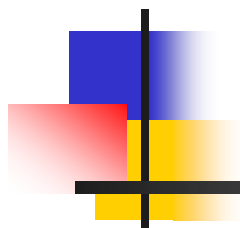


Понятие «Агента», информация, сигналы, данные, знания, управление



Олег Николаевич Граничин

Мультиагентные технологии, Лекция 2
Осень 2012

Санкт-Петербургский государственный
университет



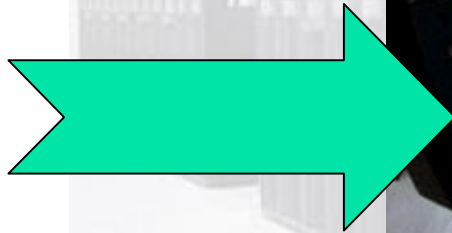
Задачи кибернетики на ближайшие 50 лет

Доклад Мюррея (Сидней, CDC-2000)

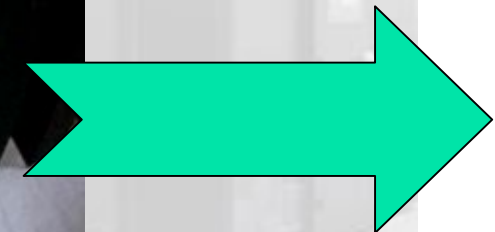
- динамически реконфигурируемое интеллектуальное управление,
- асинхронная теория управления,
- управление через Интернет,
- перепрограммирование системы управления бактерией,
- создание футбольной команды роботов, которая выиграет у победителя кубка мира среди людей

Проблема современных суперкомпьютеров — организация эффективного взаимодействия с ними

Входные
данные

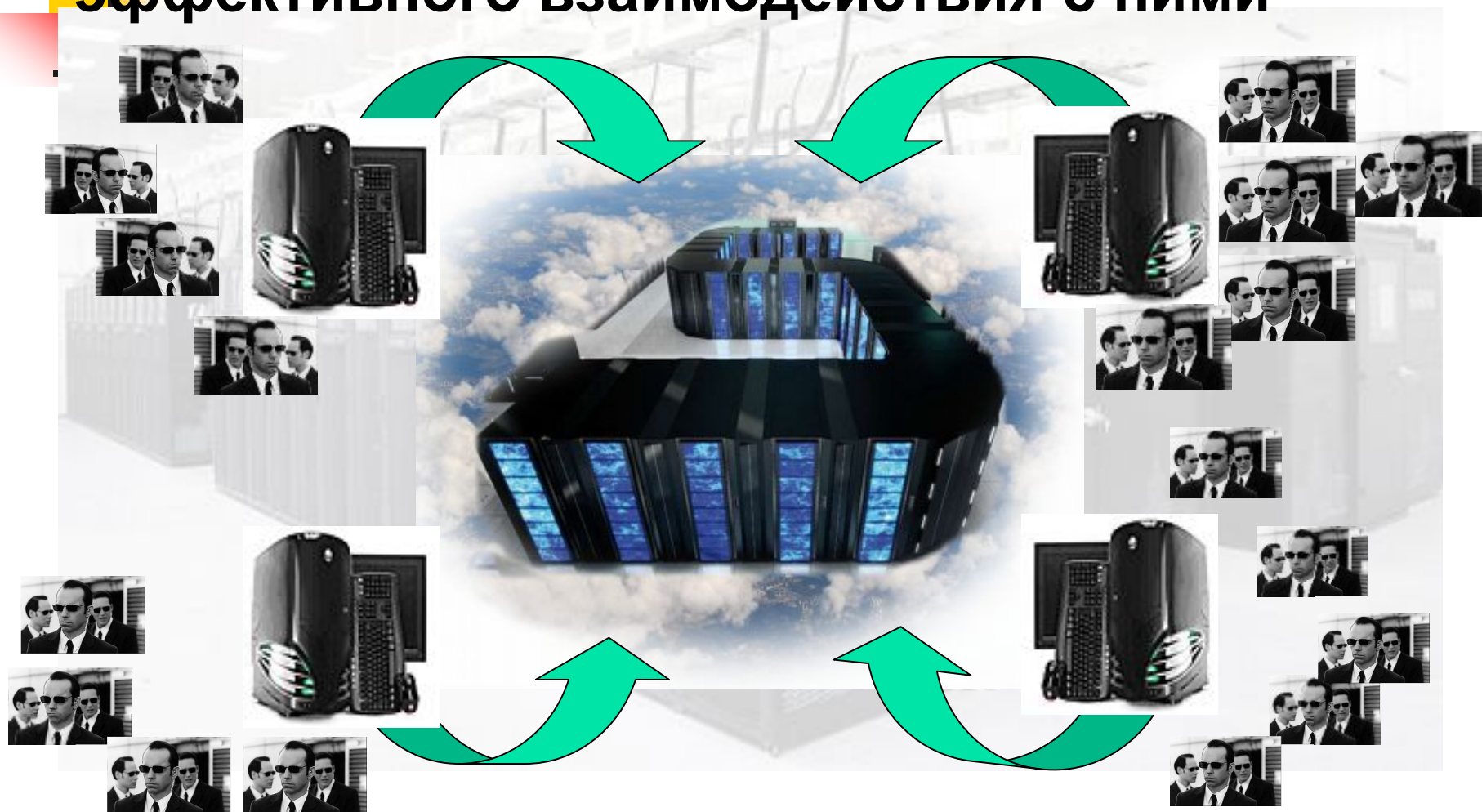


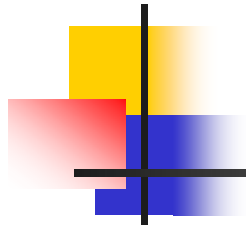
Выходные
данные



Время сбора и подготовки данных становится критичным

Проблема современных суперкомпьютеров — организация эффективного взаимодействия с ними



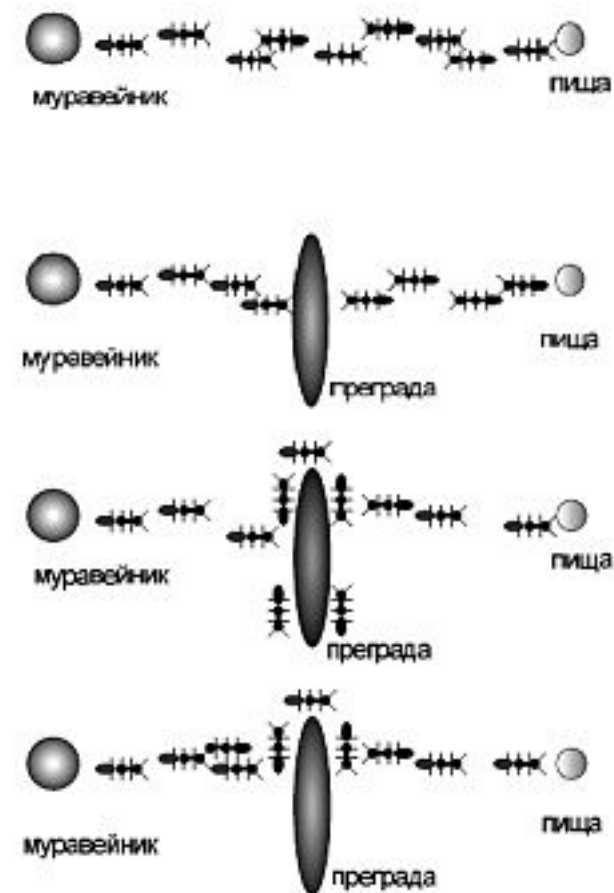


Интеллект

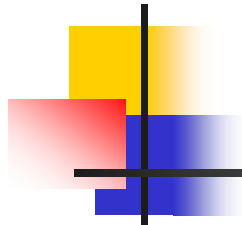
Интеллект – это то, что

- 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.

