Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Кафедра математического обеспечения и суперкомпьютерных технологий

Разработка эффективных структурх хранения данных для алгоритмов глобальной оптимизации

Выполнил:
студент группы 381503м2
Сврасов В. В.

Нижний Новгород, 2016

Содержание

Постановка задачи

$$D = \{y \in R^N : a_i \leqslant x_i \leqslant b_i, 1 \leqslant i \leqslant N\}$$

$$\phi(y^*) = \min\{\phi(y) : y \in D\}$$

Предполагается, что целевая функция удовлетворяет условию Липшица в *D*:

$$|\phi(y_1) - \phi(y_2)| \leqslant L\|y_1 - y_2\|, y_1, y_2 \in D, 0 < L < \infty$$

Численное решение задачи означает построение оценки $\widetilde{y}$, близкой по какой-либо норме к точке y^* на основе конечного числа значений целевой функции задачи, вычисленных в точках области *D*.

Редукция размерности

Основные подходы:

- Использование разверток

$$\{y \in R^N : -2^{-1} \leqslant y_i \leqslant 2^{-1}, 1 \leqslant i \leqslant N\} = \{y(x) : 0 \leqslant x \leqslant 1\}$$

$$\varphi(y(x^*)) = \min\{\varphi(y(x)) : x \in [0;1]\}$$

- Многошаговая схема

$$\min \phi(y) : y \in D = \min_{a_1 \leqslant y_1 \leqslant b_1} \min_{a_2 \leqslant y_2 \leqslant b_2} \dots \min_{a_1 \leqslant y_N \leqslant b_N} \phi(y)$$

- Блочная многошаговая схема

$$u_1 = (y_1, y_2, \dots, y_{N_1}), u_2 = (y_{N_1+1}, y_{N_1+2}, \dots, y_{N_1+N_2}), \dots,$$
$$u_M = (y_{N-N_M+1}, y_{N-N_M+2}, \dots, y_N)$$

$$\min \phi(y)_{y \in D} = \min_{u_1 \in D_1} \min_{u_2 \in D_2} \dots \min_{u_M \in D_M} \phi(y)$$

Метод глобальной оптимизации

- Общая схема характеристического метода:

пусть имеется *k* результатов испытаний

Шаг 1. Упорядочить поисковую информацию по возрастанию координат точек испытаний $a = x_0 < x_1 < \ldots < x_{k+1} = b$

Шаг 2. Вычислить для каждого интервала $(x_{i-1}, x_i), 1 \leqslant i \leqslant k$

величину *R*(*i*), называемую характеристикой.

Шаг 3. Выбрать интервал номер *t* с наибольшей характеристикой и провести в нем испытание:

$$x^{k+1} = d(t) \in (t_{t-1}, x_t)$$

Критерий остановки: $||x_t - x_{t-1}|| < \varepsilon$

Метод глобальной оптимизации

Формулы для вычисления характеристик:

$$R(1) = 2\Delta_1 - 4\frac{z_1}{M}, R(k+1) = 2\Delta_{k+1} - 4\frac{z_k}{M}$$

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{M^2\Delta_i} - 2\frac{z_i + z_{i-1}}{M}, 1 < i < k+1$$

$$\Delta_i = (x_i - x_{i-1})^{\frac{1}{N}}$$

Правило выбора очередной точки:

