Структурный синтез и планирование вычислений

Костенко Валерий Алексеевич

kost@cs.msu.su

Задачи комбинаторной оптимизации

Множество решений включает всевозможные:

- размещения исходно заданных объектов,
- упорядочивания исходно заданных объектов,
- разбиения на группы исходно заданных объектов.

Комбинаторная оптимизация включает в себя задачи оптимизации, в которых множество допустимых решений дискретно или может быть сведено к дискретному множеству.

Задача о рюкзаке

Дан рюкзак заданного объема и набор предметов.

Каждый предмет характеризуется:

- стоимостью
- и объемом.

Требуется упаковать рюкзак так, чтобы:

- суммарная стоимость упакованных предметов была максимальной;
- суммарный объем упакованных предметов не превышал объем рюкзака.

Задача о рюкзаке

Дано:

•*b* - объем рюкзака*,*

•$\{(a_i\ ,\ c_i),\ I = 1,\ldots,n\}$

n – число предметов,

a_i и c_i. - объем и стоимость предмета *i*.

Решение будем описывать вектором:

$X = (x_{1,},\ x_2,\ \ldots.,x_n)$

Если предмет упакован в рюкзак, то $x_i=1$, если нет $x_i=0$.

Задача о рюкзаке

Целевая функция:

$$\max_X \left(\sum_{i=1}^{n} c_i \cdot x_i \right)$$

Функция ограничений:

$$\sum_{i=1}^{n} a_i \cdot x_i \le b$$

.

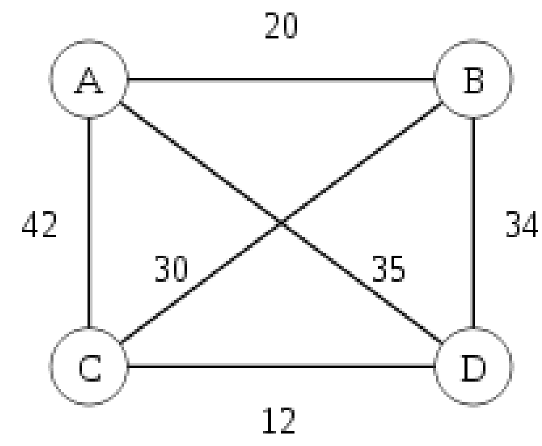
Задача о рюкзаке

Если все коэффициенты a_i целые числа и b целое число, то задача является *NP*-трудной.

Если все коэффициенты a_i вещественные, то задача может быть решена жадным алгоритмом сложности $O(n \cdot log\ n)$.

Задача коммивояжера

Задачу коммивояжёра можно представить в виде нахождения маршрута на графе.



Вершины графа – города.

Ребра - пути сообщения между этими городами.

Каждому ребру сопоставлен вес.

Задача заключается в отыскании кратчайшего маршрута, в который входит по одному разу каждая вершина графа.

Длина маршрута - сумма весов входящих в него ребер.

Задача коммивояжера

Маршрут может быть описан циклической перестановкой номеров городов:

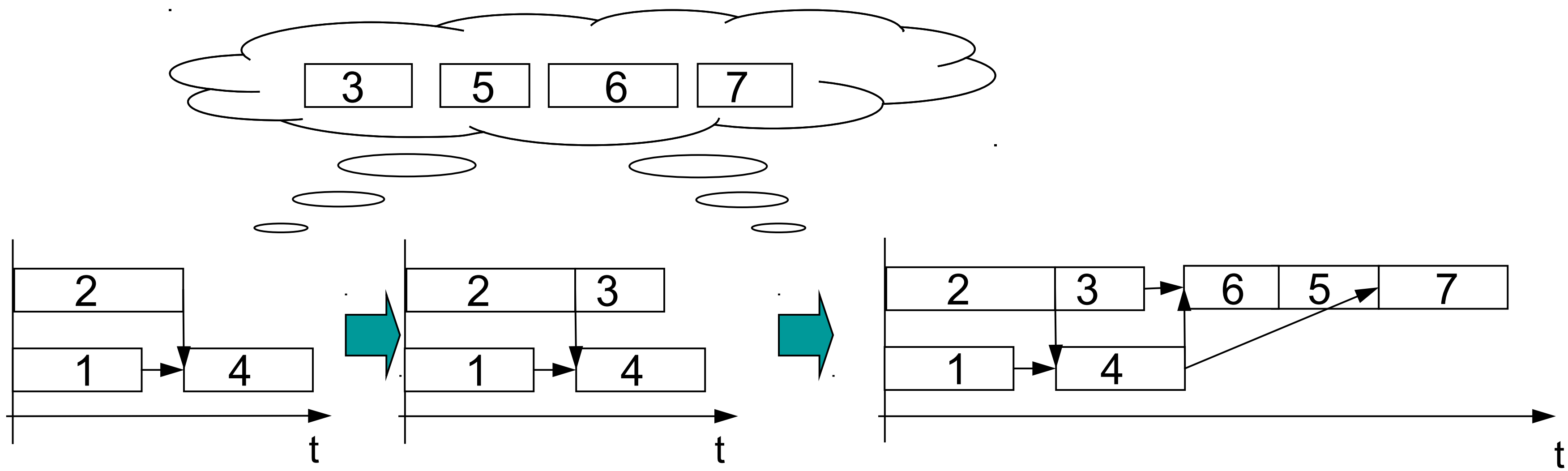
$$J = (j_1, j_2, \ldots j_n, j_{n+1})$$

где j_i - номер города находящийся в *i* –ой позиции перестановки.

Пространством поиска решений является множество перестановок *n* городов, таких, что все $j_1, j_2, \ldots j_n$ разные и $j_1 = j_{n+1}$

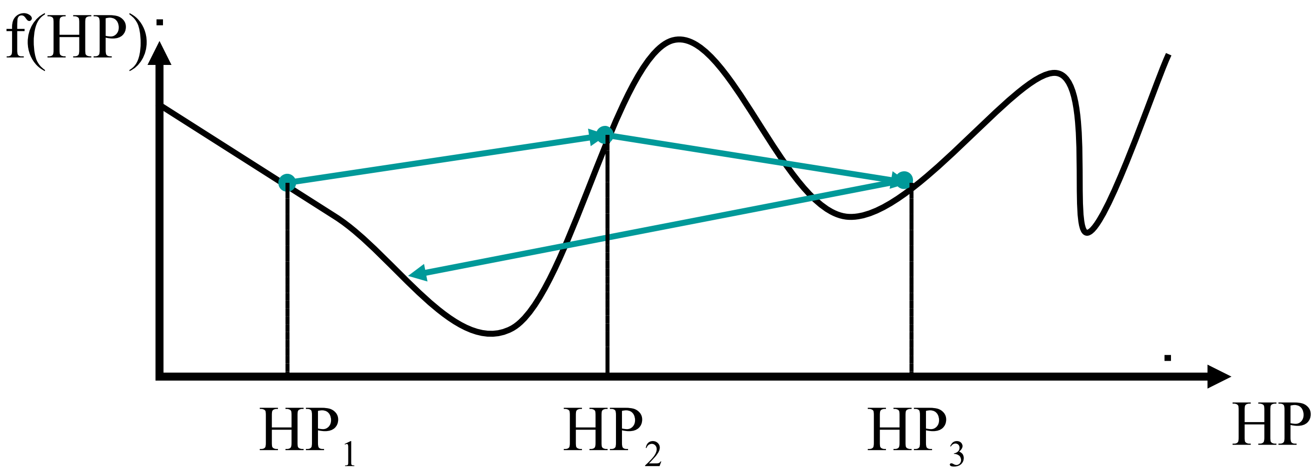
Данная задача принадлежит к классу *NP*-трудных задач.