А есть ли интеллект у колонии муравьев?



Всероссийская суперкомпьютерная конференция «Научный сервис в сети Интернет
Абрау-Дюрсо, 21-25 сентября 2010 г.



Эмерджентный интеллект

Активные агенты: миллион муравьев побеждают слона!

Эмерджентный интеллект (интеллектуальный резонанс, интеллект роя)— это

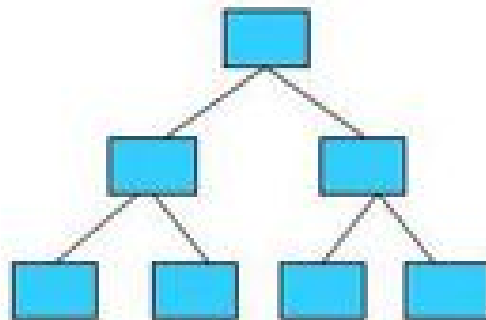
- проявление неожиданных свойств, которыми обладает система, но не обладает ни один из входящих в нее отдельных элементов

Ключевая особенность –

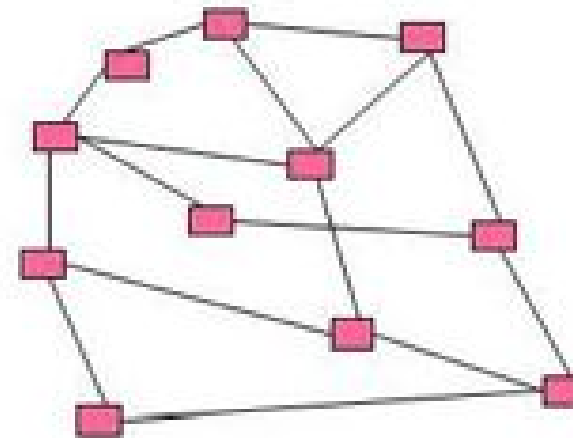
- динамика и непредсказуемость процесса принятия решений

Две схемы построения ПО

Традиционная
(иерархическая)



Мультиагентная
система



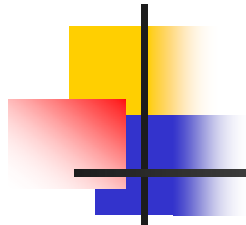
- В мультиагентной системе каждой сущности ставится в соответствие программный агент, который представляет ее интересы



Мультиагентные системы (МАС)

В качестве основы были взяты достижения в следующих областях:

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



Агенты

Эволюция понятия «агента» в
контексте разработки ПО

- интеллектуализации
пользовательского интерфейса
- автономные агенты (персональные
ассистенты)

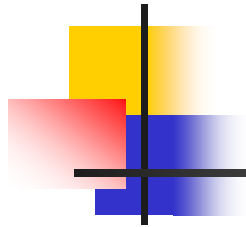


Свойства агентов

Агент – некто или нечто,
прикладывающее усилия для
достижения эффекта

- совершает действия
- действует автономно и рационально

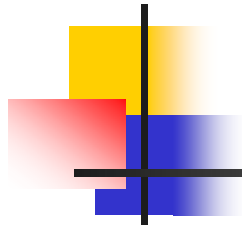
Агент помещен во внешнюю среду, с
которой он способен
взаимодействовать



Рациональность агента

Агент имеет

- цели (желания, desires)
- представления о внешнем мире (убеждения, beliefs),
на которые агент опирается при выборе
- действия (реализации намерений, intentions – множество избранных, совместимых и достижимых желаний)



Интеллектуальный агент

Агент, обладающий следующими свойствами:

- реактивность (reactivity) – ощущает внешнюю среду и реагирует на изменения в ней, совершая действия, направленные на достижение целей
- проактивность (pro-activeness) – показывает управляемое целями поведение, проявляя инициативу, совершая действия направленные на достижение целей
- социальность (social ability) – взаимодействует с другими сущностями внешней среды (другими агентами, людьми и т.д.) для достижения целей

