

Жадные алгоритмы.

**Алгоритмы, сочетающие жадные стратегии
и ограниченный перебор.**

Жадные алгоритмы (общая схема)

$$\min_{(x,y)} f(x, y)$$

1. В соответствии с жадным критерием (локальной целевой функцией) $k1$ выбрать переменную x или $y \equiv$ очередную работу.
2. В соответствии с жадным критерием $k2$ Присвоить значение переменной \equiv выбрать место размещения работы в расписании.

Жадные алгоритмы (общая схема)

Разложимые функции:

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

1. В соответствии с жадным критерием выбрать переменную x или $y \equiv$ очередную работу.
2. Присвоить в соответствии с локальной функцией (f_1, f_2) значение переменной \equiv выбрать место размещения работы.

Задача построения расписания выполнения работ в одноприборном устройстве

Дано множество работ:

$$J = \{ \langle t_j, s_j, f_j \rangle, j = 1 \dots n \}$$

Расписание представляет собой упорядоченное множество работ:

$$H = \{ s_j^*, j \in [1 \dots n] \}$$

$$f_j^* = s_j^* + t_j, \quad |H| \leq n$$

Задача построения расписания выполнения работ в одноприборном устройстве

- Множество корректных расписаний H^* определим набором ограничений:

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

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

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

Задача построения расписания выполнения работ в одноприборном устройстве

- Задача:

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

- известна в теории расписаний как задача о выборе максимального числа совместимых заявок и является NP –трудной.

Задача построения расписания выполнения работ в одноприборном устройстве

- Для частной задачи:

$$\max_{H \in H^{*'}} |H|$$
$$\forall j: t_j = f_j - s_j$$

- известен оптимальный жадный алгоритм сложности $O(n \cdot \log n)$.

Жадные алгоритмы

(GrA - алгоритм построения расписания для одноприборного устройства)