Конструктивные алгоритмы построения расписаний



- Жадные алгоритмы
- Метод ветвей и границ
- Алгоритмы сочетающие жадные стратегии и стратегии ограниченного перебора

Итерационные алгоритмы построения расписаний



- Алгоритмы случайного поиска
- Имитация отжига
- Алгоритмы локальной оптимизации
- Генетические и эволюционные алгоритмы

Алгоритмы имитации отжига

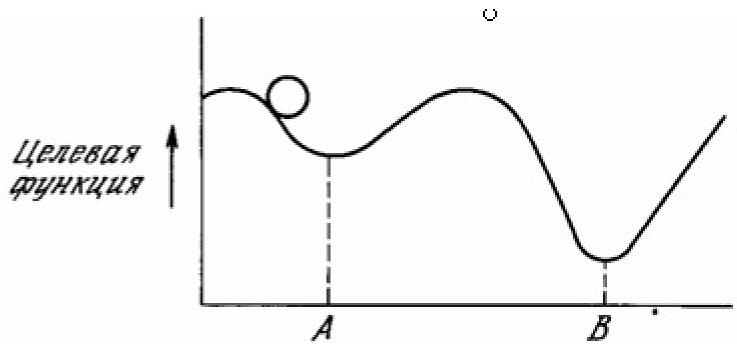
Алгоритм основывается на имитации физического процесса, который происходит при отжиге металлов.

Предполагается, что атомы уже выстроились в кристаллическую решётку, но ещё допустимы переходы отдельных атомов из одной ячейки в другую.

Переход атома из одной ячейки в другую происходит с некоторой вероятностью, причём вероятность уменьшается с понижением температуры.

Устойчивая кристаллическая решётка соответствует минимуму энергии атомов, поэтому атом или переходит в состояние с меньшим уровнем энергии, или остаётся на месте.

Вероятностный закон принятия нового решения текущим решением



- Алгоритмы имитации отжига с некоторой вероятностью допускают переход в состояние с более высоким значением целевой функции:

$$P(X^k \to X^{k+1}) = \begin{cases} 1, & \Delta F \le 0 \\ \exp(-\Delta F / T), & \Delta F > 0 \end{cases}$$

- где T - некоторая температура, ΔF - изменение целевой функции.
- В ходе работы алгоритма температура постепенно снижается.

Общая схема

1. Задать начальное решение $X_0 \in S$ и считать его текущим $X=X_0$.
2. Установить начальную температуру T_0, приняв ее текущей: $T=T_0$
3. Применить операции преобразования к X и получить новый вариант решения $X' \in S$, если оно является лучшим из ранее найденных, то запомнить его.
4. Найти изменение функции оценки качества решения : $\Delta F = F(X') - F(X)$:
   – если $\Delta F \leq 0$ (решение улучшилось), то $X=X'$;
   – если $\Delta F > 0$, то принять с вероятностью в качестве текущего решения X'.
5. Повторить заданное число раз шаги 3 и 4 без изменения текущей температуры.
6. Если критерий останова выполнен, то завершение работы.
7. Понизить текущую температуру и перейти к шагу 3..

Вероятность принятия решения на шаге 4:

$$p = e^{\frac{-\Delta F}{T}}$$

Законы понижения температуры на шаге 7:

- Закон Больцмана: $$T = \frac{T_0}{\ln(1+i)}$$
- Закон Коши: $$T = \frac{T_0}{1+i}$$
- $$T = T_0 \frac{\ln(1+i)}{1+i}$$

i - номер итерации алгоритма

Асимптотическая скорость сходимости

Чтобы достичь наперед заданной точности, нужно совершить число итераций, пропорциональное квадрату от размера пространства поиска.

Повышение эффективности алгоритма

- Динамические законы понижения температуры.

- Распараллеливание алгоритмов.

- Эвристические (направленные) операции преобразования текущего решения.

Генетический алгоритм Холланда (SGA)

- Holland J.N. Adaptation in Natural and Artificial Systems. Ann Arbor, Michigan: Univ. of Michigan Press, 1975.

Генетический алгоритм Холланда (SGA)

- Основан на использовании механизмов естественной эволюции:
  1. Изменчивость → операция мутации
  2. Наследственность → операция скрещивания
  3. Естесственный отбор → операция селекции

Основные понятия

- ***Популяция*** - это множество битовых строк.
- ***Каждая строка*** - одно из возможных решений задачи.
- По строке может быть вычислена ***функция выживаемости***, которая характеризует качество решения.
- Основные операции алгоритма: ***селекция, скрещивание и мутация*** выполняются над элементами популяции.

Схема ГА

1. Сгенерировать случайным образом начальную популяцию.
2. Вычислить функцию выживаемости для каждой строки популяции.
3. Выполнить операцию селекции.
4. Выполнить операцию скрещивания:
   - Выбрать пары для скрещивания.
   - Для каждой выбранной пары выполнить скрещивание, получить двух потомков и произвести в популяции замену родителей на их потомков.
5. Выполнить операцию мутации.
6. Если критерий останова не достигнут, перейти к п.2, иначе завершить работу.

Требования к кодированию решений

- Для задач непрерывного и целочисленного мат. программирования, оптимизируемые параметры задаются:
  - ***двоичным кодом числа,***
  - ***кодами Грея***.

Создание начальной популяции

- Случайным образом генерируется начальная популяция в пределах допустимых значений (в области поиска):

$$X_1[10100..01], \quad X_2[11100..11], \quad \ldots, \quad X_N[01010..10]$$

Функция выживаемости

- Выбирается согласно предметной области
- Определяет качество решения
- Применяется ко всем элементам популяции

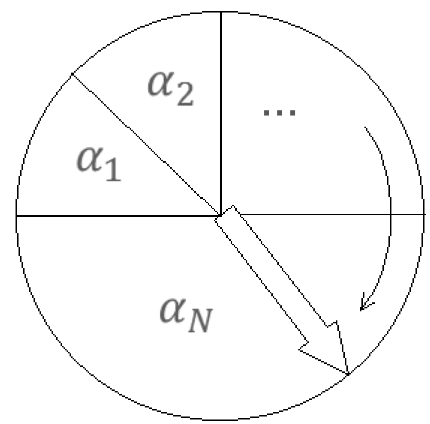
Операция селекции

- Схема пропорциональной селекции:

$$\bar{F} = \frac{\sum F_i}{N}, \quad r_i = \left\lfloor {}^{F_i}/_{\bar{F}} \right\rfloor$$

