

Параллельные вычислительные технологии (ПаВТ'2019)

2-4 апреля 2019 г. :: Калининград

Балтийский федеральный университет

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ
ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СХЕМ
ДЛЯ РЕШЕНИЯ ЗАДАЧ ПРИНЯТИЯ РЕШЕНИЙ**

Гергель В.П., д.т.н., директор ИТММ, ННГУ

Козин Е.А., ассистент каф. МОСТ, ИТММ, ННГУ

Содержание

- ❑ Постановка задачи принятия решений
- ❑ Основы предлагаемого подхода к решению задач принятия оптимальных решений
- ❑ Параллельные алгоритмы решения задач принятия решений
- ❑ Результаты вычислительных экспериментов



Актуальность задачи

- ❑ Задачи принятия оптимальных решений редко можно решить на основе только интуиции и накопленного опыта исследователя.
- ❑ Задачи принятия оптимальных решений во многих случаях могут быть сведены к задачам многокритериальной оптимизации (МКО).
- ❑ Решение задачи МКО осложнено следующими факторами:
 - критерии являются **многоэкстремальными**,
 - критерии могут быть **вычислительно трудоемкими**,
 - в предельном случае необходимо отыскание **всей области Парето**,
 - постановка задачи МКО **может меняться в ходе вычислений**.

Математическая постановка задачи принятия оптимальных решений

Объект принятия оптимальных решений

$$w(y) = (w_1(y), w_2(y), \dots, w_M(y)),$$

- $w(y)$ – **вектор характеристик**,
- $y = (y_1, y_2, \dots, y_s) \in D$ – **вектор конструктивных параметров**,
- $D = \{ y \in R^N: a_i \leq y_i \leq b_i, 1 \leq i \leq N \}$.

$w_1(y)$ = Прочность машины

$w_2(y)$ = Стоимость машины



$w_3(y)$ = Вес машины



$w_4(y)$ = Расход топлива

Математическая постановка задачи принятия оптимальных решений

Требования к оптимальному выбору

$$f(y) = (f_1(y), f_2(y), \dots, f_s(y)),$$

- $f_j(y) = w_{ij}(y)$ – **векторный критерий эффективности**,

$$g(y) = (g_1(y), g_2(y), \dots, g_m(y))$$

- $g_l(y) = w_{jl}(y) - q_l$ – **вектор-функция ограничений**,
- q_l – допуск на значения критериев.



Множество задач многокритериальной оптимизации

Каждое требование задает новую задачу оптимизации

$$\vec{Z} = \{Z_i : 1 \leq i \leq v\}. \quad Z_i \in \vec{Z} : f(y) \rightarrow \min, y \in Q, Q = \{y \in D : g(y) \leq 0\}$$

Прочность машины $\rightarrow \max$



Стоимость машины $\rightarrow \min$



Вес машины $< 1,5$ т.



Расход топлива < 11 л.

Основы предлагаемого подхода:

Сведение задач МКО к одномерным задачам ГО

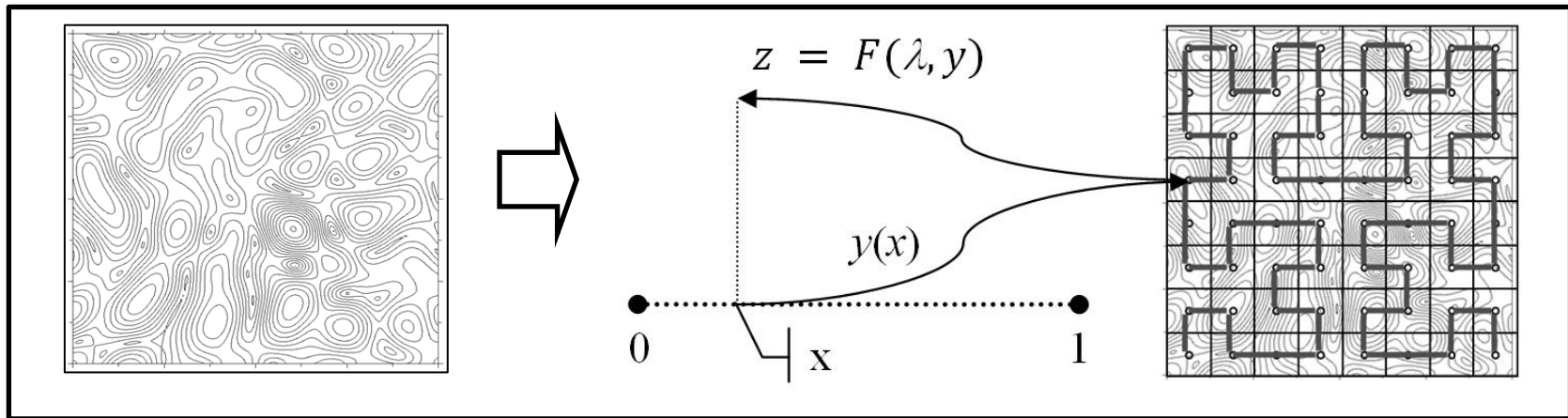
□ В рамках разработанного подхода применяются:

- Свертка набора частных критериев $f_i(y)$

$$F(\lambda, y) = \max(\lambda_i * f_i(y), 1 \leq i \leq s), \sum_{i=1}^s \lambda_i = 1, \lambda_i \geq 0, 1 \leq i \leq s.$$

- Редукция размерности на основе *кривых (разверток) Пеано* $y(x)$

$$\min_{x \in [0, 1]} \varphi(x), \text{ где } \varphi(x) = F(\lambda, y(x)), y(x) \in Q.$$



- Учет ограничений выполнен на основе использования индексной схемы.

Основы предлагаемого подхода:

Базовый алгоритм глобального поиска (АГП*)

- ❑ Испытание – вычисление значений вектор-функции $w(y(x_i))$ в точке x_i .
- ❑ Общая схема алгоритма поиска глобального минимума:

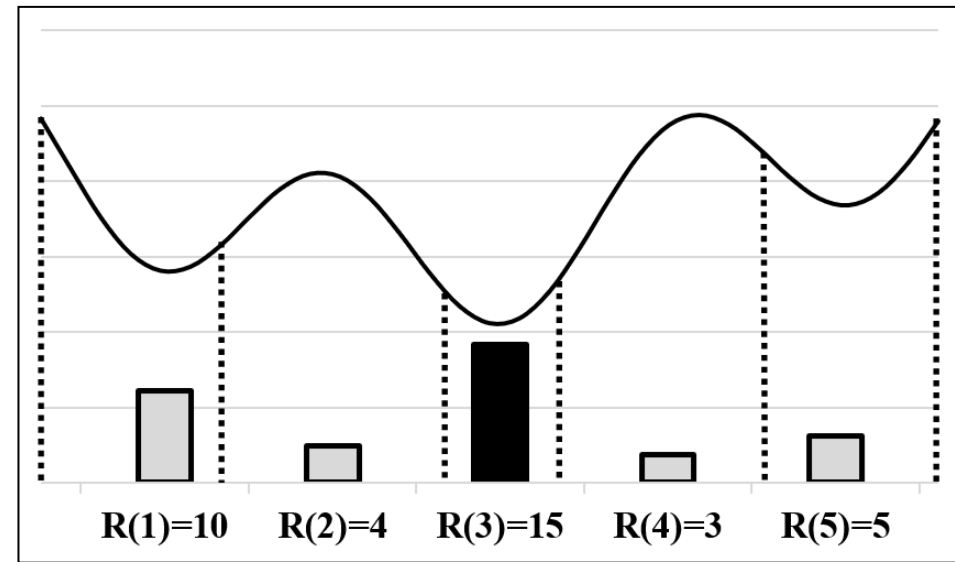
Первое испытание проводится в произвольной точке $x^1 \in (0,1)$. Далее:

1. Отсортировать точки испытаний в порядке возрастания их координат
 $0 = x_0 < x_1 < \dots < x_i < \dots < x_k < x_{k+1} = 1$.
2. Для каждого интервала (x_{i+1}, x_i) вычислить значение характеристики $R(i)$.

3. Определить интервал (x_{t-1}, x_t) ,
которому соответствует максимальная характеристика
 $R(t) = \max \{R(i): 1 \leq i \leq k+1\}$.

4. Провести очередное испытание в точке интервала $x^{k+1} \in (x_{t-1}, x_t)$.

5. Условие остановки $\rho_t \leq \varepsilon$, где
 $\rho_j = \sqrt[N]{x_i - x_{i-1}}$.



