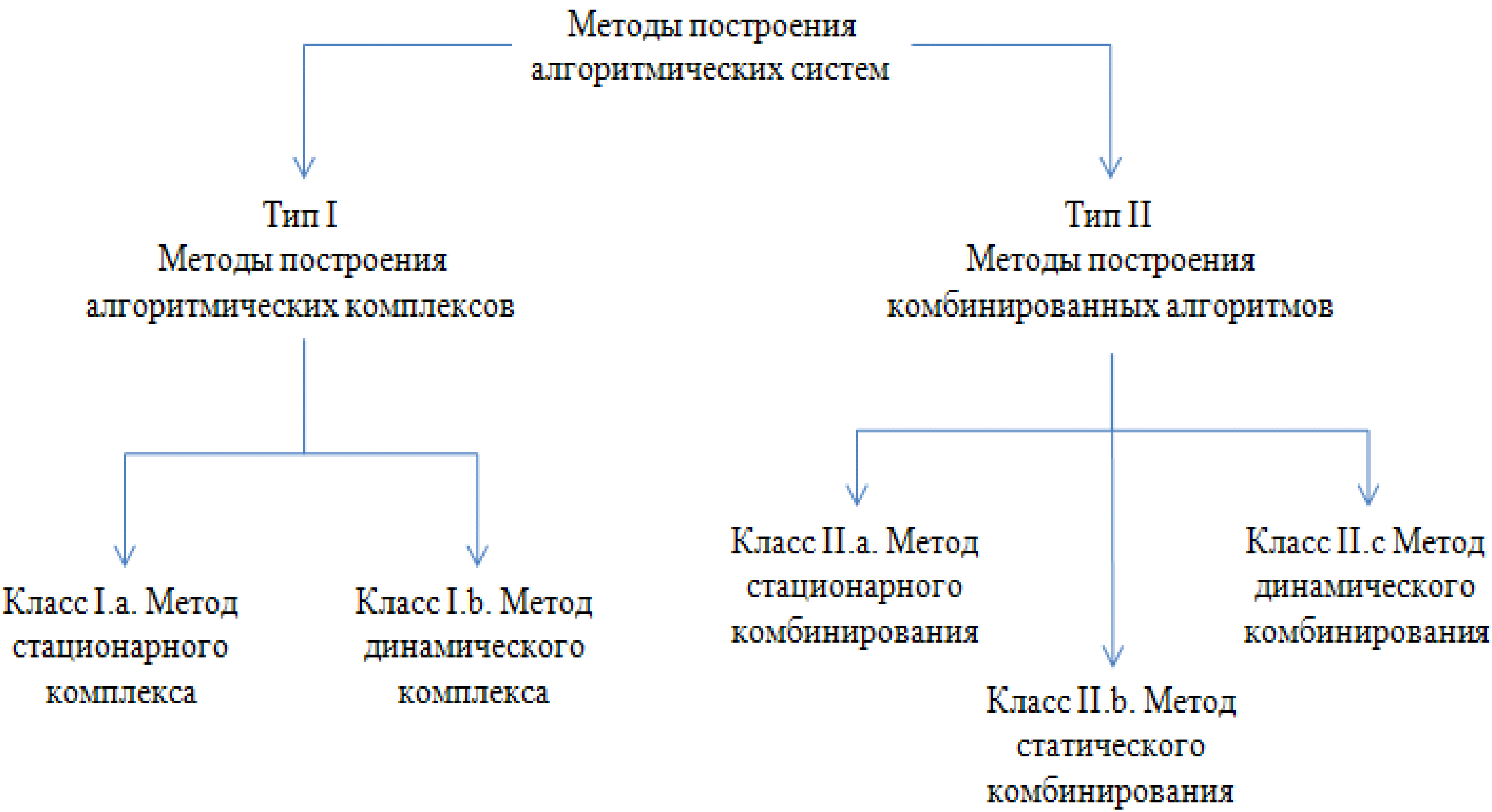
1. Оптимальный порог переключения между комбинируемыми алгоритмами определяется **в динамике** их выполнения на конкретном входе **на основе вычисления** значения специальной **пороговой функции** для которой известно критическое значение (*простая динамическая адаптация к входу*).
2. Оптимальный порог переключения определяется на основе совокупной адаптации к предыдущим входам (*динамическая адаптация с памятью входов*).