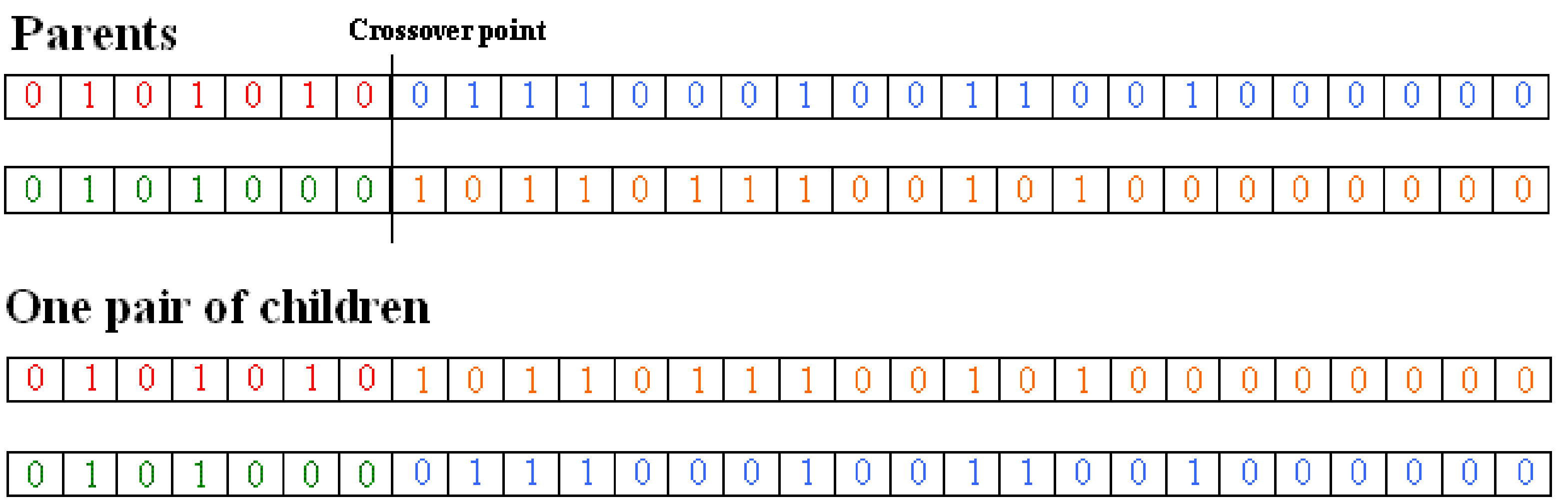
- Схема рулетки:

$$\alpha_i = 2\pi \frac{F_i}{\bar{F}}$$



Операция скрещивания

- Параметр – порог вероятности скрещивания p_{cr}
- Одноточечное скрещивание:



Операция мутации

- Параметр – порог вероятности мутации p_{mut}
- Если $p < p_{mut}$, то бит инвертируется:

$$X = (100110\ldots001)$$

$$p < p_{mut}$$

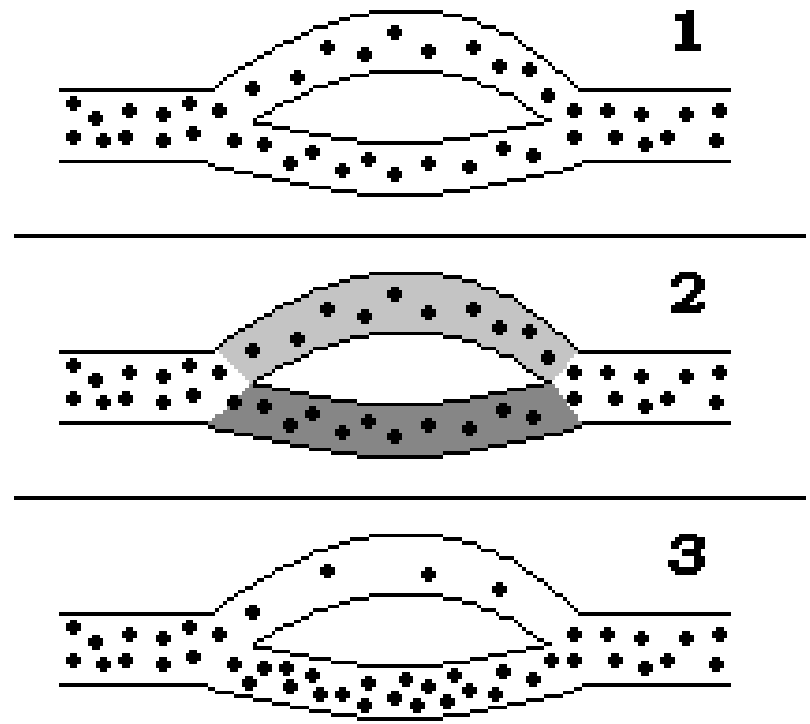
$$\hat{X} = (100010\ldots001)$$

Критерий останова

- Процесс продолжается итерационно
- Варианты критерия останова:
  - Выполнение заданного числа итераций
  - Выполнение заданного числа итераций без улучшения
  - Достижение заданного значения функции выживаемости

Муравьиные алгоритмы (биологическая модель)

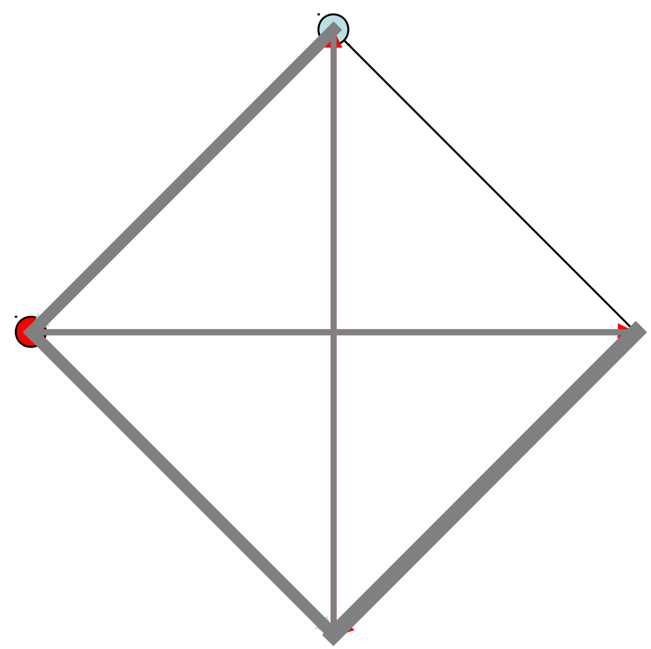
1. Изначально вероятности выбора маршрутов равны
2. Муравьи, выбравшие кратчайший маршрут, возвращаются быстрее, количество феромона на коротких маршрутах больше
3. Вероятность выбора кратчайшего маршрута повышается



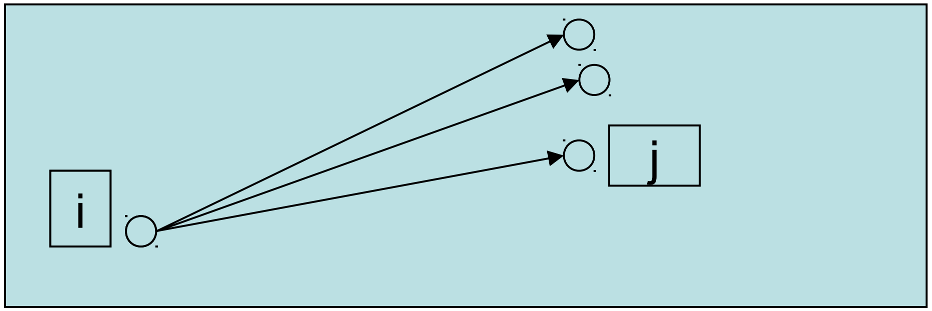
Муравьиные алгоритмы
(МА для решения задачи коммивояжера)

Общая схема итерации:

- «Создание» муравьев
- Построение маршрутов муравьями
- Обновление феромонного следа на найденных маршрутах



Муравьиные алгоритмы
(построение маршрута и обновление феромонов)