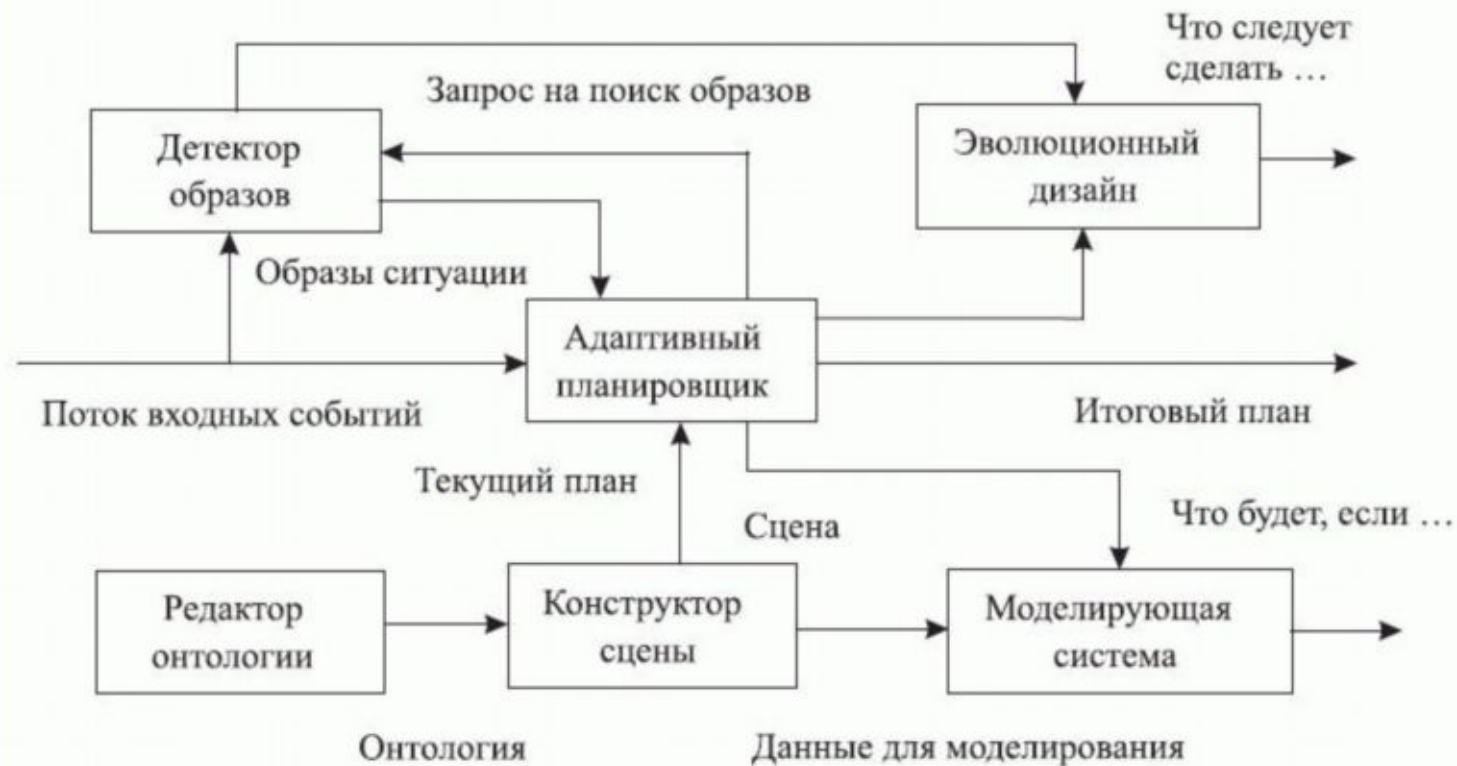


Адаптивность агента

Сложность формулирования содержательных практически значимых задач и невозможность априорного точного задания всех условий функционирования выдвигают адаптивные постановки проблем, выделяя такую особенность агентов, как

- адаптивность – способность автоматически приспособливаться к неопределенным и изменяющимся условиям в динамической среде

Схема взаимодействия модулей МАС фирмы "Генезис знаний"



<http://www.kg.ru/>



Задачи модулей МАС фирмы "Генезис знаний"

- **Детектор образов** распознает типовые ситуации, возникающие в ходе поступления заявок и вырабатывает прогноз заявок и рекомендации по планированию с учетом предистории
- **Адаптивный планировщик** обрабатывает поток входящих событий (поступлений заявок, ввода новых ресурсов, выхода из строя ресурсов и т.п.)
- **Конструктор сцены** позволяет редактировать начальную конфигурацию сети и определить все параметры ресурсов компании
- Конструктор сцены основывается на общей базе знаний **онтологии**, описывающей деятельность компании, в которой присутствуют базовые понятия и отношения между ними, и которая при развитии бизнеса может расширяться с использованием редактора онтологии
- **Редактор онтологии** позволяет ввести и изменить общую онтологию компании, описывающую модель знаний предметной области, которая затем применяется в редакторе сети для описания конфигурации бизнеса. Онтология содержит базовые знания и отношения между ними, представляемые в форме семантической сети.
- **Моделирующая система** – программный модуль, позволяющий осуществлять моделирование ситуации по принципу «Что если?»
- **Эволюционный дизайн** – модуль, вырабатывающий предложения по улучшению конфигурации сети в части увеличения или уменьшения определенного числа ресурсов, изменению географии ресурсов и т.д.

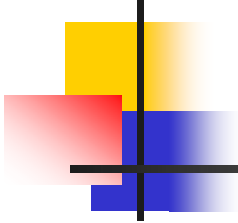


Информация, сигналы, данные, знания и управления

- Что такое информация?