Основы предлагаемого подхода: Ускорение вычислений на основе повторного использования информации...

□ Решение задач – последовательность испытаний $w^i = w(y^i)$.

- Вся доступная информация о решаемой задаче оптимизации (множество поисковой информации, МПИ):

$$\Omega_k = \left\{ \left(y^i, w^i = w(y^i) \right)^T : 1 \leq i \leq k \right\}.$$

- МПИ преобразуется к матрице состояния поиска (МСП) :

$$A_k = \{ (x_i, z_i, l_i)^T : 0 \leq i \leq k \}$$

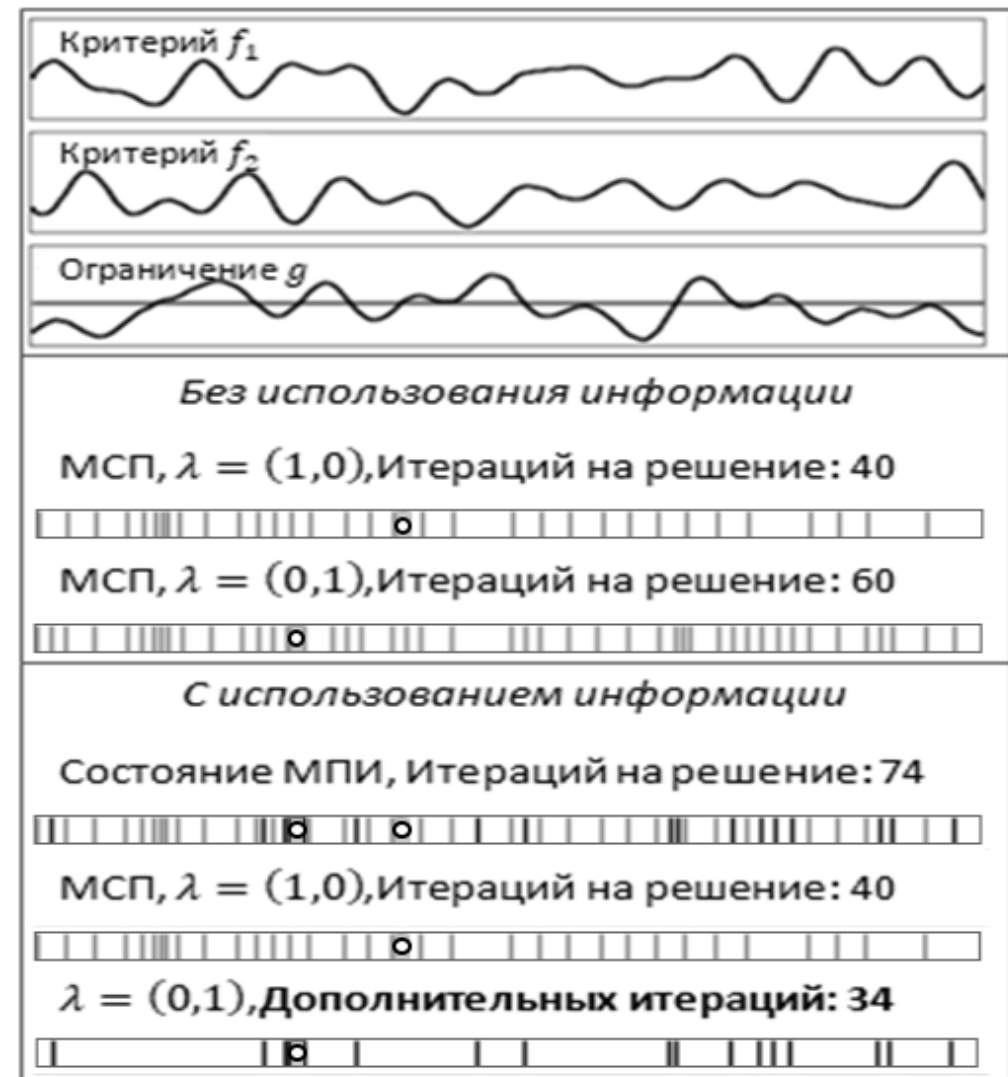
- $x_i = y(y_i)$ – редуцированные точки,
- $z_i = \max(\lambda^j f_i^j, 1 \leq j \leq s)$,
- l_i - номера итераций.