Типизация алгоритмических систем

АЛГОРИТМИЧЕСКИЕ СИСТЕМЫ

- Тип I. Алгоритмические комплексы
  - Класс I.a. Алгоритмические комплексы без памяти
    - Подкласс. Комплексы без памяти без выбора
    - Подкласс. Комплексы без памяти с выбором
  - Класс I.b. Алгоритмические комплексы с памятью
    - Подкласс. Простые комплексы с памятью
    - Подкласс. Адаптивные комплексы с памятью
- Тип II. Комбинированные алгоритмы
  - Класс II.a. Стационарно-адаптивные комбинированные алгоритмы
  - Класс II.b. Статически-адаптивные комбинированные алгоритмы
  - Класс II.c. Динамически-адаптивные комбинированные алгоритмы

Классификация методов построения алгоритмических систем

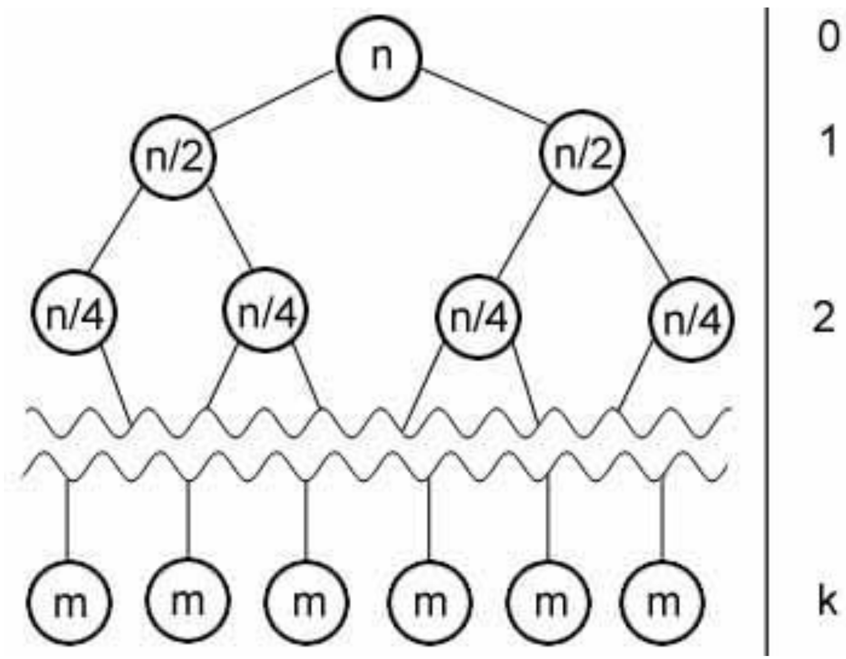


ПРИМЕРЫ КОМБИНИРОВАННЫХ АДАПТИВНЫХ АЛГОРИТМИЧЕСКИХ РЕШЕНИЙ

СТАЦИОНАРНЫЙ ПОДХОД

Сортировка слиянием в комбинации с сортировкой вставками

Схема комбинированного алгоритма



m – длина массива, при достижении которой происходит останов рекурсии и сортировка массива вставками;

k – уровень, на котором происходит останов рекурсии;

Теоретическое определение порога переключения (1)

