

ФАКУЛЬТЕТСКИЙ КУРС: “Проектирование систем”

**Факультет радиотехники и кибернетики, Московский физико-технический институт
(государственный университет) (2004-2008)**

**Марк Шмуилович ЛЕВИН
Институт проблем передачи информации РАН**

Email: mslevin@acm.org / mslevin@iitp.ru

1. Лекции (по 2 часа): 16 занятий

2. Семинары / лабораторные работы (по 4 часа): 16 занятий

*(типовая задача, модель, схема решения, алгоритм, программа, численный пример,
отчет – статья/выступление на конференции)*

3. Домашняя работа

4. Экзамен

[Http://www.mslevin.iitp.ru/SYSD.HTM](http://www.mslevin.iitp.ru/SYSD.HTM)

ЛЕКЦИЯ 1. КУРС: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.1. Системы, структура, жизненный цикл, примеры

ПЛАН:

1. Профиль специалиста
2. О курсе
3. Иллюстративный пример системы и жизненного цикла
4. Роль математики (модели, алгоритмы)
5. Жизненный цикл и логистическая кривая
6. Инженерный опыт в России
7. Уровни системной сложности
8. Простые примеры систем
9. Системы мониторинга

Сент. 3, 2004

1. Профиль специалиста

СТРУКТУРА:

А. Базовые научные дисциплины

1. Математика

2. Физика, физико-химические процессы и др.

В. Специальные инженерные дисциплины

1. Радиотехника и др.

С. Информационная технология

Д. Управление / экономика



Е. Системное мышление



Ф. Творческие способности



Г. Опыт в прикладных областях

2.0 курсе

A. Системы, много-дисциплинарные системы

(самолеты, станки, радары, бригады, планы, производственные системы и др.)

B. Жизненный цикл (технология жизненного цикла)

C. Схемы проектирования, поддержка жизненного цикла

D. Структура курса:

(1) лекции (схемы, модели / методы,

технологические проблемы, прикладные примеры)

(2) упражнения (простые предварительные самостоятельные работы)

(3) проекты (реалистичные прикладные системы)

E. Близкие курсы :

* системный инжиниринг (system engineering)

* системное проектирование

* управление технологиями (technology management)

* многокритериальное принятие решений (multicriteria decision making)

* комбинаторная оптимизация

* инженерия знаний

* приложения (техника, управление, информационная технология)

3.Иллюстративный пример жизненного цикла

А.Жизненный цикл:

- *предварительное исследование
- *проектирование
- *производство
- *тестирование
- *исследование рынка и реализация рыночных стратегий
- *использование и техобслуживание
- *утилизация

В.Система (самолет):

- *корпус
- *двигатель
- *электроника (управление, коммуникации и др.)
- *среда для человека (пилот, пассажиры)

Дополнительные подсистемы поддержки:

- *техобслуживание
- *обучение и тренировка персонала
- *утилизация подсистем
- *др.

4.Роль математики

А.Модели

- *структурные модели (графы, сети)
- *оптимизационные модели
- *модели многокритериального принятия решений
- *дифференциальные уравнения (динамика)
- *теория игр
- *модели неопределенности (вероятность, размытые множества и др.)

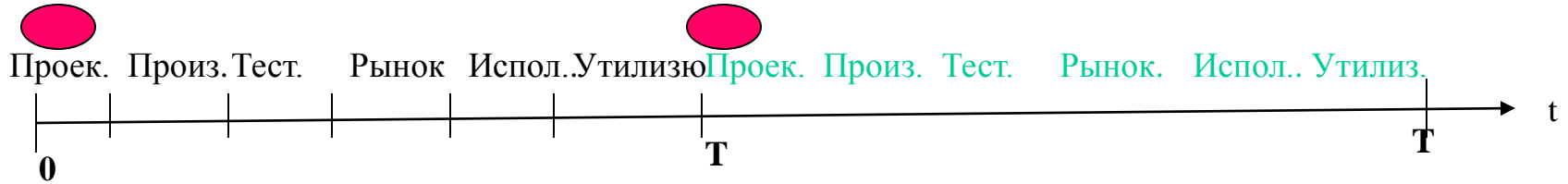
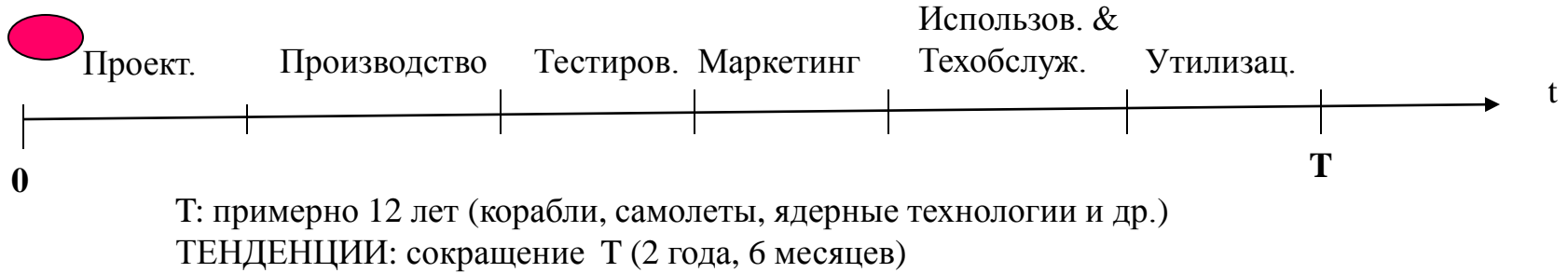
В.Алгоритмы

С.Схемы решения

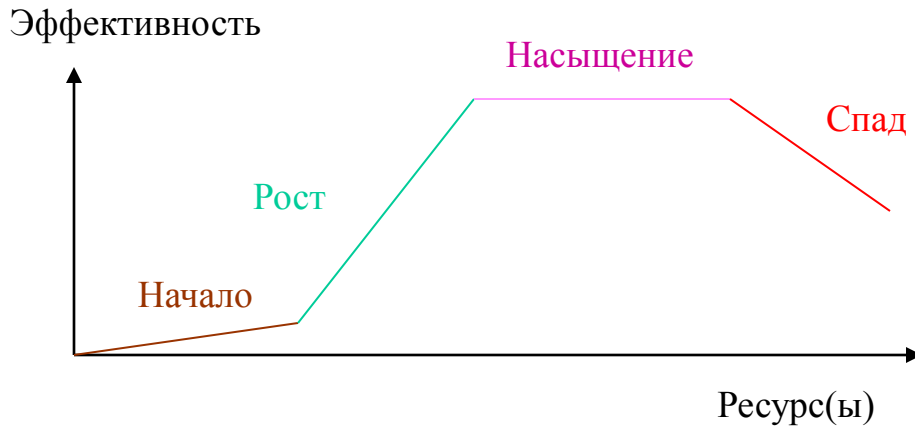
Реальное Новое Приложение =>

Новые или модифицированные модели / алгоритмы

5. Жизненный цикл и логистическая кривая



РЕЗУЛЬТАТ: потребность в специалистах в системном проектировании & специалистах в жизненном цикле



Сложные системы:

- 1.Самолеты**
- 2.Космические системы**
- 3.Коммуникационные системы**
- 4.Ядерные технологии**
- 5.Оборонные системы (радары и др.)**
- 6.Т.Д.**

Факторы :

- 1.Творческие люди**
- 2.Образовательная система**
- 3.Инженерные традиции (в проектировании сложных систем)**
- 4.Сложные проблемы**
(огромная территория, различные среды и др.)

7. Уровни сложности систем (A. Shenhar)

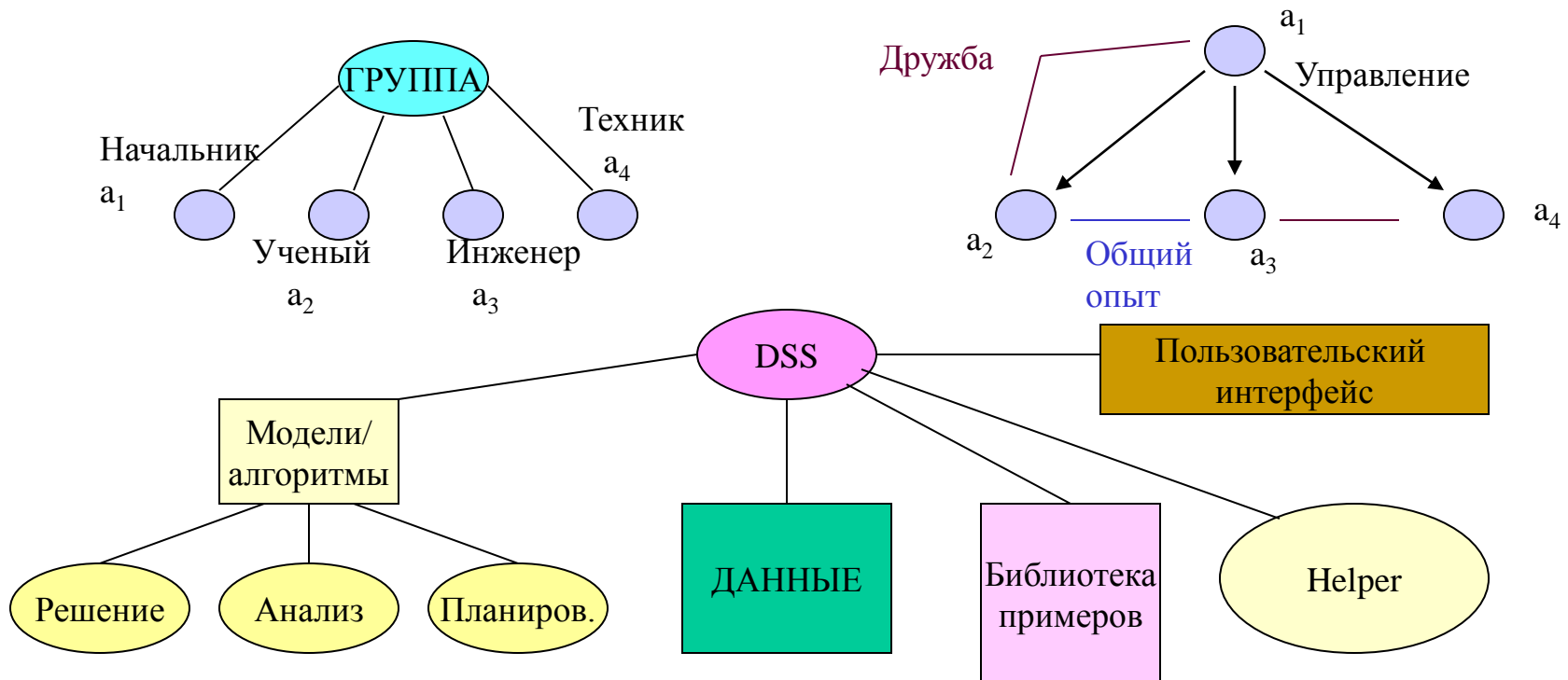
Уровень 1. Agrays (сеть систем, например, сеть радарных оборонных систем)

Уровень 2. Система (много функций; радар, оборонная система)

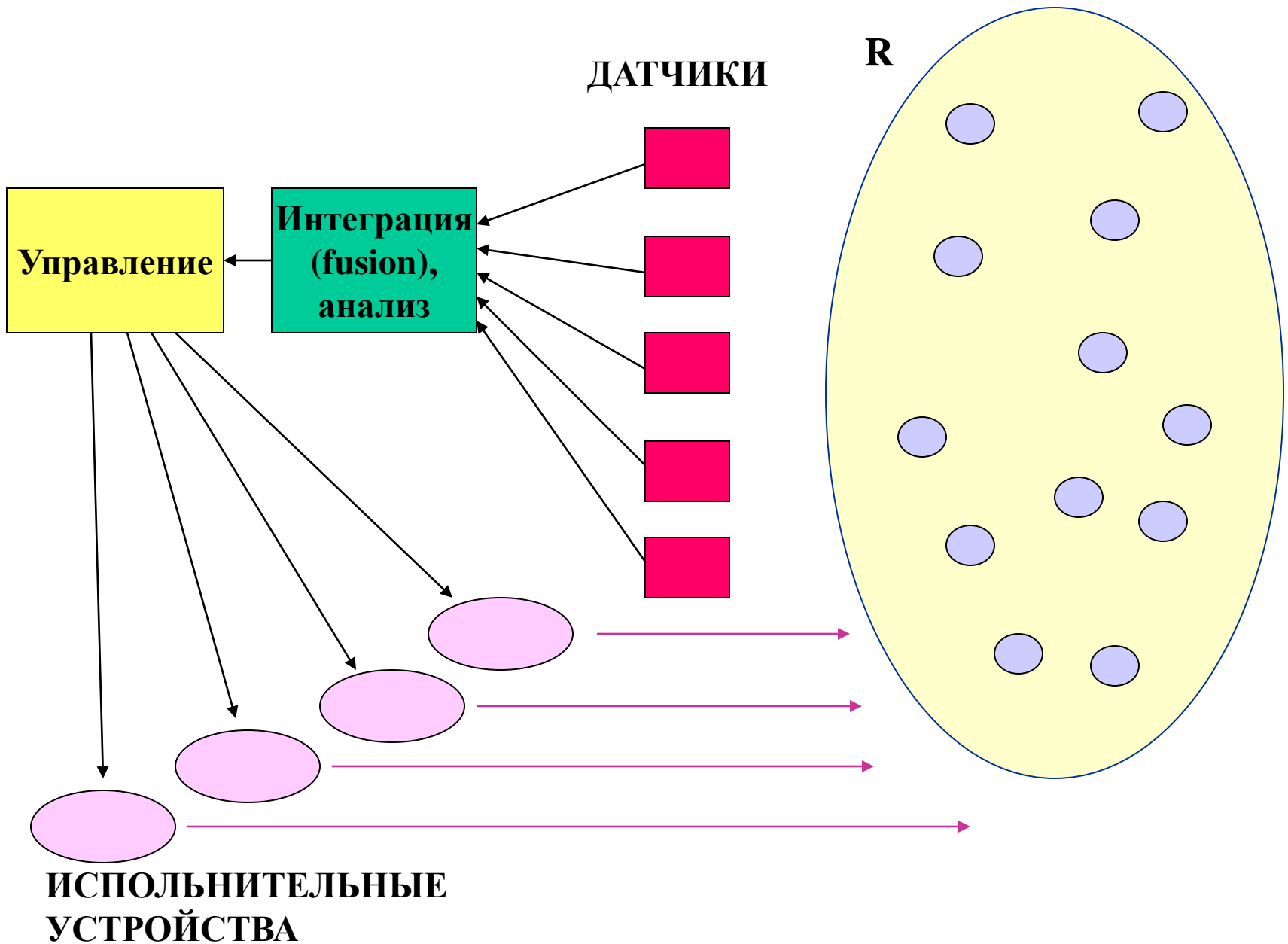
Уровень 3. Assembly (составная система) (1 функция: TV)

Уровень 4. Компонент

8. Простые примеры систем



9. Системы мониторинга



ЛЕКЦИИ 2-3. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.2. Модульность, декомпозиция (пример)

Л.3. Структурные модели (графы, сети, бинарные отношения), примеры

ПЛАН:

1. Декомпозиция (разбиение) систем: *декомпозиция – разбиение; *иллюстративные примеры; *подходы
2. Вопросы модульности: *описание и базовый лингвистический аналог,
*прикладные примеры (механика, космос и др.), *цели и результаты
3. Структурные модели: *графы (графы, ориентированные графы - оргграфы, знаковые графы)
*простые структуры (цепи, деревья, параллельно-последовательные графы и др.)
*задачи на графах (метрика/близость, оптимизация, перспективные модели)

Сент. 4, 2004

1. Декомпозиция / разбиение систем

Декомпозиция: последовательный процесс (например, динамическое программирование)

Разбиение: параллельный процесс / разбиение
(пример: комбинаторный синтез)

Методы разбиения:

*физическое разбиение

*функциональное разбиение

Примеры (для самолета, для человека)

Примеры для программных систем:

1. Последовательный процесс обработки информации
(вход, процесс решения, анализ, выход)

2. Архитектура:

Подсистема поддержки данных, процесс решения, пользовательский интерфейс, подсистема обучения, коммуникации

3. Дополнительная часть: visualization (для данных, для процесса решения)

4. Дополнительная вспомогательная часть для управления моделями (model management) :

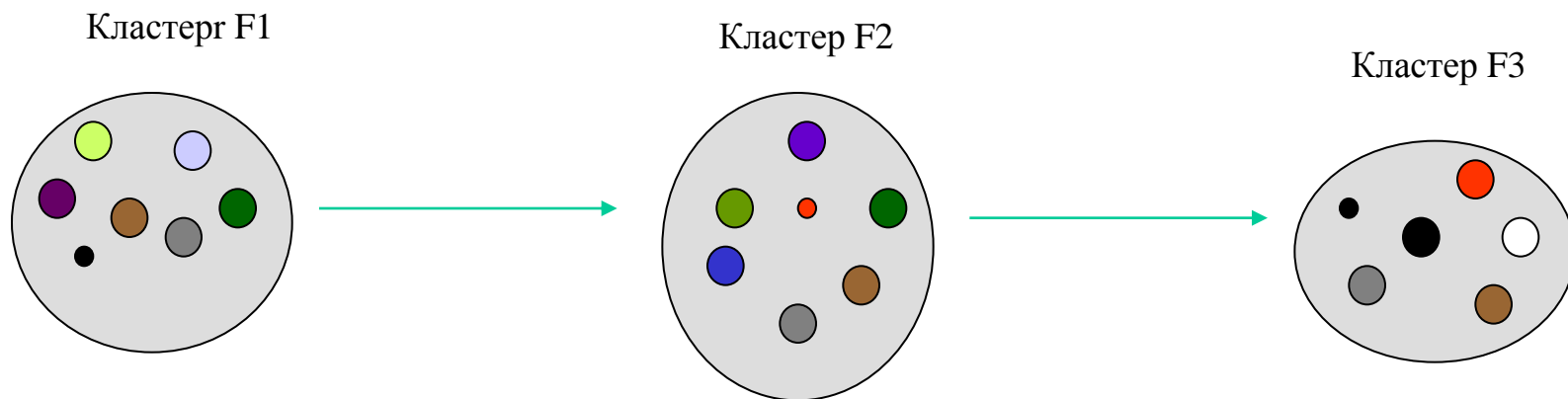
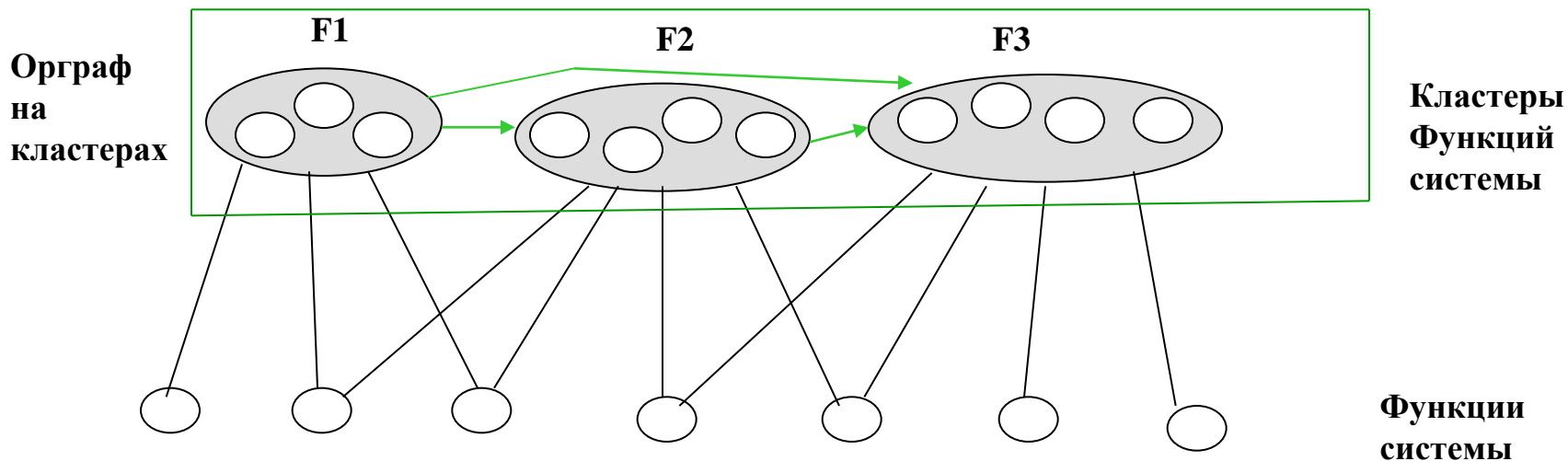
*анализ исходной прикладной ситуации,

*библиотека моделей / методов,

*выбор / проектирование моделей / методов,

*выбор / проектирование много-модельных стратегий решения

1. Декомпозиция / разбиение систем: Пример для много-функционального тестирования систем



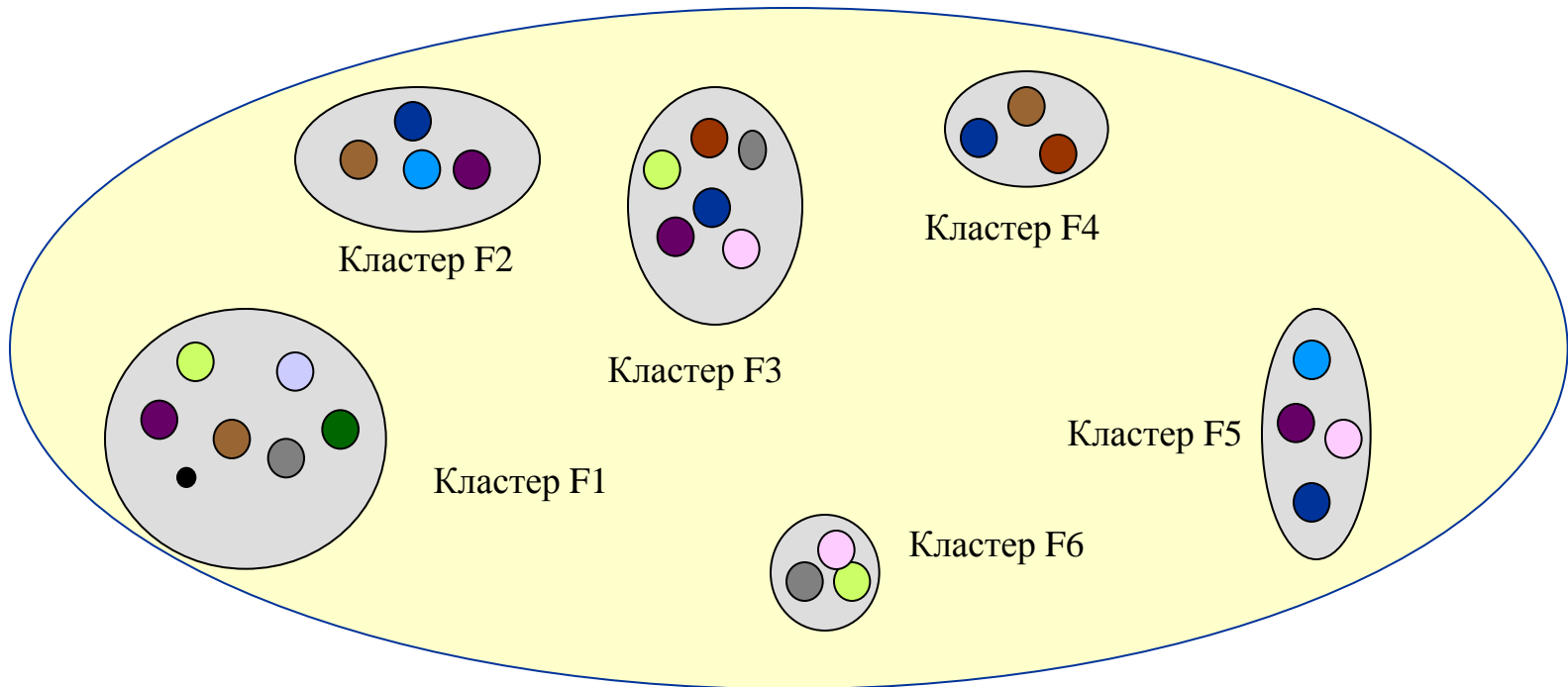
Главные подходы к разбиению систем

А.Содержательный анализ и опыт:

*по функциям (основные функции, дополнительные функции)

*по системным частям (физическое разбиение)

В.Кластер-анализ (кластеризация)



2. Вопросы модульности

ПРИНЦИПЫ УПРАВЛЕНИЯ СЛОЖНОСТЬЮ :

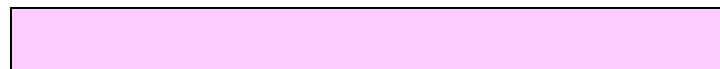
*дискретные части (модули)

*стандартные интерфейсы для коммуникации модулей

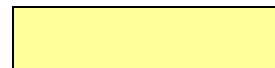
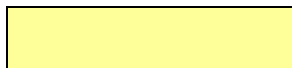
Приложения: *проектирование новых технологий * организационное проектирование

ЛИНГВИСТИЧЕСКАЯ СИСТЕМА

ТЕКСТЫ



ФРАЗЫ



СЛОВА



АЛФАВИТ



Прикладные примеры использования модульности

1.Генетика

2.Реконфигурируемое (Reconfigurable) производство

3.Программные библиотеки стандартных модулей

4.Комбинаторная химия:

*молекулярное проектирование в химии и биологии

*проектирование лекарств

*технология проектирования материалов

*др.

5.Проектирование в механических / космических систем

6.Электроника

7.Строительство

Основные цели модульности и резюме

ОСНОВНЫЕ ЦЕЛИ:

- 1. Управление сложностью**
- 2. Параллельная работа**
- 3. Подготовка к будущей неопределенности**
- 4. Разнообразие результирующих модульных систем**
- 5. Гибкость, адаптивность, способность к изменению конфигурации результирующих модульных систем**

РЕЗЮМЕ:

- 1. Упрощение процесса проектирования & упрощение всех фаз жизненного цикла**
- 2. Короткий жизненный цикл продукции, долгий жизненный цикл модулей**
- 3. Способность к изменению конфигураций систем (Reconfigurable systems)**
(например, для производственных систем):
продолжительный жизненный цикл поколений системы
- 4. Упрощенное проектирование и поддержка семейств продуктов**
(самолеты, автомобили и др.)
- 5. Упрощенное проектирование и поддержка различных продуктов**
(на основе библиотек модулей как повторное использование)

3. Структурные модели

А. ГРАФЫ

1. Графы

2. Оргграфы (направленные графы или ориентированные графы)

3. Графы / оргграфы с весами (для вершин, для ребер / дуг)

4. Простые графы: цепи, деревья, параллельно-последовательные графы, иерархии

5. Знаковые графы

Б. СЕТИ

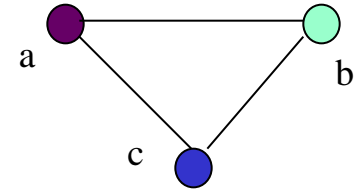
В. АВТОМАТЫ

Г. БИНАРНЫЕ ОТНОШЕНИЯ

Иллюстрация для графов / орграфов

Граф: $G = (A, E)$ где множество узлов (вершин) $A = \{1, \dots, n\}$
и множество ребер $E \subseteq A \times A$ (пары вершин)

Пример: $A = \{a, b, c\}$, $E = \{(a, b), (b, c), (a, c)\}$

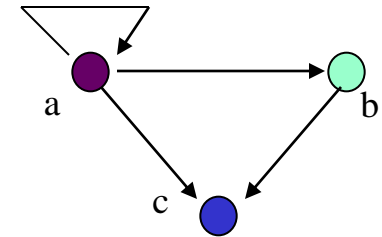


Матрица

	a	b	c
a		1	1
b	1		1
c	1	1	

Орграф: $G = (A, E)$ где множество узлов (вершин) $A = \{1, \dots, n\}$
и множество дуг $E \subseteq A \times A$ (пары вершин)

Пример: $A = \{a, b, c\}$, $E = \{(a, a), (a, b), (b, c), (a, c)\}$

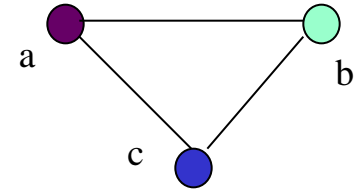


Матрица

	a	b	c
a	1	1	1
b			1
c			

Иллюстрация для графов с весами

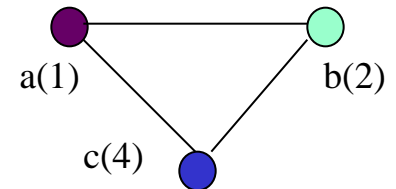
Граф (веса ребер): $G = (A, E)$ где множество узлов (вершин)
 $A = \{1, \dots, n\}$ и множество ребер $E \subseteq A \times A$ (пары вершин)
Пример: $A = \{a, b, c\}$, $E = \{(a, b), (b, c), (a, c)\}$



Матрица

	a	b	c
a		2	5
b	2		3
c	5	3	

Граф (веса ребер & вершин): $G = (A, E)$ где множество ребер
(узлов) $A = \{1, \dots, n\}$ и множество ребер $E \subseteq A \times A$ (пары вершин)
Пример: $A = \{a, b, c\}$, $E = \{(a, b), (b, c), (a, c)\}$
(веса вершин указаны в скобках)

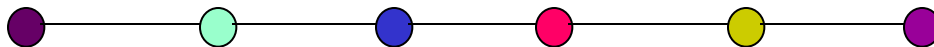


Матрица

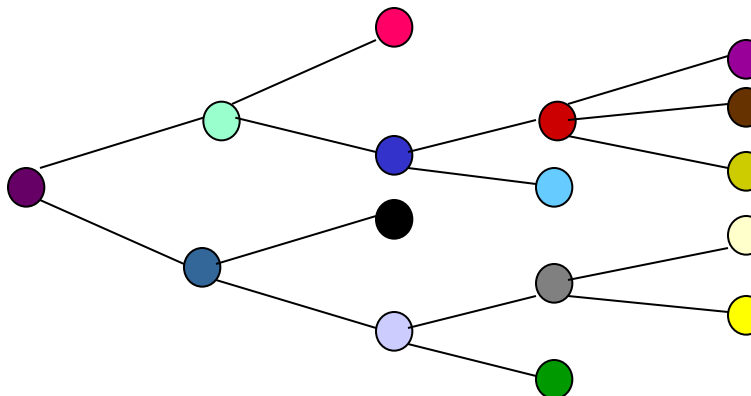
	a	b	c
a		2	5
b	2		3
c	5	3	

Простые структуры (цепи, деревья, параллельно-последовательные графы)

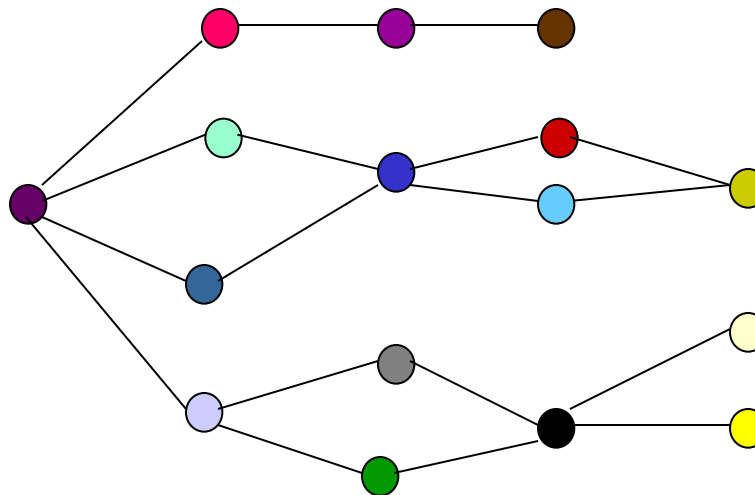
ЦЕПЬ



ДЕРЕВО



ПАРАЛЛЕЛЬНО-ПОСЛЕДОВАТЕЛЬНЫЙ ГРАФ



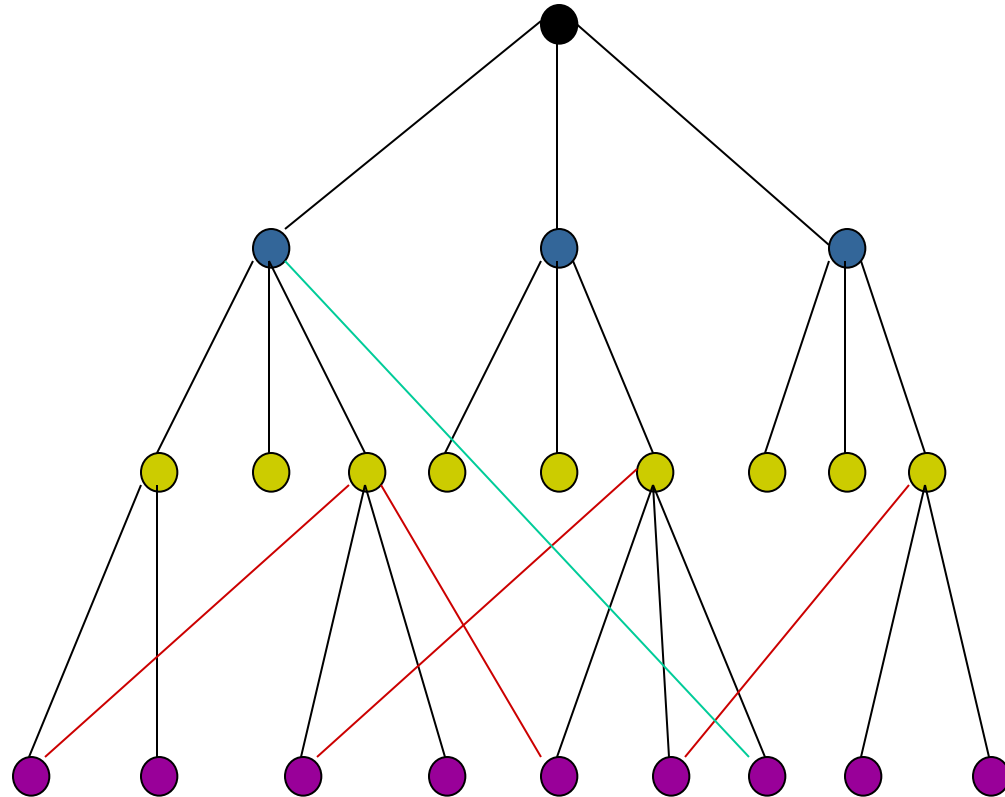
Простые структуры (иерархия)

Уровень 4

Уровень 3

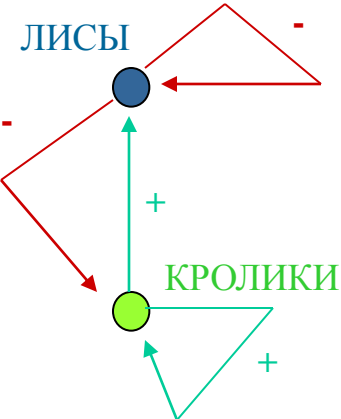
Уровень 2

Уровень 1



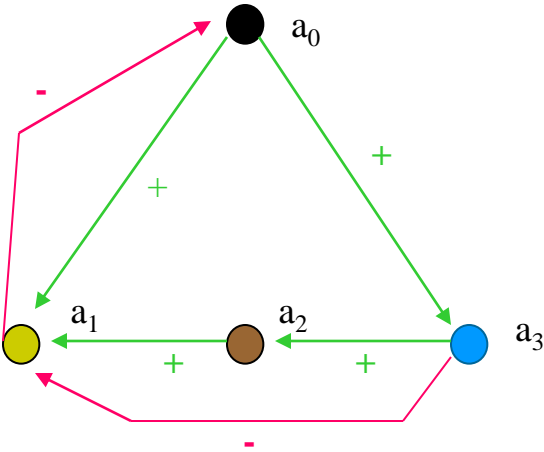
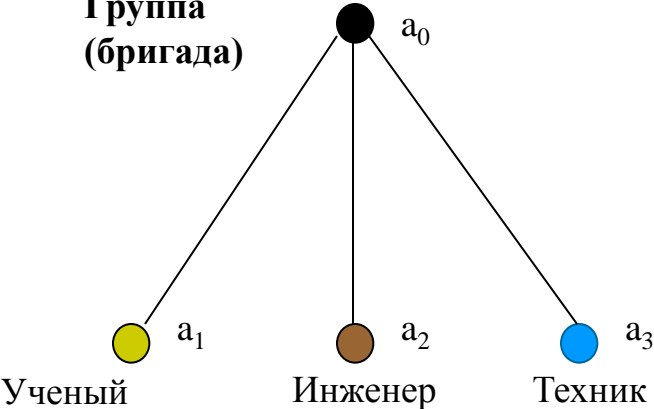
Знаковый граф: иллюстративный пример

Экологическая система



Руководитель

Группа (бригада)



1.Мультиграфы

2.Графы с версиями вершин (узлов)

3.Графы с весами в виде векторов

4.Графы с «размытыми» весами

А.Метрика / близость (в графе между вершинами, между графам)

Близость между графами:

- 1.метрики,**
- 2.расстояние редактирования (минимальная “цена” трансформации),**
- 3.общая часть**

Б.Оптимизация на графах:

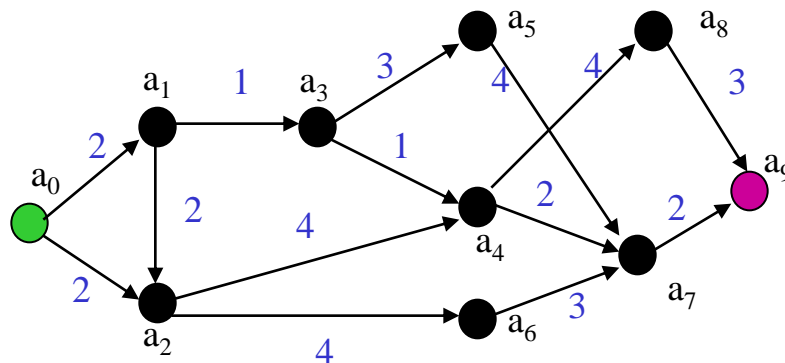
- 1.Кратчайший путь**
- 2.Покрывающее дерево (& близкие аппроксимационные задачи:
Покрытие другими простыми структурами)**
- 3.Задача о коммивояжере**
- 4.Минимальное дерево Штейнера**
- 5.Упорядочение вершин**
- 6.Размещение на графах**
- 7.Задачи о покрытиях**

В.Задачи балансов для знаковых графов

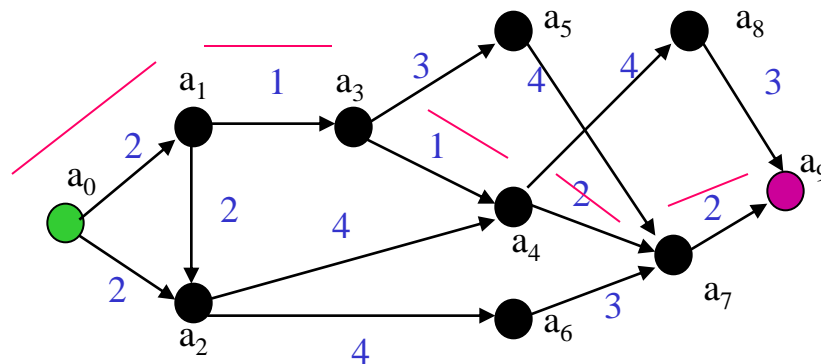
Г.Кластеризация (разбиение на группы взаимосвязанных / близких элементов)

Оптимизация на графах: иллюстрации

БАЗОВЫЙ ГРАФ (ОРГРАФ):
Веса для дуг (или ребер)

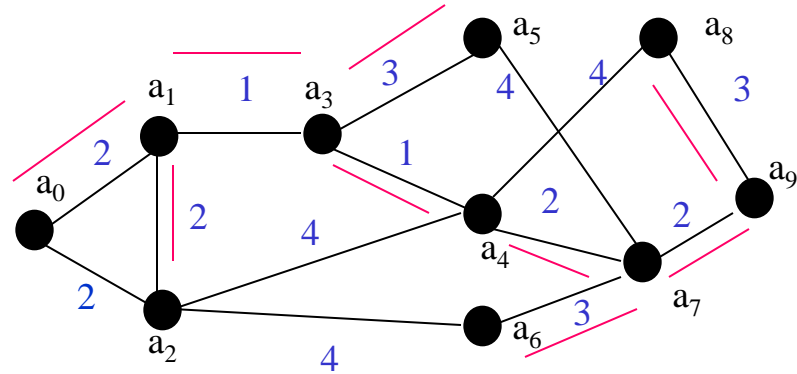
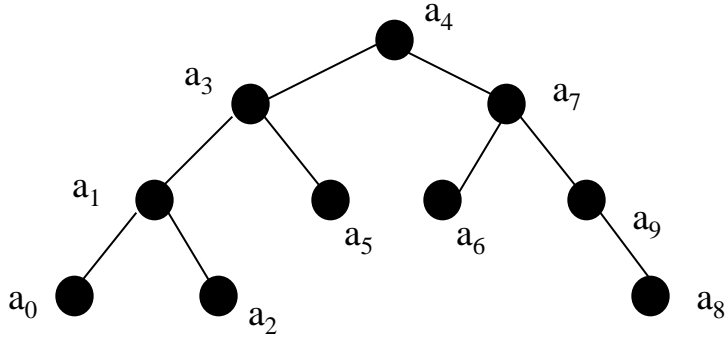


Кратчайший путь $\langle a_0, a_9 \rangle$:
 $L = \langle a_0, a_1, a_2, a_3, a_4, a_7, a_9 \rangle$
 $2+1+1+2+2 = 8$



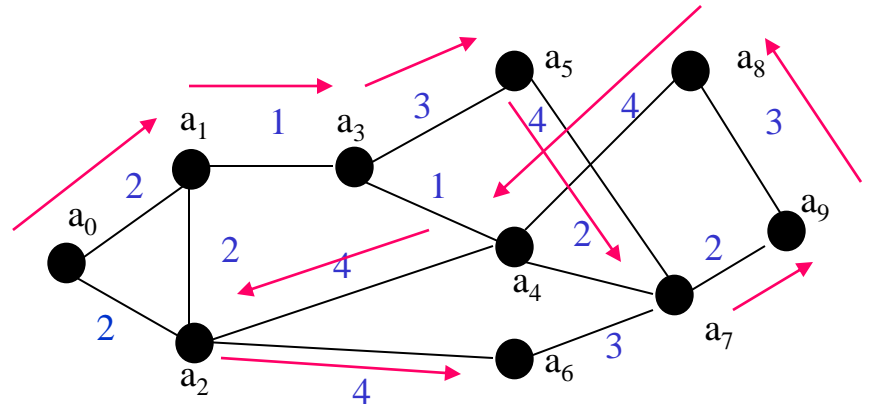
Оптимизационные задачи на графах: иллюстрации

Покрывающее дерево (длина = 19):



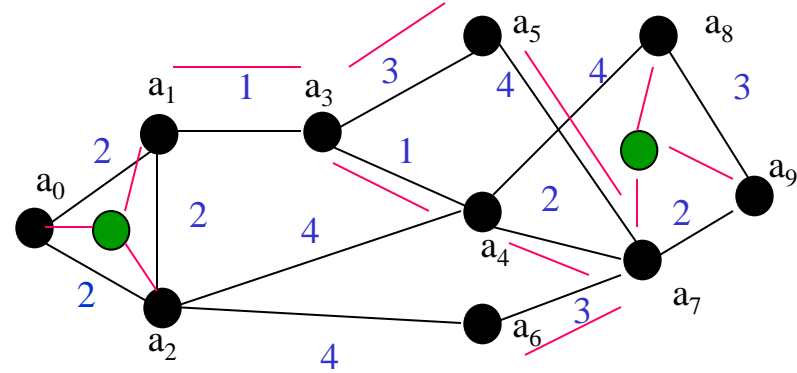
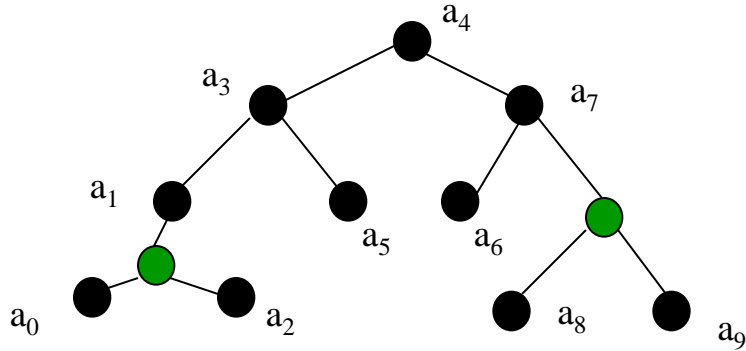
Задача коммивояжера :

$L = \langle a_0, a_1, a_3, a_5, a_7, a_9, a_8, a_4, a_2, a_6 \rangle$
 $2+1+3+4+2+2+3+4+4+4$

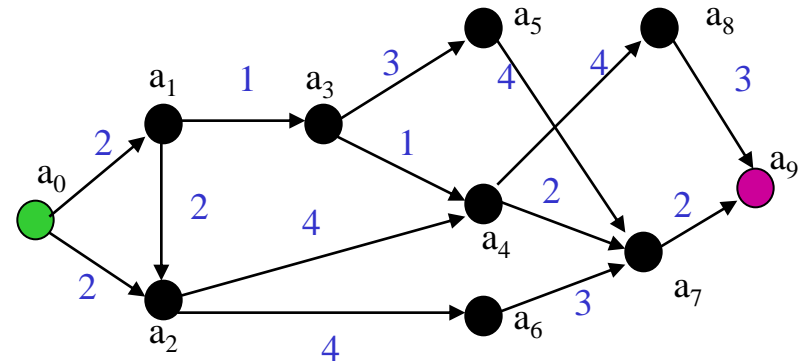
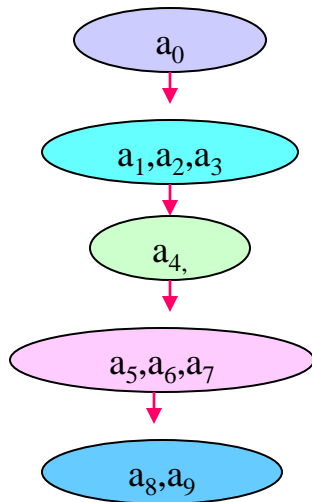


Оптимизационные задачи на графах: иллюстрации

Дерево Штейнера (пример):



Задача “Упорядочения” (близкие задачи: задача календарного планирования):



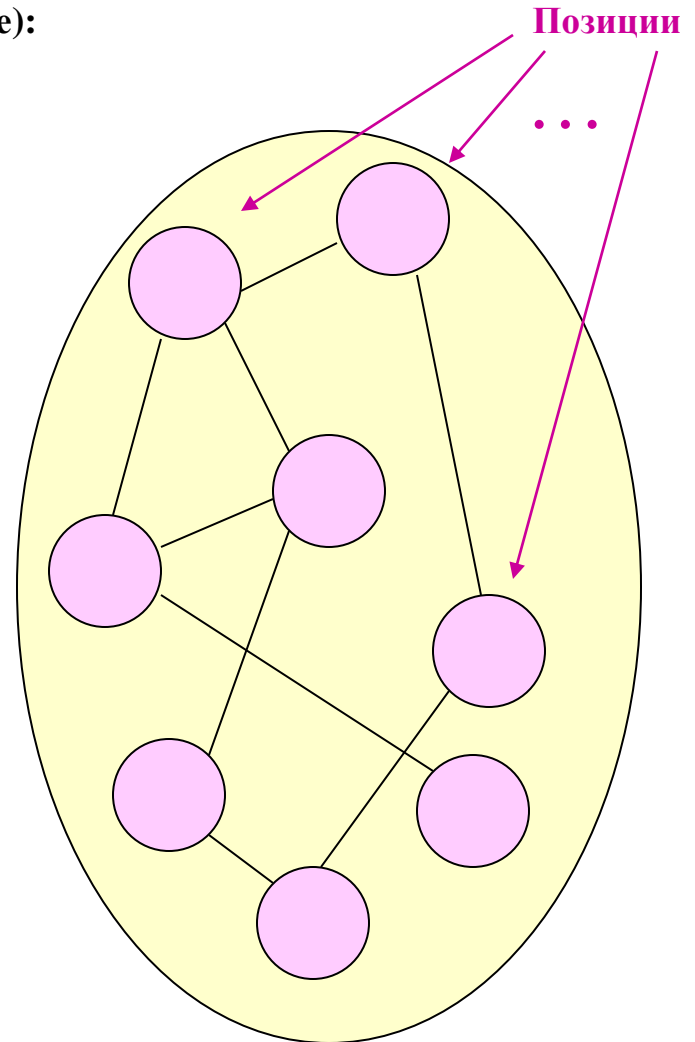
Оптимизационные задачи на графах: иллюстрации

Размещение (назначение, сопоставление, отображение):

Множество
элементов



РАЗМЕЩЕНИЕ
(отображение,
назначение)



Пример: кластеры функций системы и их покрытие цепями (покрытие дуг)

Оргграф кластеров функций системы

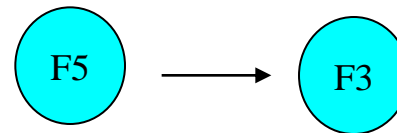
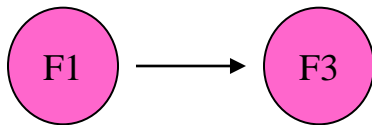
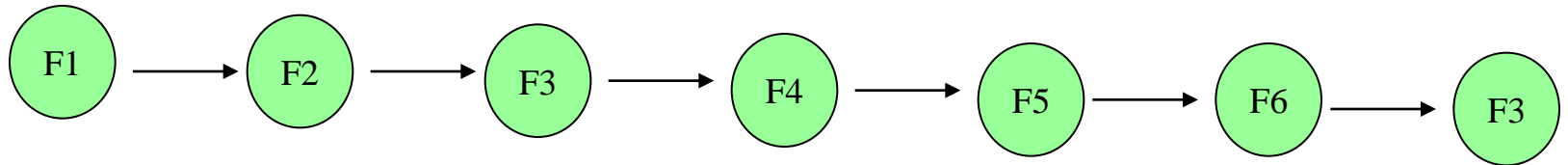
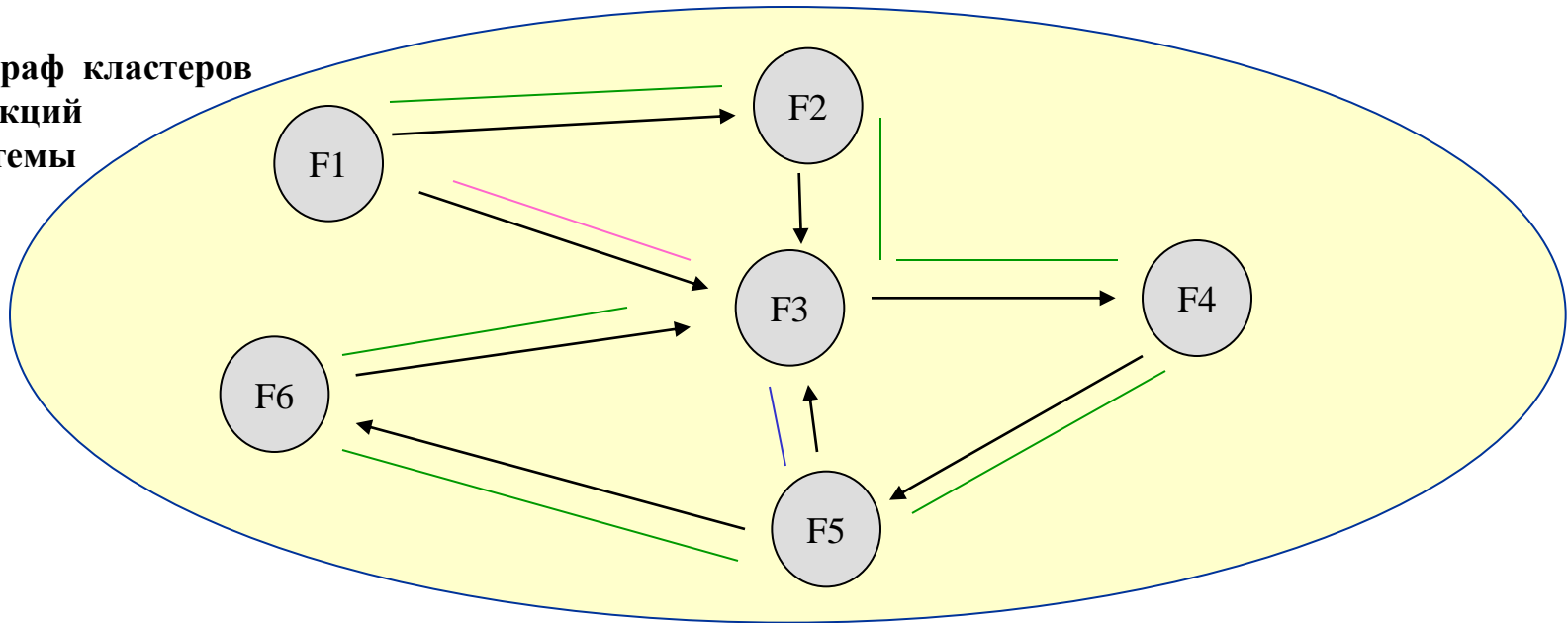
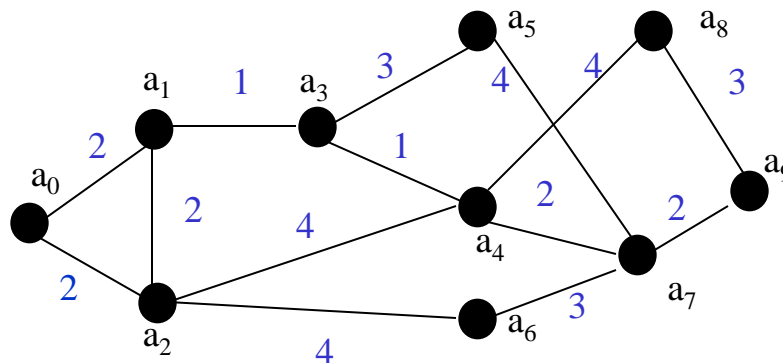


Иллюстрация кластеризации

Базовый граф



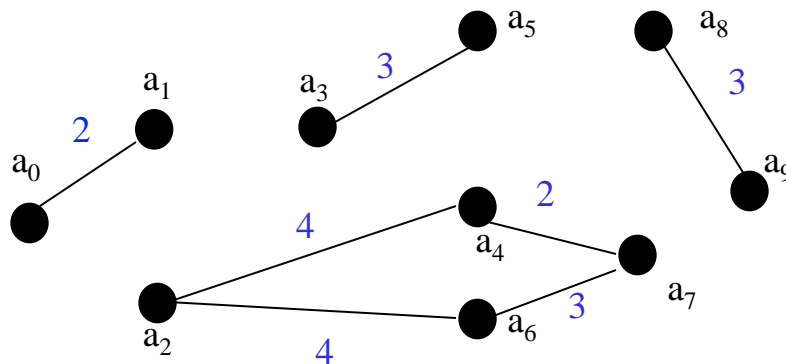
Кластеры (вариант решения):

$$C_1 = \{ a_0, a_1 \}$$

$$C_2 = \{ a_3, a_5 \}$$

$$C_3 = \{ a_8, a_9 \}$$

$$C_4 = \{ a_2, a_4, a_6, a_7 \}$$



Бинарные отношения

Исходное множество $A = \{1, 2, \dots, n\}$, $B = A \times A$ ($\forall (x, y), x, y \in A$)

Определение. Бинарное отношение R это подмножество B

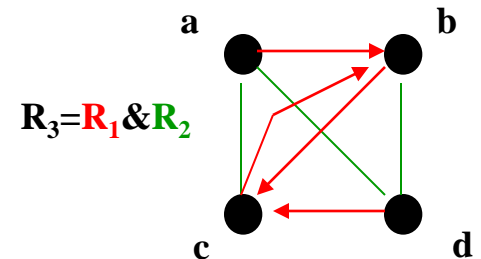
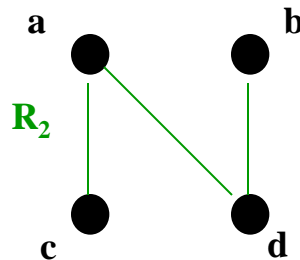
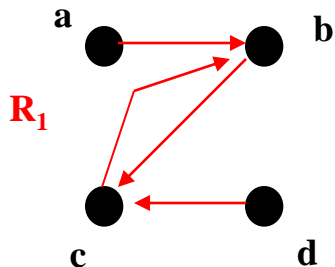
ПРИМЕР: $A = \{a, b, c, d\}$

$B = \{(a, a), (a, b), (a, c), (a, d), (b, a), (b, b), (b, c), (b, d), (c, a), (c, b), (c, c), (c, d), (d, a), (d, b), (d, c), (d, d)\}$

$R_1 = \{(a, b), (b, c), (c, b), (d, c)\}$

$R_2 = \{(a, d), (b, d), (a, c)\}$

$R_3 = R_1 \& R_2$



Бинарные отношения

Смысловой пример:

1. "Лучше" (доминирование)
2. "Лучше & Равно" (доминирование & эквивалентность)
3. "Равно" (эквивалентность)

НЕКОТОРЫЕ СВОЙСТВА:

1. Симметрия: $(x, y) \in R \Rightarrow (y, x) \in R \quad (\forall x \in R, \forall y \in R)$
2. Рефлексивность: $(x, x) \in R \quad \forall x \in R$
3. Транзитивность: $(x, y) \in R, (y, z) \in R \Rightarrow (x, z) \in R \quad (\forall x \in R, \forall y \in R, \forall z \in R)$

ПРИЛОЖЕНИЯ *Дружба, *Партнерство, *Похожесть и др.