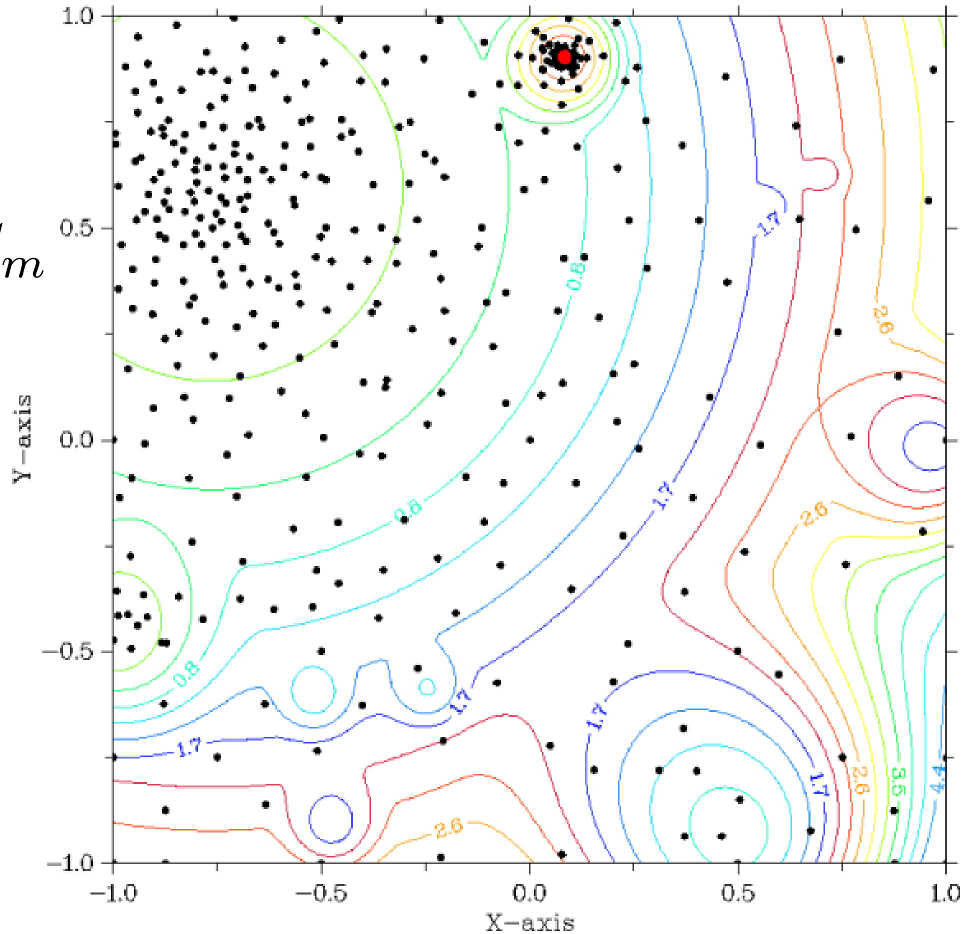
$$x_{k+1} = \frac{x_t + x_{t-1}}{2}, t = 1, t = k+1$$

$$x_{k+1} = \frac{x_t + x_{t-1}}{2} - (z_t - z_{t-1})\frac{1}{2r}\left[\frac{|z_t - z_{t-1}|}{\mu}\right]^N, 1 < t < k+1$$

Класс тестовых задач

Генератор GKLS

$$f(x) = \begin{cases} C_i(x), x \in S_i, i \in 2, \ldots, m \\ ||x - T||^2 + t, x \notin S_2, \ldots, S_m \end{cases}$$



Использование методов локальной оптимизации

- Способы использования локального поиска (метод Хука-Дживса):

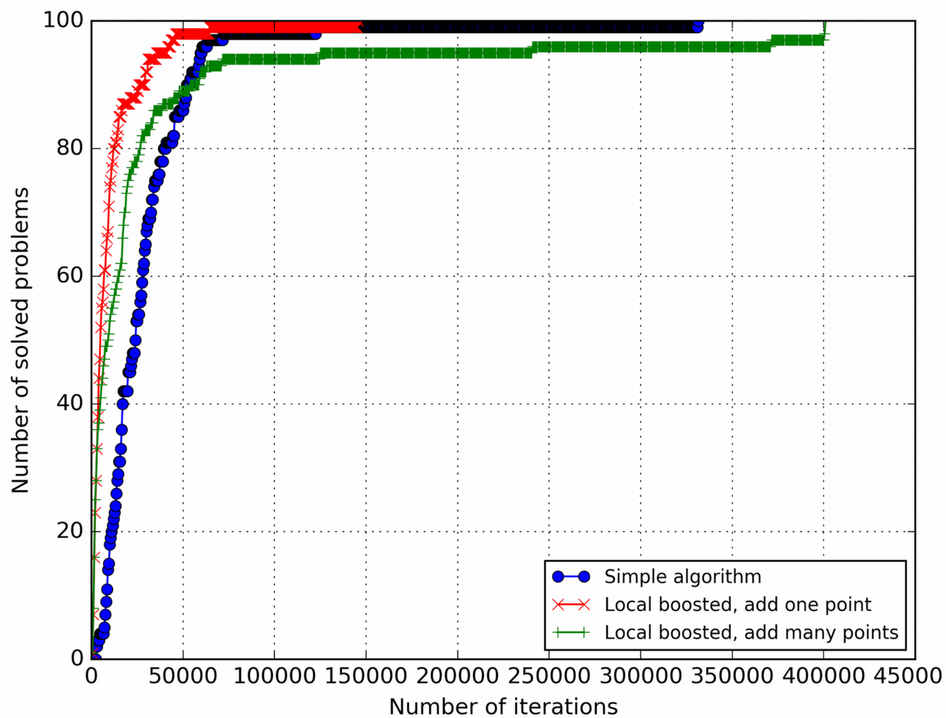
1) Запуск из лучшей найденной точки после окончания работы АГП.
2) Запуски из текущих лучших точек в процессе работы АГП.