Чтобы определить рациональный порог переключения, то есть оптимальную длину массива $m_{оптим}$, при достижении которой в ходе работы алгоритма происходит останов рекурсии, необходимо предварительно вычислить трудоемкость комбинированного алгоритма на входных данных длиной (n):

$$f_{комб}(n,m) = f_{рекурсии} + f_{вставки};$$

$$\frac{\partial f}{\partial m} = 0$$, откуда получим $m_{оптим}$

$$f_{рекурсии} = f_R(n) + f_{CL}(n) + f_{CV\ Msort}(n) + f_{CV\ Merge}(n) =$$

$$= 36\frac{n}{m} - 18 + 4\frac{n}{m} + 28\frac{n}{m} - 28 + 18 \cdot n \cdot \log_2\frac{n}{m} + 23\frac{n}{m} - 23 =$$

$$= 18 \cdot n \cdot \log_2\frac{n}{m} + (36 + 4 + 28 + 23)\frac{n}{m} - (18 + 28 + 23) = \underline{18 \cdot n \cdot \log_2\frac{n}{m} + 91\frac{n}{m} - 69}$$ где

$f_R(n)$ - трудоемкость вызова рекурсии; $f_{CL}(n)$ - трудоемкость останова рекурсии;

$f_{CV\ Merge}(n)$, $f_{CV\ Msort}(n)$ - трудоемкость внутренних вершин процедур Merge и MSort;

Теоретическое определение порога переключения (2)

$$f_{вставки}(n) = R_L(n)\cdot f_{вставки}(1) = \frac{n}{m}\cdot\left(2.5m^2 + 11.5m - 10\right) = \underline{2.5nm + 11.5n - 10\frac{n}{m}}$$, где

$R_L(n)$ - число листьев дерева рекурсии; $R_L(n) = 2^k = \frac{n}{m}$

Таким образом, функция трудоемкости комбинированного алгоритма имеет вид:

$$f(n,m) = f_{рекурсии} + f_{вставки} = 18\cdot n\cdot\log_2\frac{n}{m} + 91\frac{n}{m} - 69 + 2.5nm + 11.5n - 10\frac{n}{m} =$$

$$\underline{= 18\cdot n\cdot\log_2\frac{n}{m} + 2.5nm + 11.5n + 81\frac{n}{m} - 69}$$

Продифференцировав по (m), получим $m_{оптим}$:

$$\frac{\partial f}{\partial m} = -\frac{18n}{m\ln 2} + 2.5n - \frac{81n}{m^2}$$

$$\frac{\partial f}{\partial m} = 0 \Rightarrow -\frac{18n}{m\ln 2} + 2.5n - \frac{81n}{m^2} = 0 \Big| \cdot m^2$$