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

Иллюстрация работы алгоритма

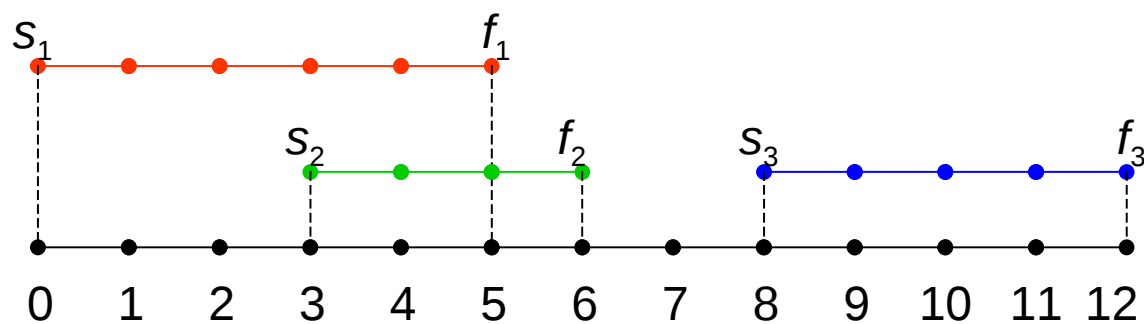


Иллюстрация работы алгоритма

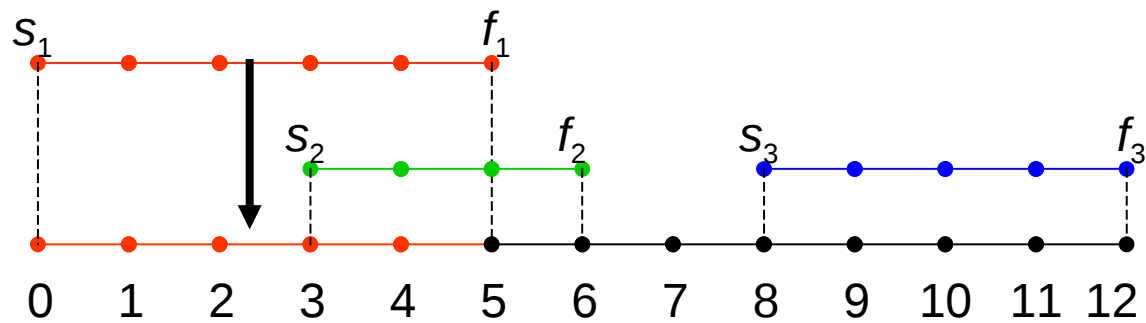


Иллюстрация работы алгоритма

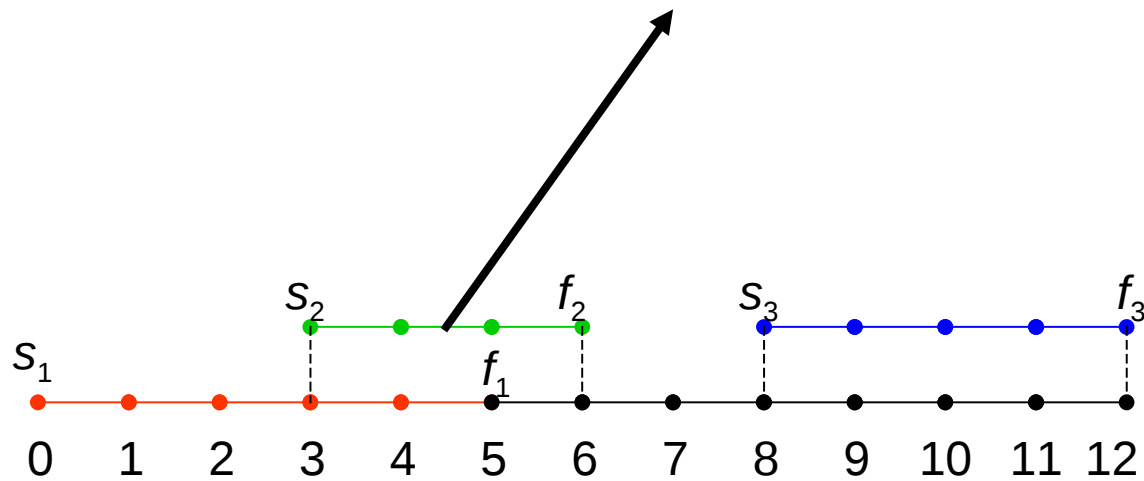


Иллюстрация работы алгоритма

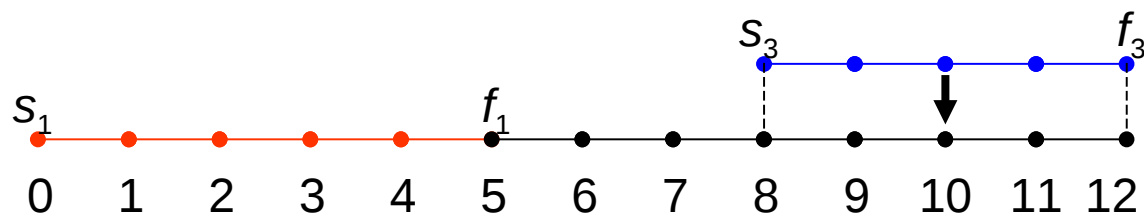
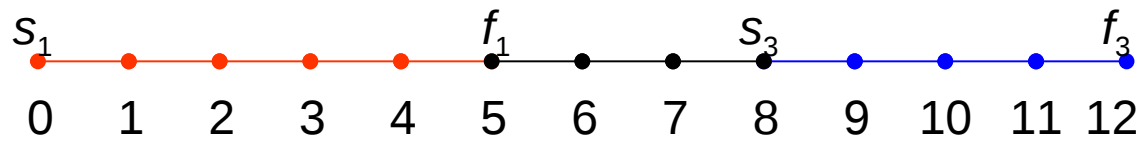


Иллюстрация работы алгоритма



Жадные алгоритмы

(оптимальность алгоритма *GrA*)

Теорема. Алгоритм *GrA* включает в расписание наибольшее возможное количество совместимых работ.

Доказательство.

