Санкт-Петербургский государственный университет

ТАМАСЯН Григорий Шаликович

ПРОСОЛУПОВ Евгений Викторович

АНАЛИЗ АЛГОРИТМОВ ПРОЕКТИРОВАНИЯ ТОЧКИ НА СТАНДАРТНЫЙ СИМПЛЕКС

III международная конференция
«УСТОЙЧИВОСТЬ И ПРОЦЕССЫ УПРАВЛЕНИЯ»
посвященная 85-летию со дня рождения профессора,
чл.-корр. РАН В. И. Зубова

г. Санкт-Петербург
5 – 9 октября 2015 г.

СОДЕРЖАНИЕ ДОКЛАДА

- Постановка задачи
- Краткая история вопроса
- Алгоритм Малозёмова – Певного
- Алгоритм Maculan – de Paula
- Численные эксперименты

ПОСТАНОВКА ЗАДАЧИ

Задача ортогонального проектирования точки $c = (c_1, \ldots, c_n)$ на стандартный симплекс $\Lambda \subset \mathbb{R}^n$, определяемый условиями

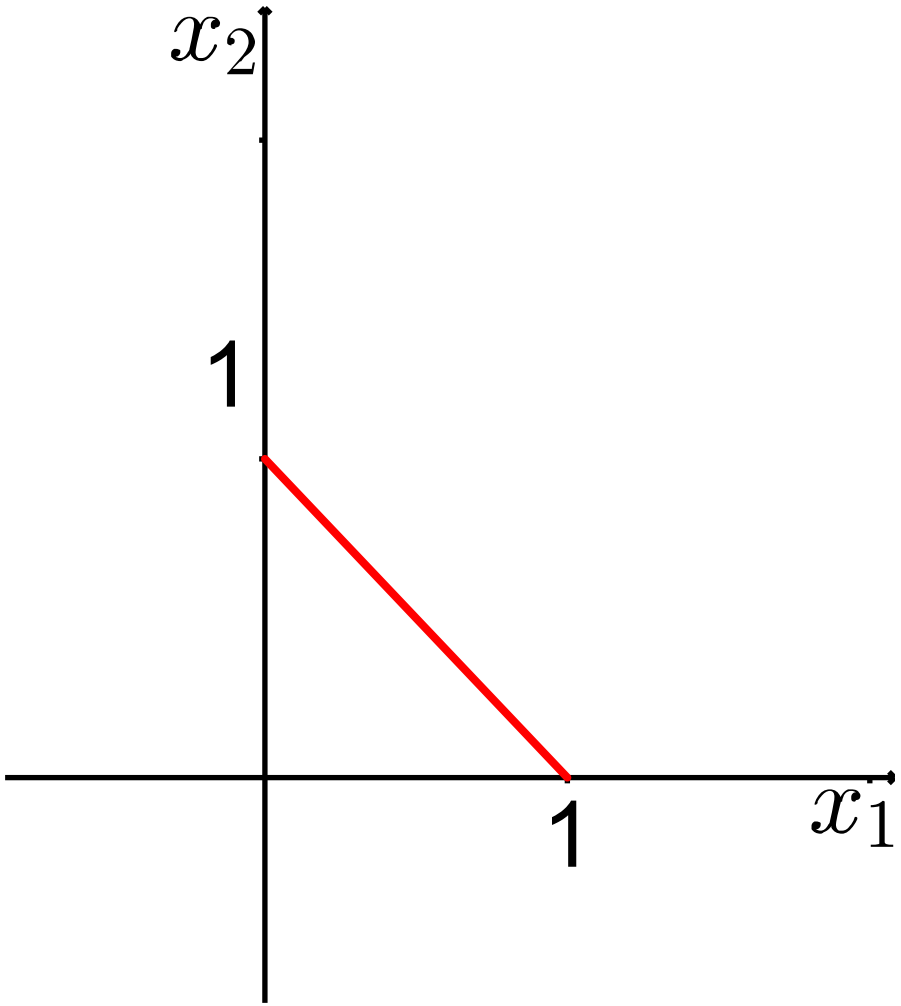
$$\sum_{i=1}^{n} x_i = 1; \quad x_i \geqslant 0,\ i \in 1:n,$$

ставится следующим образом:

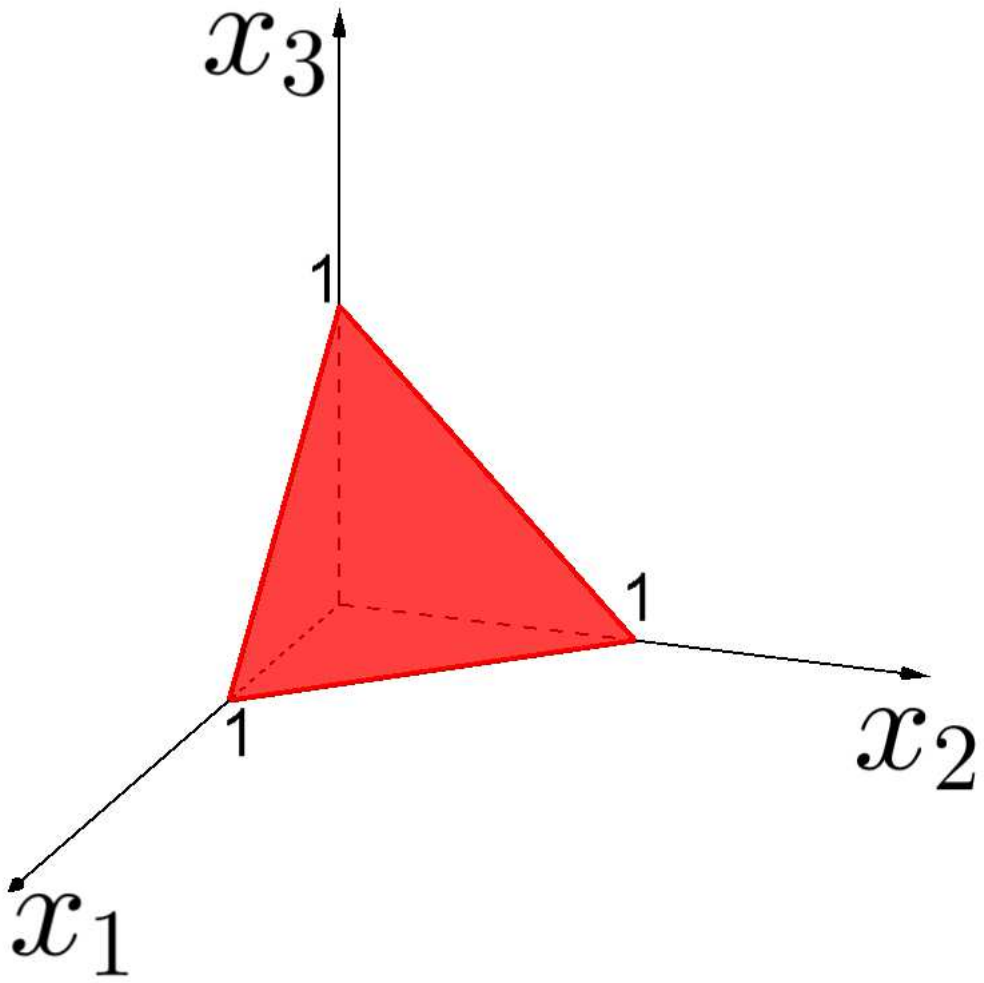
$$Q(x) := \tfrac{1}{2} \sum_{i=1}^{n} (x_i - c_i)^2 \to \min_{x \in \Lambda}. \tag{1}$$

Решение этой задачи существует и единственно. Обозначим его x^*.

