



УНИВЕРСИТЕТ  
ЛОБАЧЕВСКОГО

# Исследование схем ускорения сходимости алгоритмов глобальной оптимизации

**В.В. Соврасов**

ННГУ им. Н.И. Лобачевского

23 июня 2017 г.  
Нижний Новгород

## Постановка задачи

$D = \{y \in R^N : a_i \leq x_i \leq b_i, 1 \leq i \leq N\}$  — некоторый гиперинтервал, на котором определены функции задачи.

$$Q = \{y \in D : g_j(y) \leq 0, 1 \leq j \leq m\}$$
$$\varphi(y^*) = \min\{\varphi(y) : y \in Q\}$$

Предполагается, что целевая функция  $\varphi(y)$  и ограничения  $g_j(y)$  удовлетворяют условию Липшица в области  $D$ :

$$|\varphi(y_1) - \varphi(y_2)| \leq L\|y_1 - y_2\|, y_1, y_2 \in D, 0 < L < \infty$$

Численное решение задачи означает построение оценки  $\tilde{y}$ , 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 развёрток:

$$\{y \in R^N : -2^{-1} \leq y_i \leq 2^{-1}, 1 \leq i \leq N\} = \{y(x) : 0 \leq x \leq 1\},$$
$$\varphi(y(x^*)) = \min\{\varphi(y(x)) : x \in [0; 1]\}$$

# Редукция размерности

Основные подходы:

- ▶ Использование развёрток:

$$\{y \in R^N : -2^{-1} \leq y_i \leq 2^{-1}, 1 \leq i \leq N\} = \{y(x) : 0 \leq x \leq 1\},$$
$$\varphi(y(x^*)) = \min\{\varphi(y(x)) : x \in [0; 1]\}$$

- ▶ Многошаговая схема:

$$\min_{(x_1, \dots, x_n) \in D} f(x_1, \dots, x_n) = \min_{a_1 \leq x_1 \leq b_1} \min_{a_2 \leq x_2 \leq b_2} \dots \min_{a_n \leq x_n \leq b_n} f(x_1, \dots, x_n)$$

# Редукция размерности

Основные подходы:

- ▶ Использование развёрток:

$$\{y \in R^N : -2^{-1} \leq y_i \leq 2^{-1}, 1 \leq i \leq N\} = \{y(x) : 0 \leq x \leq 1\},$$
$$\varphi(y(x^*)) = \min\{\varphi(y(x)) : x \in [0; 1]\}$$

- ▶ Многошаговая схема:

$$\min_{(x_1, \dots, x_n) \in D} f(x_1, \dots, x_n) = \min_{a_1 \leq x_1 \leq b_1} \min_{a_2 \leq x_2 \leq b_2} \dots \min_{a_n \leq x_n \leq b_n} f(x_1, \dots, x_n)$$

- ▶ Блочная многошаговая схема:

$$u_1 = (y_1, y_2, \dots, y_{N_1}), u_2 = (y_{N_1+1}, y_{N_1+2}, \dots, y_{N_1+N_2}), \dots, u_M = (y_{N-N_M+1}, y_{N-N_M+2}, \dots, y_N)$$
$$\min_{y \in D} \varphi(y) = \min_{u_1 \in D_1} \min_{u_2 \in D_2} \dots \min_{u_M \in D_M} \varphi(y)$$

# Метод глобальной оптимизации

Общая схема характеристического метода: пусть имеется  $k$  результатов испытаний, далее:

- Шаг 1. Упорядочить поисковую информацию по возрастанию координат точек испытаний
- Шаг 2. Вычислить для каждого интервала величину  $R(i)$ , называемую характеристикой.
- Шаг 3. Выбрать интервал номер  $t$  с наибольшей характеристикой и провести в нем испытание:

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

Критерий остановки:

$$x_t - x_{t-1} < \varepsilon$$

Формулы для вычисления характеристик:

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

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

Правила выбора следующей точки:

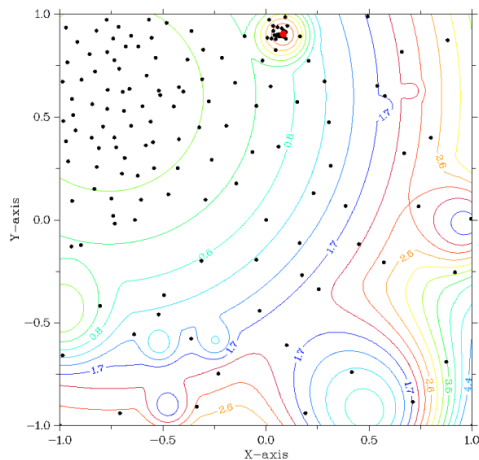
$$x_{k+1} = \frac{x_t + x_{t-1}}{2}, t = 1, t = k+1$$

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

# Класс тестовых задач

Генератор GKLS:

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





Способы использования локального поиска (метод Хука-Дживса):

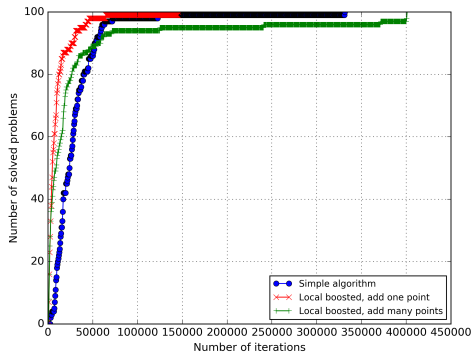
1. Запуск из лучшей найденной точки после окончания работы АГП.
2. Запуски из текущих лучших точек в процессе работы АГП.