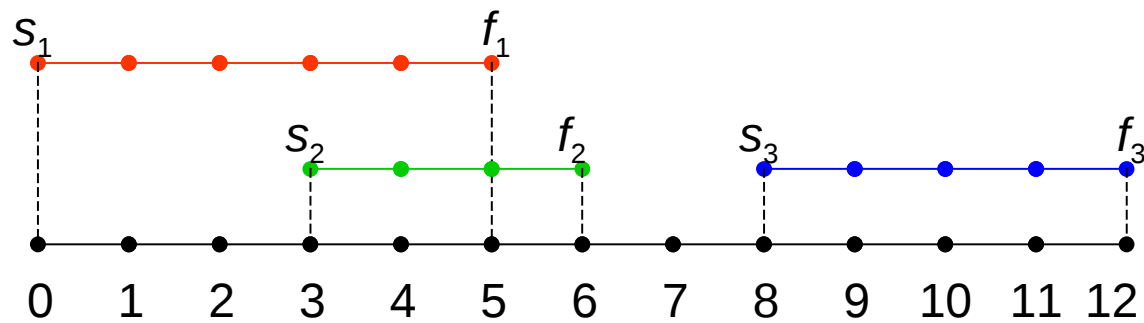
- ✓ Если в каком-то оптимальном расписании работа 1 отсутствует, то можно в нем заменить заявку с самым меньшим f на работу 1.
- ✓ Не нарушится совместимость работ и не изменится их количество в расписании.

Жадные алгоритмы

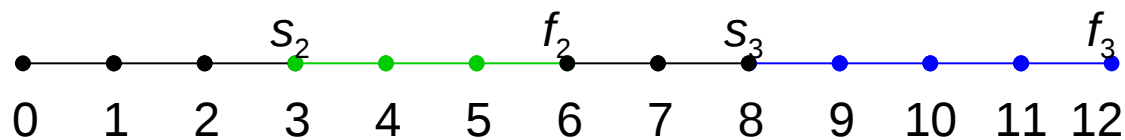
(оптимальность алгоритма *GrA*)

- ✓ Т.е. можно искать оптимальное расписание содержащее работу 1 → существует оптимальное расписание, начинающееся с жадного выбора.
- ✓ Из исходного набора можно удалить все работы несовместимые с 1.
- ✓ Задача сводится, к выбору набора работ из множества оставшихся работ, т.е. мы свели задачу к аналогичной задаче с меньшим числом работ.
- ✓ Рассуждая по индукции, получаем, что делая на каждом шаге жадный выбор, мы придем к оптимальному расписанию.

