Стохастическое программирование и мультиагентные технологии

Олег Николаевич Граничин

Санкт-Петербургский государственный университет

1-я лекция на VIII Традиционной школе Б.Т.Поляка
“Управление, информация и оптимизация”
14 июня 2016

План

1. Введение (Introduction)
2. Рандомизированные алгоритмы в управлении и при извлечении знаний (Randomized algorithms in automatic control and data mining)
   - Оценивание при произвольных внешних помехах (Estimation under arbitrary external noise)
   - Оптимизация функционала среднего риска (Mean-Risk optimization)
   - Рандомизированные алгоритмы стохастической аппроксимации (Randomized stochastic approximation, SPSA)
   - Рандомизированные алгоритмы и квантовые вычисления (Randomized algorithms and quantum computing)
   - Приложения (Applications)
3. Мультиагентные технологии (Multiagent technologies)
   - Консенсусное управление (Consensus control)
   - Планирование группы БПЛА (UAV Group Planning)

План

1. Введение (Introduction)
2. Рандомизированные алгоритмы в управлении и при извлечении знаний (Randomized algorithms in automatic control and data mining)
   - Оценивание при произвольных внешних помехах (Estimation under arbitrary external noise)
   - Оптимизация функционала среднего риска (Mean-Risk optimization)
   - Рандомизированные алгоритмы стохастической аппроксимации (Randomized stochastic approximation, SPSA)
   - Рандомизированные алгоритмы и квантовые вычисления (Randomized algorithms and quantum computing)
   - Приложения (Applications)
3. Мультиагентные технологии (Multiagent technologies)
   - Консенсусное управление (Consensus control)
   - Планирование группы БПЛА (UAV Group Planning)

Цифровые технологии в сравнении с “механическими”

Сдвиги в технологиях XXI в.

Стадии	Ключевые ресурсы	Распре-деление	Область	Факторы успеха
Урбанизация Индустриальная экономика Технологии массовых продуктов	Капитал	Транспорт	Региональная	Экономия при масштабировании
↓	↓	↓	↓	↓
Глобализация общества Экономика знаний Цифровые технологии	Знания / Информация	Цифровые сети	Глобальная	**Адаптивность**

- Степень сложности в экономике и обществе растет

Следствия роста сложности

Для живущих и работающих в сложных условиях частых непредсказуемых разрушительных событий:

- хорошо структурированные общества и бизнесы не могут реагировать быстро на непредвиденные события
- неопределенности генерируют тревогу
- неопределенности дают возможности

Принятие решений в реальном времени

Критические факторы:

- скорость — надо принять и реализовать решение до наступления следующего “разрушительного” события
- интеллект (природный и/или искусственный) — надо достичь цели в условиях неопределенности

Критическая технологии:

- Стохастическое программирование и мультиагентные технологии для поддержки (или замены) процесса принятия решения человеком

Интеллект

Интеллект — это то, что:

- отличает человека от компьютера, действующего строго по заложенной в него программе
- позволяет человеку ориентироваться в сложной обстановке, иметь дело с нечетко поставленными задачами, адаптироваться к меняющимся условиям

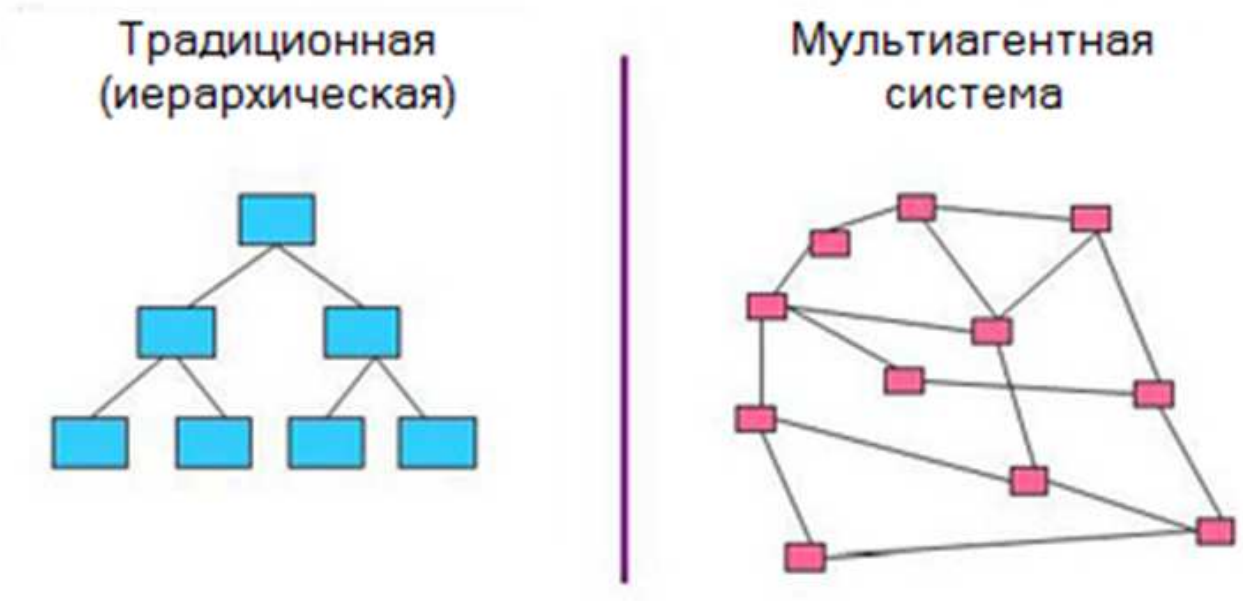
Стохастическое программирование

Стохастичность — греческое слово ($\sigma\tau o\chi\alpha\sigma\tau\iota\kappa\acute{\epsilon}\zeta$)

- целиться, метить (целиться во что (в кого)-либо)
- стремиться к тому, чтобы судьями были самые влиятельные люди
- применяться, приспособляться
- умозаключать, судить, догадываться, разгадывать (догадываться о том, что требуется; заключать на основании чего-либо; путем догадок)

Стохастическое программирование — подход в математическом программировании, позволяющий учитывать неопределённость в оптимизационных моделях

Мультиагентные системы (начиная с 1990х)



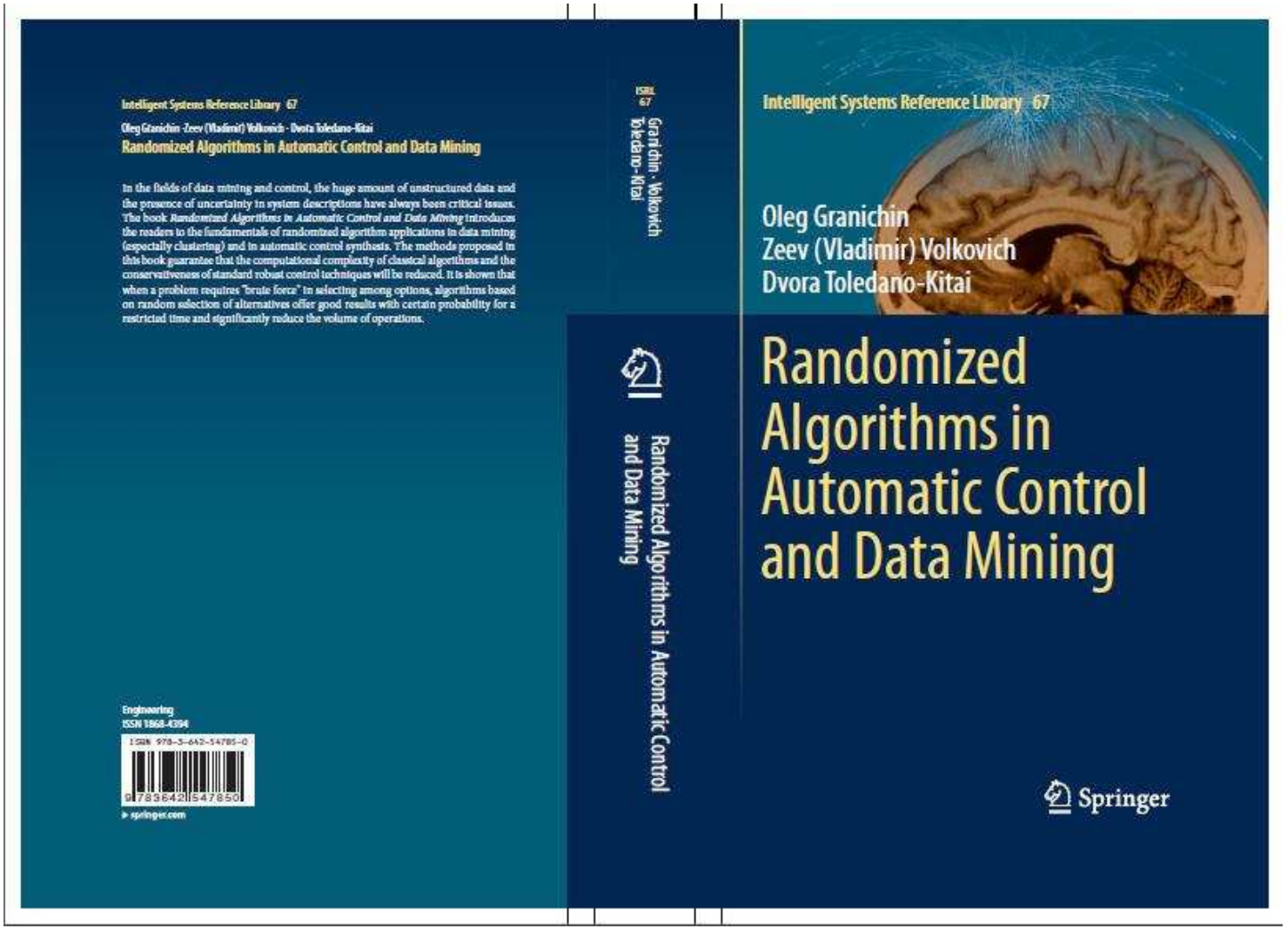
В качестве основы были взяты достижения в следующих областях:

- системы искусственного интеллекта (Artificial Intelligence)
- параллельные вычисления (Parallel Computing)
- распределенное решение задач (Distributed Problem Solving)

План

1. Введение (Introduction)
2. Рандомизированные алгоритмы в управлении и при извлечении знаний (Randomized algorithms in automatic control and data mining)
   - Оценивание при произвольных внешних помехах (Estimation under arbitrary external noise)
   - Оптимизация функционала среднего риска (Mean-Risk optimization)
   - Рандомизированные алгоритмы стохастической аппроксимации (Randomized stochastic approximation, SPSA)
   - Рандомизированные алгоритмы и квантовые вычисления (Randomized algorithms and quantum computing)
   - Приложения (Applications)
3. Мультиагентные технологии (Multiagent technologies)
   - Консенсусное управление (Consensus control)
   - Планирование группы БПЛА (UAV Group Planning)

Randomized Algorithms in Automatic Control and Data Mining

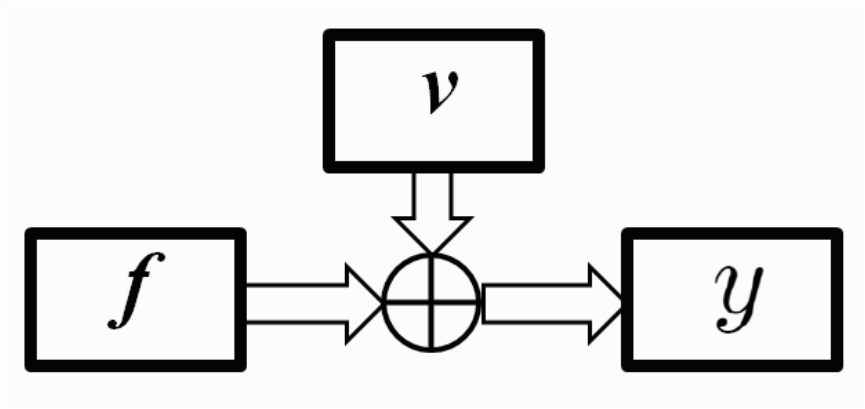


Springer, 2015

Произвольные внешние помехи

Если сигнал f регистрируется с помехой v, тогда "мгновенные" наблюдения y_t:

$$y_t = f_t + v_t. \tag{1}$$



- $v = 0$.
- $v \approx 0$.
- $v_t \to 0$ as $t \to \infty$.
- v_t, $t = 1, \ldots, T$, — i.i.d., $Ev_t = 0$ и $\sigma_v^2 < \infty$
- v_t — произвольные внешние помехи

Усреднение данных

Если регистрирующая аппаратура усредняет сигналы f_t, поступающие в моменты времени $t = 1, \ldots, T$, тогда

$$y = \frac{1}{N}\sum_{t=1}^{T} f_t + \frac{1}{N}\sum_{t=1}^{T} v_t \tag{2}$$

Если v_t, $t = 1, \ldots, T$, — i.i.d., $Ev_t = M_v$ и $\sigma_v^2 < \infty$, тогда

$$\mathrm{Prob}\left\{ |\frac{1}{N}\sum_{t=1}^{T} v_t - M_v| > \varepsilon \right\} \to 0 \ \ as \ \ t \to \infty.$$

Следовательно, можно использовать оценку

$$\hat{f} = \frac{1}{N}\sum_{t=1}^{T} y_t - M_v.$$

Возможно ли оценивание при произвольных помехах?