Стратегии сохранения информации (для п. 2):

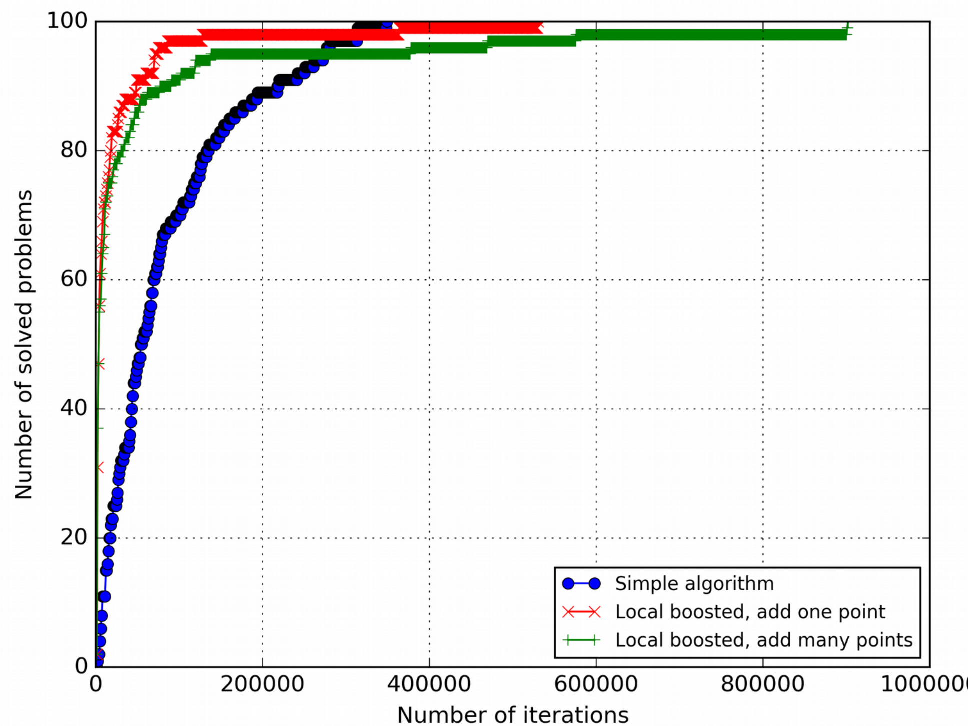
- добавлять в поисковую информацию все точки
- добавлять только лучшие точки