СТАНДАРТНЫЙ СИМПЛЕКС



$n = 2$



$n = 3$

КРАТКАЯ ИСТОРИЯ ВОПРОСА

1. Held M., Wolfe P., Crowder H. P. Validation of the subgradient optimization // Math. Programming. 6, pp. 62–88 (1974).

2. Brucker P. An $O(n)$ algorithm for quadratic knapsack problems // Operations Research Letters 3 (1984) 163–166.

3. Maculan, N., de Paula Jr., G.G.: A linear-time median-finding algorithm for projecting a vector on the simplex of $\mathbb{R}^n$ // Operations Research Letters, 8 (4), pp. 219–222 (1989).

4. Малоземов В. Н., Певный А. Б. *Быстрый алгоритм проектирования точки на симплекс* // Вестник СПбГУ. Сер. 1. 1992. Вып. 1 (No 1). С. 112–113.

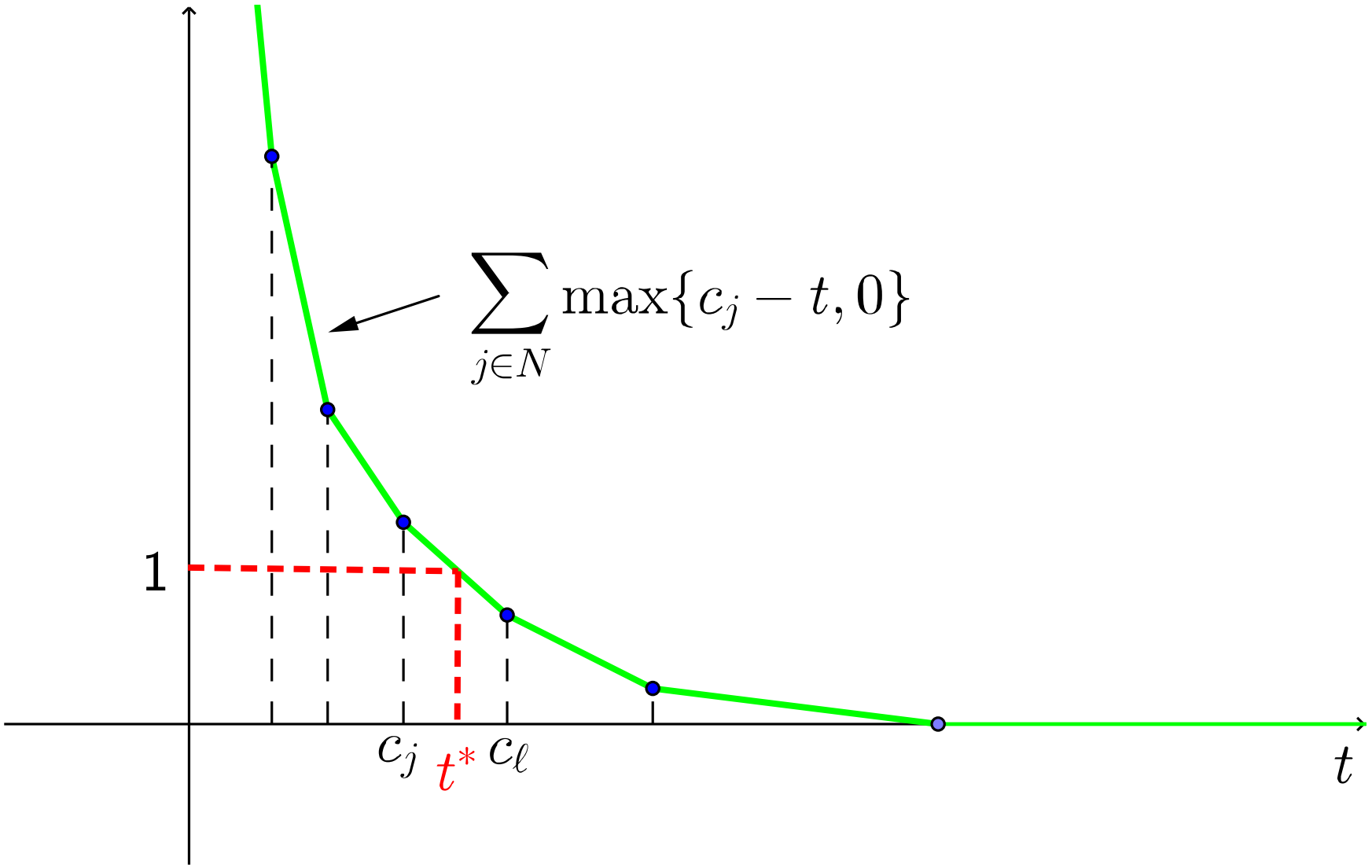
5. Michelot C. *A finite algorithm for finding the projection of a point onto the canonical simplex of* $\mathbb{R}^n$ // JOTA. 1986. Vol. 50. No 1. P. 195—200.

6. Causa A., Raciti F. *A purely geometric approach to the problem of computing the projection of a point on a simplex* //JOTA. 2013. Vol. 156. No 2. P. 524—528.

7. Patriksson M. A survey on the continuous nonlinear resource allocation problem. Eur. J. Oper. Res, Vol. 185, No. 1 (Feb., 2008), pp. 1–46.

8. Малоземов В. Н., Тамасян Г. Ш. Ещё один быстрый алгоритм проектирование точки на стандартный симплекс // Семинар «DHA & CAGD». Избранные доклады. 5 сентября 2013 г. (`http://dha.spb.ru/reps13.shtml#0905`)

9. Тамасян Г. Ш. Сравнительное изучение двух быстрых алгоритмов проектирования точки на стандартный симплекс // Семинар «CNSA & NDO». Избранные доклады. 15 мая 2014 г. (`http://www.apmath.spbu.ru/cnsa/reps14.shtml#0515`)


$\sum_{j \in N} \max\{c_j - t, 0\}$
1
c_j
t^*
c_ℓ
t

АЛГОРИТМ МАЛОЗЁМОВА – ПЕВНОГО

1) Меняем знаки у компонент c_j точки c и числа $\{-c_j\}$ упорядочиваем по неубыванию. Получаем последовательность $a_1 \leqslant \ldots \leqslant a_n$.