Стратегии сохранения информации (для п. 2):

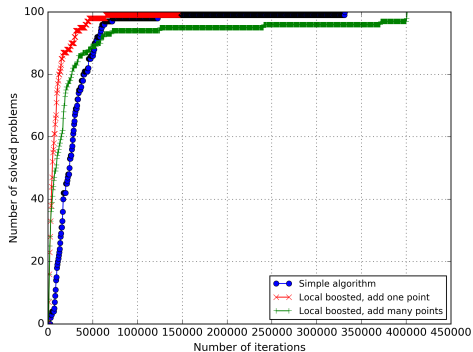
- ▶ добавлять только лучшие точки
- ▶ добавлять в поисковую информацию все точки

# Использование методов локальной оптимизации

Результаты применения различных стратегий сохранения информации:



GKLS 4d Simple



GKLS 5d Simple

## Смешанный алгоритм

Метод является модификацией АГП. Каждый интервал имеет две характеристики  $R(i)$  и  $R^*(i)$ .

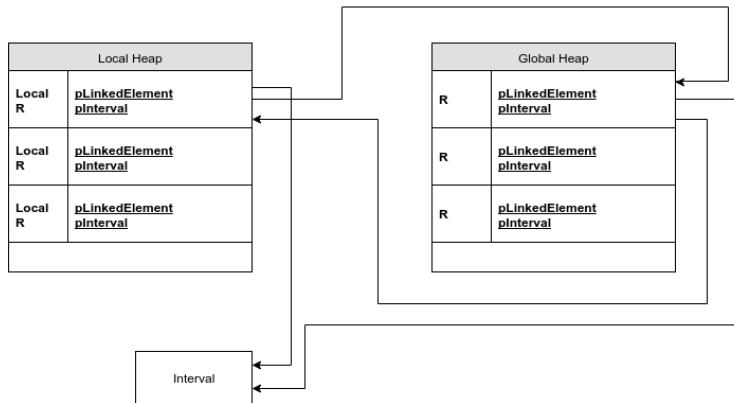
$$R^*(i) = \frac{R(i)}{\sqrt{(z_i - z^*)(z_{i-1} - z^*)/\mu + 1.5^{-\alpha}}}$$

Для эффективной реализации АГП используется приоритетная очередь интервалов. Ключ –  $R(i)$ . Для смешанного АГП – две связанные очереди. Операции с очередями:

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

# Смешанный алгоритм

Схема перекрёстных ссылок на элементы очередей:

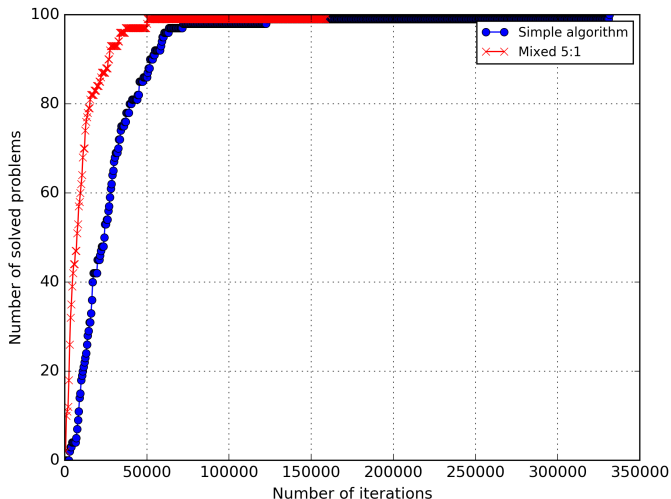


## Смешанный алгоритм

Алгоритм обмена элементов при погружении/всплытии:

```
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



Операционные характеристики обычного и смешанного АГП на классе GKLS 4d Simple

## Пример прикладной задачи

$$\dot{x} = (A + B_u \Theta)x + B_v v, x(0) = 0$$

Выходы системы описываются выражениями:

$$z_k = (C_k + B_u \Theta), k = \overline{1, N}$$

Критерии оптимальности:

$$J_k(\Theta) = \sup_{v \in L_2} \frac{\max_{1 \leq i \leq n_k} \sup_{t \geq 0} |z_k^{(i)}(\Theta, t)|}{\|v\|_2} \rightarrow \min_{\Theta}$$

Ограничение на устойчивость системы:

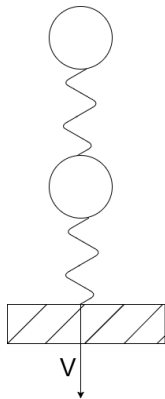
$$g_0(\Theta) = \min_j \operatorname{Re}(\lambda_j(A + B_u \Theta)) < 0$$

Многокритериальная задача сводится к скалярной методом главного критерия или с помощью свёртки Гермейера.

## Пример прикладной задачи

Рассматривается система из  $n$  материальных точек, связанных упруго-диссипативными элементами. При  $n = 2$ :

$$\begin{cases} \ddot{\xi}_1 = -\beta(\dot{\xi}_1 - \dot{\xi}_2) - \xi_1 + \xi_2 + u + v \\ \ddot{\xi}_2 = -\beta(\dot{\xi}_2 - \dot{\xi}_1) - \xi_2 + \xi_1 + v \\ \xi_1(0) = \xi_2(0), \dot{\xi}_1(0) = \dot{\xi}_2(0) = 0 \end{cases}$$





## Пример прикладной задачи (результаты)

Постановка задачи в методе главного критерия:

$$J_2(\Theta^*) = \min\{\Theta : J_1(\Theta) < \varepsilon\}$$

Решается серия задач при различных значениях  $\varepsilon$ .