□ Поисковая информация может быть приведена к новой постановке задачи **без дополнительных вычислений значений характеристик:**

$$z'_i = \max(\lambda'_j f_i^j, 1 \leq j \leq s), \quad 1 \leq i \leq k.$$

Основы предлагаемого подхода: Ускорение вычислений на основе повторного использования информации

- ❑ Пример решения задачи:
 - **Серый** – точки проведения испытаний.
 - **Черный** – точки проведения испытаний при повторном использовании информации.
 - **Черные окружности** – точки принадлежащие области Парето.
- ❑ Основной результат применения МАМГП:
 - Первая подзадача решается за равное количество итераций (40 итераций).
 - **Вторая подзадача за счет повторного использования информации решается почти в два раза быстрее (34 против 60 итераций).**

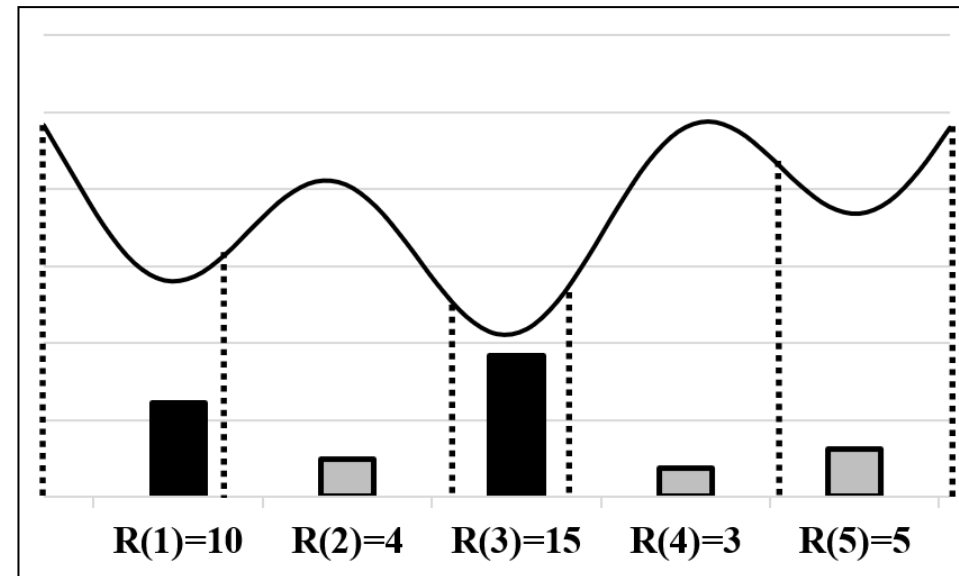


Параллельные вычисления для вычислительных систем с общей памятью

- Параллельный многомерный алгоритм многокритериального глобального поиска для общей памяти (ПАМГП-ОП).

Параллельный алгоритм, дополненный возможностью повторного использования информации:

1. Отсортировать точки испытаний в порядке возрастания их координат
 $0 = x_0 < x_1 < \dots < x_i < \dots < x_{k \cdot p} < x_{k \cdot p + 1} = 1$.
2. Для каждого интервала (x_{i-1}, x_i) вычислить характеристики интервалов $R(i)$.
3. Отсортировать интервалы по убыванию характеристик, взять p интервалов
 $R(t_1) \geq R(t_2) \geq \dots \geq R(t_p)$.
4. Провести p испытаний параллельно
 $(x_{t_1-1}, x_{t_1}), (x_{t_2-1}, x_{t_2}), \dots, (x_{t_p-1}, x_{t_p})$.
5. Критерий остановки:
 $\rho_{t_i} \leq \varepsilon, 1 \leq t_i \leq p$.



Параллельные вычисления для вычислительных систем с распределенной памятью...