Модернизируем задачу, добавив в модель наблюдений "управляемый вход" u. Пусть сигнал f в момент времени t определяется "текущим" входом u_t и некоторым неизвестным параметром $\theta^\star$ (неизвестный коэффициент усиления/ослабления входов):

$$f_t = u_t \theta^\star. \tag{3}$$

Задача: *требуется оценить* неизвестный параметр $\theta^\star \in \mathbb{R}$ по последовательности входов и выходов $\{u_t, y_t\}$ без существенных ограничений на последовательность внешних помех $\{v_t\}$.

Схема задачи

Модель наблюдений (1) можно переписать в виде:

$$y_t = u_t \theta^\star + v_t. \tag{4}$$

Можно

- выбирать входы (управления) u_t, $t = 1, 2, ..., T$,
- измерять выходы y_t (см. Рис. 1).

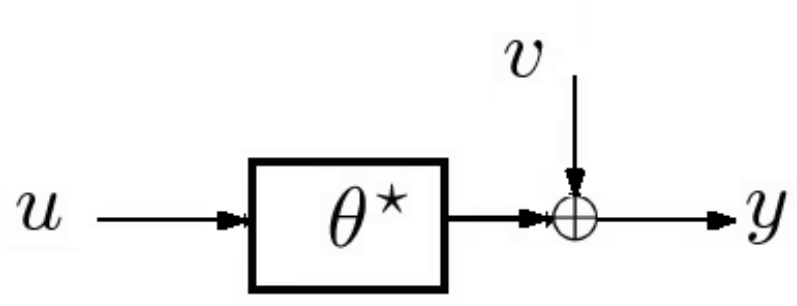
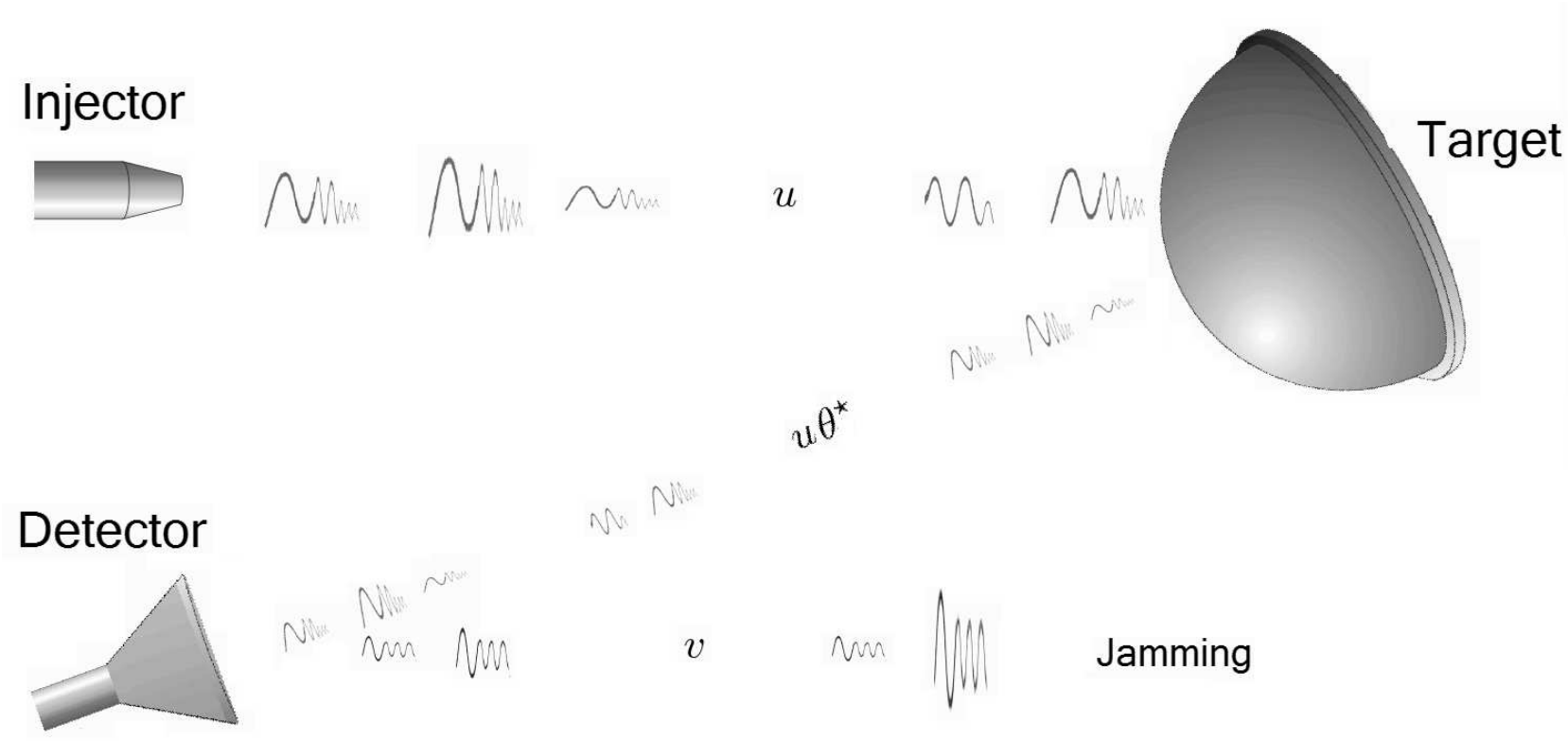


Рис.: Модель наблюдений.

При $u_t \equiv 1$ получаем обычную проблему оценивания неизвестного параметра $\theta^\star$ по наблюдениям с помехами.

Источник, мишень и детектор отраженного сигнала



Алгоритм оценивания $\theta^\star$

1. Выбор входа (управления) u_t и подача его на вход системы.
2. Получение отчета от системы y_t.
3. Оценивание параметра $\theta^\star$ на основе полученных данных u_t, y_t (вычисление, например, оценки $\widehat{\theta}_t$ или множества $\widehat{\Theta}_t$, содержащего $\theta^\star$).
4. Повторение шагов 1–3.

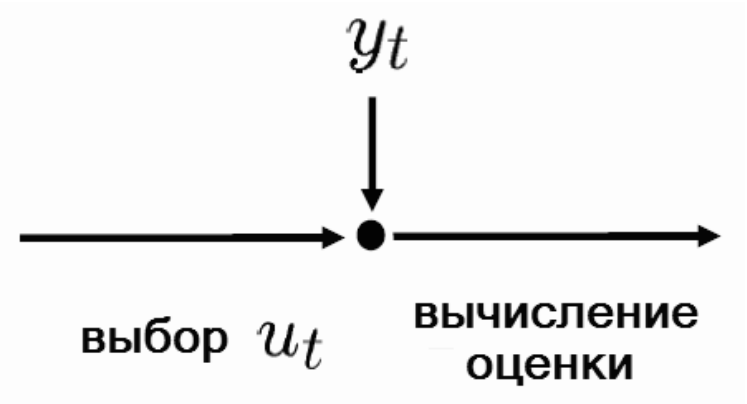


Рис.: Модель детерминированного алгоритма.

Детерминированный алгоритм

Definition

Алгоритм называется *детерминированным* если все его шаги, определяемые пользователем, задаются детерминированными правилами, использующими результаты предшествующих шагов, и выдают новые данные (результаты), которые можно использовать на последующих шагах алгоритма.

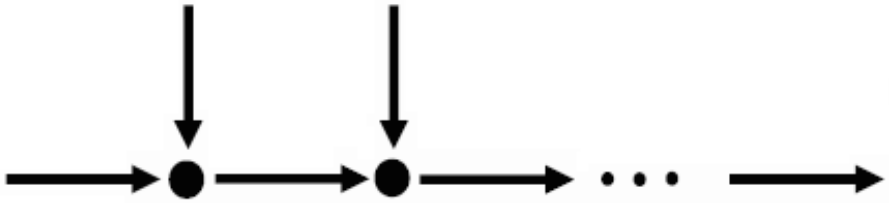


Рис.: Модель детерминированного алгоритма.

Детерминированные алгоритмы часто не справляются с решением задач !

- В теории и на практике возникают существенные трудности при аналитическом исследовании “сложных” систем.
- Во многих практических приложениях традиционные детерминированные методы часто не справляются с решением задач, когда система сложная.
- В частности, это приводит к NP-сложным задачам.

При произвольных внешних помехах нет детерминированного алгоритма оценивания!

Пусть $\theta^\star = 3$

$$\widehat{\theta}_t = \frac{1}{t}\sum_{i=1}^{t} y_i$$

Таблица

t	1	2	3	4	5	6	7
u_t	1	1	1	1	1	1	1
$v_t = rand() - 0.5$							
y_t	2.9	2.8	3.2	3.3	2.6	3.4	2.7
$\widehat{\theta}_t$	2.9	2.85	2.97	3.05	2.96	3.03	2.99
$v_t = rand() - 0.5 + M_v,\ M_v = 1$							
y_t	3.9	3.8	4.2	4.3	3.6	3.9	4.2
$\widehat{\theta}_t$	3.9	3.85	3.97	4.05	3.96	4.03	3.99

Рандомизированные алгоритмы

Рандомизация — мощное средство для решения целого ряда проблем, которые кажутся нерешаемыми детерминированными методами.

Definition

Рандомизированным называется алгоритм, в котором выполнение пользователем одного или несколько шагов основано на случайном правиле (т. е. среди многих детерминированных правил одно выбирается пользователем случайно в соответствии с вероятностью P).

“Обогащение” наблюдений

Рассмотрим рандомизированное правило выбора входа на первом шаге

$$u_t = \begin{cases} +1, \text{ with probability } \frac{1}{2}, \\ -1, \text{ with probability } \frac{1}{2}. \end{cases} \tag{5}$$

т. е. на первом шаге случайно выбирается один из 2^7 возможных наборов входов (управлений). На втором шаге по известным парам значений (u_t, y_t) формируем величины

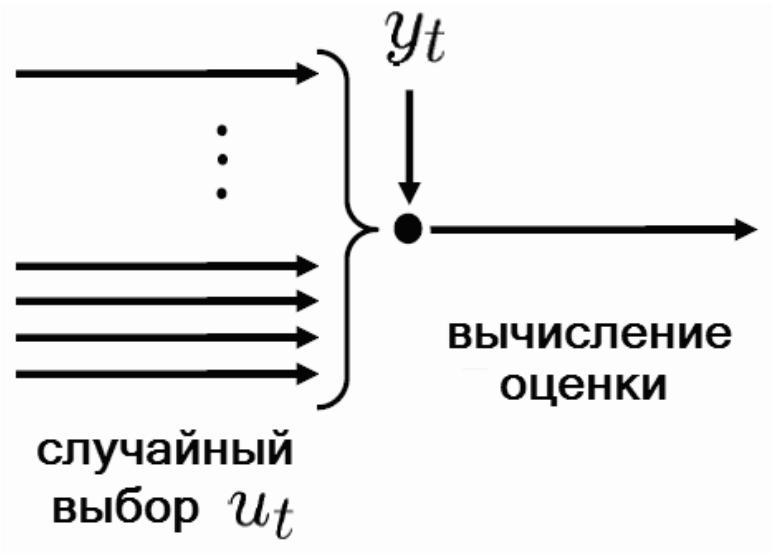
$$\tilde{y}_t = u_t \cdot y_t.$$

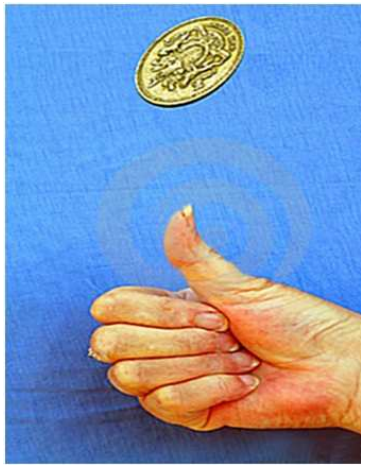
Для “новой” последовательности наблюдений справедлива похожая на (4) модель

$$\tilde{y}_t = \tilde{u}_t \cdot \theta^\star + \tilde{v}_t,$$

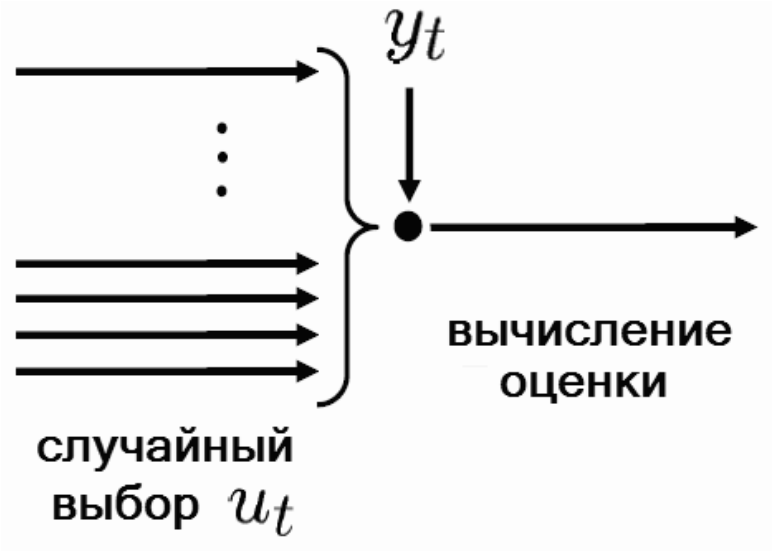
в которой $\tilde{u}_t = u_t^2 \equiv 1$ и $\tilde{v}_t = u_t \cdot v_t$.