Расширенные модели:

1. Взвешенные бинарные отношения (например, сила доминирования)
2. К-отношения

Перспективное использование : Задача комбинаторной оптимизации на графах с дополнительными бинарными отношениями (на вершинах / узлах, на ребрах / дугах, на элементах / позициях)

Лекция 4. Курс: “Проектирование систем: Структурный подход”

Каф.. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики
Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН
Институт проблем передачи информации, РАН

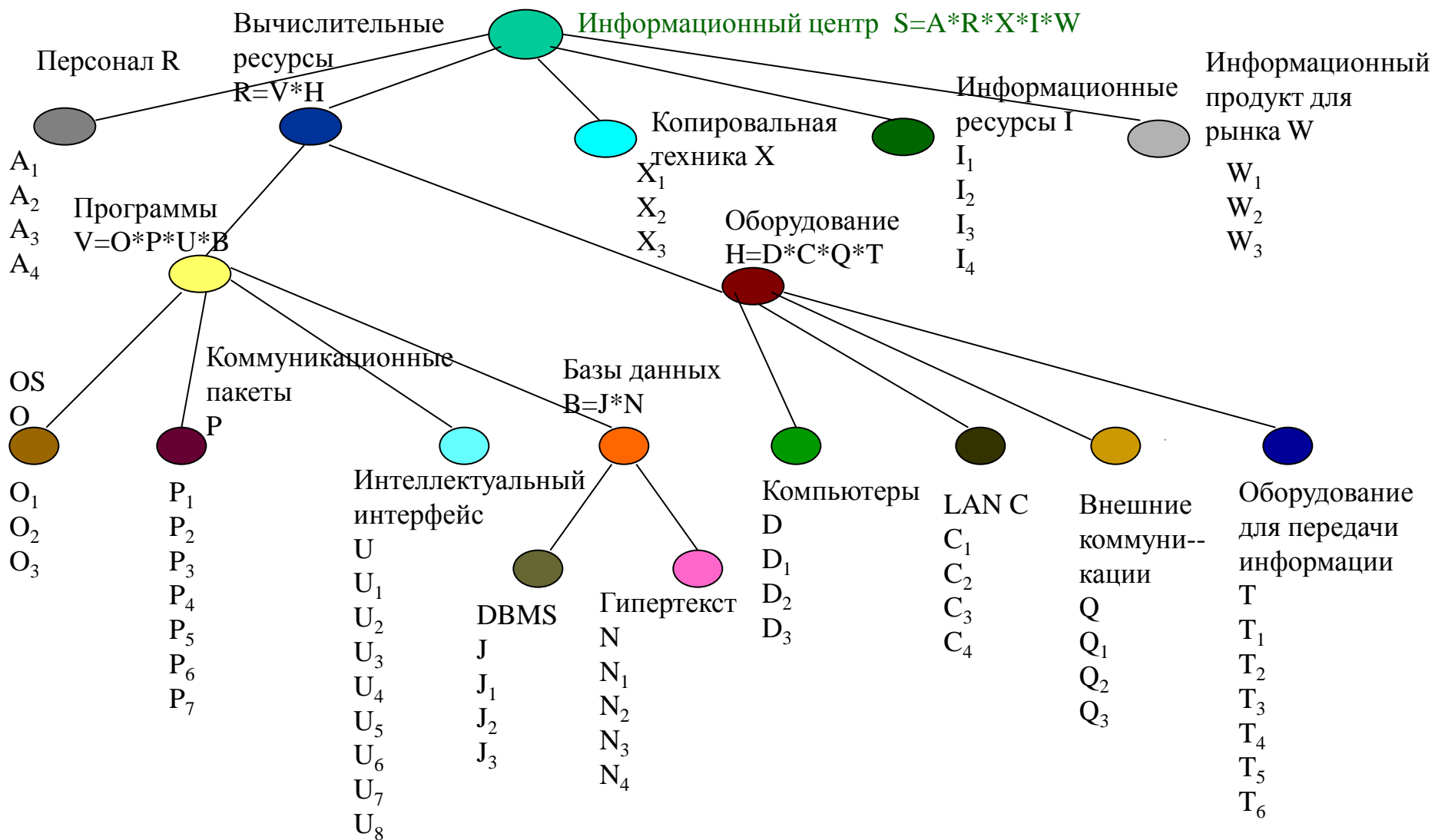
Email: mslevin@acm.org / mslevin@iitp.ru

Л.4. Пример: проектирование иерархической системы. Система связи

ПЛАН:

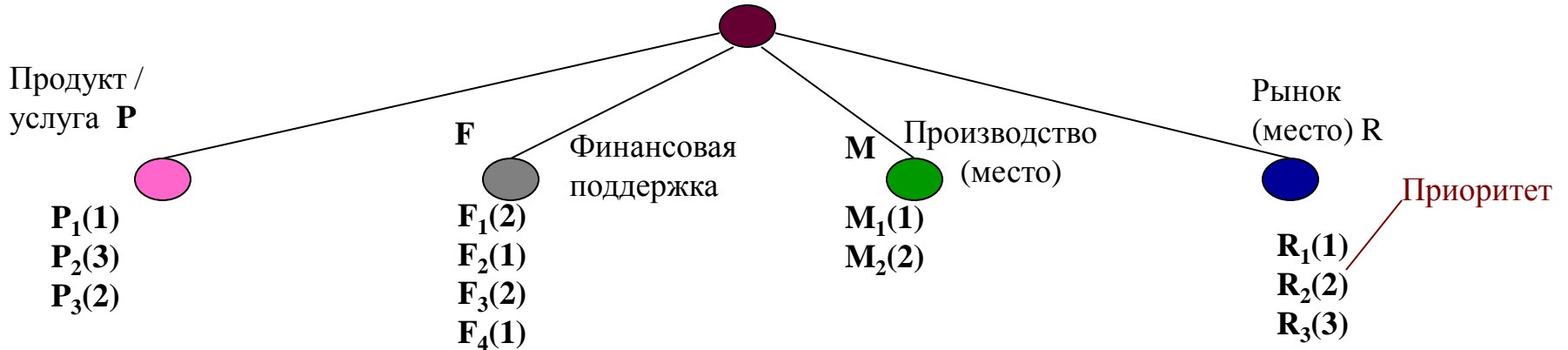
1. Совместное проектирование иерархической системы: *структурная схема, *иерархическая модель, *генерация альтернатив для системных частей, *генерация критериев для оценивания альтернатив, *многокритериальный выбор альтернатив, *синтез составных систем
2. Обсуждение перспективных исследовательских направлений в коммуникационных системах

Пример иерархической системы: информационный центр



Совместное проектирование иерархической системы: студенческий бизнес

Студенческий бизнес $S=P * F * M * R$



Альтернативы:

- P_1 простая еда (например, каша)
- P_2 услуги по поддержке персональных компьютеров
- P_3 специальное консультирование для поиска персонала (для компаний, для специалистов)
- F_1 самофинансирование
- F_2 финансирование с помощью родственников, друзей
- F_3 финансовая поддержка из банка
- F_4 финансовая поддержка из компании
- M_1 в университете
- M_2 специальное помещение в Долгопрудном
- R_1 Долгопрудный
- R_2 Москва
- R_3 Нью Йорк

Оценивание альтернатив и приоритеты

	Стоимость производства (-)	Объем рынка (+)	Перспективы(+)	Оценка	Приоритет
P ₁	1	5	3	7	1
P ₂	3	5	2	4	3
P ₃	4	4	5	5	2

	Возможный объем(+)	Ответственность(-)	Оценка	Приоритет
F ₁	1	1	0	2
F ₂	2	1	1	1
F ₃	5	5	0	2
F ₄	4	3	1	1

	Стоимость(-)	Полезность(+)	Оценка	Приоритет
M ₁	1	5	4	1
M ₂	3	5	2	2

	Объем (+)	Возможная конкуренция (-)	Расстояние (-)	Оценка	Приоритет
R ₁	1	0	0	1	1
R ₂	3	2	1	0	2
R ₃	3	3	5	-5	3

Оценивание совместимости между альтернативами

	F₁	F₂	F₃	F₄	M₁	M₂	R₁	R₂	R₃
P₁	5	4	0	0	5	3	5	2	0
P₂	4	5	1	2	5	5	4	5	3
P₃	0	1	4	3	0	5	2	5	5
F₁					5	1	5	3	0
F₂					5	1	5	3	0
F₃					3	5	1	5	5
F₄					3	5	1	5	5
M₁							5	4	5
M₂							3	5	4

Замечание: **5** соответствует лучшему уровню совместимости
0 соответствует несовместимому случаю

Результирующее составное решение

$$S_1 = P_1 * F_2 * M_1 * R_1 ; \text{ вектор качества } N(S_1) = (4; 4,0,0);$$

Описание: (a) все элементы на лучшем уровне (1); (b) уровни совместимости (5,5,5,5,5,4)

Другое перспективное решение:

$$S_2 = P_3 * F_4 * M_1 * R_2 ; \text{ вектор качества } N(S_2) = (4; 2,2,0);$$

Описание: (a) уровни элементов (1,1,2,2); (b) оценки совместимости (3,3,4,5,5,0)

Замечание: это решение является недопустимым по совместимости

Узкие места (проблемы для улучшения исходной ситуации):

- 1.совместимость (P_3, M_1) equals 0 (\Rightarrow увеличить)
- 2.совместимость (F_4, M_1) equals 3 (\Rightarrow увеличить)
- 3.совместимость (M_1, R_2) equals 3 (\Rightarrow увеличить)
- 4.приоритет P_3 (\Rightarrow увеличить)
- 5.приоритет R_2 (\Rightarrow увеличить)

Перспективные направления исследований в коммуникационных системах

- 1.Общее проектирование коммуникационных сетей**
- 2.GRID-подобная сетевая среда (GRID-вычисления, GRID-коммуникации)**
- 3.Расширение коммуникационных сетей:**
 - 3а.Улучшение существующей коммуникационной сети**
 - 3б.Расширение существующей коммуникационной сети (дополнительная территория)**
- 4.Размещение ресурсов в коммуникационных сетях (прикладные ситуации, задачи, модели, подходы)**
- 5.назначение частот в коммуникационных сетях (прикладные ситуации, задачи, модели, подходы)**
- 6.Вопросы надежности коммуникационных сетей**
 - 6а.Оценивание надежности**
 - 6б.Проектирование надежных коммуникационных сетей**
- 7.Маршрутизация**

8.Тестирование коммуникационных систем: задача зондирования

9.Техническое обслуживание в коммуникационных системах

10.Мобильные коммуникационные системы

11а.Движение пользователей

11б.Движение всех компонентов системы

11.Системы сжатия информации (алгоритмическая часть)

12.Коммуникационные сети и free scale networks

13.Планирование доступа к информационным / вычислительным ресурсам

(информационные базы на серверах)

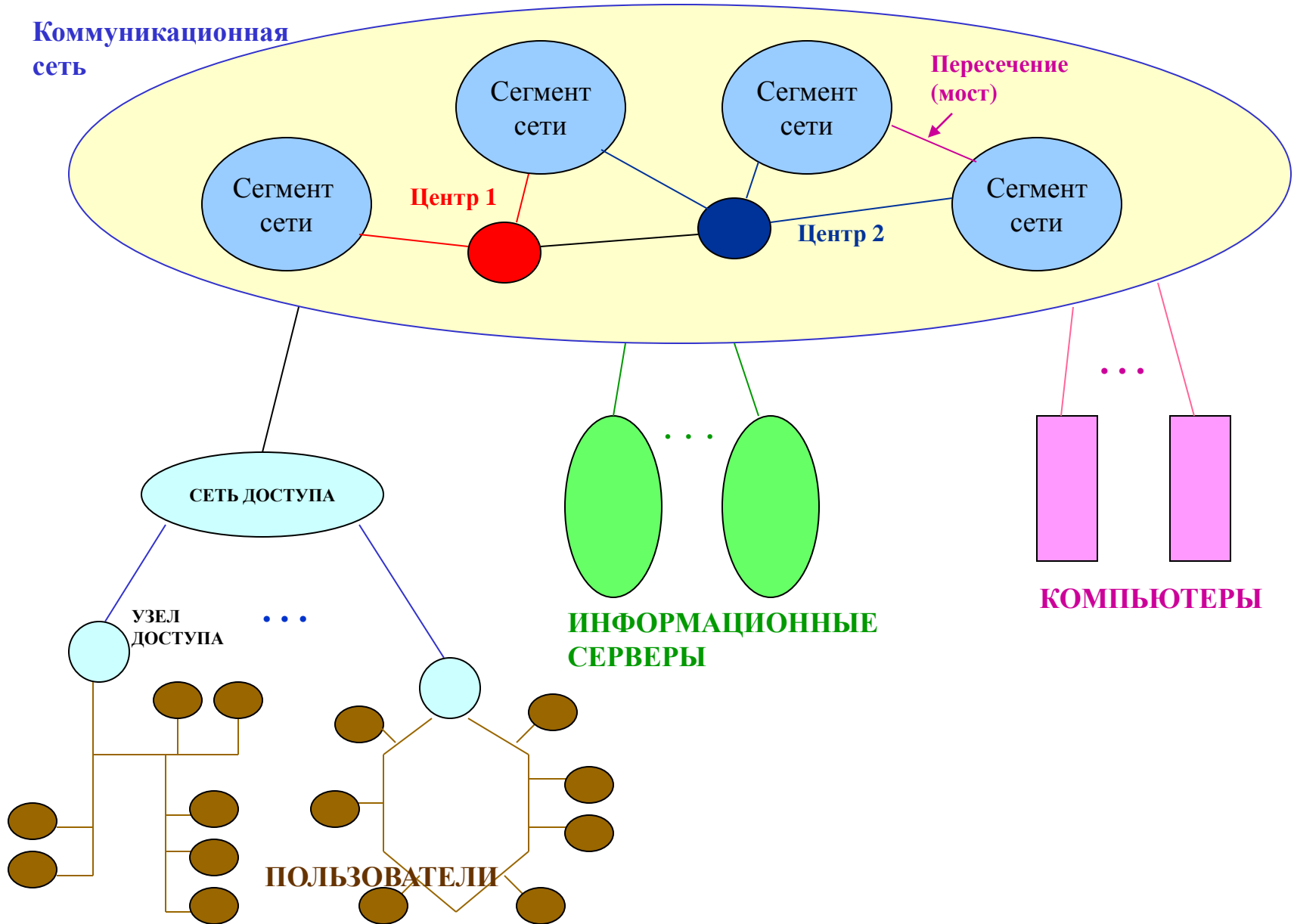
в распределенных информационных / вычислительных средах

(пользователи, коммуникационные сети, информационные серверы / компьютеры)

14.Проектирование топологии коммуникационных систем

Упрощенная схема коммуникационной системы

Коммуникационная сеть



ЛЕКЦИЯ 5-6. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Ин-т проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.5. Информационная технология. Участие человека.

Л.6. Схемы проектирования. Задачи проектирования.

ПЛАН:

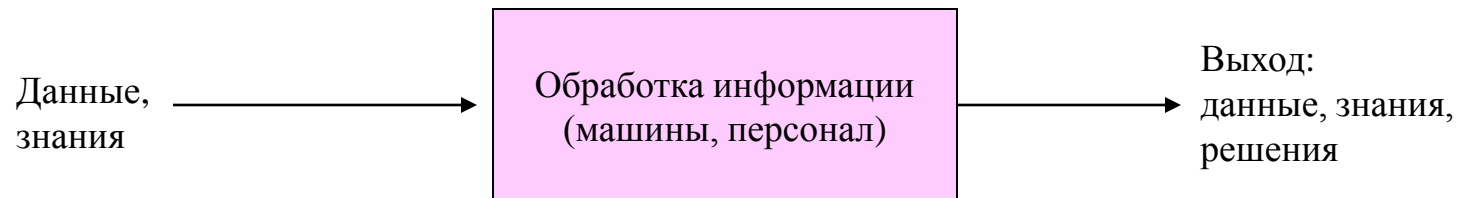
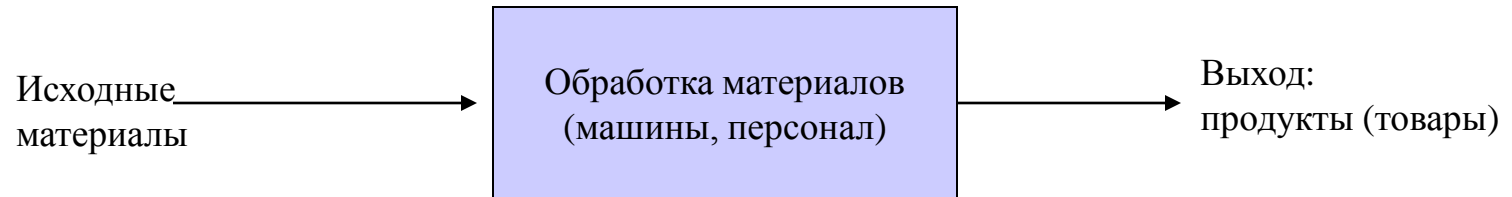
1. Информационная технология и ее свойства
2. Организационно-технические системы . Участие человека (в системе, в проектировании)
3. Схемы проектирования (последовательный процесс, каскадная схема).
Близкие схемы при обработке информации
4. Основные проблемы проектирования (проектирование, пере-проектирование, много-стадийное проектирование, оценивание, поиск узких мест, модели эволюции / развития систем)

Сент. 11, 2004

Информационная технология: структура

	Исслед.	Произв.	Тестиров.	Рынок	Используй./Техобслуж.	Утилизация
ОБОРУДОВАНИЕ						
*VLSI			●			●
*компьютеры			●			
*коммуникации			●			
ПРОГРАММЫ						
*опер. системы						
*DBMSs						
*коммун. пакеты						
МАТ. /АЛГОРИТМЫ						
*модели	●					
*алгоритмы	●					
ИНФОРМАЦИЯ						
*данные						●
*знания	●	●			●	●
ОРГ. ЧАСТЬ						
*специалисты						●
*пользователи						
*чел.-маш. интерфейс						
*групповая работа		●				
ПРИКЛ. СИСТЕМЫ						
*MISs						
*DSSs & ESs					●	
*etc.						

“Обработка” (как конвейер)



Сравнение: обработка материалов & обработка информации

СТАДИИ	Технология обработки древесины	Информационная технология
Источник исходных материалов	Лес	1.Книги, газеты 2.Базы данных / знаний 3.Люди
Исходные материалы	Древесина	1.Данные 2.Знания
Транспортирование	Автомобили, поезда	Коммуникационные системы
Производство: *машины *персонал	Машины Инженеры, рабочие	Компьютеры, программы, коммуникации 1.Специалисты 2.Пользователи
Выход	Доски и др.	1.Данные 2.Знания 3.Решения
Хранение	Склады	1.Базы данных 2.Базы знаний
Пользователи	Строительные фирмы, Частные лица	1.Государство 2.Фирмы 3.Университеты и др. 4.Исследовательские организации 5.Частные лица

Особенности информационной технологии

1. Различные источники: *статистика, книги, базы данных *специалисты, население
2. Сохранение исходной информации и возможность повторной обработки
3. Возможность параллельной обработки
4. Возможность использования различных методов
5. Возможность накапливать результаты обработки
6. Высокая “экологичность”
7. Высокие требования к персоналу
8. Уникальная роль человека
9. Высокие требования к представлению информации (визуализация)
10. Интеграция:
 - *точные науки
 - *техника
 - *психология
 - *обучение
 - * искусство (телевидение, кино)
11. Широкий круг пользователей:
 - *наука
 - *промышленность
 - *управление, экономика
 - *образование
 - *искусство
 - *частная жизнь

Морфологическая схема специалиста

Системная часть /область	Уровень специалиста по Ж.Пиаже (J. Piaget)			
	А. До-объектный /до-операционный (0..5)	В. Конкретные объекты/операции(5...11)	С. Абстрактные объекты/операции(11...)	
Прикладная задача	●	→	● ●	
Модель			● ● ●	
Алгоритм	●	→	●	●
Программы	● ●	→		●
Оборудование	●		● ●	
Другое			● ● ●	

Последовательная схема проектирования (J.R. Dixon)

ТРЕБОВАНИЯ
(государство,
рынок)



ПРОЕКТНЫЕ РЕШЕНИЯ

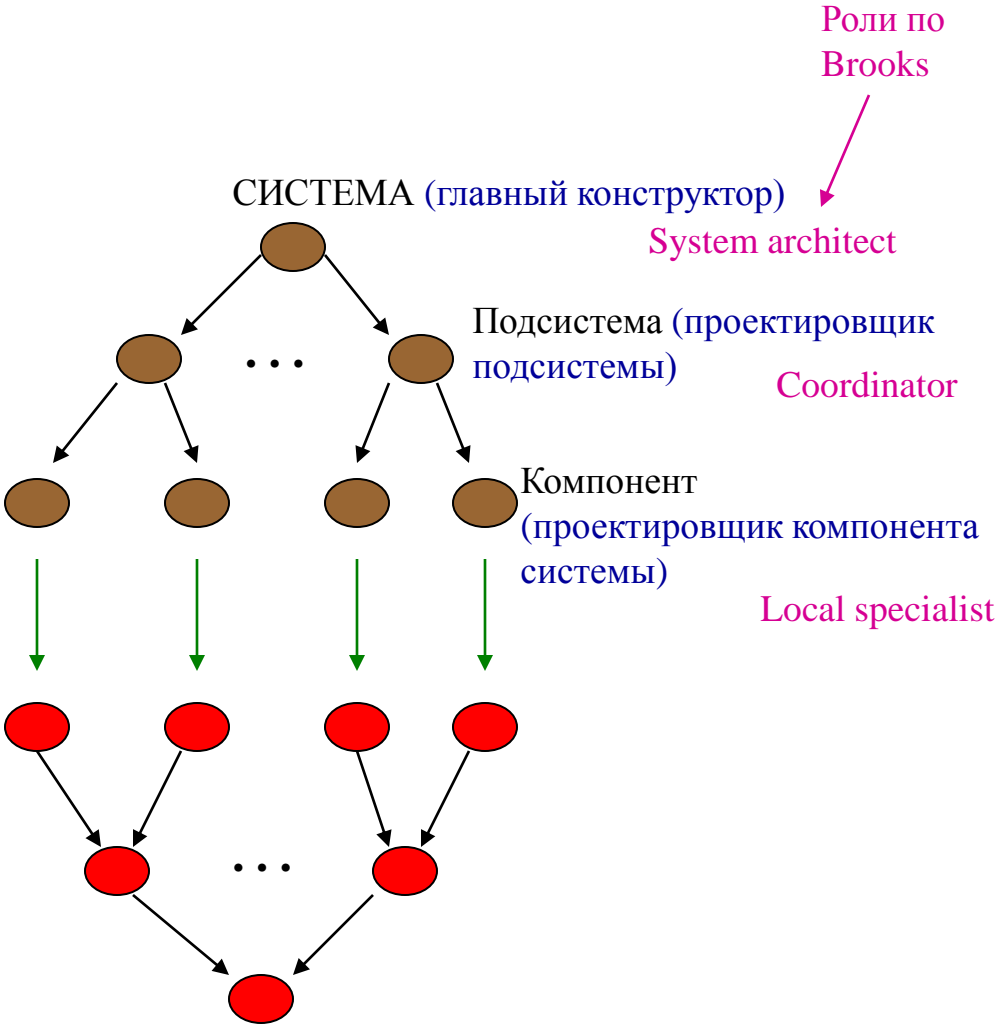
Каскадная схема проектирования

ТРЕБОВАНИЯ
(государство,
рынок)



ПРОЦЕСС
(Сверху-Вниз)
разбиение:
*системы
*требований

ПРОЦЕСС
(Снизу-Вверх):
*генерация
*оценивание
*выбор
*синтез



Уровни творчества (по Г. Альтшуллеру)

**УРОВЕНЬ 1. Использование известного объекта
(продукта, технологии, решения и др.)**

**УРОВЕНЬ 2. Поиск и выбор лучшего объекта
из некоторого исходного набора**

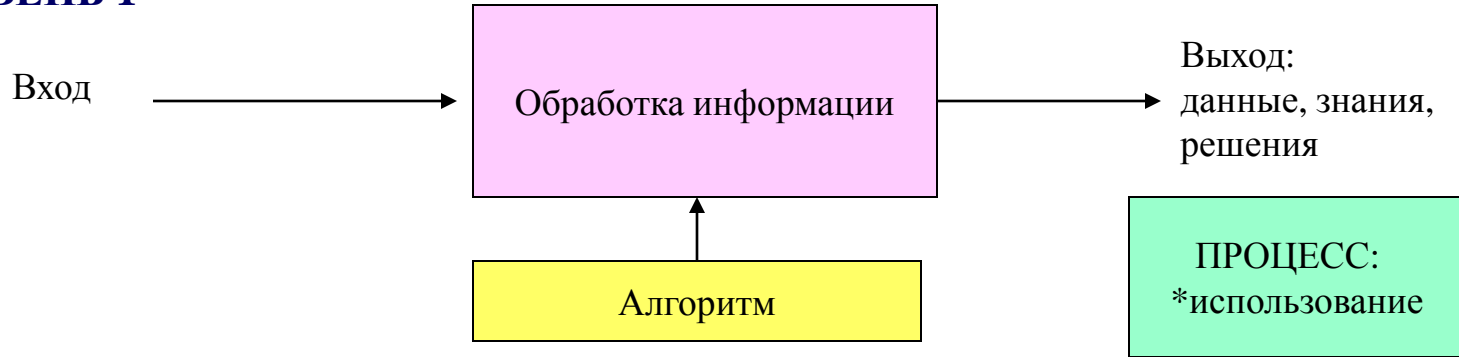
УРОВЕНЬ 3. Улучшение (модификация) существующего объекта

УРОВЕНЬ 4. Проектирование нового объекта

УРОВЕНЬ 5. Проектирование системы объектов

Иллюстрация «творческого уровня» для процесса обработки информации

УРОВЕНЬ 1



УРОВЕНЬ 2

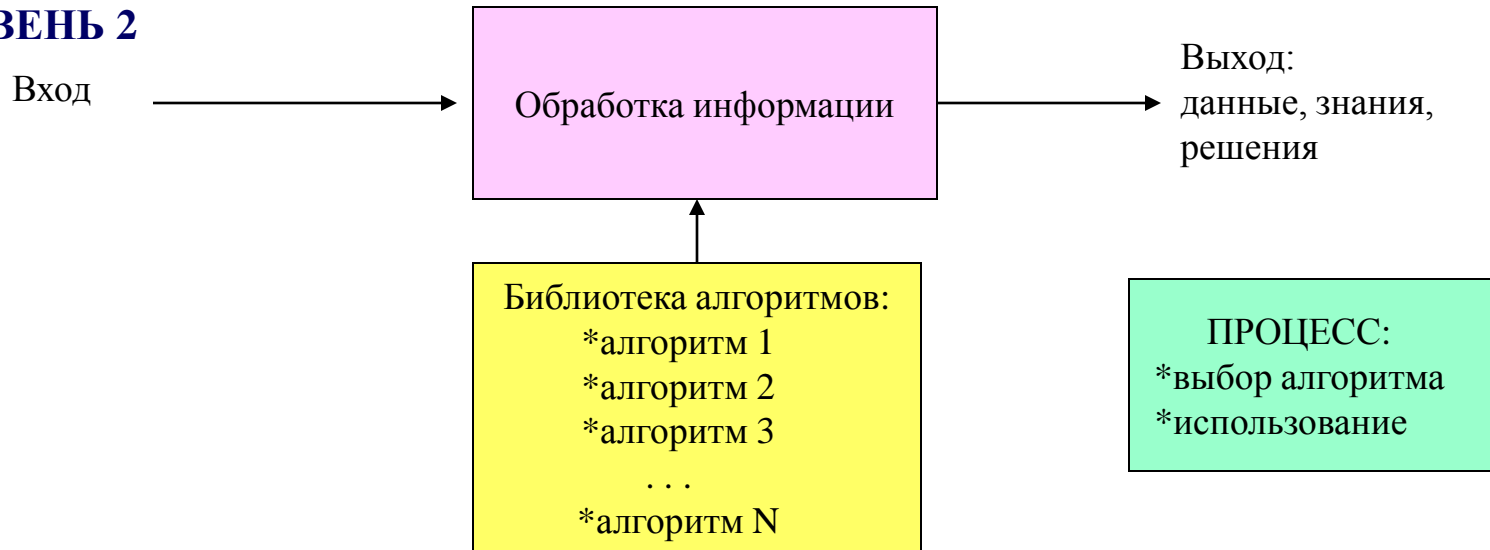


Иллюстрация «творческого уровня» для процесса обработки информации

УРОВЕНЬ 3

Вход

Обработка информации

Выход:
данные, знания,
решения

Модифицированный
алгоритм

ПРОЦЕСС:

- *выбор
- *модификация
- *использование

УРОВЕНЬ 4

Вход

Обработка информации

Выход:
данные, знания,
решения

Новый спроектированный
алгоритм

ПРОЦЕСС:

- *выбор
- *проектирование
нового алгоритма
- *использование

Иллюстрация «творческого уровня» для процесса обработки информации

УРОВЕНЬ 5

Вход

Обработка информации

Выход:
данные, знания,
решения

Новая алгоритмическая
система(библиотека
алгоритмов и др.),

ПРОЦЕСС:
*проектирование
новой системы
алгоритмов
*использование



1.Проектирование

2.Пере-проектирование (улучшение)

3.Многостадийное проектирование

4.Оценивание

5.Поиск «узких мест»

6.Моделирование эволюции / развития (& прогнозирование)

ЛЕКЦИЯ 7. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

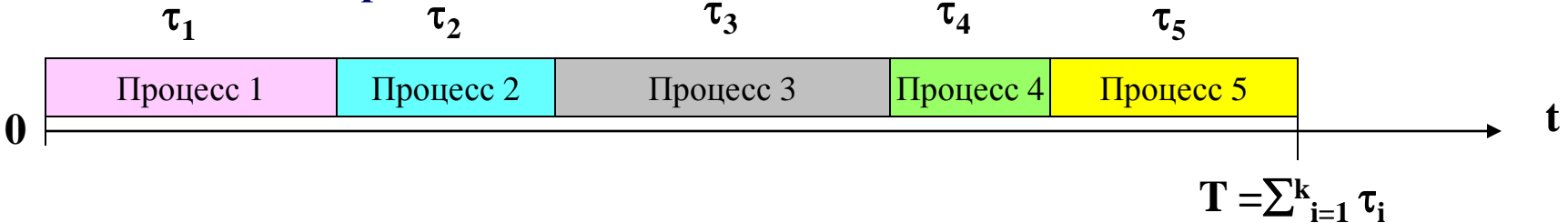
Л.7. Параллельный инжиниринг. Традиционная схема иерархического проектирования

ПЛАН:

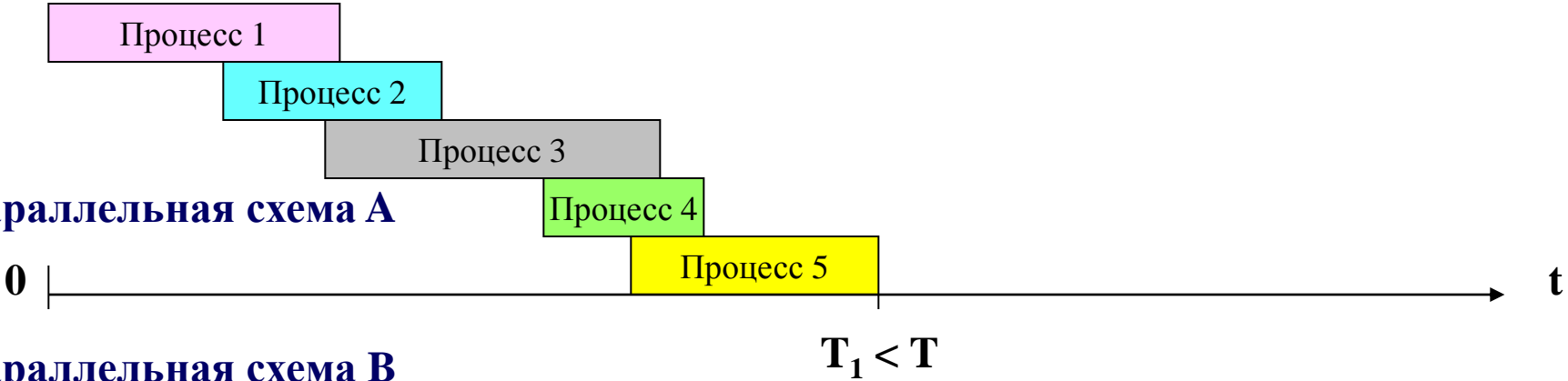
1. Параллельный (concurrent) подход (параллельные процессы, параллельность в жизненном цикле)
2. Иерархическое проектирование систем
3. Подходы к проектированию систем

Иллюстрация параллельного подхода

Последовательный процесс



Параллельная схема А



Параллельная схема В



Параллельность в жизненном цикле

цикл 1

цикл 2



Схема А



Схема В

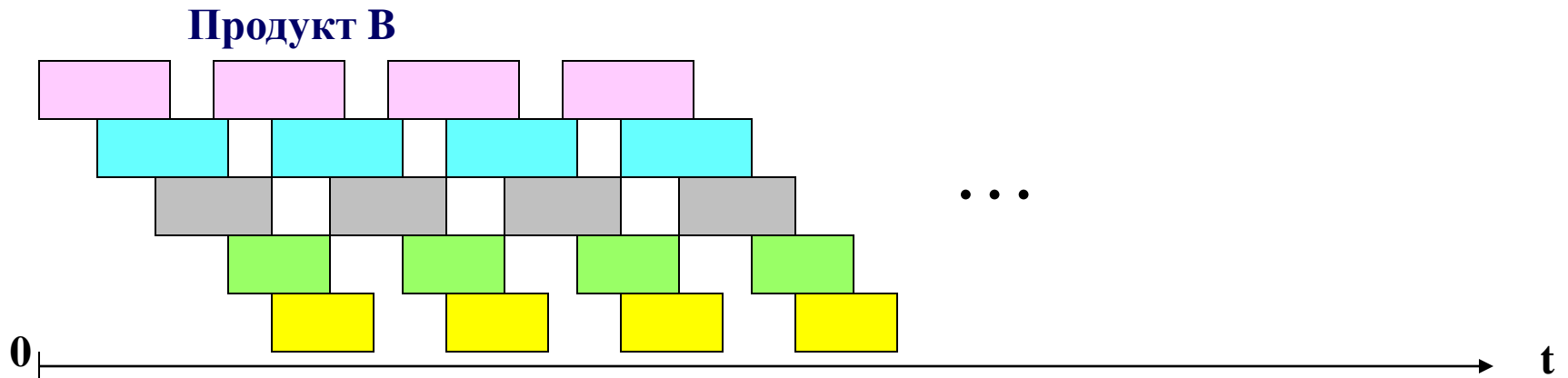
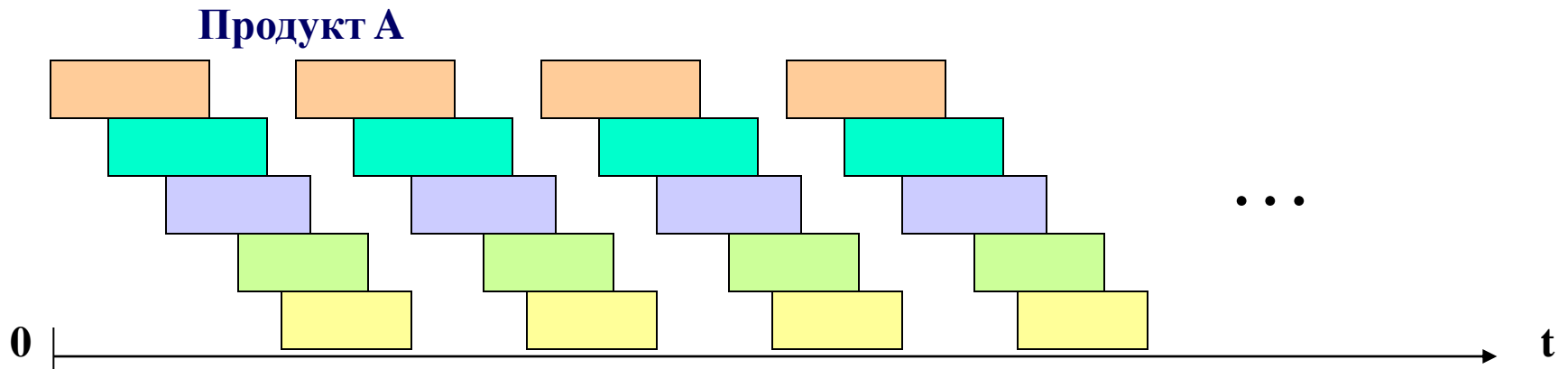


ПЛЮС: модульность,
управление конфигурациями

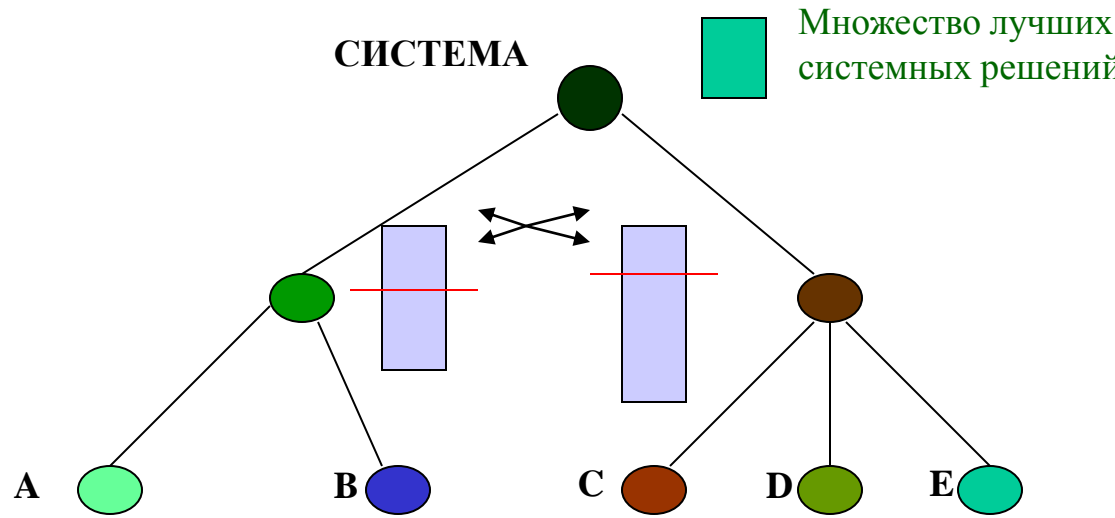
Параллельность в жизненном цикле для нескольких продуктов

ОРГАНИЗАЦИОННОЕ НАПРАВЛЕНИЕ:

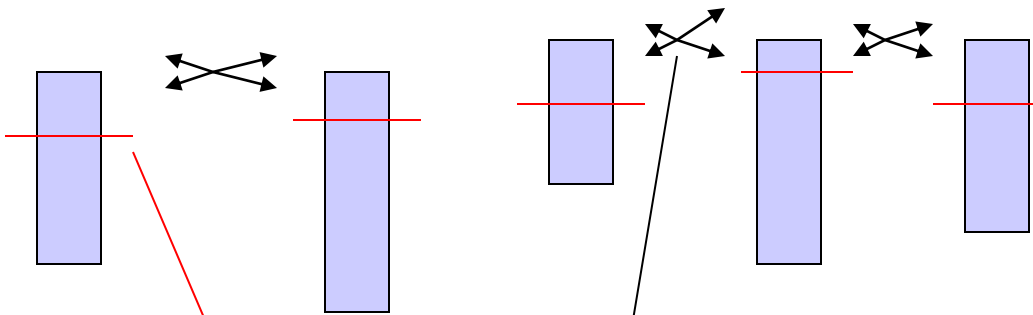
параллельность, модульность, управление конфигурациями и координация для продуктов А, В и т.д.



Иерархическое проектирование систем



ПРОЦЕСС
СНИЗУ-ВВЕРХ



Множество
альтернатив

Композиция
выбранных альтернатив

Выбор лучших
альтернатив

Процесс выбора:

1. Ограничения
2. Многокритериальный выбор (ранжирование)
3. Оптимизация
4. Экспертные процедуры

Основные подходы к проектированию систем

- 1. Многодисциплинарная оптимизация (Multidisciplinary optimization) (например, проектирование космических и авиационных систем, проектирование конструкций в строительстве)**
- 2. Смешанное целочисленное нелинейное программирование (например, проектирование химических процессов)**
- 3. Нелинейная многокритериальная оптимизация включая эволюционную многокритериальную оптимизацию**
- 4. Формальные методы в проектировании (например, в механике)**
- 5. Методы глобальной оптимизации**
- 6. «Грамматическое» проектирование (использование формальных грамматик при описании составных систем, например, в архитектуре)**
- 7. Методы искусственного интеллекта (системы на основе знаний, нейронные сети, генетические алгоритмы) (например, проектирование в электронике)**
- 8. Исследование пространства параметров PSI (проектирование в механике, атомных технологиях и др.)**
- 9. Иерархическое проектирование систем (традиционный организационный подход, модульное проектирование, комбинаторный синтез)**
- 10. Гибридные методы**

ЛЕКЦИИ 8-9. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

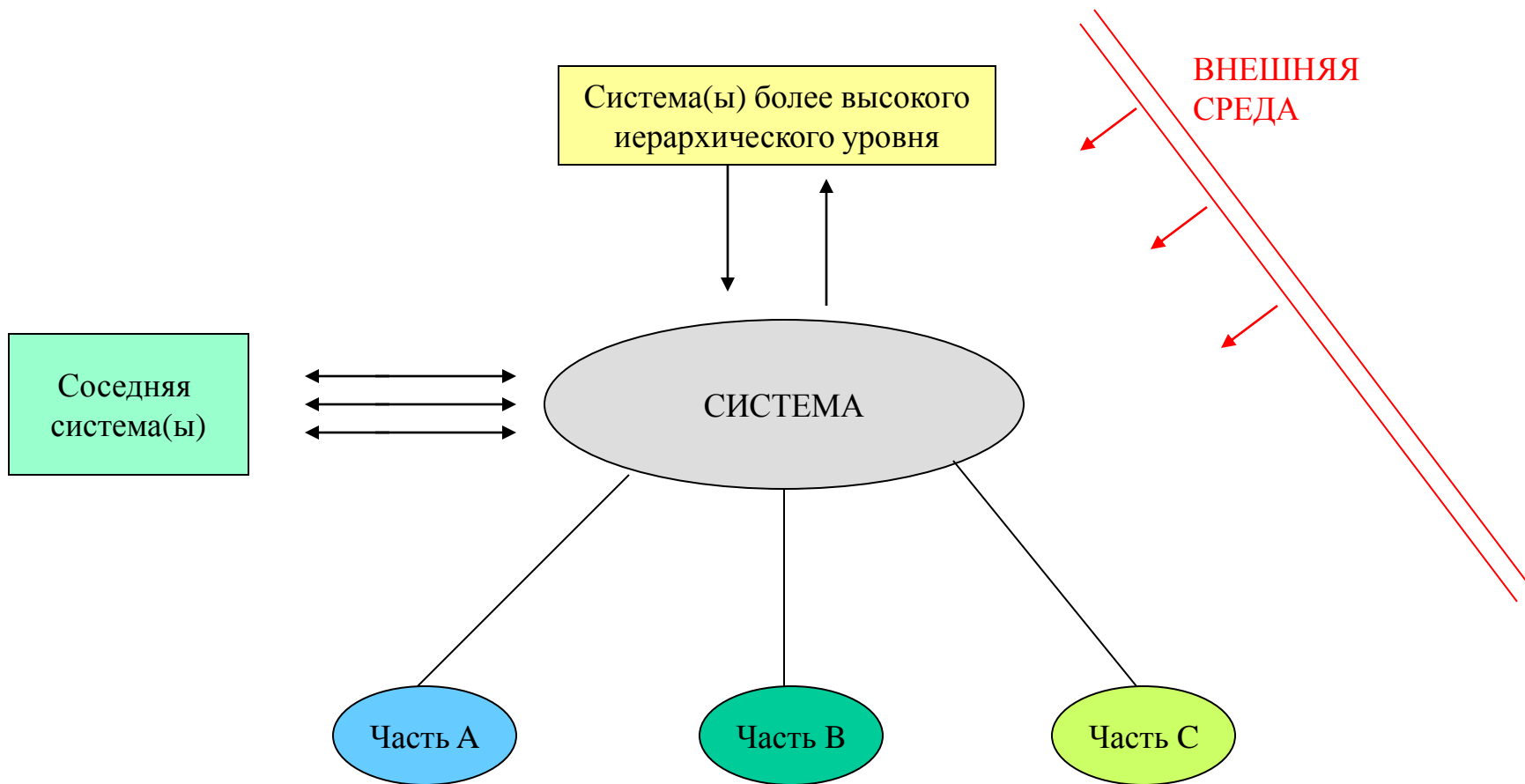
Л.8. Принципы системного анализа. Парадигма принятия решений. Задачи принятия решений.

Л.9. Типы шкал. Парето-эффективные решения. Оценивание, иерархия требований

ПЛАН:

- 1.Принципы системного анализа. 2.Парадигма принятия решений. 3.Основные задачи принятия решений.
- 4.Типы шкал. 5.Парето-эффективные решения 6.Оценивание систем
- 7.Иерархия требований / критериев 8.Роли в процессе принятия решений. Пример.

Схема системы



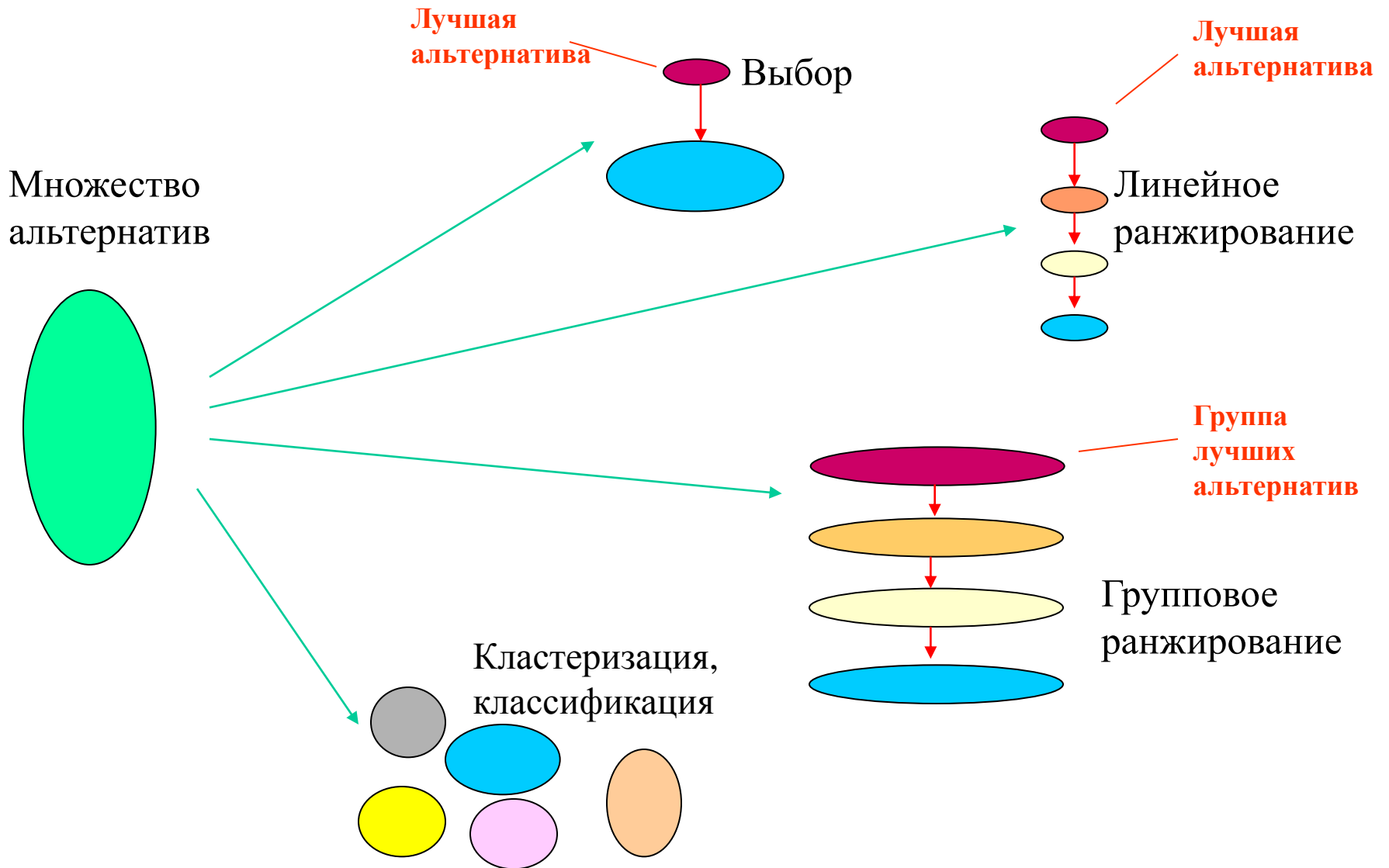
Основные принципы системного анализа

- 1. Исследование жизненного цикла (т.е., проектирование, производство, тестирование, маркетинг, использование & техобслуживание, утилизация)**
- 2. Исследование эволюции / развития систем (т.е., динамические аспекты)**
- 3. Исследование связей с средой (природа, общество, другие системы)**
- 4. Исследование связей внутри системы между частями / компонентами (физические части, функции, информация, энергия и др.)**
- 5. Анализ системных изменений (близко к принципу 2)**
- 6. Выявление и исследование главных системных параметров**
- 7. Интеграция различных методов (декомпозиция, иерархия, композиция и др.)**
- 8. Исследование главных системных противоречий (технических, экономических, экологических, политических и др.)**
- 9. Интеграция различных моделей и алгоритмов (физические эксперименты, математическое моделирование, эвристики, экспертные процедуры)**
- 10. Взаимодействие между специалистами из различных профессиональных областей и иерархических уровней (техника, компьютерные науки, математика, управление, социальные науки и др.)**

Парадигма принятия решений (стадии) по Герберту А. Саймону

- 1. Анализ прикладной проблемы (понять проблему: главные противоречия и др.)**
- 2. Структуризация проблемы:**
 - 2.1. Генерация альтернатив**
 - 2.2. Проектирование критериев**
 - 2.3. Проектирование шкал для оценивания альтернатив по критериям**
- 3. Оценивание альтернатив по критериям**
- 4. Выбор лучшей альтернативы (или нескольких лучших)**
- 5. Анализ результатов**

4 основные задачи принятия решений



Классификация проблем по Г.А. Саймону

I. СТАНДАРТНЫЕ ПРОБЛЕМЫ

II. ФОРМУЛИЗУЕМЫЕ ПРОБЛЕМЫ

(математические модели как уравнения, оптимизация и др.)