2) Проводим последовательные вычисления по рекуррентной формуле

$$\begin{gathered} \varphi_1 = 0, \\ \varphi_{k+1} = \varphi_k + k(a_{k+1} - a_k), \quad k = 1, \ldots, n-1, \end{gathered} \tag{2}$$

пока не встретим индекс k_0, на котором

$$\varphi_{k_0} < 1 \leqslant \varphi_{k_0+1}.$$

Если $\varphi_n < 1$, то полагаем $k_0 = n$.

3) Вычисляем λ^* по формуле

$$\lambda^* = a_{k_0} + \tfrac{1}{k_0}(1 - \varphi_{k_0}). \tag{3}$$

Компоненты x_i^* проекции точки c на стандартный симплекс имеют вид

$$x_i^* = (\lambda^* + c_i)_+, \quad i \in 1:n, \tag{4}$$

где $(u)_+ = \max\{0, u\}$.

КОММЕНТАРИИ К АЛГОРИТМУ

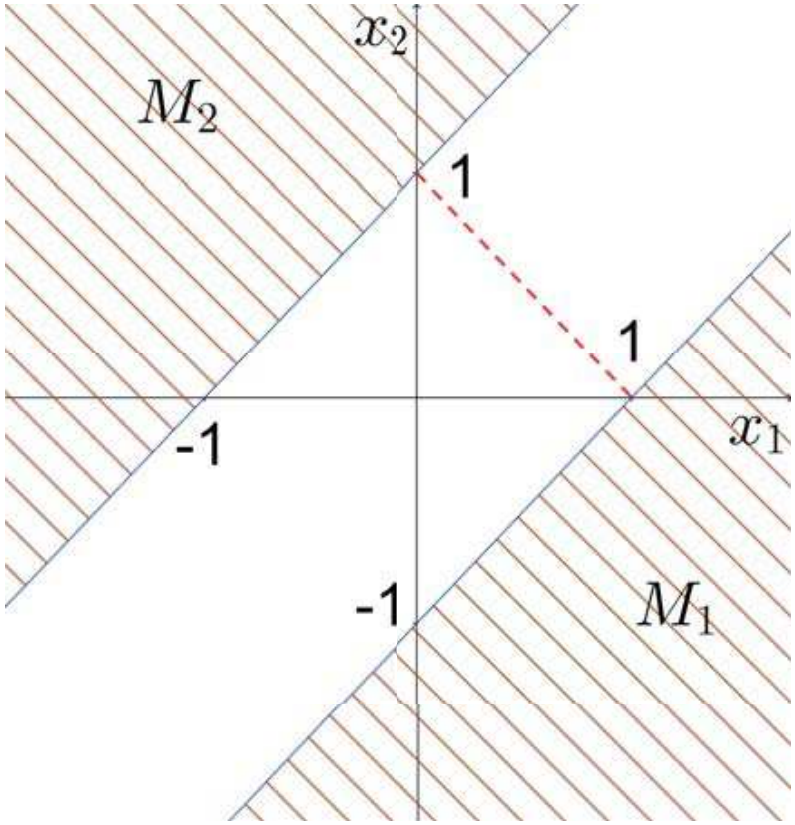
- Алгоритм сходится не более, чем за n шагов.

- В основе алгоритма лежит рекуррентное соотношение (2) для скалярных величин.

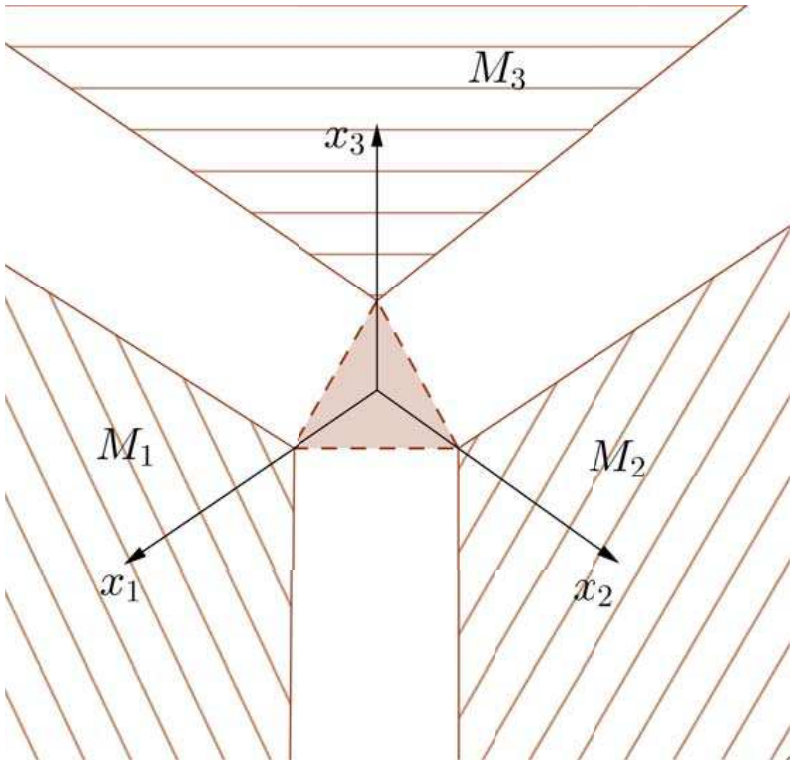
- В общем случае алгоритм требует сортировку элементов массива длиной n и не более $4n - 2$ арифметических операций.