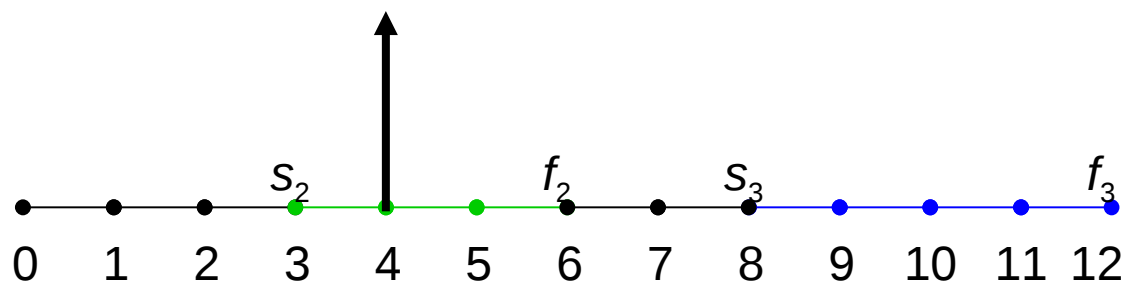
Доказательство оптимальности алгоритма



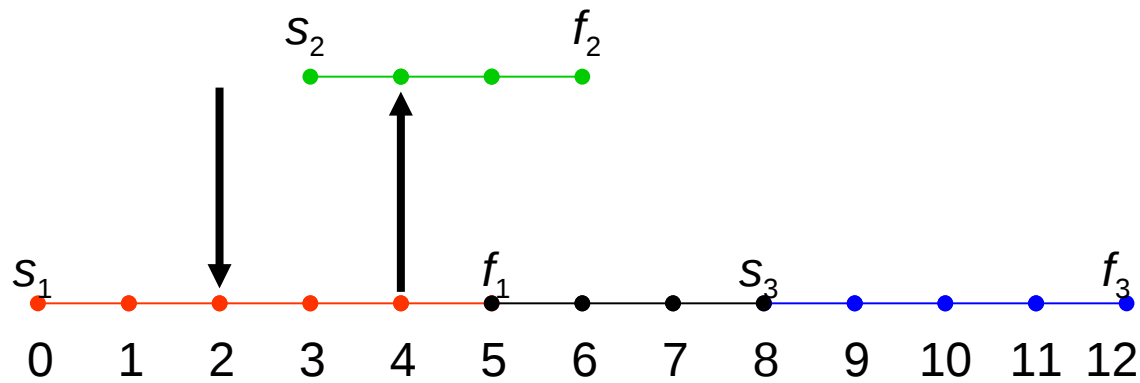
Доказательство оптимальности алгоритма



Доказательство оптимальности алгоритма



Доказательство оптимальности алгоритма



Жадные алгоритмы (как доказать, что алгоритм получает оптимальное решение)

1. Доказываем что жадный выбор на первом шаге не исключает возможности получения оптимального решения.
2. Показываем, что подзадача, возникающая после жадного выбора на первом шаге, аналогична исходной.
3. Рассуждение завершается по индукции.

- **Кормен Т., Лейзерсон Ч., Ривест Р.
Алгоритмы: построение и анализ. М.:
МЦНМО, 2000.**

Жадные стратегии + ограниченный перебор.

Принцип построения алгоритмов.

- На каждом шаге алгоритм выбирает заявку из множества еще не размещенных заявок и место ее размещения в соответствии с жадным критерием.
- **Подход 1.** Процедура ограниченного перебора вызывается, если на очередном шаге выбранная заявка не может быть размещена.
- **Подход 2.** процедура ограниченного перебора вызывается, если после пробного размещения на очередном шаге выбранной заявки в множестве неразмещенных заявок появляются заявки, которые не могут быть размещены.
- Для процедуры ограниченного перебора задается максимально допустимая глубина перебора. Она определяет максимально возможное количество заявок, которые могут участвовать в переборе.

Жадные стратегии + ограниченный перебор.

Схема алгоритма (подход 1).

1. Выбрать в соответствии с жадным критерием очередную заявку из множества еще не размещенных заявок.
2. Выбрать в соответствии с жадным критерием ресурс (место в расписании) для размещения заявки:
 - если такой ресурс не найден, то к п. 3,
 - если такой ресурс найден, то разместить на нем заявку, удалить заявку из множества неразмещенных заявок и, если это множество не пусто, перейти к п.1, в противном случае – завершение работы.
3. Вызов процедуры ограниченного перебора (или эвристики):
 - успешное завершение процедуры: сохранить полученное процедурой размещение, удалить заявку из множества неразмещенных заявок и, если это множество не пусто, перейти к п.1, в противном случае – завершение работы.
 - неуспешное завершение процедуры: удалить текущую заявку из множества неразмещенных заявок и, если множество неразмещенных заявок не пусто, перейти к п.1, в противном случае – завершение работы.

Жадные стратегии + ограниченный перебор.

Схема алгоритма (подход 2).

1. Выбрать в соответствии с жадным критерием очередную заявку из множества еще не размещенных заявок.
2. Выбрать в соответствии с жадным критерием ресурс для размещения заявки (всегда есть хоть один ресурс), сделать пробное размещение и проверить оптимальность жадного выбора:
 - если в множестве неразмещенных заявок есть заявки, которые не могут быть размещены после размещения выбранной заявки, то отменить размещение и перейти к п.3,
 - если в множестве неразмещенных заявок нет заявок, которые не могут быть размещены после размещения выбранной заявки, то удалить заявку из множества неразмещенных заявок и, если это множество не пустое, перейти к п.1.
3. Вызов процедуры ограниченного перебора (или эвристики):
 - успешное завершение процедуры: сохранить полученное процедурой размещение, удалить заявку из множества неразмещенных заявок и если, это множество не пусто, перейти к п.1, в противном случае – завершение работы,
 - неуспешное завершение процедуры: удалить текущую заявку из множества неразмещенных заявок и, если множество неразмещенных заявок не пусто, перейти к п.1, в противном случае – завершение работы.

Жадные стратегии + ограниченный перебор. Свойства алгоритмов.

Функция называется разделяемой на f_1 и f_2 , если она представима в виде:

$$f(x, y) = f_1(x, f_2(y))$$

Функция называется разложимой на f_1 и f_2 , если она разделяема на f_1 и f_2 и функция f_1 монотонно не убывает по последнему аргументу.