III. СЛАБО-СТРУКТУРИЗУЕМЫЕ ПРОБЛЕМЫ

*человеческие факторы, информация от экспертов & лицо принимающее решение

*неопределенность

IV. ПРОГНОЗИРОВАНИЕ (решения для будущего)

Проблемы
принятия
решений



Прикладные проблемы принятия решений

1. УРОВЕНЬ ГОСУДАРСТВА:

- *выбор исследовательских проектов
- *инвестиции в инфраструктуру (транспорт, коммуникации, образование)
- *выбор политических решений

2. УРОВЕНЬ КОМПАНИИ:

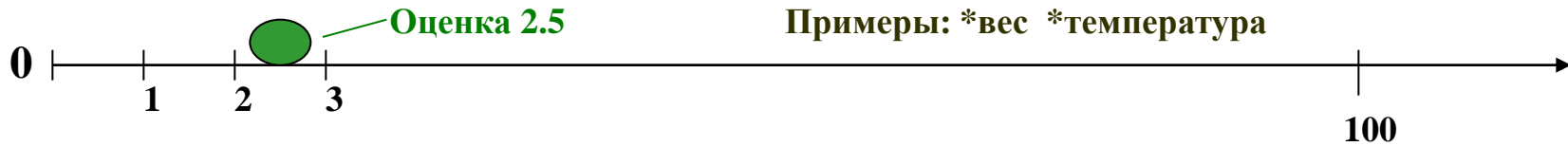
- *выбор продукта
- *выбор рынка
- *выбор персонала
- *выбор партнеров
- *выбор места для нового завода

3. УРОВЕНЬ ЧАСТНОЙ ЖИЗНИ:

- *выбор квартиры
- *выбор университета
- *выбор автомобиля
- *выбор программы в банке
- *выбор места отдыха

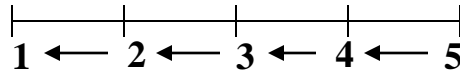
Типы шкал

1. Количество: количественная шкала

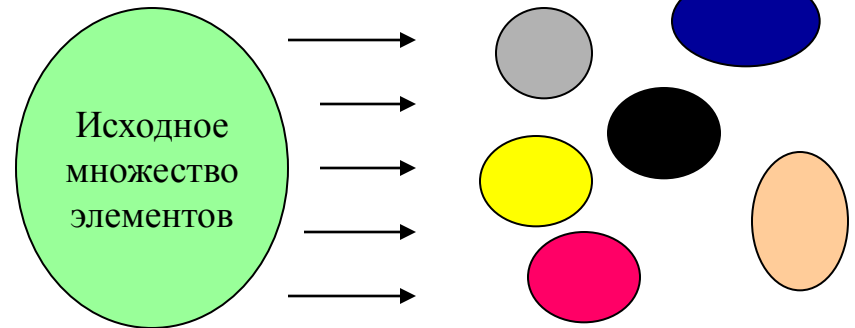


2. Качество: качественная шкала (уровень, порядок, класс)

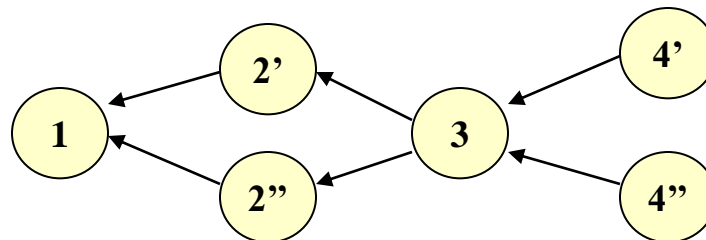
2а. Порядковая шкала



2б. Номинальная шкала (для классов, кластеров)



2с. Шкала как частичный порядок (обобщение)



Описание проблемы принятия решений

Альтернативы $A=(A_1, \dots, A_i, \dots, A_n)$ и критерии $C=(C_1, \dots, C_j, \dots, C_k)$,
 $\forall A_i$ вектор оценок $z_i = (z_{i1}, \dots, z_{ij}, \dots, z_{ik})$

Матрица оценок:

$$Z = \begin{array}{|c|} \hline z_{11}, \dots, z_{1j}, \dots, z_{1k} \\ \hline \dots \\ \hline z_{i1}, \dots, z_{ij}, \dots, z_{ik} \\ \hline \dots \\ \hline z_{n1}, \dots, z_{nj}, \dots, z_{nk} \\ \hline \end{array} \begin{array}{l} \longrightarrow P(A_1) \\ \longrightarrow P(A_i) \\ \longrightarrow P(A_n) \end{array}$$

Наша цель заключается в получении “приоритета” для каждой альтернативы: $P(A_i)$

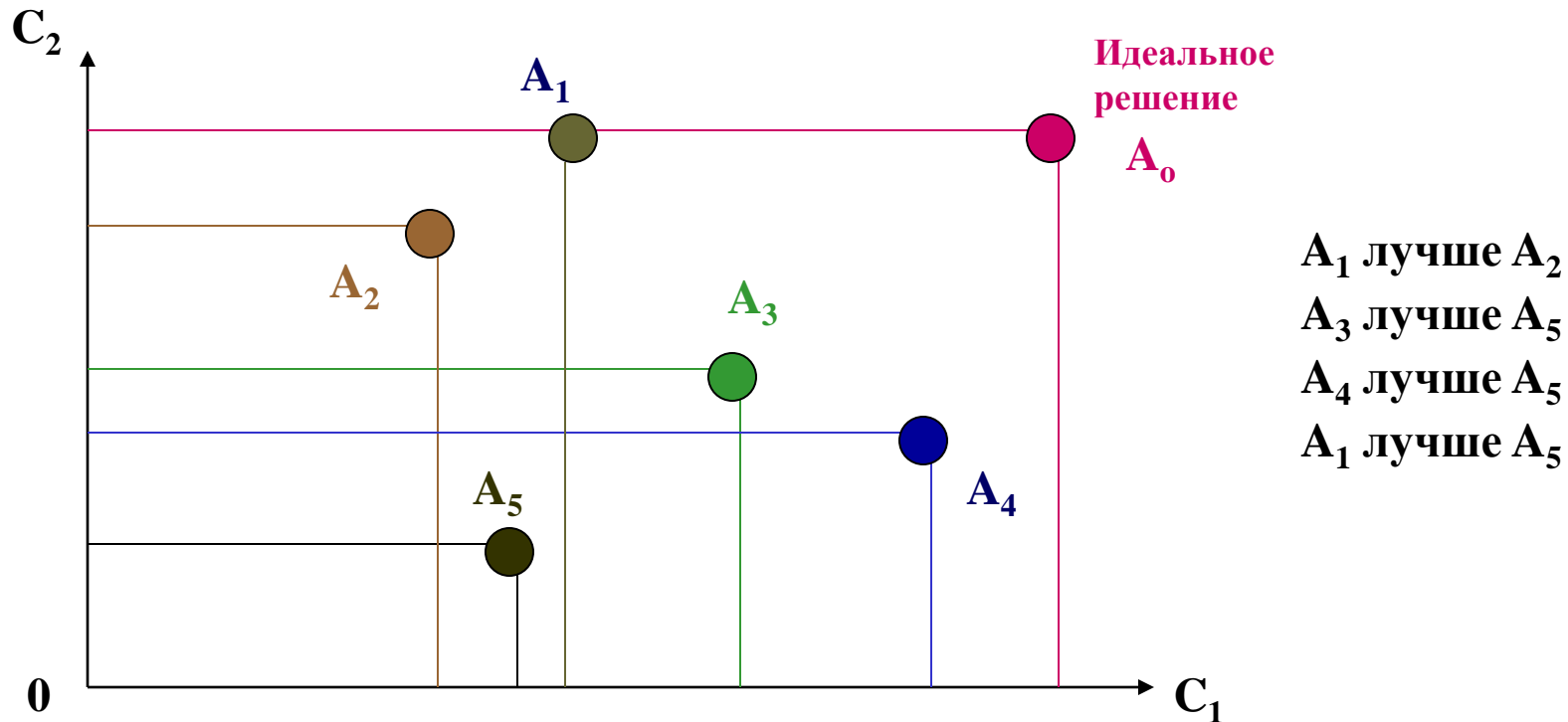
Оценивание $P(A_i)$ может быть основано на следующем:

1. Количественная шкала
2. Порядковая шкала
3. Шкала как частичный порядок

Парето-эффективные (Парето-оптимальные) решения

ПРАВИЛО ПАРЕТО:

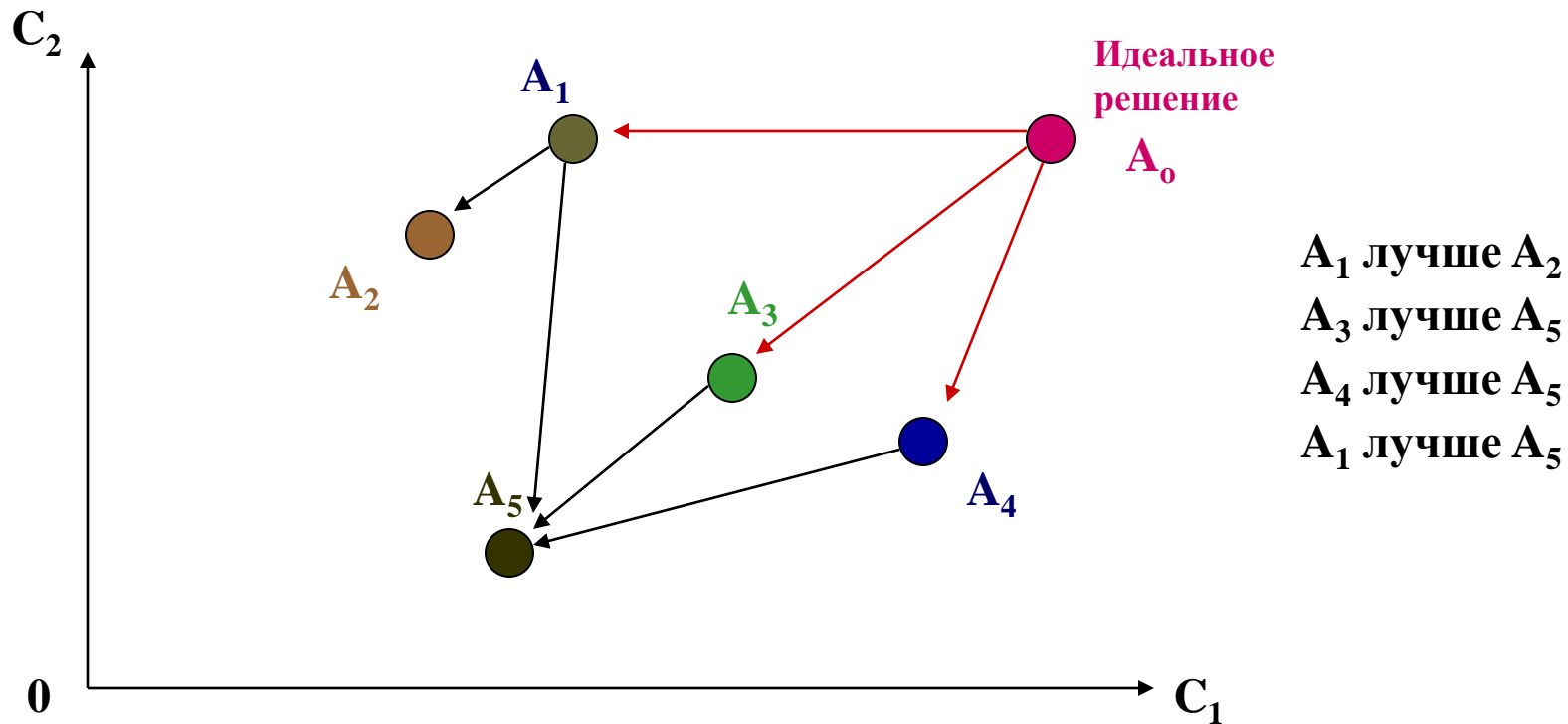
Альтернатива $X=(x_1, \dots, x_j, \dots, x_k)$ и альтернатива $Y=(y_1, \dots, y_j, \dots, y_k)$,
 X лучше чем Y если $\forall j \quad x_j \geq y_j$ и $\exists i \quad (1 \leq i \leq k)$ такой что $x_i > y_i$



A_1, A_3, A_4 несравнимы и не имеют доминирующих элементов (только A_0)

A_1, A_3, A_4 являются Парето-эффективными решениями для множества $\{A_1, A_2, A_3, A_4, A_5\}$

Частичный порядок на альтернативах



A_1, A_3, A_4 несравнимы и не имеют доминирующих элементов (только A_0)

A_1, A_3, A_4 являются Парето-эффективными решениями на множестве $\{A_1, A_2, A_3, A_4, A_5\}$

Оценивание сложных систем может быть основано на следующем:

1. Количественная шкала

2. Порядковая шкала

3. Шкала как частичный порядок

(включая специальные дискретные пространства)

1. Экология, политика

2. Экономика, рынок

3. Технология (т.е., вопросы производства, вопросы техобслуживания)

4. Непосредственно инженерное проектирование

Главные роля в процессе принятия решений

1.ЛИЦО ПРИНИМАЮЩЕЕ РЕШЕНИЕ (ЛПР)

(выбрать окончательное решение, оценить альтернативы и др.)

2.СПЕЦИАЛИСТ – ОРГАНИЗАТОР ПРОЦЕДУРЫ

(организовать процедуру принятия решений включая поддержку всех стадий)

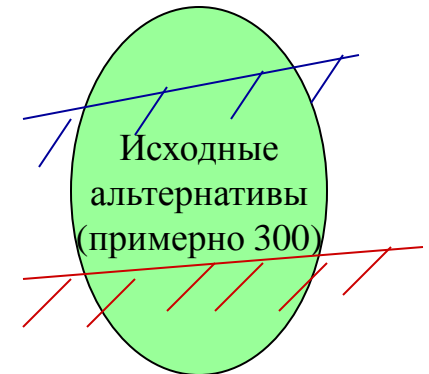
3.ЭКСПЕРТ(Ы) (оценивать альтернативы)

Фазы процесса принятия решений: Пример выбора лучшей компании (P. Humphreys)

Фаза 1. Анализ исходных заявок (т.е., альтернатив) & исключение наихудших материалов примерно 1/3 всех заявок



Фаза 2. Проектирование специального метода для многокритериального выбора, оценивание альтернатив по критериям, выбор группы лучших альтернатив (примерно 20...30) (участие группы экспертов)



Фаза 3. Выбор лучшей альтернативы (или нескольких): special procedure of expert judgment (групп)



Лекция 10. Курс: “Проектирование систем: Структурный подход”

Каф.”Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.10. Многокритериальное принятие решений

ПЛАН:

1. Многокритериальное принятие решений: *функция полезности, * метод парных сравнений, *метод уровней несравнимости (эквивалентности), *метод порогов несравнимости (ELECTRE), *метод аналитических иерархий (АНП) и др.
2. Интегрирование результатов, полученных на основе нескольких методов (или систем критериев)

Примеры функции полезности (свертки)

Альтернативы $A=(A_1, \dots, A_i, \dots, A_n)$ и критерии $C=(C_1, \dots, C_j, \dots, C_k)$,
 $\forall A_i$ вектор оценок $z_i = (z_{i1}, \dots, z_{ij}, \dots, z_{ik})$, μ_j вес для критерия j

Арифметическая $F_a = \sum_{j=1}^k \mu_j z_j / z_{jb}$

Геометрическая $F_g = \prod_{j=1}^k (z_j / z_{jb})^{\mu_j}$

Квадратичная $F_q = \sum_{j=1}^k \mu_j (z_j / z_{jb})^2$

Гармоническая $F_h = 1 / (\sum_{j=1}^k \mu_j (z_j / z_{jb}))$

Степенная $F_p = \sum_{j=1}^k \mu_j (z_j / z_{jb})^k$

Общий случай $F_o = \sum_{j=1}^k \mu_j \varphi (z_j / z_{jb})$

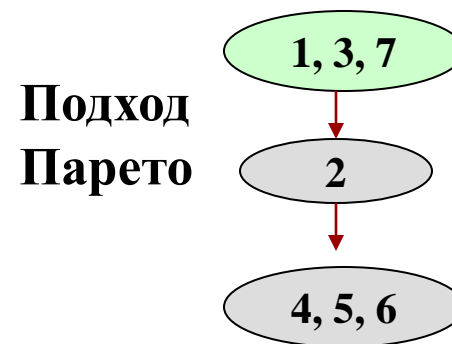
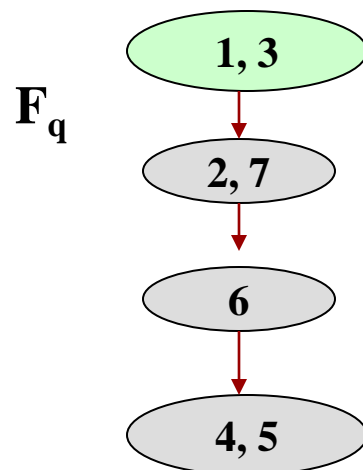
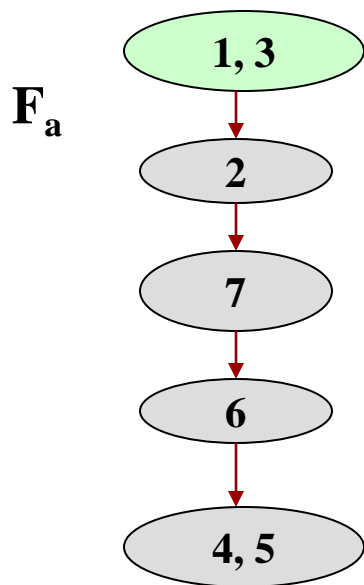
где φ является дифференцируемой функцией,

z_{jb} является базовой оценкой

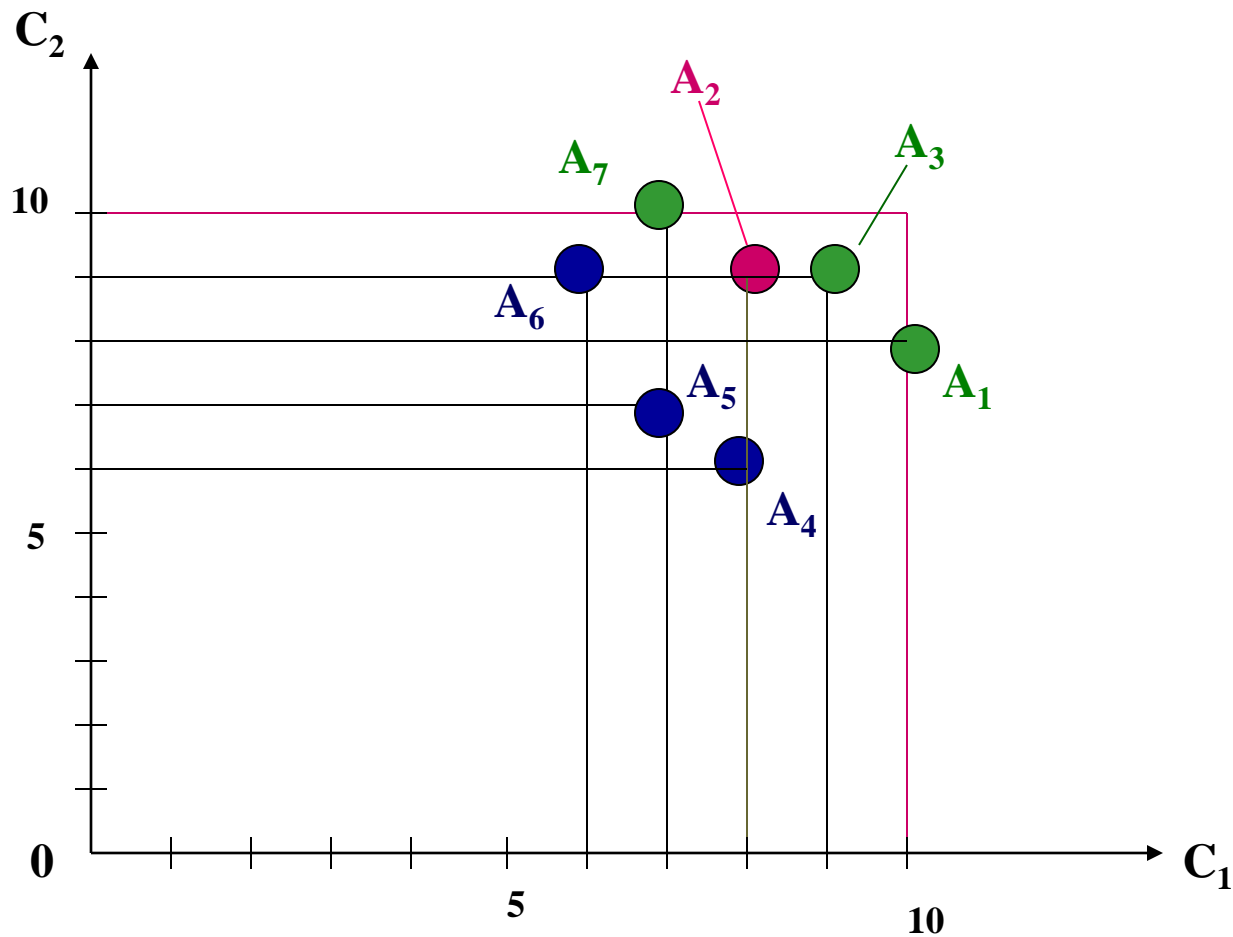
Иллюстративный численный пример для многокритериального ранжирования

	Математика C_1	Спорт C_2	F_a	F_q	Подход Парето
--	---------------------	----------------	-------	-------	------------------

A_1	10	8	18 / 1	164 / 1	1
A_2	8	9	17 / 2	145 / 2	2
A_3	9	9	18 / 1	162 / 1	1
A_4	6	8	14 / 5	100 / 4	3
A_5	7	7	14 / 5	98 / 4	3
A_6	9	6	15 / 4	117 / 3	3
A_7	10	7	16 / 3	149 / 2	1

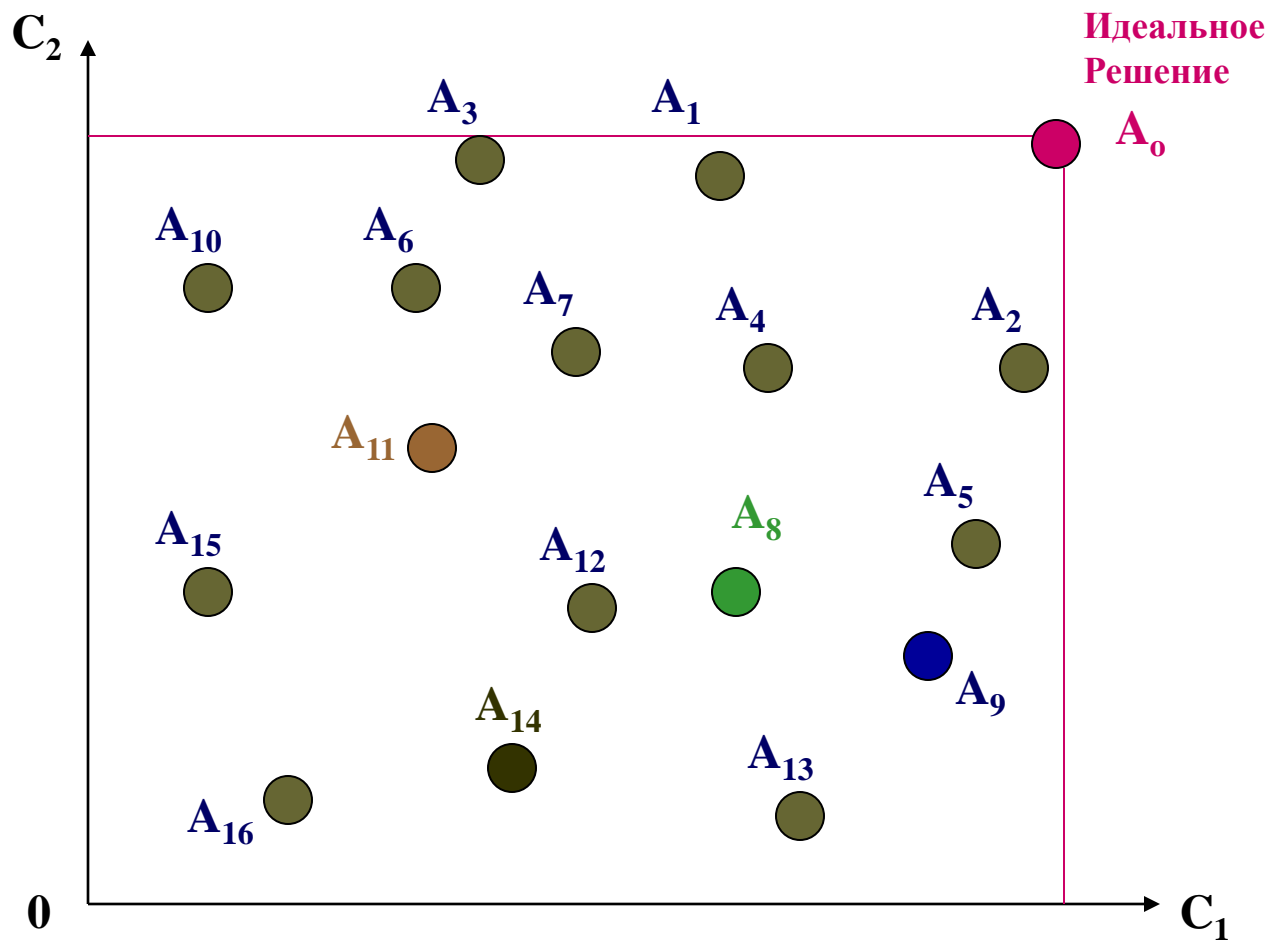


Подход Парето для предыдущего примера

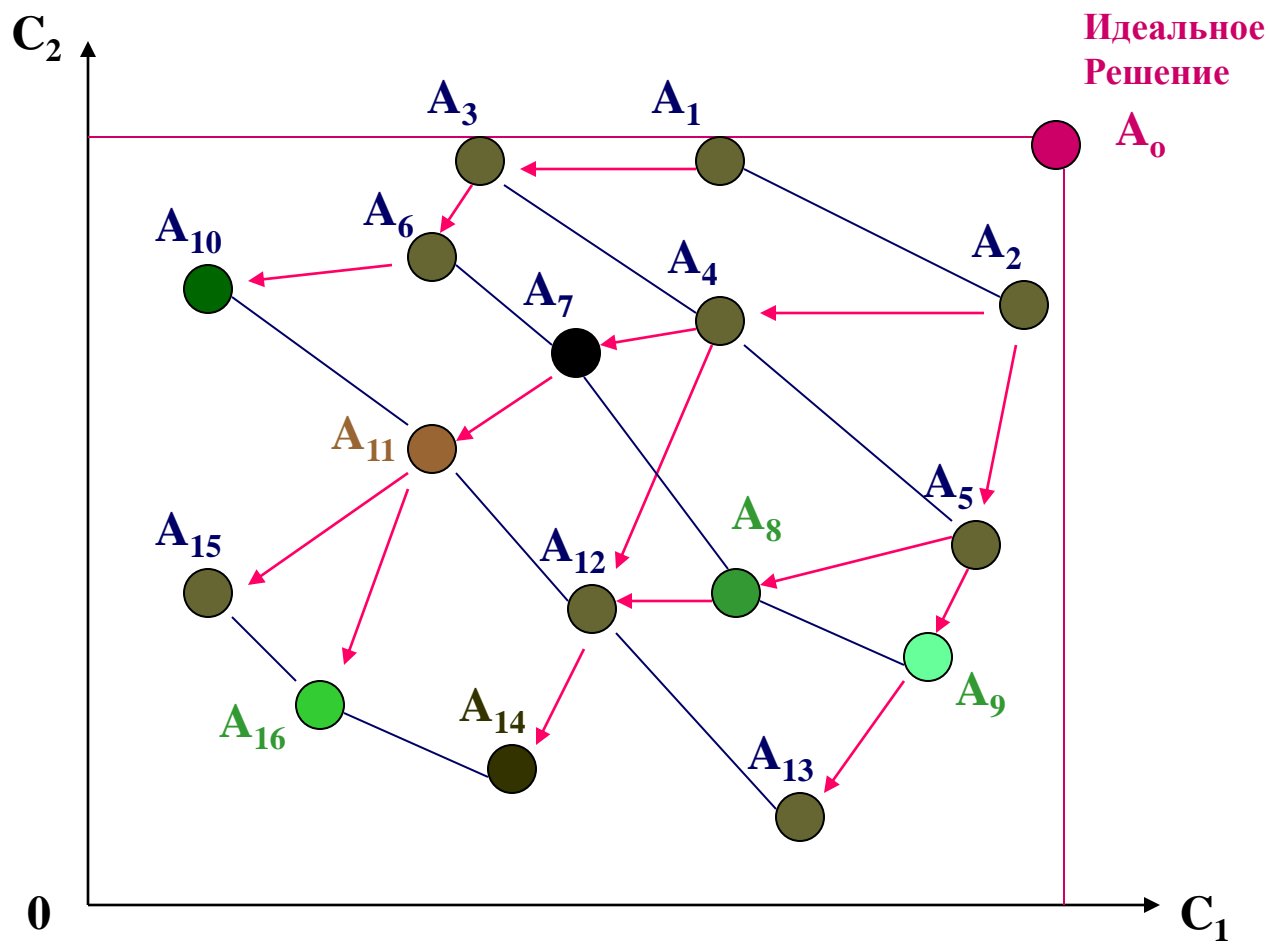


- A3 лучше A2
- A3 лучше A5
- A1 лучше A4
- A7 лучше A5
- A2 лучше A4
- A2 лучше A6
- A3 лучше A6
- A3 лучше A4
- A1 лучше A5
- A7 лучше A6
- A2 лучше A5

Метод уровней «эквивалентности» (несравнимости): Исходные альтернативы

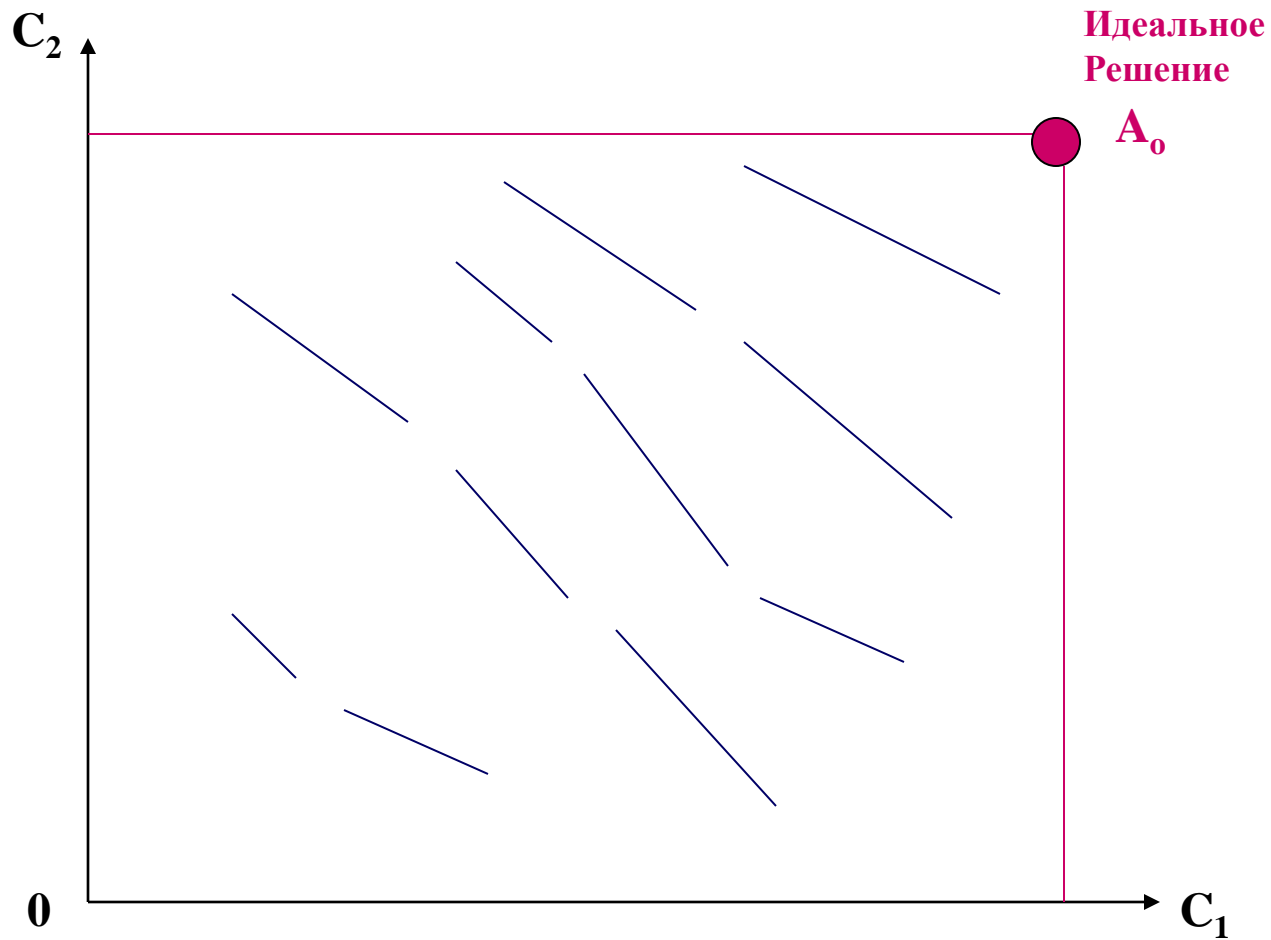


Метод уровней «эквивалентности» (несравнимости): Парные сравнения

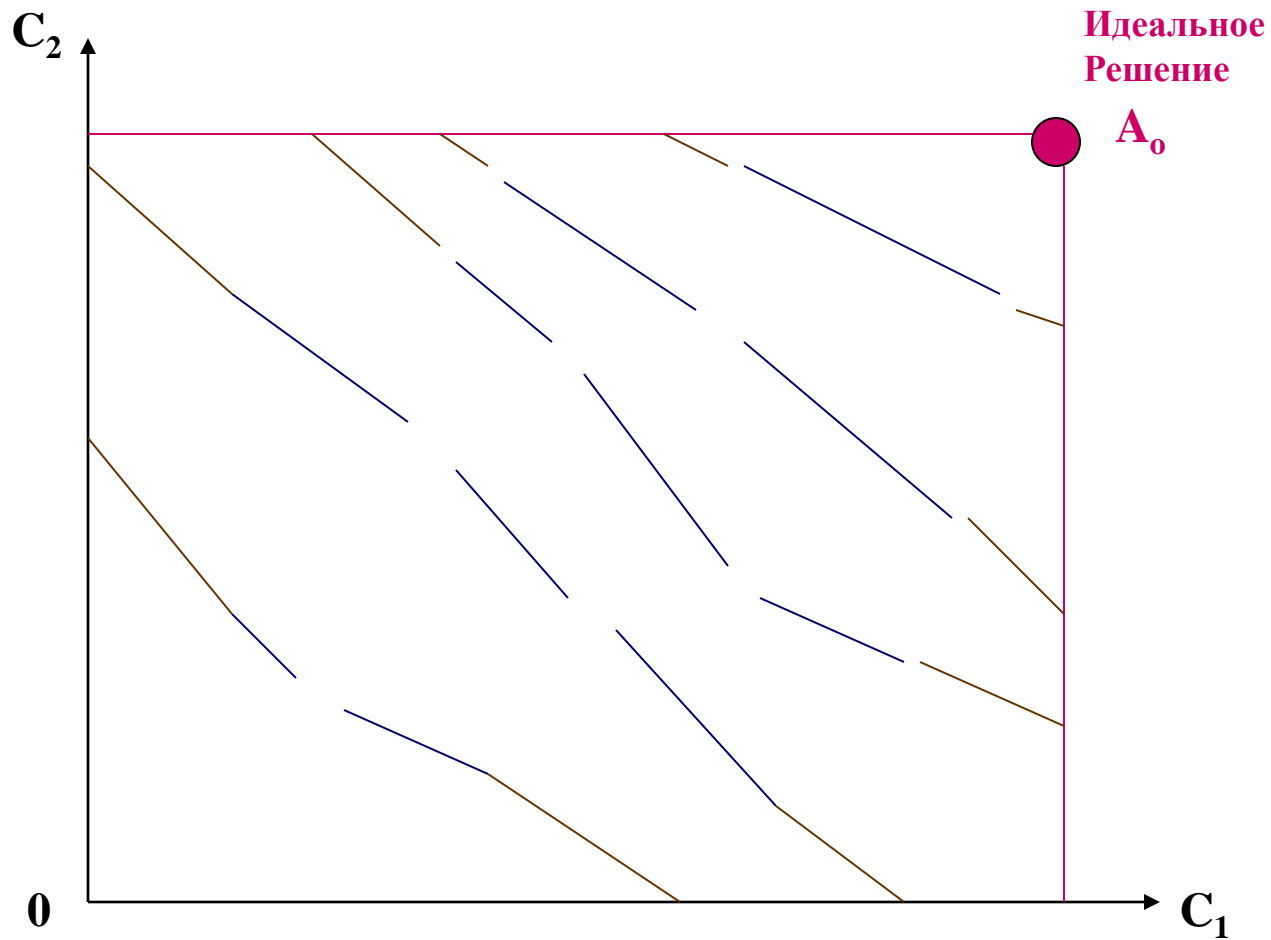


Парные сравнения:
1. Доминирование
2. Несравнимость

Метод уровней “эквивалентности” (несравнимости): уровни несравнимости



Метод уровней “эквивалентности” (несравнимости): расширение уровней несравнимости



Метод уровней “эквивалентности” (несравнимости): оценивание новых альтернатив

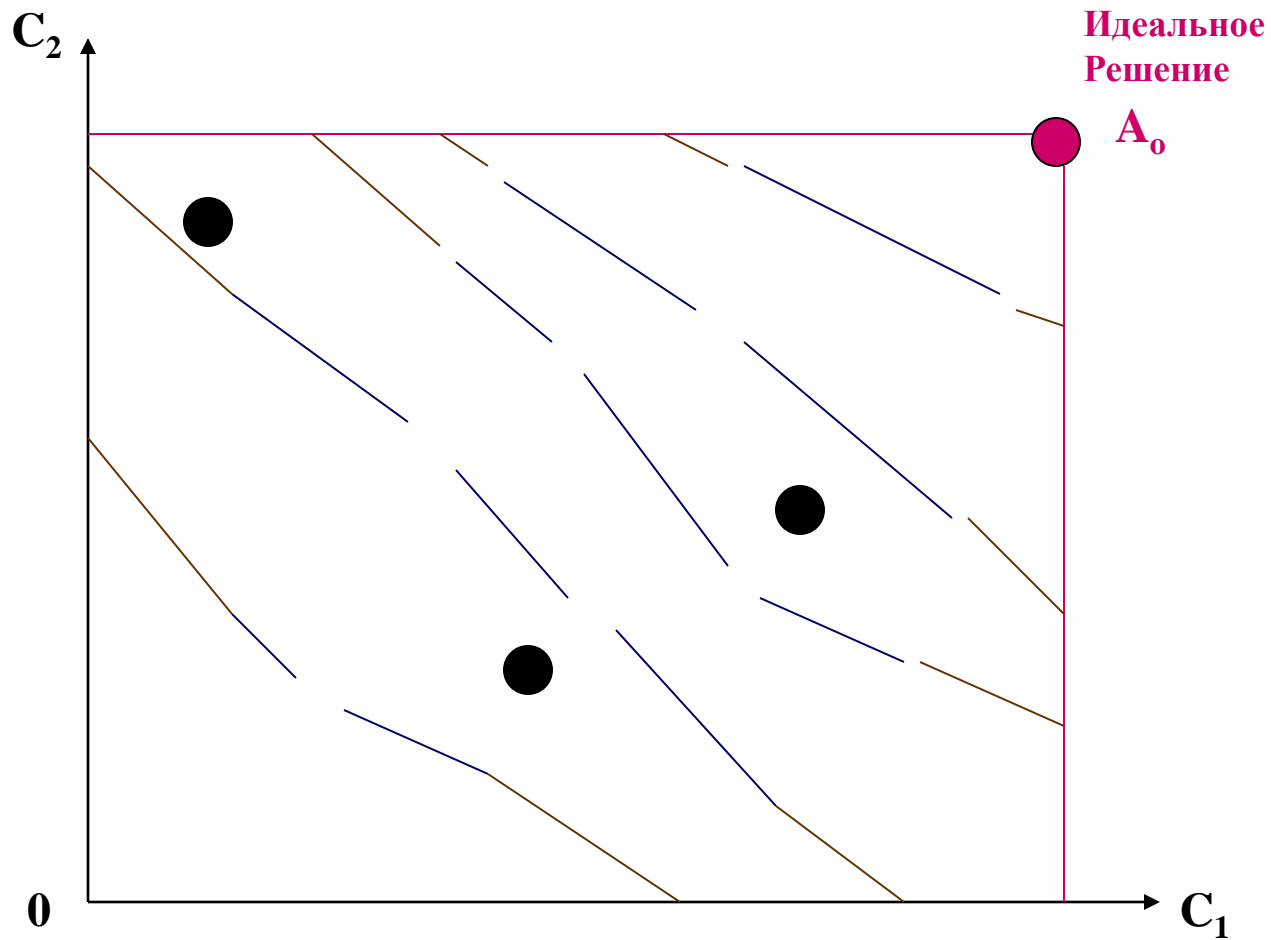


Иллюстрация для арифметической функции полезности: уровни несравнимости

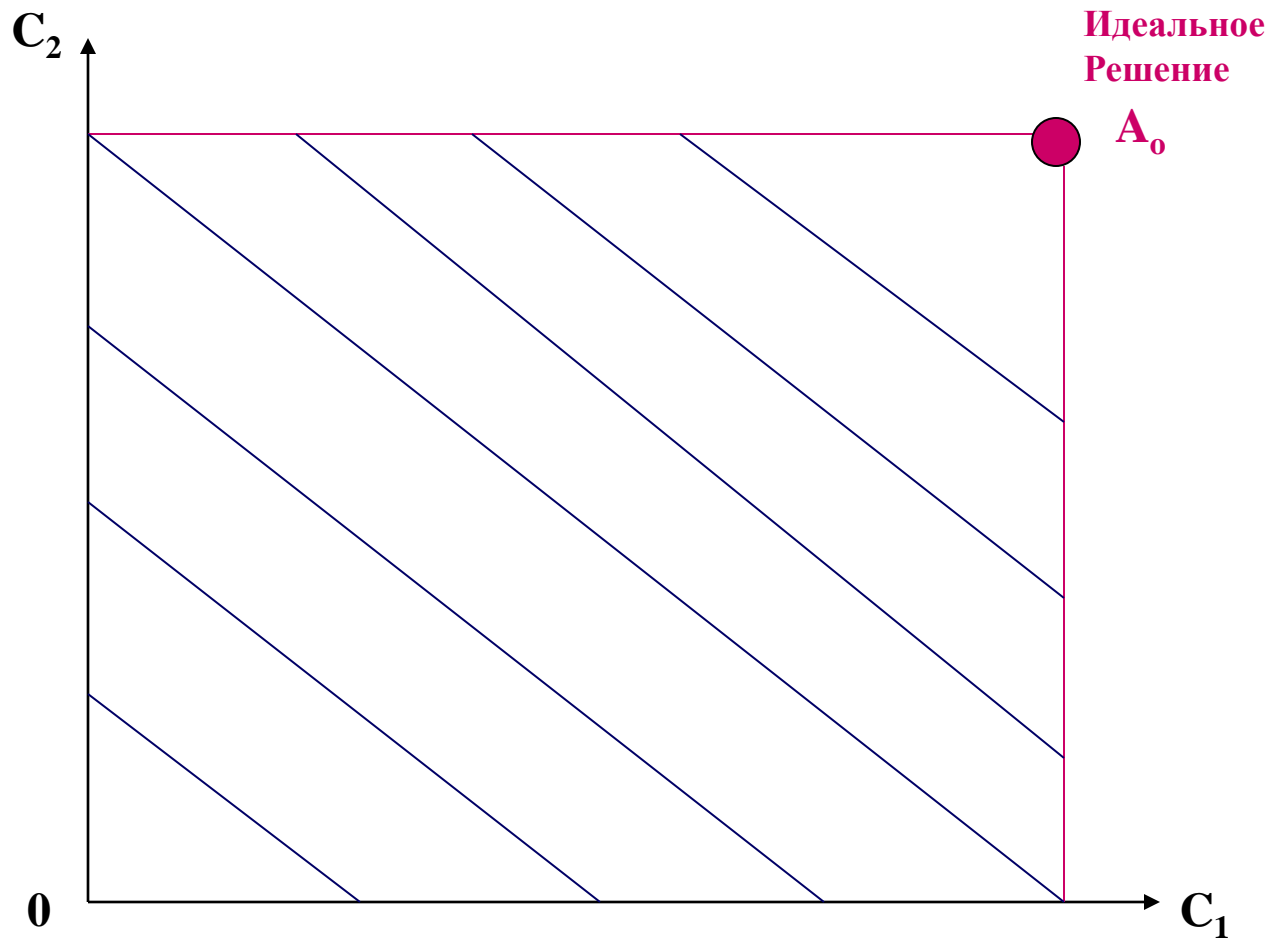
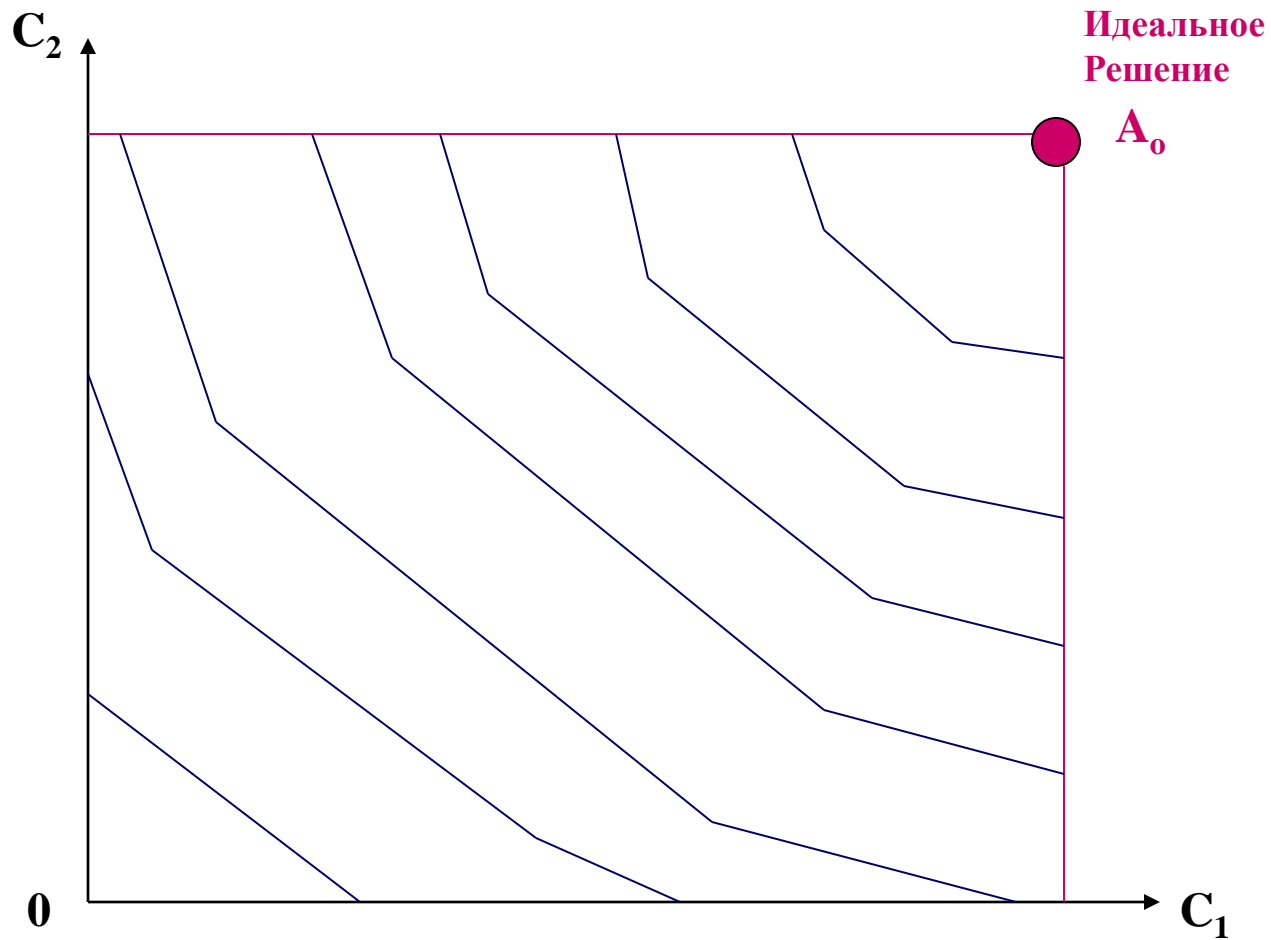
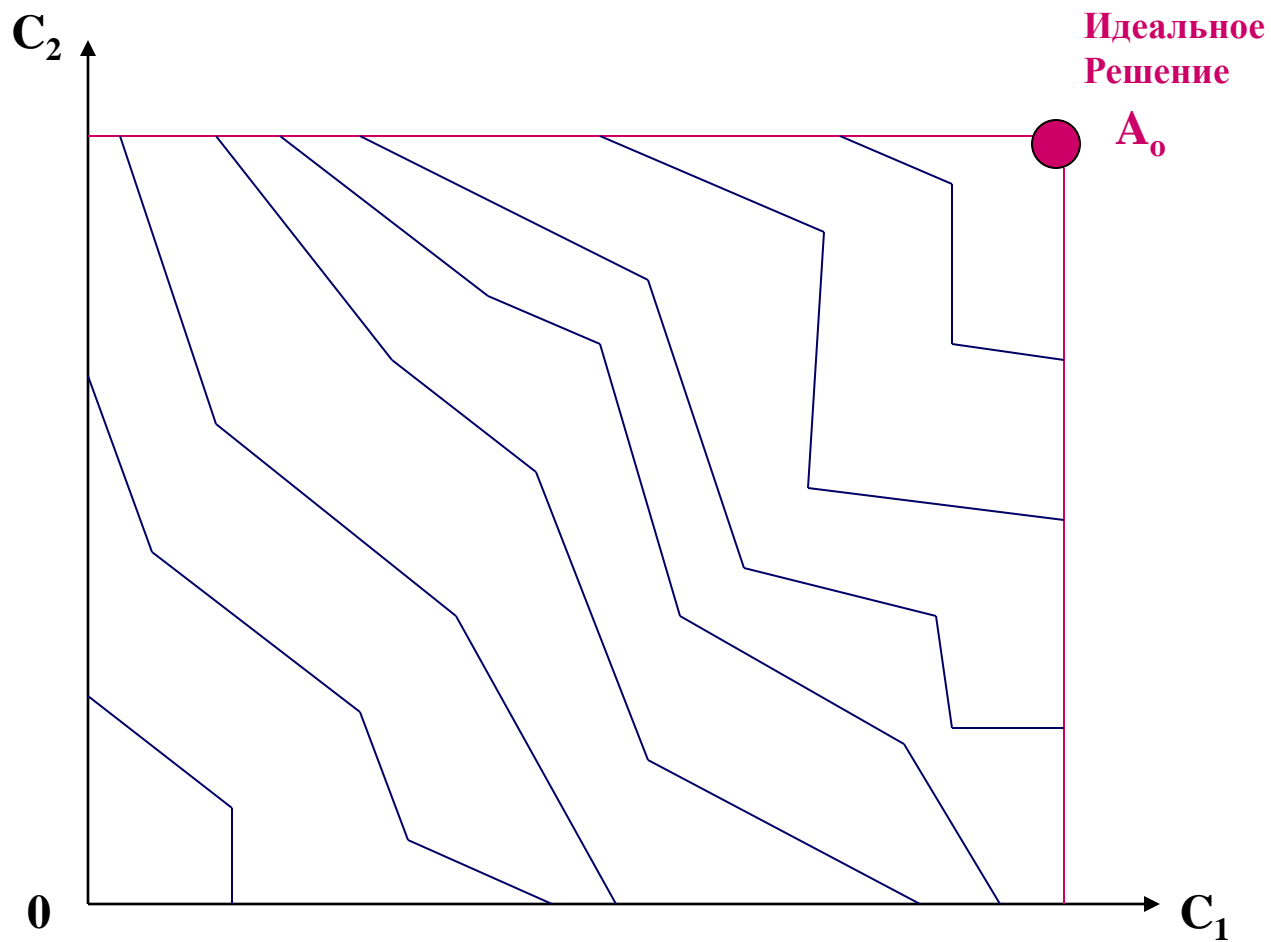


Иллюстрация для квази-квадратичной функции полезности: уровни несравнимости



Иллюстративный пример для “сложной” ситуации уровней несравнимости



Альтернативы $A=(A_1, \dots, A_i, \dots, A_n)$ и критерии $C=(C_1, \dots, C_j, \dots, C_k)$,
 $\forall A_i$ вектор оценок $z_i = (z_{i1}, \dots, z_{ij}, \dots, z_{ik})$, μ_j вес для критерия j

\forall пары $A_u, A_v \in A$ вычислить:

Коэффициент “согласия”

$$\alpha_{uv} = (1 / \sum_{j=1}^k \mu_j) \sum_{(j \in X(u, v))} \mu_j$$

Коэффициент “несогласия”

$$\beta_{uv} = 0 \quad \text{если } |Y(u, v)| = 0 \quad \text{иначе}$$
$$\max_j ((\mu_j |z_{uj} - z_{vj}|) / (d_j \sum_{j=1}^k \mu_j))$$

$X(uv) = \{j \mid z_{uj} \geq z_{vj}\}$, $Y(uv) = \{j \mid z_{uj} < z_{vj}\}$, d_j размер шкалы

ПРАВИЛО: A_u лучше A_v если $(\alpha_{uv} \geq p) \& (\beta_{uv} \leq q)$

где p, q - пороги (например, $p = 0.9$ и $q = 0.2$)

Иллюстративный численный пример для многокритериального ранжирования

	C_1 0.1	C_2 0.1	C_3 0.15	C_4 0.4	C_5 0.25	критерии {j} веса μ_j
A_1	10	8	8	10	4	
A_2	1	9	7	5	3	
A_3	0	9	10	6	1	
A_4	10	2	14	3	2	
A_5	7	7	5	8	3	
d_j	11	8	10	8	4	

$$u = 1, v = 3$$



$$X(1,3) = \{ 1,4,5 \}$$

$$Y(1,3) = \{ 2,3 \}$$

$$\alpha_{13} = (1 / 1) (0.1 + 0.4 + 0.25) = 0.75$$

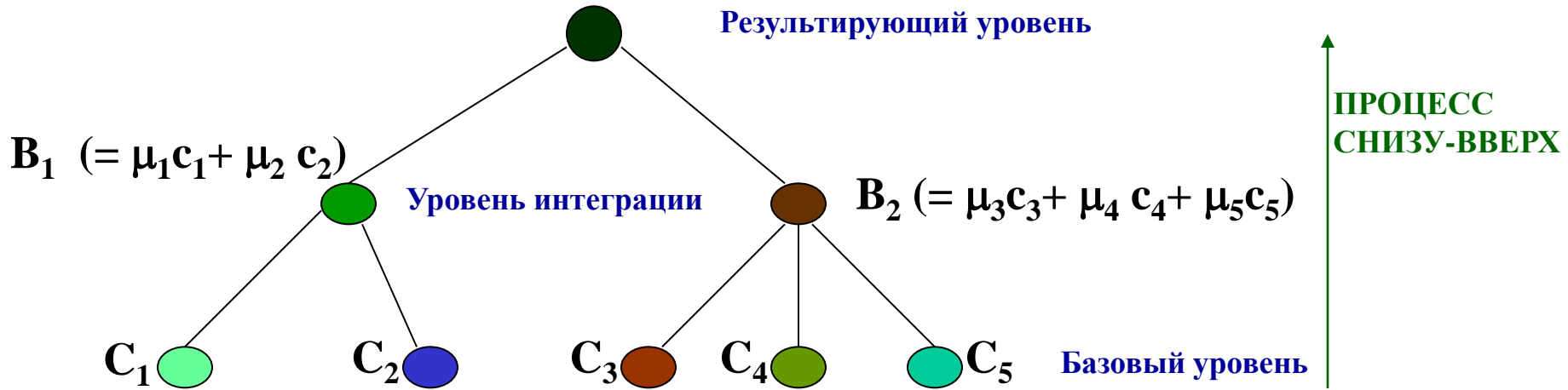
$$\beta_{13} = \max \{ (0.1 (9-8) / 8), (0.15 (10 - 8) / 10) \} = \max \{ 0.125, 0.03 \} = 0.125$$

Версия результата 1: $p = 0.7$ $q = 0.3 \Rightarrow A_1$ лучше A_3

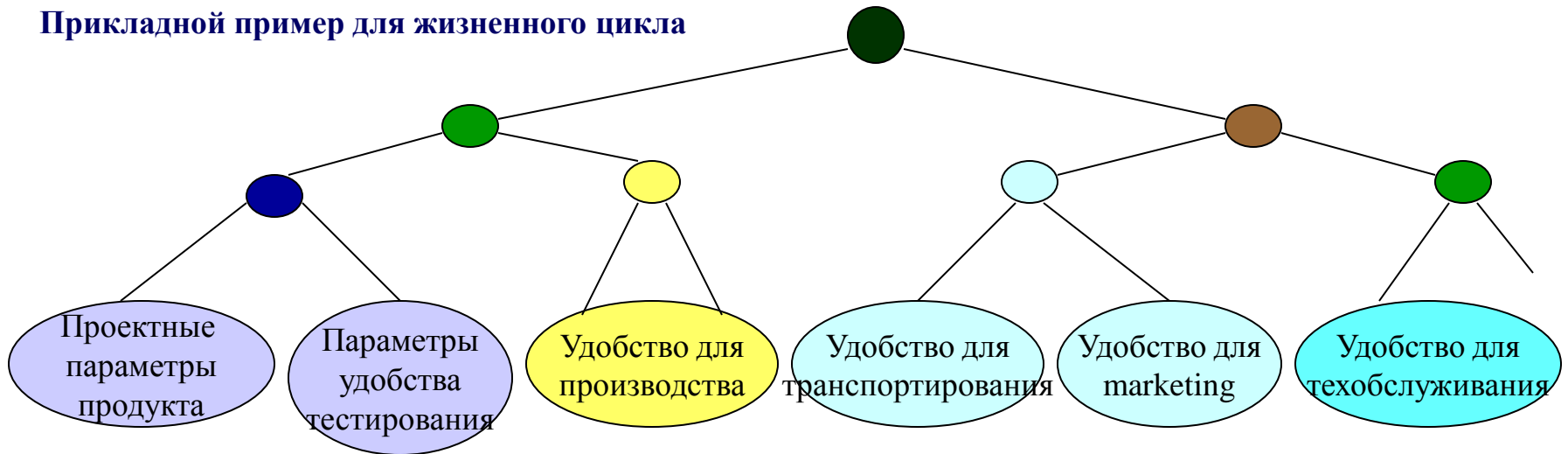
Версия результата 2: $p = 0.8$ $q = 0.2 \Rightarrow$ несравнимы

Метод аналитических иерархий АНП (Т.Л. Saaty)

$$J^* (= \lambda_1 b_1 + \lambda_2 b_2)$$

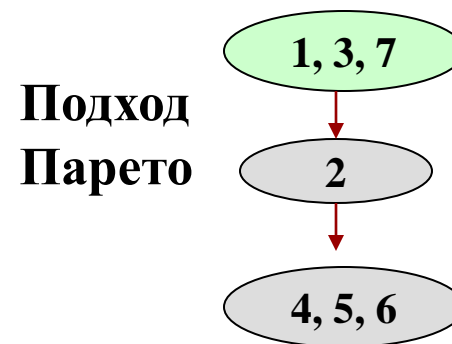
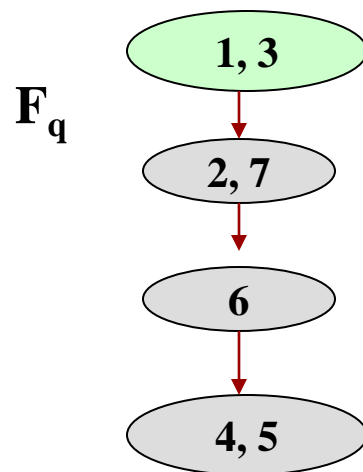
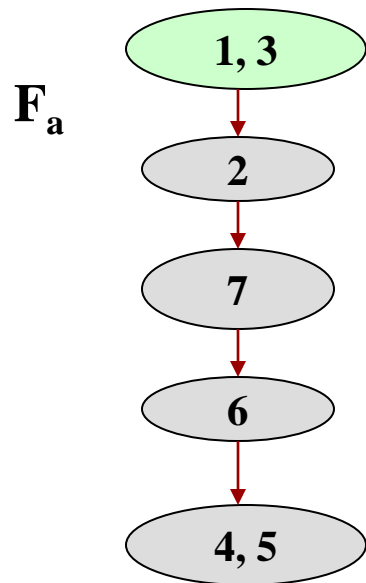


Прикладной пример для жизненного цикла

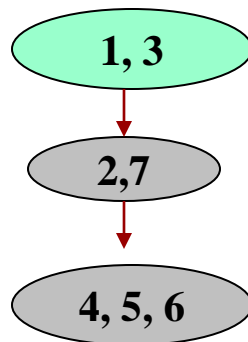


Интегрирование(агрегирование) результатов

Предыдущий пример:



Интуитивная интеграция:



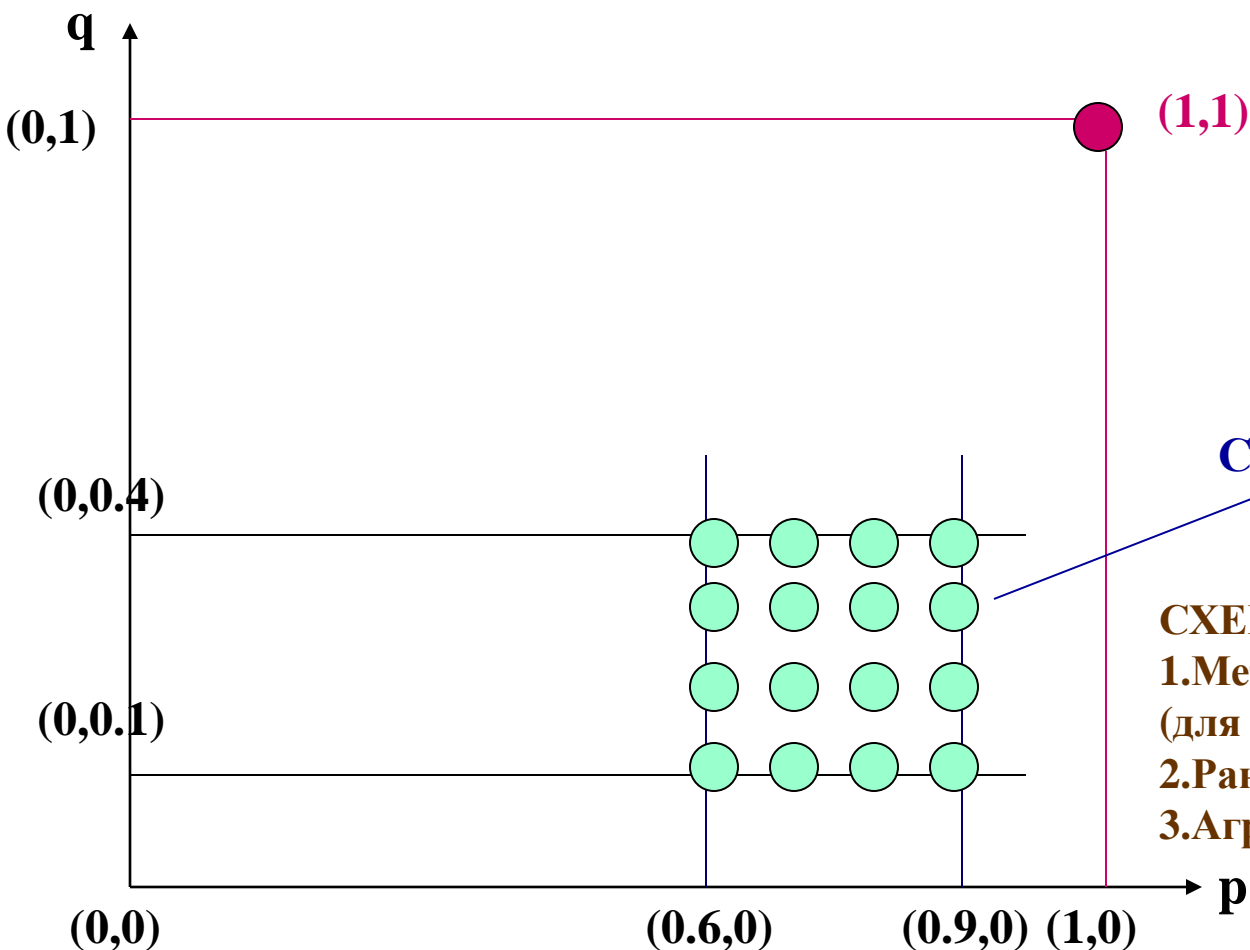
1.Правила типа «голосования»

2.Правила типа «голосования» & удаление “маргинальных результатов”

3.Многокритериальные методы, рассмотренные выше

4.Функции принадлежности (как в размытых множествах)

Подход интегрирования: Пример использования ELECTRE (М.Ш. ЛЕВИН, DSS COMBI)



Сетка порогов

СХЕМА РЕШЕНИЯ:

1.Метод ELECTRE

(для каждой пары порогов)

2.Ранжирование (получение уровней)

3.Агрегирование результатов

ЛЕКЦИИ 11-12. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Макр Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

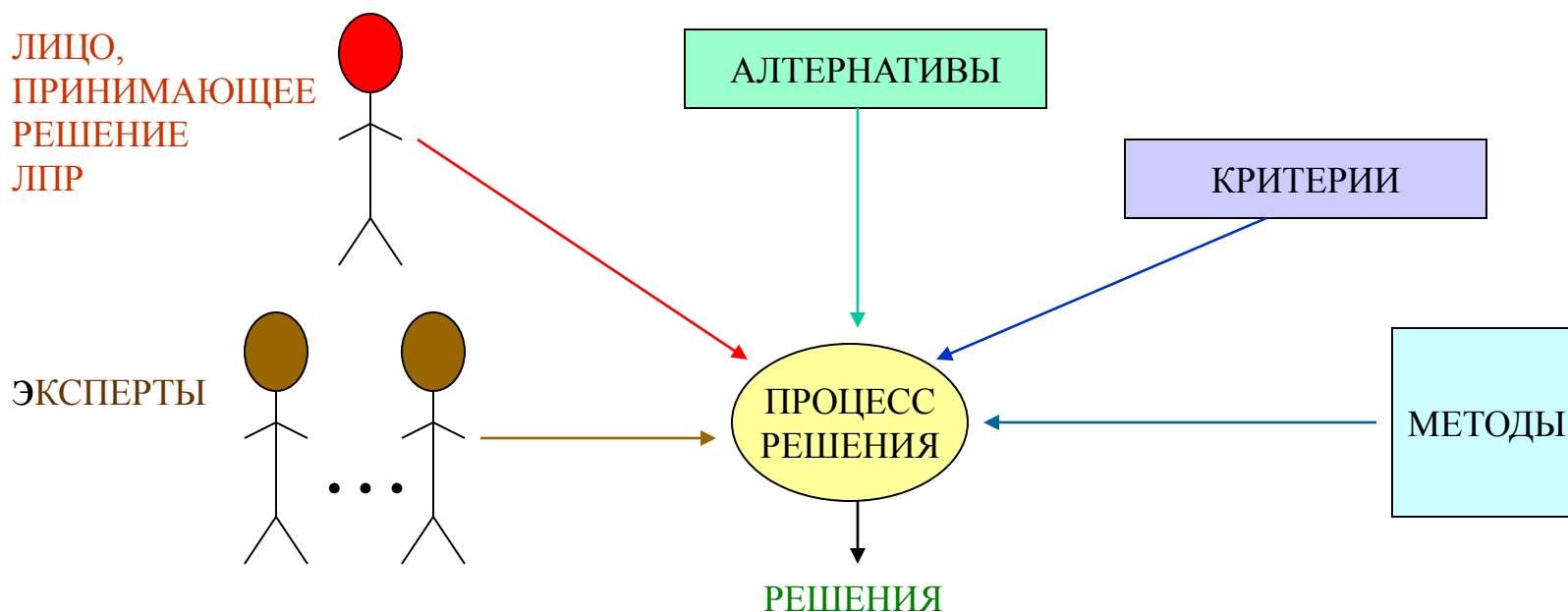
Л.11. Схема принятия решений, примеры

Л.12. Функция, отображение. Модели оптимизации

ПЛАН:

- 1.Схема многокритериального принятия решений
- 2.Разбиение процедуры многокритериального принятия решений
- 3.Числовые примеры: *разбиение (декомпозиция) исходной задачи *агрегирование результатов
- 4.Отображение

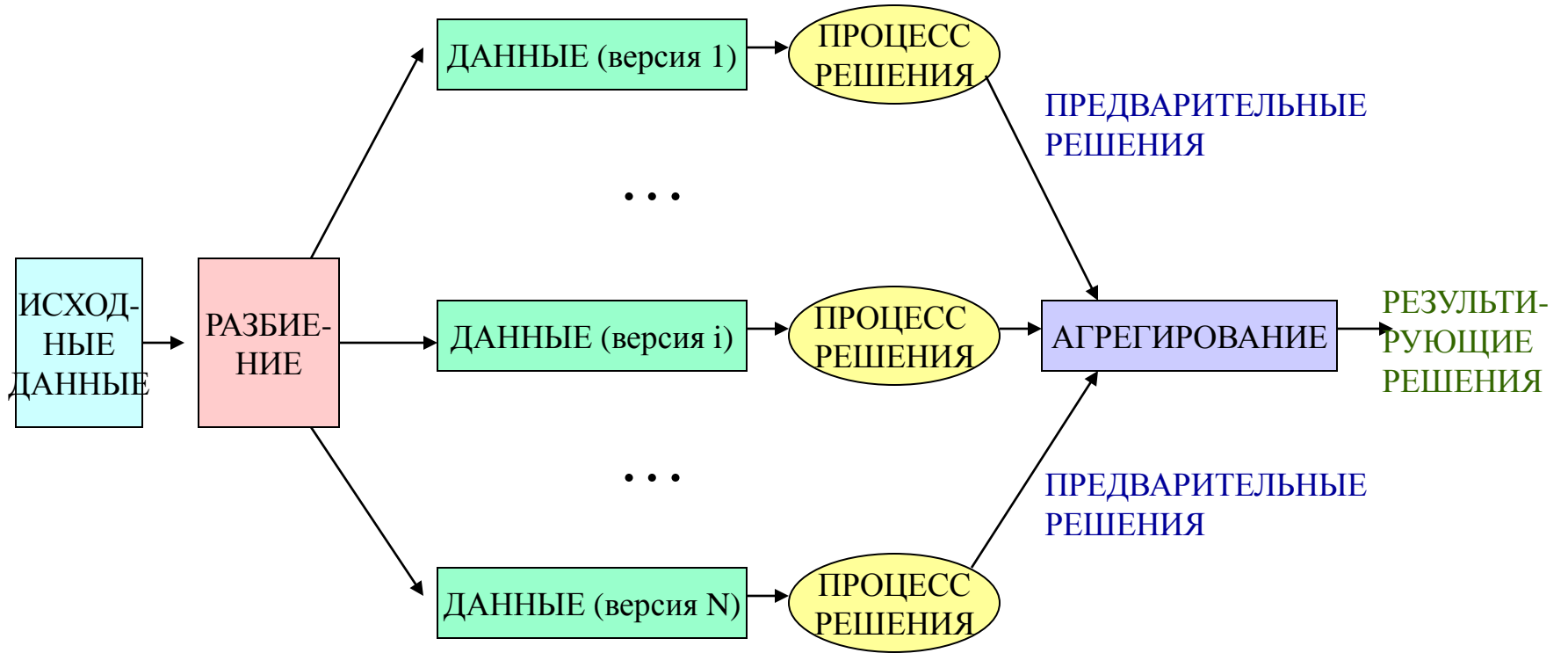
Схема многокритериального принятия решений & ее разбиение (распараллеливание)



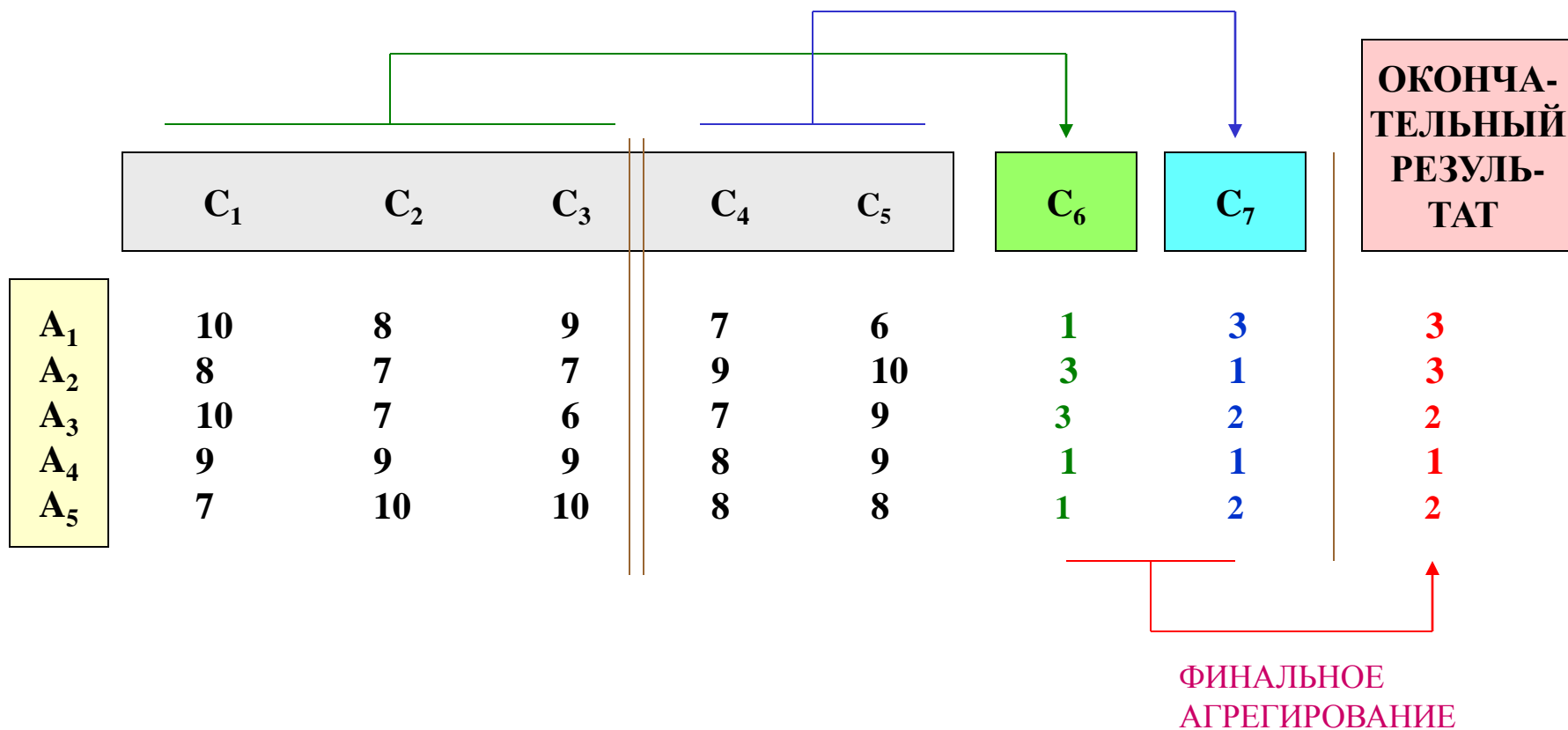
ПОДХОДЫ К РАЗБИЕНИЮ (ДЕКОМПОЗИЦИИ) СХЕМЫ ПРИНЯТИЯ РЕШЕНИЙ:

1. По критериям
2. По альтернативам
3. По экспертам
4. По методам
5. Гибридные подходы

Параллельная схема обработки



Пример: Разбиение по группам критериев



Пример: Разбиение по группам альтернатив

ПРЕДВАРИТЕЛЬНАЯ СТАДИЯ

	C ₁	C ₂	C ₃	1 st шаг
A ₁	10	8	9	1
A ₂	8	7	7	3
A ₃	10	7	6	3
A ₄	9	9	9	1
A ₅	7	10	10	1
A ₆	10	10	6	1
A ₇	6	8	9	2
A ₈	7	7	9	3
A ₉	9	8	9	1

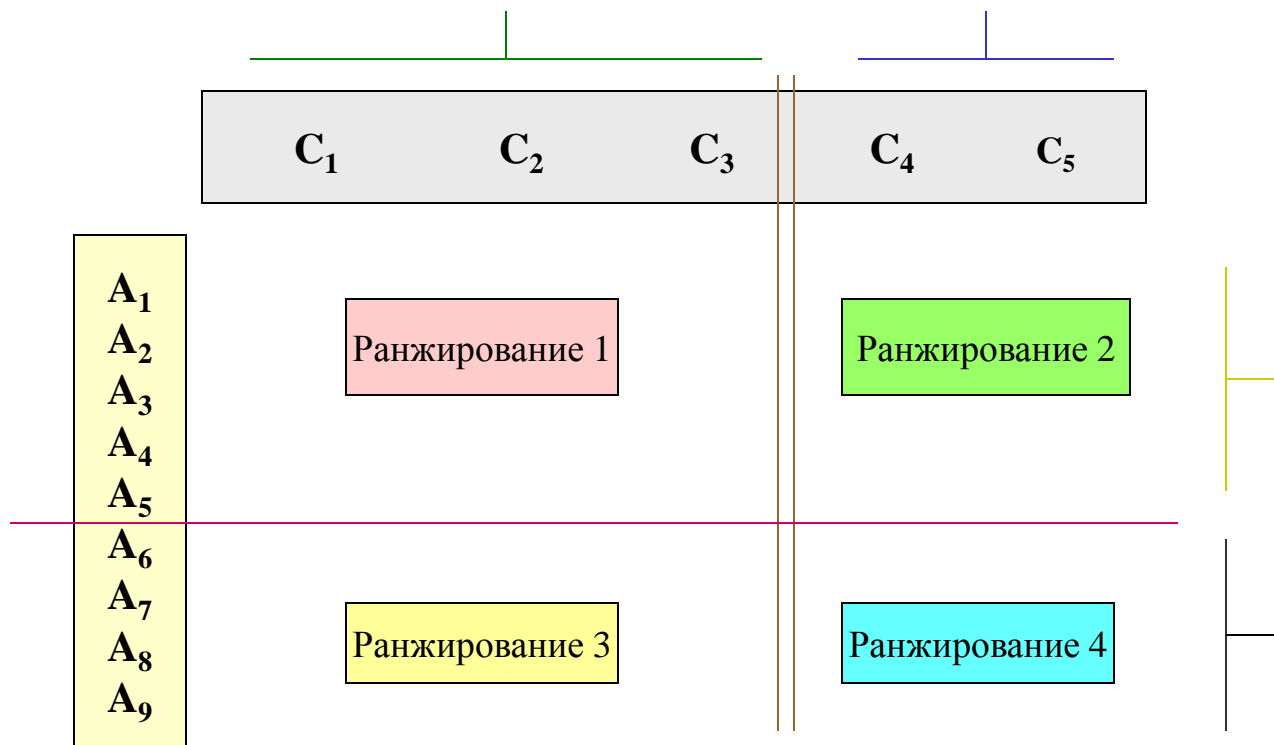
ФИНАЛЬНАЯ СТАДИЯ

C ₁	C ₂	C ₃		
A ₁	10	8	9	2
A ₄	9	9	9	3
A ₅	7	10	10	1
A ₆	10	10	6	1
A ₉	9	8	9	4

Пример: совместное разбиение по группам критериев & группам альтернатив

	C_1	C_2	C_3	C_4	C_5
A_1	10	8	9	7	6
A_2	8	7	7	9	10
A_3	10	7	6	7	9
A_4	9	9	9	8	9
A_5	7	10	10	8	8
A_6	10	10	6	7	9
A_7	6	8	9	10	8
A_8	7	7	9	10	10
A_9	9	8	9	9	9

Возможные схемы для совместного разбиения по группам критериев & группам альтернатив



Возможные схемы для совместного разбиения по группам критериев & группам альтернатив

СХЕМА 1

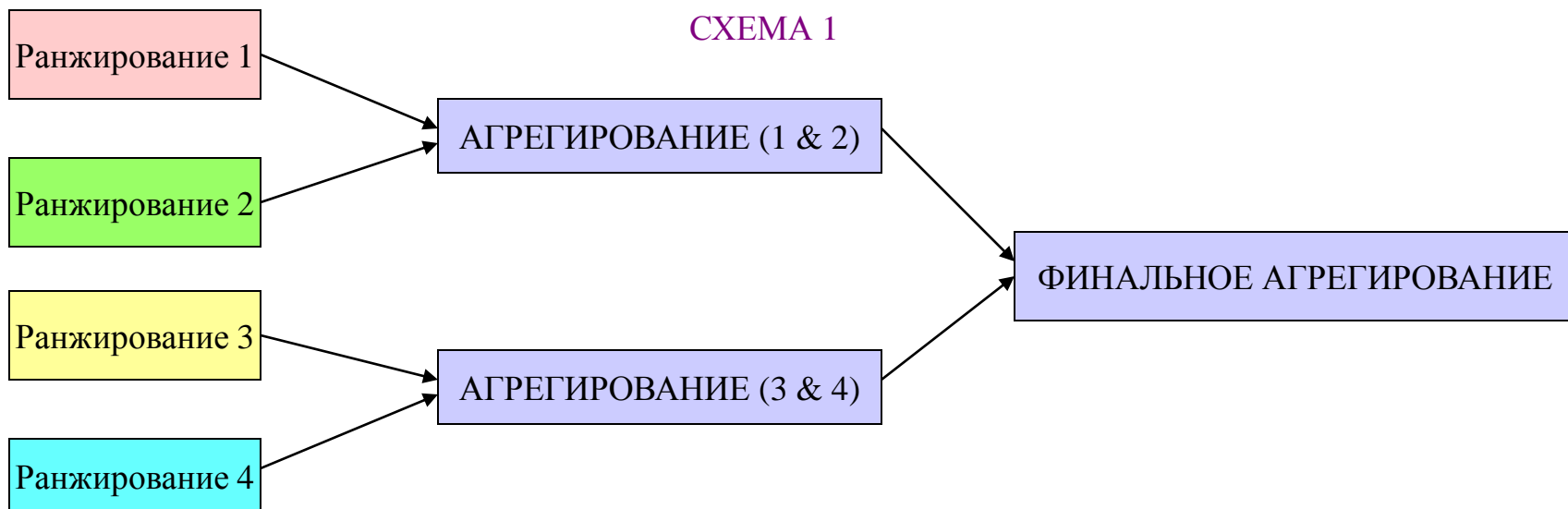
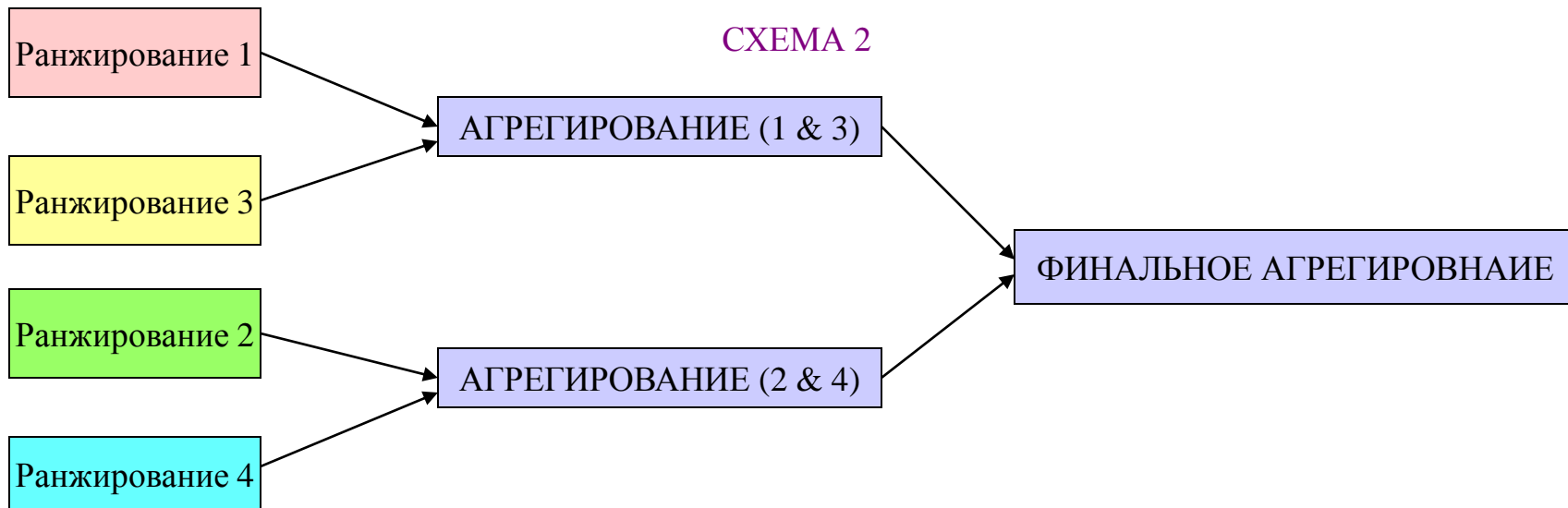
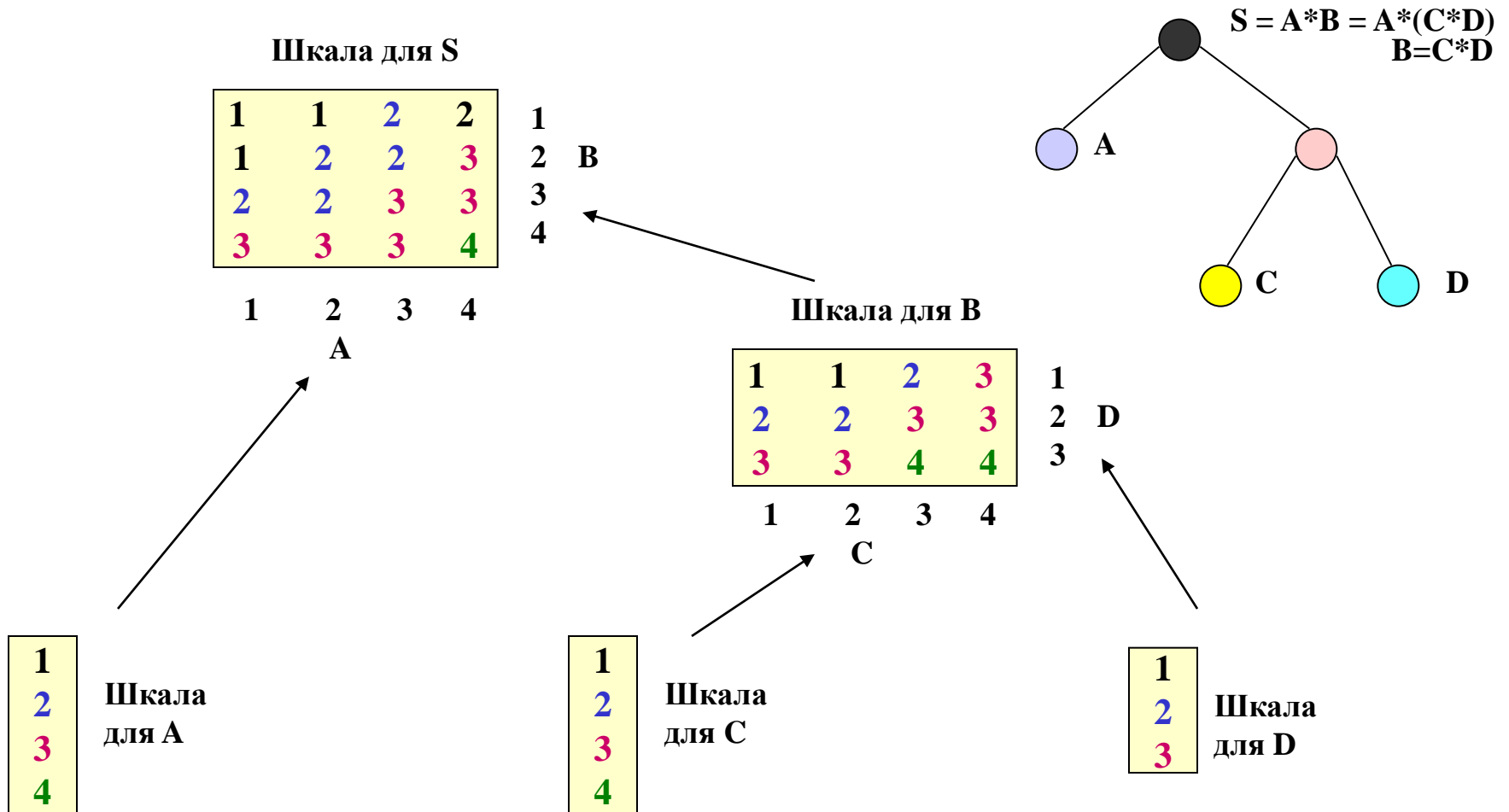


СХЕМА 2



Пример: Интегрирующие таблицы (Готов & Павелъев)

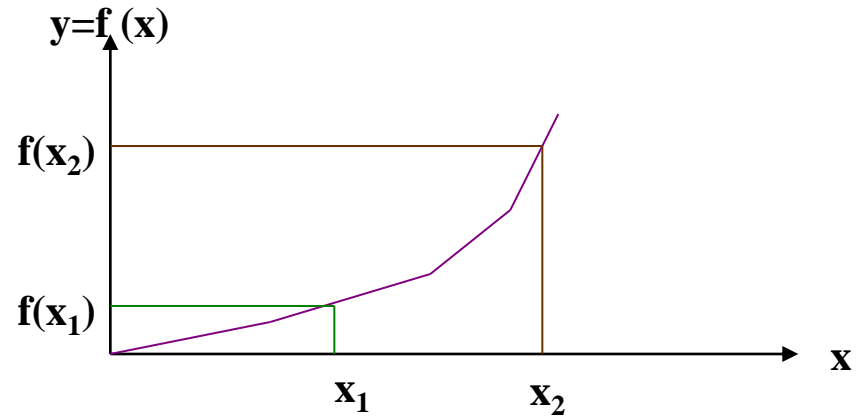


ПРИМЕР: Базовые оценки следующие: **4** для A, **3** для C, **1** для D;
 промежуточные оценки для B - **2**;
 результирующая оценка для S - **3**.

ЗАМЕЧАНИЕ: много-размерные интегрирующие таблицы возможны (и полезны) тоже .

Отображение & Оптимизационные Модели

$x \in \mathbf{R}$



$x \in X \subseteq \mathbf{R}$



ОТОБРАЖЕНИЕ $y=f(x)$
 $X \Rightarrow Y$



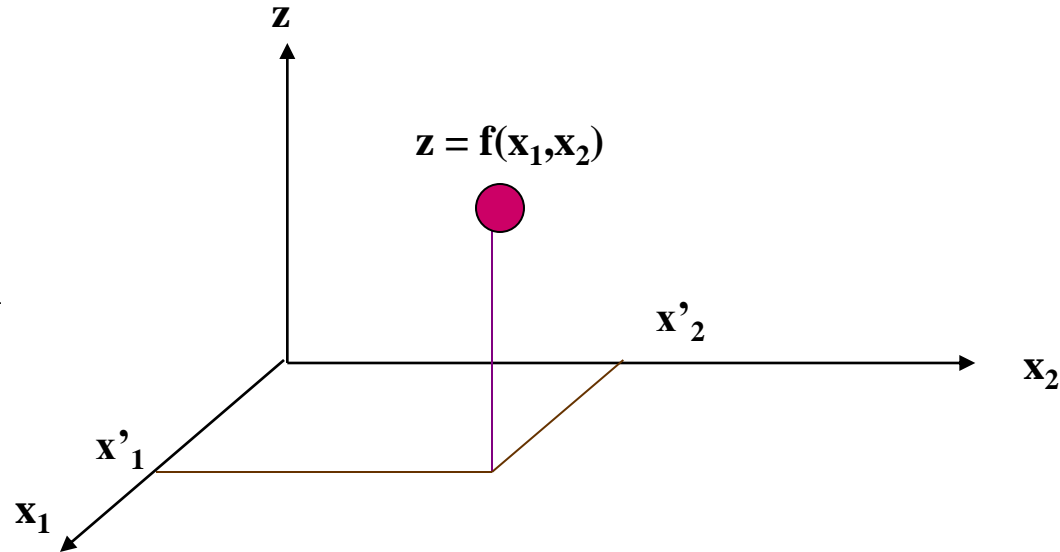
$y \in Y \subseteq \mathbf{R}$

Отображение & Оптимизационные Модели

$$z = f(x_1, x_2) \in \mathbb{R}^2$$

$$(x_1, x_2) \in X \subseteq \mathbb{R}^2$$

$$f(x_1, x_2) \Rightarrow Z \subseteq \mathbb{R}$$



ОБЩИЙ СЛУЧАЙ:

$$(x_1, \dots, x_m) \in X \subseteq \mathbb{R}^m$$

$$X \xrightarrow{\text{ОТОБРАЖЕНИЕ}} Y$$

$$(y_1, \dots, y_n) \in Y \subseteq \mathbb{R}^n$$

Отображение & Оптимизационные Модели

$$\mathbf{x} \in \mathbf{X} \subseteq \mathbf{R}$$

$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2]$ (допустимая область)

$f(\mathbf{x})$ - целевая функция

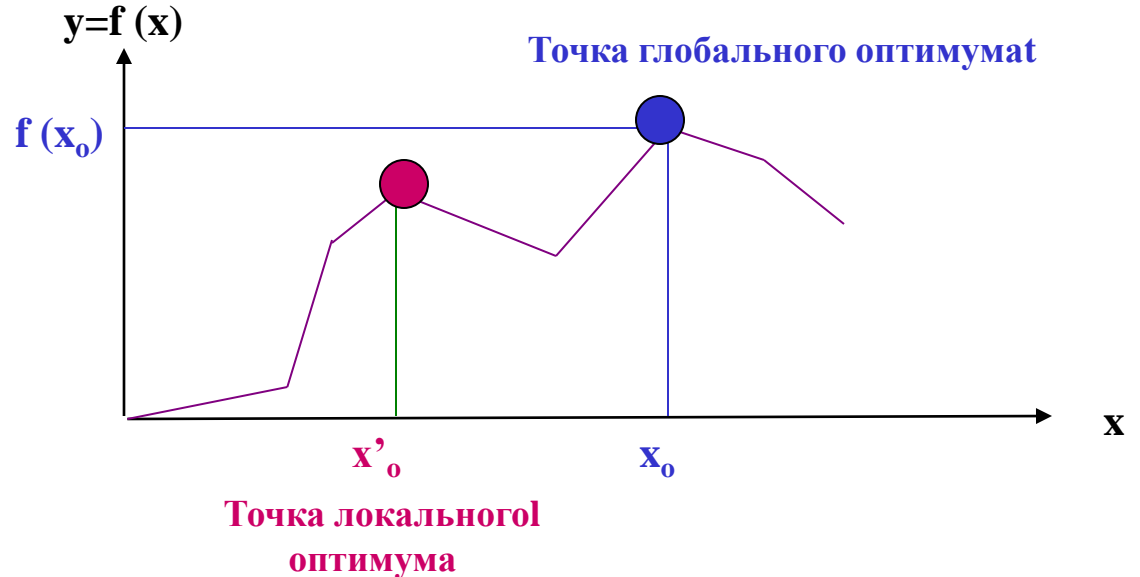
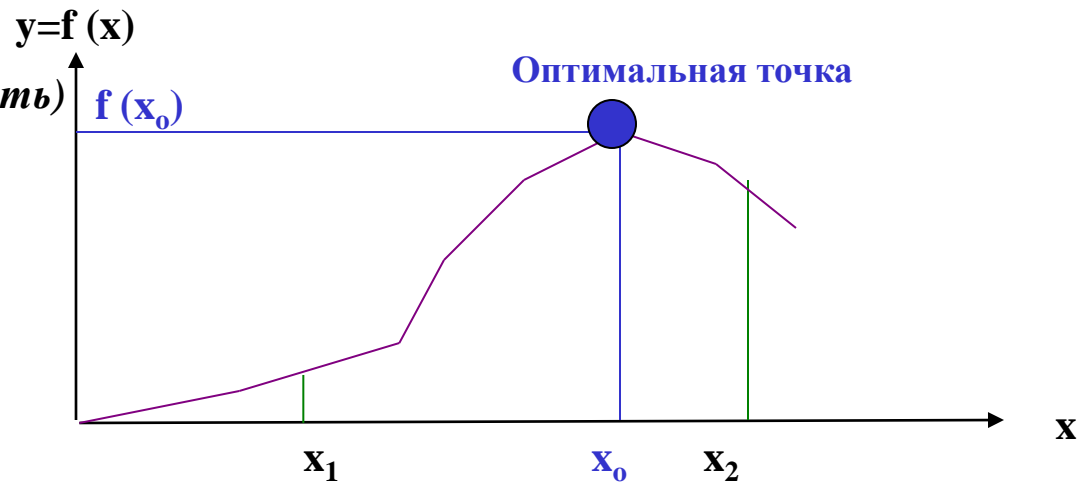
$$\begin{array}{l} \max f(\mathbf{x}) \\ \text{subject to} \\ \mathbf{x} \in \mathbf{X} \end{array}$$

$$\begin{array}{l} \max f(\mathbf{x}) \\ \text{subject to} \\ \mathbf{x} \geq \mathbf{x}_1 \\ \mathbf{x} \leq \mathbf{x}_2 \end{array}$$

ОБЩИЙ СЛУЧАЙ:

$$\begin{array}{l} \max f(\mathbf{x}) \\ \text{subject to} \\ \varphi_1(\mathbf{x}) \leq 0 \\ \dots \\ \varphi_k(\mathbf{x}) \leq 0 \end{array}$$

$\varphi_j(\mathbf{x})$ - функция ограничения ($1 \leq j \leq k$)



Иллюстрирующая таблица для оптимизационных моделей

Функция цели $f(x)$	Ограничение $\varphi_j(x)$	Тип модели	Метод
Линейная	Линейное	Линейная	*симплекс *метод эллипсоида *метод Кармаркара
Квадратичная	Линейное	Квадратичная	*симплекс *эллипсоид
Выпуклая	Линейное	Выпуклая	*метод градиента *метод эллипсоида
...

ЗАМЕЧАНИЯ:

- 1. Целевые функции могут рассматриваться как векторные (многокритериальная оптимизация)**
- 2. Ограничения могут рассматриваться и как бинарные отношения**
- 3. В дискретной оптимизации рассматриваются дискретные пространства**
- 4. В моделях стохастической оптимизации параметры / функции являются стохастическими**
- 5. Возможно учитывать неопределенность:**
 - *стохастические параметры / функции**
 - *параметры / функции на основе размытых множеств**

ЛЕКЦИЯ 13. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.13. Основные модели комбинаторной оптимизации I.

ПЛАН:

1. Базовые задачи комбинаторной оптимизации:

*задача о рюкзаке, *схемы решения для многокритериальной задачи о рюкзаке, *блочная задача о рюкзаке.

2. алгоритмы:

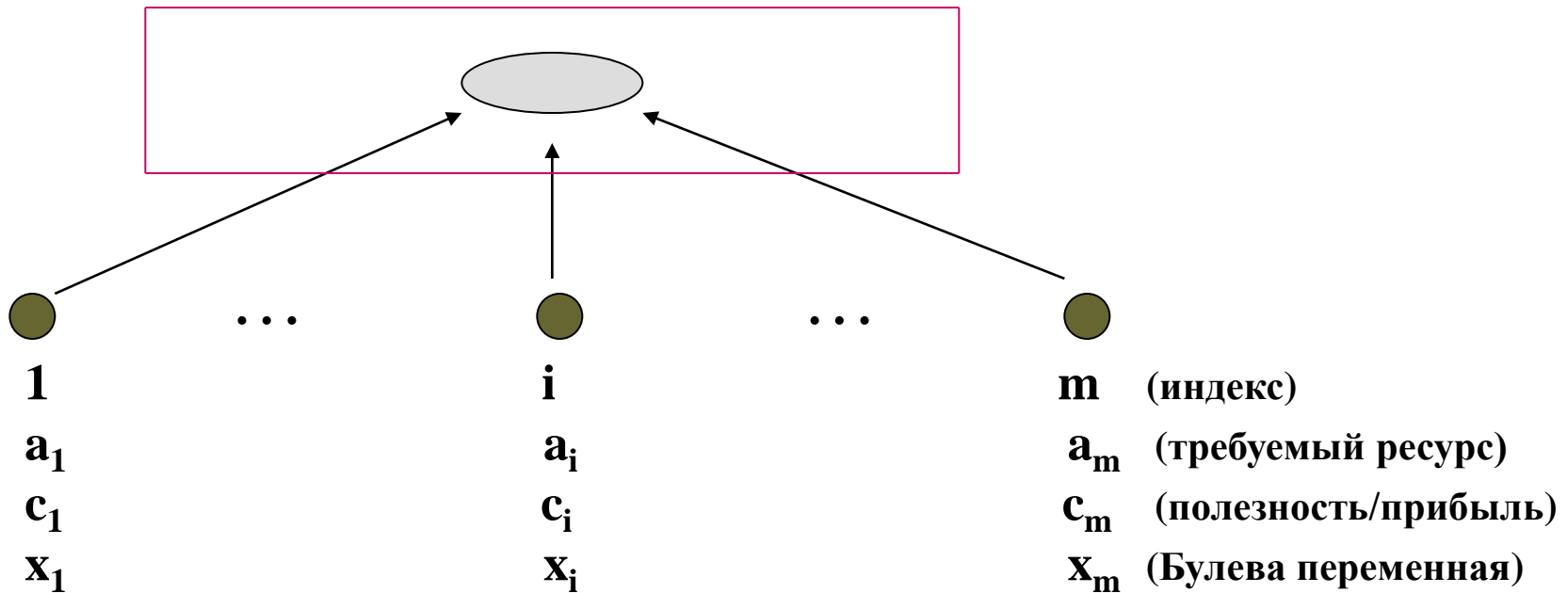
*типы решений (точные, приближенные), *типы алгоритмов (полиномиальные и переборные алгоритмы)

3. Сложность задач.

4. Глобальные подходы и локальные приемы

Окт. 1, 2004

Задача о рюкзаке



$$\begin{aligned} \max \quad & \sum_{i=1}^m c_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^m a_i x_i \leq b \end{aligned}$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, m$$

Возможные дополнительные ограничения

$$\sum_{i=1}^m a_{ik} x_i \leq b_k, \quad k = 1, \dots, l$$

- 1. Упорядочение по невозрастанию c_i / a_i (алгоритм Данцига, эвристика)**
- 2. Метод ветвей и границ**
- 3. Динамическое программирование (точное решение)**
- 4. Динамическое программирование (схема приближенного решения)**
- 5. Вероятностные методы**
- 6. Гибридные схемы**

1. $c_i = c_0$ (*равные полезности*)
2. $a_i = a_0$ (*равные требуемые ресурсы*)

Полиномиальный алгоритм:

1. *Упорядочение по неубыванию a_i*
2. *Упорядочение по невозростанию c_i*

«Расширенные» версии задачи о рюкзаке

1. Задача о рюкзаке с целевой функцией на *min*
2. Задача о рюкзаке с несколькими “рюкзаками”
3. Задача о рюкзаке с дополнительными структурными (логическими) ограничениями на элементах (например, различные виды деревьев)
4. Многокритериальная задача о рюкзаке
5. Задача о рюкзаке с «размытыми» параметрами

Эвристическая схема для многокритериальной версии задачи о рюкзаке

АЛГОРИТМИЧЕСКАЯ СХЕМА (случай линейного ранжирования):

ШАГ 1. Многокритериальное ранжирование элементов
(получить линейное ранжирование)

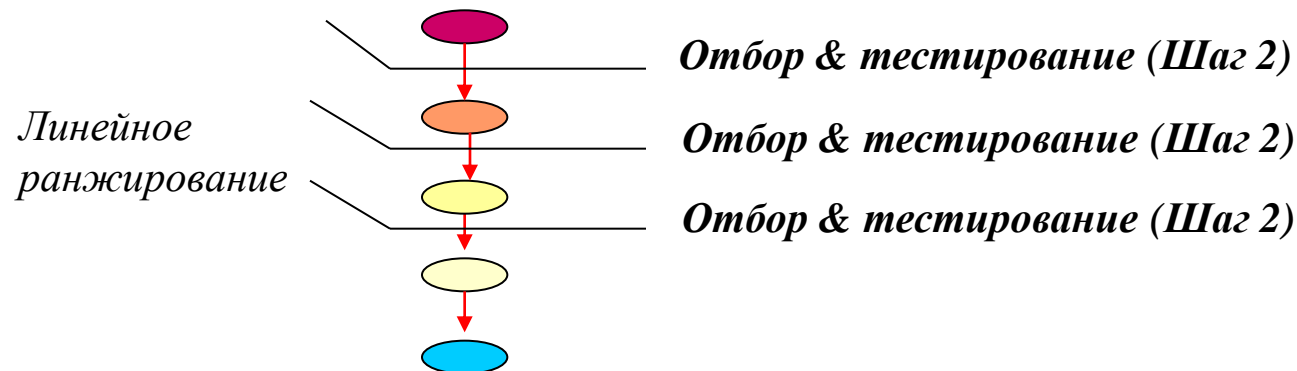
ШАГ 2. Последовательный отбор элементов
(лучший элемент, следующий элемент, и т.д.)

После каждого отбора: тестирование ограничения по ресурсу ($\leq b$).

Если ограничение не выполняется, то необходимо исключить последний отобранный элемент и СТОП.

Иначе: ШАГ 2.

СТОП.



Эвристическая схема для многокритериальной версии задачи о рюкзаке

АЛГОРИТМИЧЕСКАЯ СХЕМА (случай группового ранжирования):

ШАГ 1. Многокритериальное ранжирование элементов
(получить групповое ранжирование)

ШАГ 2. Последовательные отбор элементов

(элементы «лучшей» группы, элементы следующей группы и т.д.)

После каждого отбора: тестирование ограничения по ресурсу ($\leq b$).

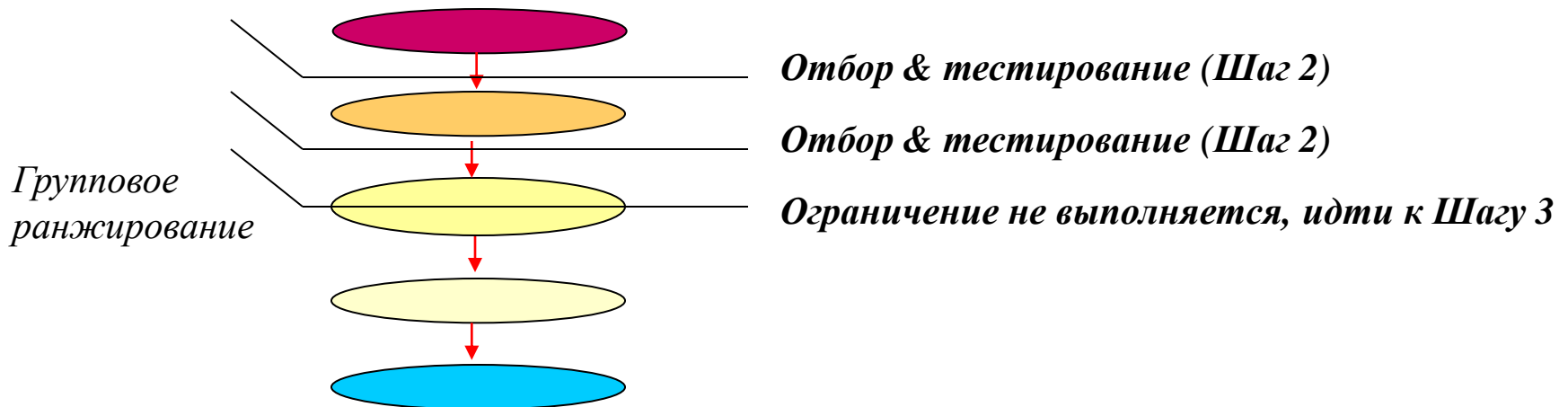
Если ограничение не выполняется, то ШАГ 3. Иначе: ШАГ 2.

ШАГ 3. Решение для последней анализируемой группы специального случая задачи о рюкзаке (с равными полезностями) как последовательный отбор элементов из списка (невозрастание по a_i).

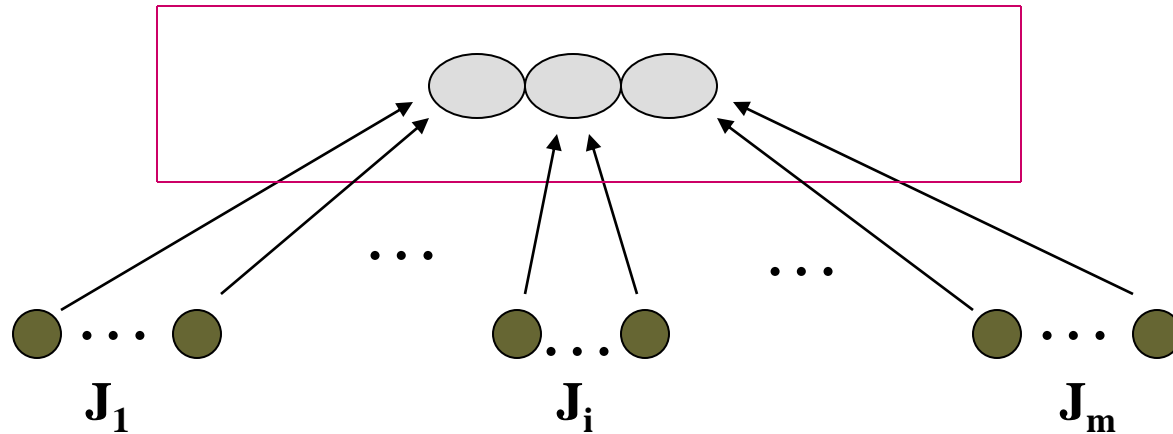
Здесь ограничение следующее: $\leq b - \sum_{(i \in Q)} a_i$

(где Q – множество отобранных элементов из предыдущих групп)

СТОП.



Блочная задача о рюкзаке



$$\forall i \quad |J_i| = q_i, \quad j = 1, \dots, q_i$$

$$\mathit{max} \quad \sum_{i=1}^m \sum_{j=1}^{q_i} c_{ij} x_{ij}$$

$$\mathit{s.t.} \quad \sum_{i=1}^m \sum_{j=1}^{q_i} a_{ij} x_{ij} \leq b$$

$$\sum_{j=1}^{q_i} x_{ij} \leq 1, \quad i = 1, \dots, m$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, q_i$$

- 1. Упорядочение по невозростанию of c_{ij} / a_{ij} (эвристика)*
- 2. Метод ветвей и границ*
- 3. Динамическое программирование (точное решение)*
- 4. Динамическое программирование (приближенная схема решения)*
- 5. Вероятностные методы*
- 6. Гибридные схемы*

Иллюстрация для динамического программирования

- Последовательное построение решения:
1. От точки НАЧАЛО к точке КОНЕЦ
 2. От точки КОНЕЦ к точке НАЧАЛО

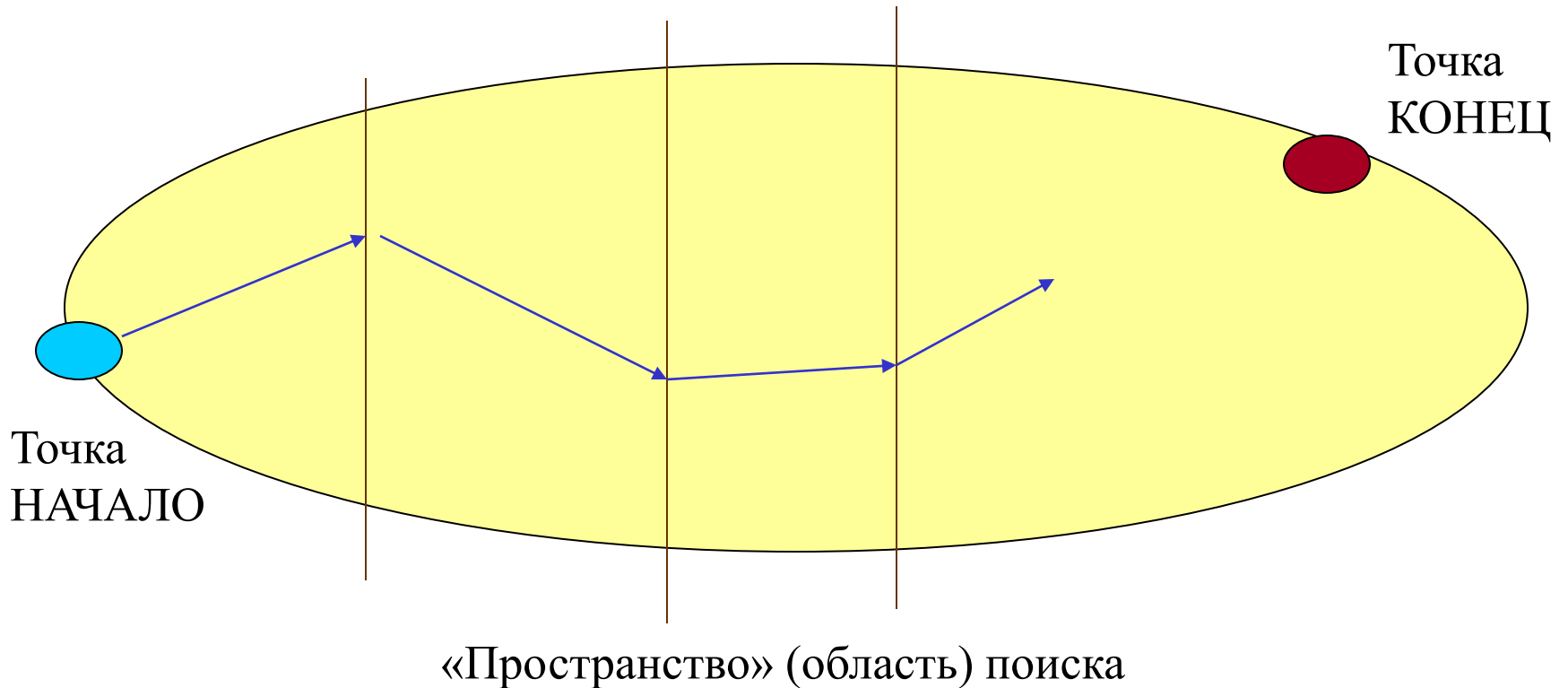
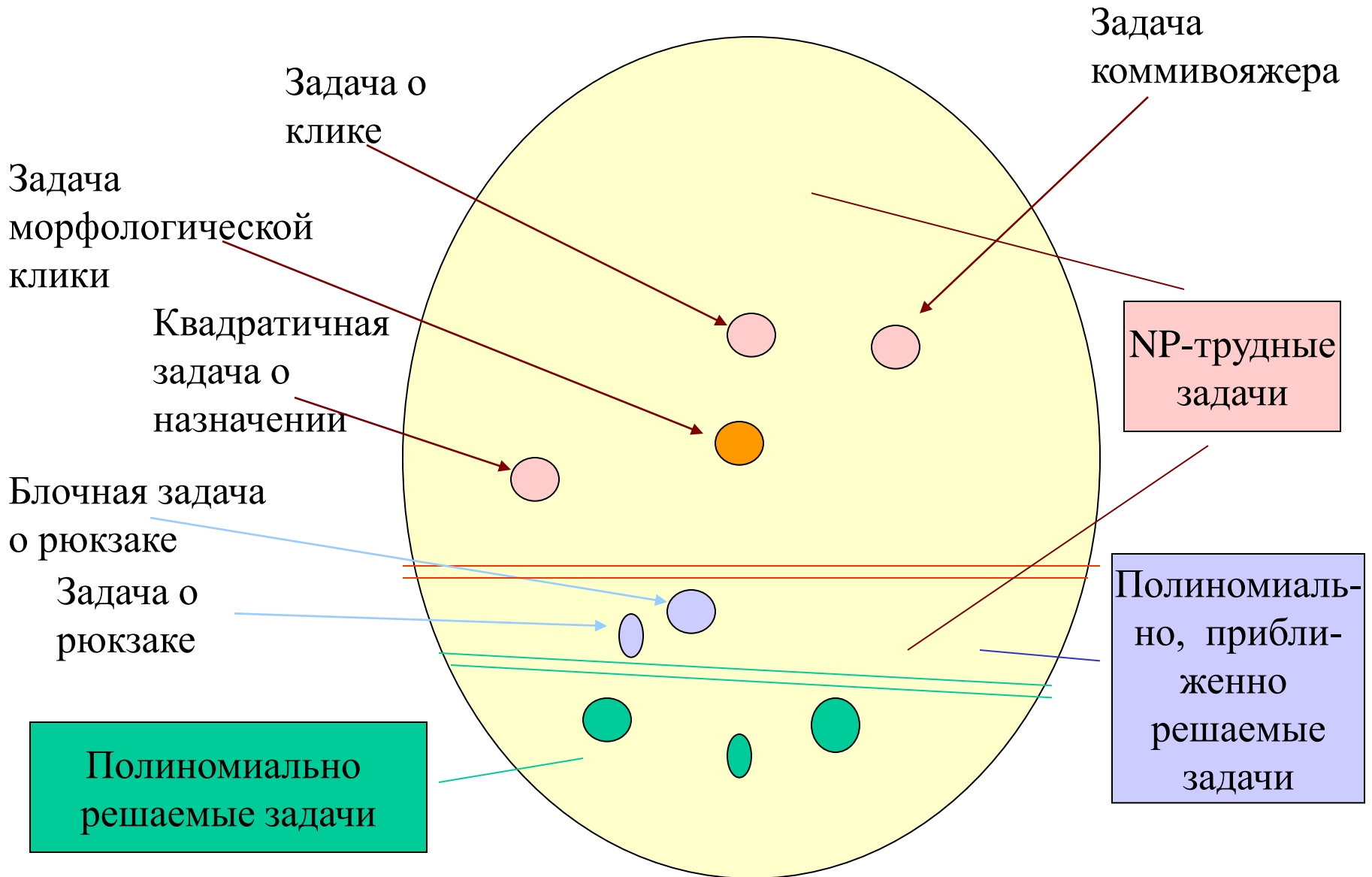


Иллюстрация сложности задач комбинаторной оптимизации



Классификация алгоритмов

ТОЧНОСТЬ РЕЗУЛЬТАТА (решение):

1. Точное решение
2. Приближенное решение (для наихудшего случая): *ограниченная ошибка (абсолютная), *ограниченная ошибка (относительная), *др.
3. Приближенное решение (статистически)
4. Эвристика (без оценок точности)

ПО СЛОЖНОСТИ ПРОЦЕССА РЕШЕНИЯ (например, число шагов):

1. Полиномиальные алгоритмы (по длине входа, например:
 $O(n \log n)$, $O(n)$, $O(1)$, $O(n^2)$)
2. Полиномиальные приближенные схемы (для заданной точности / ограниченной ошибки, например: $O(n^2/\varepsilon)$
где $\varepsilon \in [0,1]$ - относительная точность по целевой функции)
3. Статистически «хорошие» алгоритмы (статистически полиномиальные)
4. Переборные алгоритмы . . .

БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ РЕСУРСЫ:

1. Число шагов (вычислительные операции)
2. Требуемый объем памяти
3. Требуемое число взаимодействия со специалистом (оракулом)
(для получения дополнительной информации)
4. Требуемые коммуникации между процессорами
(для многопроцессорных алгоритмов)

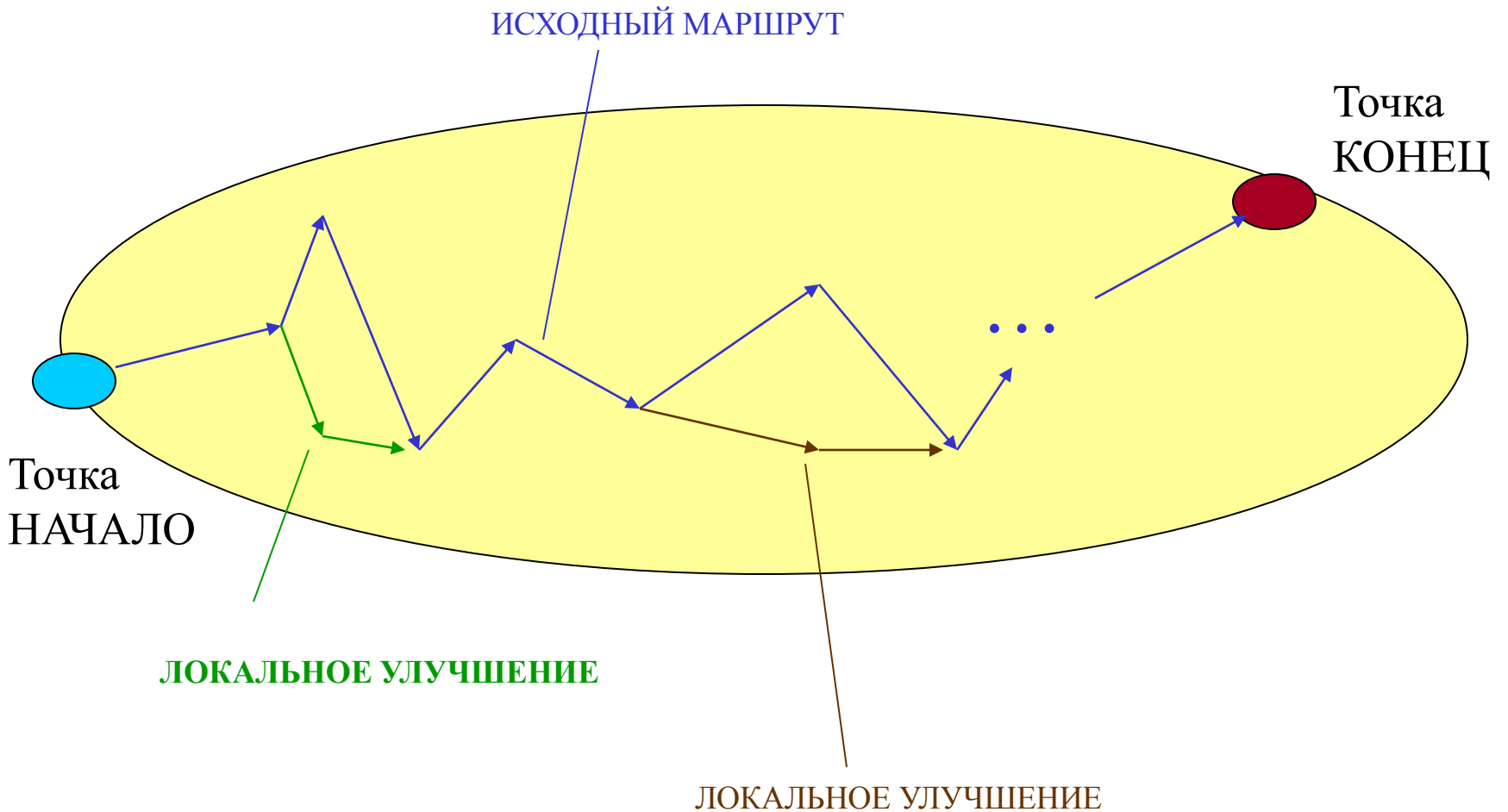
ГЛОБАЛЬНЫЕ ПОДХОДЫ:

- 1.Разбиение на подзадачи**
- 2.Декомпозиция (расширение «хорошего» локального решения и др.)**
(например: динамическое программирование, метод ветвей и границ)
- 3.Сеточный подход с удалением «плохих точек»**
- 4.Приближенные (аппроксимационные) подходы**
(т.е., аппроксимация исходной задачи или ее частей на основе более простой «конструкции»)

ЛОКАЛЬНЫЕ ПРИЕМЫ:

- 1.Локальная оптимизация как улучшение решения или его части**
- 2.Вероятностные шаги**
- 3. «Жадные» алгоритмы (выбор “простого” / “близкого” / и т.д.)**
- 4.Рекурсия**

Иллюстрация улучшения решения (локальная оптимизация)



ЛЕКЦИИ 14-15. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем управления, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.14. Основные модели комбинаторной оптимизации II

Л.15. Схема многокритериального проектирования PSI

ПЛАН:

1. Базовые задачи комбинаторной оптимизации:

*целочисленное нелинейное программирование (специальная постановка и пример),

*задачи упаковки & задача упаковки в контейнеры (иллюстрация),

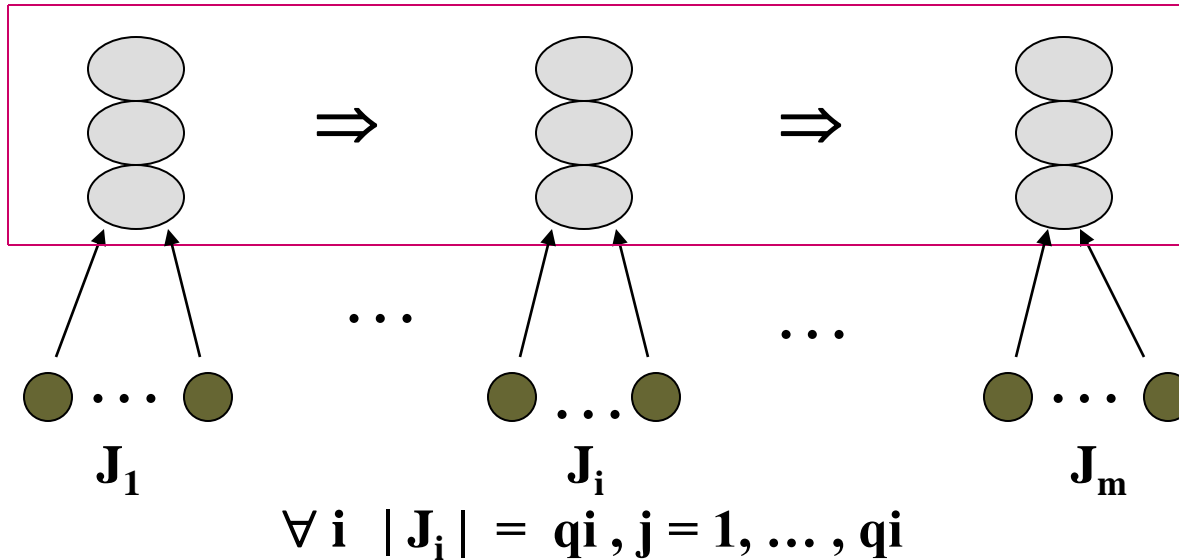
*задачи теории расписаний (задача и алгоритм для «сборки»), 3 примера для 1-процессорного расписания,

*задача максимальной клики (иллюстрация)

2. Схема многокритериального проектирования (PSI – исследование пространства параметров)

Окт. 2, 2004

Целочисленное нелинейное программирование (модульное проектирование последовательной системы с точки зрения надежности) (by Berman & Ashrafi)



$$\max \quad \prod_{i=1}^m (1 - \prod_{j=1}^{q_i} (1 - p_{ij} x_{ij}))$$

$$s.t. \quad \sum_{i=1}^m \sum_{j=1}^{q_i} d_{ij} x_{ij} \leq b$$

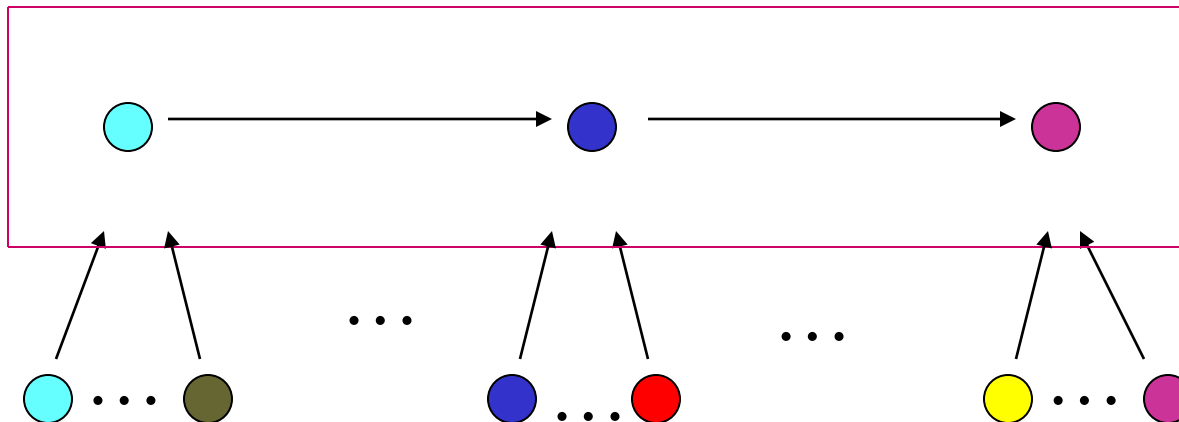
$$\sum_{j=1}^{q_i} x_{ij} \geq 1, \quad i = 1, \dots, m$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, q_i$$

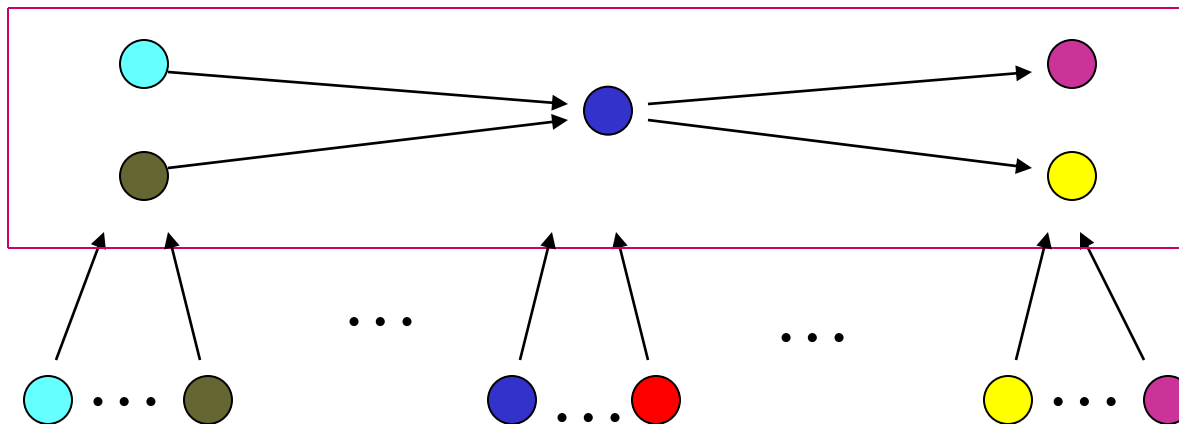
p_{ij} - надежность, d_{ij} - стоимость

Целочисленное нелинейное программирование (модульное проектирование последовательной системы с точки зрения надежности) (by Berman & Ashrafi)

ПРИМЕР 1
(последовательная схема)



ПРИМЕР 2
(параллельно-последовательная схема)



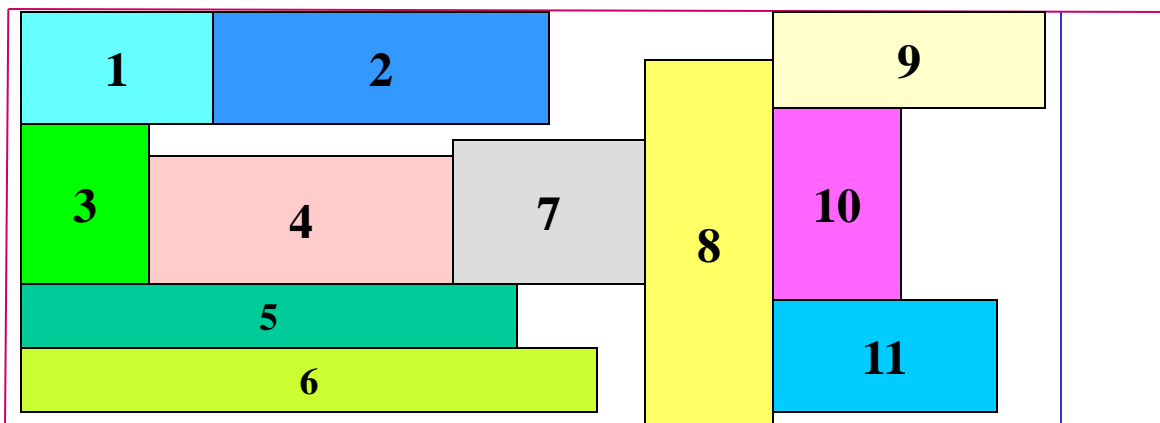
1.Метод ветвей и границ

2.Динамическое программирование

4.Эвристики (например, сведение задачи к непрерывной)

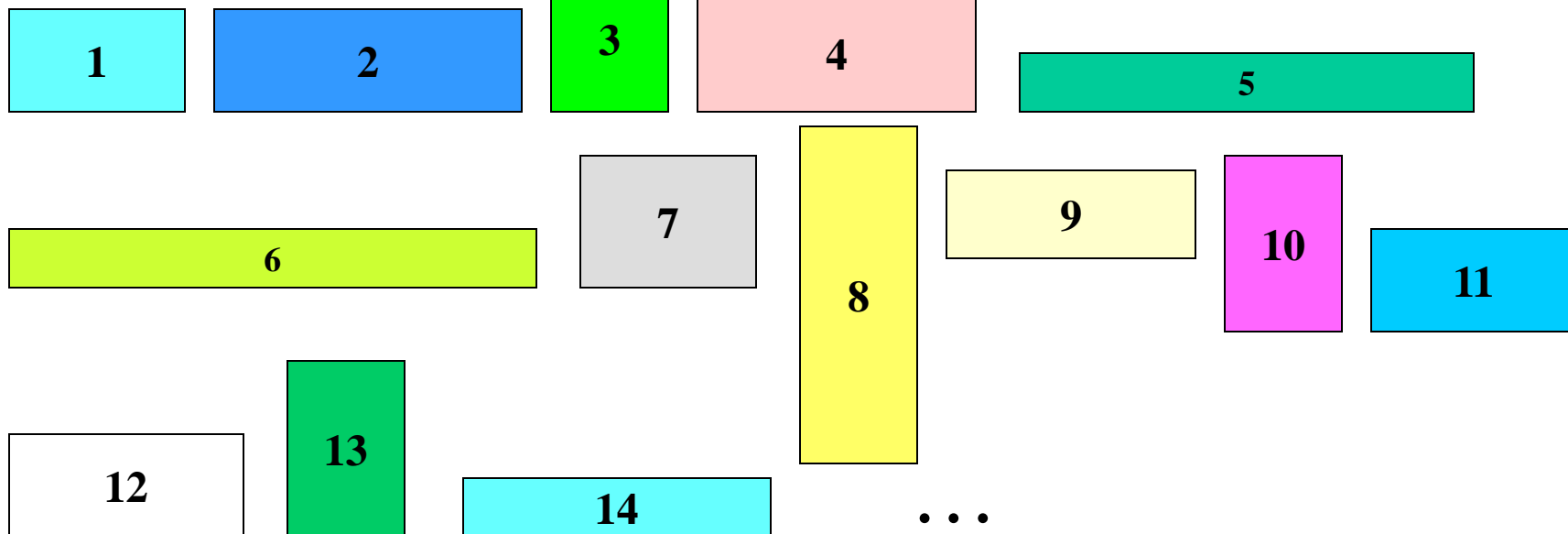
Задача упаковки (иллюстрация)

ОБЛАСТЬ
УПАКОВКИ



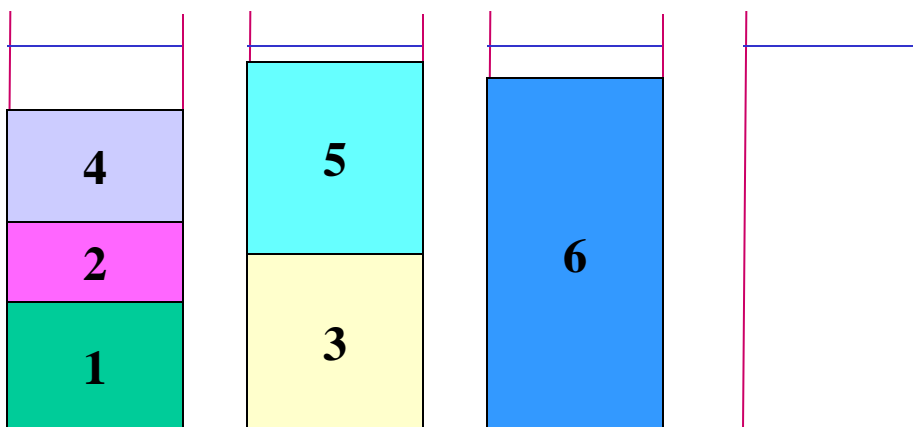
ЦЕЛИ:
*Максимум
упакованных
элементов
*Минимум
свободного
пространства

ЭЛЕМЕНТЫ



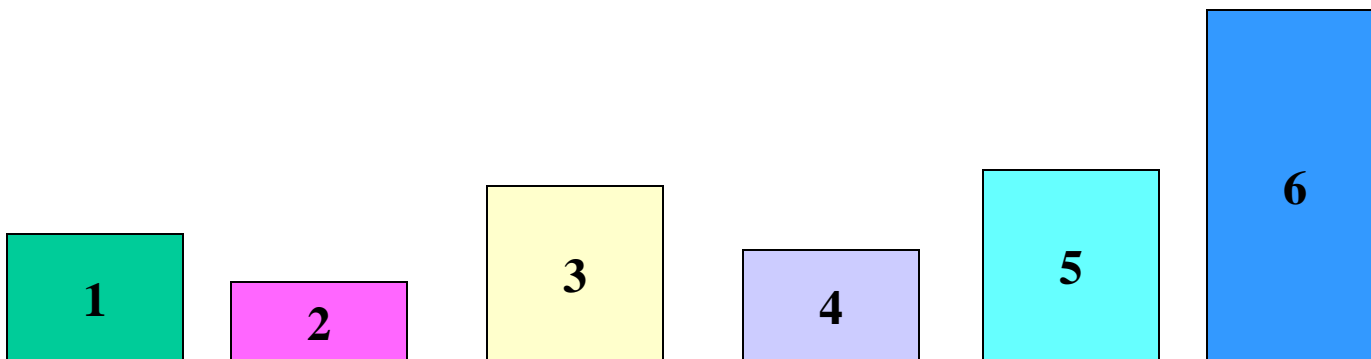
Задача упаковки в контейнеры Bin-packing (иллюстрация)

КОНТЕЙНЕРЫ
ДЛЯ
УПАКОВКИ

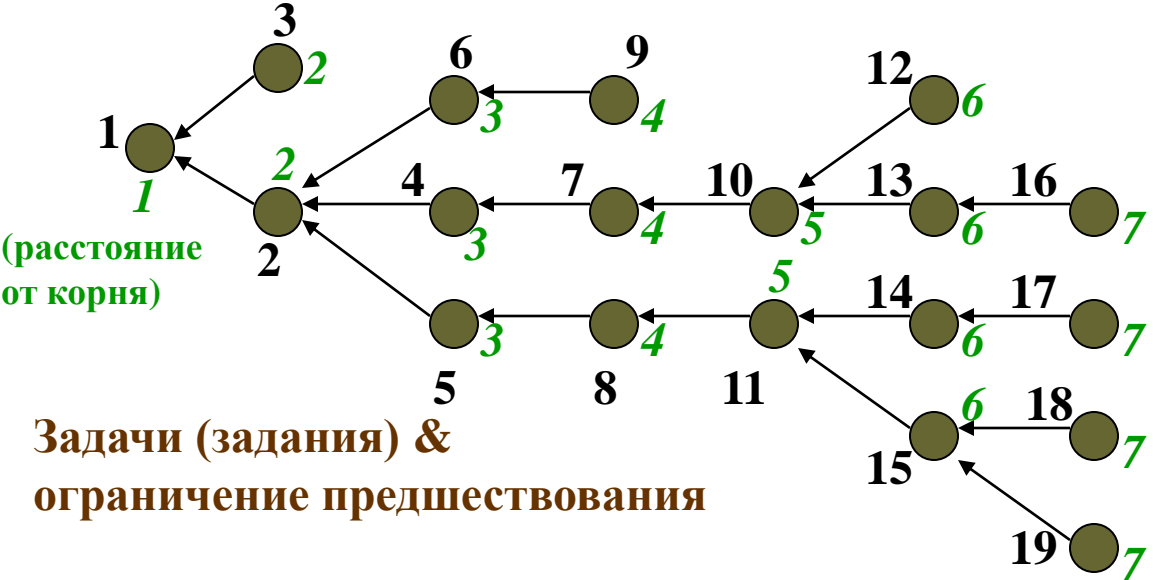


ЦЕЛЬ:
Использование
минимального
числа
контейнеров

ЭЛЕМЕНТЫ



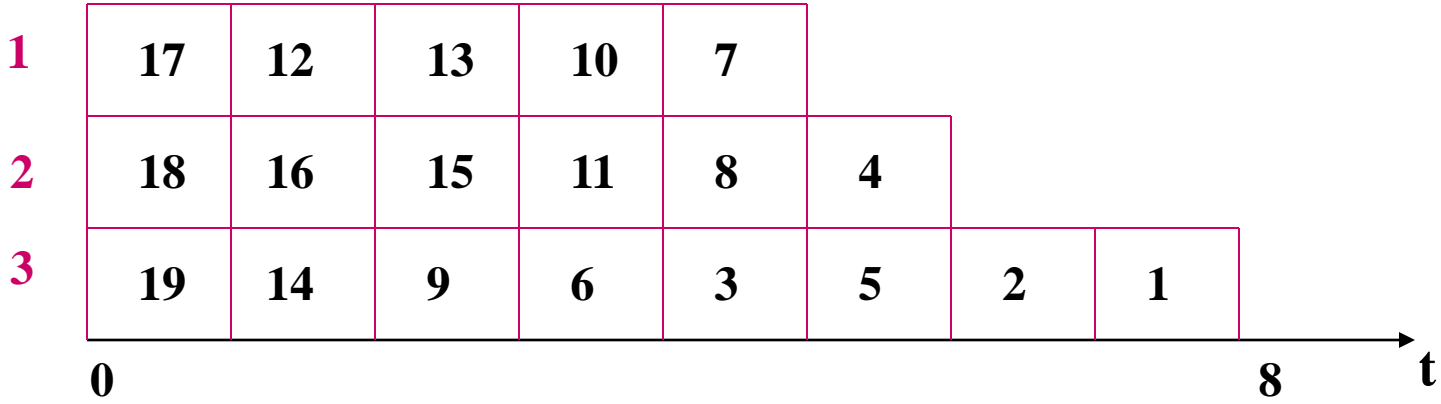
Календарное планирование (теория расписаний):
 иллюстративный пример для «сборки» (алгоритм длиннейших хвостов)



ЦЕЛЬ:
 Минимальное
 общее время
 выполнения
 (всех заданий)

**Задачи (задания) &
 ограничение предшествования**

3 процессора:



Простая задача календарного планирования (теории расписаний) для одной машины (процессор): Формулировка

Исходное множество элементов (заданий): $R = \{ 1, \dots, i, \dots, n \}$

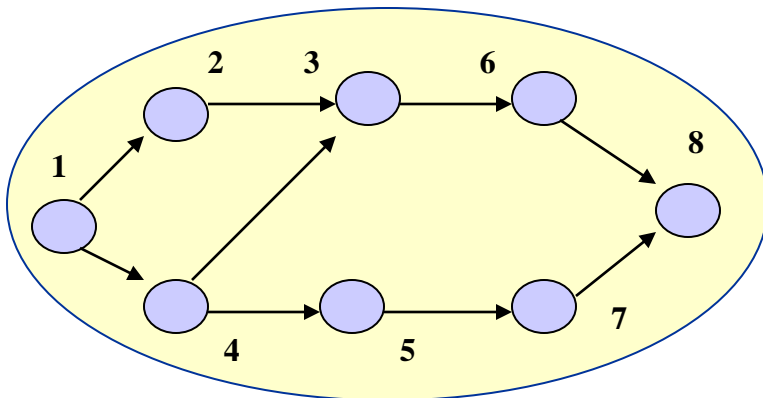
Расписание (линейное упорядочение): $S = \langle s[1], \dots, s[i], \dots, s[n] \rangle$

$s[i]$ – номер элемента на позиции i в расписании S

$f(S)$ это положительная функция с действительными значениями

Задача:

Найти оптимальное расписание S^* : $f(S^*) = \min f(S) \quad \forall S$



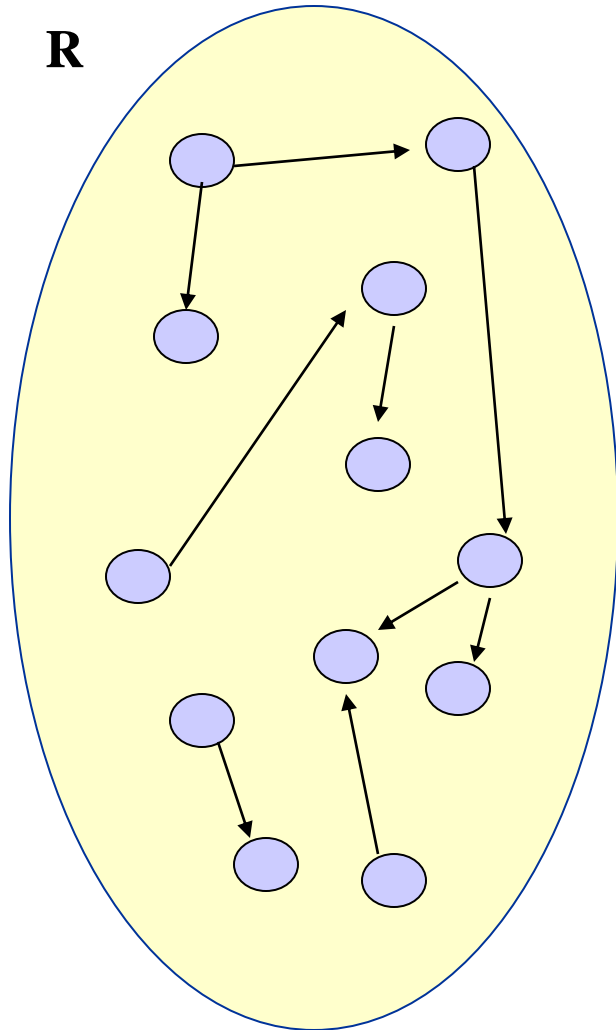
Ограничение предшествования:

$G = (R, E)$

(обычно: без циклов)

E - отображение R на R

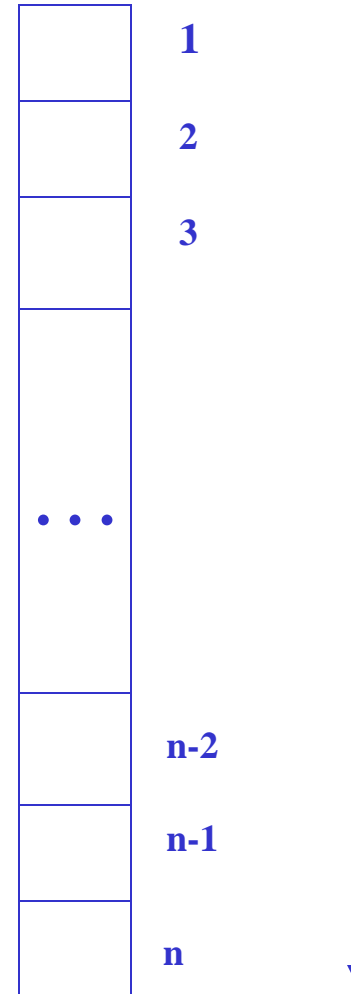
Базовый иллюстративный рисунок



отображение



R \Rightarrow **S**



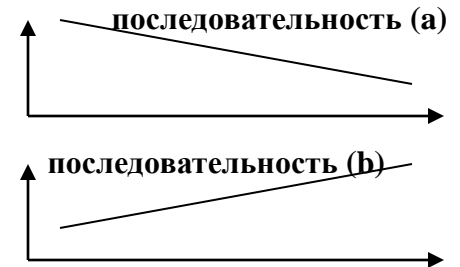
Базовая задача теории расписаний [Hardy, Littlewood, Polya, 1930]

Последовательность $(a) = \{ a_1, a_2, \dots, a_n \}$
 Последовательность $(b) = \{ b_1, b_2, \dots, b_n \}$

\Rightarrow

$S = \langle b(s[1]), b(s[2]), \dots, b(s[n]) \rangle$
 $f(S) = \sum_{i=1}^n a_i b(s[i]) \Rightarrow \min$

ЛЕММА: S^* - оптимальное расписание если
 (1) последовательность (a)
 упорядочена по невозрастанию
 (2) последовательность (b)
 упорядочена по неубыванию



ДОКАЗАТЕЛЬСТВО: Пусть $a_1 > a_2 > \dots > a_n$ & $b_1 < b_2 < \dots < b_n$
 Пусть существуют j и k такие, что $a_j > a_k$ & $b_j > b_k$ ($j < k$)
 Тогда мы переставляем b_j и b_k в последовательности (b) и имеем:
 $\Delta f = a_j b_k + a_k b_j - (a_j b_j + a_k b_k) = (a_j - a_k)(b_k - b_j) \leq 0$.

Последовательность (a)



Последовательность (b)



Базовая задача теории расписаний (Smith, 1956)

$$R = \{ 1, \dots, i, \dots, n \}$$

$$S = \langle s[1], \dots, s[i], \dots, s[n] \rangle$$

$$f(S) = \sum_{i=1}^n f_i(C_i) \Rightarrow \min \quad (1)$$

C_i is a completion time for job (task) i
 $f_i(C_i) = a_i C_i + b_i$ (penalty function, $a_i > 0$)

ТЕОРЕМА 1: S^* - оптимальное расписание ($f(S^*) \leq f(S) \forall S$) если

1. Существует действительная функция $g(i,j)$ такая, что

$$g(i,j) < g(j,i) \Rightarrow f(S) < f(S^*)$$

2. В расписании S^* $i < j$ if $g(i,j) < g(j,i)$

Это Задача 1 (P1)

Базовая задача теории расписаний (Танаев, 1966)

$$R = \{ 1, \dots, i, \dots, n \}$$

$$S = \langle s[1], \dots, s[i], \dots, s[n] \rangle$$

$$f(S) = \sum_{i=1}^n f_i \Rightarrow \min \quad (2)$$

C_i – момент выполнения задания (работы) i

$$f_i(C_i) = a_i \exp(\lambda C_i) \quad (\lambda > 0)$$

Это Задача 2 (P2)

Задача 1 P1: $\omega(i) = \tau_i / a_i$

Задача2 P2: $\omega(i) = a_i \exp(\lambda \tau_i) / (1 - \exp \lambda \tau_i)$

ЗАМЕЧАНИЕ:

Рассмотренные алгоритмы (на основе упорядочения)

можно использовать

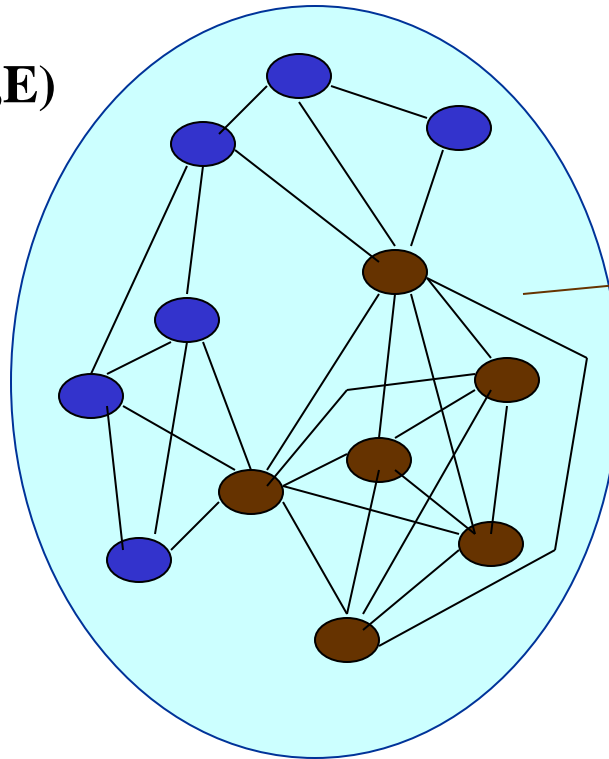
**(расширенная версия) в случае ограничений предшествования
в виде дерева или параллельно-последовательного графа**

Задача максимальной клики (иллюстрация)

Исходный граф $G = (R, E)$, R – это множество вершин,
 E – это множество ребер

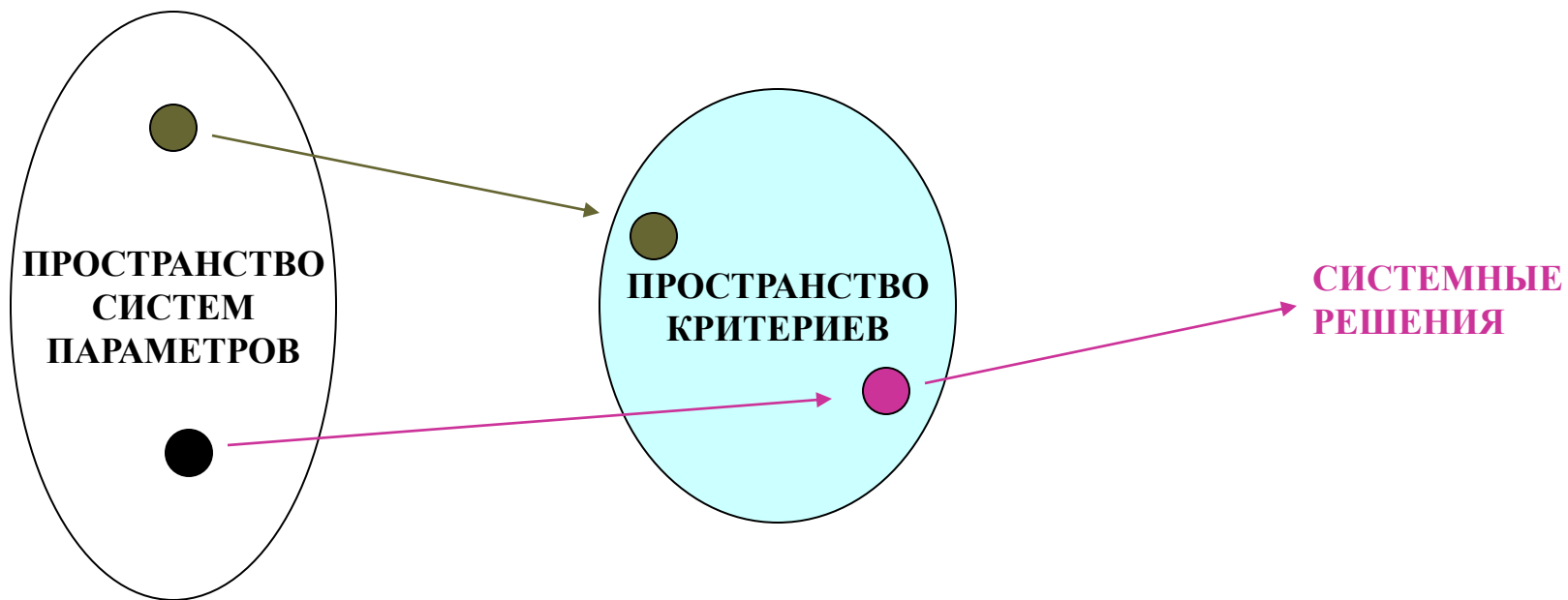
Задача: Найти максимальную (по числу вершин) клику
(т.е., полный подграф)

$G = (R, E)$



Клика,
состоящая из
6 вершин
(максимальный
полный
подграф)

Схема многокритериального системного проектирования



Иерархия требований / критериев

1. Экология, политика

2. Экономика, маркетинг

3. Технология (производство, техобслуживание и др.)

4. Инженерное проектирование

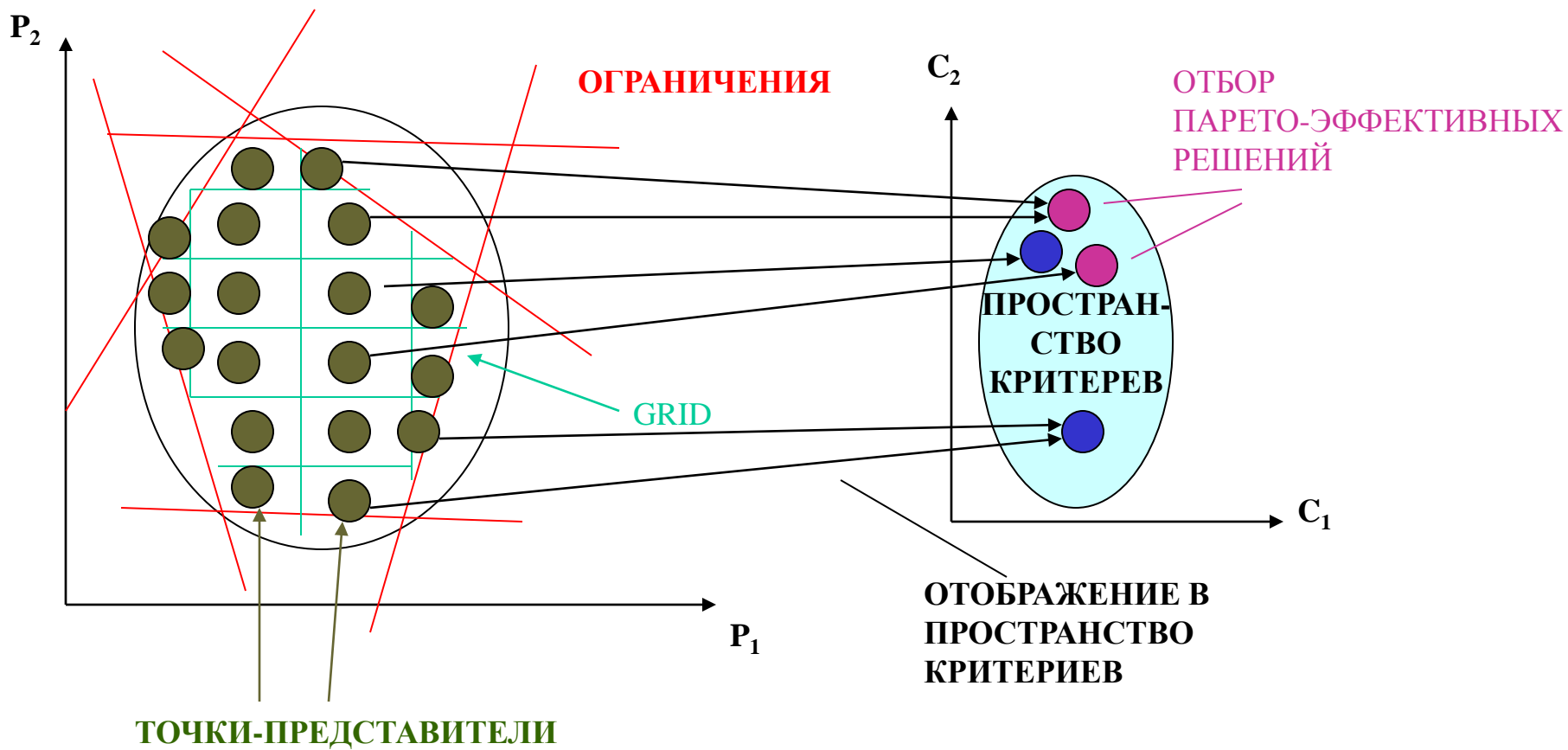


**ПРОСТРАНСТВО
КРИТЕРИЕВ**



**ПРОСТРАНСТВО
СИСТЕМ
ПАРАМЕТРОВ**

Схема проектирования PSI (исследование пространства параметров) (Соболь & Статников)



ЛЕКЦИЯ 16. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.16. Перестановочный прием, генетический алгоритм и др.

ПЛАН:

- 1.Техническая документация (опыт в России)
- 2.Типы перестановочного приема
- 3.Генетические алгоритмы
- 4.Многокритериальная эволюционная оптимизация
- 5.Multidisciplinary optimization
- 6.Смешанное целочисленное нелинейное программирование

Oct. 8, 2004

БАЗОВАЯ ВЕРСИЯ

- 1.Предварительный “аван”-проект**
- 2.”Аван”-проект**
- 3.Техническое предложение**
- 4.Техническое задание**
- 5.Технический проект**
- 6.Рабочий проект**
- 7.Отчет об 1-ом этапе опытной эксплуатации
(включая предложения по улучшению)**
- 8.Итоговый отчет об эксплуатации
(включая предложения по улучшению)**

«СОКРАЩЕННАЯ» ВЕРСИЯ

- 1.Техническое предложение**
- 2.Техно-рабочий проект**
- 3.Отчет об эксплуатации**

Типы перестановочного приема (иллюстрация)

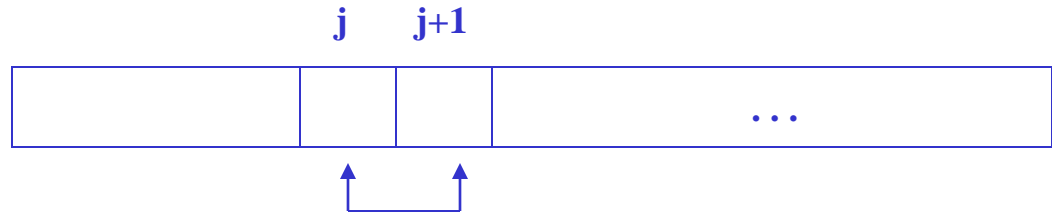
БАЗОВАЯ ВЕРСИЯ

Последовательность



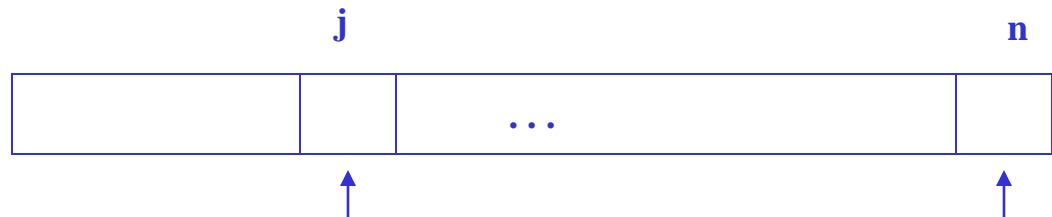
ВЕРСИЯ ДЛЯ СОСЕДНИХ ЭЛЕМЕНТОВ

Последовательность



ВЕРСИЯ ДЛЯ ПОСЛЕДНЕГО (ИЛИ ПЕРВОГО) ЭЛЕМЕНТА

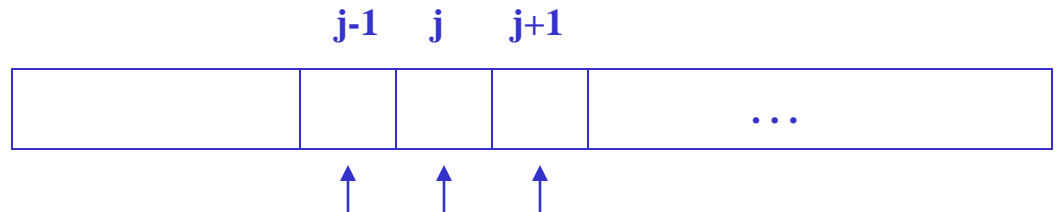
Последовательность



Перестановочный прием с 3-мя и 4-мя элементами (иллюстрация)

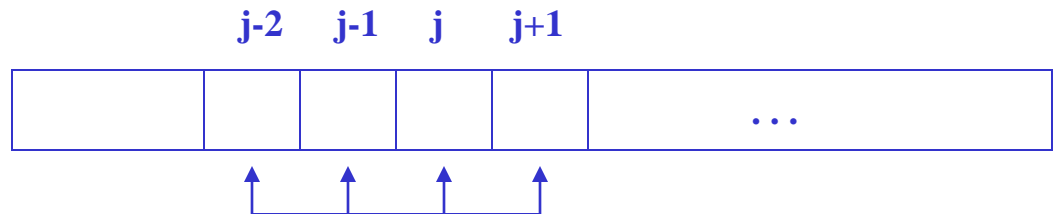
ВЕРСИЯ ДЛЯ ПЕРЕСТАНОВОЧНОГО ПРИЕМА С 3-МЯ ЭЛЕМЕНТАМИ

Последовательность



ВЕРСИЯ ПЕРЕСТАНОВОЧНОГО ПРИЕМА С 4-МЯ ЭЛЕМЕНТАМИ

Последовательность



Перестановочный прием: случай размерности 2 (иллюстрация)

2-ЭЛЕМЕНТ ПЕРЕСТАНОВОЧНЫЙ СЛУЧАЙ

«Площадь»

						...	
		●	●			...	
					●	...	●
		●				...	
			●			...	

4-ЭЛЕМЕНТ ПЕРЕСТАНОВОЧНЫЙ СЛУЧАЙ

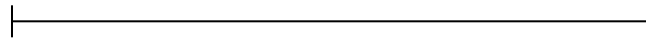
«Площадь»

						...	
		●	●			...	
		●	●		●	...	●
						...	
	●			●		...	

Базовая задача о рюкзаке:

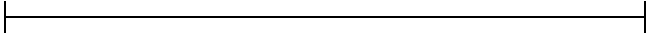
$$\begin{aligned} \mathit{max} \quad & \sum_{i=1}^m c_i x_i \\ \mathit{s.t.} \quad & \sum_{i=1}^m a_i x_i \leq b \\ & x_i \in \{0, 1\}, \quad i = 1, \dots, m \end{aligned}$$

Решение $x_0 = (x_1, \dots, x_i, \dots, x_m)$



Генетические алгоритмы (иллюстрация для задачи о рюкзаке)

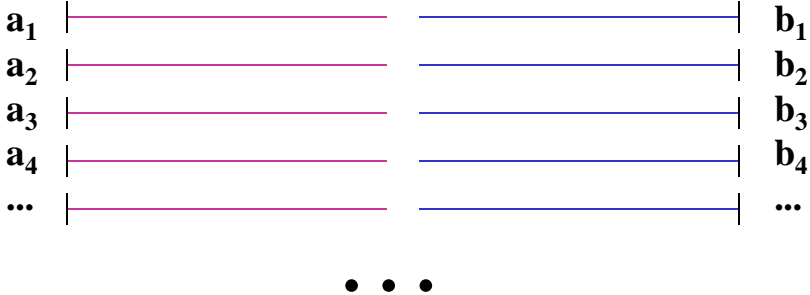
ШАГ 1. ИСХОДНОЕ РЕШЕНИЕ x_0



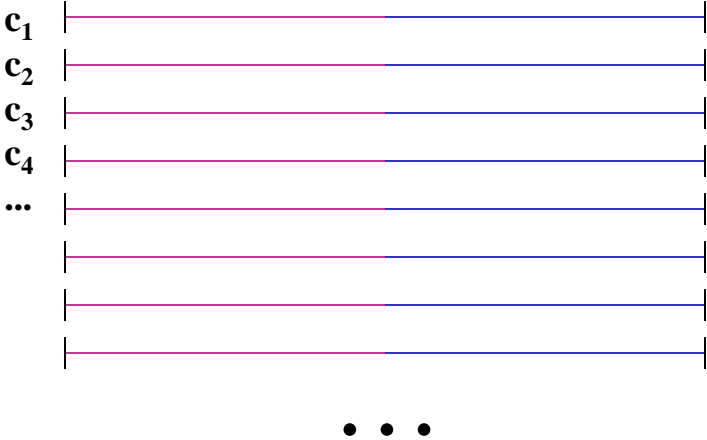
ШАГ 2. РАЗБИЕНИЕ x_0 НА 2 ЧАСТИ



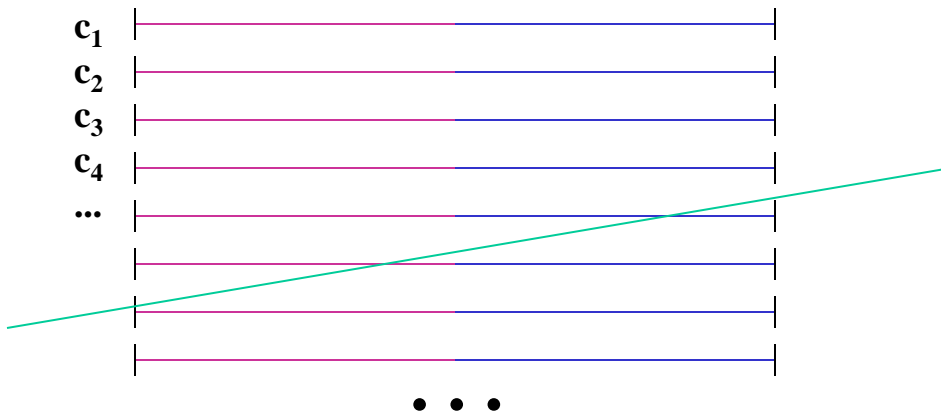
ШАГ 3. МУТАЦИЯ (ГЕНЕРАЦИЯ ВЕРСИЙ): *перестановка элементов,
*изменение элементов и др..