ЛУЧШИЙ И ХУДШИЙ СЛУЧАИ АЛГОРИТМА МАЛОЗЁМОВА – ПЕВНОГО

минимальная трудоёмкость — $\varphi_2 \geqslant 1$



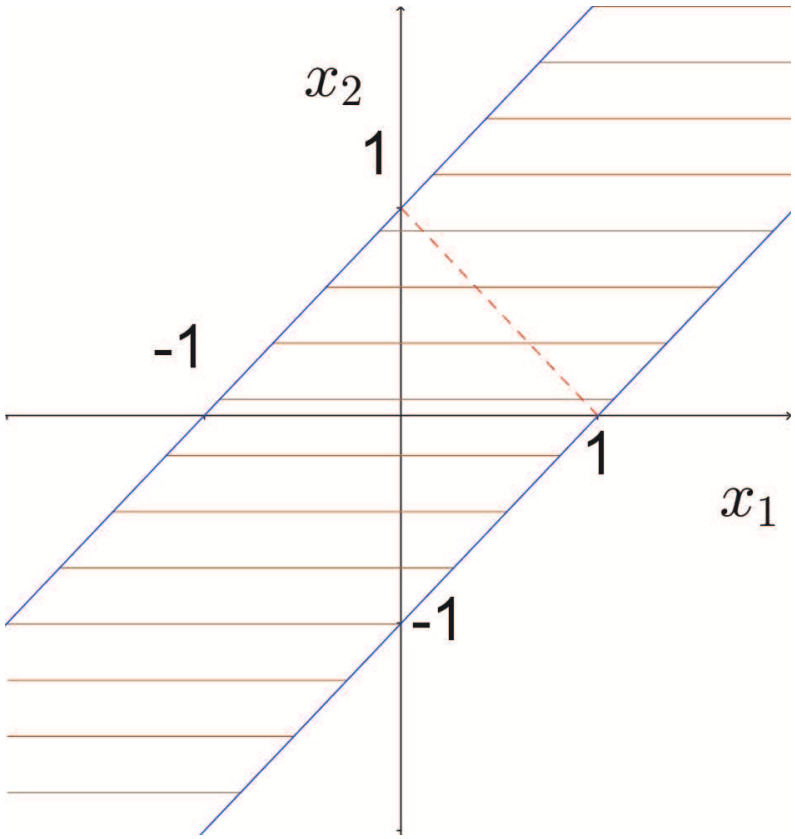
$n = 2$



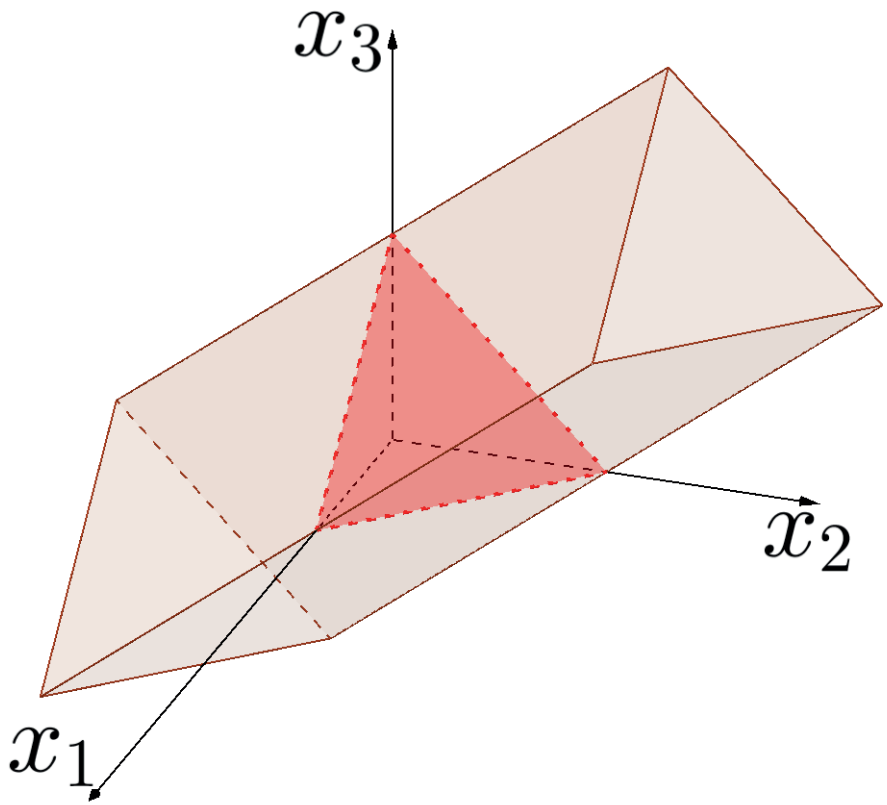
$n = 3$

ЛУЧШИЙ И ХУДШИЙ СЛУЧАИ АЛГОРИТМА МАЛОЗЁМОВА – ПЕВНОГО

максимальная трудоёмкость — $\varphi_{n-1} < 1$



$n = 2$



$n = 3$

МОДИФИКАЦИИ АЛГОРИТМА МАЛОЗЁМОВА – ПЕВНОГО

1. Очередь с приоритетом (наивная реализация, MalozemovNaivePQ).
   - предварительная подготовка не требуется;
   - нахождение очередного значения $O(n)$;
   - в худшем случае $O(n^2)$.

2. Очередь с приоритетом на основе сортирующего дерева (двоичной кучи, MalozemovPQ).
   - предварительное построение дерева $O(n)$;
   - нахождение очередного значения $O(\log(n))$;
   - в худшем случае $O(n \log(n))$.

МЕДИАНА МНОЖЕСТВА

Определение. *Пусть* $S = \{a_1, \ldots, a_\ell\}$, *где*

$$a_1 \leqslant \ldots \leqslant a_\ell.$$

Под медианой M *множества* S *понимается величина* $a_m \in S$, *которая стоит на позиции* $m = \left[\frac{\ell+1}{2}\right]$, *т. е.*

$$M = a_m,$$

где $[z]$ — *целая часть числа* z.

АЛГОРИТМ MACULAN – de PAULA (MACULAN).

Инициализация. $S := \{c_1, \ldots, c_n\}, \quad J := N,$

$$v := 0, \quad p := 0, \quad q := 0.$$

Определим функцию

$$\Phi(t, x, L, v, p, q) := \sum_{j \in L} (x_j - t) + v + p(q - t), \tag{5}$$

где $x = (x_1, \ldots, x_n)$, $L \subseteq N$.

Шаг 1. Рассмотрим множество $S = \{c_j \mid j \in J\}$. Найдём медиану M множества S и построим следующие индексные множества

$$L_= := \{j \in J \mid c_j = M\}, \quad L_> := \{j \in J \mid c_j > M\}, \quad L_< := \{j \in J \mid c_j < M\}.$$

Пусть $m \in L_=$, т. е. $c_m = M$. Вычислим

$$z = \Phi(M, c, L_>, v, p, q). \tag{6}$$

Если $z \geqslant 1$, то переходим к **Шагу 2**; иначе к **Шагу 3**.

Шаг 2. Положим

$$J := L_> \cup \{m\}, \qquad S := \{c_j \mid j \in J\},$$

где m — индекс медианы S найденный на **Шаге 1**.

Если $|J| \geqslant 3$, то переходим к **Шагу 1**;
иначе при $v = z$ и $q = M$ к **Шагу 4**.