$$-\frac{18nm}{\ln 2} + 2.5nm^2 - 81n = 0 \Big| : n$$

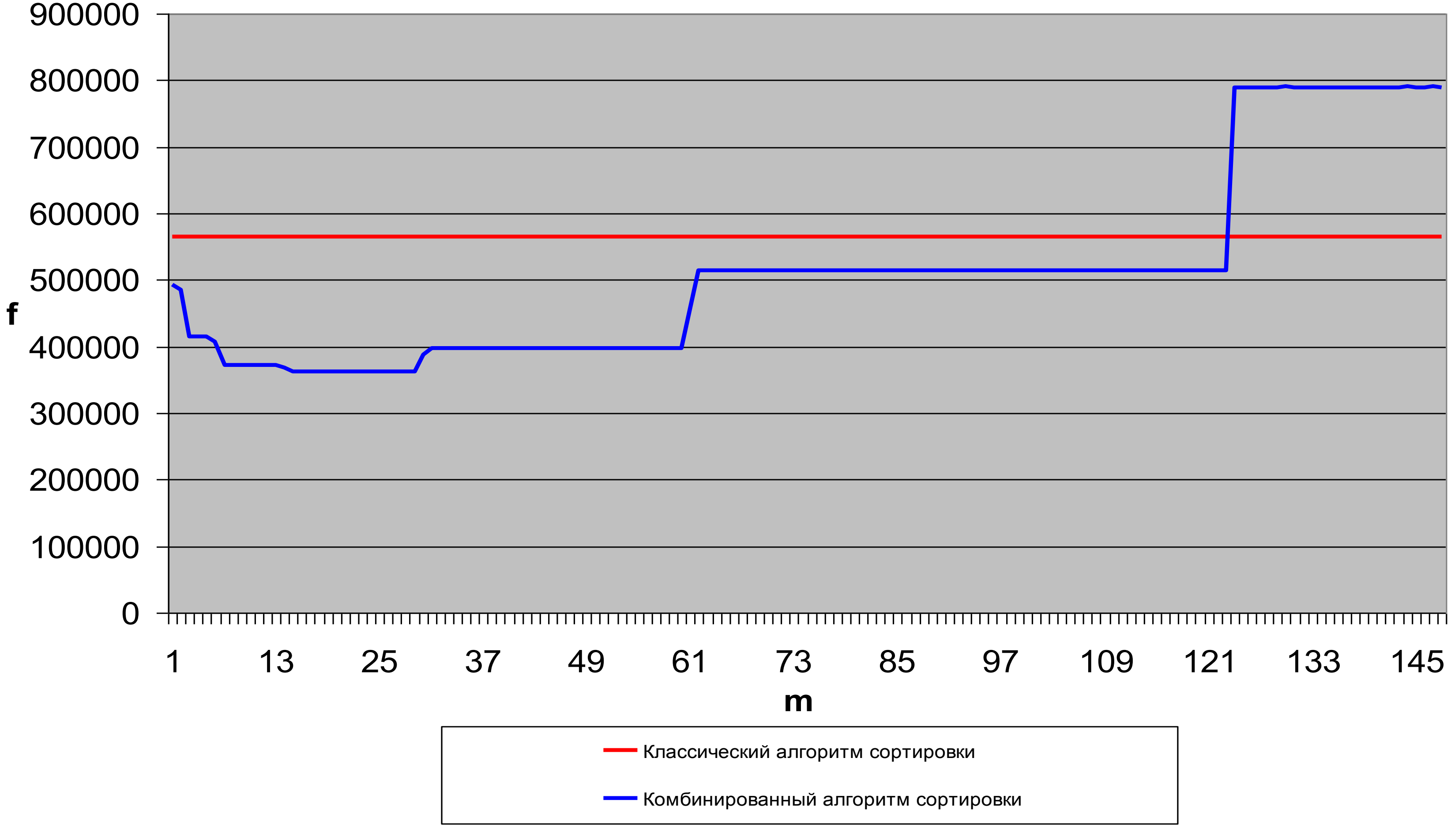
$$2.5m^2 - \frac{18m}{\ln 2} - 81 = 0$$

$$\frac{D}{4} = \frac{81}{(\ln 2)^2} - (-81\cdot 2.5) = \frac{81 + 202.5\ln^2 2}{\ln^2 2} = \left(\frac{13.35}{\ln 2}\right)^2$$

$$m = \frac{\frac{9}{\ln 2} + \frac{13.35}{\ln 2}}{2.5} = \frac{22.35}{2.5\ln 2} \approx \underline{13}$$

Экспериментальные результаты

Зависимость трудоемкости f от порога переключения m по результатам экпериментов с массивом из 2000 элементов



Снижение трудоемкости

Комбинированный алгоритм				Классическая сортировка слиянием				
n	*f* эксп.	*f* теор.	Δ, %	n	*f* эксп.	*f* теор.	Δ, %	Снижение трудоемкости, %
1000	167945	162938	2,98	1000	264488	264318	0,06	36,5
2000	371924	361945	2,68	2000	565043	564702	0,06	34,2
3000	574311	574540	0,04	3000	882867	878674	0,47	34,9
4000	815930	795959	2,45	4000	1202147	1201470	0,06	32,1
5000	1031411	1023939	0,72	5000	1536973	1530828	0,40	32,9
6000	1256688	1257149	0,04	6000	1873794	1865415	0,45	32,9
7000	1549763	1494707	3,55	7000	2210937	2204350	0,30	29,9
8000	1775971	1735987	2,25	8000	2548369	2547007	0,05	30,3
9000	2007022	1980522	1,32	9000	2899480	2892919	0,23	30,8
10000	2242972	2227948	0,67	10000	3254015	3241722	0,38	31,1

Значение $m=13$ определено на основе теоретического анализа, не изменяется для различных входов, и является ***стационарным***, что дает сокращение трудоемкости ≈100n.