Использование методов локальной оптимизации

GKLS 4d Simple



GKLS 5d Simple



Смешанный алгоритм

Метод является модификацией АГП. Каждый интервал имеет две характеристики

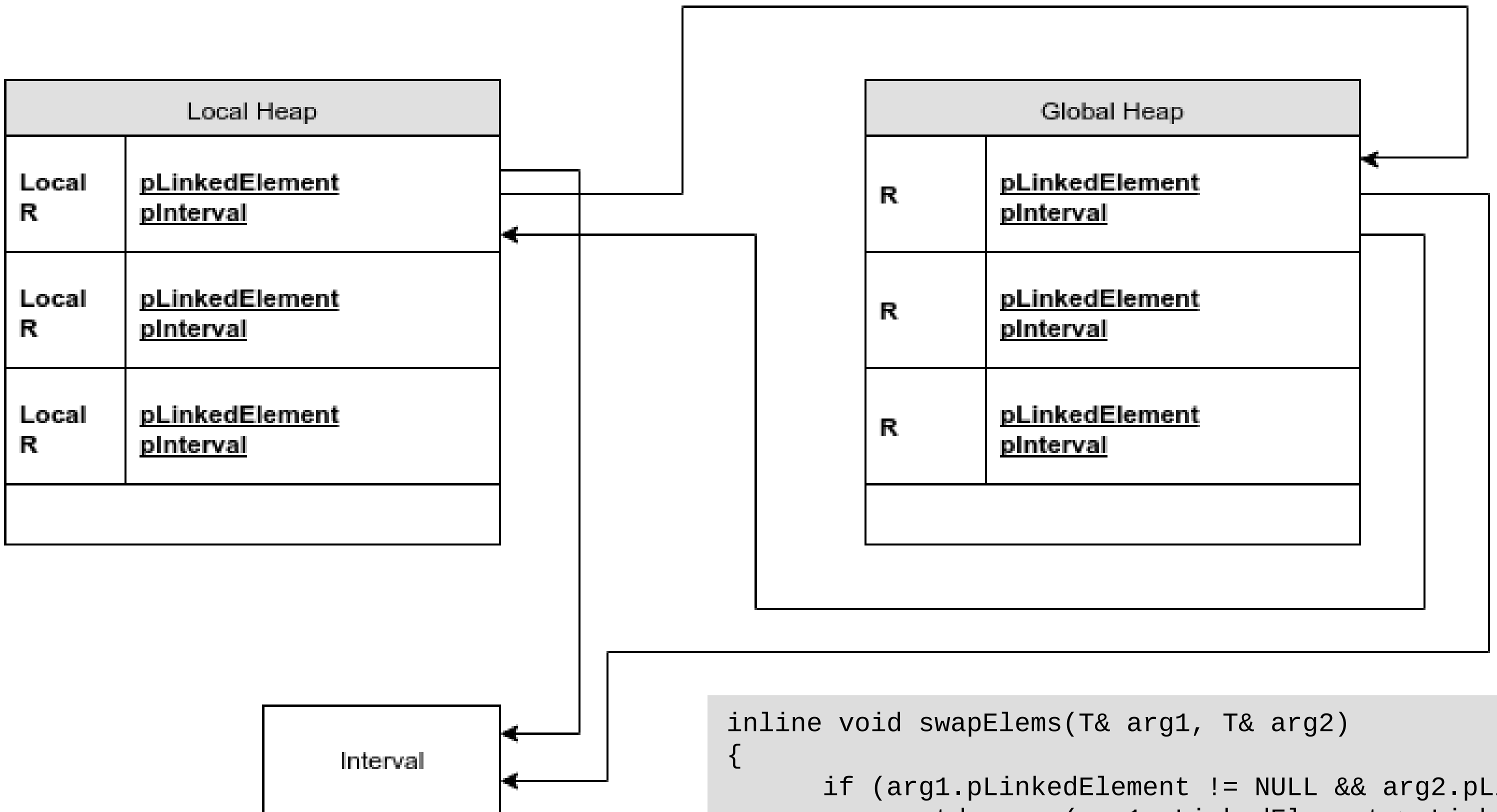
$$R^*(i) = \frac{R(i)}{\sqrt{(z_i - z^*)(z_{i-1} - z^*)}/\mu + 1.5^{-\alpha}}$$

Для эффективной реализации АГП используется приоритетная очередь интрервалов. Ключ – *R(i).*
Для смешанного АГП – две связанные очереди.

Операции с очередями:

- Синхронная вставка
- Синхронное удаление
- Обновление перекрестных ссылок при восстановлении кучеобразности