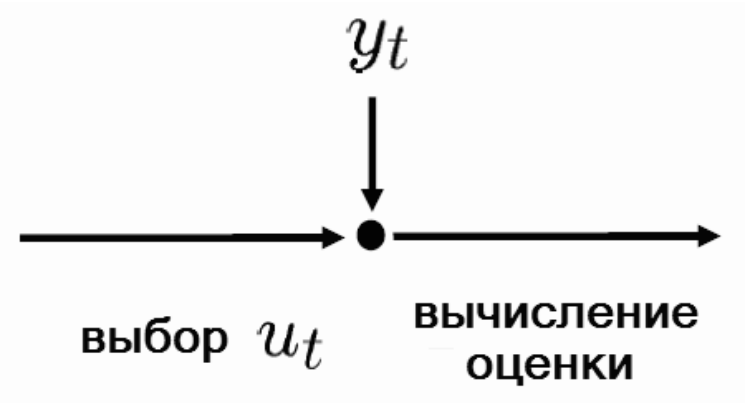
Схема рандомизированного алгоритма





Два типа алгоритмов





Результаты моделирования

$$\theta^\star = 3, \qquad \widehat{\theta}_t = \frac{1}{t}\sum_{i=1}^{t} \tilde{y}_i = \frac{1}{t}\sum_{i=1}^{t} u_i y_i$$

Таблица

t	1	2	3	4	5	6	7
u_t	-1	1	-1	1	1	1	-1
$v_t = rand() - 0.5 + m,\ m = 1$							
y_t	-2.1	3.8	-1.8	4.3	3.6	4.4	-2.3
$\tilde{u}_t$	1	1	1	1	1	1	1
$\tilde{y}_t$	2.1	3.8	1.8	4.3	3.6	4.4	2.3
$\widehat{\theta}_t$	2.1	2.95	2.57	3.00	3.12	3.33	3.19

$$\forall t,\ \forall \varepsilon > 0\ \ Prob\{|\widehat{\theta}_t - \theta^\star| \geq \varepsilon\} \leq \frac{1}{t}\frac{\mathrm{E}\{v_t^2\}}{\varepsilon^2} + o\left(\frac{1}{t}\right).$$

[Granichin, IEEE TAC, 2004]

Неасимптотический результат

Пусть $T = 7$ — конечное число наблюдений, v_t — произвольная внешняя помеха. LSCR метод (M. Campi, EJC, 2010):

1. Фиксируем $M = 8$ и $q = 1$. Выберем случайно семь $(= M - 1)$ различных наборов по четыре индекса $T_1, \ldots, T_7$.
2. Вычислим семь частичных сумм $\bar{s}_i = \frac{1}{4} \sum_{j \in T_i} \bar{y}_j$, $i = 1, \ldots, 7$.
3. Сформируем доверительный интервал

$$\widehat{\Theta} = [\min_{i \in 1:7} \bar{s}_i; \max_{i \in 1:7} \bar{s}_i],$$

содержащий $\theta^\star$ с вероятностью $p = 75\%$ $(= 1 - 2 \cdot q/M)$.

Доверительный интервал

Для смоделированных ранее данных $\{(u_t, y_t)\}$ получаем:

Таблица

i	T_i	$\bar{s}_i$
1	$\{2, 3, 4, 5\}$	3.375
2	$\{1, 3, 4, 6\}$	3.15
3	$\{2, 3, 5, 6\}$	3.4
4	$\{1, 2, 6, 7\}$	3.15
5	$\{1, 4, 5, 7\}$	3.075
6	$\{2, 3, 5, 7\}$	2.875
7	$\{1, 4, 6, 7\}$	3.275

и, следовательно,

- интервал $\widehat{\Theta} = [2.875; 3.4]$ содержит с вероятностью $p = 75\%$ неизвестный параметр $\theta^\star$.

Вывод: рандомизация входов дает вполне разумные результаты при произвольных внешних помехах.

Рандомизация стратегии управления

Amelin and Granichin IEEE TAC 2016

$$A_\star(z^{-1})y_t = B_\star(z^{-1})u_t + v_t,\ t = 1, 2, \ldots, T$$

$A_\star(\lambda) = 1 + a_\star^{(1)}\lambda + \cdots + a_\star^{(n_a)}\lambda^{n_a}$, $B_\star(\lambda) = b_\star^{(l)}\lambda^l + b_\star^{(l+1)}\lambda^{l+1} + \cdots + b_\star^{(n_b)}\lambda^{n_b}$

$\tau_\star = col(a_\star^{(1)}, \ldots, a_\star^{(n_a)}, b_\star^{(l)}, \ldots, b_\star^{(n_b)})$ — вектор параметров

$$\text{Goal: } \overline{\lim_{t\to\infty}}|y_t| \to \min,\ \sup_t |y_t| + |u_t| < \infty$$

$s \in \mathbb{N},\ T = s \cdot N_\Delta$

$$u_{sn+i-l} = \begin{cases} \Delta_n + \bar{u}_{sn-l}, & i = 0, \\ \bar{u}_{sn+i-l}, & i = 1, 2, \ldots, s-1, \end{cases} \quad n = 0, \ldots, N_\Delta - 1$$

$$\bar{u}_t = \mathscr{U}_t(y_t, y_{t-1}, \ldots, \bar{u}_{t-1}, \ldots),\ t \geq 0,\ \bar{u}_{-k} = 0,\ k > 0$$

Δ_n — измеримое пробное возмущение

$$\text{Например,} \quad \Delta_n = \begin{cases} +1, \text{with probability} \frac{1}{2}, \\ -1, \text{with probability} \frac{1}{2}. \end{cases}$$

Перепараметризация

Перепишем модель (6) в виде линейной регрессии

$$y_{sn+k-1} = \Delta_n \theta_\star^{(k)} + \bar{v}_{sn+k-1} = \Delta_n \theta_\star^{(k)} + \sum_{i=0}^{k-1} \theta_\star^{(k-i)} \bar{u}_{sn-l+i} + \ldots, \qquad (6)$$

с входамиΔ_n и регрессорами $\theta_\star^{(k)}$, $n = 1, \ldots, N_\Delta$, $k = 1, \ldots, s$.

$$\theta_\star = \theta(\tau_\star),\ \theta(\tau) = \mathbb{A}^{-1}(\tau)\mathbb{B}(\tau), \qquad (7)$$

$$\mathbb{A} = \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 \\ a_\star^{(1)} & 1 & \ldots & 0 & 0 \\ a_\star^{(2)} & a_\star^{(1)} & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & a_\star^{(n_a)} \ldots & a_\star^{(1)} & 1 \end{pmatrix},\ \mathbb{B} = \begin{pmatrix} b_\star^{(l)} \\ \vdots \\ b_\star^{(n_b)} \\ \vdots \\ 0 \end{pmatrix}.$$

(Granichin, Fomin, ARC, 1986)

Заметим, что Δ_n и $\bar{v}_{sn+k-1}$ независимы, если пользователь может выбирать Δ_n.

Как выбрать s?

A2. $\qquad s : \exists \;\; \tau(\theta) = \theta^{-1}(\tau)$

$s = n_a + n_b$, если полиномы $A_\star(\lambda)$ и $B_\star(\lambda)$ взаимно просты

Пример перепараметризации Рассмотрим ОУ второго порядка

$$y_t + a_\star^{(1)} y_{t-1} + y_{t-2} = b_\star^{(1)} u_{t-1} + 1.6 u_{t-2} + v_t, \tag{8}$$

$t = 1, 2, \ldots, N$, с неизв. коэффициентами $a_\star^{(1)}$ и $b_\star^{(1)} \neq 0$. Обозначим

$$\tau_\star = col(a_\star^{(1)}, b_\star^{(1)}).$$

Пусть $s = 2$, вектор $\theta_\star$, вычисленный по обратной функции $\tau(\theta)$, равен

$$\theta_\star = \begin{pmatrix} b_\star^{(1)} \\ 1.6 - a_\star^{(1)} b_\star^{(1)} \end{pmatrix} \in \mathbb{R}^2, \;\; \tau(\theta) = \begin{pmatrix} \frac{1.6 - \theta^{(2)}}{\theta^{(1)}} \\ \theta^{(1)} \end{pmatrix}.$$

Алгоритм. Leave-out Sign-dominant Correlation Regions (LSCR) method [M. Campi and E. Weyer, TAC, 2010]

1. $\widehat{y}_{sn+k-1}(\theta) = \Delta_n \theta^k + \sum_{i=0}^{k-1} \theta^{k-i} \bar{u}_{sn+k-l-i}$.
2. $\varepsilon_t(\theta) = y_t - \widehat{y}_t(\theta),\ t = 1, \ldots, N$.
3. $f_{sn+k-1}(\theta) = \mathrm{sign}[\Delta_n \varepsilon_{sn+k-1}(\theta)],\ n = 0, \ldots, N_\Delta - 1, k = 1, \ldots, s$.
4. Выбираем $M > 2s$ и разные случайные двоичные строки $(h_{i,1}, \ldots, h_{i,N}),\ i = 0, \ldots, M-1$.
   Вычисляем $g_i^k(\theta) = \sum_{n=0}^{N_\Delta - 1} h_{i,ns+k} \cdot f_{ns+k-1}(\theta),\ i = 0, \ldots, M-1$,
5. Выбираем $q \in [1; M/2s]$ и для $k = 1, \ldots, s$ строим области:

$$\widehat{\Theta}^{(k)} = \{\theta : \text{ at least } q \text{ of } g_i^k(\theta) \text{ are } > 0 \text{ and at least } q \text{ are } < 0\}.$$

$$\widehat{\Theta} = \bigcap_{k=1}^{s} \widehat{\Theta}^{(k)}. \tag{9}$$

Теоремы

Theorem

Пусть условие **A1** *выполнено. Рассмотрим* $k \in \{1, 2, \ldots, s\}$ *и допустим, что* $Prob(g_i^k(\theta_\star) = 0) = 0$. **Тогда**

$$Prob\{\theta_\star \in \widehat{\Theta}^{(k)}\} = 1 - 2q/M, \tag{10}$$

где M, q *и* $\widehat{\Theta}^{(k)}$ *из пп. 4 и 5.*

Theorem

$$Prob\{\theta_\star \in \widehat{\Theta}\} \geq 1 - 2sq/M. \tag{11}$$

Лемма [M. Campi and E. Weyer, TAC, 2010]

Fix $k \in [1, \ldots, s]$. ПустьH стохастическая $M \times N_\Delta$ матрица с элементами $h_{i,ns+k}$, $i = 0, 1, \ldots, M-1$, $n = 0, \ldots, N_\Delta - 1$, из п. 4 алгоритма LSCR, и пусть $\eta = col(\eta_1, \ldots, \eta_{N_\Delta})$ — вектор независимый с H, состоящий из взаимно независимых случайных величин симметрично распр. вокруг нуля.
Возьмем $i \in [0, M-1]$, пусть H_i — $M \times N$ матрица со строками равными i-ой строке H. **Тогда**, $H\eta$ и $(H - H_i)\eta$ имеют одинаковое M-размерное распределение.

Основная идея доказательства

Обозначим $\eta_n := \Delta_{n-1}\varepsilon_{(n-1)s+k-1}(\theta_\star)$.
Для корреляции между η_i and η_j, $i > j$,:

$$E[\eta_i\eta_j] = E[\Delta_{i-1}]E[\varepsilon_{(i-1)s+k-1}(\theta_\star)\Delta_{j-1}\varepsilon_{(j-1)s+k-1}(\theta_\star)] = 0$$

$E[\Delta_{i-1}] = 0$ ($\eta_1, \ldots, \eta_{N_\Delta}$ взаимно независимы).
Возьмем $g_{\bar{i}}^{(k)}(\theta_\star)$ в r-ой позиции.

$$g_i^{(k)}(\theta_\star) - g_{\bar{i}}^{(k)}(\theta_\star) = \sum_{n=0}^{N_\Delta}(h_{i,ns+k} - h_{\bar{i},ns+k})\eta_n < 0$$

для $r-1$ выборов из $i \in [0, M-1]$.
По Лемме: $Prob\{$"$r-1$ entries of $(H - H_{\bar{i}})\eta$ are negative"$\} = Prob\{$"$r-1$ entries of $H\eta$ are negative"$\}$, и не зависит от $\bar{i}$.

Пример 1

$$y_t - 2y_{t-1} + y_{t-2} = b_\star u_{t-1} + 1.6u_{t-2} + v_t,\ t = 1, \ldots, 15,$$

$y_0 = y_{-1} = u_{-1} = 0$, $b_\star$ — неизвестный коэффициент, v_t — неизвестный внешний произвольный шум.