СТАТИЧЕСКИ-АДАПТИВНЫЙ ПОДХОД

КОМБИНИРОВАННЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ УПАКОВКИ

Постановка задачи упаковки

Пусть задано множество типов грузов

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

где каждый элемент, соотнесенный с типом груза, обладает целочисленным линейным размером или «объемом», и ценовой характеристикой , которая содержательно отражает практически значимые предпочтения для загрузки объектов данного типа.

Максимизировать

$$P_n(\mathbf{x}) = \sum_{i=1}^{n} C_i(x_i) = \sum_{i=1}^{n} x_i \cdot c_i \to \max \qquad \textbf{при} \qquad \sum_{i=1}^{n} x_i \cdot v_i \le V$$

Функциональное уравнение Беллмана для задачи упаковки

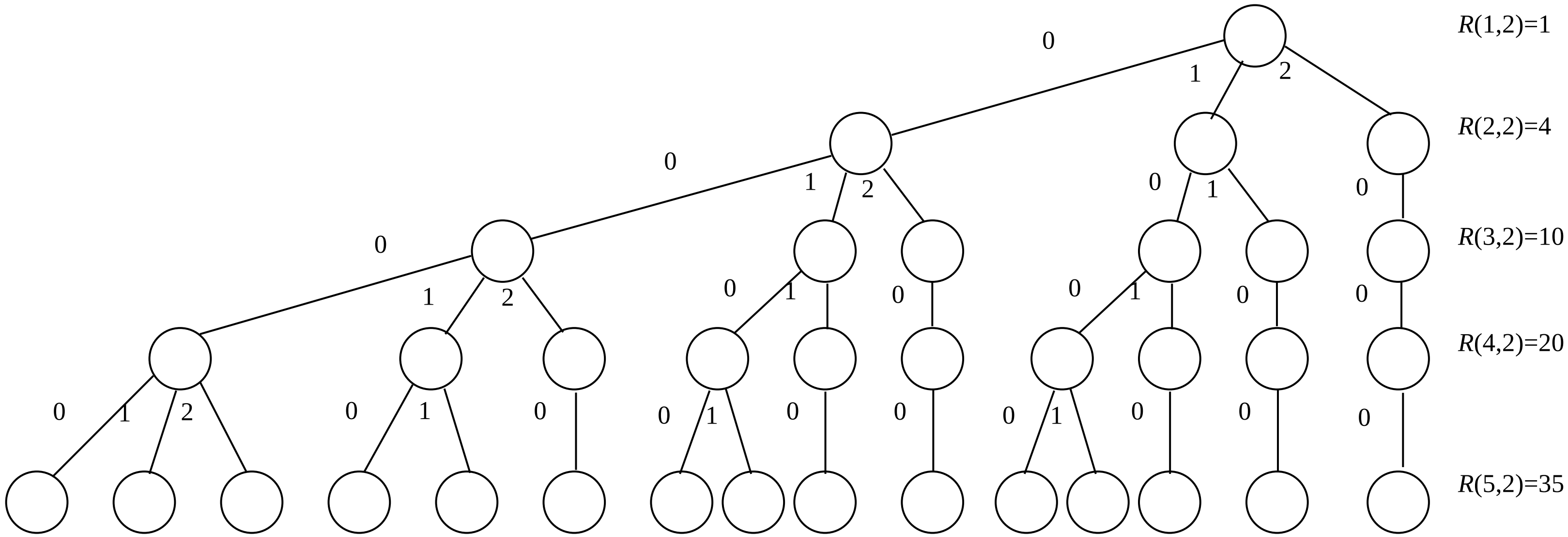
$$\begin{cases} f_1(v) = \left\lfloor \dfrac{v}{v_k} \right\rfloor \cdot c_1 \, , m = 1; \\ f_m(v) = \max\limits_{x_m} \{ x_m \cdot c_m + f_{m-1}(v - x_m \cdot v_m) \}, m \geq 2, \, x_m = 0,1,\ldots,\left\lfloor \dfrac{v}{v_m} \right\rfloor. \end{cases}$$