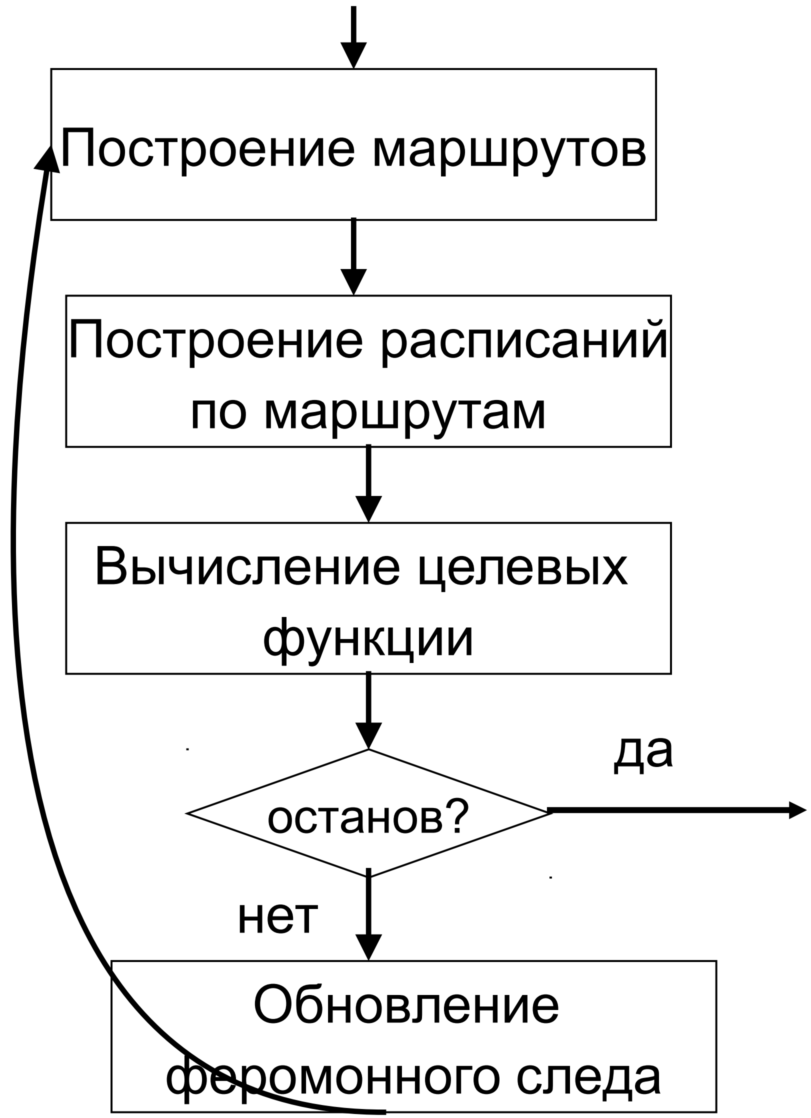


$$P_{ij,k}(t)=\begin{cases}\dfrac{\left(\tau_{ij}(t)\right)^{\alpha}\cdot\left(\eta_{ij}(t)\right)^{\beta}}{\sum\limits_{l\in J_k}\left(\tau_{il}(t)\right)^{\alpha}\cdot\left(\eta_{il}(t)\right)^{\beta}}, & j\notin L_k\\ 0, & j\in L_k\end{cases}$$

$$\tau_{ij}(t+1)=(1-p)\cdot\tau_{ij}(t)+\sum_{k=1}^{m}\Delta\tau_{ij,k}(t)$$

$$\Delta\tau_{ij,k}(t)=\begin{cases}F\left(T_k(t)\right), & (i,j)\in T_k(t)\\ 0, & (i,j)\notin T_k(t)\end{cases}$$

Муравьиные алгоритмы (использование для построения расписаний)



Алгоритмы случайного поиска

- Основой методов случайного поиска служит итерационный процесс:

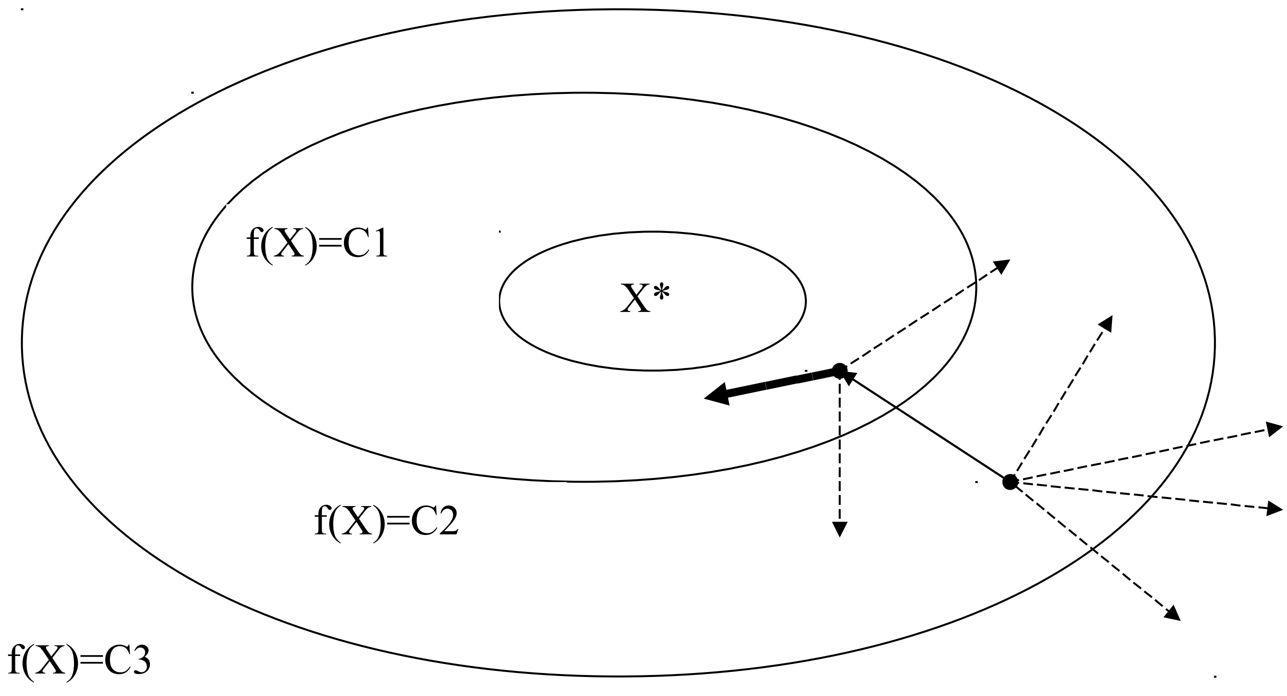
$$X_{k+1} = X_k + \alpha_k \cdot \frac{\xi}{\|\xi\|}, k = 0,1,\ldots,$$