Теорема оптимальности для разложимых функций:

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

Жадные стратегии + ограниченный перебор. Свойства алгоритмов.

У т в е р ж д е н и е . Если:

- целевая функция задачи $f(x_1, x_2, \dots, x_n)$ является разложимой на f_1, f_2, \dots, f_n ,
- в качестве жадного критерия выбора очередной оптимизируемой переменной (заявки) используется номер функции,
- в качестве жадного критерия выбора значения переменной (места размещения заявки) используется глобальный минимум соответствующей функции,
- существует решение, содержащее все заявки из исходно заданного набора,

то алгоритмы, построенные по схеме 1 или 2, получат оптимальное решение и процедура ограниченного перебора вызываться не будет..

Жадные стратегии + ограниченный перебор. Свойства алгоритмов.

Схема 1 будет иметь меньшую вычислительную сложность для задач, в которых возможно размещение большинства заявок по жадному критерию.

В случаи, когда жадные критерии «плохо подходят» для частной задачи, точность алгоритма, построенного по схеме 1, при одинаковой глубине перебора будет хуже точности алгоритма, построенного по схеме 2.

Модель физических ресурсов ЦОД

Модель физических ресурсов: $H = (P \cup M \cup K, L)$

- P – множество вычислительных узлов: $vh(p)$, $qh(p)$
- M – множество систем хранения данных: $uh(m)$, $type(m)$
- K – множество коммутационных элементов сети обмена:
 $th(k)$
- L – множество физических каналов передачи данных: $rh(l)$

Модель запроса

$$G = (W \cup S, E)$$

- W – множество виртуальных машин реализующих приложения: $v(w)$, $q(w)$
- S – множество storage-элементов: $u(s)$, $type(s)$
- E – множество виртуальных каналов: $r(e)$

Размещение элементов запроса на физические ресурсы ЦОД

Размещением запроса будем называть отображение

$$A: G \rightarrow H = \{W \rightarrow P, S \rightarrow M, E \rightarrow \{k, l\}\}$$

которое удовлетворяет ограничениям:

$$\sum_{w \in W_p} v(w) \leq v_h(p), \quad \sum_{w \in W_p} q(w) \leq q_h(p)$$

$$\sum_{e \in E_l} r(e) \leq r_h(l)$$

$$\sum_{e \in E_k} r(e) \leq \tau_h(k)$$

$$\sum_{s \in S_m} u(s) \leq u_h(m), \quad \forall s \in S_m : \text{type}(s) = \text{type}(m)$$

Граф остаточных ресурсов H_{res}

- Для графа H_{res} переопределены значения функций:
 $vh(p)$, $qh(p)$, $rh(l)$, $\tau h(k)$, $uh(m)$

$$vh_{res}(p) = vh(p) - \sum_{w \in W_p} v(w) \quad qh_{res}(p) = qh(p) - \sum_{w \in W_p} q(w)$$