Информация

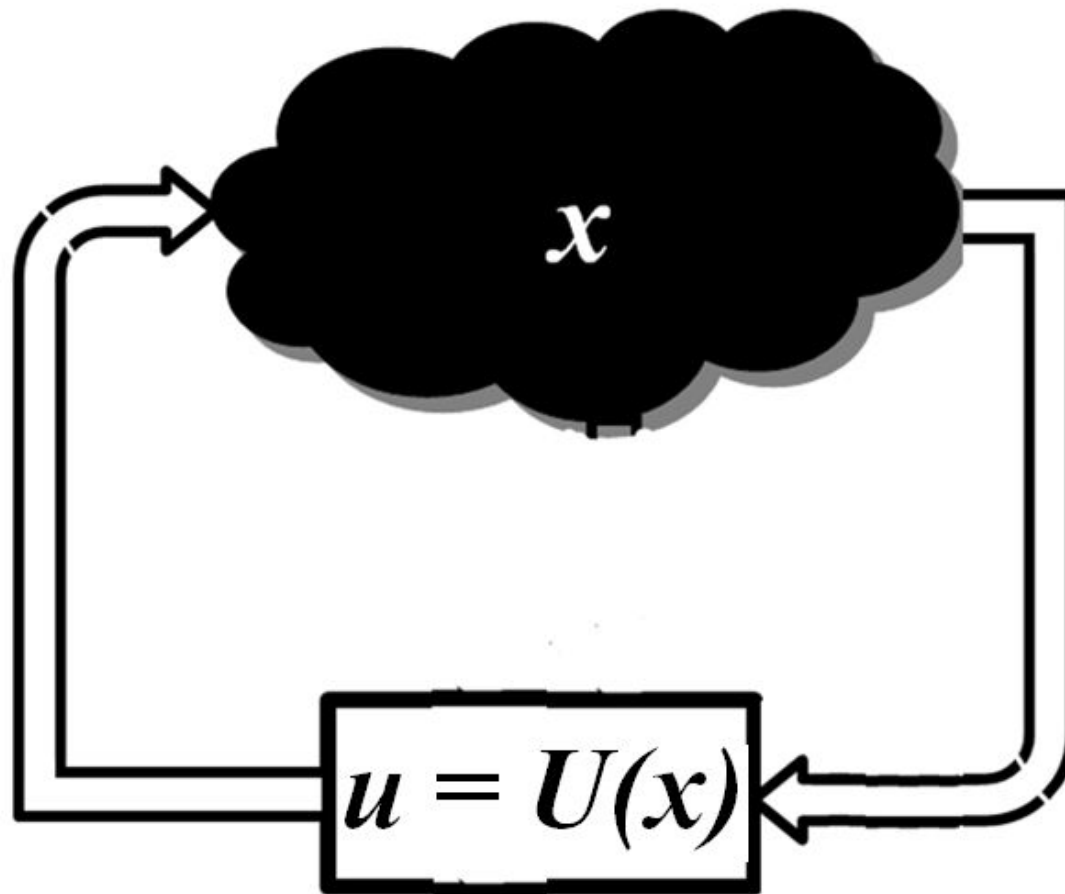


Обратная связь по состоянию

- x – информация (состояние системы)
- u – управление

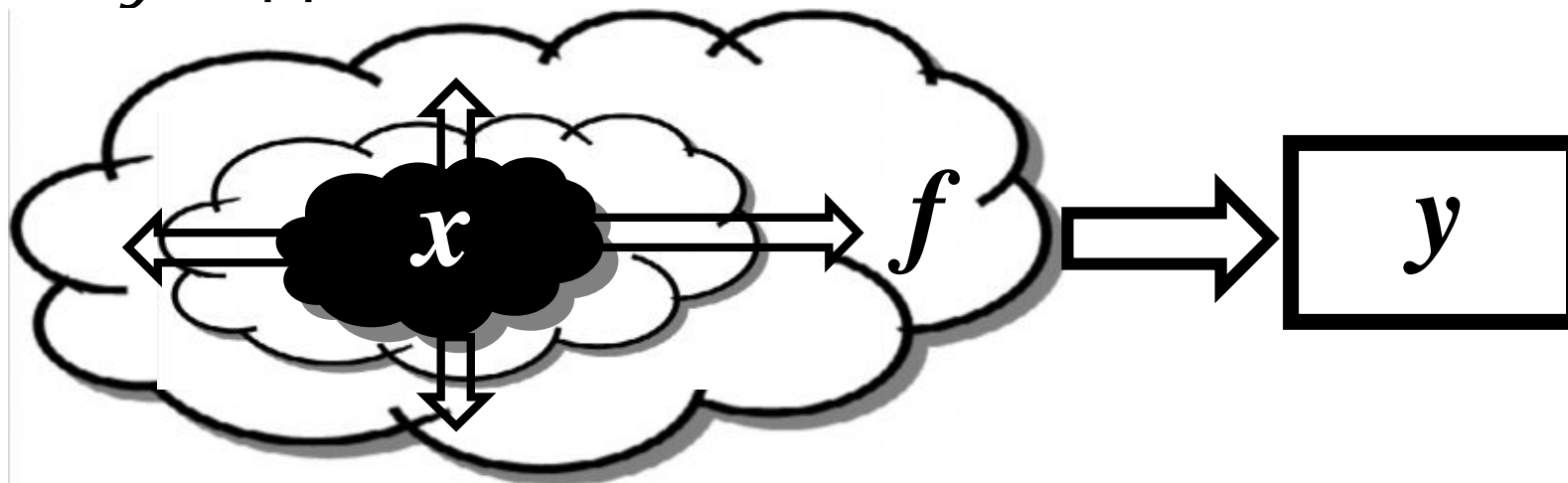
$$u = U(x)$$

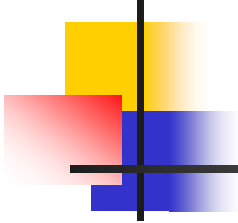
Взаимное влияние информации и управления



Сигналы и данные

- f – сигналы
- y – данные

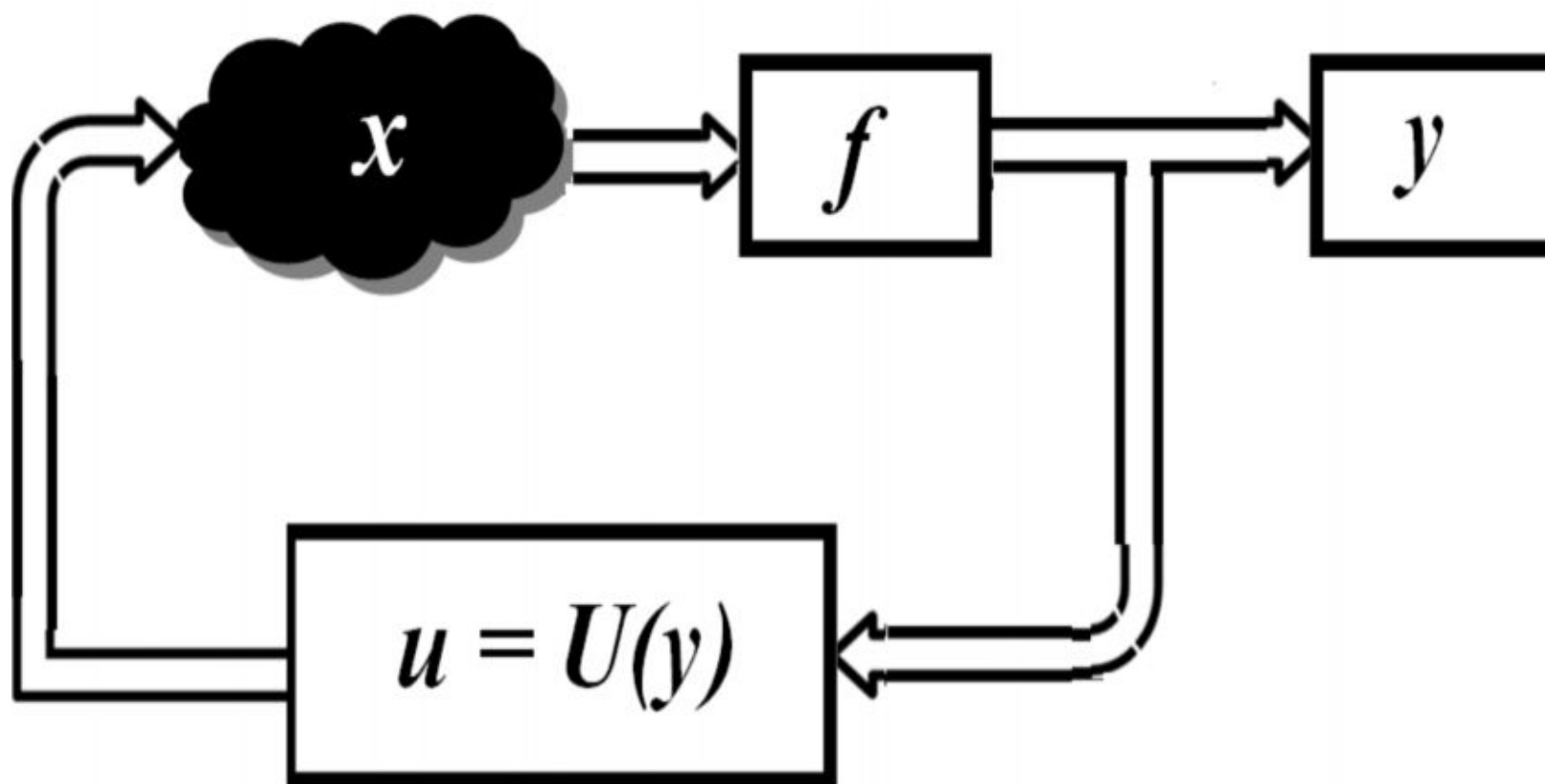
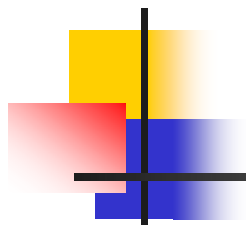