Шаг 3. Положим

$$J := L_< \cup \{m\}, \qquad v := z, \qquad q := M,$$
$$p := p + |L_=| - 1 + |L_>|.$$

Если $|J| \geqslant 2$, то переходим к **Шагу 1**; иначе к **Шагу 4**.

Шаг 4. Вычисляем t^* по формуле

$$t^* := q - \frac{1 - v}{1 + p}. \tag{7}$$

Компоненты x_i^* проекции точки c на стандартный симплекс имеют вид

$$x_i^* = (t^* + c_i)_+, \quad i \in 1 : n. \tag{8}$$

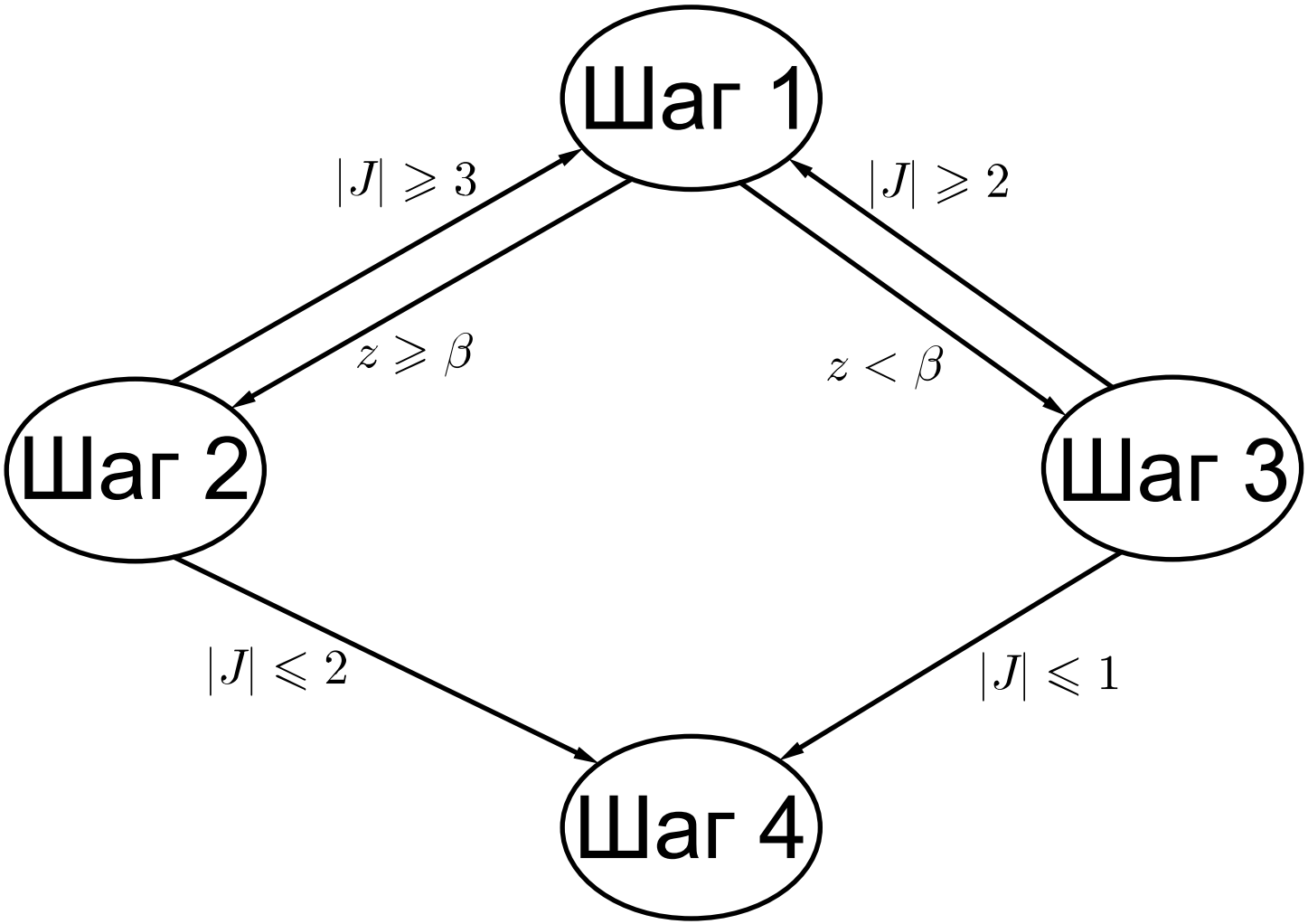


Схема алгоритма Maculan.

КОММЕНТАРИИ К АЛГОРИТМУ

Замечание 1. *Величина z вычисляемая на* ***Шаге 1*** (*см.* (6)) *является значением функции* Φ *в точке* t *равной медиане* M.

Замечание 2. *Несложно понять, что мощность индексного множества* J *от итерации к итерации убывает. Это гарантирует конечность процесса.*

О лучших и худших случаях для алгоритма Maculan – de Paula.

- Лучший случай: при $c_i = c_j$, $\forall i \neq j$, алгоритм завершается за одну итерацию.

- Худший случай: при $c_i \neq c_j$, $\forall i \neq j$, количество итераций определено размерностью задачи с точностью до константы.

$$\log_2(\widehat{n}) - 0.5849626 < \widehat{k} < \log_2(\widehat{n}) + 2 \tag{9}$$

$$k_{\max}(\widehat{n}) - k_{\min}(\widehat{n}) \leq 2 \tag{10}$$

Минимальная и максимальная размерности для данного числа итераций

Пусть $d(k)$ — минимальная, а $D(k)$ — максимальная размерность задачи, которую алгоритм Maculan решает за k итераций.

$$k_{\min}(\widehat{n}) = \operatorname*{argmin}_k \{D(k) \mid D(k) \geq \widehat{n}\}, \tag{11}$$

$$k_{\max}(\widehat{n}) = \operatorname*{argmax}_k \{d(k) \mid d(k) \leq \widehat{n}\}. \tag{12}$$

k	1	2	3	4	5	6	7	8	9	10	11	12	13
$d(k)$	1	3	4	6	10	18	34	66	130	258	514	1026	2050
$D(k)$	3	6	12	24	48	96	192	384	768	1536	3072	6144	12288

Например, задача размерности 100 потребует
$k_{\min}(100) = 7$ или $k_{\max}(100) = 8$ итераций алгоритма Maculan.

Численные эксперименты

10000 точек в n-мерном евклидовом пространстве
при n равном 10, 10^2, 10^3, 10^4, 10^5 и 10^6.