$$rh_{res}(l) = rh(l) - \sum_{e \in E_l} r(e)$$

$$\tau h_{res}(k) = \tau h(k) - \sum_{e \in E_k} \tau(e)$$

$$uh_{res}(m) = uh(m) - \sum_{s \in S_m} u(s)$$

Задача распределения ресурсов

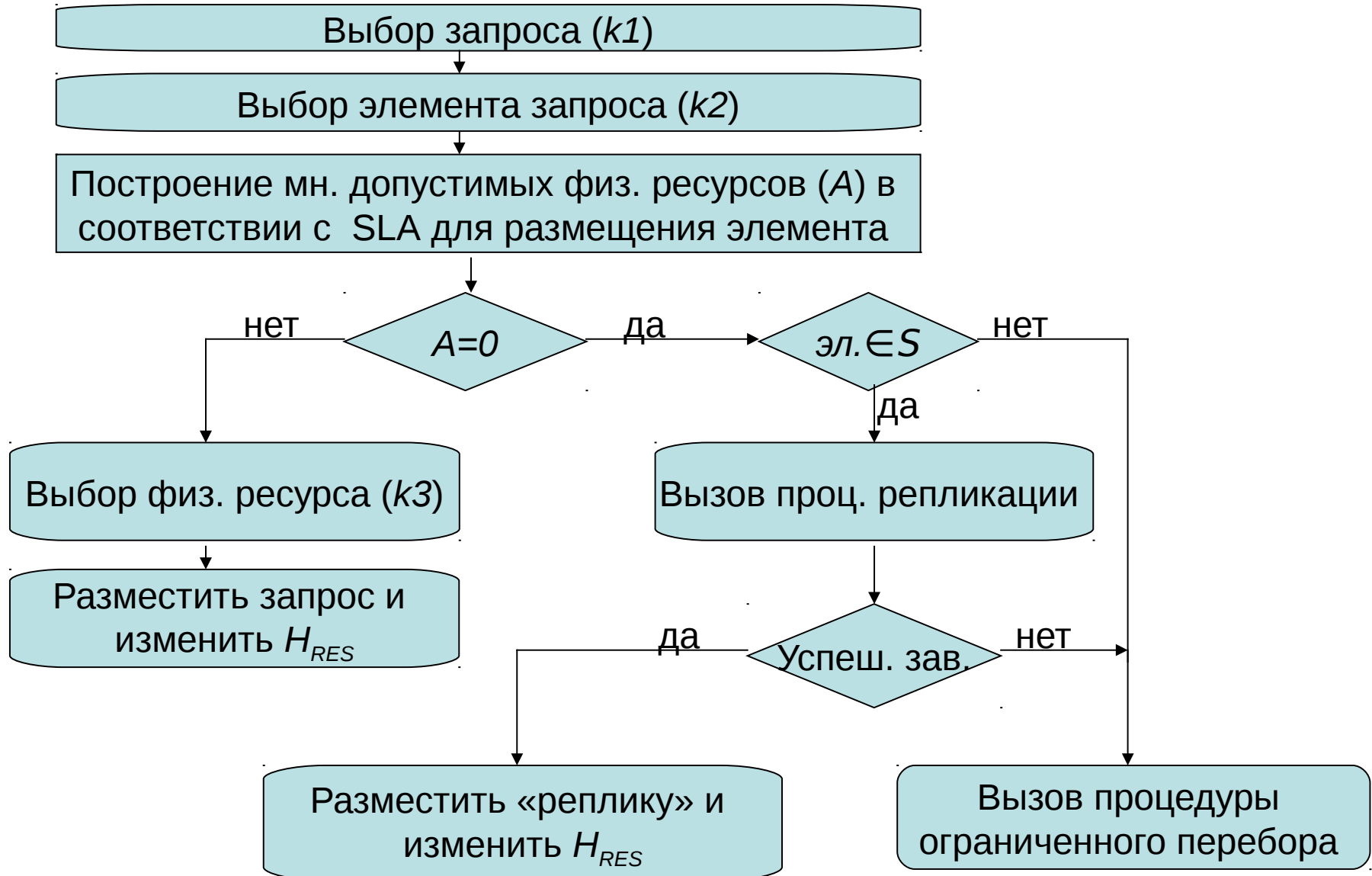
Дано:

- Множество поступивших запросов $Z = \{G_{ij}\}$.
- Множество выполняемых запросов для которых допустима миграция $M = \{G_{ij}\}$ и их отображение $A_M: M \rightarrow H$.
- Граф остаточных ресурсов ЦОД: H_{res} .

Требуется:

- определить максимальное число запросов из Z , которые можно разместить не нарушая SLA (множество $L \in Z$), и построить отображения:
 - $A_L: L \in Z \rightarrow H_{res} \cup H_M$
 - $A_M^*: M \rightarrow H_{res} \cup H_M$