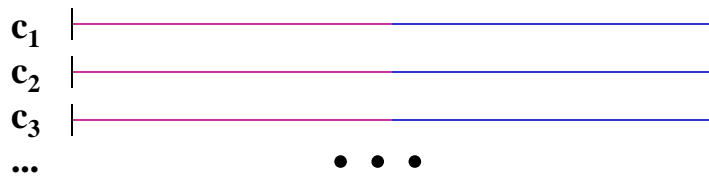
ШАГ 4. ГЕНЕРАЦИЯ НОВЫХ РЕШЕНИЯ НА ОСНОВЕ ПАР (a_i, b_j)



ШАГ 5. Исключение некоторых решений по ресурсным ограничениям ($\leq b$)



ШАГ 6. Отбор наилучших решений



Отбор на основе двух подходов:

1. Отбор на основе функции полезности
2. Отбор на основе правила Парето

ЭТО -

Многокритериальная эволюционная оптимизация

ШАГ 7. Повторение шагов 2, 3, 4, 5 и 6 для отобранных решений

Multidisciplinary optimization (проектирование систем для космоса, авиации, строительства, кораблестроения)

**Общая модель оптимизации с ограничениями
Соответствующими определенным дисциплинам
(в частности, вес, длина, надежность и др.):**

$$\begin{array}{ll} \max f(\mathbf{x}) & (\text{or } \text{extr } f(\mathbf{x})) \\ \text{subject to} & \\ \varphi_1(\mathbf{x}) \leq W & \text{вес} \\ B_1 \leq \varphi_2(\mathbf{x}) \leq B_2 & \text{высота} \\ C_1 \leq \varphi_3(\mathbf{x}) \leq C_2 & \text{температура} \\ D_1 \leq \varphi_4(\mathbf{x}) \leq D_2 & \text{надежность} \\ & \dots \\ & \varphi_k(\mathbf{x}) \leq 0 \end{array}$$

$\varphi_j(\mathbf{x})$ - функция ограничения ($1 \leq j \leq k$)

Int. Society for Structural and Multidisciplinary Optimization
(civil engineering, ship engineering, marine engineering, aerospace engineering)

[//www.issmo.org](http://www.issmo.org)

Optimal design of structures (including issues of fluids)

Смешанное целочисленное нелинейное программирование (process systems engineering & chemical engineering)

Общая модель оптимизации включающая целые и непрерывные переменные:

$$\min F(x, y)$$

subject to

$$h(x, y) = 0$$

$$g(x, y) \leq 0$$

где x - вектор бинарных переменных (выбор подсистем)

*y - вектор непрерывных переменных / параметров
(например, размер)*

***Global optimization**

***Process Systems Engineering (chemical engineering, etc.)**

***Prof. C.A. Floudas (Princeton Univ, Chemical Engineering)**

***Prof. I.E. Grossmann (Carnegie Mellon Univ., Chemical Engineering)**

1.Метод ветвей и границ

2.Комбинаторные гибридные методы

3.Метод градиента

***4.Метод внутренней точки
(Interior point method)***

ЛЕКЦИИ 17-18. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.17. Основные модели комбинаторной оптимизации III

Л.18. основные модели комбинаторной оптимизации IV

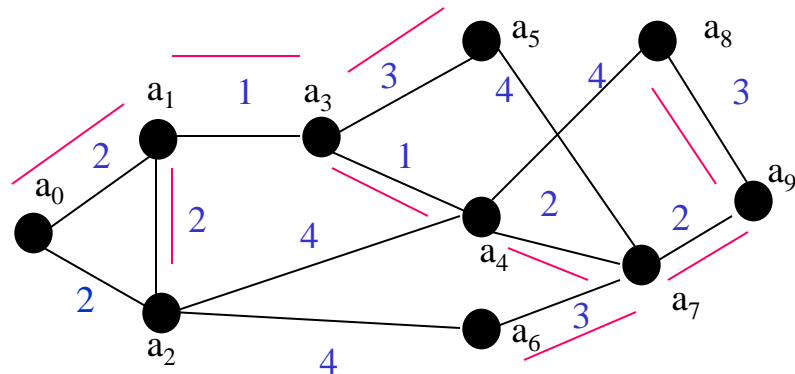
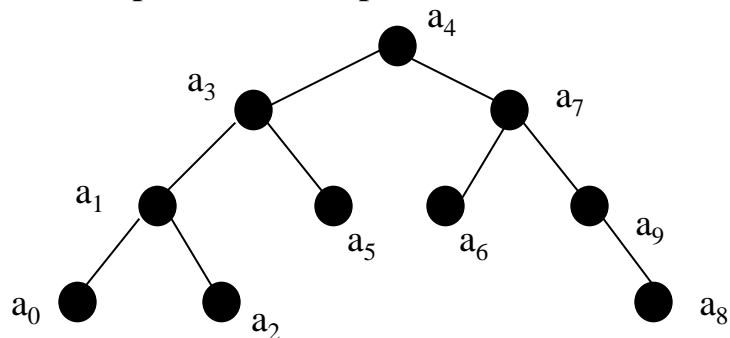
ПЛАН:

- 1.Покрытие (иллюстрации): *покрывающее дерево, *минимальное дерево Штейнера, *2-связный граф,
- 2.Задача коммивояжера, задача о назначении (паросочетании) (формулировки)
- 3.Задача сопоставления (иллюстрации) , применение в обработке экспериментальных данных
- 4.раскраска графа, задачи покрытия (иллюстрации и приложения)
- 5.Выравнивание (Alignment), максимальные подструктуры, минимальные надструктуры
(иллюстрации, приложения)
- 6.задачи составления расписаний (Timetabling)

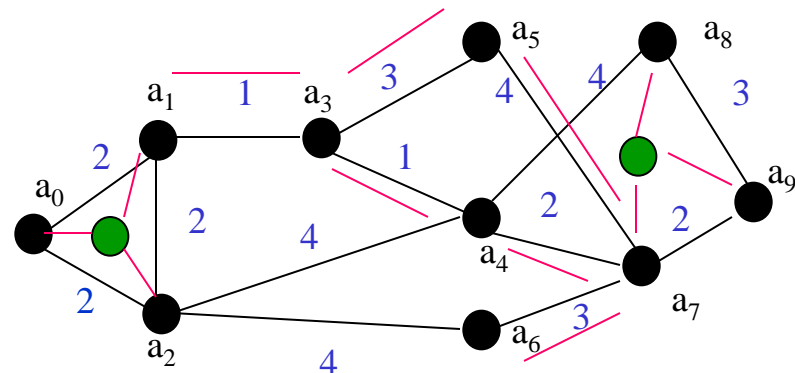
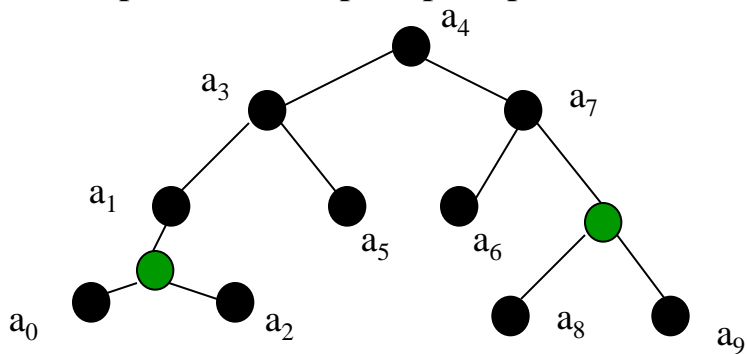
Окт. 9, 2004

Покрывание (иллюстрация): 1-связный граф

Покрывающее дерево (длина = 19):

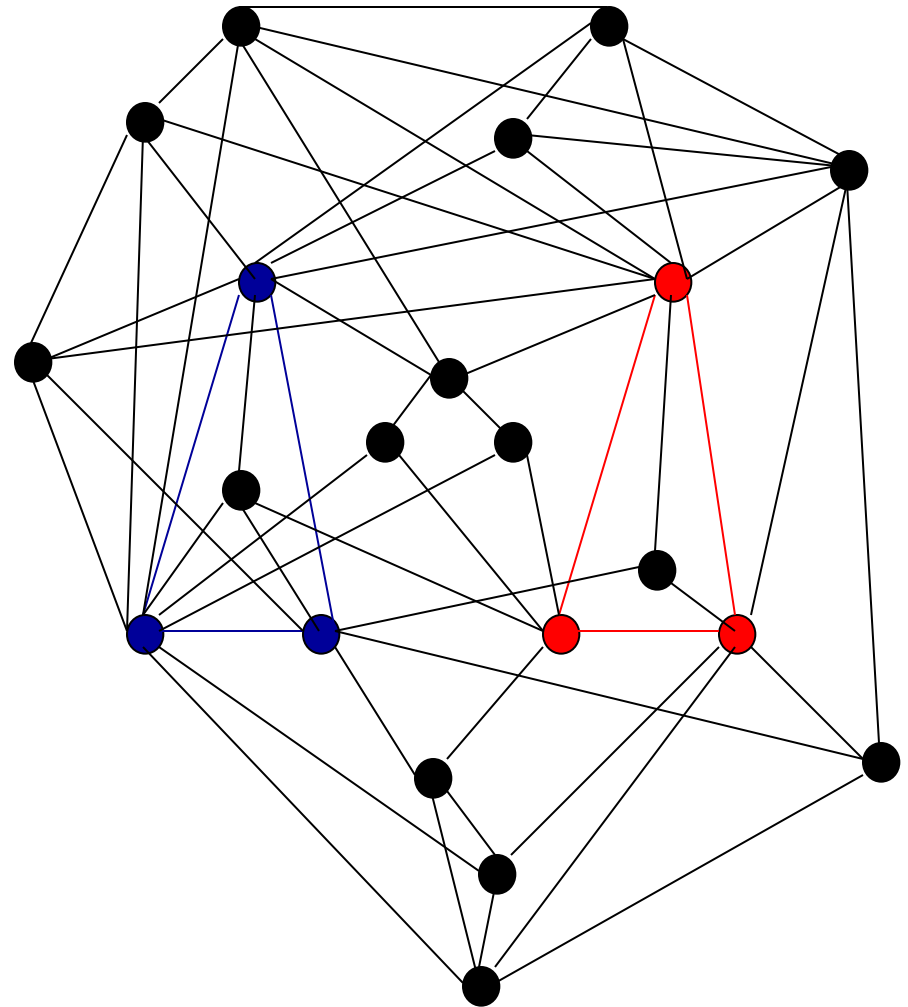


Дерево Штейнера (пример):



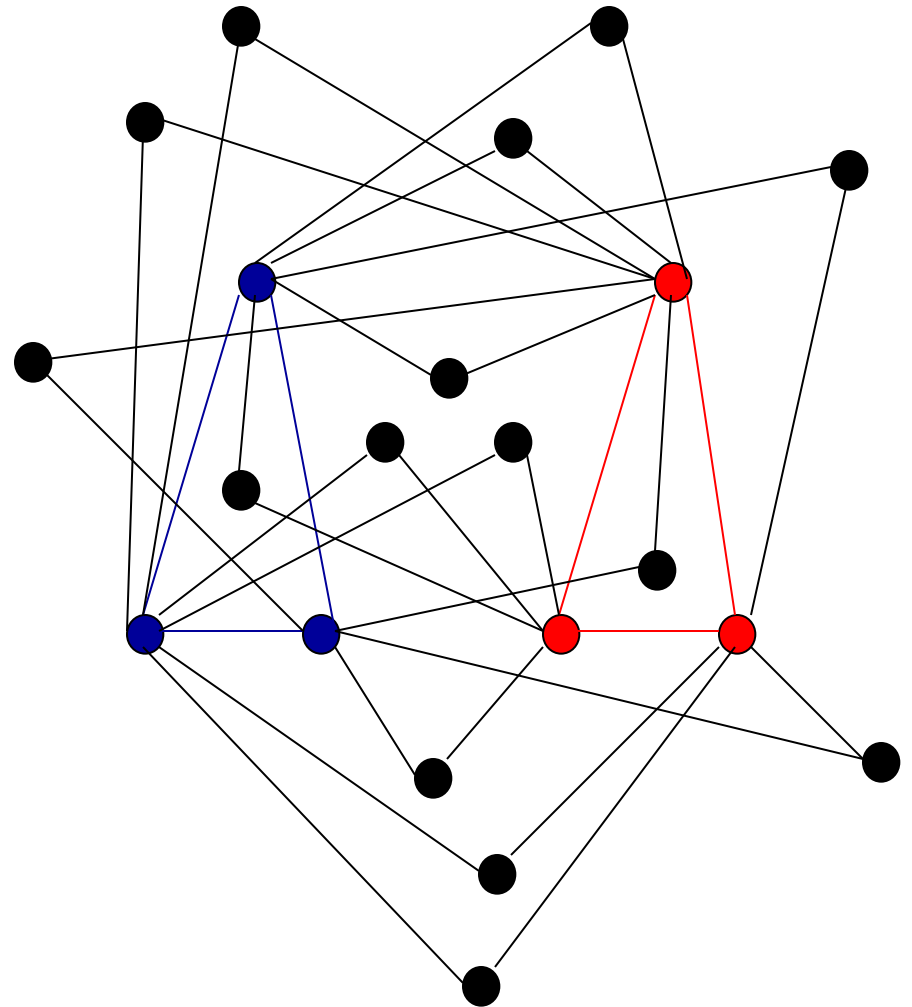
Покрытие (иллюстрация): 2-связный граф

Покрытие 2-связным графом:
Выделение двух клик из 3 узлов
(центры)

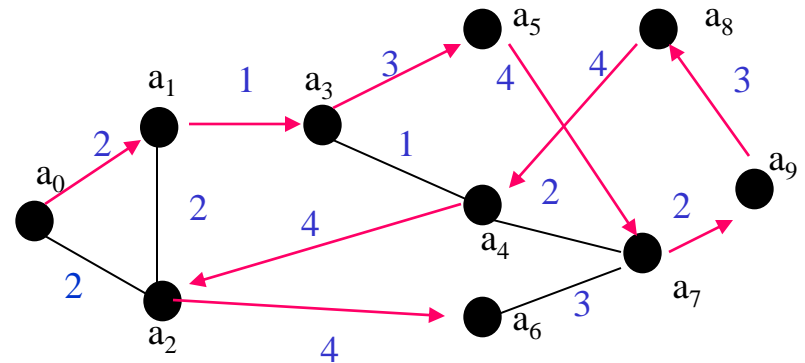
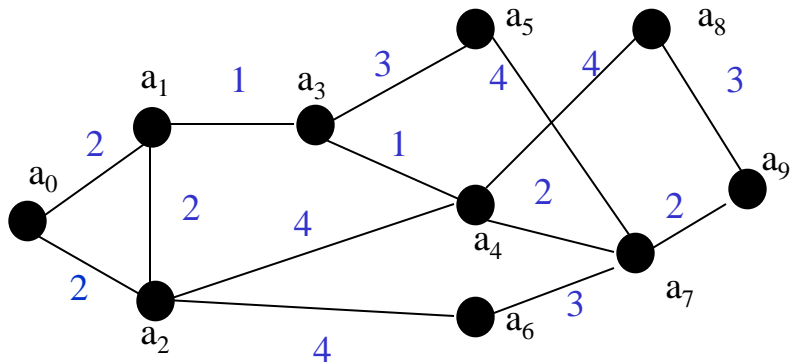


Покрытие (иллюстрация): 2-связный граф

Покрытие 2-связным графом
*Соединение каждого узла с
двумя центрами*



Задача коммивояжера



$$L = \langle a_0, a_1, a_3, a_5, a_7, a_9, a_8, a_4, a_2, a_6 \rangle$$

$$2+1+3+4+2+2+3+4+4+4$$

ФОРМУЛИРОВКА:

Множество городов: $A = \{ a_1, \dots, a_i, \dots, a_n \}$

Расстояние между городами i и j : $\rho(a_i, a_j)$

Π - множество перестановок элементов A ,
перестановка

$$s^* = \langle a(s^*[1]), \dots, a(s^*[i]), \dots, a(s^*[n]) \rangle$$

$$\min_{(s \in \Pi)} f(s) = f(s^*)$$

$$f(s) = \sum_{i=1}^{n-1} \rho(a(s[i]), a(s[i+1])) + \rho(a(s[n]), a(s[1]))$$

Задача коммивояжера

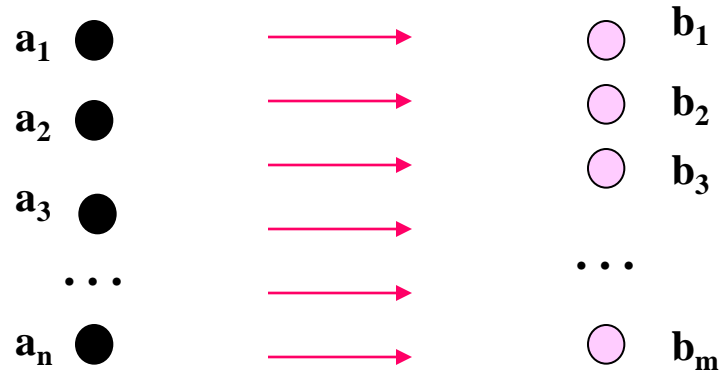
АЛГОРИТМЫ:

1. Жадный алгоритм
 2. На основе покрывающего дерева
 3. Метод ветвей и границ
- Др.

ВЕРСИИ (много):

1. Цикл или путь
 2. m -коммивояжеров
 3. асимметричная задача (*i.e.*, *расстояния*
 $\rho(a_i, a_j)$ и $\rho(a_j, a_i)$ - различны)
 4. Различные типы пространств (метрические пространства и др.)
 5. Многокритериальные задачи
- Др.

Задача о назначении (о паросочетании)



ФОРМУЛИРОВКА:

Множество элементов: $A = \{ a_1, \dots, a_i, \dots, a_n \}$

Множество позиций $B = \{ b_1, \dots, b_j, \dots, b_m \}$ (здесь $n = m$)

Эффективность пары i и j : $z(a_i, b_j)$

$\Pi = \{s\}$ - множество перестановок (назначений) элементов A на позиции множества B :

$s^* = \langle (s^*[1]), \dots, (s^*[i]), \dots, (s^*[n]) \rangle$, т.е., элемент a_i на позицию $s[i]$ в B

Целевая функция:

$$\max \sum_{i=1}^n z(i, s[i])$$

Задача о назначении (паросочетаниях)

АЛГОРИТМЫ:

1. Полиномиальный ($O(n^3)$)

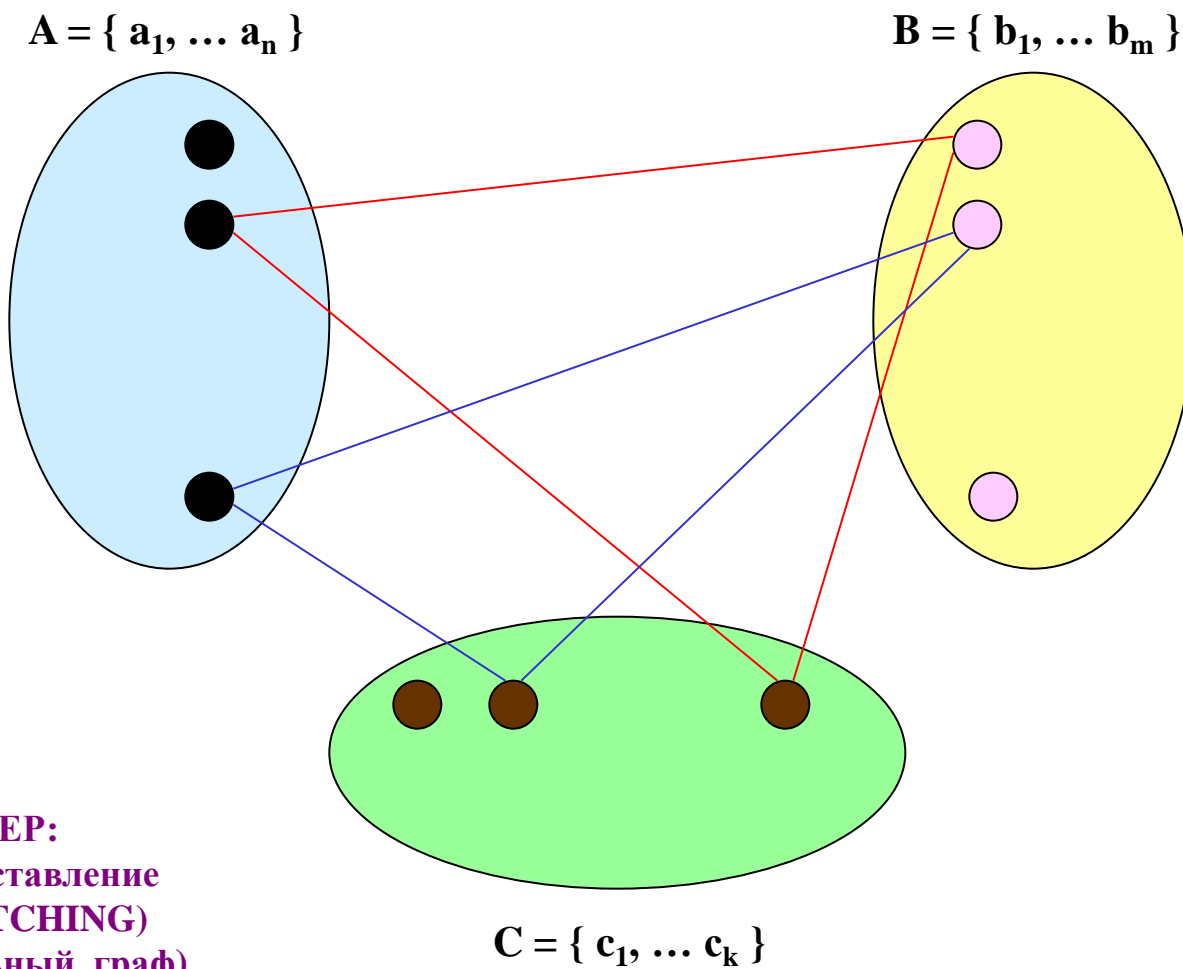
ВЕРСИИ:

1. *Min max* задача

2. Многокритериальные задачи

Др.

Задачи сопоставления



ПРИМЕР:
3-сопоставление
(3-MATCHING)
(3-дольный граф)

АЛГОРИТМЫ:

1. Эвристики

(т.е., жадные алгоритмы, локальная оптимизация, гибридные эвристики)

2. Переборные алгоритмы

(например, метод ветвей и границ)

3. Морфологический подход

ВЕРСИИ:

1. Динамические задачи (сопоставление трасс целей от нескольких радаров)

2. Задачи с ошибками в исходных данных

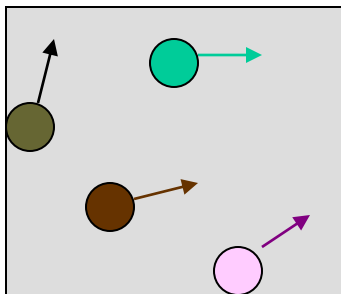
4. Задачи с неопределенностью

(вероятностные параметры, размытые параметры)

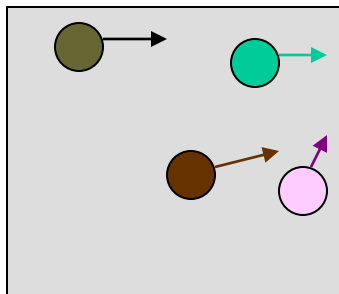
Др.

Прикладной пример: применение задачи о назначении для определения скоростей частиц

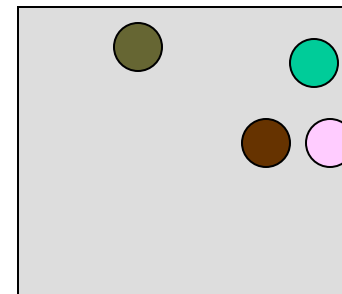
КАДР 1



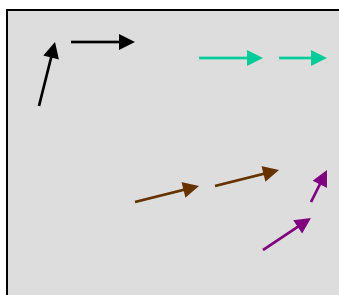
КАДР 2



КАДР 3



ПРОСТРАНСТВО СКОРОСТЕЙ



Прикладной пример: применение задачи о назначении для определения скоростей частиц

МОДЕЛИ & АЛГОРИТМЫ:

- 1. Корреляционные функции (из радиотехники: обработка сигналов)**
- 2. Задача о назначении между двумя соседними кадрами**
(алгоритмические схемы: генетические алгоритмы ,
другие алгоритмы для задачи о назначениях, гибридные схемы)
- 3. Многостадийная задача о назначениях (например, анализ трех кадров и др.)**
(алгоритмические схемы: генетические алгоритмы,
Другие алгоритмы для задачи о назначениях, гибридные схемы)

ВЕРСИИ :

- 1. Базовая задача**
 - 2. Задачи с ошибками**
 - 3. Задача с неопределенными оценками**
(вероятность, размытые множества)
- Др.

Прикладной пример: применение задачи о назначении для определения скоростей частиц

ПРИЛОЖЕНИЯ (среда газа или жидкости):

- 1. Физические эксперименты**
- 2. Науки о климате (например, исследование облаков)**
- 3. Химические процессы**
- 4. Биотехнология**

Источники:

- 1. PIV системы (лазер/оптические системы)**
 - 2. Фотографии со спутников**
 - 3. Электронные микроскопы**
- Др.**

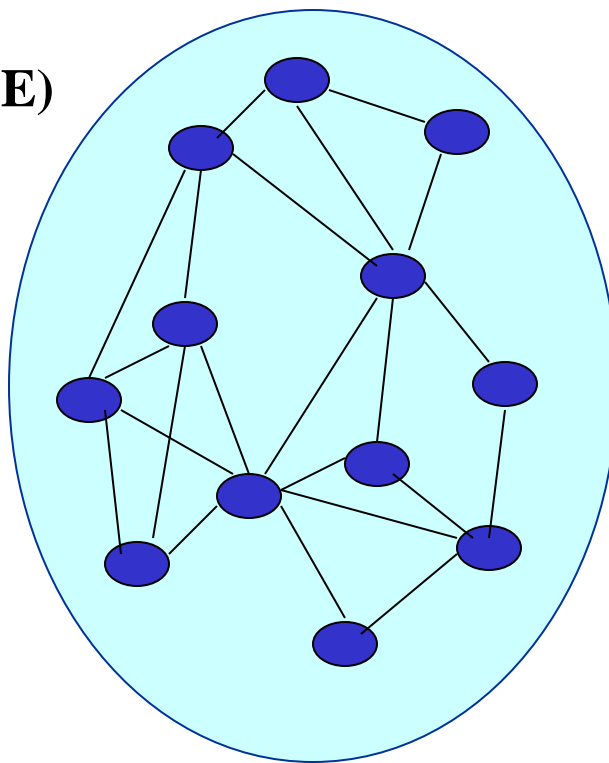
Задача раскраски графа (иллюстрация)

Исходный граф $G = (A, E)$, A – множество вершин, E – множество ребер

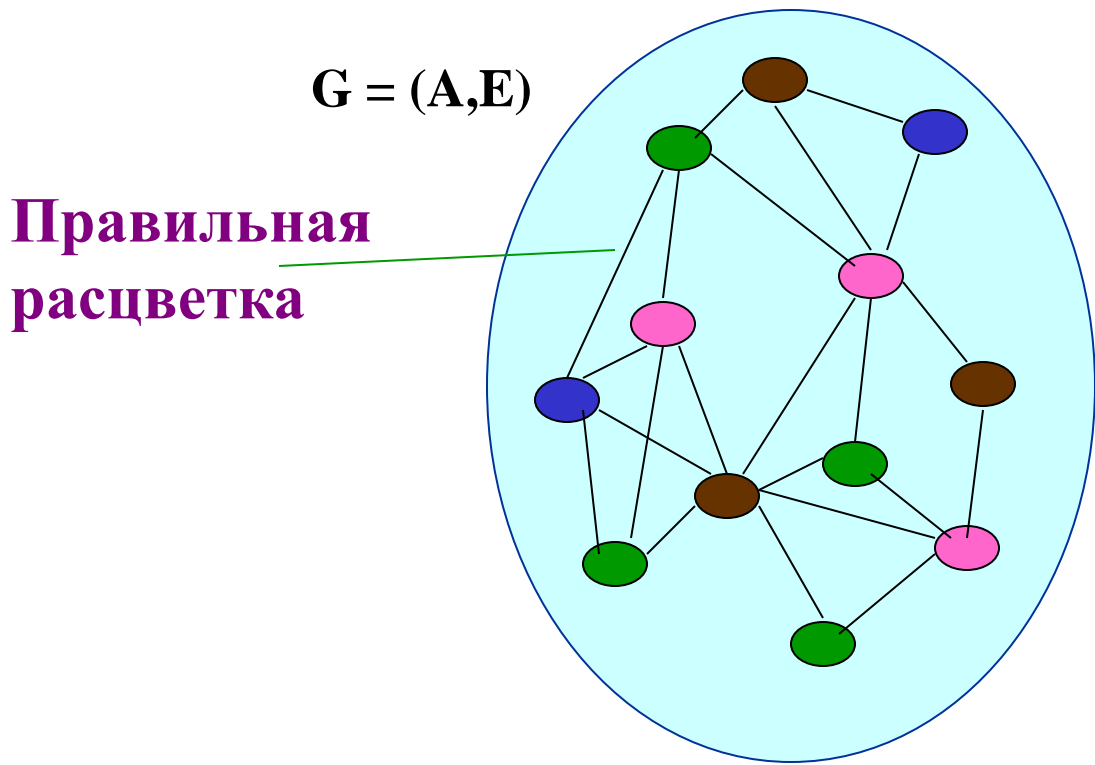
Задача:

Назначить цвета для каждой вершины с минимальным общим числом цветов и ограничением: соседние вершины имеют различный цвет

$G = (A, E)$



Задача раскраски графа (иллюстрация)



ПРИЛОЖЕНИЯ:

1. Назначение регистров в процессе компиляции (А.П. Ершов, 1959)

2. Назначение частот / каналов связи

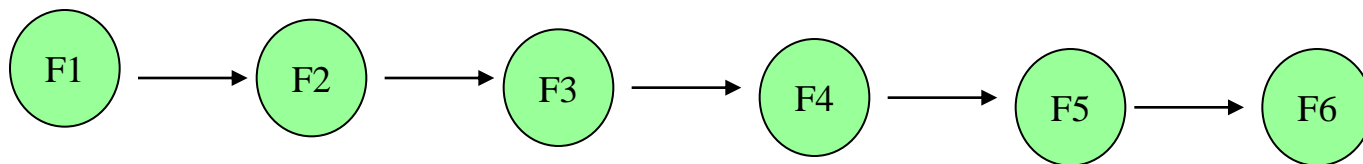
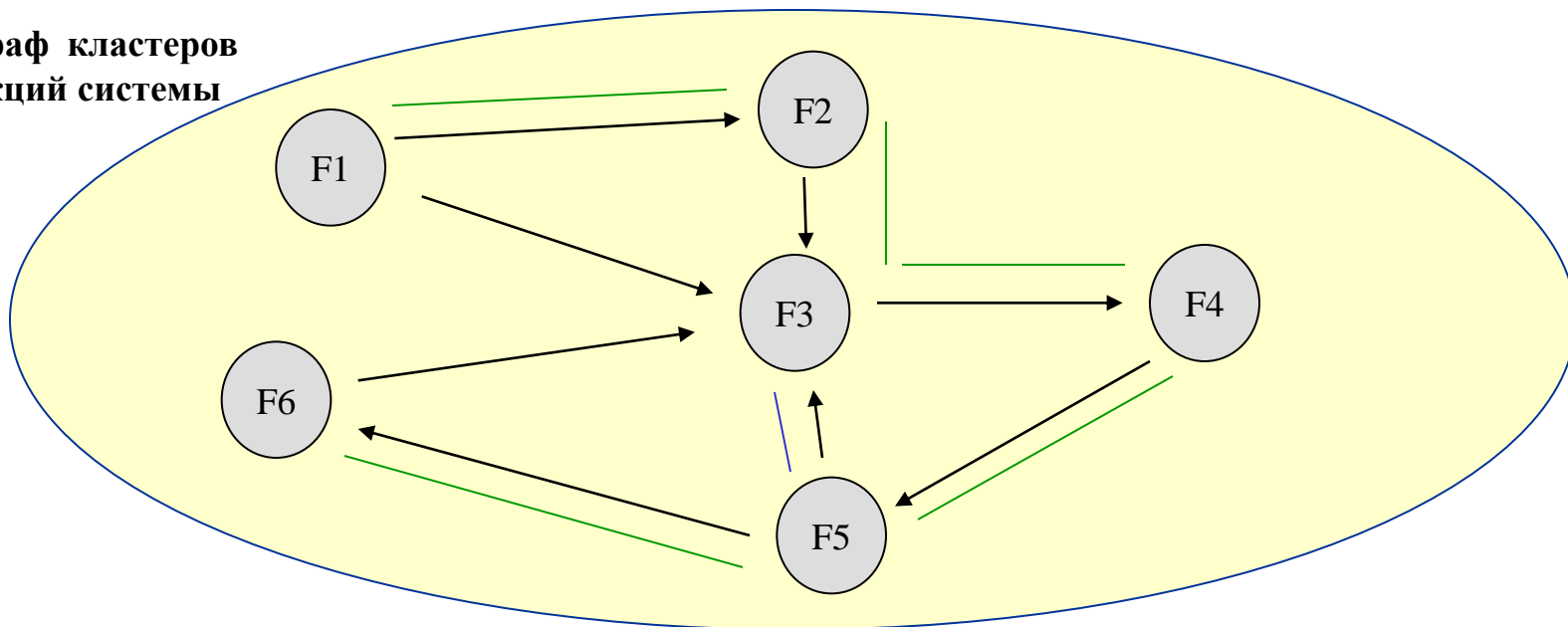
(статическая задача, динамическая задача и др.)

3. Проектирование больших интегральных схем (VLSI design)

Др.

Пример: кластеры функций системы и покрытие цепочками (покрытие вершин)

Орграф кластеров функций системы

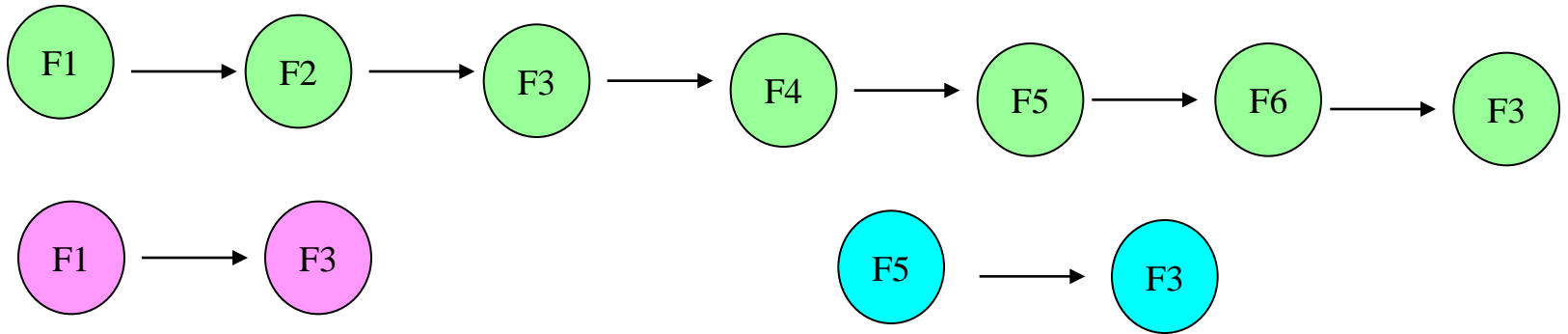
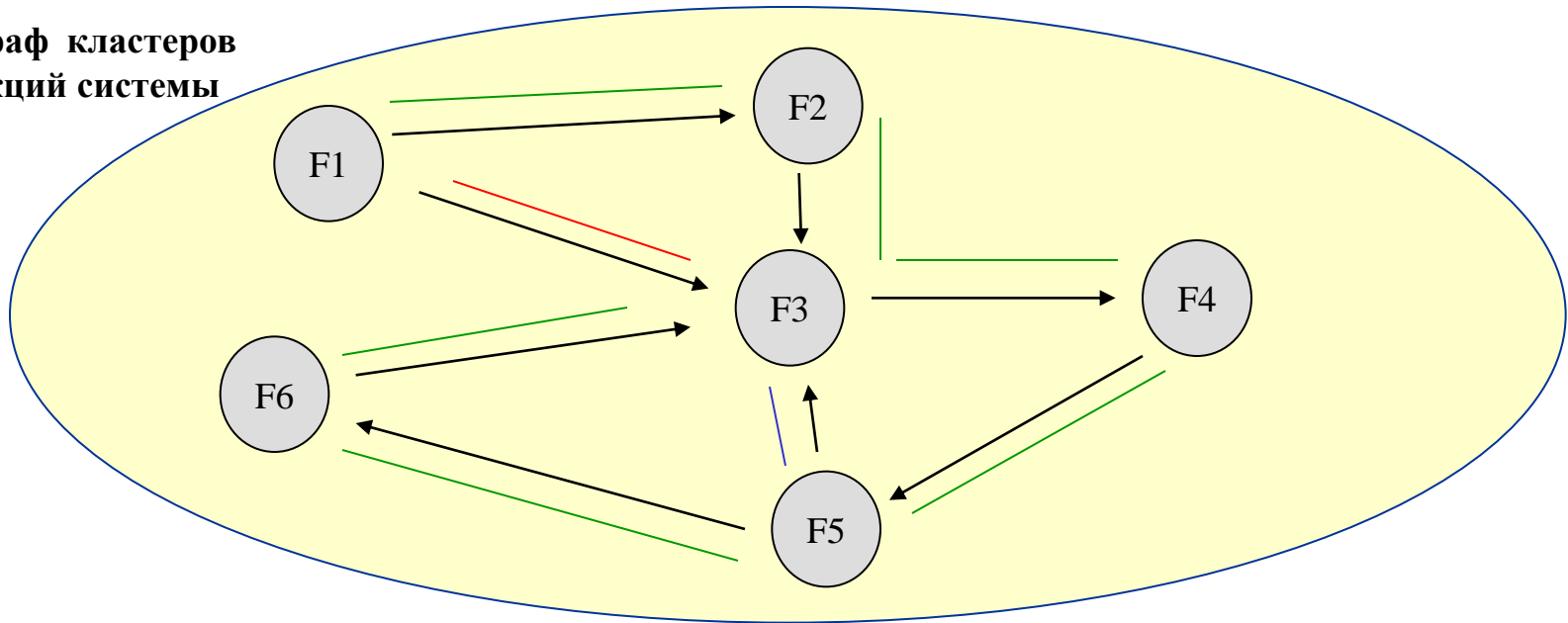


ДЛИННЕЙШИЙ ПУТЬ

Приложение: тестирование систем

Примере: кластеры функций системы и покрытие цепочками (покрытие дуг)

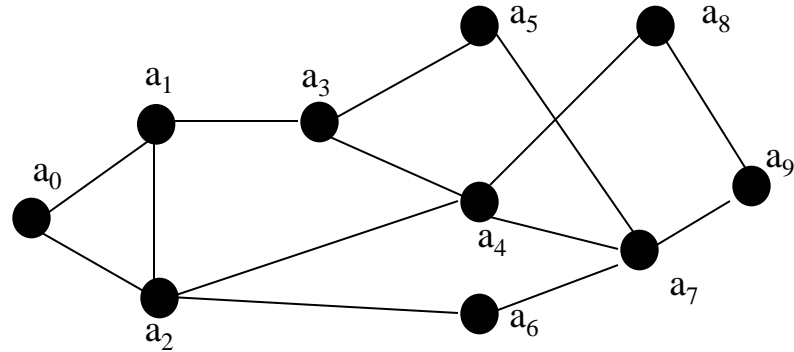
Орграф кластеров функций системы



ПРИЛОЖЕНИЕ: ТЕСТИРОВАНИЕ “ИЗМЕНЕНИЙ”

Иллюстрация: покрытие кликами

Исходный граф



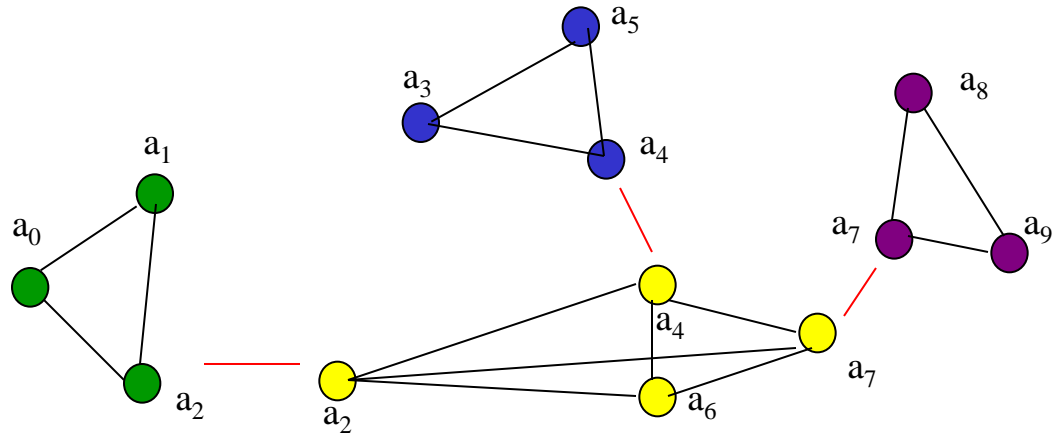
Клики (версия):

$$C_1 = \{ a_0, a_1, a_2 \}$$

$$C_2 = \{ a_3, a_5, a_4 \}$$

$$C_3 = \{ a_7, a_8, a_9 \}$$

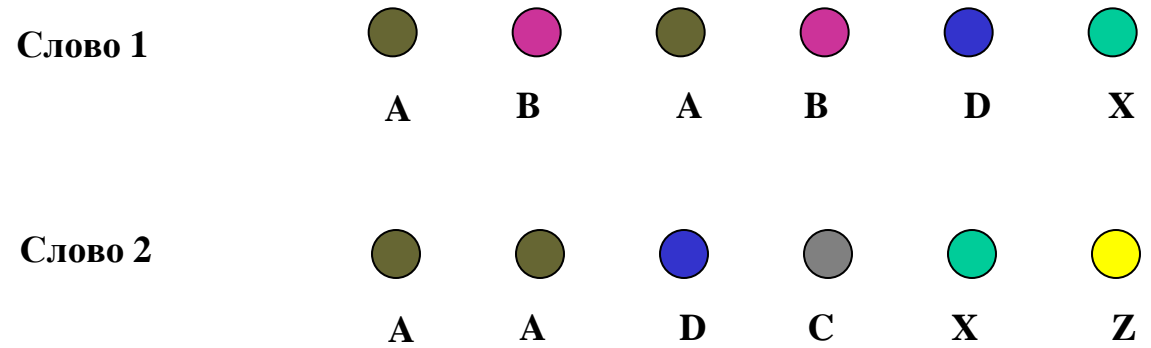
$$C_4 = \{ a_2, a_4, a_6, a_7 \}$$



ПРИЛОЖЕНИЕ: РАЗМЕЩЕНИЕ СЕРВИСА (например, коммуникационные центры)

Задача выравнивания (Alignment) (иллюстрация)

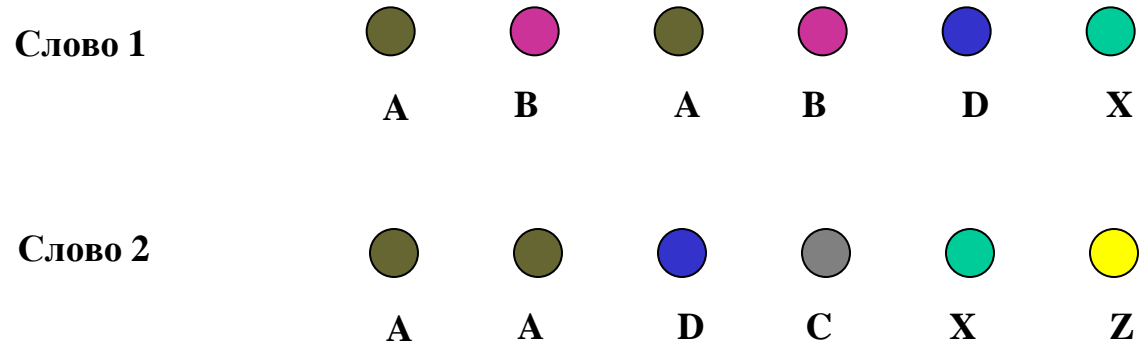
СЛУЧАЙ 2 СЛОВ:



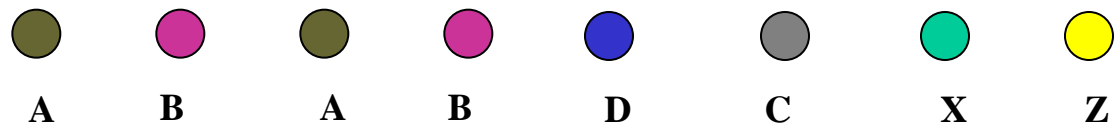
ЗАДАЧА ВЫРАВНИВАНИЯ: *минимальное число дополнительных элементов*

Задача выравнивания (Alignment) (иллюстрация)

СЛУЧАЙ 2 СЛОВ:



Минимальная Надструктура



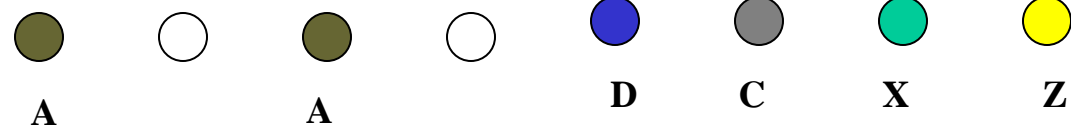
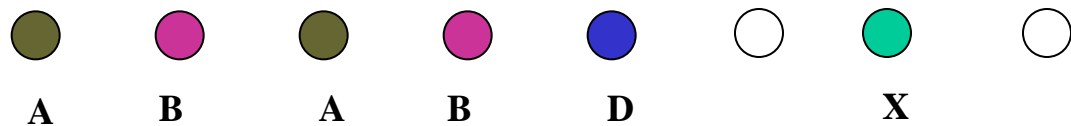
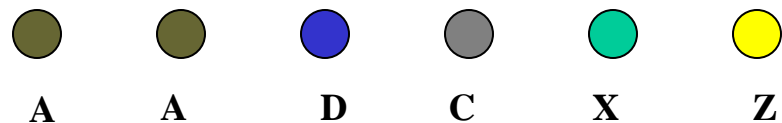
Задача выравнивания (Alignment) (иллюстрация)

СЛУЧАЙ 2 СЛОВ:

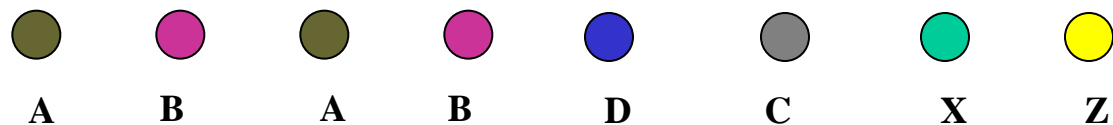
Слово 1



Слово 2



Надструктура



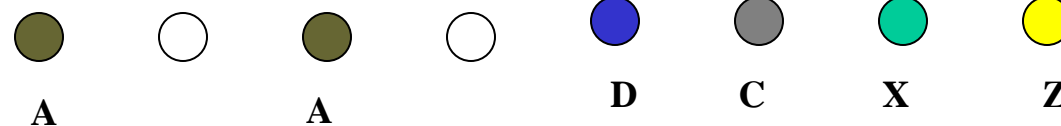
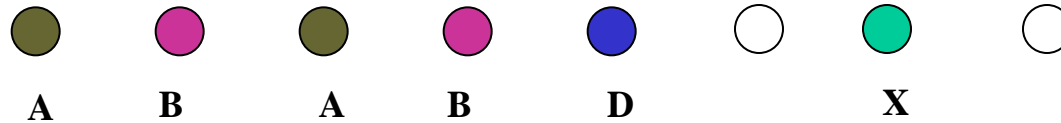
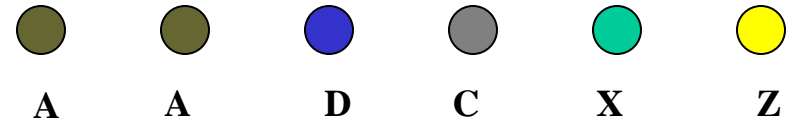
Задача выравнивания (Alignment) (иллюстрация)

СЛУЧАЙ 2 СЛОВ:

Слово 1



Слово 2



- ПРИЛОЖЕНИЯ:**
1. Лингвистика
 2. Биоинформатика (исследование генов и др.)
 3. Обработка последовательностей кадров (обработка изображений)
 4. Моделирование конвейерных систем

Задача выравнивания (Alignment) (иллюстрация)

ДРУГИЕ ВЕРСИИ ЗАДАЧИ:

***СЛУЧАЙ N СЛОВ**

***СЛУЧАЙ 2 2-МЕРНЫХ СЛОВА**

***СЛУЧАЙ N 2-МЕРНЫХ СЛОВ**

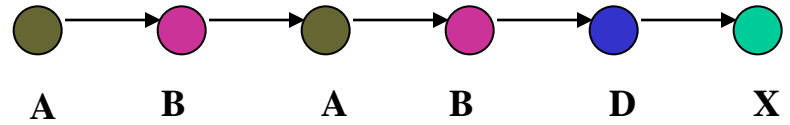
***M-РАЗМЕРНЫЕ СЛУЧАИ**

***ДР.**

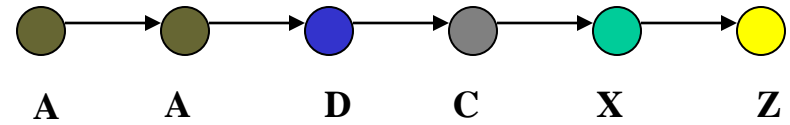
Подструктура и надструктура (иллюстрация)

СЛУЧАЙ 2 ЦЕПЕЙ:

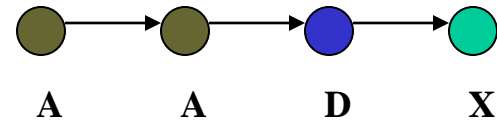
Цепь 1



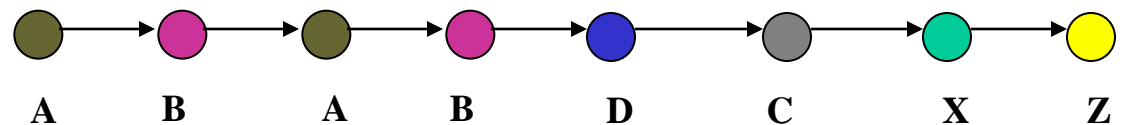
Цепь 2



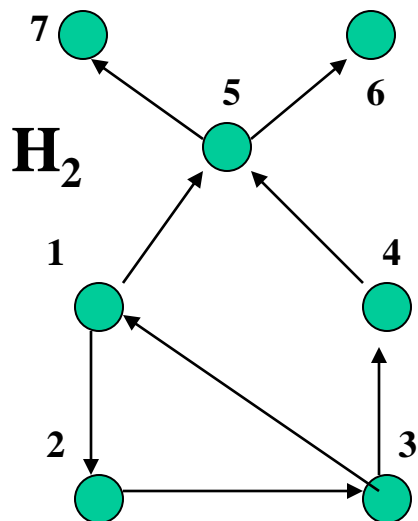
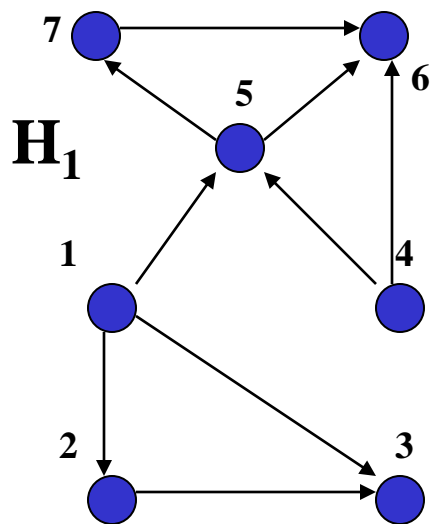
Задача 1: *Максимальная Подструктура*



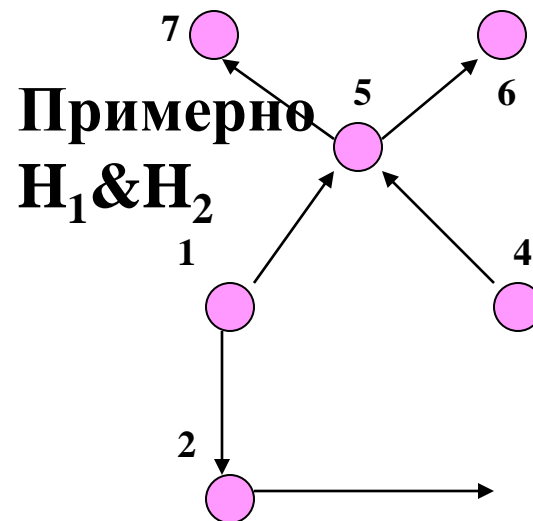
Задача 2: *Минимальная Надструктура*



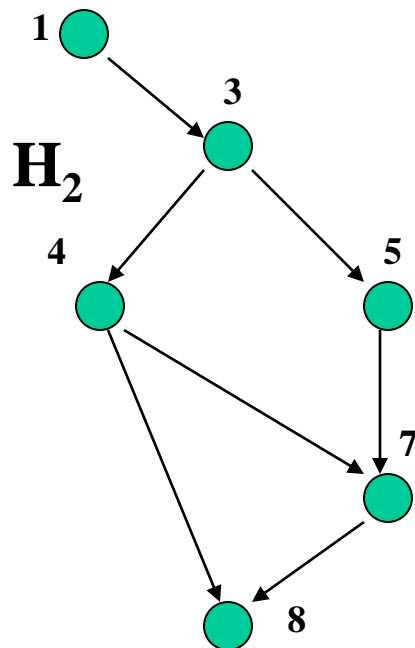
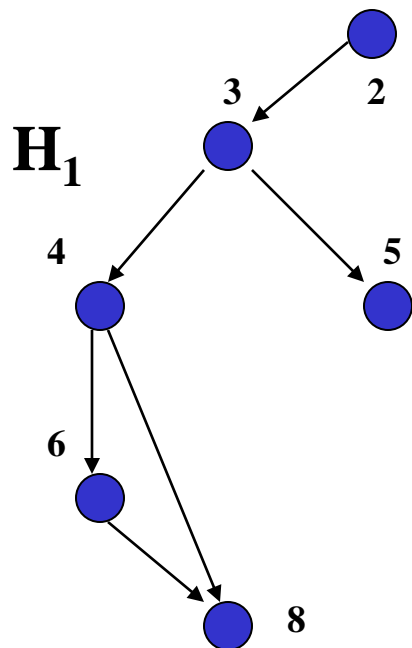
Подструктура и надструктура (иллюстрация): случай 2 оргграфов



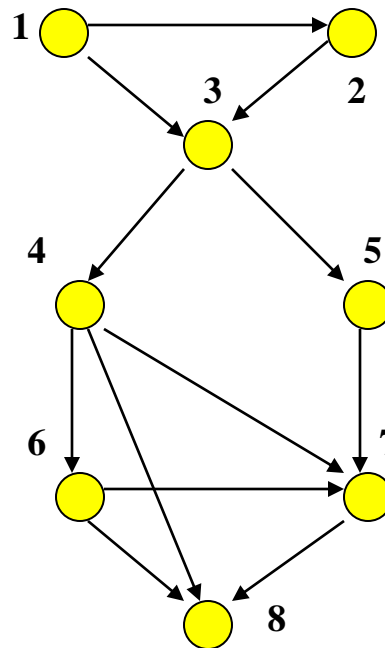
*“Максимальная” Подструктура
(по дугам)*



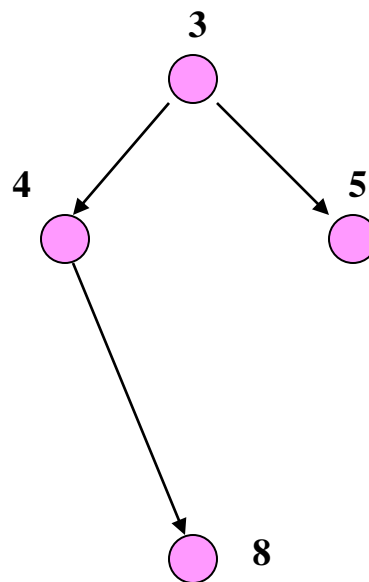
Подструктура и надструктура (иллюстрация): случай 2 орграфов



Минимальная Надструктура



Максимальная Подструктура



- ПРИЛОЖЕНИЯ:**
- 1.Принятие решений / Экспертные суждения (отношение доминирования)**
 - 2.Информационные структуры (базы данных)**
 - 3.Информационные структуры (базы знаний)**
 - 4.Биоинформатика**
 - 5.Химические структуры**
 - 6.Сетевые системы (например, социальные сети, программы)**
 - 7.Образы, основанные на графовых моделях**
 - 8.Изображения (графовые модели для изображений)**
 - 9.Лингвистика**
 - 10.Организационные структуры**
 - 11.Инженерные системы**
 - 12.Архитектура**
 - 13.Поиск информации**
 - 14.Распознавание образов**
 - 15.Близость графо-подобных систем**
- Др.

Подструктура и надструктура (иллюстрация)

ДРУГИЕ ВЕРСИИ ЗАДАЧИ:

- *СЛУЧАЙ N ГРАФОВ (БИНАРНЫХ ОТНОШЕНИЙ)**
- *СЛУЧАЙ ВЗВЕШЕННЫХ ГРАФОВ**
- *СЛУЧАЙ СО СПЕЦИАЛЬНЫМИ ОГРАНИЧЕНИЯМИ**
- *ДР.**

ПРИЛОЖЕНИЯ:

1. Расписания в образовании
(университеты, школы)
2. Расписания в медицине
3. расписания в спорте
(например, баскетбол)

ДР.

СОСТАВНЫЕ АЛГОРИТМИЧЕСКИЕ СХЕМЫ

На основе комбинации моделей:

1. Раскраска графов
 2. Назначение / Размещение
 3. Планирование эксперимента (Combinatorial design)
 4. Традиционные модели теории расписаний
- Др.