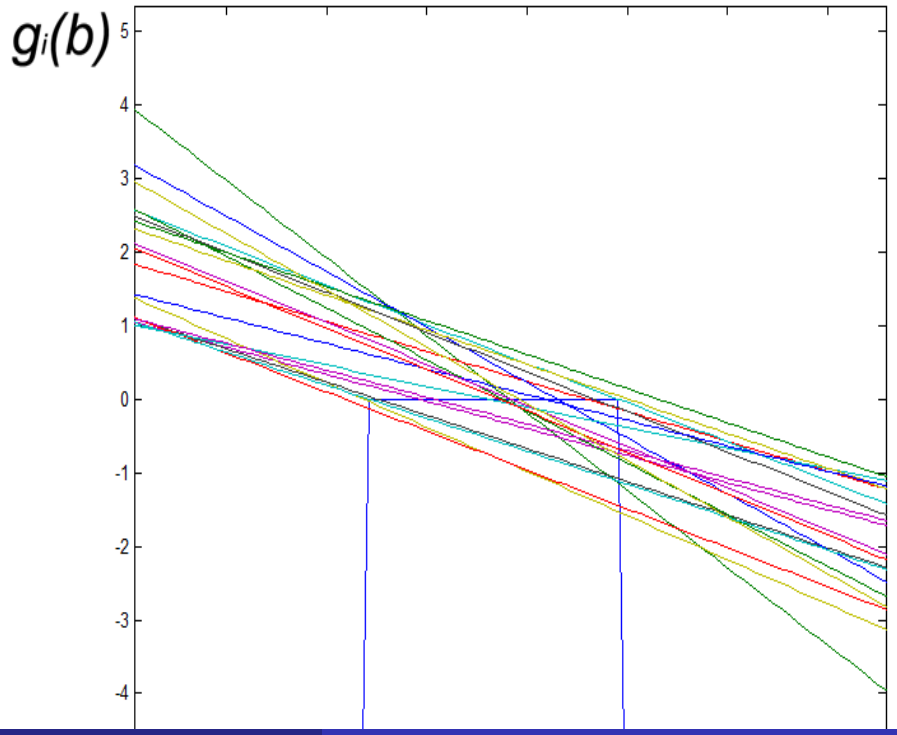
LSCR

$b_\star = 1$, $Ev_t = 0.5$, $\sigma_v = 0.1$

$$g_i(b) = \sum_{t=1}^{15} h_{i,t} \cdot \Delta_{t-1}\varepsilon_t(b),$$

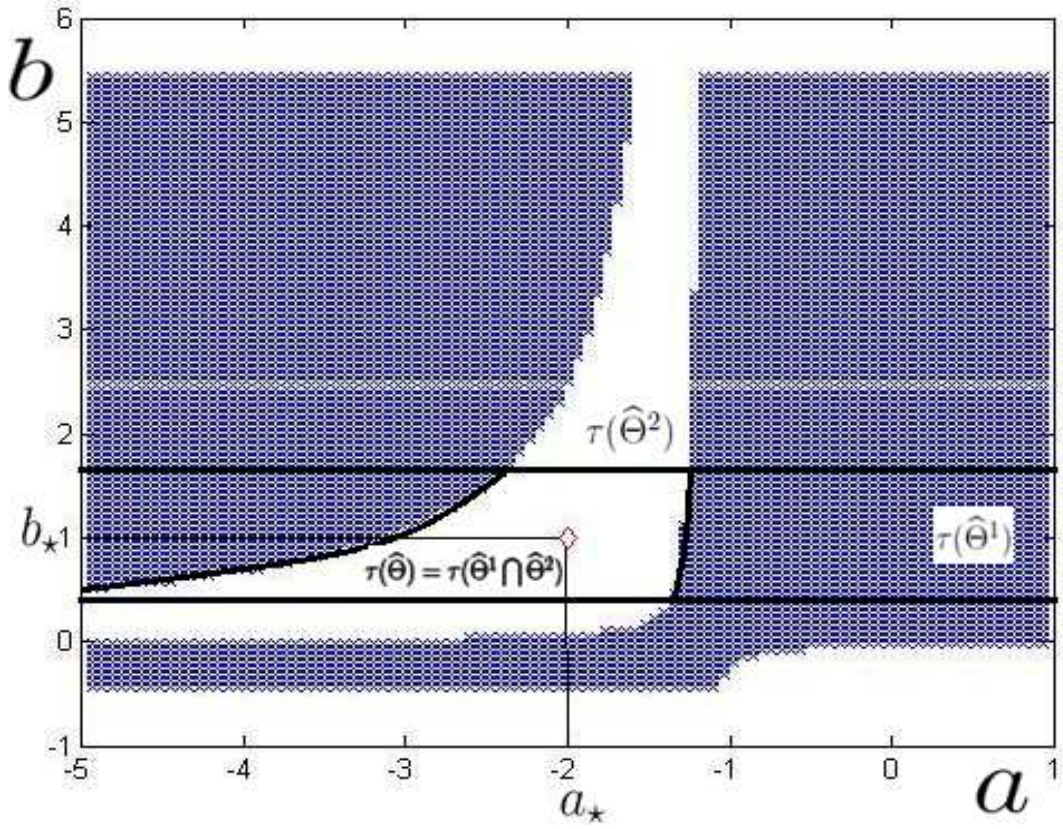
$$h_{i,t} \in \{0, 1\},\ i = 1, \ldots, 19.$$

Доверительный интервал $[0.834; 1.090]$

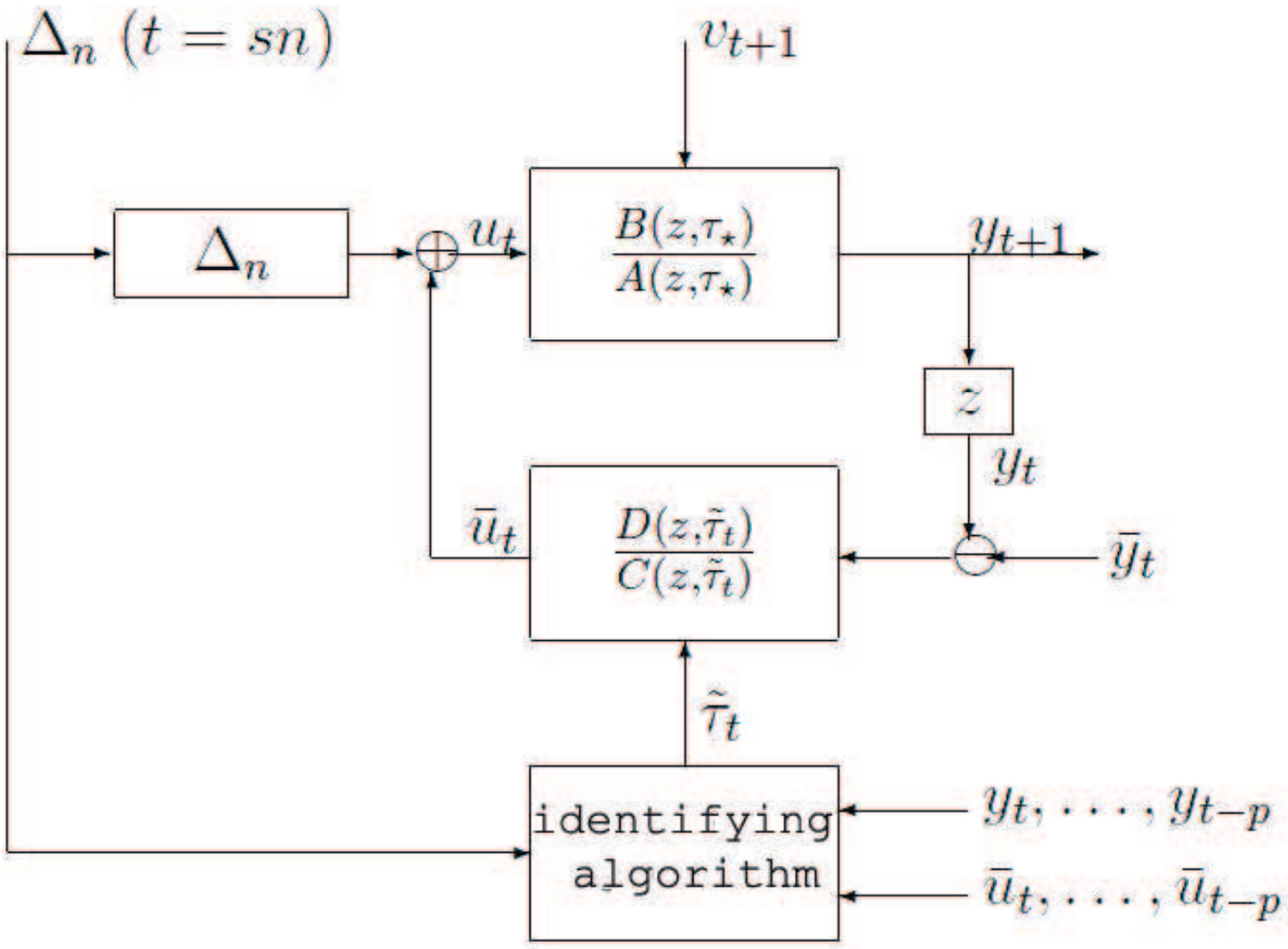


Пример 2

$$y_t + a_\star^1 y_{t-1} + y_{t-2} = b_\star^1 u_{t-1} + 1,6 u_{t-2} + v_t,\ t = 1, 2, \ldots, 960,$$
$$Prob\{\theta_\star \in \widehat{\Theta}^{(k)}\} = 1 - 2q/M,$$
$$95\% = (1 - 2 \cdot 2 \cdot 6/480) \cdot 100\%.$$



Блок-схема адаптивной системы управления



Рандомизированный и байесовский подходы

Альтернативный вероятностный подход — байесовский, при котором помехам v_t "приписывается" распределение вероятностей Q.
НО рандомизированный и байесовский подходы совершенно различны с практической точки зрения!!!
При байесовском подходе Q описывает вероятности значений v_t, т. е. выбор Q — часть модели задачи.
В противоположность, вероятность P в рандомизированном подходе выбирается искусственно. P существует только в нашем алгоритме, следовательно, нет традиционной проблемы "плохой модели" как это часто бывает с Q при байесовском подходе.

Рандомизация как способ сокращения вычислительной сложности

Общая идея:
Предположим, что детерминированный алгоритм тебует “огромного” набора вычислительных ресурсов для обработки всей доступной информации.
Тогда можно попробовать взять только часть информации и получить решение упрощенной задачи с частичной информацией.
В этом случае, однако, качества решения как у детерминированного алгоритма можно и не достичь, но рандомизированный подход позволят говорить о *вероятностно-успешных алгоритмах*, если удается получить решение с высокой вероятностью. Компромисс между гарантированным успехом и вычислительными возможностями получить разумный ответ за ограниченное время.

Рандомизация ...

1928-30 ...

- von Neumann (minimax theorem), Fisher (remove bias)

1950 ... 1975

- Metropolis, Ulam (method Monte-Carlo)
- Rastrigin, Kirkpatrick, Holland (random search, simulation annealing, genetic algorithm)

1980 ... 1999

- Granichin, Fomin, Chen, Guo (randomized control strategies)
- Polyak, Thzubakov, Luing, Guffi, Spall (fast algorithms)
- Granichin (arbitrary noise)
- Vadyasagar (randomized learning theory)

2000 ...

- Tempo, Campi, Calafiore, Dabbene, Polyak, Sherbakov *etc.* (probabilistic methods in a control syntheses, scenario approach)
- Candes, Donoho, Romberg, Tao (compressive sensing)

Замечательные свойства

- Существенное сокращение числа операций
- Снижение негативного влияния систематических ошибок
- Точность обычно не существенно зависит от размерности данных

Оптимизация функционала среднего риска

Пусть $\mathbf{x}_1, \mathbf{x}_2, \ldots \in \mathbb{R}^d$ — точки измерений

$$y_t = f(\mathbf{x}_t, \mathbf{w}_t) + v_t, \tag{12}$$

где $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$,
$\mathbf{w}_t$ — случайный вектор, v_n — произвольная внешняя помеха.
Задача стохастического программирования:

$$F(\mathbf{x}) = \int f(\mathbf{x}, \mathbf{w}) \mathrm{P}(d\mathbf{w}) \to \min_{\mathbf{x}}. \tag{13}$$

Конечно-разностный подход

Процедура Кифера-Вольфовица
Число наблюдений на итерации $M = 2d$

$$\widehat{\theta}_0 \in \mathbb{R}^d$$

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} - \frac{\alpha_n}{2\beta_n}(\mathbf{Y}_n^+ - \mathbf{Y}_n^-),$$

$$x_n^{(i,\pm)} = \widehat{\theta}_{n-1} \pm \beta_n e_i$$

$$\mathbf{Y}_n^{\pm} = \begin{pmatrix} f(x_n^{(1,\pm)}, w_n^{(1,\pm)}) + v_n^{(1,\pm)} \\ f(x_n^{(2,\pm)}, w_n^{(2,\pm)}) + v_n^{(2,\pm)} \\ \vdots \\ f(x_n^{(d,\pm)}, w_n^{(d,\pm)}) + v_n^{(d,\pm)} \end{pmatrix}$$

Рандомизированные алгоритмы стохастической аппроксимации

Сокращение числа наблюдений до 1 или 2 вместо $2d$ (!)