αk – величина шага,

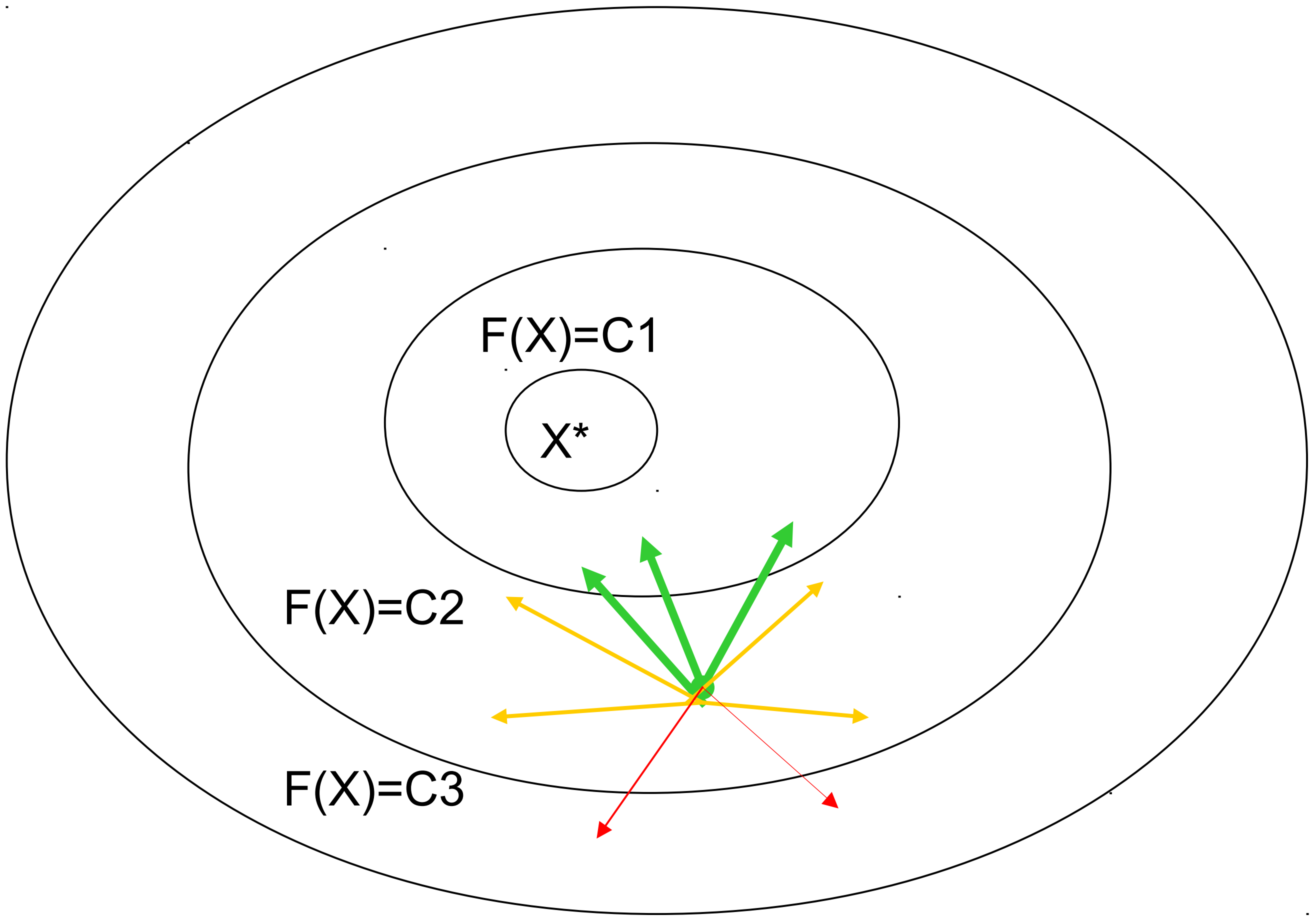
$\xi = (\xi 1, \cdots, \xi n)$ – некоторая реализация *n*-мерного случайного вектора ξ.

- Л.А. Растригин. Статистические методы поиска.- М.: Наука, 1968.