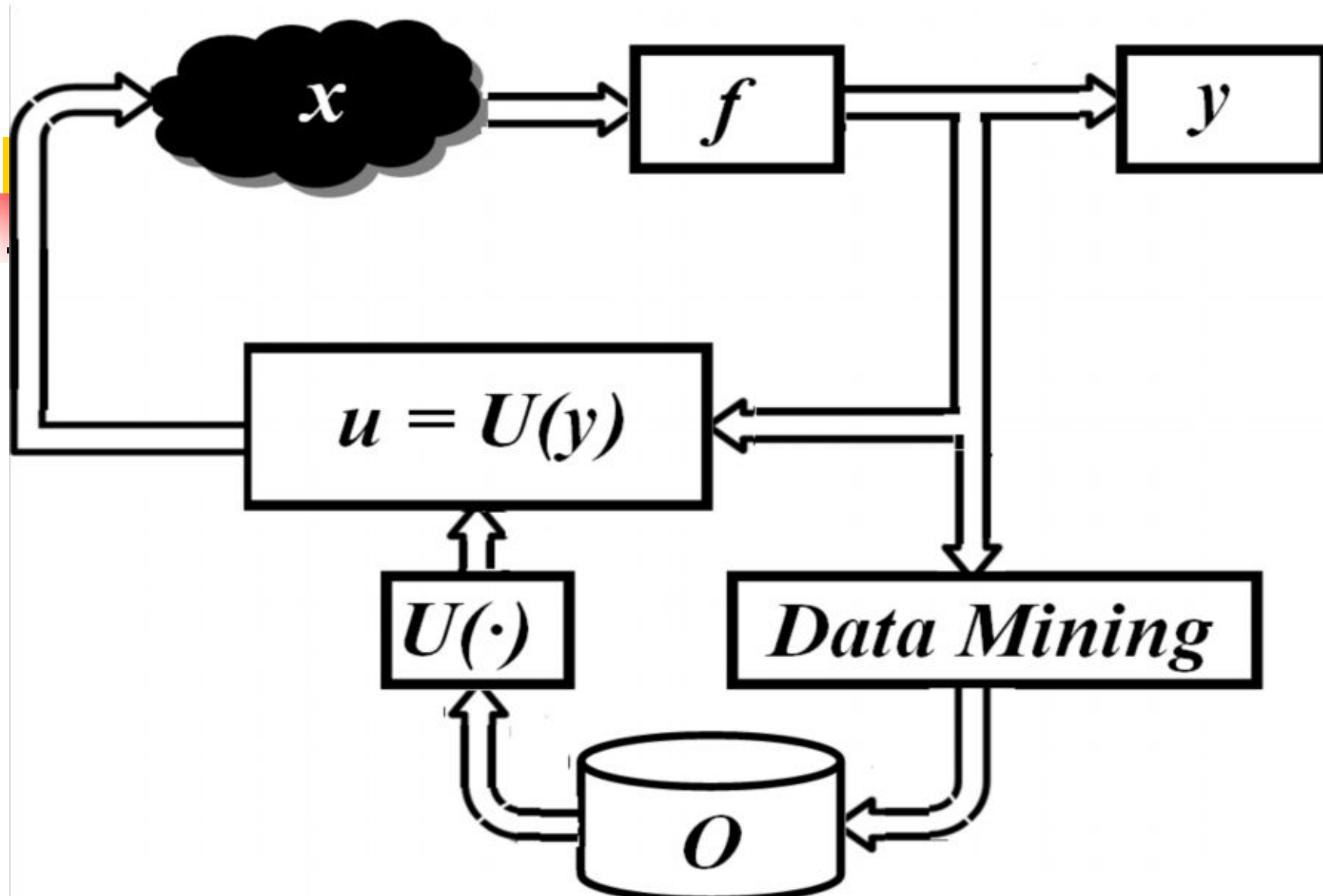


Обратная связь по наблюдениям

- y – наблюдения (данные)
- u – управление

$$u = U(y)$$

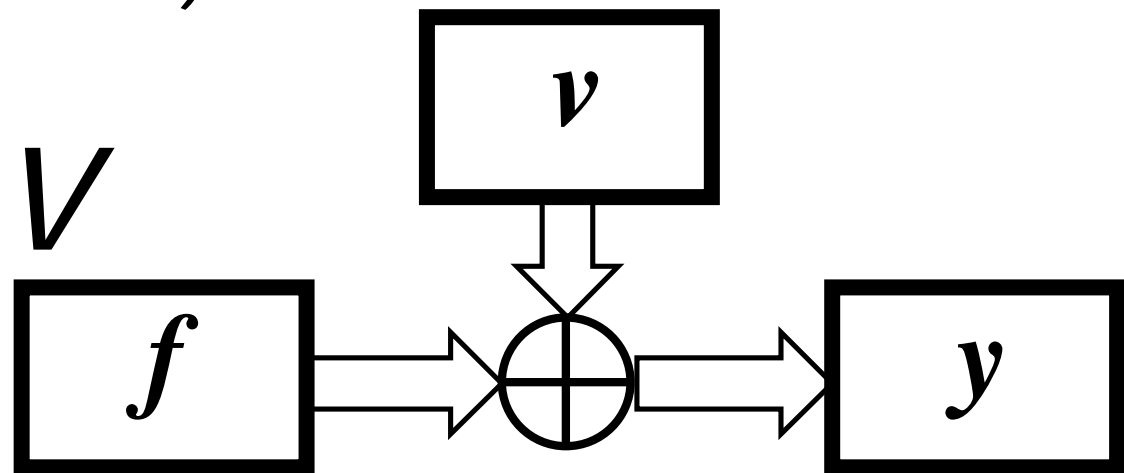




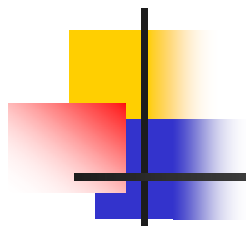
Наблюдения с помехами

- f – сигнал
- y – наблюдения (данные)
- v – ошибки (помехи)