Параметризация задачи

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

Рекурсивный алгоритм

Непосредственная рекурсивная реализация функционального уравнения Беллмана



Общее число вершин дерева рекурсии

$$R(n,k)=C_{n+k}^{n-1}$$

Трудоемкость рекурсивного алгоритма

Трудоемкость данного алгоритма в базовых операциях процедурного языка высокого уровня определена методом подсчета вершин порожденного дерева рекурсии

$$f_{A1}(n,k) = \left(6 \cdot n + 9 \cdot n \cdot \alpha_f\right) \cdot R(n,k) + \left(46 + 5 \cdot \alpha_V + 7 \cdot \alpha_f\right) \cdot R(n,k) - 17$$

где

$$R(n,k) = \frac{\Gamma(n+k+1)}{\Gamma(n) \cdot \Gamma(k+2)}$$

$$\alpha_V = \frac{n-1}{n+k}, \qquad \alpha_f = 1 - \frac{k+1}{2 \cdot (n+k)}$$

Идея табличного алгоритма

Пример для трех типов грузов и объема 10

i	v_i	c_i
1	2	3
2	3	5
3	4	7

Таблица оптимальной упаковки

v	x_1	$f_1(v)$		v	x_1	x_2	$f_2(v)$		v	x_1	x_2	x_3	$f_3(v)$
1	0	0		1	0	0	0		1	0	0	0	0
2	1	3		2	1	0	3		2	1	0	0	3
3	1	3		3	0	1	5		3	0	1	0	5
4	2	6		4	2	0	6		4	0	0	1	7
5	2	6	$\Rightarrow$	5	1	1	8	$\Rightarrow$	5	1	1	0	8
6	3	9		6	0	2	10		6	0	2	0	10
7	3	9		7	2	1	11		7	0	1	1	12
8	4	12		8	1	2	13		8	0	0	2	14
9	4	12		9	0	3	15		9	0	3	0	15
10	5	15		10	2	2	16		10	?	?	?	?

Трудоемкость табличного алгоритма

Детальный анализ табличного алгоритма позволяет получить его функцию трудоемкости в выбранной модели вычислений с учетом введенной параметризации. Окончательный результат имеет следующий вид:

$$f_{A2}(n,V,k) = 4n^2V + 5nVk + 20nV + 10nk + 2Vn\ln k - 2{,}86Vn - 2V\ln k + 2{,}86V - 5Vk + 8n - 2V - 10k + 16.$$

Сравнительный анализ трудоемкости алгоритмов

Вид функций трудоемкости (для табличного алгоритма V=const) (рис.1)
и кривая разграничения рационального выбора (рис.2).