Координаты точек $c = (c_1, \ldots, c_n)$ формировались пятью способами:

(A) генерировались по непрерывному равномерному распределению на отрезке $[-10000, 10000]$;

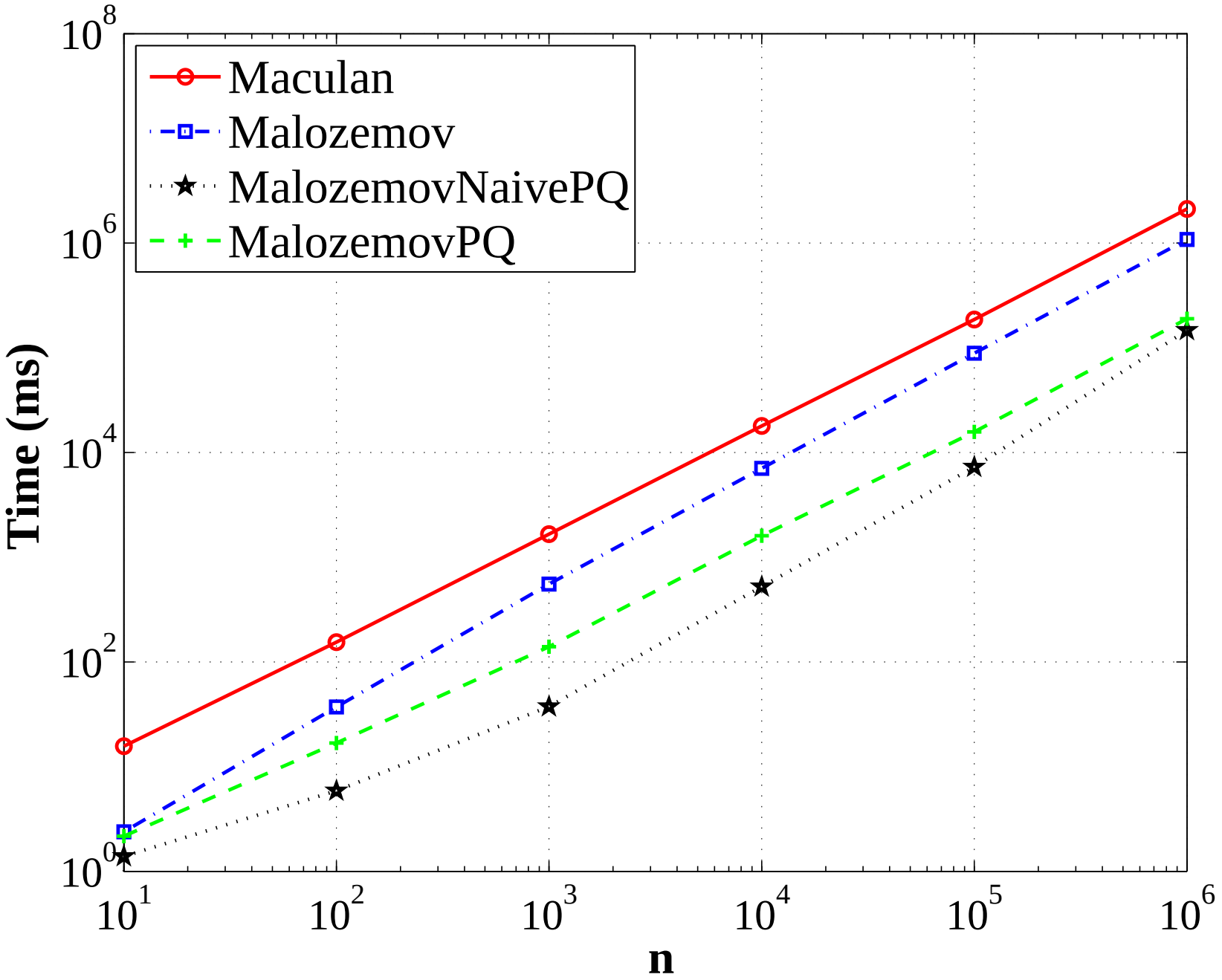
(B) $c_j = \xi + v_j$, $j \in 1 : n$, где $(v_1, \ldots, v_n)$ — любой элемент из Λ, ξ — произвольное фиксированное число из отрезка $[-10000, 10000]$;

(C) фиксировалось произвольное целое число k от 1 до n и вещественное c_k, тогда c_j при $j \neq k$ выбирались удовлетворяющие неравенству