- Решение задачи МКО может быть сформулировано как проблема решения семейства задач

$$\vec{\Phi}(x) = \left\{ \begin{array}{l} \Phi_1(x) \leftarrow (F(\lambda_1, y(x)), g(y)), \\ \Phi_2(x) \leftarrow (F(\lambda_2, y(x)), g(y)), \\ \dots, \\ \Phi_\tau(x) \leftarrow (F(\lambda_\tau, y(x)), g(y)) \end{array} \right\}.$$

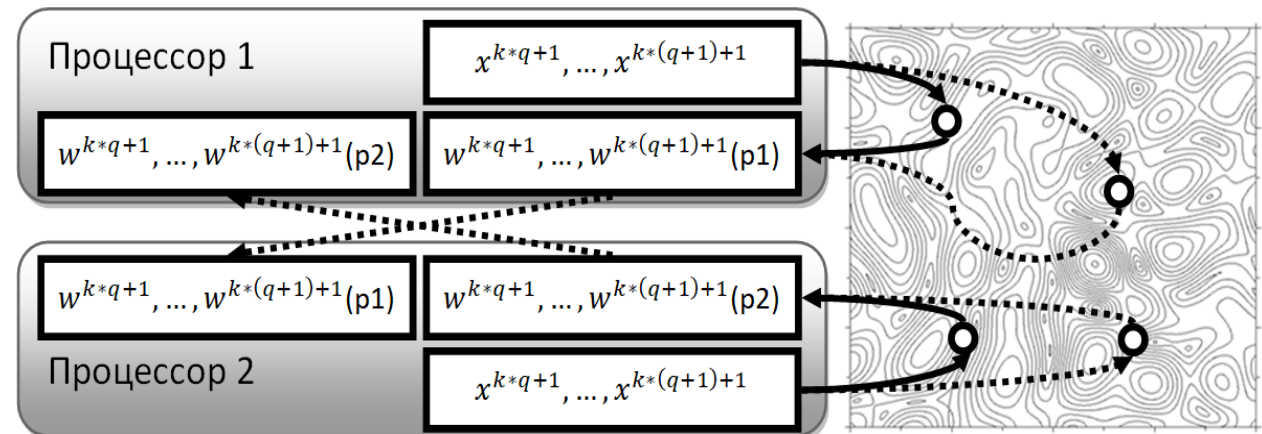
- Семейство подзадач $\vec{\Phi}(x)$ можно решать как последовательно, так и параллельно:
 - Семейство подзадач $\vec{\Phi}(x)$ является *информационно-связанным*.
 - Значения оптимизируемых функций, вычисленные для любой подзадачи $\Phi_l(x)$, $1 \leq l \leq \tau$, *могут быть приведены* к значениям всех остальных задач этого семейства.



Параллельные вычисления для высокопроизводительных вычислительных систем

□ Параллельный многомерный алгоритм многокритериального глобального поиска (ПАМГП).

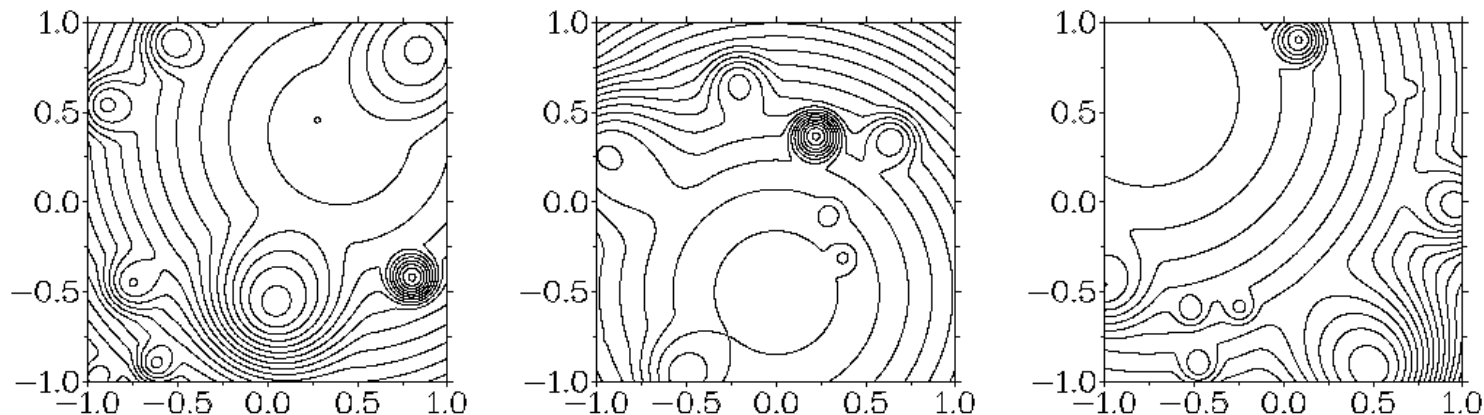
- В начале вычислений подзадачи $\Phi_i(x)$ распределяются по вычислительным процессорам.
- Перед каждой итерацией происходит проверка на наличие вычисленных значений критериев и ограничений от соседних процессоров.
- При наличии вычисленных значений обновляется МПИ и МСП.
- Выполняется итерация глобального поиска согласно ПАМГП-ОП.
- Вычисленные значения критериев и ограничений рассылаются всем процессорам.