Алгоритмы сочетающие жадные стратегии и стратегии ограниченного перебора

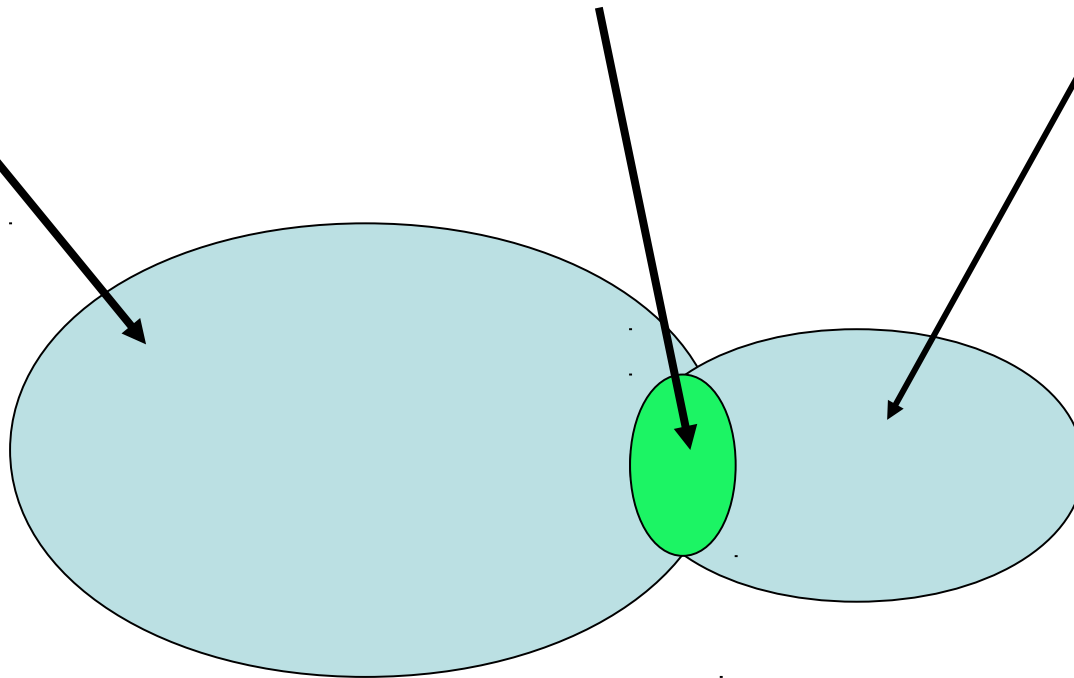


Построение множества допустимых физических ресурсов

Мн. допустимых
физ. ресурсов
(***r1*** – число ядер VM)

Мн. допустимых
мест размещения

Мн. допустимых
физ. ресурсов
(***r2*** - объем памяти VM)



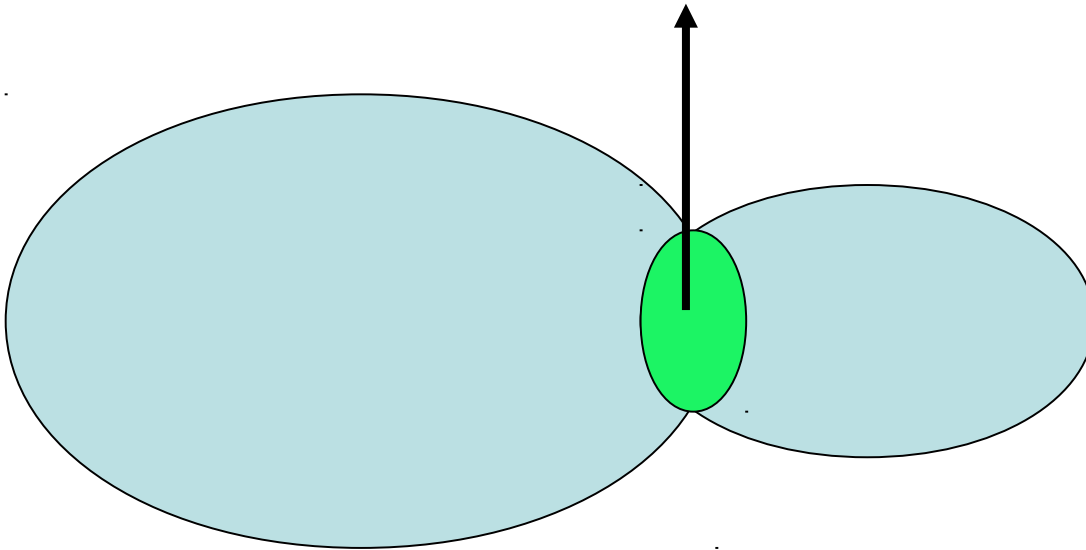
Жадный выбор места размещения

Характеристики _ ресурса :