Алгоритм с одним измерением

$$\mathbf{x}_n = \widehat{\theta}_{n-1} + \beta_n \Delta_n, \;\; \Delta_n = \begin{pmatrix} \pm 1 \\ \pm 1 \\ \vdots \\ \pm 1 \end{pmatrix}$$

$$y_n = f(\mathbf{x}_n, \mathbf{w}_n) + v_n$$

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n} \mathscr{K}_n(\Delta_n) y_n$$

Алгоритм с двумя измерениями

$$\mathbf{x}_n^{\pm} = \widehat{\theta}_{n-1} \pm \beta_n^{\pm} \Delta_n, \; y_n^{\pm} = f(\mathbf{x}_n^{\pm}, \mathbf{w}_n^{\pm}) + v_n^{\pm}$$

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n^{+} + \beta_n^{-}} \mathscr{K}_n(\Delta_n)(y_n^{+} - y_n^{-})$$

Преимущества алгоритма

- Состоятельность при почти произвольных внешних помехах [Granichin, 1989, 1992, 2002]
- Асимптотически-оптимальная скорость сходимости [Polyak and Tsybakov, 1990]
- Минимальное число наблюдений на итерации [Spall, 1992, 1997]
- Применимость в задаче об отслеживании изменений параметров [Granichin et al., 2009, 2015]
- "Естественная" реализация на квантовом компьютере [Granichin, Sysoev, 2006]

Обоснование "псевдоградиентности" при произвольных внешних помехах

$$E\{\widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n}\Delta_n y_n | \mathscr{F}_{n-1}\} =$$

$$= \widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n}(E\{\Delta_n f(\mathbf{x}_n)|\mathscr{F}_{n-1}\} + E\{\Delta_n\}E\{v_n|\mathscr{F}_{n-1}\}) =$$

$$= \widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n}(E\{\Delta_n f(\widehat{\theta}_{n-1} + \beta_n\Delta_n)|\mathscr{F}_{n-1}\} \approx$$

$$\approx \widehat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n}(E\{\Delta_n f(\widehat{\theta}_{n-1}) + \frac{\beta_n\Delta_n\Delta_n^{\mathrm{T}}\nabla f(\widehat{\theta}_{n-1})}{2}|\mathscr{F}_{n-1}\} =$$

$$= \widehat{\theta}_{n-1} - \frac{\alpha_n}{2}\nabla f(\widehat{\theta}_{n-1})$$

Отслеживание изменений параметров (трекинг)

Пусть Ξ — множество, $\{f_\xi(\mathbf{x},\mathbf{w})\}_{\xi\in\Xi}$ — семейство дифференцируемых функций

$$y_t = f_{\xi_t}(\mathbf{x}_t,\mathbf{w}_t) + v_t$$

$$F_t(\mathbf{x}) = \int f_{\xi_t}(\mathbf{x},\mathbf{w}_t)\mathrm{P}(d\mathbf{w}_t)\mathrm{P}(d\xi_t) \to \min_{\mathbf{x}} \tag{14}$$

Granichin et. al. (2015, 2009) SPSA с посоянным размером шага $\alpha_n = \alpha > 0$

Пример:

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \zeta,\ \mathbf{x}_n \in \mathbb{R}^d$$

Рандомизированные алгоритмы и квантовые вычисления

Все известные эффективные квантовые алгоритмы выполняются по следующей схеме:

- сначала подготавливается “суперпозиция” входов
- потом происходит обработка входов и
- измерение результата

Выполнение алгоритма SPSA на квантовом компьютере

$$\mathbf{u} = \frac{1}{2^{\frac{d}{2}}} \sum_{\Delta_i \in \{-1,+1\}^d} |\widehat{\mathbf{x}} + \beta \Delta_i\rangle = \mathbf{H}_\beta |\widehat{\mathbf{x}}\rangle$$

$$\mathbf{U}_f |\mathbf{u}\rangle |0\rangle = \frac{1}{2^d} \sum_{\Delta_i \in \{-1,+1\}^d} |\widehat{\mathbf{x}} + \beta \Delta_i\rangle |f(\widehat{\mathbf{x}} + \beta \Delta_i)\rangle$$

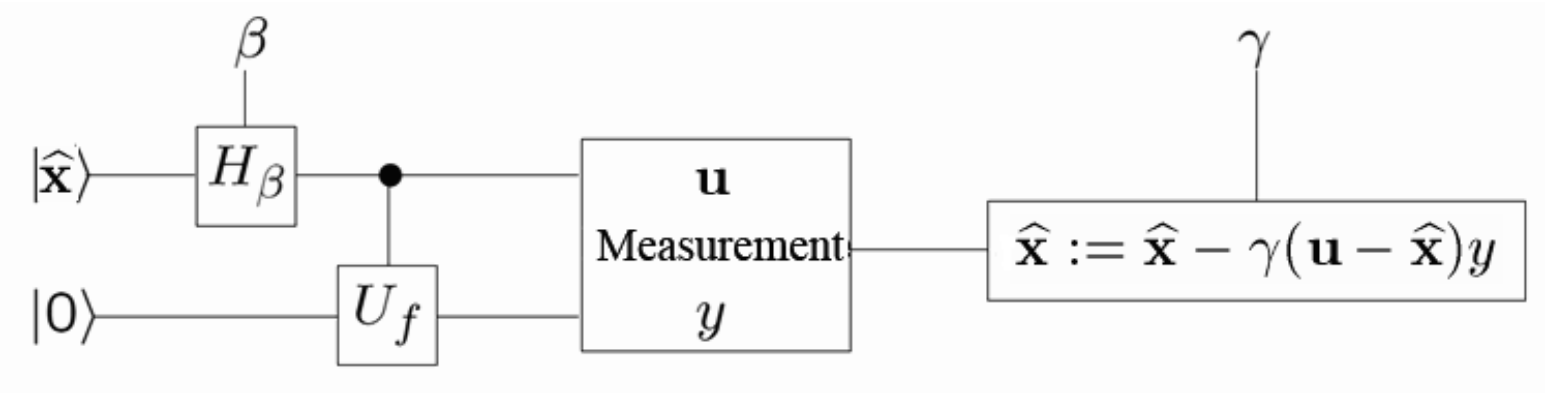


Рис.: Квантовая цепь для вычисления градиента “на лету”.

Близкие результаты

- Линейная регрессия
- Фильтрация
- Машинное обучение

Приложения

- Фотоэмиссионный экспиремент
- Парение БПЛА
- Адаптивная оптимизация сервера
- Балансировка загрузки

Рандомизация в фотоэмиссионном эксперименте

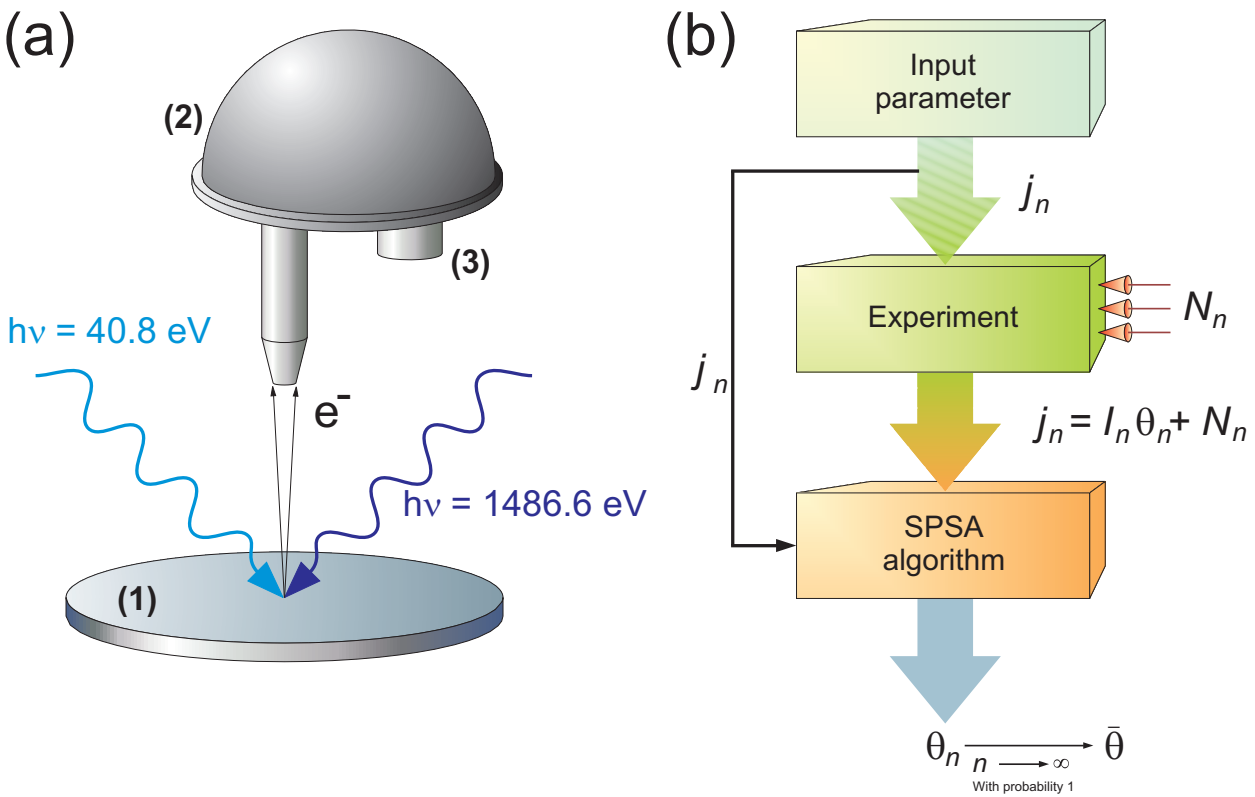


Рис.: (a) Схема экспиремента. Два источника фотонов He IIα и Al Kα, электроны анализируются фотоэлектронным спектрометром.

Результат

Granichin O., Molodtsov S. *et al.* Review of Scientific Instruments. 79, 036103. 2008.

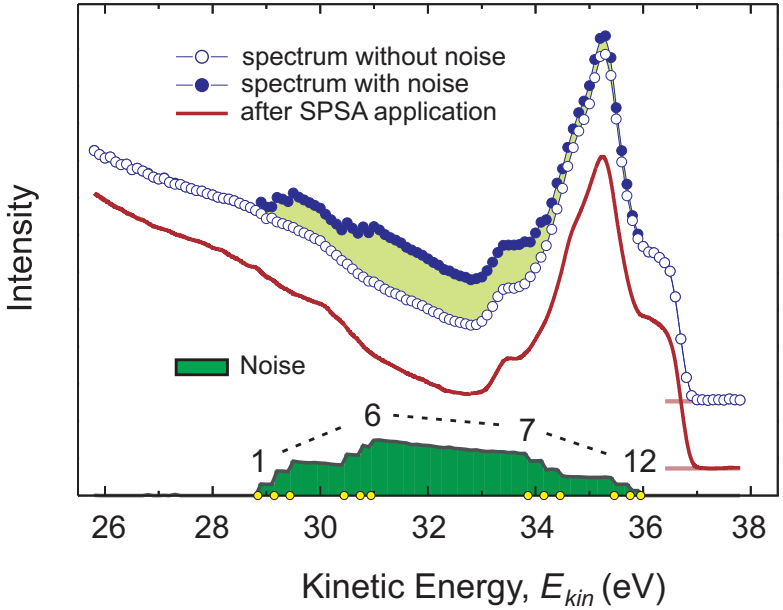


Рис.: Спектр W(110) полученный при He IIα без/с систематическим шумом и результат после 50 итераций прменения рандомизированного алгоритма.