$$y = f + v$$

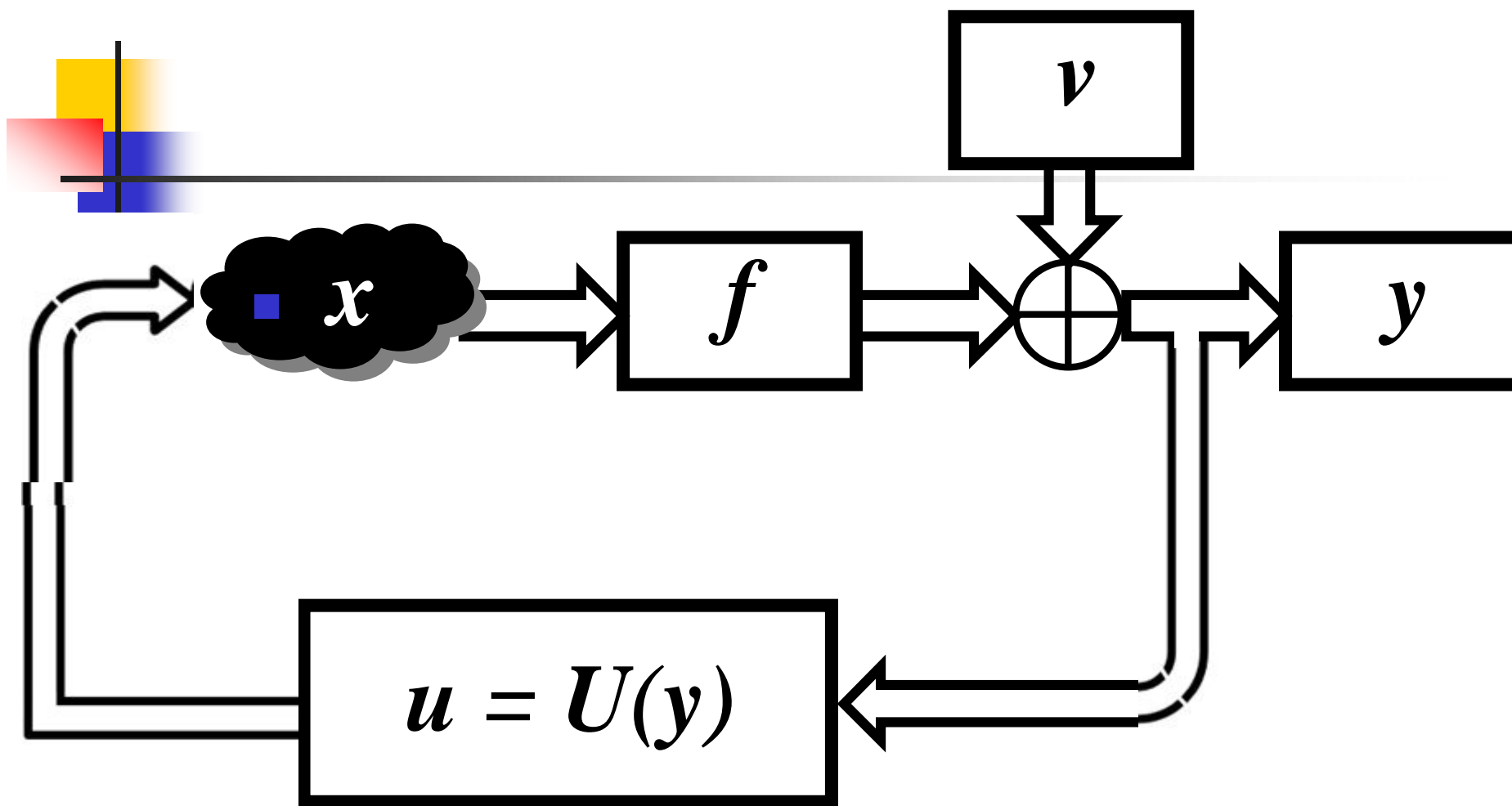


Типы помех и усреднение наблюдений

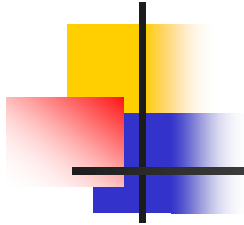


$$y = f + v$$

- помех нет
- помехи малы
- помехи малы с течением времени
- помехи гауссовские
- помехи произвольные



Характерные черты новых вычислительных устройств



Гибридность

Асинхронность

Кластерность

Мультиагентность

Стохастичность

.....



Природа стохастичности

Неопределенности

- Моменты переключений
- Действия при переключениях
- (Задание Программы, Целей?..)



Стохастичность (στοχάζομαι)

- 1) целиться, метить
ex. σ. τινος Xen., Isocr., Plat., Arst. — целиться во что (в кого)-л.
- 2) иметь в виду, стремиться
ex. (τινος Plat., Arst. и πρὸς τι Plat.)
σ. κριτῶν τῶν κρατίστων Xen. — стремиться к тому, чтобы судьями были самые влиятельные люди
- 3) применяться, приспособляться
ex. (σ. τοῦ συμβουλευομένου Plat.)
σ. τῆς τοῦ δήμου βουλήσεως Polyb. — применяться к воле народа
- 4) умозаключать, судить, догадываться, разгадывать
ex. (τινος Isocr., Plat. и τι Xen.)
σ. τὰ συμφέροντα Xen. — догадываться о том, что требуется;
σ. ἔκ и διὰ τινος Polyb. — заключать на основании чего-л.;
τῷ στοχάζεσθαι Plat. — путем догадок

Стохастичность и вычисления

- «Кто нам мешает, тот нам поможет!»

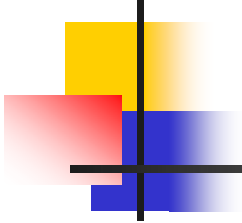


- Стохастичность является основой самоорганизации



Детерминированный алгоритм

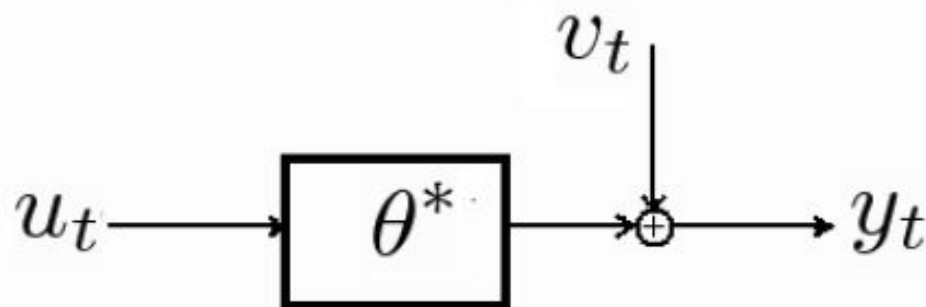
Каждый шаг задается
детерминированным
правилом с
использованием
результатов предыдущих
шагов, и полученная новая
информация о системе
(выход) возвращается для
использования в
последующих шагах
алгоритма



Детерминированные подходы часто «проваливаются»

- В теории и практике возникает много трудностей, когда мы пытаемся исследовать “сложные” системы.
Во многих практических приложениях традиционные детерминированные алгоритмы не дают результатов, когда система сложная. Часто это ведет к так называемым *NP*-сложным задачам

Пример: оценивание неизвестного параметра



$$y_t = \theta^* \cdot u_t + v_t,$$

- можно выбирать входы (управления) u_t
- измерять выходы y_t , $t = 1, 2, \dots, N$

При произвольных внешних помехах

НЕТ РЕШЕНИЯ в классе

детерминированных алгоритмов!