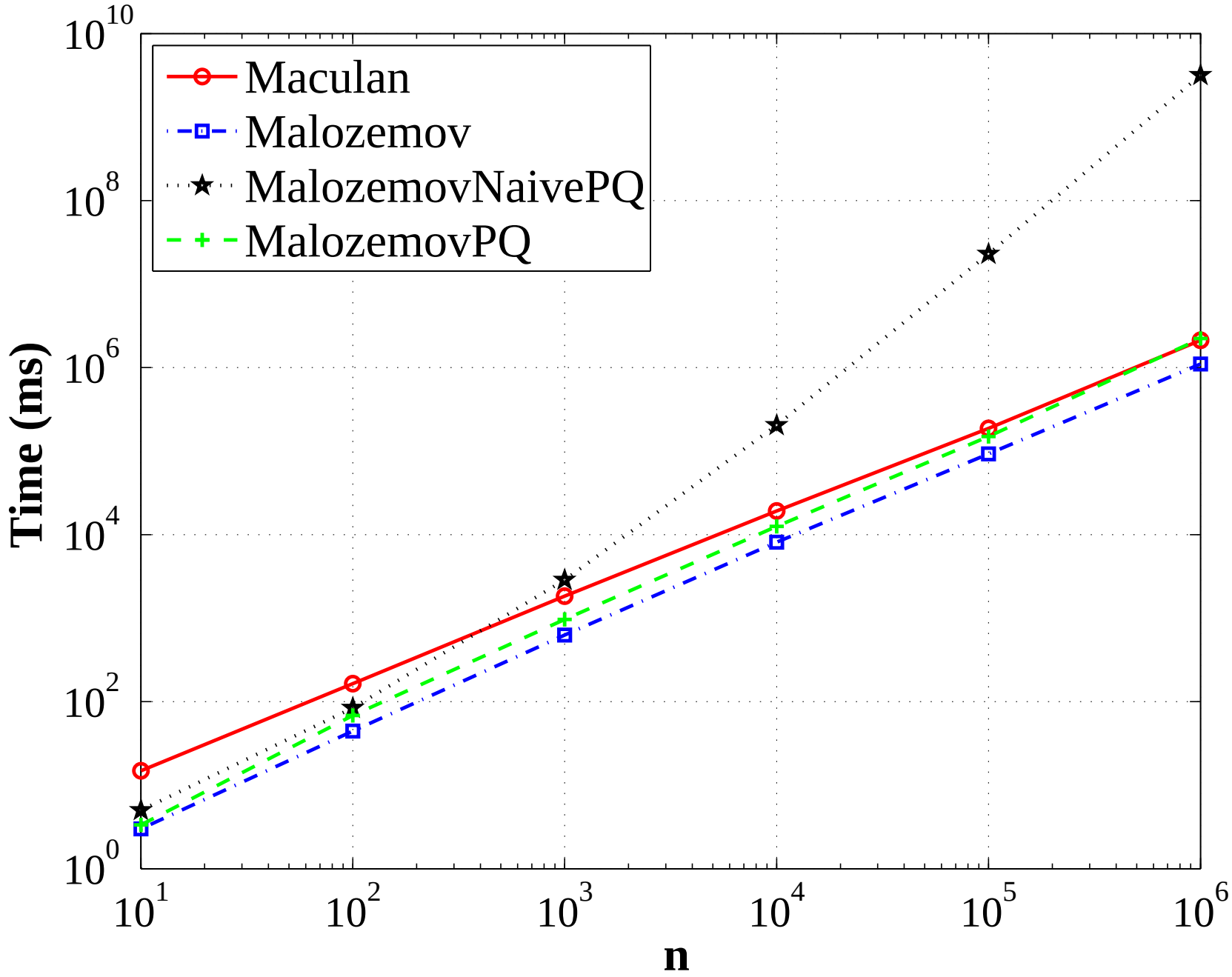
$$c_j \leq c_k - 1;$$

(D) $c_j \neq c_\ell$ для всех $1 \leq j \neq \ell \leq n$, т. е. все компоненты точки c различны;

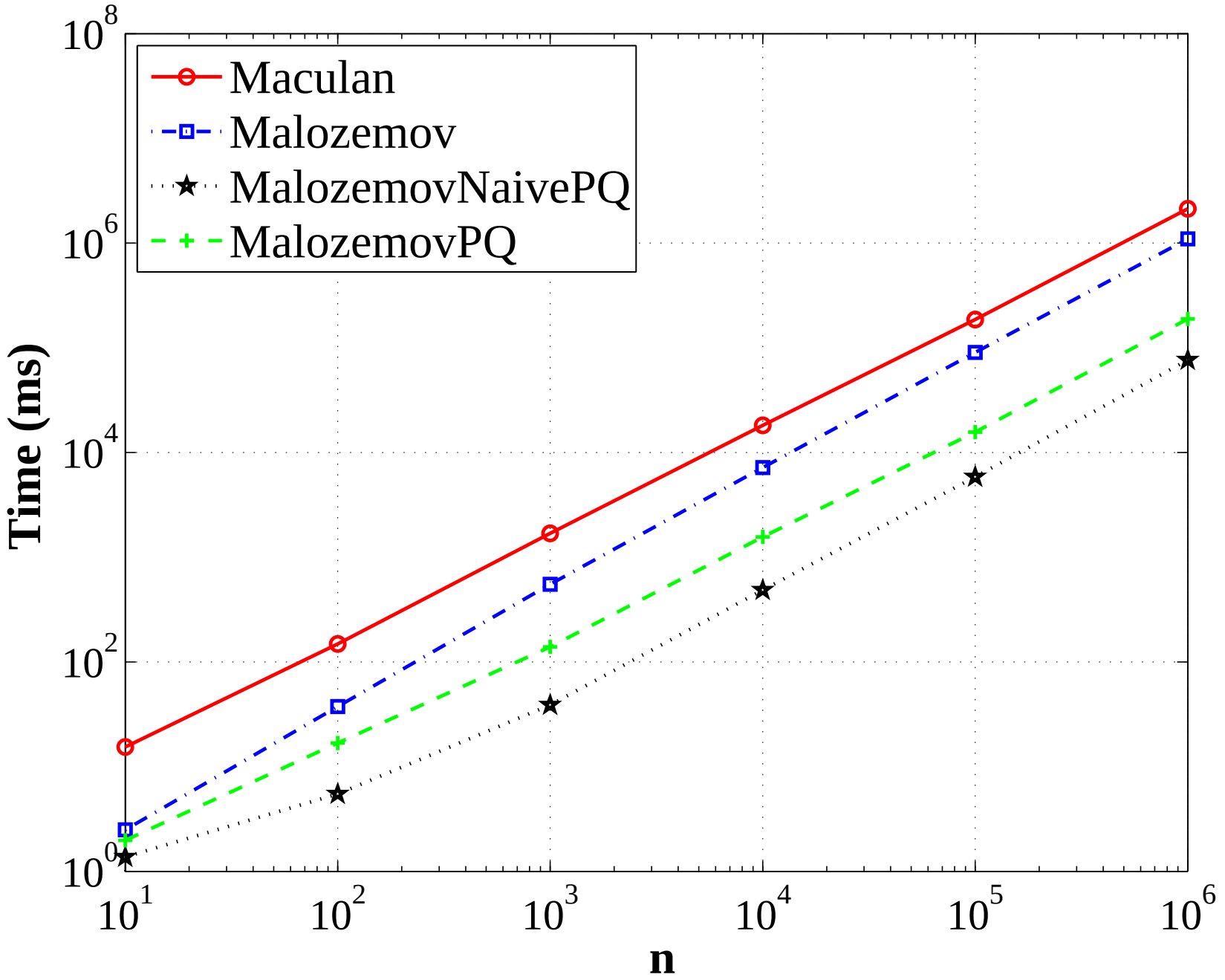
(E) $c_j = \xi$ для всех $1 \leq j \leq n$, где ξ — произвольное фиксированное число из отрезка $[-10000, 10000]$.