Парение БПЛА

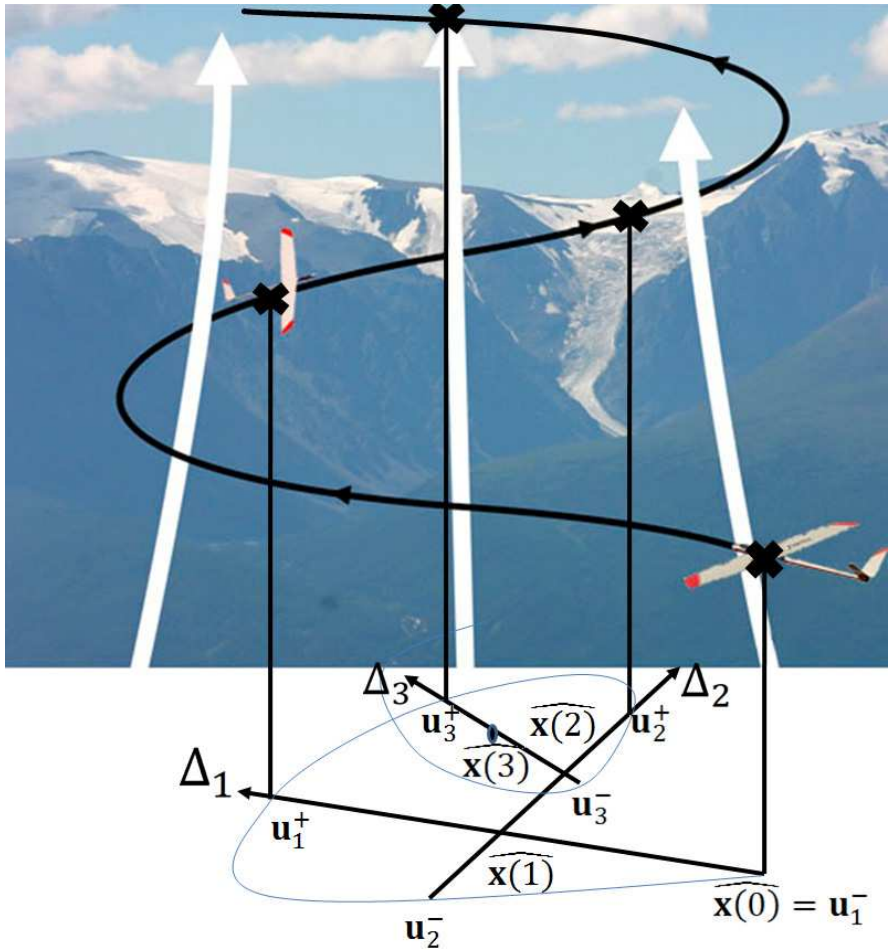
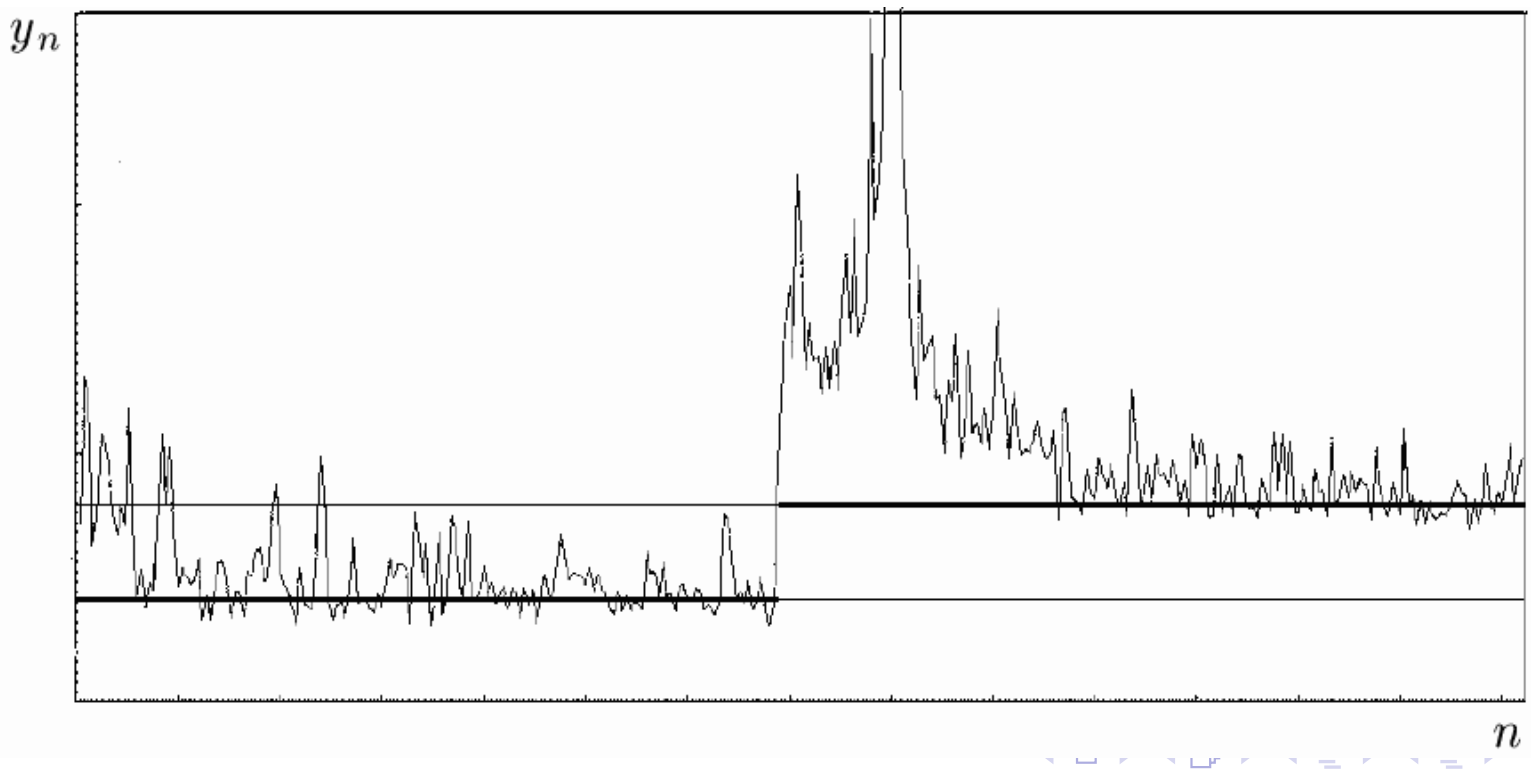


Рис.: Последовательность оценок и точек измерений.

Адаптивная оптимизация сервера

$$F(x) = q(x) + L(x) \equiv q(x) + \lim_T \frac{1}{T} \sum_{t=1}^{T} Ey_t(x) \to \min_x$$

$$y_n = \frac{t_n - t_{n-1}}{x} d_{load} + \frac{1}{N} \sum_{t_i^{out} \in [t_{n-1}, t_n]} (t_i^{out} - t_i^{in} - d_i)$$



Балансировка загрузки сети

Пусть $x_t^i = q_t^i / p_t^i$, где

- q_t^i — очередь заданий агента i в момент времени t
- p_t^i — производительность агента i в момент времени t

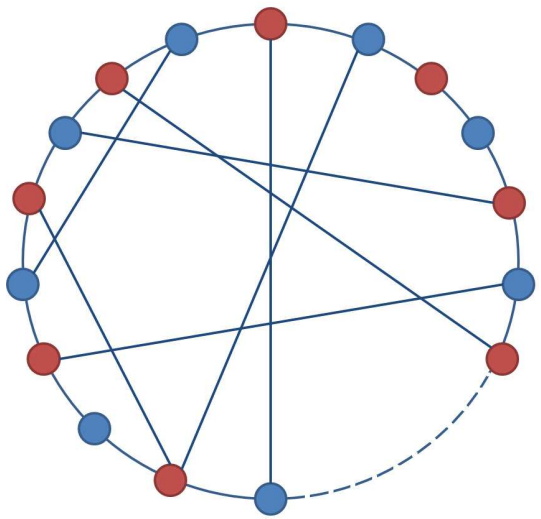
Динамическая модель:

$$x_{t+1}^i = x_t^i + f_t^i + u_t^i,\ i \in N = \{1, \ldots, n\},$$

или

$$q_{t+1}^i = q_t^i - p_t^i + z_t^i + p_t^i u_t^i;\ \ i \in N,\ t = 0, 1, \ldots$$

Пример топологии сети:



Оптимальное решение

IEEE TAC 2015, АиТ 2015 (Граничин, Амелина)

$$z_t = \sum_{i=1}^{n} \bar{u}_i, \ \bar{u}_i = ?$$

$$T(\mathbf{u}) = \max_{i \in N} \ time_i(\bar{u}_i) \to \min_{\mathbf{u}}$$

Согласованность:

$$p_1 \bar{u}_1 = p_2 \bar{u}_2 = \cdots = p_n \bar{u}_n$$

Функционал среднего риска (Потенциал Лапласа):

$$F(\mathbf{u}) = \sum_{i,j=1}^{m} a_t^{i,j} (time_t^i - time_t^j)^2 \ \to \ \min_{\mathbf{u}}$$

План

1. Введение (Introduction)
2. Рандомизированные алгоритмы в управлении и при извлечении знаний (Randomized algorithms in automatic control and data mining)
   - Оценивание при произвольных внешних помехах (Estimation under arbitrary external noise)
   - Оптимизация функционала среднего риска (Mean-Risk optimization)
   - Рандомизированные алгоритмы стохастической аппроксимации (Randomized stochastic approximation, SPSA)
   - Рандомизированные алгоритмы и квантовые вычисления (Randomized algorithms and quantum computing)
   - Приложения (Applications)
3. Мультиагентные технологии (Multiagent technologies)
   - Консенсусное управление (Consensus control)
   - Планирование группы БПЛА (UAV Group Planning)

Консенсусное управление

IEEE TIT 2015 (Амелина, Фрадков и др.)
Динамика агентов:

$$x_{t+1}^i = f(x_t^i, u_t^i),\ i \in N = \{1, \ldots, n\}$$

Согласованное поведение:

$$x_t^i \approx x_t^j,\ i,j \in N$$

Наблюдения:

$$y_t^{i,j} = x_{t-d_t^{i,j}}^j + w_t^{i,j},\ j \in N_t^i$$

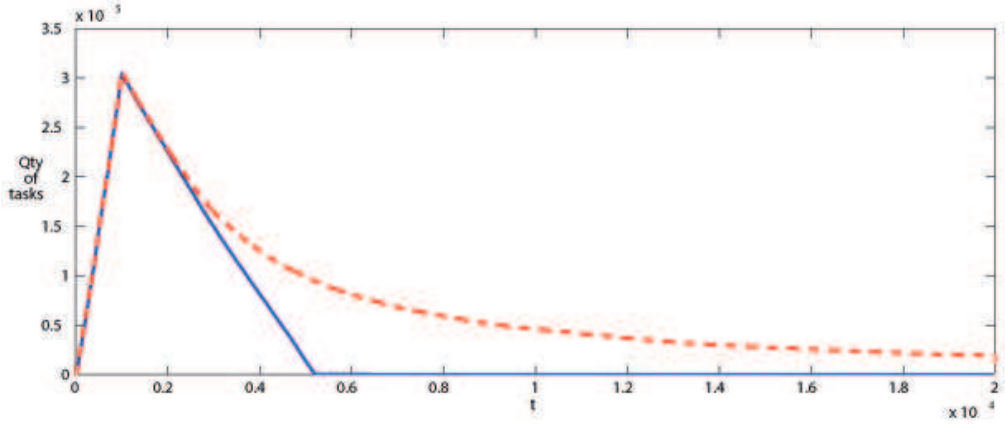
Протокол локального голосования:

$$u_t^i = \gamma \sum_{j \in \bar{N}_t^i} a_t^{i,j}(y_t^{i,j} - y_t^{i,i})$$

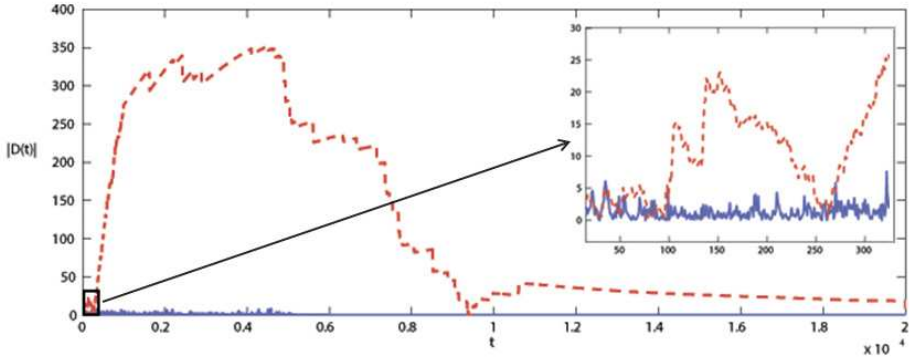
Моделирование

- $n = 1024$
- Число заданий 10^6 (поступают на интервале от 1 до 2000)

Количество заданий в очередях:



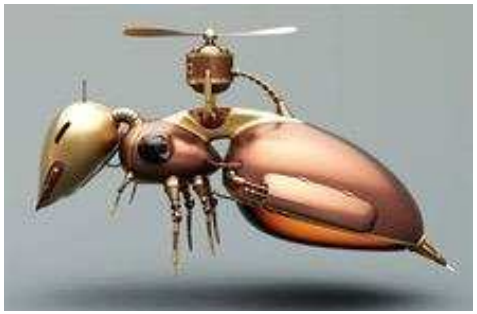
Среднее отклонение от консенсуса:



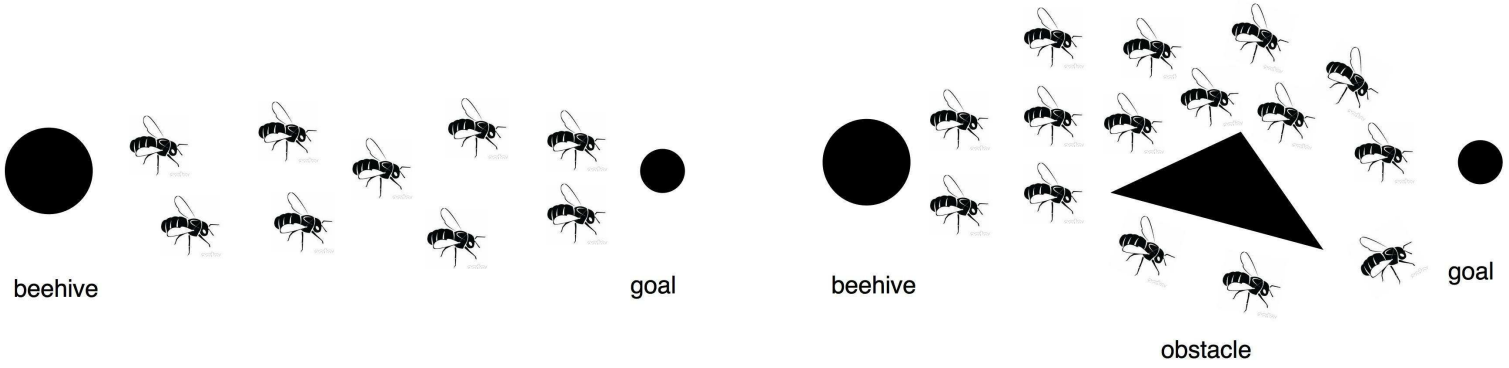
Интеллект роя

Агентно-ориентированное программирование:

- адаптивное управление сложными системами на основе мультиагентных технологий и искусственного интеллекта
- управление роем или стаей динамических объектов
- дешевые сенсоры
- масштабирование
- преодоление неопределенностей