$$\theta^* = 3$$

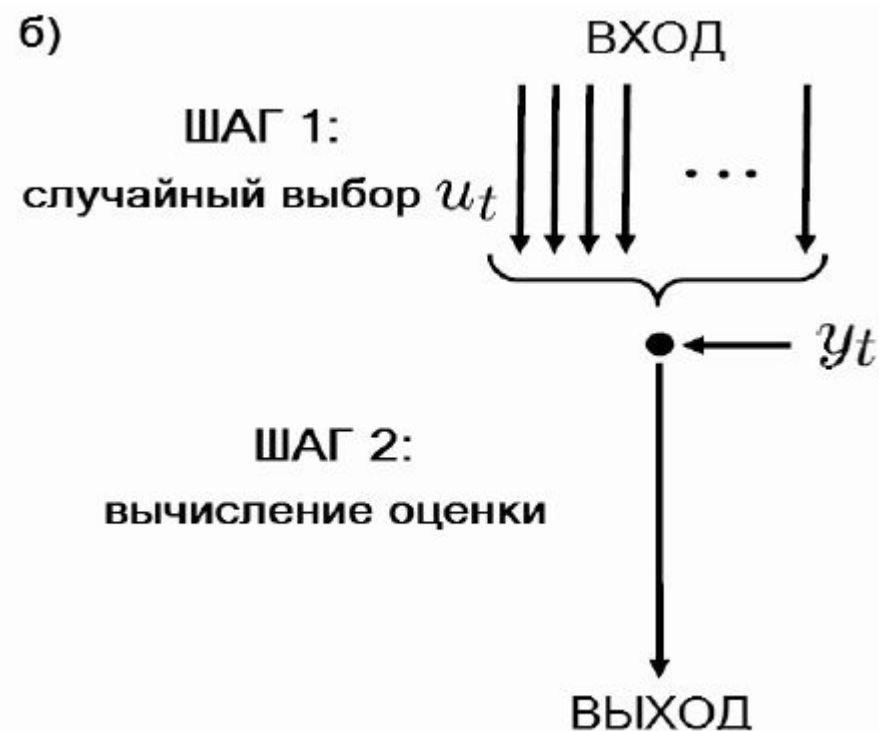
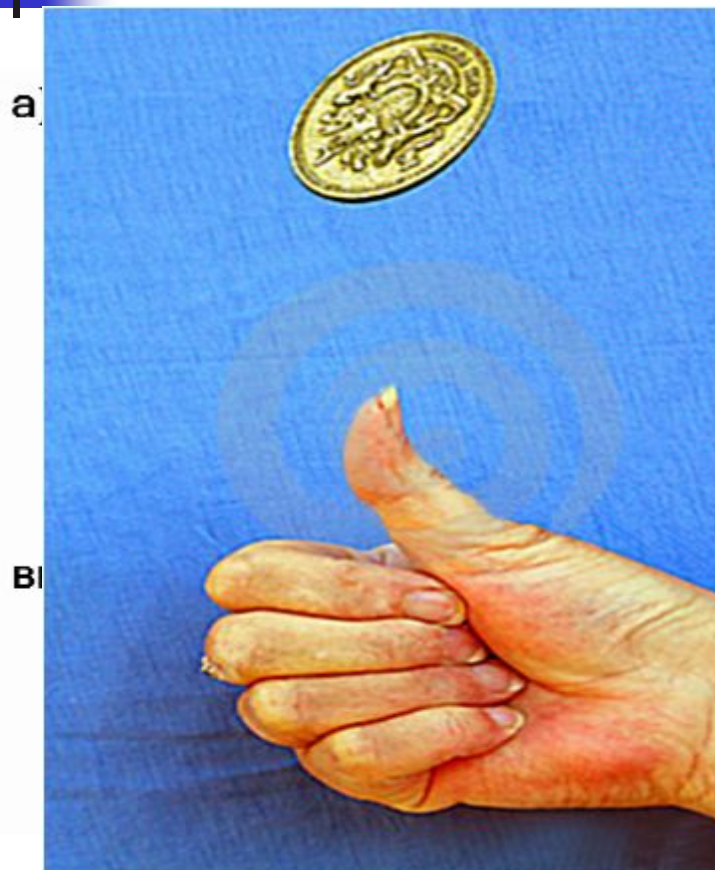
t	1	2	3	4	5	6	7
u_t	1	1	1	1	1	1	1
$v_t = rand() - 0.5$							
y_t	2.9	2.8	3.2	3.3	2.6	3.4	2.7
$\hat{\theta}_t$	2.9	2.85	2.97	3.05	2.96	3.03	2.99
$v_t = rand() - 0.5 + m, m = 1$							
y_t	3.9	3.8	4.2	4.3	3.6	3.9	4.2
$\hat{\theta}_t$	3.9	3.85	3.97	4.05	3.96	4.03	3.99



Рандомизация

- Рандомизация может оказаться мощным средством для решения ряда проблем, которые кажутся нерешаемыми детерминированными методами
- В рандомизированном алгоритме выполнение одного или несколько шагов основано на случайном правиле (т. е. среди многих детерминированных правил одно выбирается случайно в соответствии с вероятностью P)

Два типа алгоритмов





«Обогащенные» наблюдения

На первом шаге

$$u_t = \begin{cases} +1, \text{ с вероятностью } \frac{1}{2}, \\ -1, \text{ с вероятностью } \frac{1}{2}, \end{cases}$$

На втором шаге формируем величины

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

Для “новой” последовательности наблюдений

$$\bar{y}_t = \theta^* \cdot \bar{u}_t + \bar{v}_t, \quad \bar{u}_t = u_t^2 \text{ и } \bar{v}_t = u_t \cdot v_t.$$

Предварительный результат

$$\theta^* = 3$$

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
\bar{u}_t	1	1	1	1	1	1	1
\bar{y}_t	2.1	3.8	1.8	4.3	3.6	4.4	2.3
$\hat{\theta}_t$	2.1	2.95	2.57	3.00	3.12	3.33	3.19



Гарантированное множество

1. Пусть $M = 8$, выберем случайно семь ($= M - 1$) разных групп по четыре индекса T_1, \dots, T_7 .
2. Вычислим семь частичных сумм $\bar{s}_i = \sum_{j \in T_i} \bar{y}_j$, $i = 1, \dots, 7$.
3. Сформируем доверительный интервал

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

содержащий θ^* с вероятностью $p = 75\%$ ($= 1 - 2 \cdot 1/M$).



Результат

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

- интервал $\hat{\Theta} = [2.875; 3.4]$ содержит θ^* с вероятностью $p = 75\%$,



Вероятностно-успешный алгоритм

- Рандомизированный алгоритм называется вероятностно-успешным с вероятностью p , если вероятность его правильного результата не менее p
- Кроме обогащения данных наблюдений рандомизированный сценарный подход позволяет решить задачи «эффективно с высокой вероятностью» для почти всех ограничений, которых часто может быть очень много



Рандомизация...

1930...

- Fisher (remove bias)

1950...

- Metropolis, Ulam (method Monte-Carlo)
- Растрингин, Ермаков и др. (случайный поиск)
- Kirkpatrick, Holland и др. (метод отжига, генетические алгоритмы)

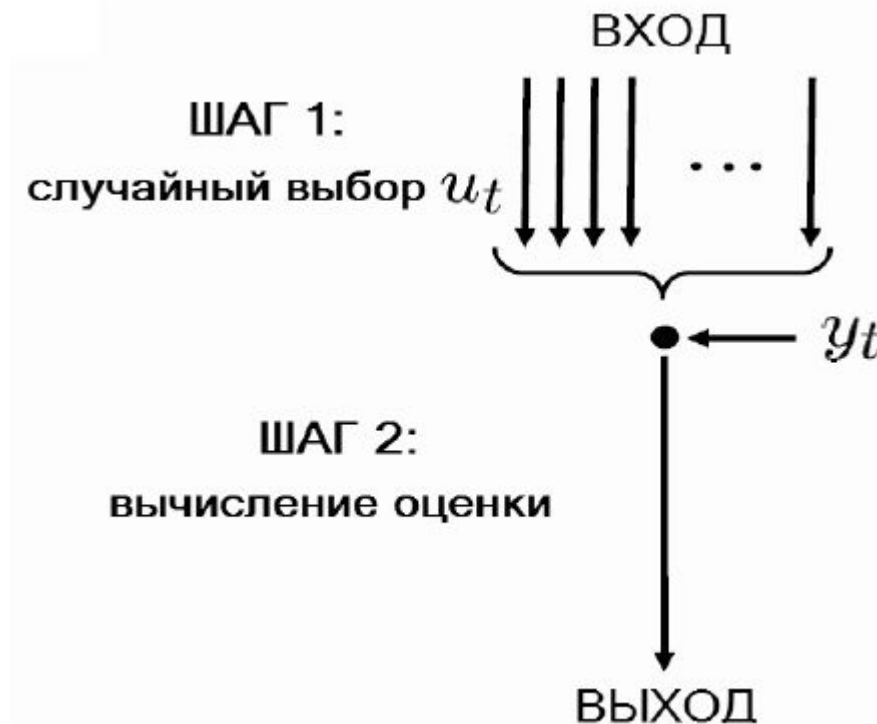
1980-90

- Поляк, Цыбаков, Luing, Guffi, Spall (fast algorithms)
- Granichin (arbitrary noise)
- Vadiyasagar (Randomized Learning Theory)

2000....