ЛЕКЦИЯ 19. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

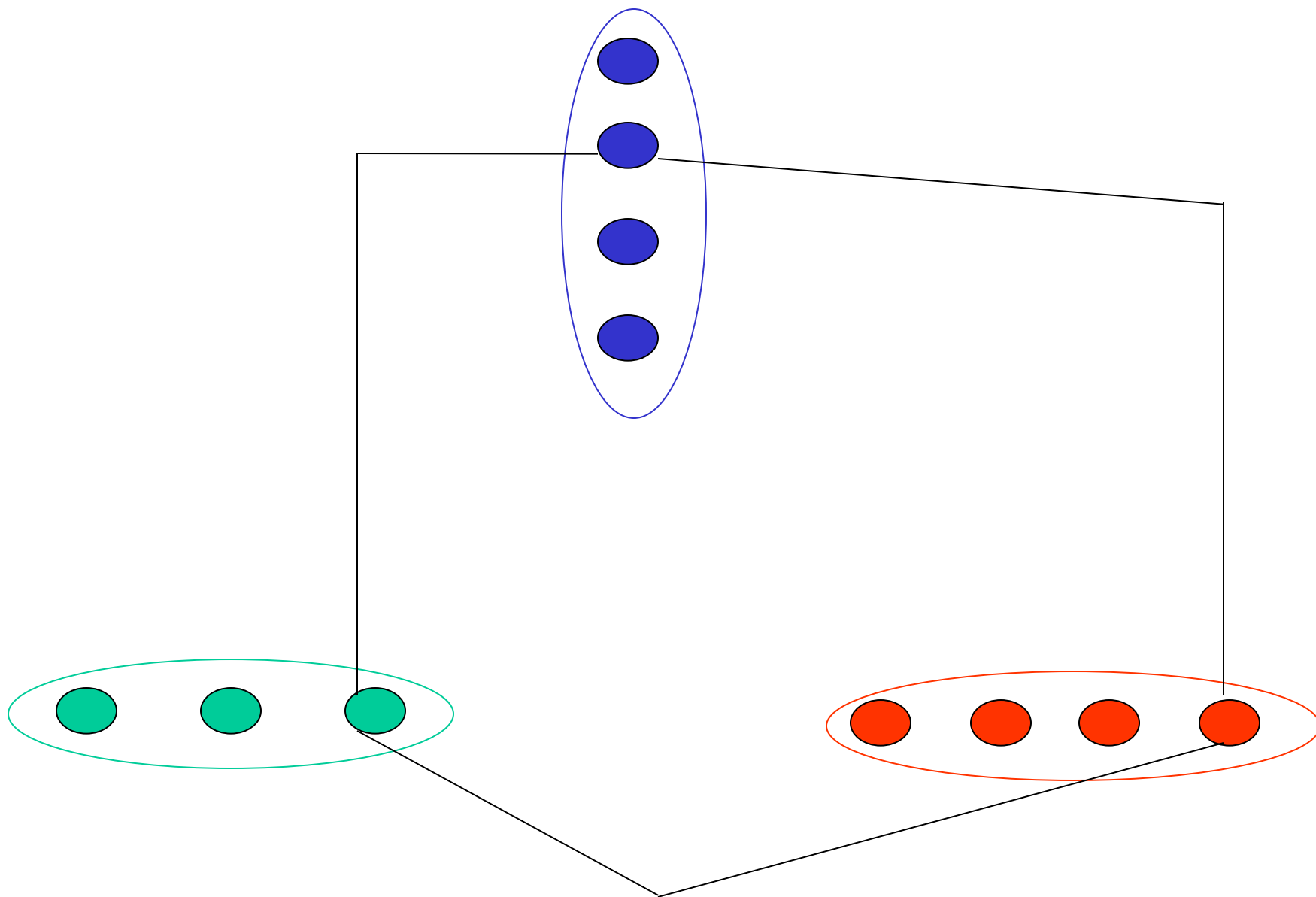
Л.19. Морфологический синтез

ПЛАН:

1. Морфологический анализ
2. Иерархическое морфологическое проектирование (HMMD)
(морфологический синтез, комбинаторный синтез): Основы
3. Пять стадий развития морфологического анализа
4. Предварительные стадии (1, 2, 3, 4)
5. HMMD: *формулировка, *схемы решения, *примеры

Окт. 15, 2004

Морфологическая клика



1. Морфологический анализ (F. Zwicky, 1943)

2. Парадигма принятия решений (H. Simon)

3. Динамическое программирование (R. Bellman)

4. Инженерный опыт иерархического проектирования
сложных многодисциплинарных систем

5. Комбинаторная оптимизация

6. Технология знаний (многодисциплинарная информация): выявление, организация, использование

1. Морфологический анализ [F. Zwicky]

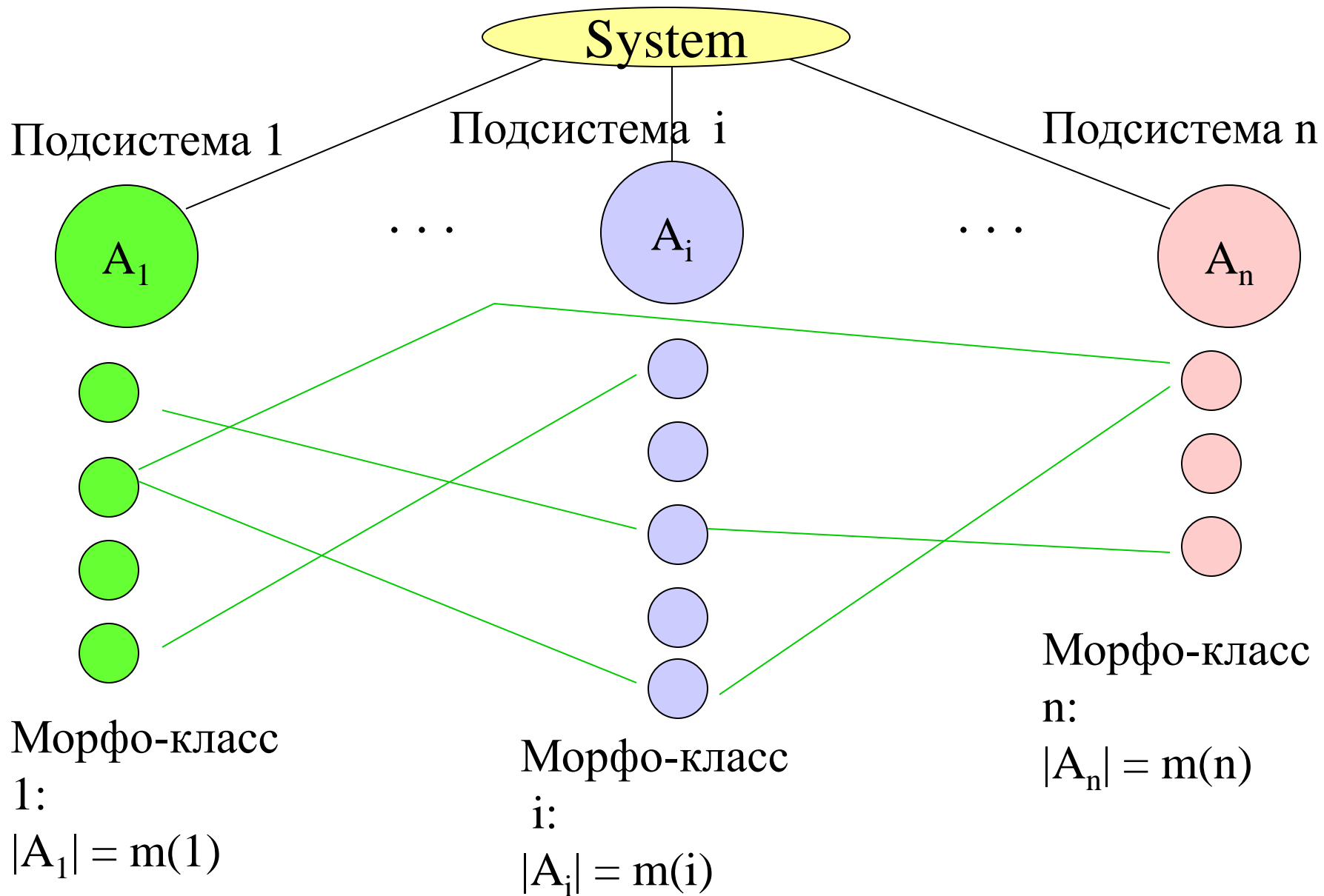
2. Близость допустимых комбинаций к “ИДЕАЛУ”
[Ayres, 1969; Якимец, Ин-т системного анализа РАН, 1977]

3. Многокритериальная оценка допустимых комбинаций
и выбор по Парето [Ин-т Проблем Управления ИПУ,
Ин-т системного анализа, Ин-т машиноведения ИМАШ,
РАН, 1972/82]

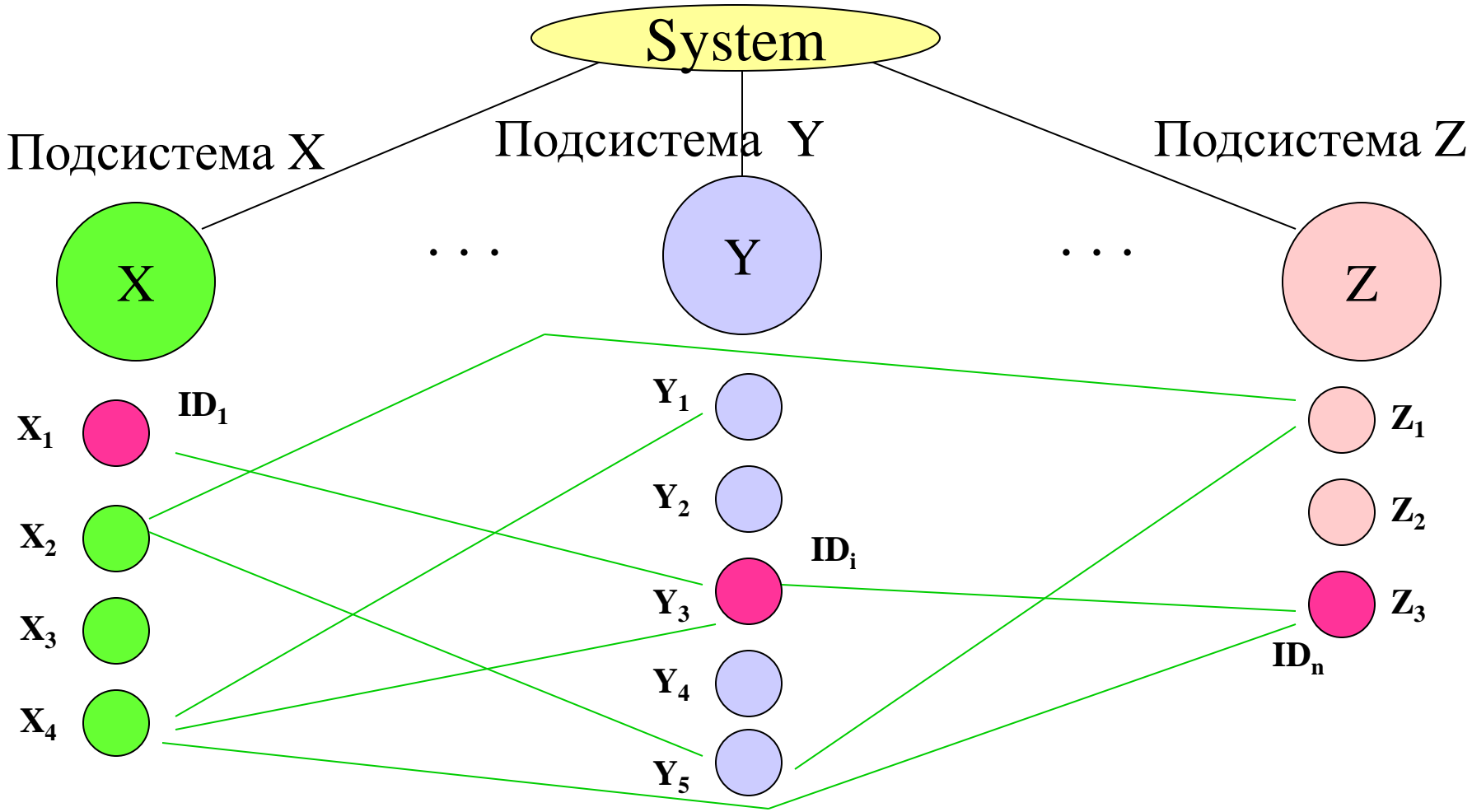
4. Иерархическое проектирование (композиция локально
оптимальных по Парето решений [Краснощеков и др.,
Выч. Центр РАН, 1979]

5. Иерархическое морфологическое проектирование
(комбинаторный морфологический синтез) [Левин, 1994...]

Морфологический анализ



Морфологический анализ: Близость к “ИДЕАЛУ” (Ayres, 1969)

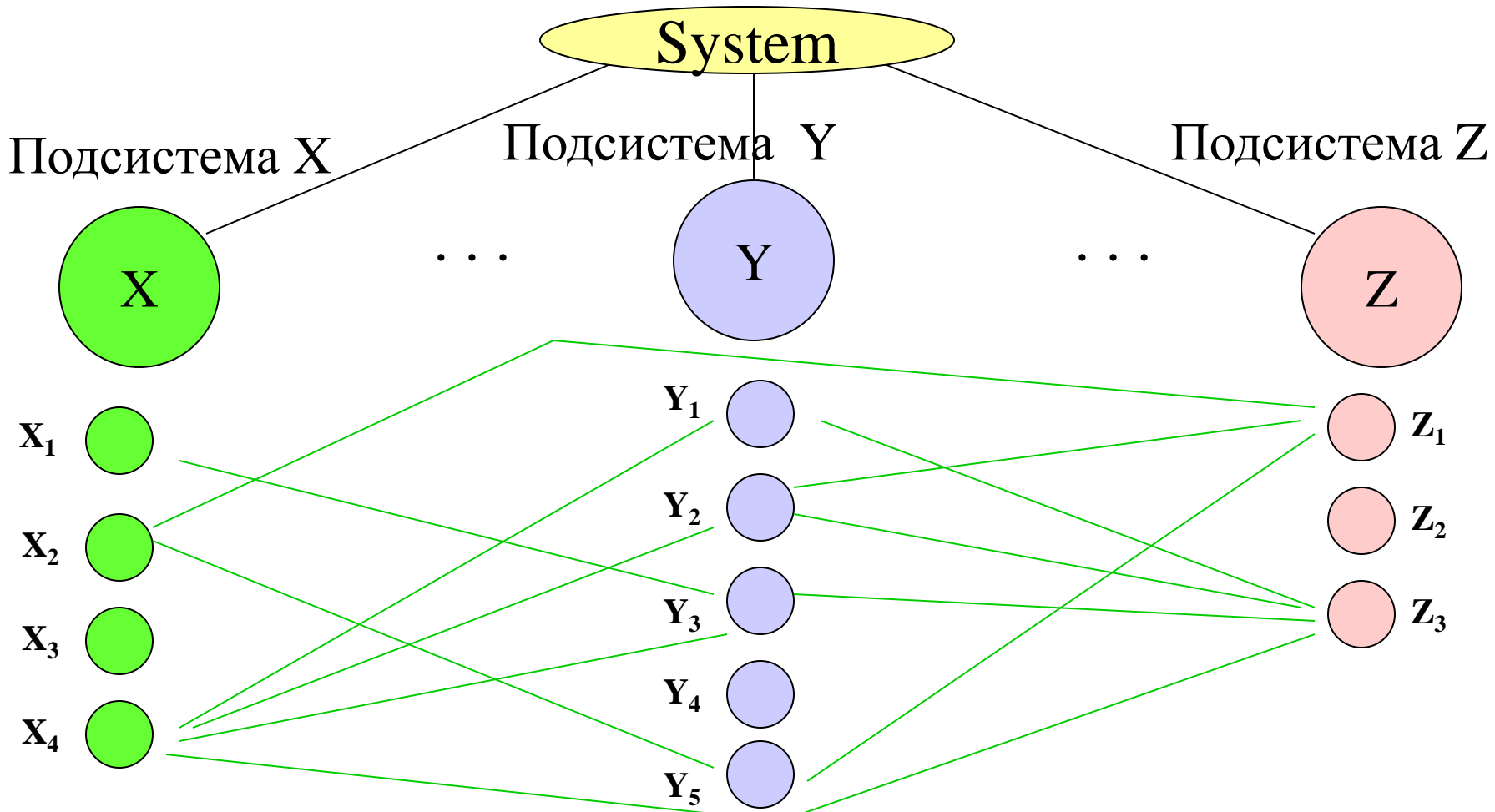


“ИДЕАЛ” : X₁ * Y₃ * Z₃
 (недопустимая комбинация)

S₁ = X₄ * Y₃ * Z₃
 S₂ = X₂ * Y₅ * Z₁

ρ (“ИДЕАЛ”, S₁) < ρ (“ИДЕАЛ”, S₂)
 ρ - близость

Многокритериальная оценка допустимых комбинаций & выбор по Парето (1972..1982)



ШАГ 1. Генерация допустимых комбинаций:

$$S_1 = X_4 * Y_3 * Z_3 \quad S_3 = X_4 * Y_2 * Z_3$$

$$S_2 = X_2 * Y_5 * Z_1 \quad S_4 = X_4 * Y_1 * Z_3$$

ШАГ 2. Оценка по критериям

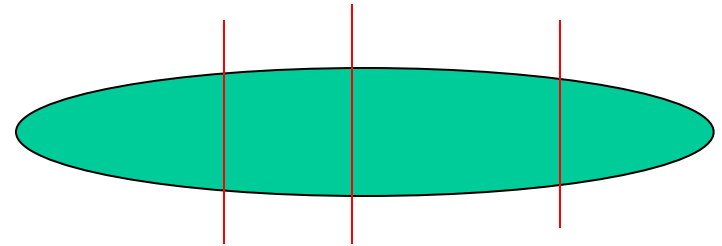
ШАГ 3. Выбор по Парето

Морфологический анализ

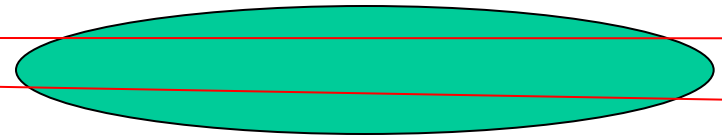
Сложность: $m(1) * \dots * m(i) * \dots * m(n)$

Уменьшение сложности:

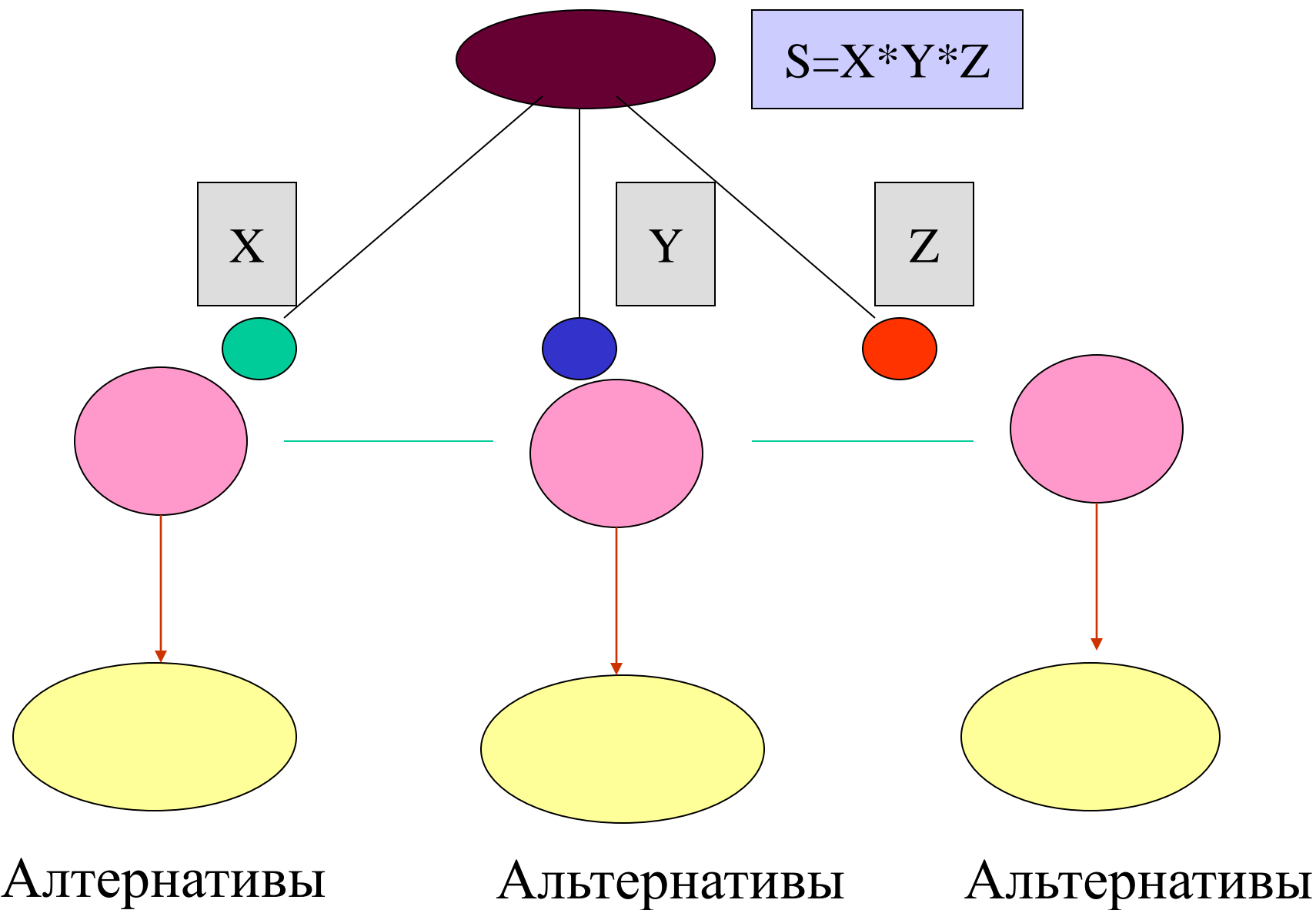
Горизонтальная декомпозиция



Вертикальная декомпозиция



Иерархическое проектирование (инженерный опыт, фазы 3, 4)

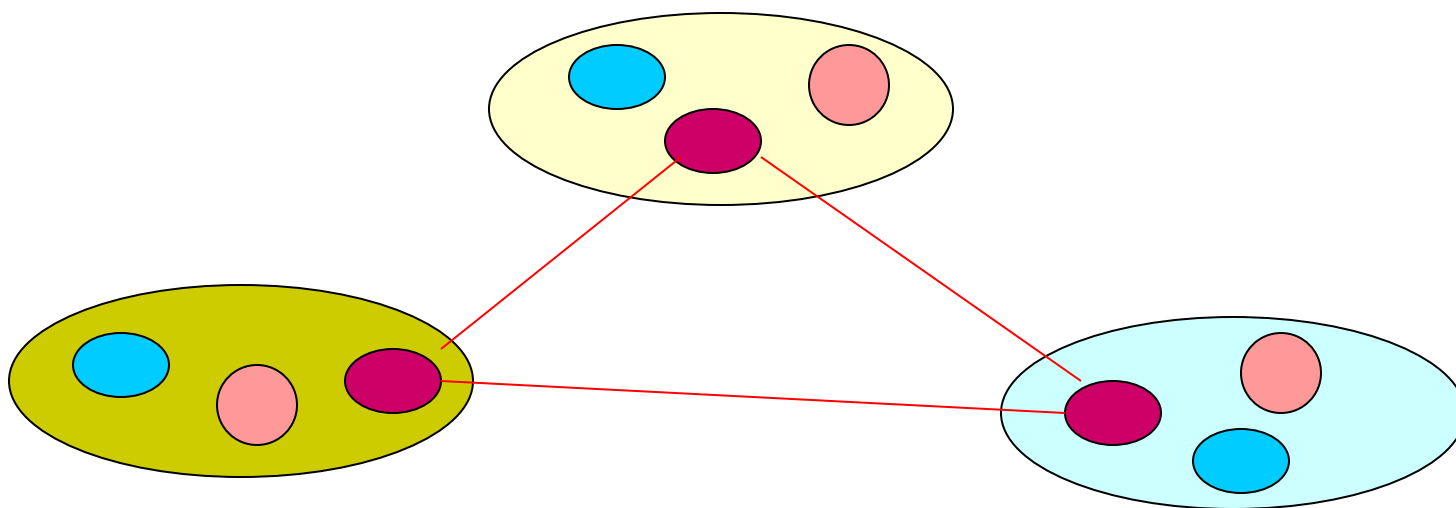
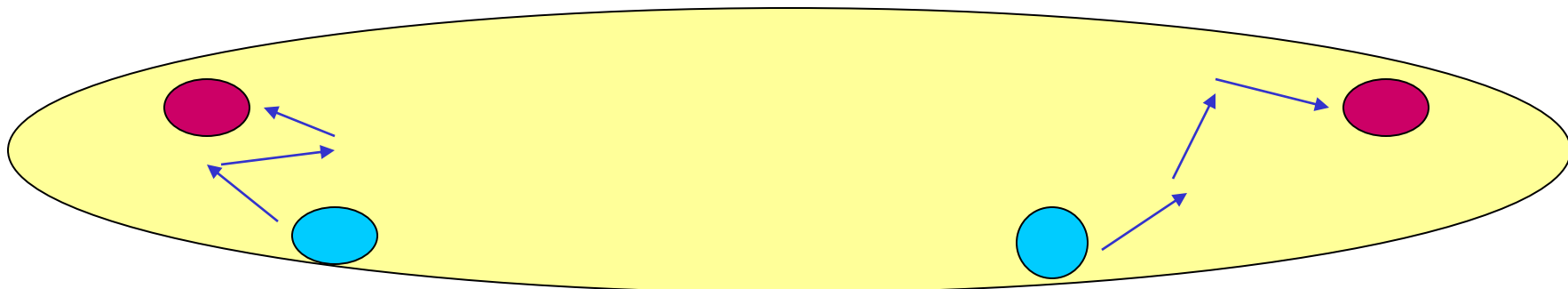


Поисковые стратегии

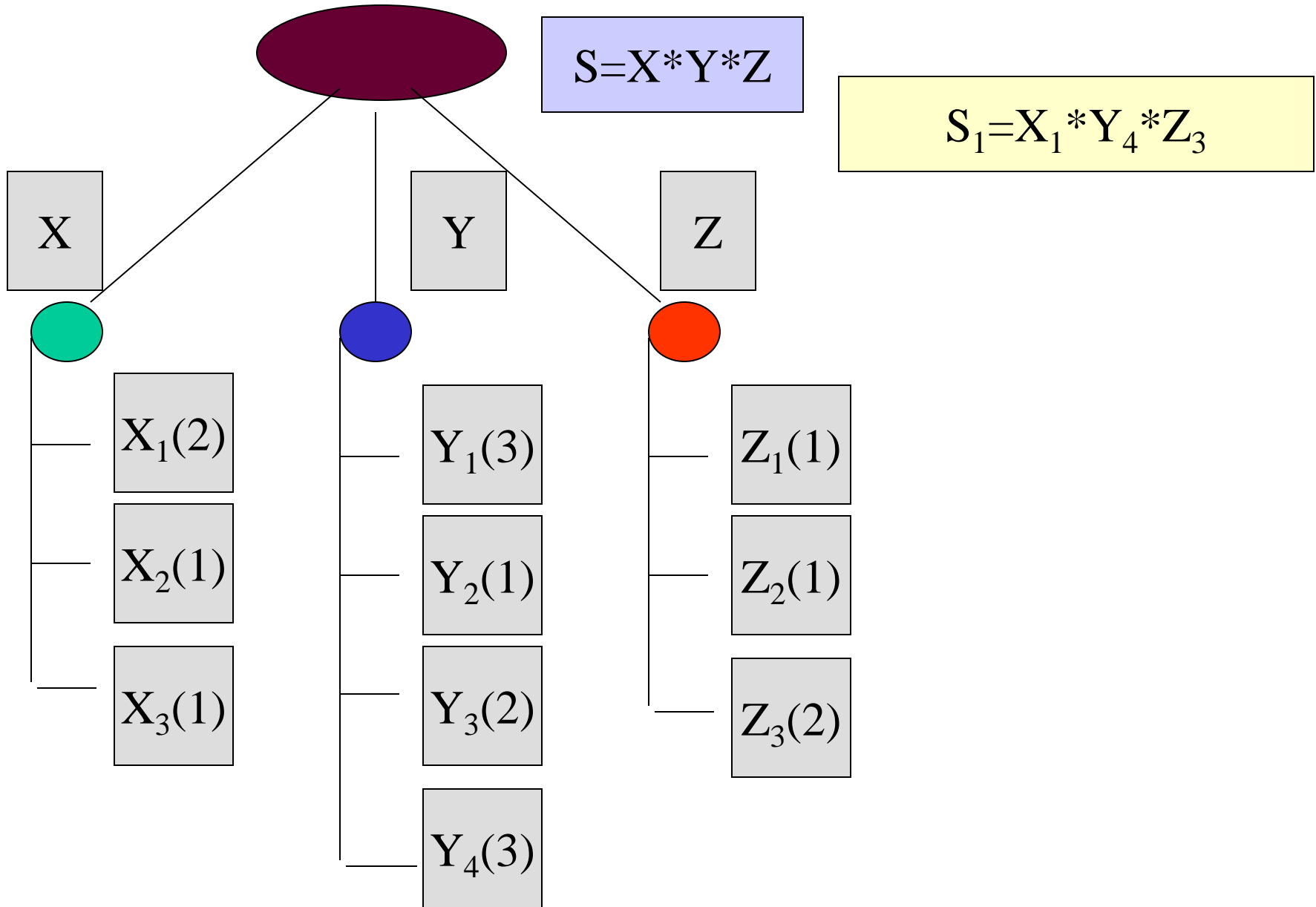
Точка
НАЧАЛО

Точка
ОПТИМУМА

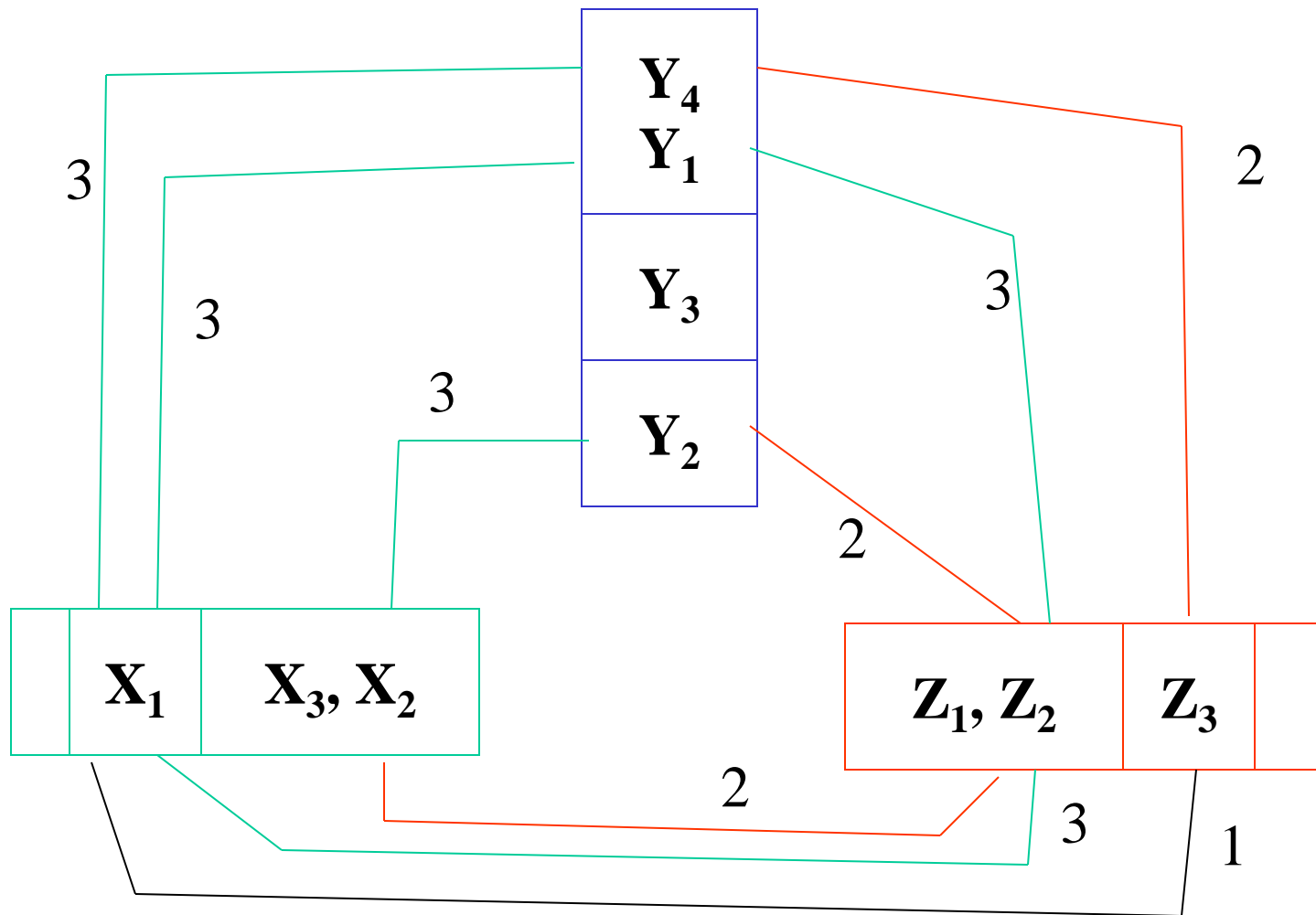
Пространство поиска



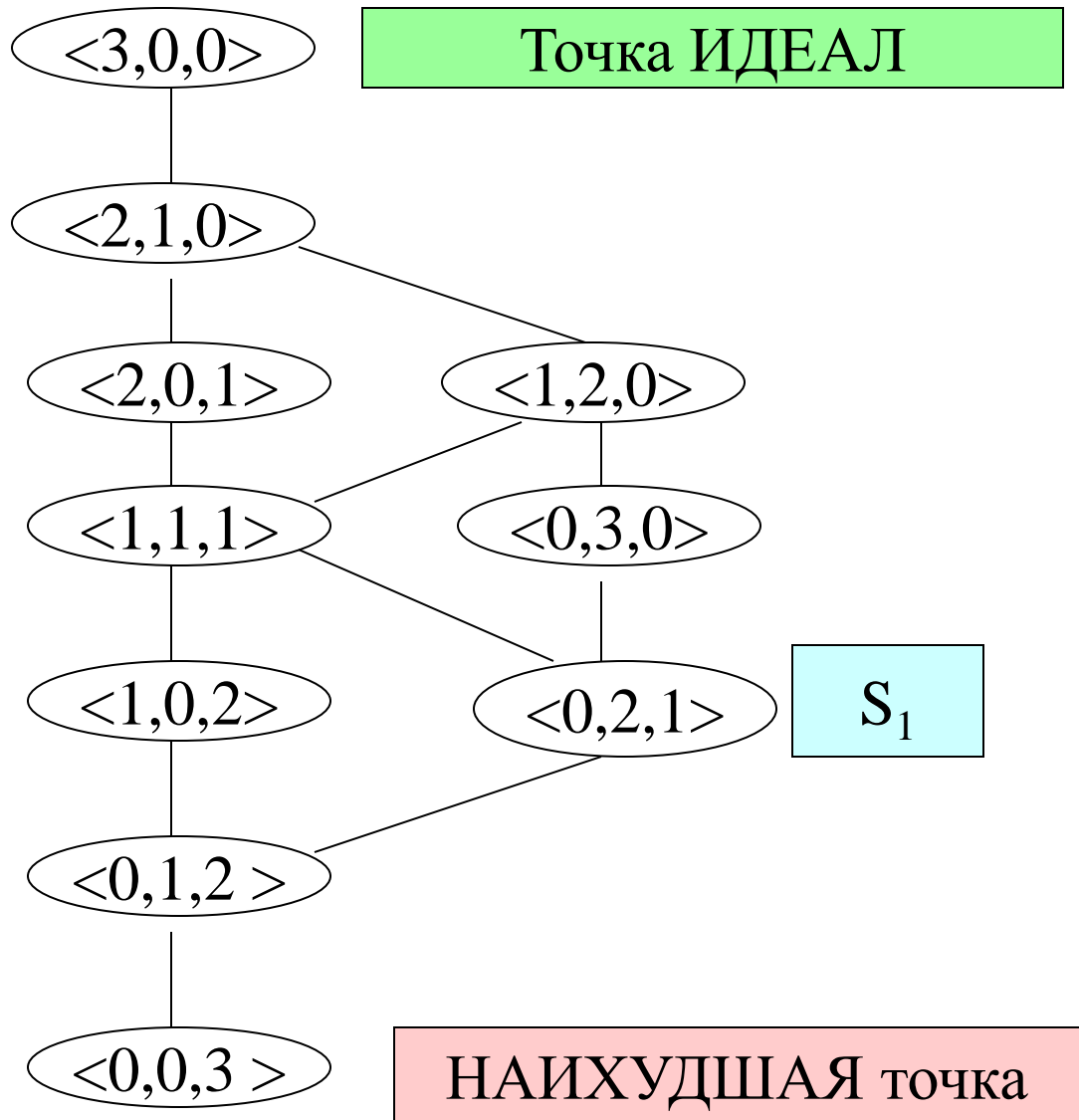
Морфологический комбинаторный синтез: пример



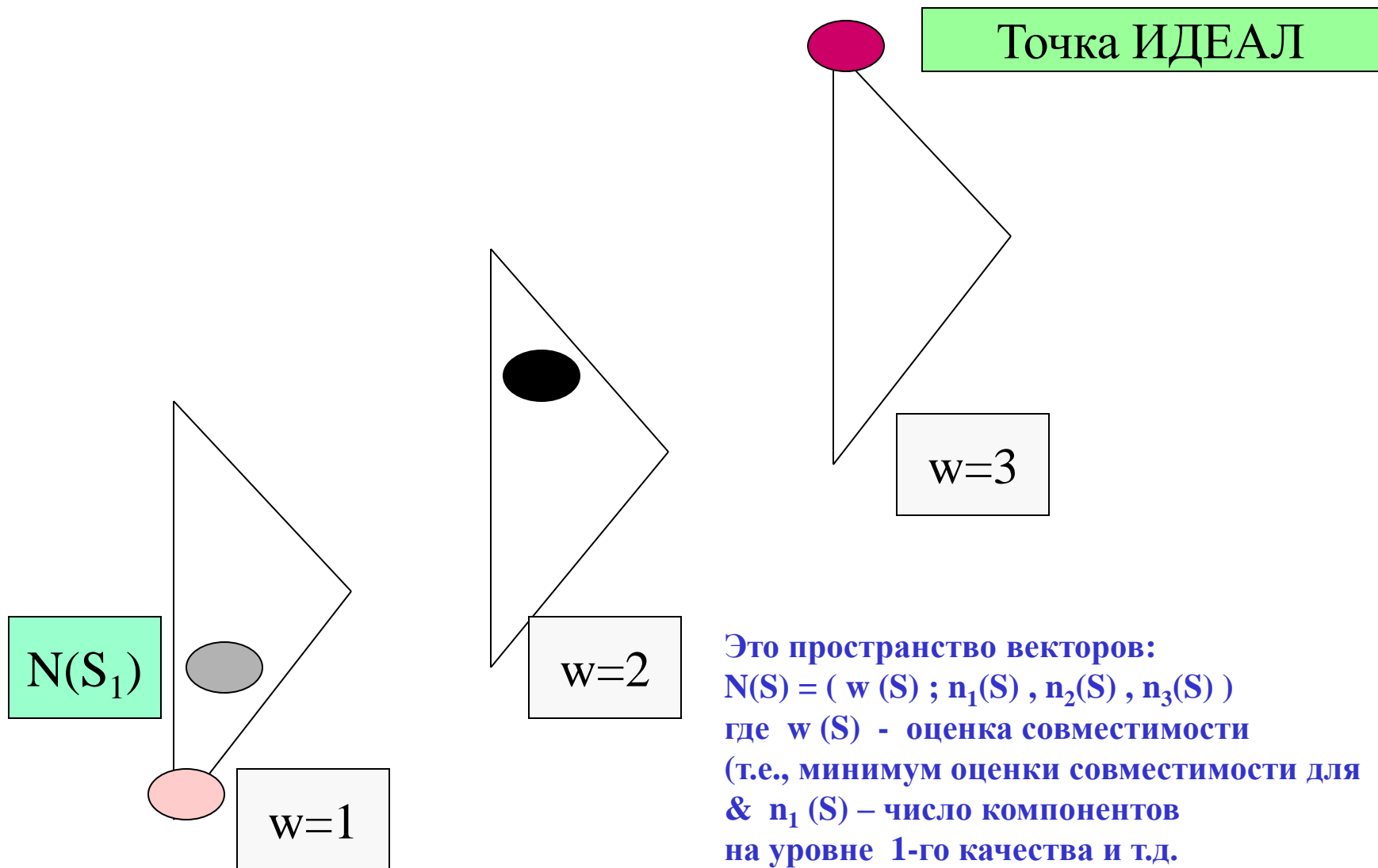
Центрическое представление морфологической клики с оценками совместимости



Дискретное пространство качества (по элементам)

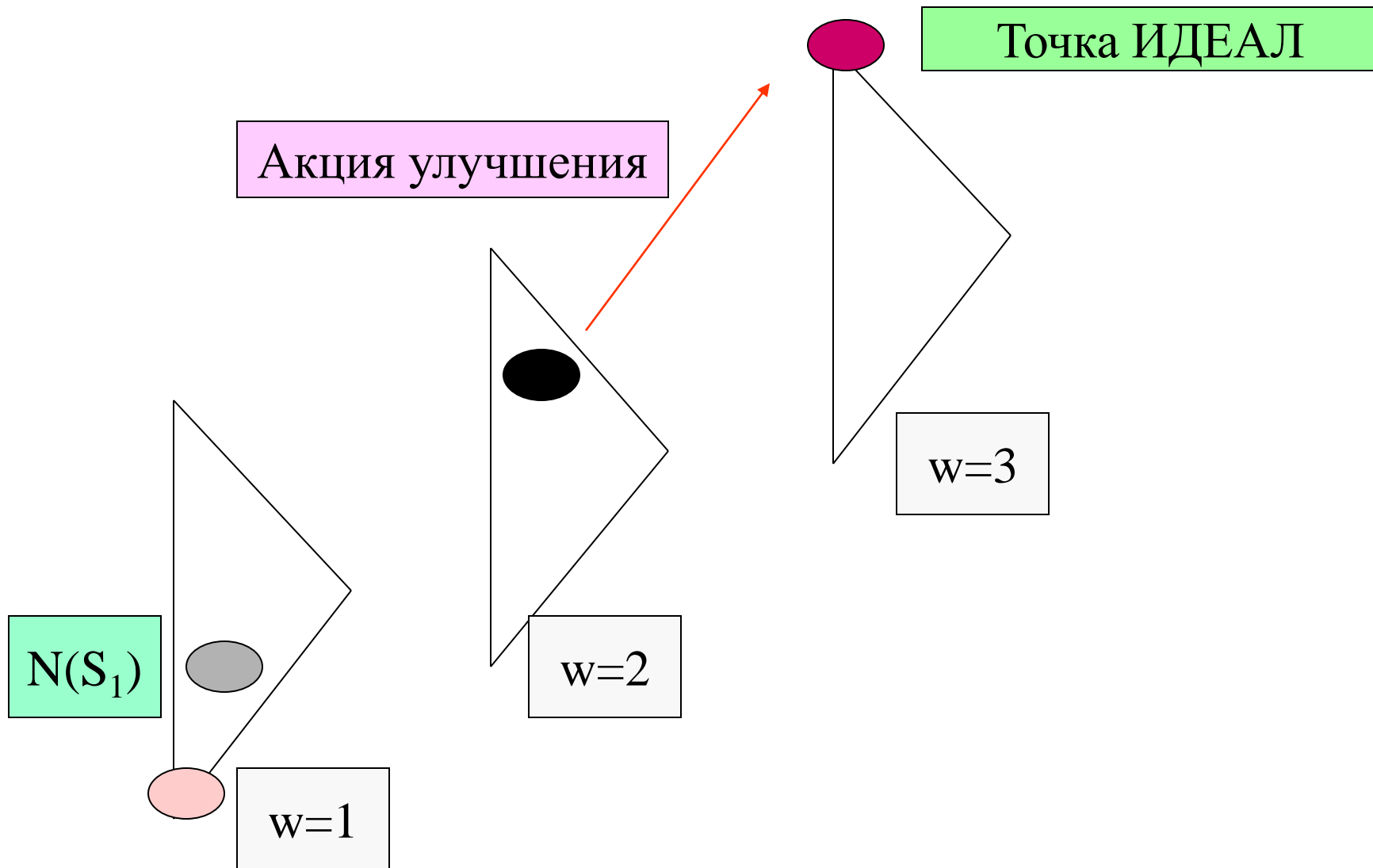


Дискретное пространство качества (по элементам, по совместимости)



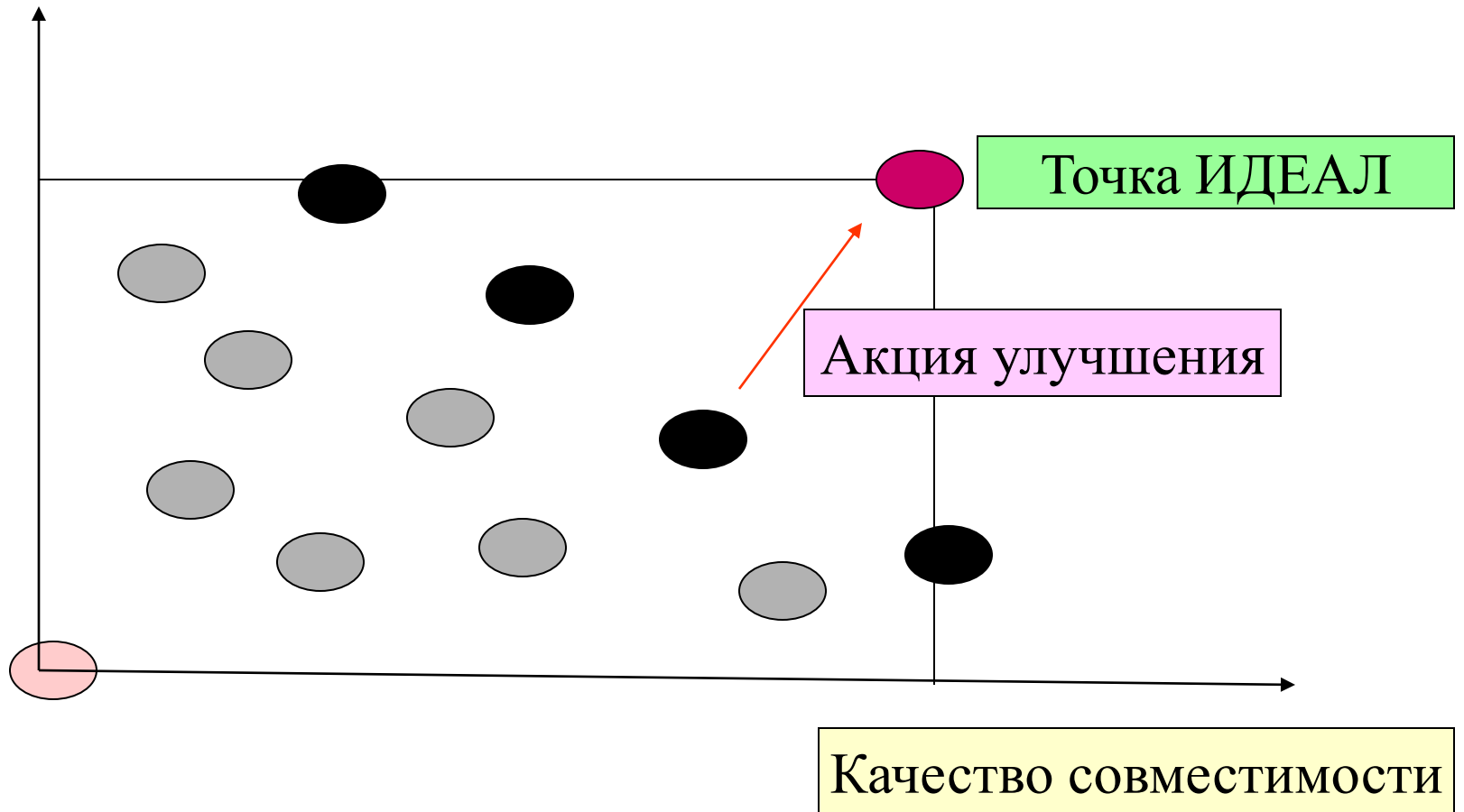
Это пространство векторов:
 $N(S) = (w(S) ; n_1(S) , n_2(S) , n_3(S))$
где $w(S)$ - оценка совместимости
(т.е., минимум оценки совместимости для пар
& $n_1(S)$ - число компонентов
на уровне 1-го качества и т.д.

Дискретное пространство качества (по элементам, по совместимости) & улучшение



Пространство качества на основе 2-х критериев & акция улучшения

Качество элементов

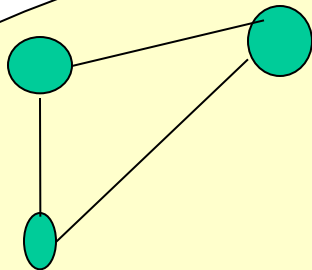


Качество совместимости

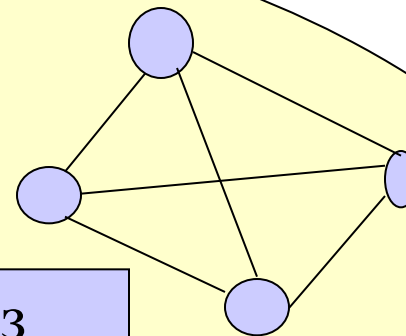
1. Переборная направленная эвристика
анализ и проверка
начиная с идеальной точки.

2. Динамическое программирование:
расширенный вариант метода для задачи
рюкзаке или блочного рюкзака

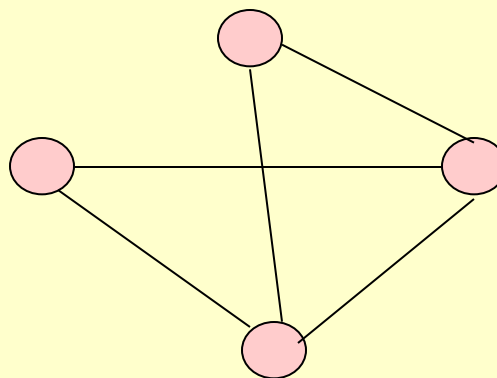
Примеры клики & квази-клики



Клика из
3-х элементов

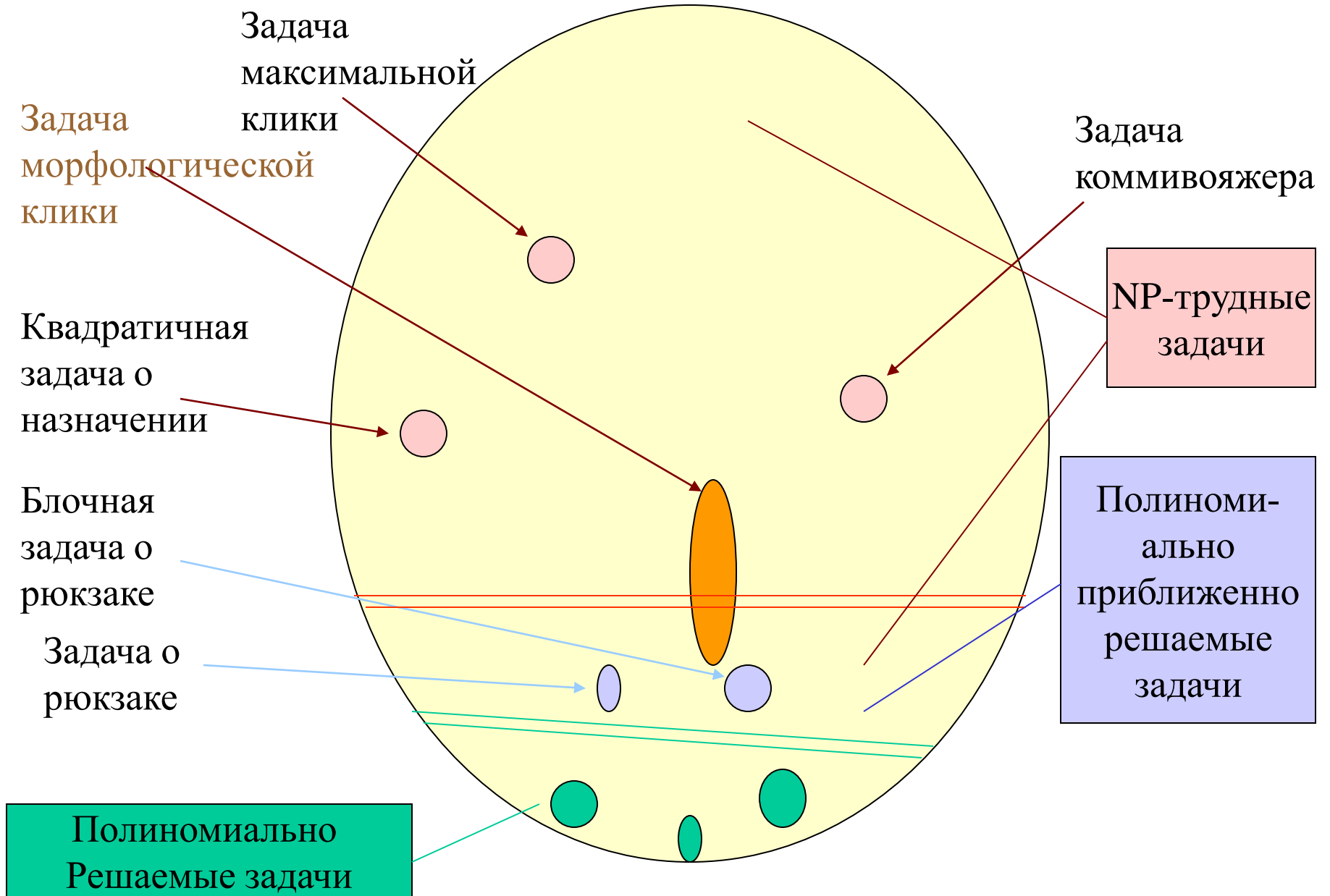


Клика из
4-х элементов



Квази-клика

«Пространство» задач комбинаторной оптимизации



1. Задача о рюкзаке

2. Задача блочного рюкзака

3. Квадратичная задача о назначении

**4. Целочисленное нелинейное
программирование**

**5. Смешанное целочисленное нелинейное
программирование**



ЛЕКЦИИ 20-21. курс: “Проектирование систем: структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.20. Применения морфологического синтеза

Л.21. «Узкие» места, улучшение, многостадийное проектирование

ПЛАН:

1 Иерархическое морфологическое проектирование (HMMD).