Пример использования протокола локального голосования



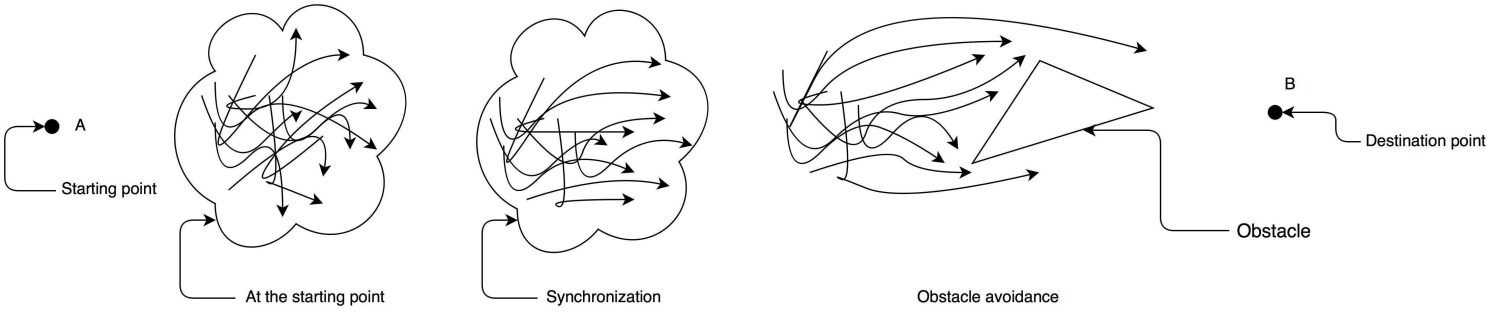


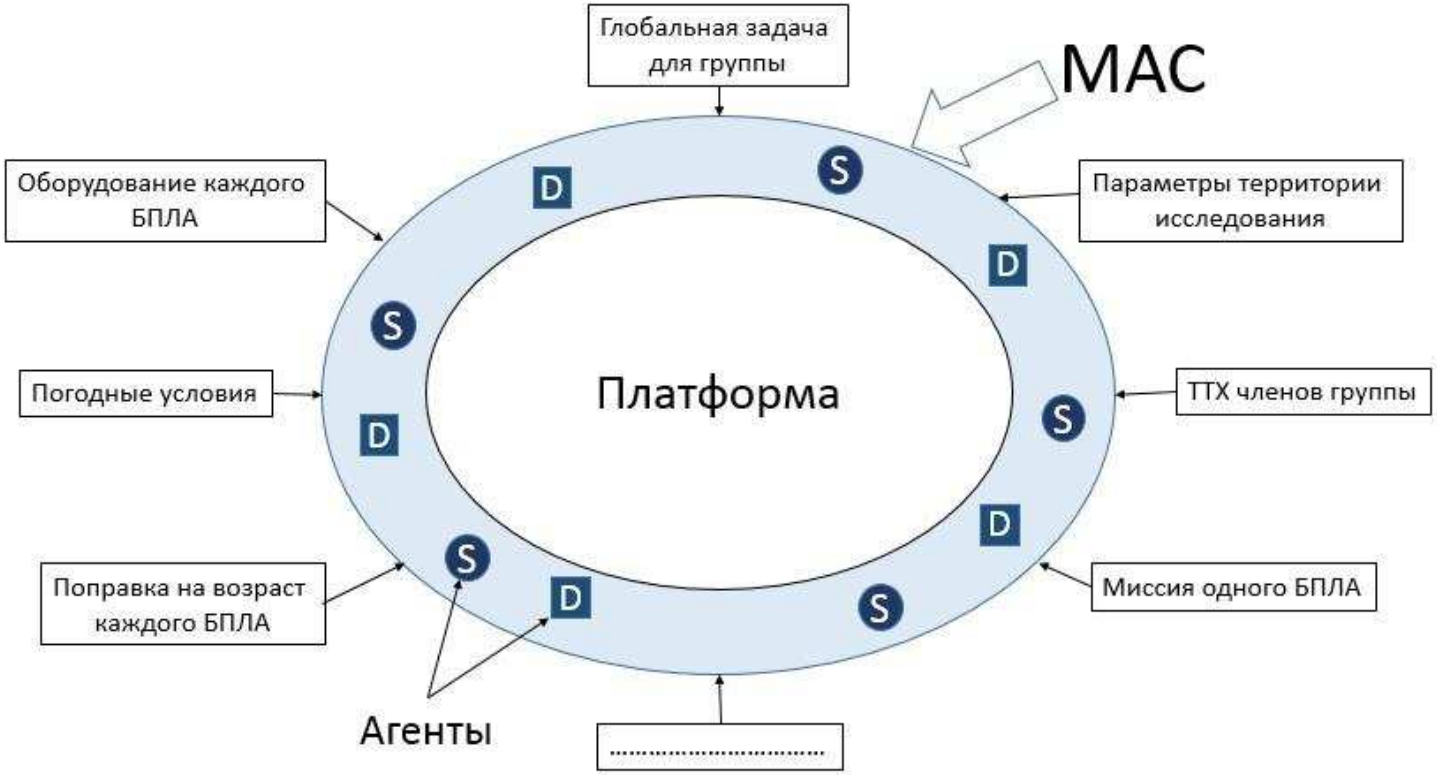
Рис.: Движение роя из точки A в точку B без препятствия и с огибанием претятствия.

Планирование группы БПЛА

Централизованный подход (классический)

При увеличении потока новой информации загрузка блока “планировщик” растет

Децентрализованный подход (МАС)



Новая информация попадает в систему, агенты ее “расхватывают” и используют. Существуют два типа агентов которые “ищут” друг друга

Согласованное решение

Две группы агентов:

- агент потребности — предоставляет в систему потребность на выполнение задания
- агент возможности — предоставляет в систему параметры и характеристики БПЛА в настоящий момент времени

Согласованное решение строится эволюционно и итеративно агентами потребностей и возможностей посредством переговоров и взаимных уступок от грубого и быстрого решения до сложного “равновесия”, когда все агенты довольны своим состоянием или не могут его улучшить

Преимущества и недостатки МАС

Преимущества над централизованной системой:

- можно добавлять новые входные параметры без перестроения системы планирования
- можно переносить в микрокомпьютеры каждого БПЛА для адаптивного перераспределения заданий непосредственно до вылета и в процессе полета
- система реагирует на все изменения параметров непосредственно в ходе планирования
- можно добавлять и убирать членов группы в процессе планирования
- исключается распределительный блок системы

Недостаток:

- сложно разрабатывать систему по сравнению с централизованным подходом

Решение:

- разработка универсальной платформы (платформа у нас есть)

Прототип системы планирования (Видео)

Задаваемые параметры

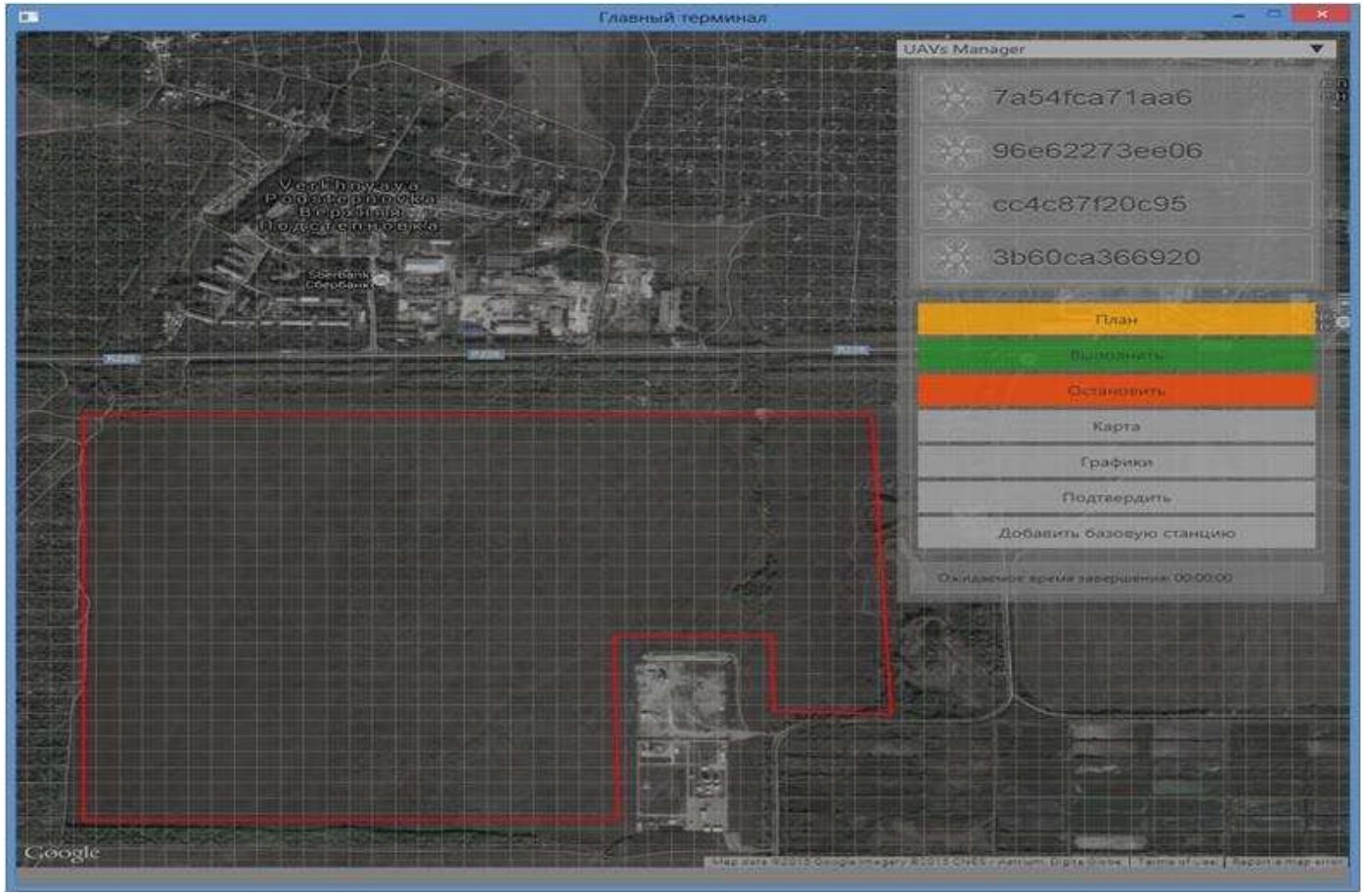
- размер территории выполнения задачи
- размер квадрата наблюдения БПЛА “за единицу времени” (например, квадрат 50x50 м)
- перекрытие фотоматериалов местности (при задачи картирования)
- скорость выполнения-наблюдения подзадачи-квадрата (определяется скоростью БПЛА, например, 50 м/с)
- время выполнения задачи (сейчас не задается)
- количество БПЛА в группе

Критерии:

- наблюдаемость квадратов —- менее наблюдаемые за предыдущие периоды планирования квадраты более “выгодные” при выборе;
- коммуникационная доступность других БПЛА — вынуждает БПЛА при планировании держать некоторое условно оптимальное расстояние до всех других БПЛА

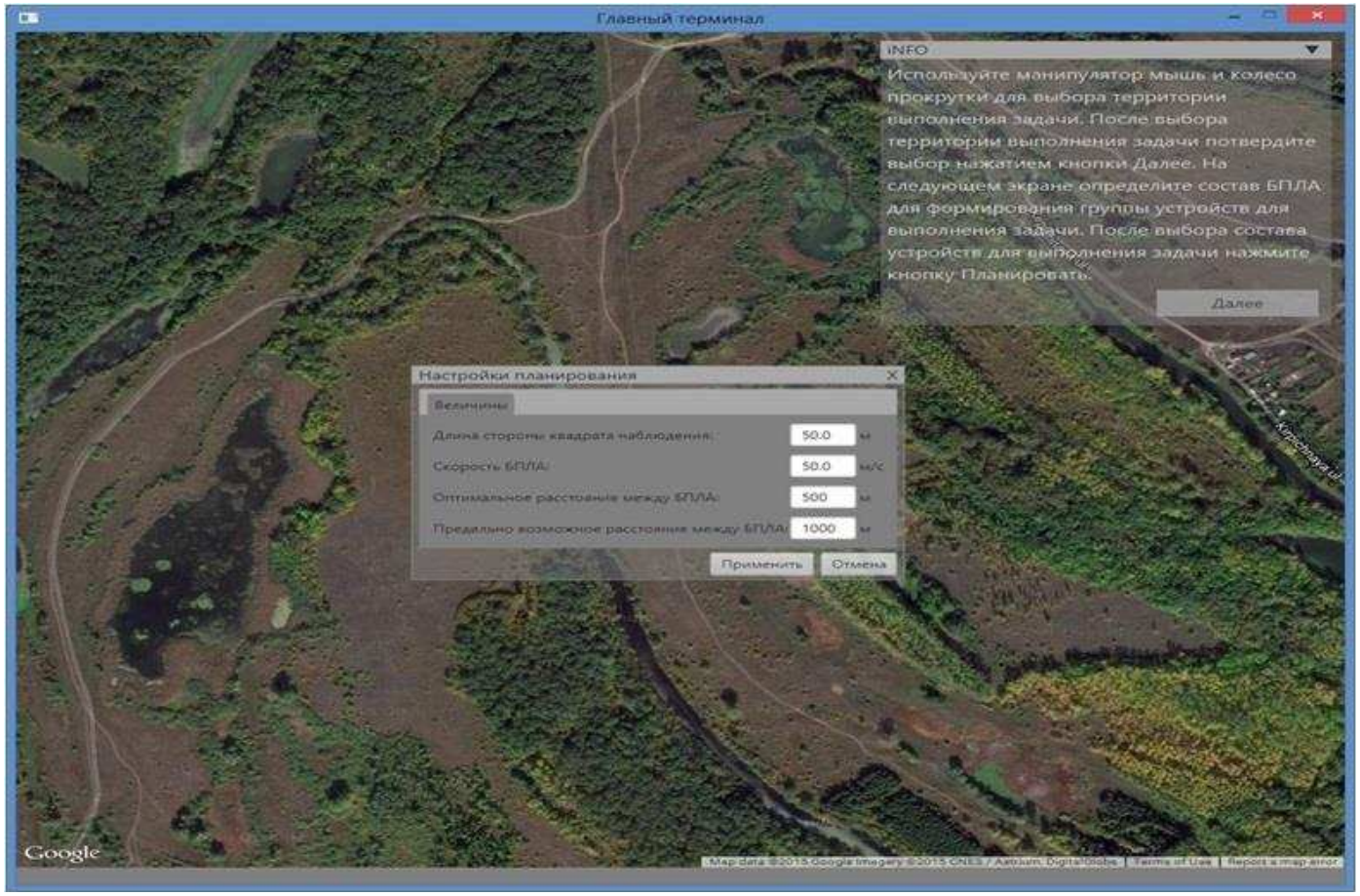
Пользовательский интерфейс

Выбор территории для исследования (форма выделяемого поля произвольная)