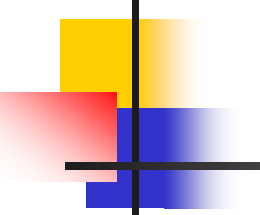
- Tempo, Campi, Calafiore, Sherbakov *etc.*
- ...

Рандомизированный алгоритм



Мультиагентность

Многомерная оптимизация



x_1, x_2, \dots — точки наблюдений $\in \mathbb{R}^d$

y_1, y_2, \dots — наблюдения $\in \mathbb{R}^1$

$$y_n = F(x_n, w_n) + v_n$$

w_1, w_2, \dots — неконтролируемые случайные
возмущения с неизвестным распределением $P(\cdot)$

v_1, v_2, \dots — неизв., но ограниченные (неслучайн.)

Цель

$$f(x) = \int F(x, w) P(dw) \rightarrow \min$$

Алгоритм SPSA

$\Delta_1, \Delta_2, \dots$ — probing sampling
with Bernoulli distribution $\Delta_n = \begin{pmatrix} \pm 1 \\ \pm 1 \\ \vdots \\ \pm 1 \end{pmatrix} \in \mathbb{R}^d$
(simultaneous perturbation)

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

$$x_n^\pm = \hat{\theta}_{n-1} \pm \beta_n^\pm \Delta_n$$

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

Алгоритм с одним измерением

$$x_n = \hat{\theta}_{n-1} - \beta_n \Delta_n$$

$$\hat{\theta}_n = \hat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n} \Delta_n y_n$$

Алгоритм с одним измерением

$$x_n = \hat{\theta}_{n-1} - \beta_n \Delta_n$$

$$\hat{\theta}_n = \hat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n} \Delta_n y_n$$

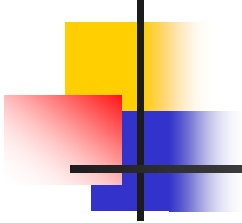
- Сокращение числа наблюдений
до 1 или 2 вместо $2d$



Преимущества SP5A

- Асимптотически оптимальная скорость сходимости
- Min число измерений на итерации
- Состоятельность при почти произвольных помехах
- Работоспособность в нестационарных задачах
- Легко может быть реализован мультиагентной системой или на квантовом компьютере

Обоснование SPSSA


$$E\left(\theta_n - \frac{\alpha_n}{\beta_n} \Delta_n f(\theta_n + \beta_n \Delta_n)\right) \approx$$

$$\theta_n - \frac{\alpha_n}{\beta_n} E\left(\Delta_n f(\theta_n) + \frac{\beta_n \Delta_n \Delta_n \nabla f(\theta_n)}{2}\right) =$$

$$\theta_n - \frac{\alpha_n}{2} \nabla f(\theta_n).$$



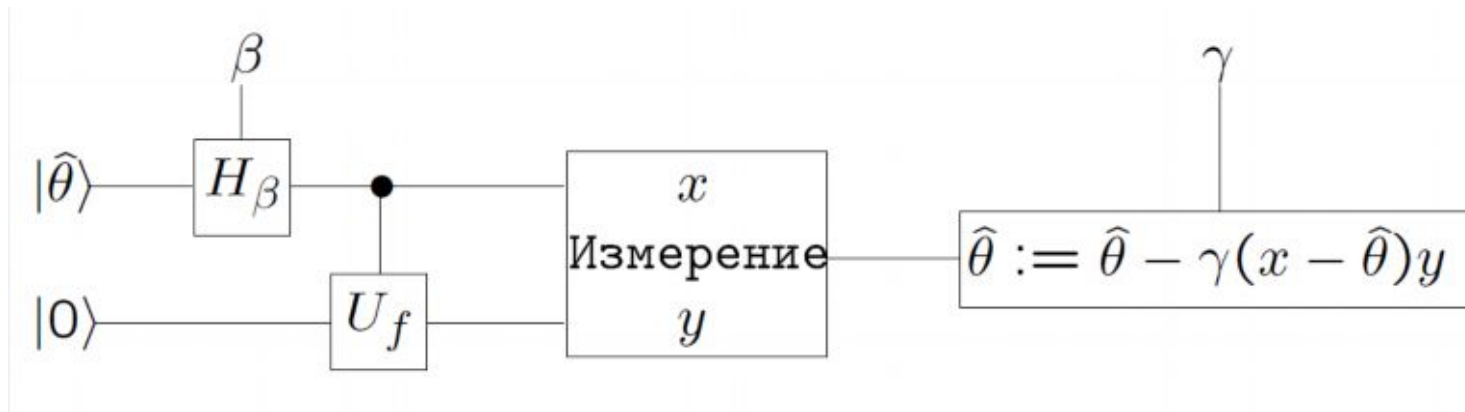
Преимущества SPSSA

- Асимптотически оптимальная скорость сходимости
 - Min число измерений на итерации
 - Состоятельность при почти произвольных помехах
 - Работоспособность в нестационарных задачах
-
- Легко может быть реализован на квантовом компьютере (или мультиагентной системой)

Вычисление градиента «за такт»

$$\hat{\theta}_k = \hat{\theta}_{k-1} - \gamma_k (x_k - \hat{\theta}_{k-1}) y_k, \quad x_k = \hat{\theta}_{k-1} + \beta_k \Delta_k, \quad y_k = f(x_k) + v_k, \quad \gamma_k = \frac{\alpha_k}{\beta_k^2}$$

$$U_f : |x\rangle |z\rangle \rightarrow |x\rangle |z \oplus f(x)\rangle, \quad x_k = H_\beta |\hat{\theta}_{k-1}\rangle = \frac{1}{2^{\frac{d}{2}}} \sum_{\Delta_i \in \{-1, +1\}^d} |\hat{\theta}_{k-1} + \beta_k \Delta_i\rangle$$



$$|\hat{\theta}_{k-1} + \beta_k \Delta_i\rangle |f(\hat{\theta}_{k-1} + \beta_k \Delta_i)\rangle, \quad \Delta_i \in \{-1, +1\}^d.$$

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



Благодарю за внимание!
Вопросы?

Санкт-Петербург, 2012 г.