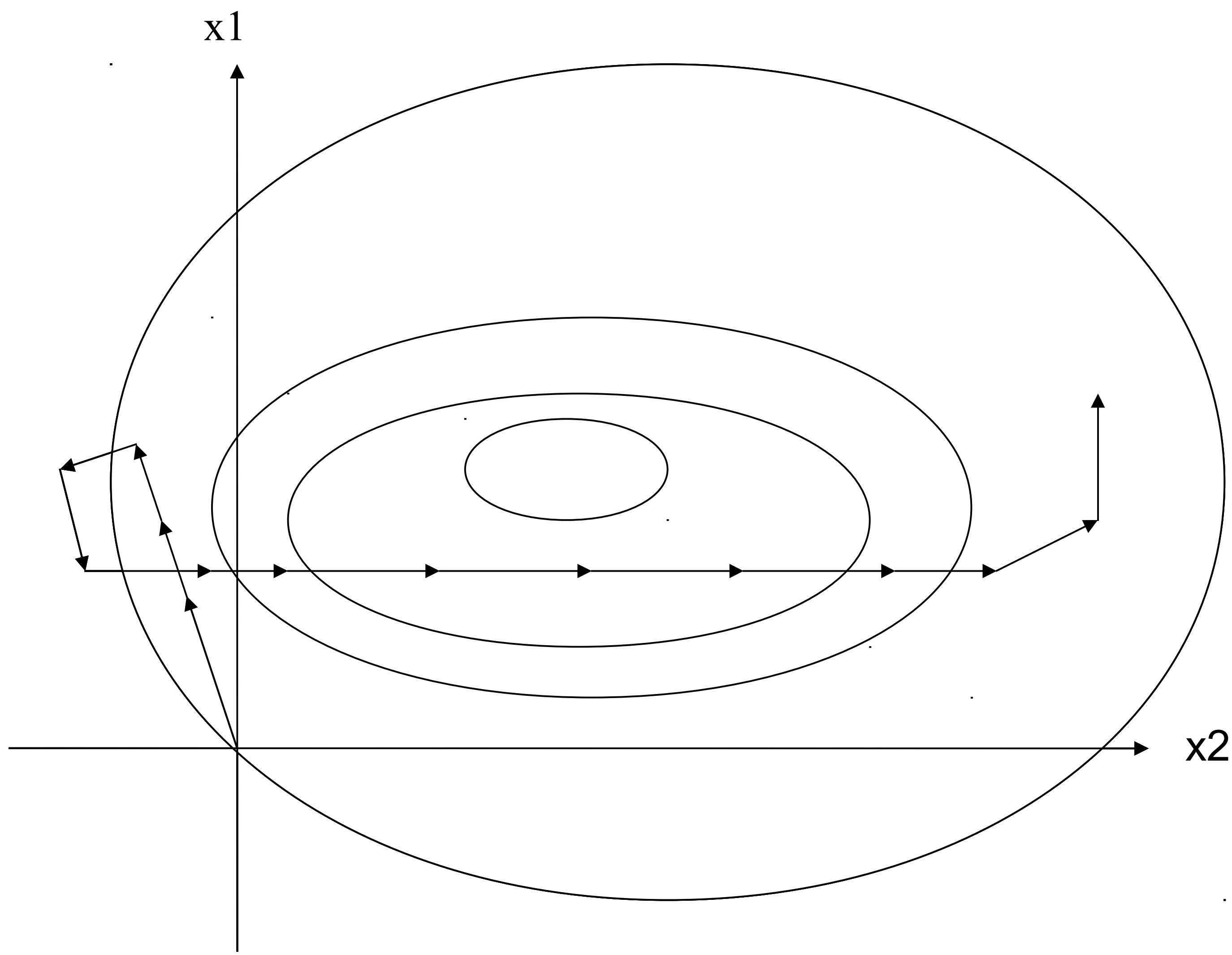
Алгоритмы случайного поиска



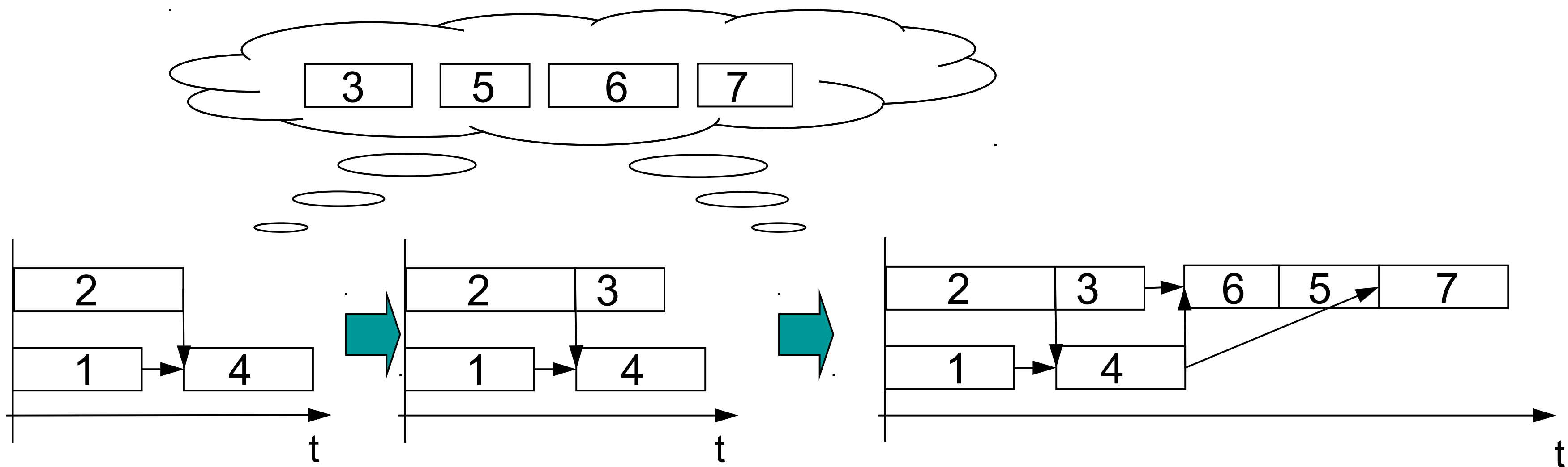
Случайный поиск с самообучением



Случайный поиск с самообучением



Конструктивные алгоритмы построения расписаний



- Жадные алгоритмы
- Метод ветвей и границ
- Алгоритмы сочетающие жадные стратегии и стратегии ограниченного перебора

Жадные алгоритмы
(общая схема)

Разложимые функции:

$$\min_{(x,y)} f(x,y) = \min_{(x)} f_1(x, \min_{(y)} f_2(y))$$

1. Выбрать очередную работу ≡ переменную *x* или *y*.

2. Выбрать в соответствии с локальной функцией (f_1,f_2) место размещения работы ≡ присвоить значение переменной.

Задача построения расписания выполнения работ в одноприборном устройстве

- Шина может рассматриваться как одноприборное устройство, обслуживающее исходно заданный набор работ без прерываний.

- Расписание выполнения работ, представляет собой упорядоченное множество

$$H = \left\{ s_j^* \right\}_{k=1}^{N_H}, \ j \in J$$

- $J = \{(t_j, s_j, f_j)\}$ - исходно заданный набор работ

Задача построения расписания выполнения работ в одноприборном устройстве

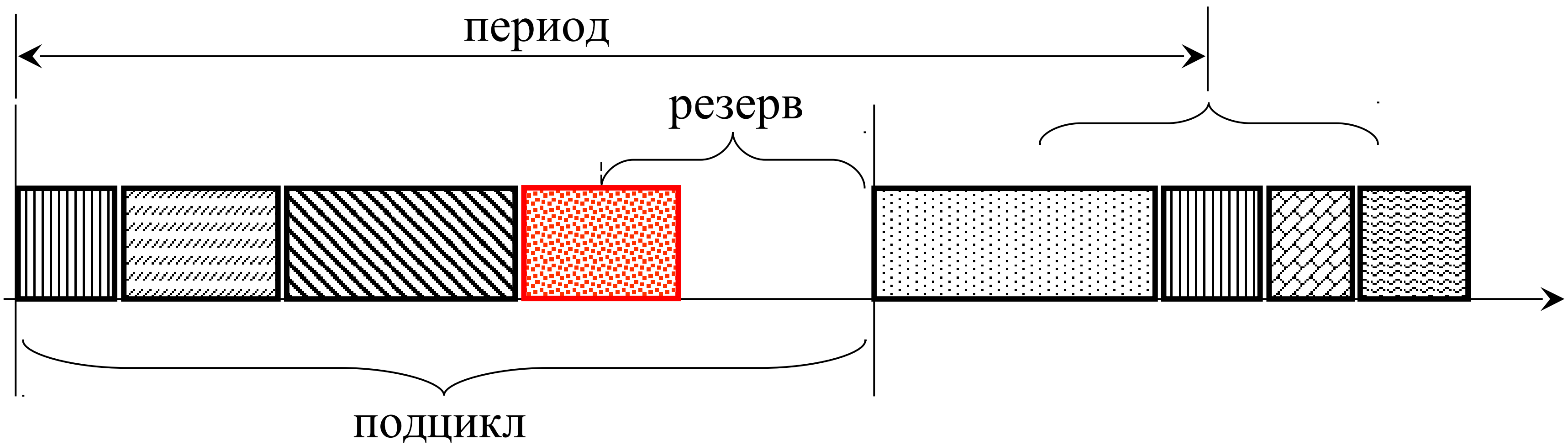
- Множество корректных расписаний $H^{*'}$ определим набором ограничений:

$$g_1 : (\forall j \in H) \Rightarrow \left((s_j^* \geq s_j) \wedge (f_j^* \leq f_j)\right)$$

$$g_2 : (\forall j \in H) \Rightarrow \left(f_j^* - s_j^* = t_j\right)$$

$$g_3 : (\forall (j,l) \in H, j \neq l) \Rightarrow \left(((s_j^* < s_l^*) \vee (s_j^* \geq f_l^*)) \wedge ((f_j^* \leq s_l^*) \vee (f_j^* > f_l^*))\right)$$

Пример: технологические ограничения на расписание передачи сообщений по шине



- Одна цепочка работ в подцикле
- Резерв времени в конце подцикла
- Максимальная длина цепочки работ
- Максимальное отклонение расстояния между последовательными работами одного сообщения от периода сообщения

Жадные алгоритмы (построение расписания выполнения работ в одноприборном устройстве)

- Для частной задачи:

$$\max_{H \in H^{*'}} |H|$$

$$\forall\, j\colon t_j = f_j - s_j$$

- известен оптимальный жадный алгоритм сложности *O(n·log n)*.

Жадные алгоритмы (*GrA* - алгоритм построения расписания для одноприборного устройства)

1. Упорядочиваем работы по возрастанию f_j. Заявки с одинаковым значением f_j располагаем в произвольном порядке. Сложность - $O(n \cdot \log n)$.
2. Размещаем в расписание работу $j=1$.
3. Ищем первую работу для которой $s_i \leq f_j$, размещаем ее в расписание и $j=i$.
4. Шаги 2, 3 повторяем пока список не исчерпан. Количество повторов - $O(n)$.