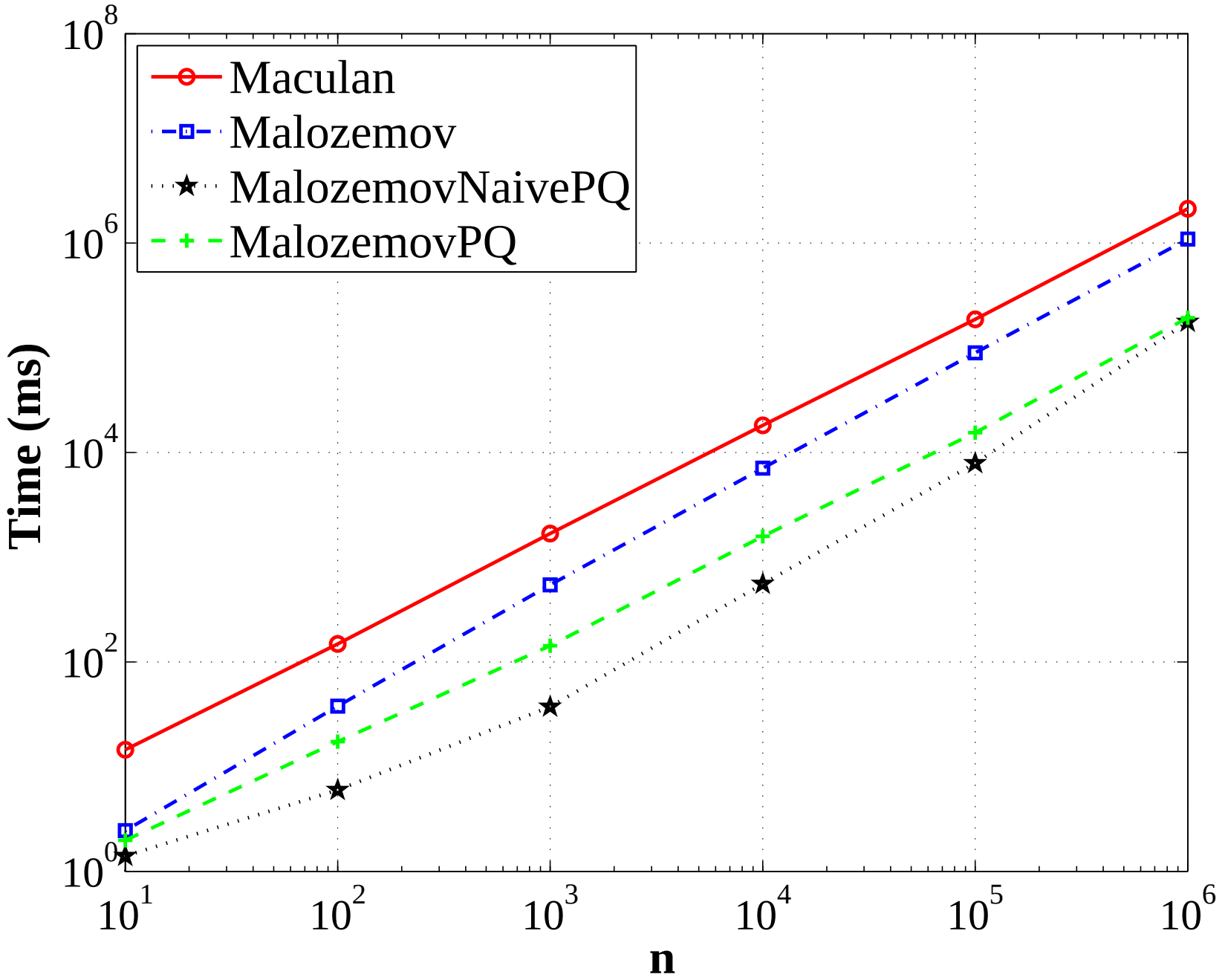
Случай (A) — равномерное распределение



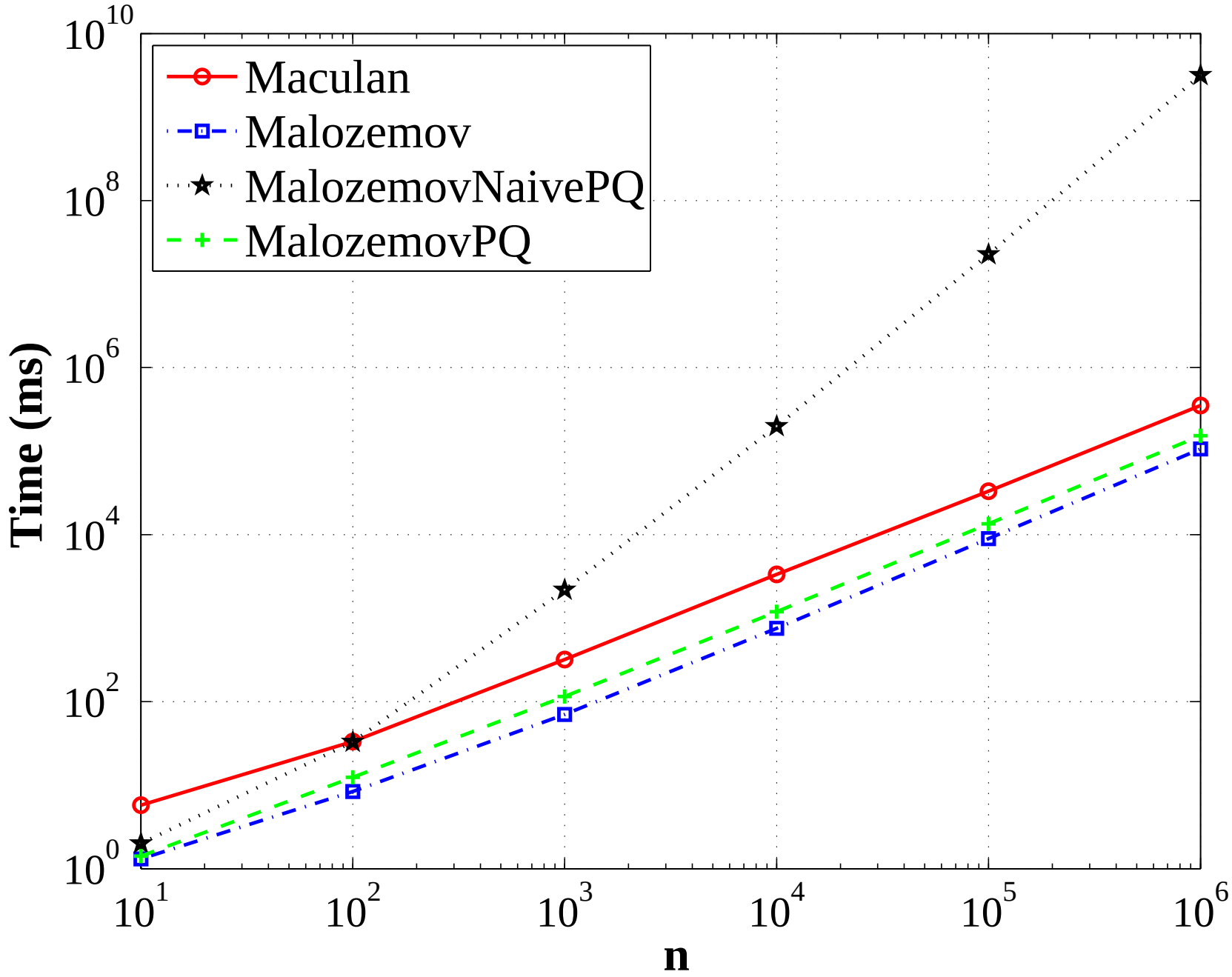
Случай (B) — худший случай для алгоритма Малоземова–Певного



Случай (C) — лучший случай для алгоритма Малоземова–Певного



Случай (D) — худший случай для алгоритма Maculan–de Paula



Случай (E) — лучший случай для алгоритма Maculan–de Paula