Решаемая задача

□ Решалась серия задач МКО

- Число задач в серии 30
- Каждая задача МКО шестимерная пятикритериальная ($N = 6, s = 5$)
- Для каждой задачи МКО использовалось 100 различных наборов коэффициентов сверток λ .
- Критерии получены генератором GKLS.
- Пример функций GKLS при размерности равной двум:



Результаты вычислительных экспериментов

□ Оценка эффективности использования одного узла кластера.

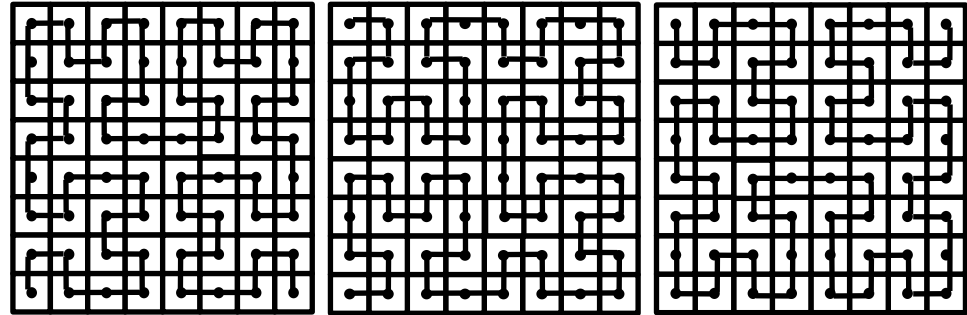
Ядер	Информация повторно	Итераций	S1	S2
1	не используется	26 813 722,7	1,0	
1	используется	9 103 069,6	2,9	1,0
8	используется	1 291 720,0	20,8	7,0
16	Используется	609 169,5	44,0	14,9

- $S1$ – ускорение параллельного алгоритма относительно алгоритма, *не использующего* МПИ.
- $S2$ – ускорение параллельного алгоритма относительно алгоритма, *использующего* МПИ.

Возможные вычислительные схемы

- Использование всех процессоров для решения одной задачи глобальной оптимизации.

- Использование множественных разверток.



- Использование процессоров для решения подзадач, отличающимися только набором коэффициентов λ .

$$\vec{\Phi}(x) = \left\{ \begin{array}{l} \Phi_1(x) \leftarrow (F(\lambda_1, y(x)), g(y)), \\ \Phi_2(x) \leftarrow (F(\lambda_2, y(x)), g(y)), \\ \dots, \\ \Phi_\tau(x) \leftarrow (F(\lambda_\tau, y(x)), g(y)) \end{array} \right\}.$$

- Использование процессоров для решения множества задач МКО.

$$Z_i \in \vec{Z}: f(y) \rightarrow \min, y \in Q, Q = \{y \in D : g(y) \leq 0\}$$

Сравнение различных вычислительных схем

□ В серии 30 задач МКО. На каждом узле используются по 16 ядер.

– Z – число параллельно решаемых задач МКО.

– Φ – число подзадач отличающихся набором λ .

– E – число используемых разверток.

– $S1$ – ускорение параллельного

алгоритма относительно алгоритма, *не использующего* МПИ.

– $S2$ – ускорение, полученное в результате использования нескольких процессоров и 16 ядер на каждом процессоре

Проц.	Z	Φ	E	Итераций	$S1$	$S2$
1	1	1	1	609 169,50	44,0	1
10	1	10	1	75 045,80	357,3	8,1
10	1	5	2	77 643,60	345,3	7,8
10	1	1	10	108 700,00	247,7	5,6
10	2	5	1	76 819,00	349,1	7,9
10	2	1	5	81 132,80	331,5	7,5
10	5	2	1	66 618,60	403,5	9,1
10	5	1	2	77 995,00	344,8	7,8
10	10	1	1	73 999,70	362,3	8,2



Спасибо за внимание!