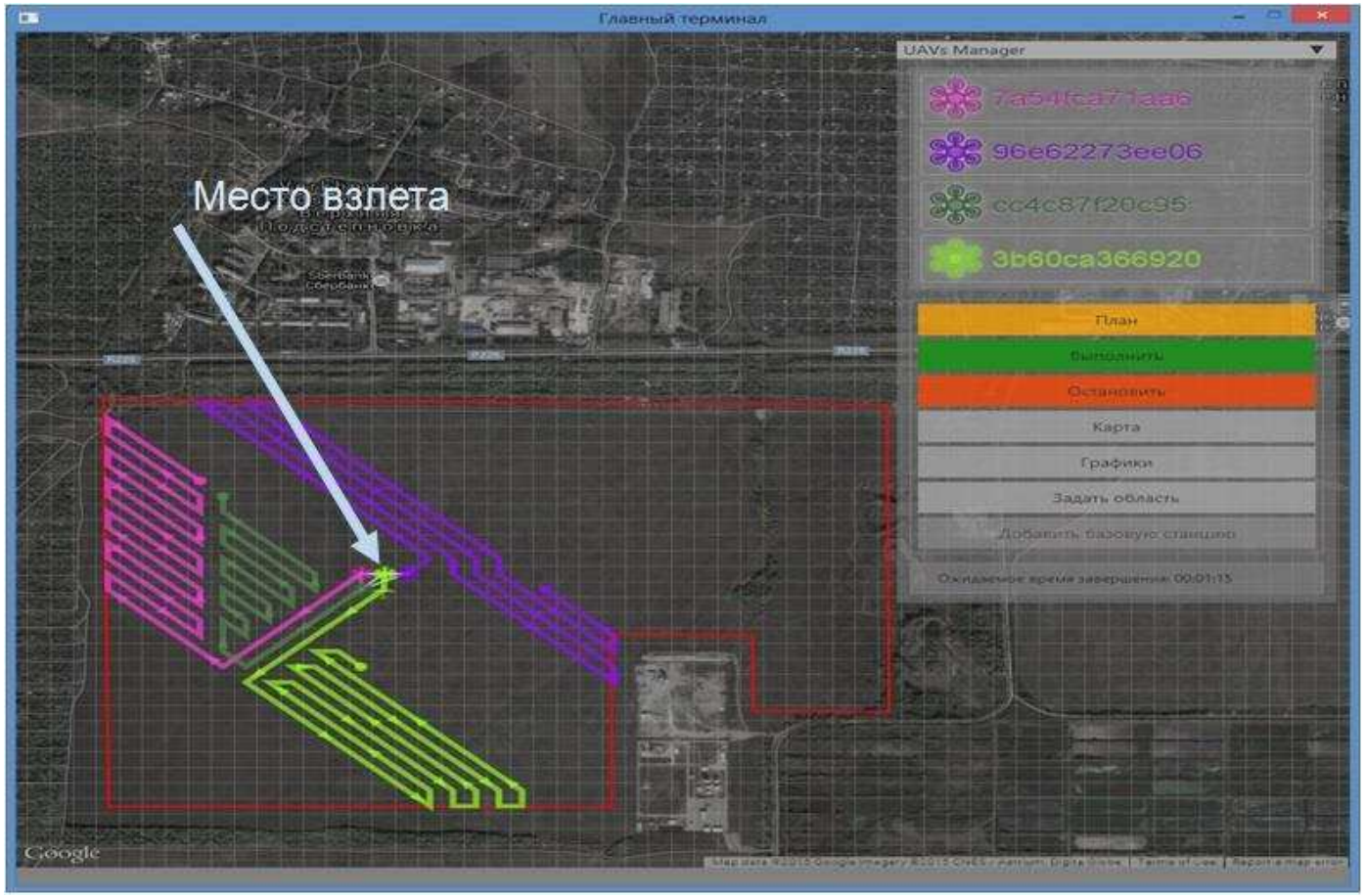
Пользовательский интерфейс

Настройка параметров планирования



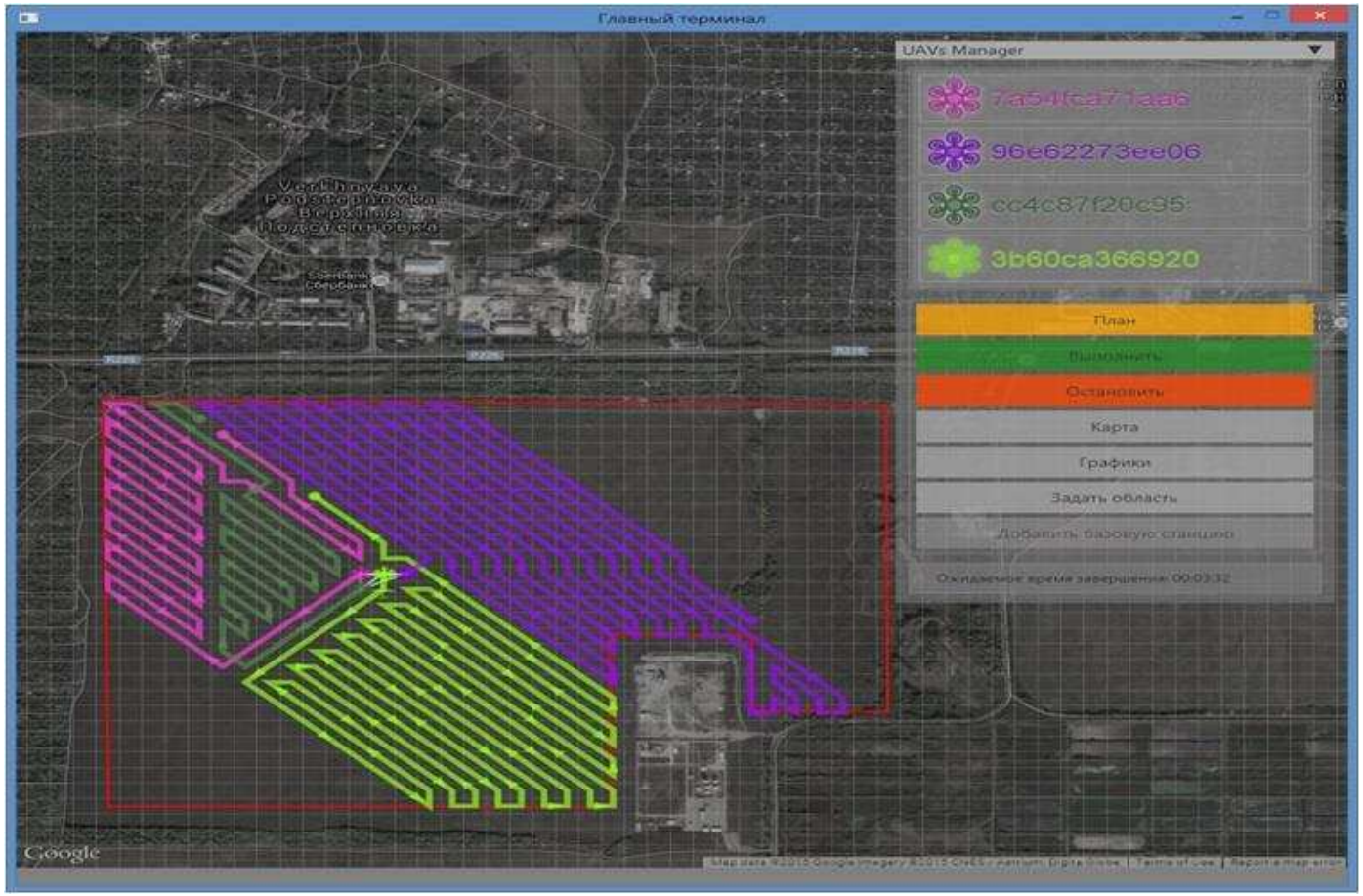
Пользовательский интерфейс

Задание количества БПЛА и формирование грубого планирования в зависимости от расположения точки взлета



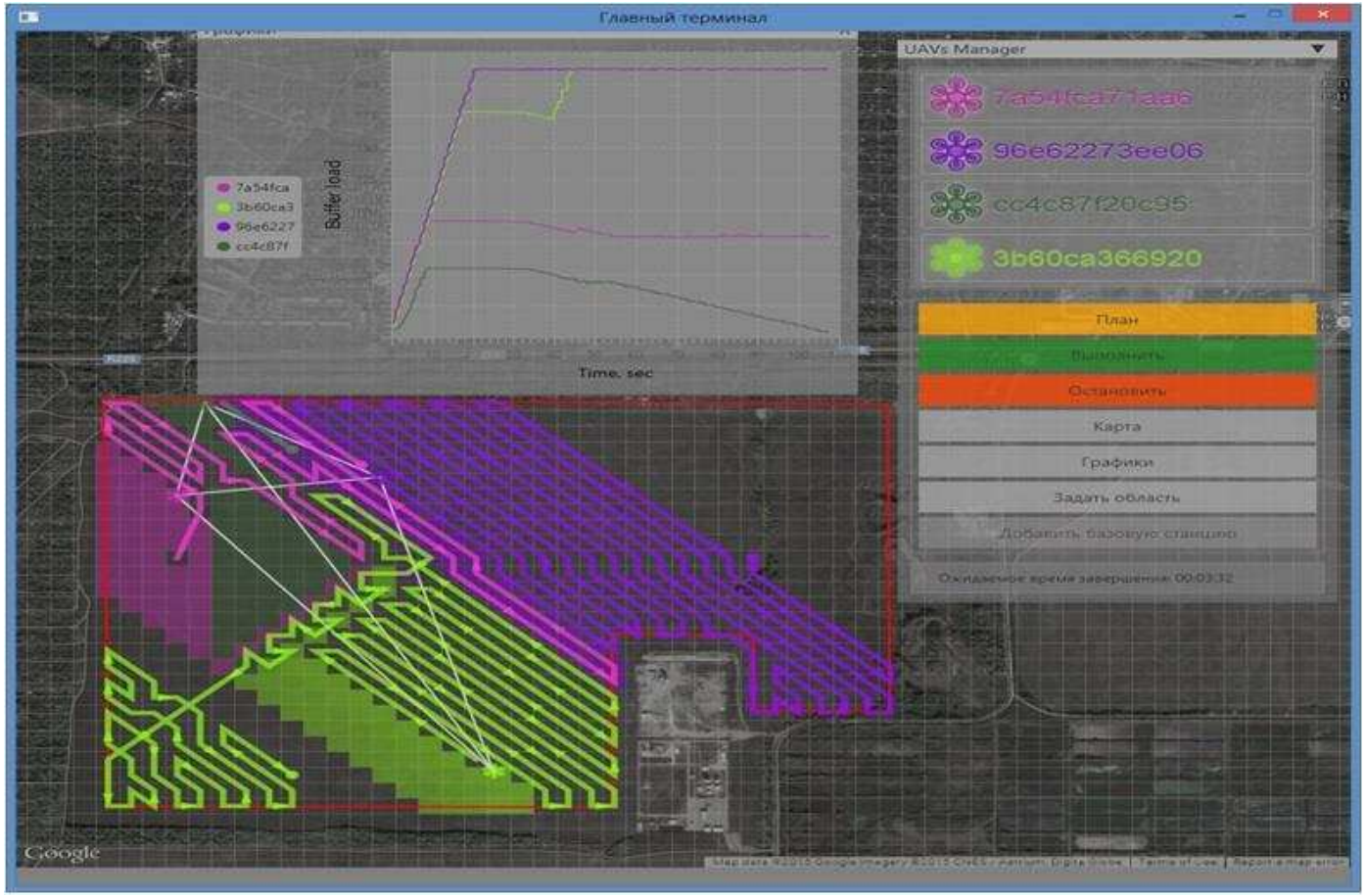
Пользовательский интерфейс

Начальное планирование для четырех БПЛА



Пользовательский интерфейс

МАС планировщик в работе. Агенты возможностей “расхватывают” задания



Пример: 3 БПЛА — 2 критерия

Видео

Пример: 3 БПЛА — 2 критерия

0-6 сек: Выбираем территорию для мониторинга группой БПЛА
6-9 сек: Выбираем сколько БПЛА будем использовать — два
9-17 сек: Задаем квадрат для исследования
17-30 сек: Выбираем функцию планирования — кнопка “План”
30-39 сек: Как только все квадраты распределены приступаем к выполнению. В процессе выполнения задания БПЛА обмениваются друг с другом информацией о выполненных заданиях
39-57 сек: Добавляем дополнительную территорию для исследования. Автоматически происходит перепланирование
57-1.16 сек: Добавляем еще один БПЛА и запускаем его в группу
1.16-1.28 сек: Включаем графическое отображение загрузки БПЛА
1.28-1.50 сек: Добавляем еще один квадрат исследования
1.50 сек: Останавливаем один БПЛА и выводим его из группы

Спасибо за внимание!

Стохастичность и вычисления

- «Кто нам мешает, тот нам поможет!»

- Стохастичность является основой самоорганизации