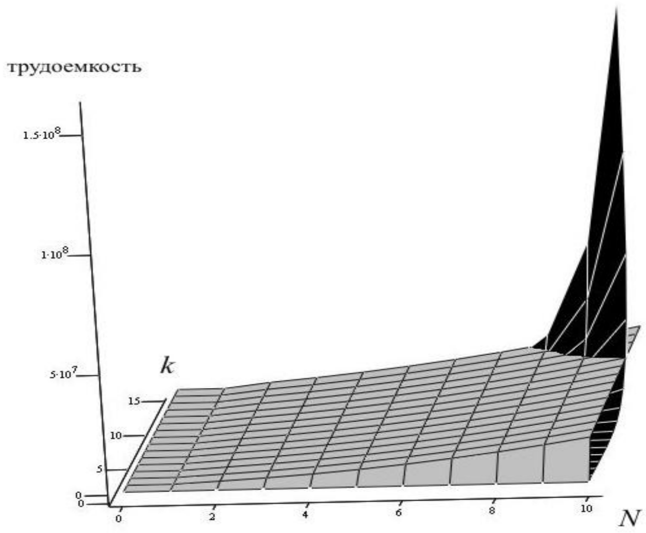


Рис.1

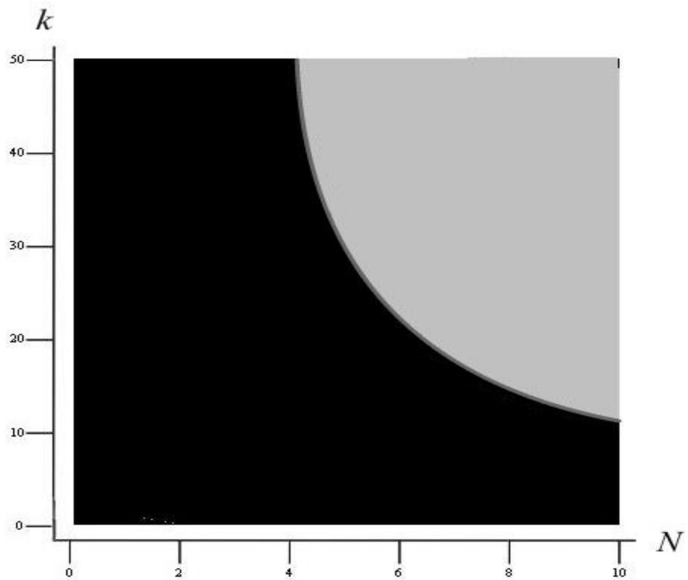


Рис.2

Построение комбинированного алгоритма

Функция $g(m)$, значением которой является совокупная трудоемкость комбинированного алгоритма