Иллюстрация работы алгоритма

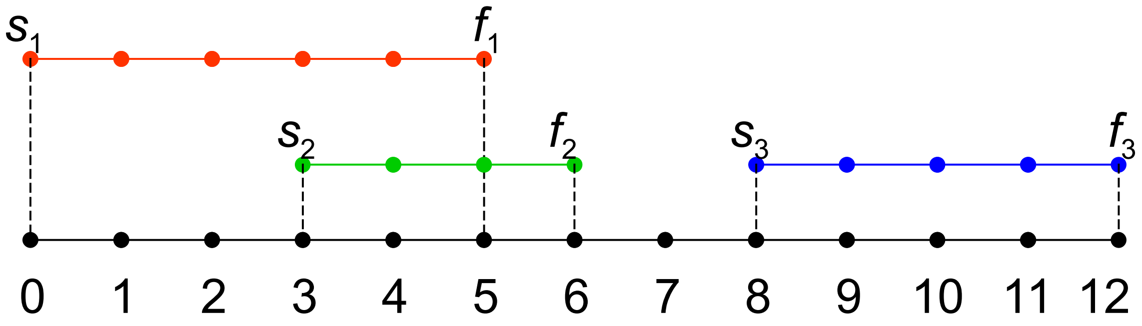


Иллюстрация работы алгоритма

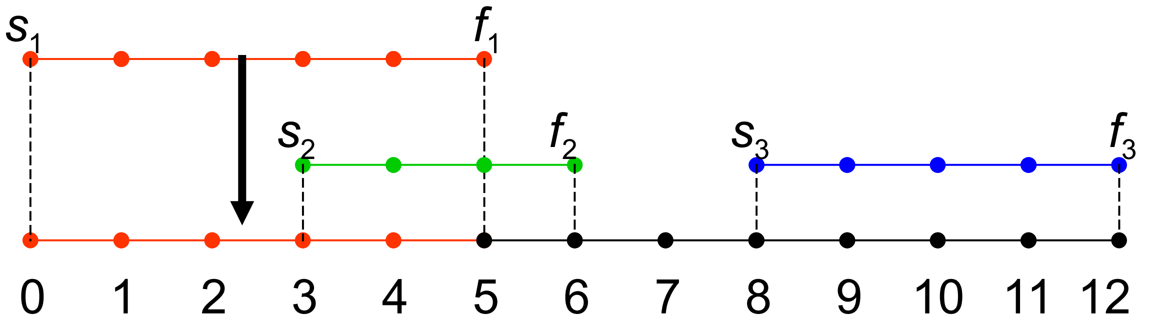


Иллюстрация работы алгоритма

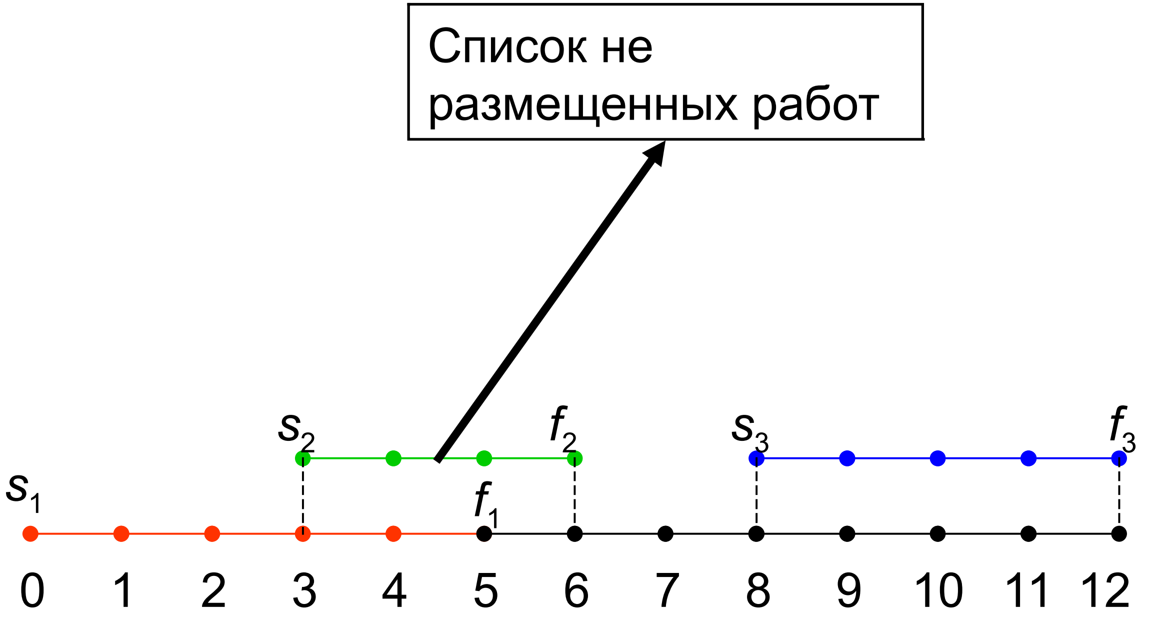


Иллюстрация работы алгоритма

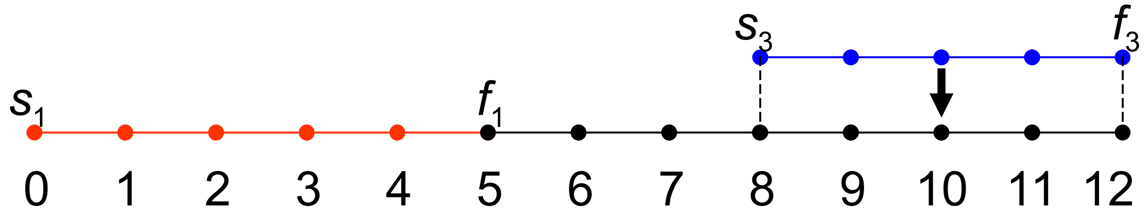
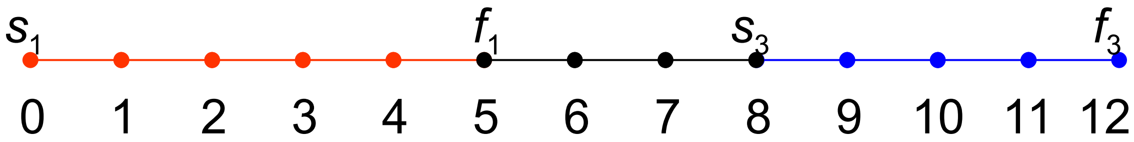
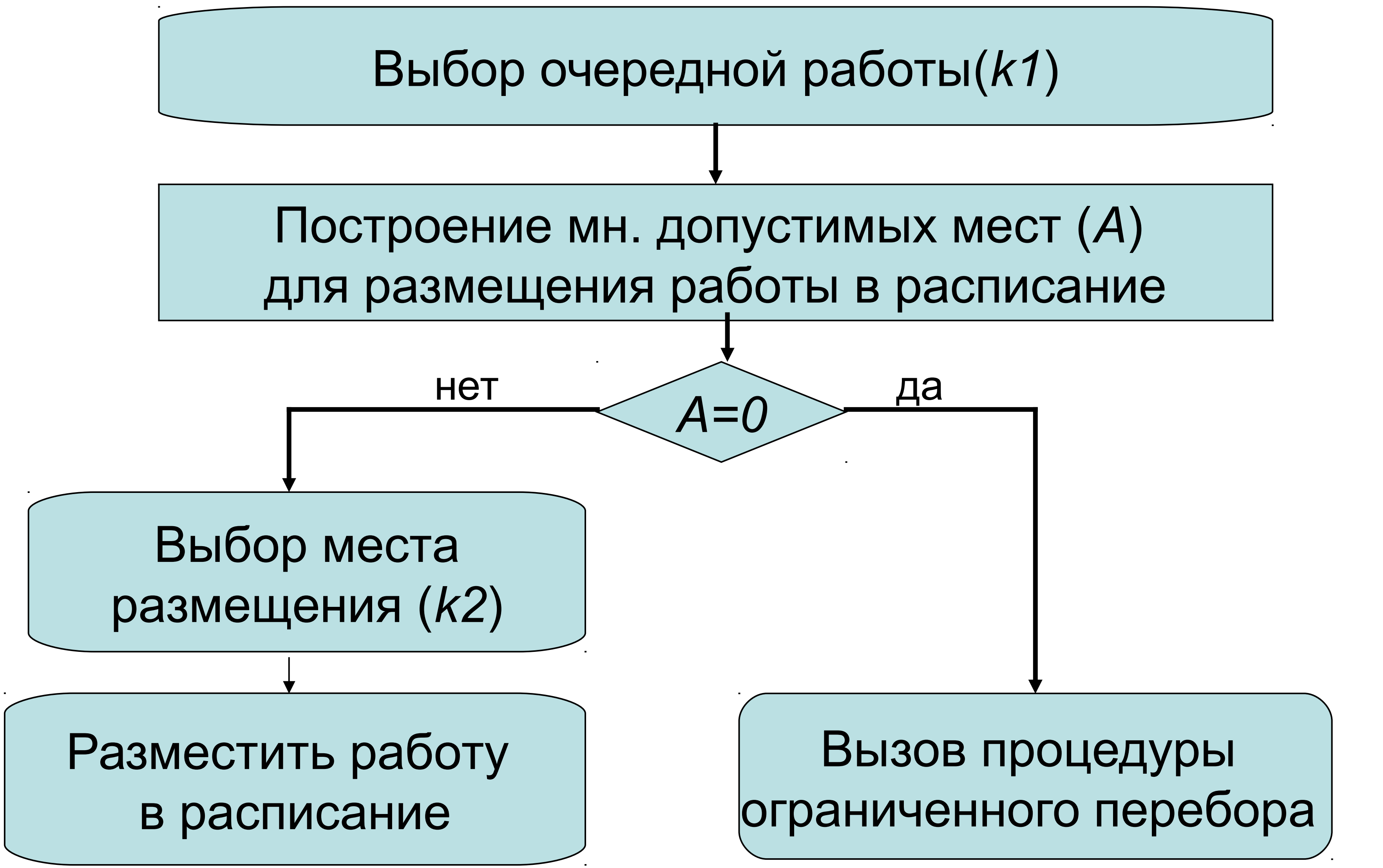