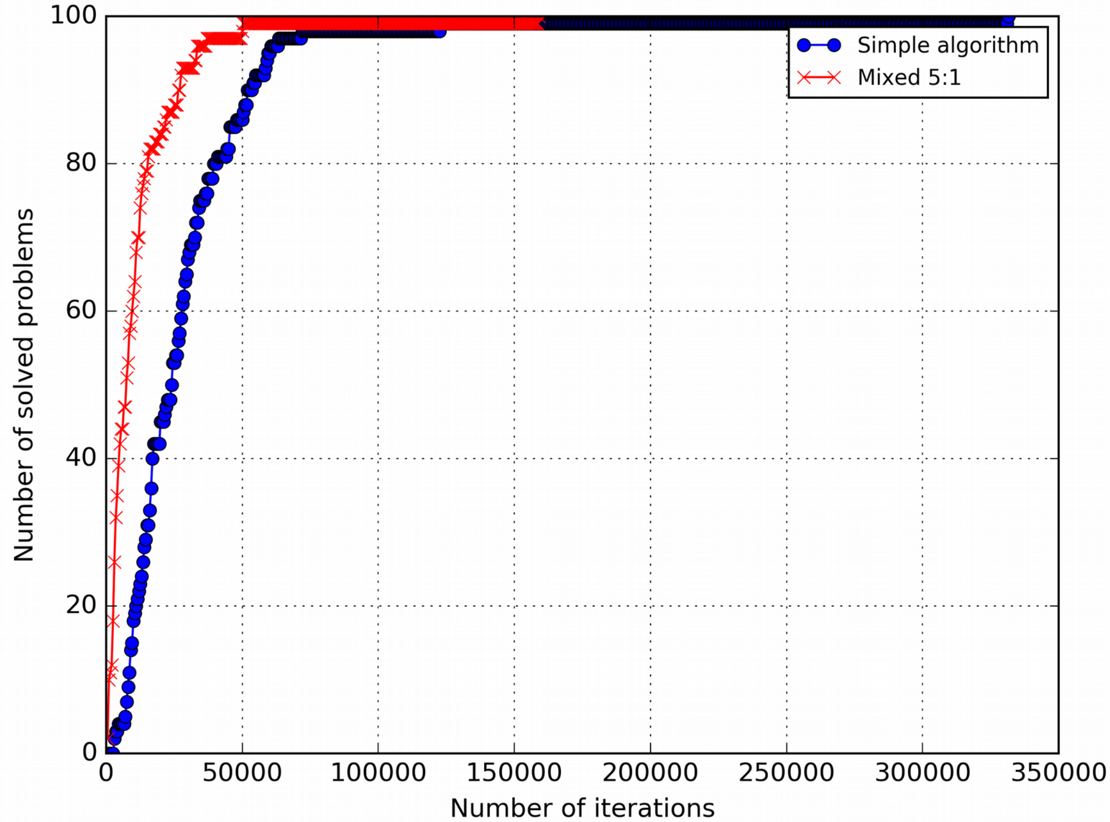
Смешанный алгоритм




```
inline void swapElems(T& arg1, T& arg2)
{
	if (arg1.pLinkedElement != NULL && arg2.pLinkedElement != NULL)
		std::swap(arg1.pLinkedElement->pLinkedElement,
			arg2.pLinkedElement->pLinkedElement);
	else if (arg1.pLinkedElement != NULL)
			arg1.pLinkedElement->pLinkedElement = &arg2;
	else if(arg2.pLinkedElement != NULL)
			arg2.pLinkedElement->pLinkedElement = &arg1;
	std::swap(arg1, arg2);
}
```

Смешанный алгоритм

Результаты на классе GKLS Simple 4d



Многоуровненые развертки

- Основная идея: использование композиции разверток меньшей размерности

Пример двухуровневой схемы:

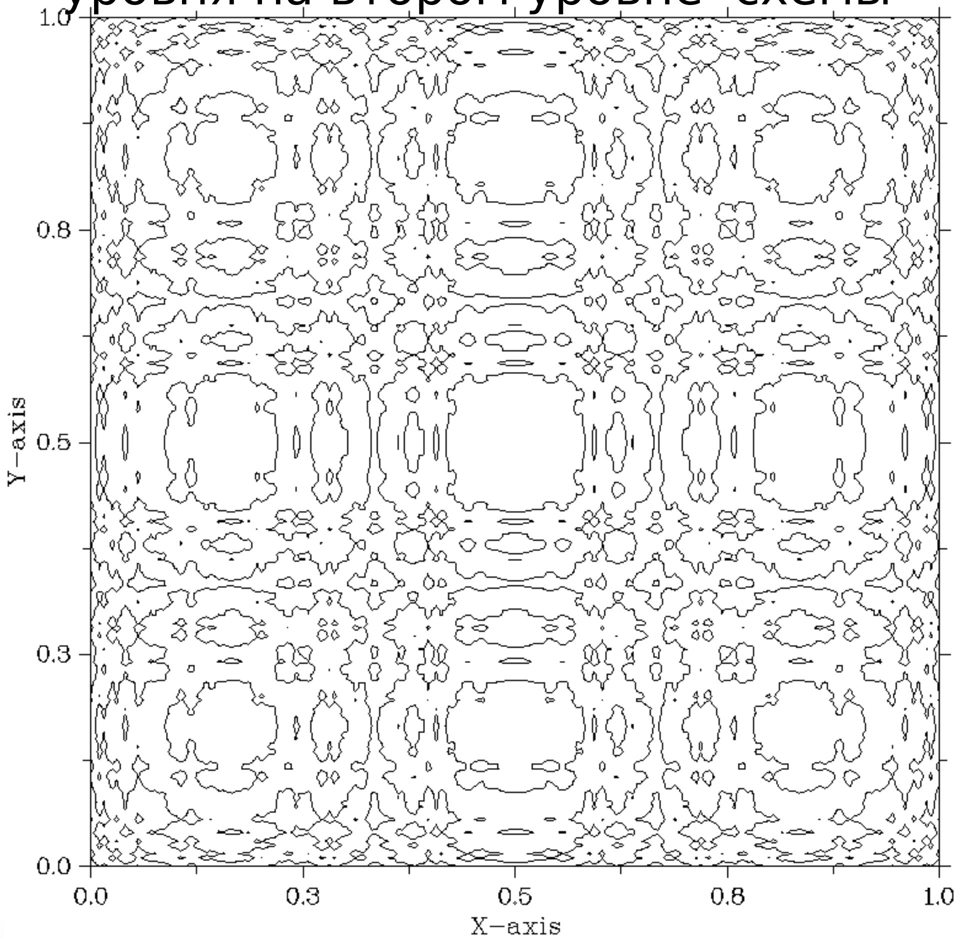
$$\psi(x_1, x_2) = \phi(y(x_1), y(x_2))$$
$$(x_1, x_2) = x(t), t \in [0; 1]$$

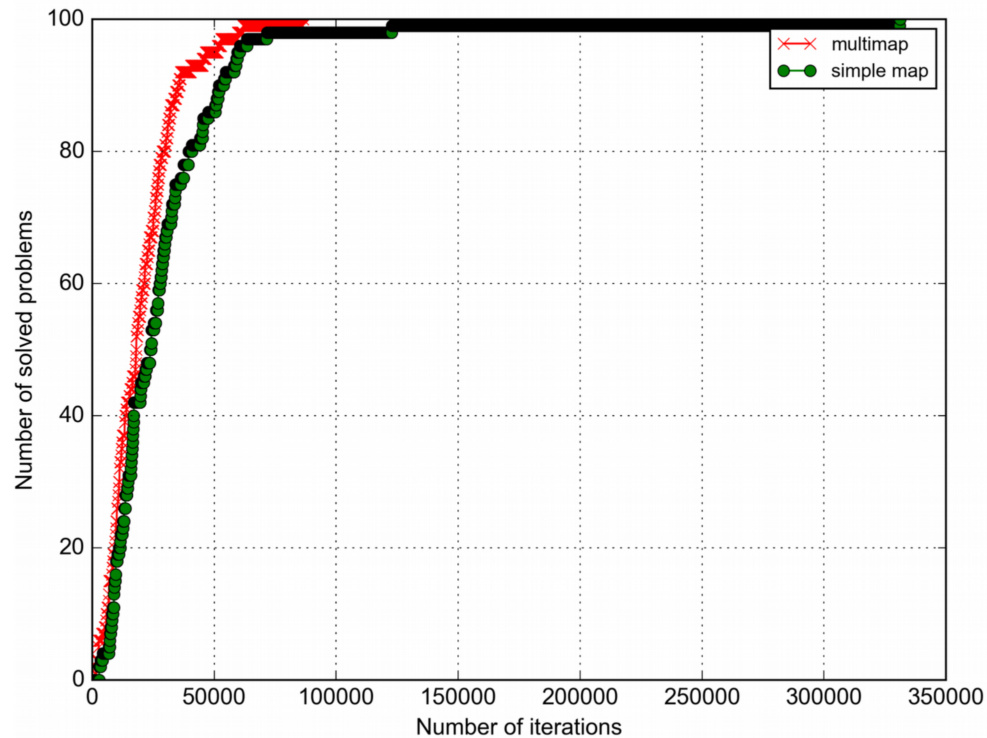
Возможные проблемы:

АГП может быть неприменим к одномерной задаче

Многоуровненые развертки

Линии развертки парабалоида уровня на втором уровне схемы





Результаты на классе GKLS 4d Simple