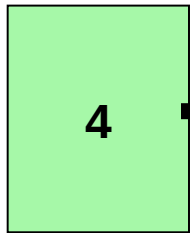
$(r_{e,1}, r_{e,2}, \dots, r_{e,n})$

Жадный выбор места размещения элемента:

- *выделение критической характеристики,*
- *взвешенная сумма с учетом дефицита ресурса по различным его характеристикам.*



Ограниченный перебор

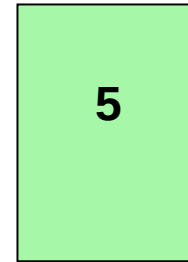


$n=2 (R1, R2)$

$R1: 5, 5$

$R2: 3, 3, 4$

$R3: 4, 4, 1$



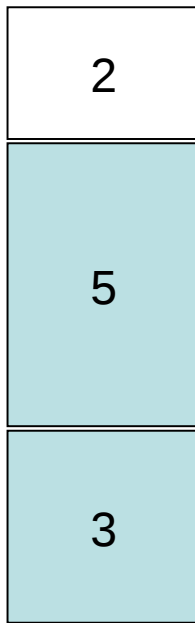
$n=3 (R1, R2, R3)$

$R1: 5, 5$

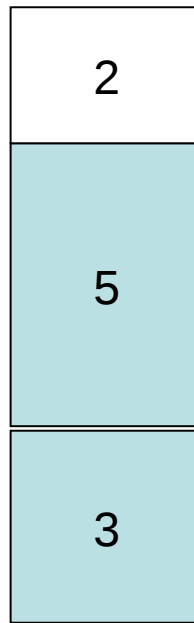
$R2: 3, 3, 4$

$R3: 4, 1, 5$

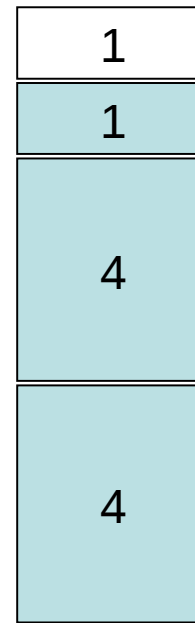
R1



R2



R3



1. В. А. Костенко. Алгоритмы комбинаторной оптимизации, сочетающие жадные стратегии и ограниченный перебор // Известия РАН. Теория и системы управления, 2017, № 2, С. 48–56. DOI: 10.7868/S0002338817020135. (*V. A. Kostenko Combinatorial Optimization Algorithms Combining Greedy Strategies with A Limited Search Procedure // Journal of Computer and Systems Sciences International, 2017, Vol. 56, No. 2, pp. 218–226. DOI: 10.1134/S1064230717020137*).
2. Зотов И. А., Костенко В. А. Алгоритм распределения ресурсов в центрах обработки данных с единым планировщиком для различных типов ресурсов // Известия РАН. Теория и системы управления, 2015., № 1, С. 61-71. (*I. A. Zotov, V. A. Kostenko. Resource Allocation Algorithm in Data Centers with a Unified Scheduler for Different Types of Resources // Journal of Computer and Systems Sciences International, 2015, Vol. 54, No. 1, pp. 59–68. DOI: 10.1134/S1064230715010141*).
3. Вдовин П.М., Костенко В.А. Алгоритм распределения ресурсов в центрах обработки данных с отдельными планировщиками для различных типов ресурсов // Известия РАН. Теория и системы управления, 2014., № 6, С. 80-93. (*Vdovin P.M., Kostenko V.A. Algorithm for Resource Allocation in Data Centers with Independent Schedulers for Different Types of Resources // J. of Computer and Systems Sciences Intern. 2014. V. 53. № 6. pp. 854–866. DOI: 10.1134/S1064230714050141*).
4. Вдовин П.М., Зотов И.А, Костенко В.А., Плакунов А.В., Смелянский Р.Л. Сравнение различных подходов к распределению ресурсов в центрах обработки данных // Известия РАН. Теория и системы управления, 2014, № 5, С.71-83. (*Vdovin P.M., Zotov I.A., Kostenko V.A., Plakunov A.V., Smelyansky R.L. Comparing Various Approaches to Resource Allocating in Data Centers // J. of Computer and Systems Sciences Intern. 2014. V. 53. № 5, pp. 689-701. DOI: 10.1134/S1064230714040145*).

■

■

■

■