$$g(m) = f_{A1}(n-m,k) + f_{A2}(m,V,k)$$

Оптимальный по трудоемкости порог переключения

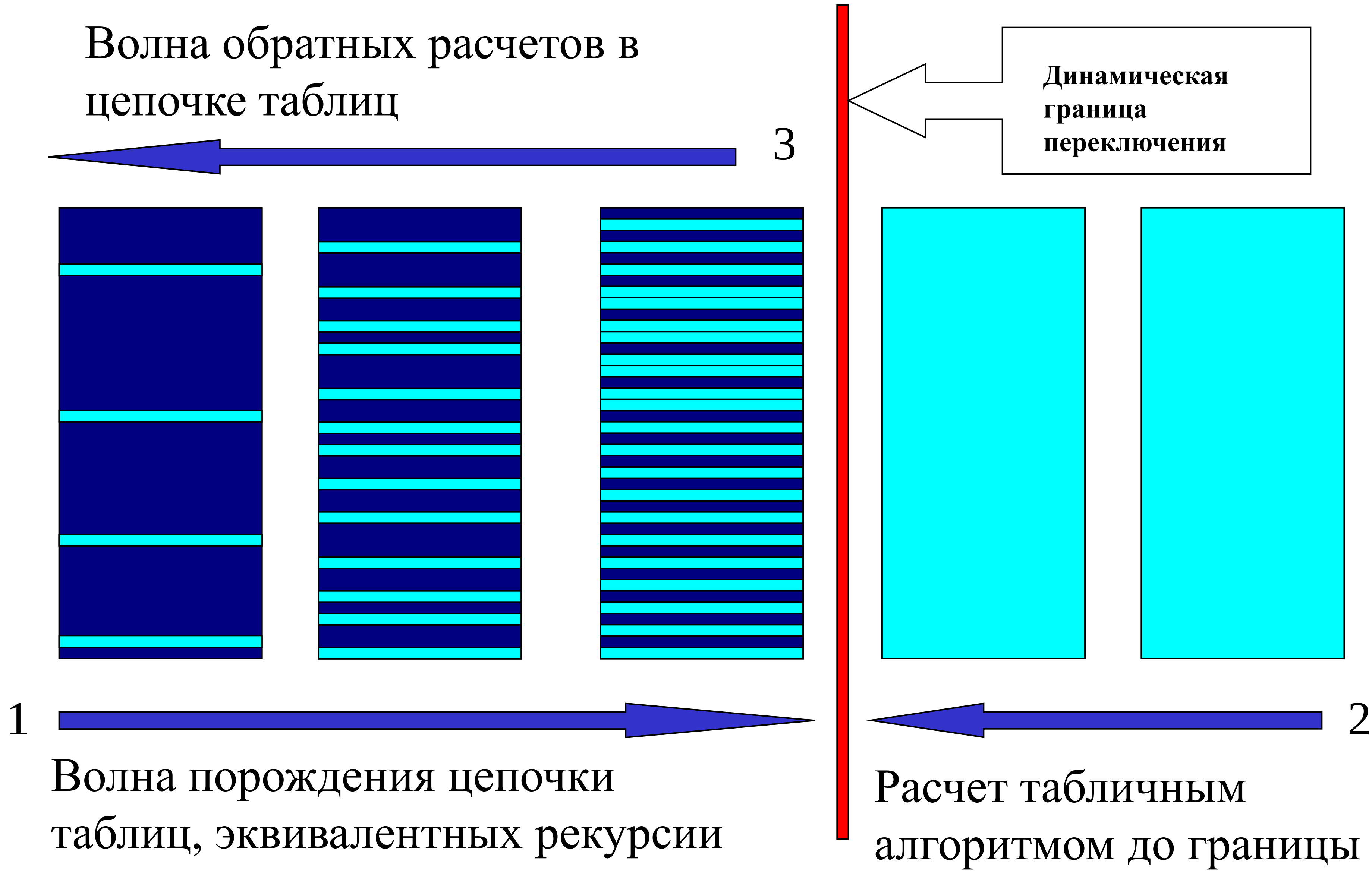
$$m^* = \arg\min_{0 \le m \le n} g(m) \qquad m^* = m^*(n,V,k)$$

Вычисления производятся при каждом входе в комбинированный алгоритм, но до начала его выполнения, т.е. значение m^* является ***статическим*** (постоянным) в момент запуска алгоритма.

ДИНАМИЧЕСКИ- АДАПТИВНЫЙ ПОДХОД

ВОЛНОВОЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ УПАКОВКИ

Идея алгоритма



Критерий переключения

Динамически для каждой таблицы , на **этапе 1** вычисляется число заполненных строк – $\boldsymbol{m}$, и определяется значение пороговой функции $\boldsymbol{\mu}$, значением которой является коэффициент заполнения текущей таблицы:

$$\boldsymbol{\mu(m,\ V)=\ m/V.}$$

Критическое значение $\boldsymbol{\mu = 0.5}$, и при $\boldsymbol{\mu \geq 0.5}$ происходит переключение на табличный алгоритм.

Библиографический список

1. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы. Разработка и анализ. — М.: «ФИЗМАТЛИТ», 2008. – 304 с.

2. Ульянов М.В., Наумова О.А. Комбинированный и волновой алгоритмы решения задачи упаковки: принципы построения и особенности // Бизнес информатика. 2009. №2(8). С. 27–33.

3. Ульянов М.В., Наумова О.А. Классификация методов построения алгоритмических систем // Вычислительные технологии. 2011. Т. 16, № 1. С. 105–118.

БЛАГОДАРЮ ЗА ВНИМАНИЕ !