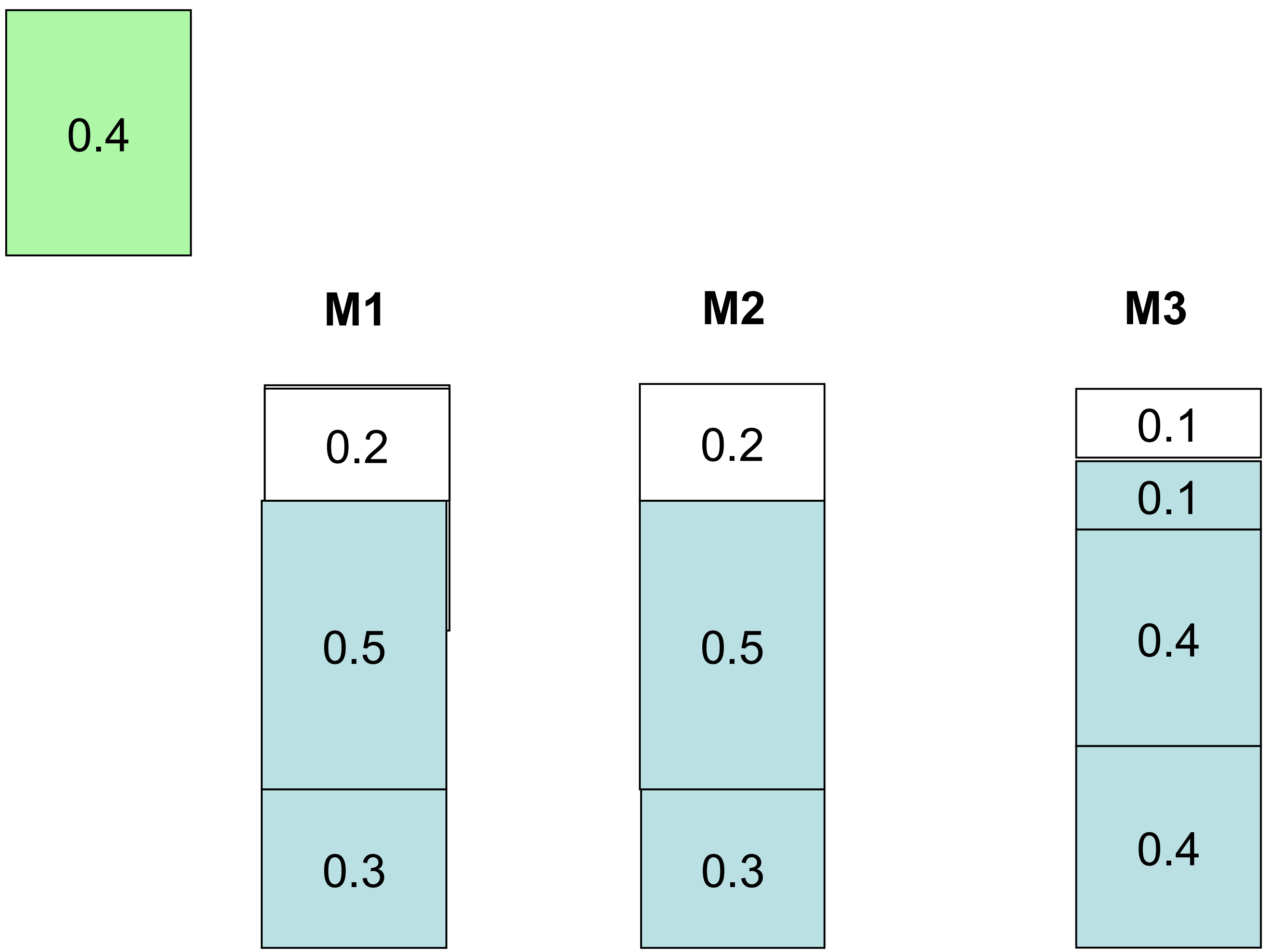
Иллюстрация работы алгоритма



Алгоритмы сочетающие жадные стратегии и стратегии ограниченного перебора



Пример процедуры ограниченного перебора



Направления работы семинара
Синтез архитектур и планирование вычислений

Разработка алгоритмов решения конкретных задач структурного синтеза и планирования вычислений:

- информационно-управляющие системы реального времени,
- центры обработки данных.

Разработка новых подходов к построению алгоритмов и модификаций известных классов алгоритмов. Примеры:

- генетический алгоритм с самообучением,
- метод построения алгоритмов сочетающих жадные стратегии и ограниченный перебор.

Аксиоматический подход к построению алгоритмов распознавания поведения динамических систем:

- алгоритмы распознавания строятся методами машинного обучения,
- повышенная устойчивость к шумам.

- Содержательная формулировка задачи
- Математическая формулировка задачи
- Анализ свойств задачи
- Выбор класса алгоритмов (или выбор известного алгоритма)
- Разработка и реализация алгоритма
- Исследование свойств алгоритма