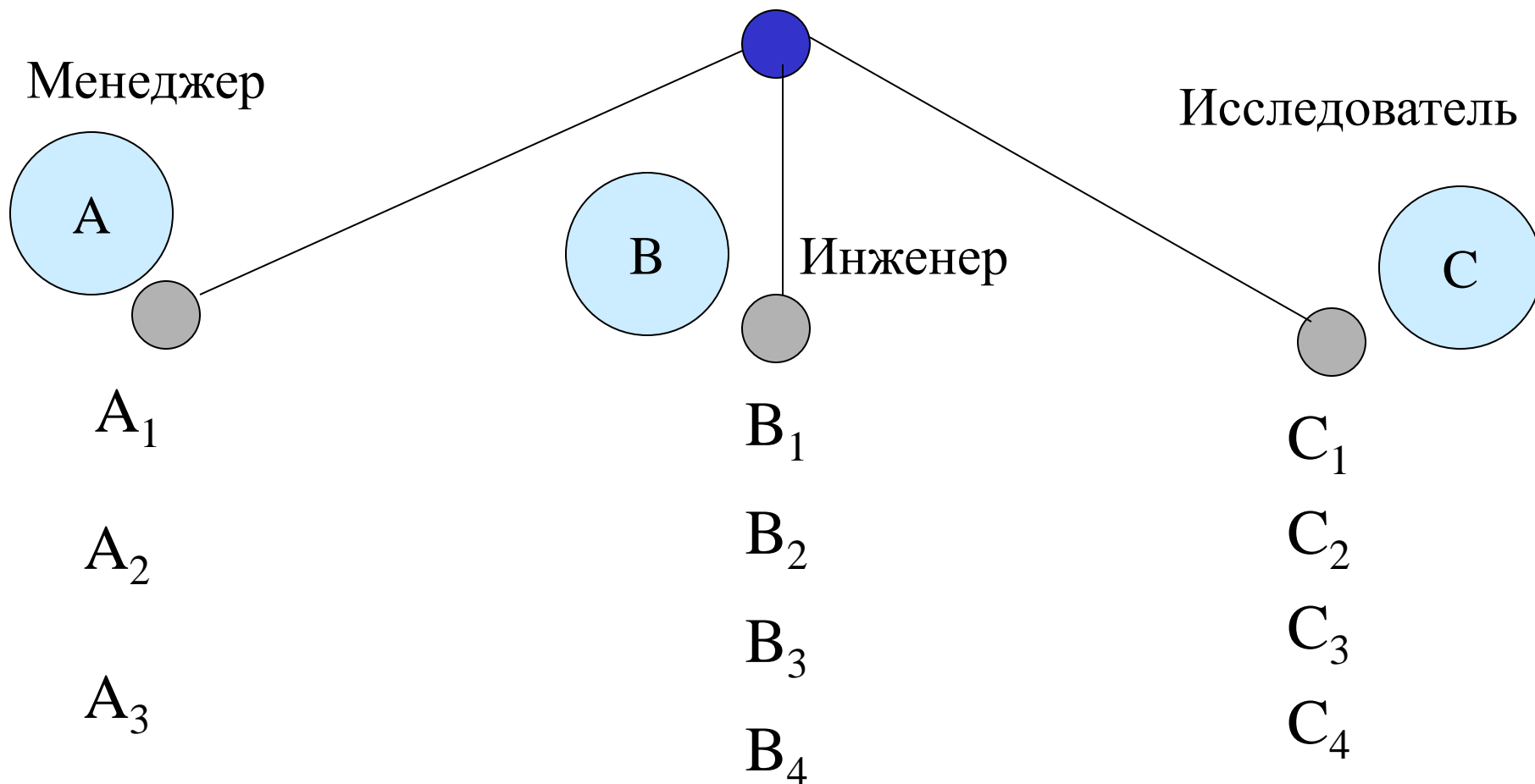
Примеры: *проектирование группы *проектирование стратегии *

2. Подходы к выявлению узких мест

3. Многостадийное проектирование

Прикладной пример 1: Проектирование группы (результатирующее решение)

Группа: $S = A * B * C$



Прикладной пример 1: Проектирование группы (результатирующее решение)

	Опыт	Рынок	Зарплата	Общая оценка
A_1	3 5	4 3	-1 10	3
A_2	15	5	20	1
A_3	10	4	15	2

Прикладной пример 1: Проектирование группы (результатирующее решение)

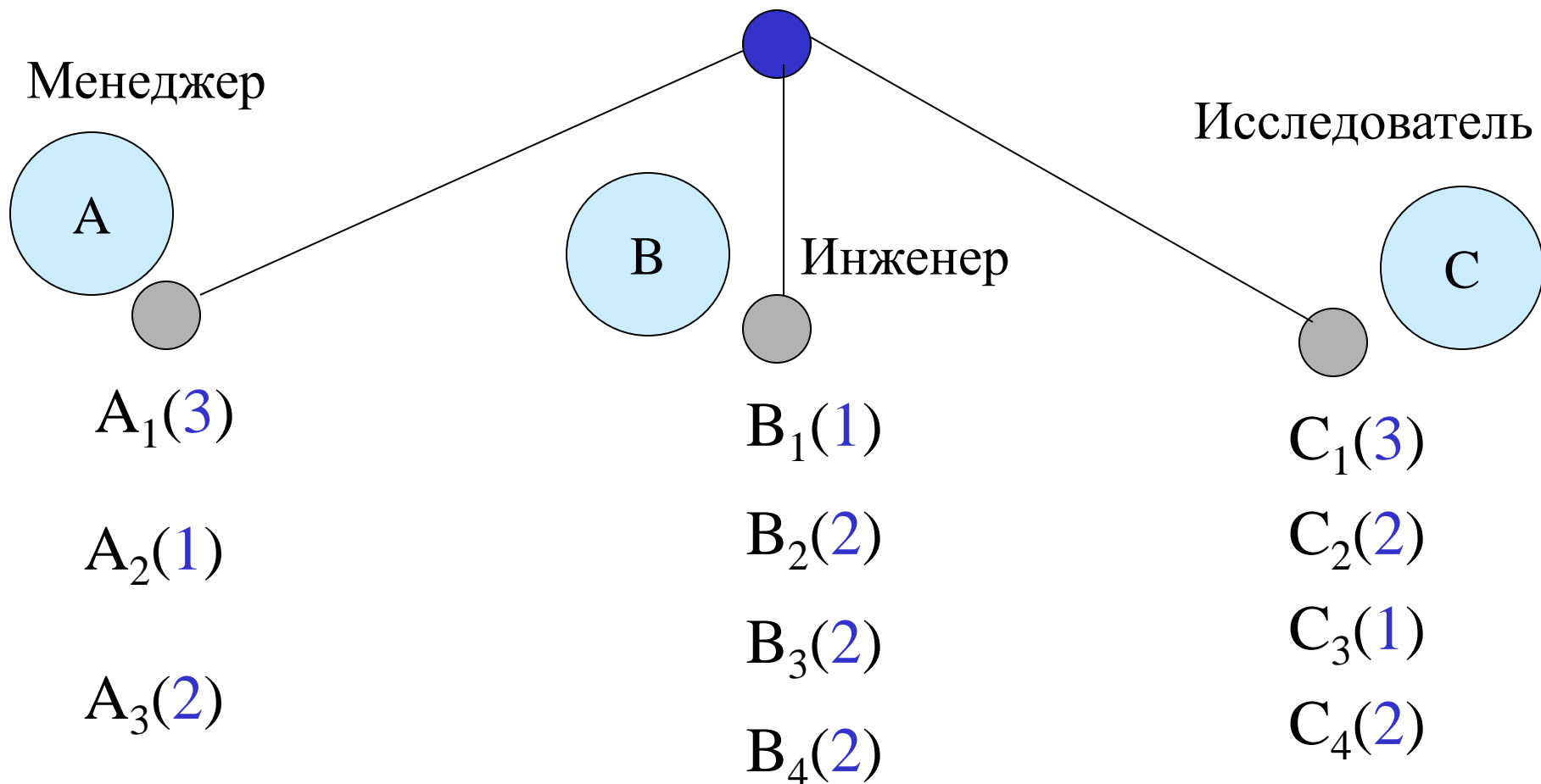
	Опыт	Западный опыт	Зарплата	Общая оценка
V_1	3 15	2 5	-3 7	1
V_2	6	6	3	2
V_3	10	0	9	2
V_4	3	0	3	2

Прикладной пример 1: Проектирование группы (результатирующее решение)

	Опыт	Творчество	Зарплата	Общая оценка
C_1	2 15	4 3	-3 10	3
C_2	5	5	7	2
C_3	4	7	4	1
C_4	6	4	6	2

Прикладной пример 1: Проектирование группы (результатирующее решение)

Группа: $S = A * B * C$

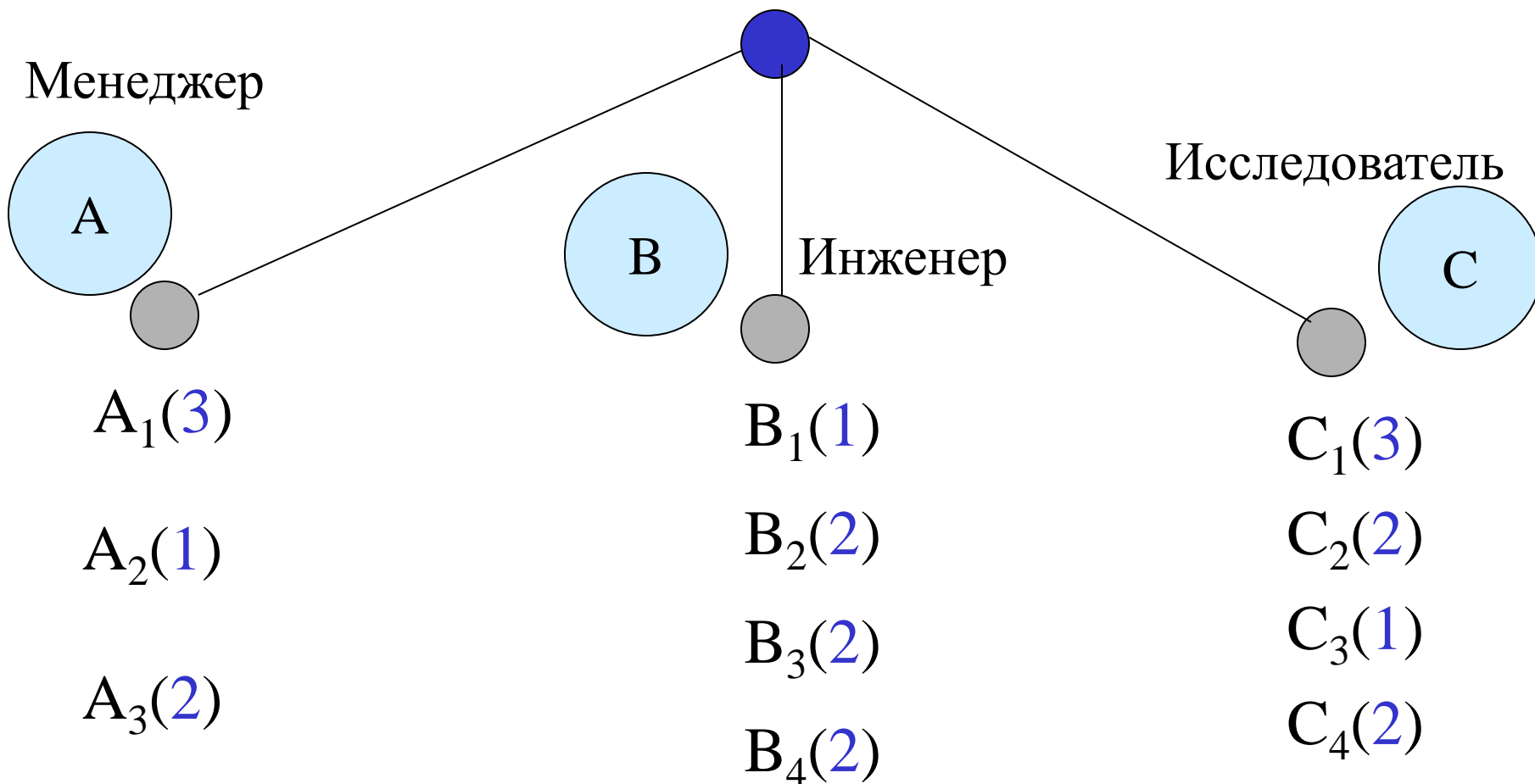


Прикладной пример 1: Проектирование группы (результатирующее решение)

	B_1	B_2	B_3	B_4	C_1	C_2	C_3	C_4
A_1	2	3	3	1	2	0	0	3
A_2	3	0	0	1	0	3	0	0
A_3	3	0	1	1	1	0	3	0
B_1					0	3	3	0
B_2					1	1	0	3
B_3					0	0	2	0
B_4					0	0	0	3

Прикладной пример 1: Проектирование группы (результатирующее решение)

Группа: $S = A * B * C$



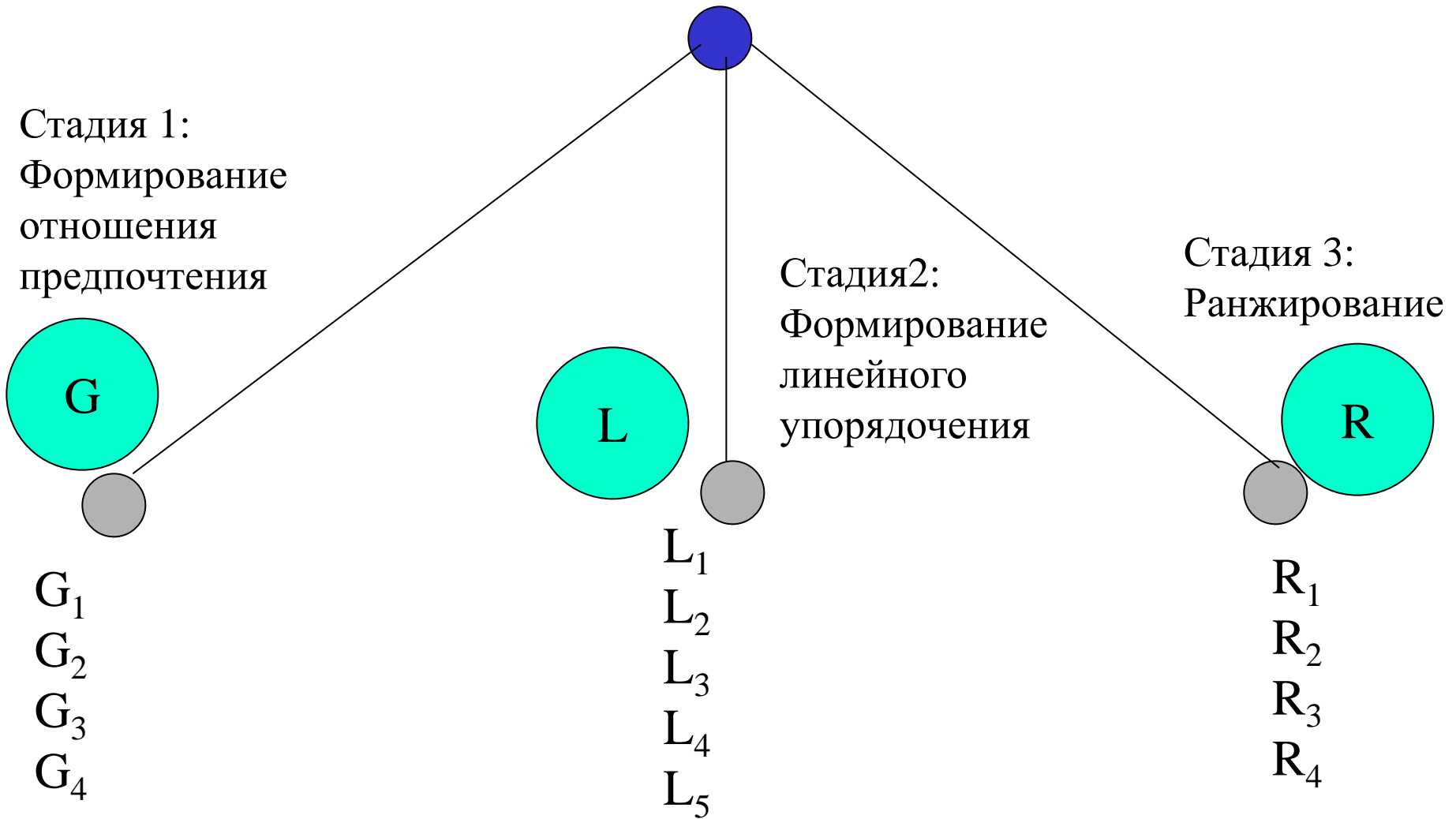
Прикладной пример 1: Проектирование группы (результатирующее решение)

$$S_1 = A_2 * B_1 * C_2 \quad N(S_1) = (3; 2, 1, 0)$$

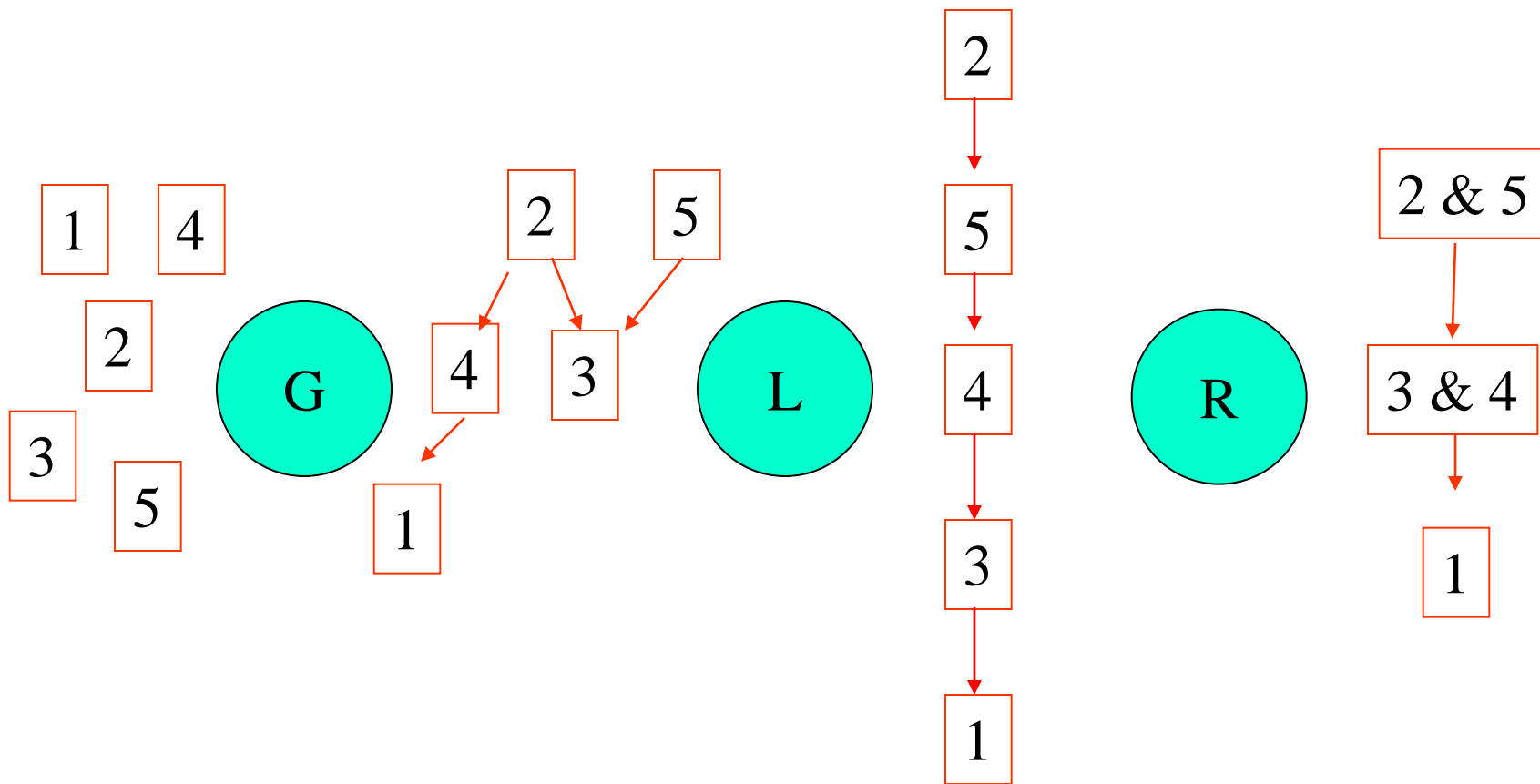
$$S_2 = A_3 * B_1 * C_3 \quad N(S_2) = (3; 2, 1, 0)$$

Прикладной пример 2: Стратегии для многокритериального ранжирования

Стратегия: $S = G * L * R$



Прикладной пример 2: Стратегии для многокритериального ранжирования

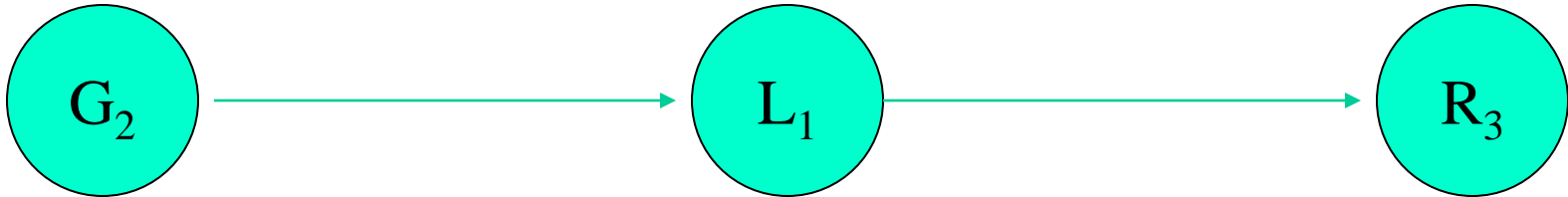


ЛОКАЛЬНЫЕ АЛЬТЕРНАТИВЫ (СОМБИ-РС, 1989...):

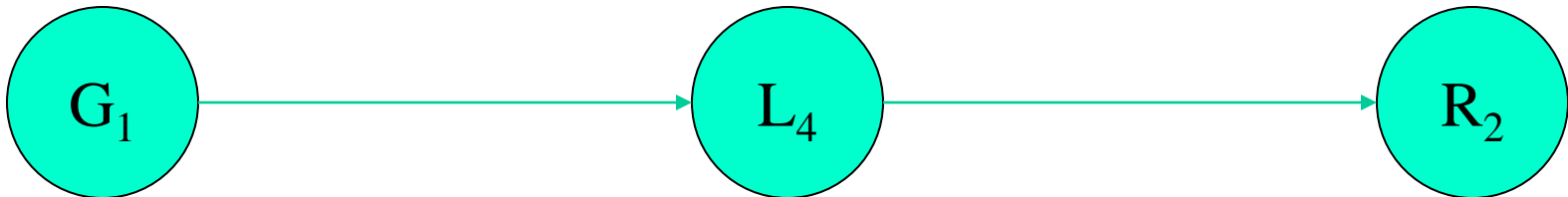
- G₁ Парные сравнения**
- G₂ метод ELECTRE**
- G₃ аддитивная функция полезности**
- G₄ экспертная стратификация**
- L₁ сумма элементов в матрице предпочтений**
- L₂ аддитивная функция полезности**
- L₃ последовательное выявление “max” элемента**
- L₄ последовательное выявление элементов Парето**
- L₅ экспертная стратификация**
- R₁ последовательное выявление “max” элемента**
- R₂ последовательное выявление элементов Парето**
- R₃ разбиение линейного упорядочения**
- R₄ экспертная стратификация**

ПРИМЕРЫ СТРАТЕГИЙ:

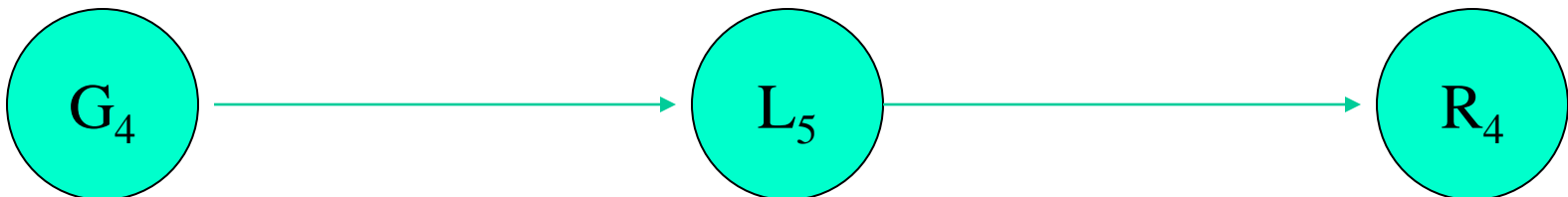
Стратегия 1



Стратегия 2

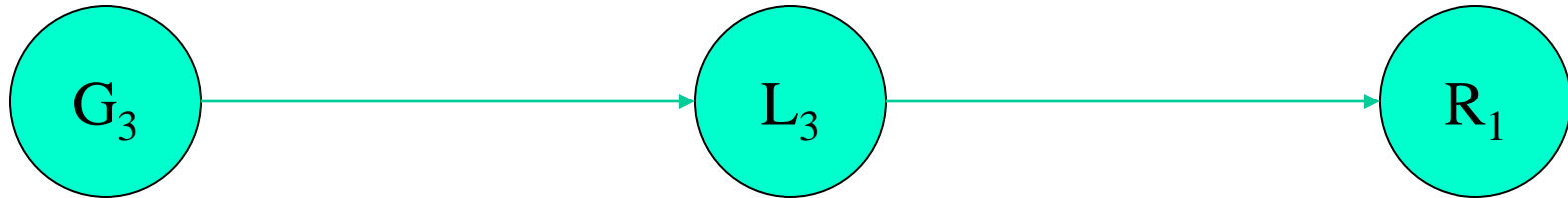


Стратегия 3

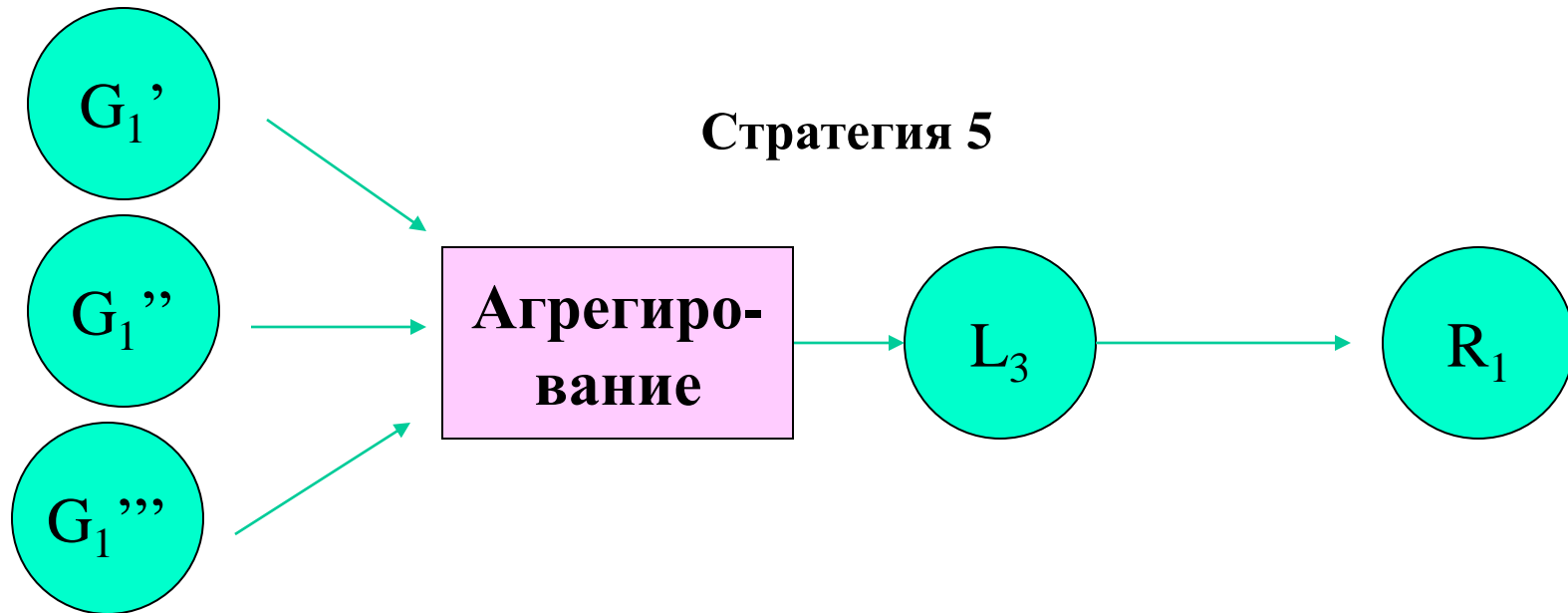


ПРИМЕРЫ СТРАТЕГИЙ:

Стратегия 4



Стратегия 5



Подход 1.

Инженерный анализ (экспертные суждения)

Подход 2.

**Построение структуры системы, оценка надежности компонентов, выбор наиболее ненадежных компонентов
(“Парето подход” из японской системы управления качеством)**

Подход 3.

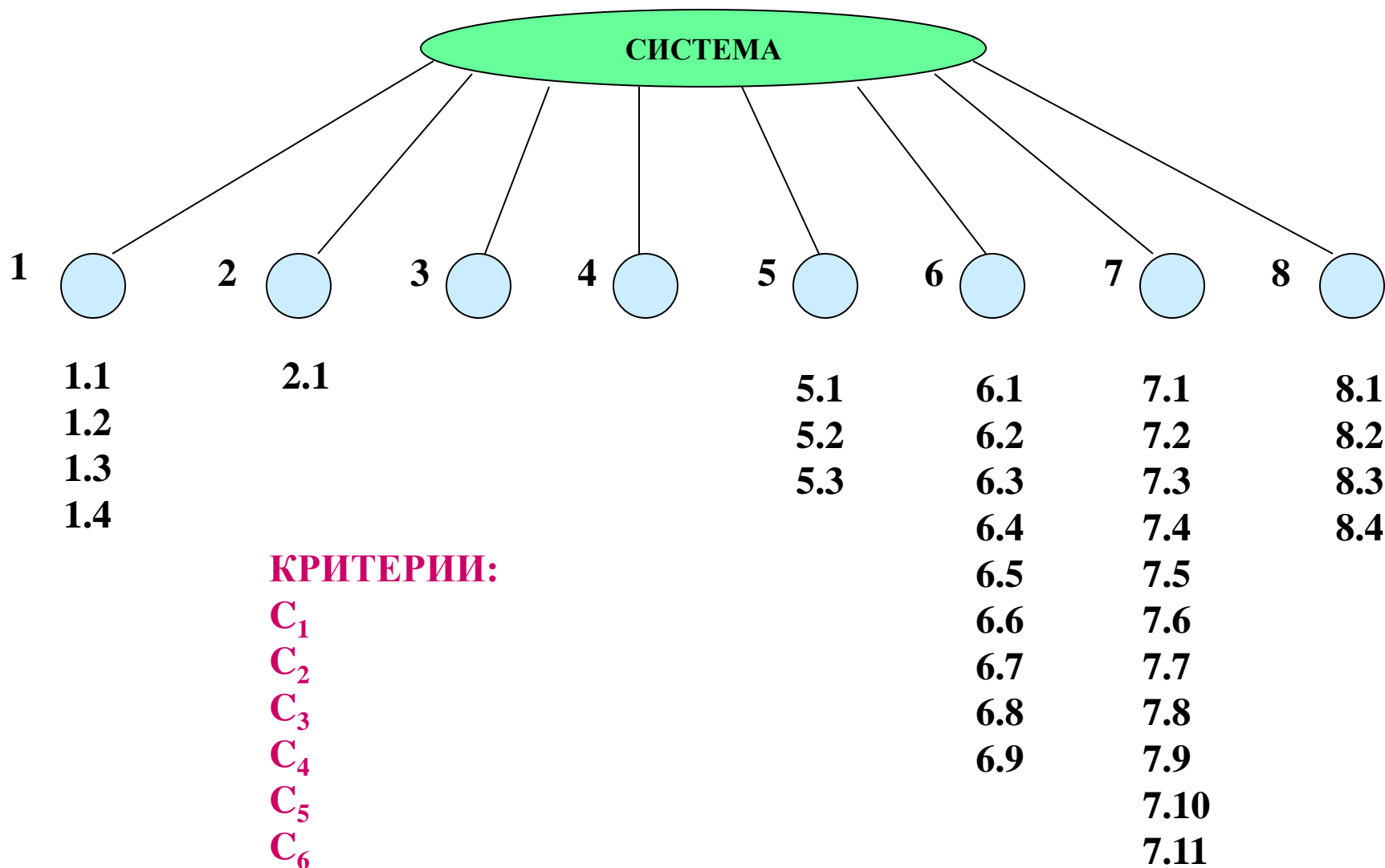
**Построение структуры системы, оценка надежности компонентов,
многокритериальное ранжирование компонентов
(для выявления наиболее важных компонентов)**

Подход 4.

Анализ общего вектора качества системы S :

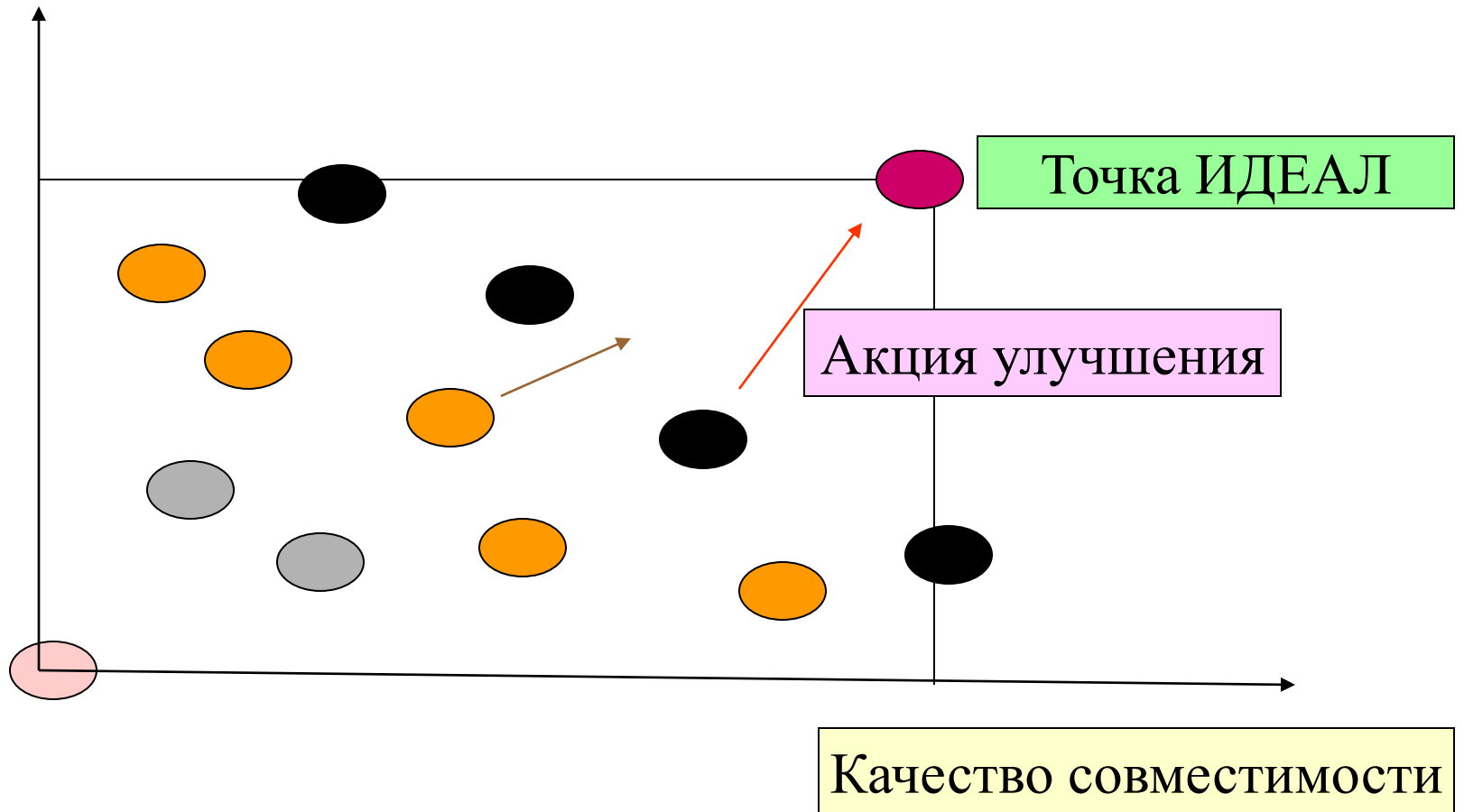
$$N(S) = (w(S); n_1(S), n_2(S), \dots)$$

Подходы к выявлению узких мест

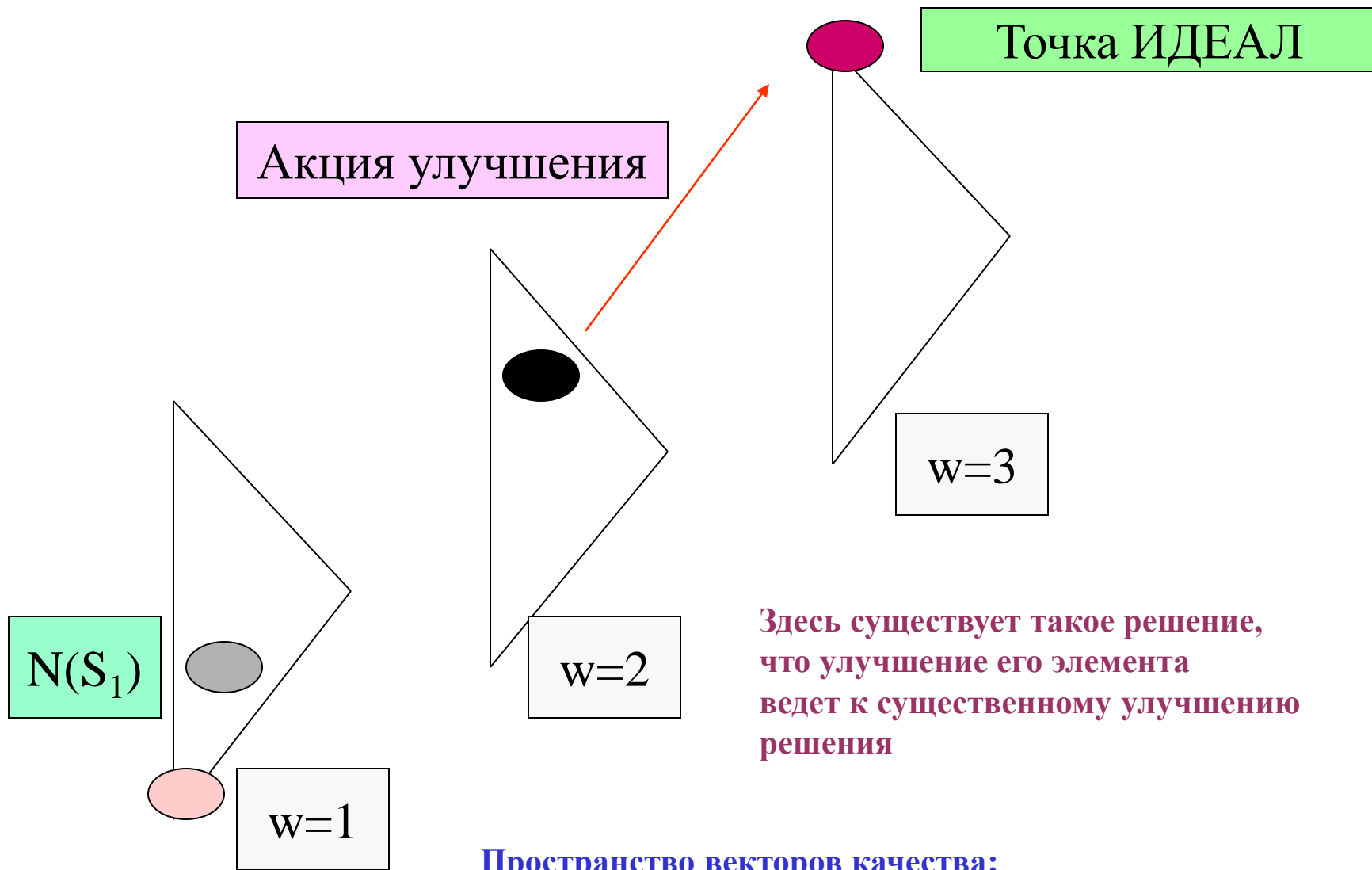


Парето-эффективные & почти Парето-эффективные решения

Качество элементов



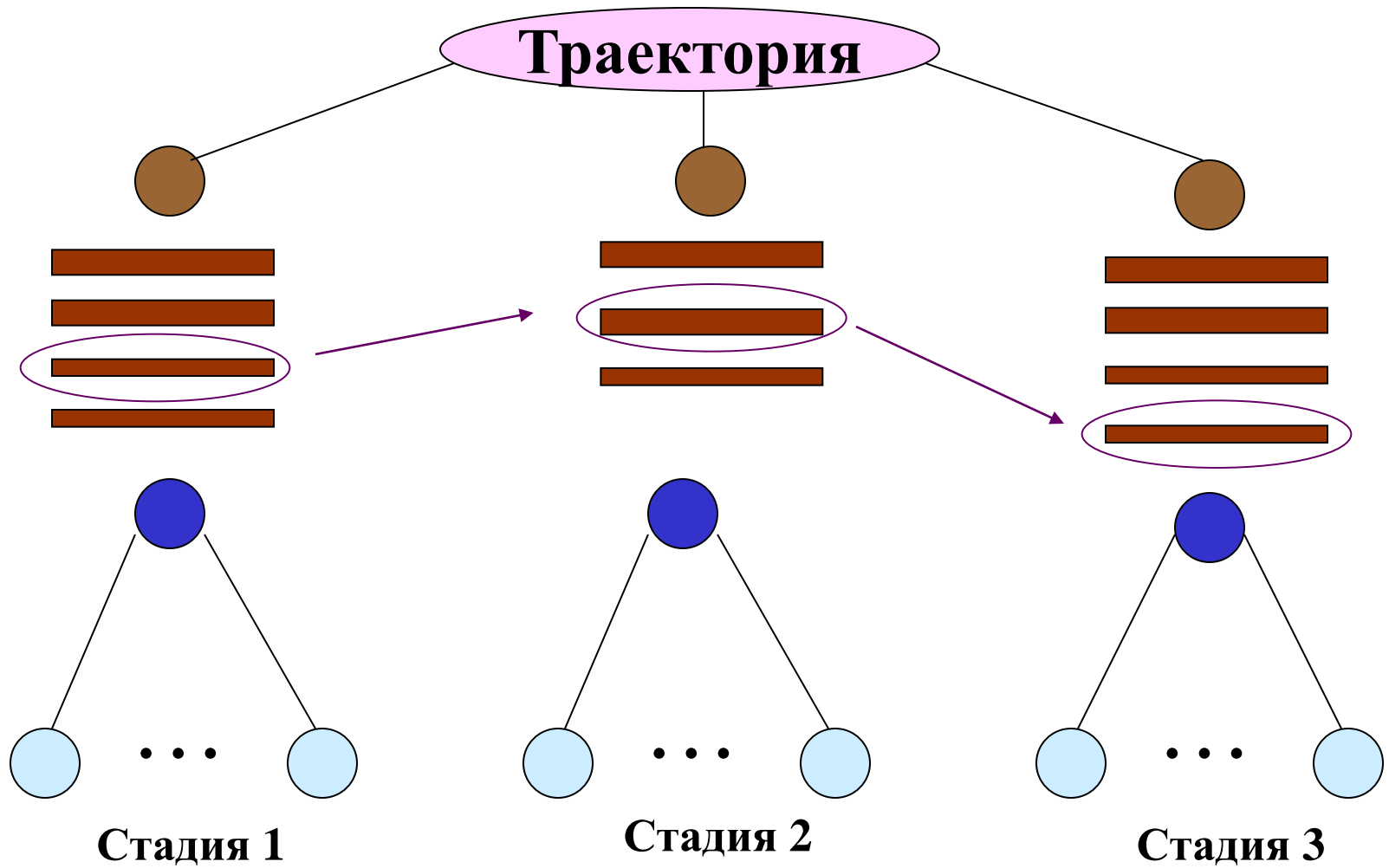
Выявление узких мест на основе $N(S)$



Здесь существует такое решение, что улучшение его элемента ведет к существенному улучшению решения

Пространство векторов качества:
 $N(S) = (w(S); n1(S) , n2(S) , n3(S))$

Прикладной пример 3: Многостадийное проектирование



0

T

ЛЕКЦИЯ 22. Курс: “Проектирование систем: Структурный подход”

Фак. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.22. Улучшение систем (примеры). Эволюция систем (пример)

ПЛАН:

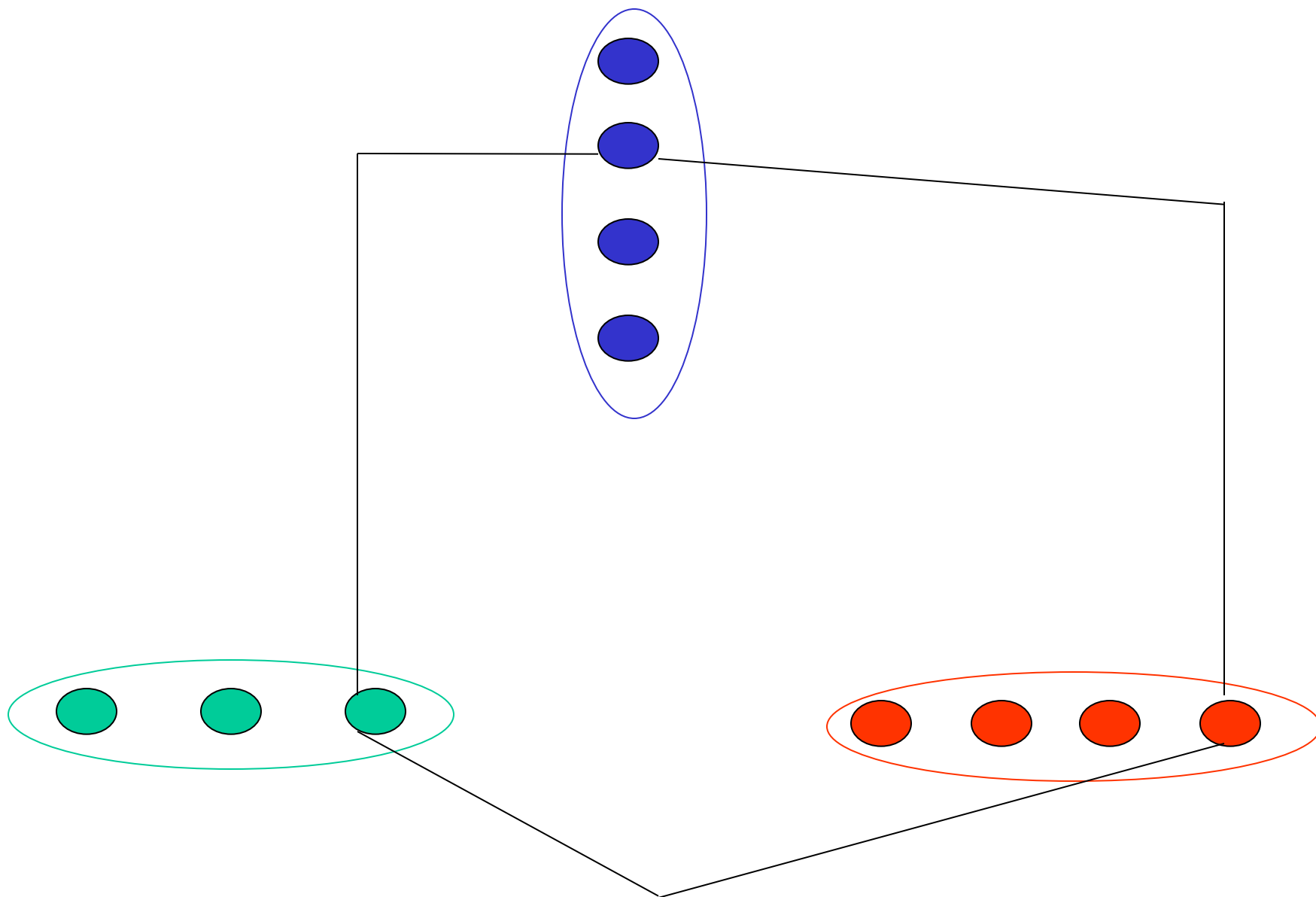
1 Иерархическое морфологическое проектирование (HMMD):

Применение к улучшению систем (улучшение, адаптация)

2. Развитие / эволюция систем:

Пример поколений системы DSS COMBI-PC

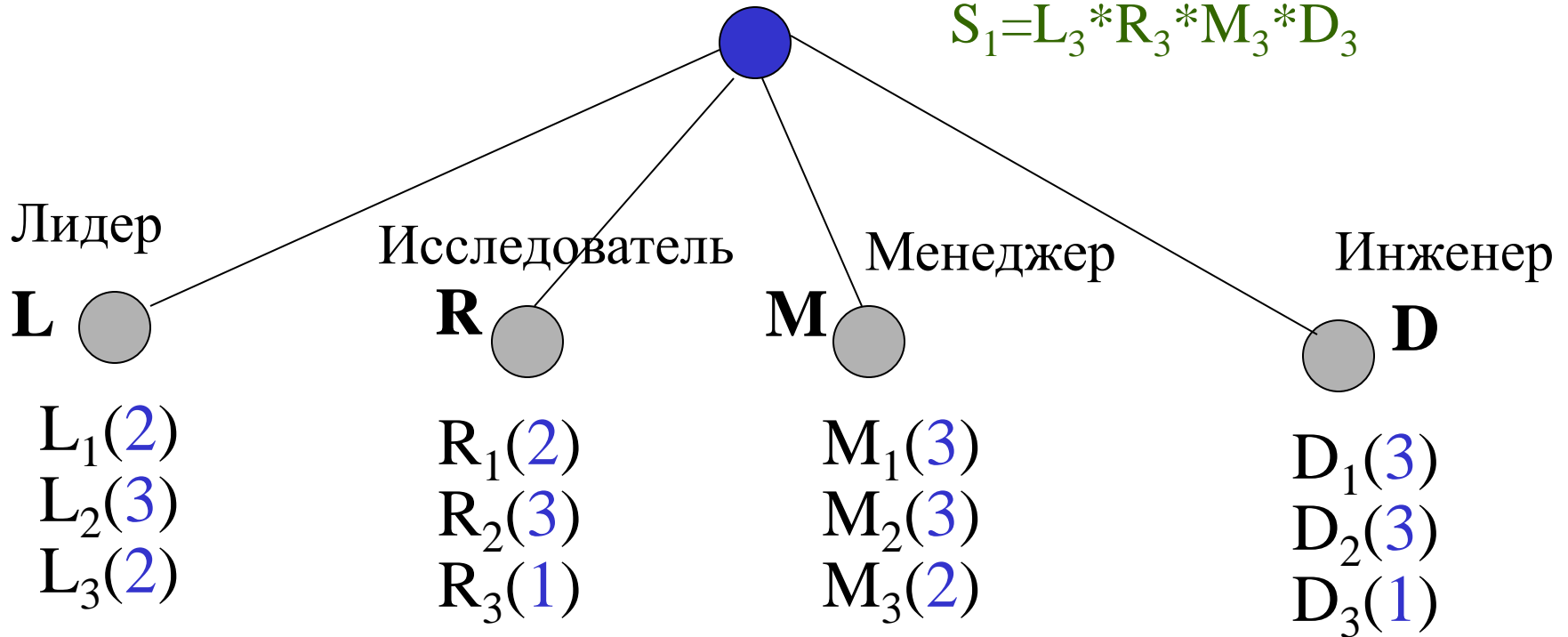
Морфологическая клика



Пример 1: Проектирование группы

Группа: $S = L * R * M * D$

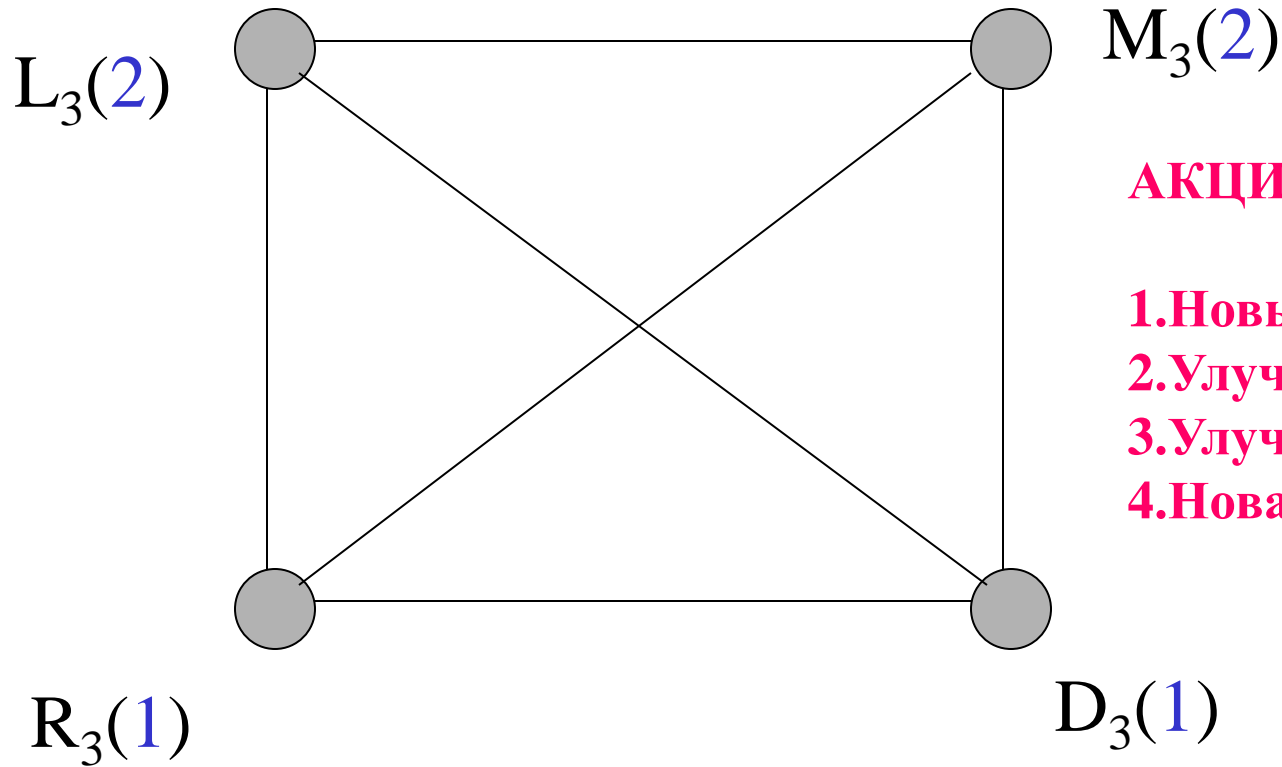
$$S_1 = L_3 * R_3 * M_3 * D_3$$



Пример 1: Проектирование группы

Группа: $S = L * R * M * D$

$$S_1 = L_3 * R_3 * M_3 * D_3$$



АКЦИИ УЛУЧШЕНИЯ:

- 1. Новый сотрудник**
- 2. Улучшение сотрудника**
- 3. Улучшение совместимости**
- 4. Новая структура**

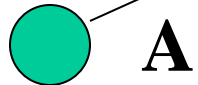
План улучшения

$$S = A * B * C$$

$$S_1 = A_3 * B_1 * C_2$$

$$S_2 = A_3 * B_1 * C_3$$

**Новые
сотрудники**



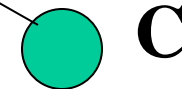
- $A_1(2)$
- $A_2(2)$
- $A_3 = A_1 \& A_2(2)$

**Профессиональные
курсы**



- $B_1(1)$
- $B_2(2)$
- $B_3(1)$
- $B_4(1)$
- $B_5 = B_3 \& B_4(2)$
- $B_6 = B_1 \& B_4(3)$
- $B_7 = B_1 \& B_2 \& B_4(3)$

**Совместные
поездки**



- $C_1(2)$
- $C_2(1)$
- $C_3(1)$
- $C_4 = C_1 \& C_3(3)$

A₁ новый лидер

A₂ новый менеджер

A₃ = A₁ & A₂

B₁ курс о достижениях в науке & технике

B₂ курс по иностранному языку

B₃ курс по системному анализу

B₄ курс по теории творчества

B₅ = B₃ & B₄

B₆ = B₁ & B₄

B₇ = B₁ & B₂ & B₄

C₁ курс по отношениям в коллективе

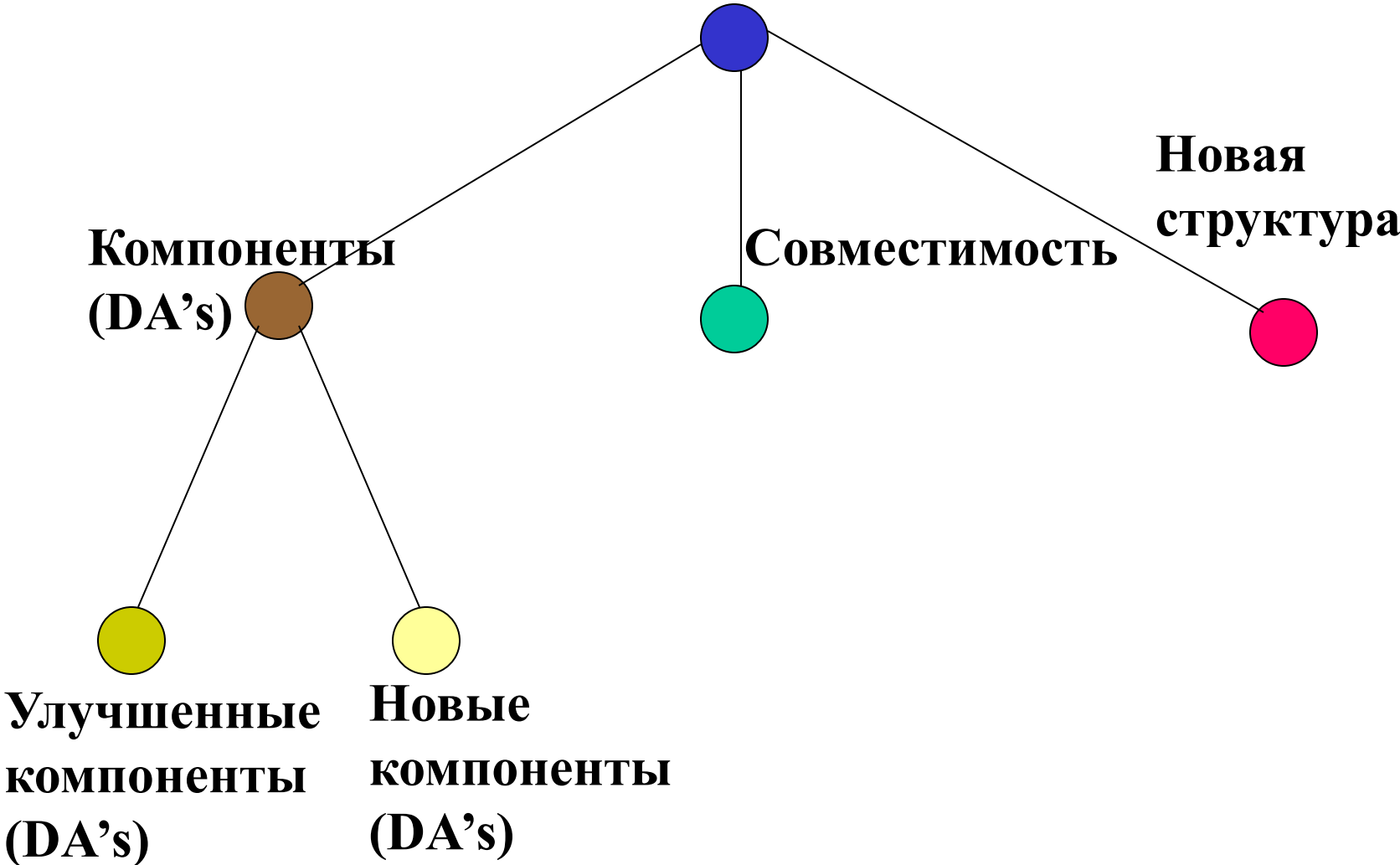
C₂ совместная поездка в дом отдыха

C₃ совместное участие в проф. конференции

C₄ = C₁ & C₂

Система улучшения

Система улучшения (изменения)



ПРОЦЕСС УЛУЧШЕНИЯ СИСТЕМЫ:

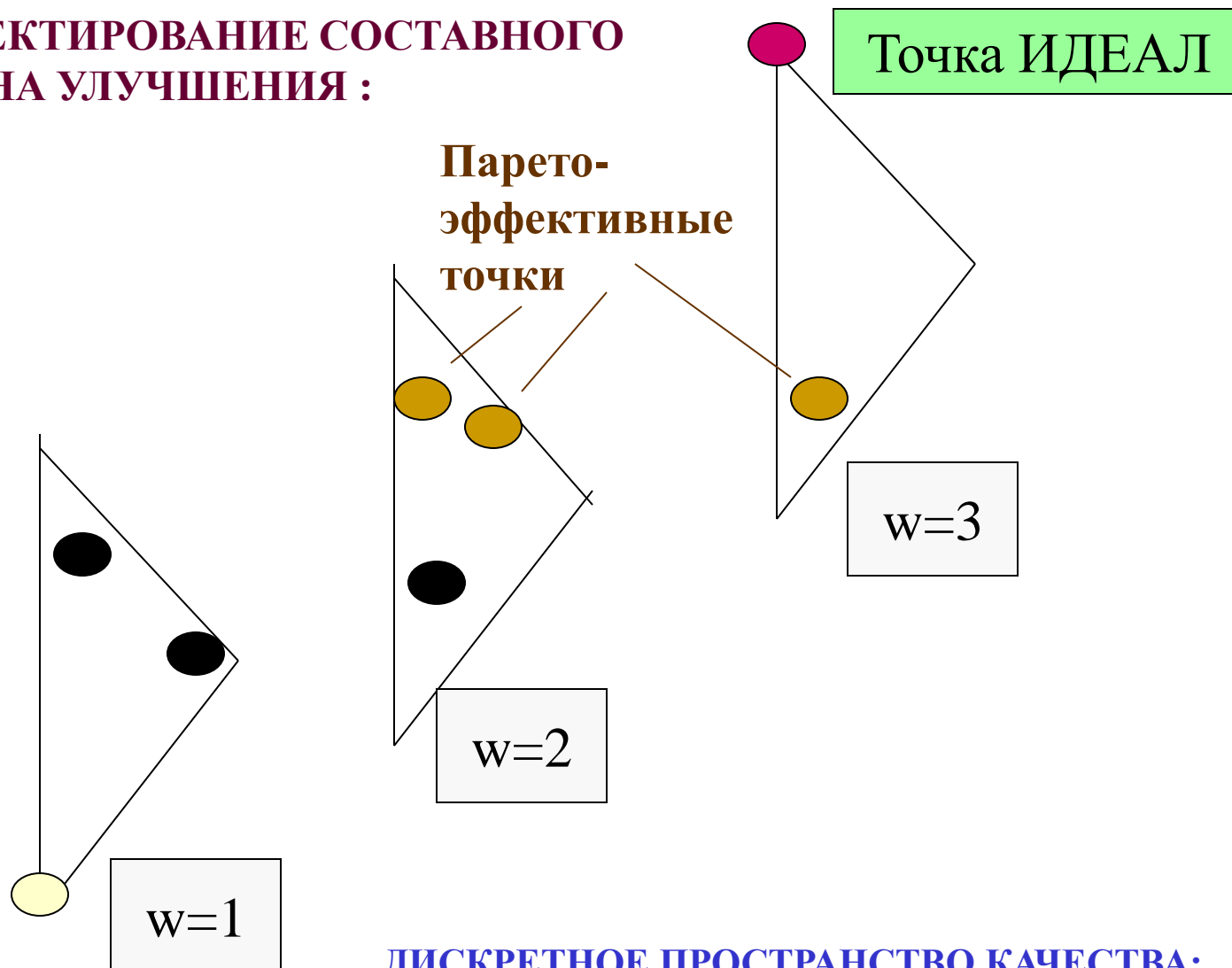
1.Улучшение/ Upgrade

**2.Системные изменения
(развитие / эволюция)**

3.Трансформирование системы

4.Адаптация (включая режим on-line)

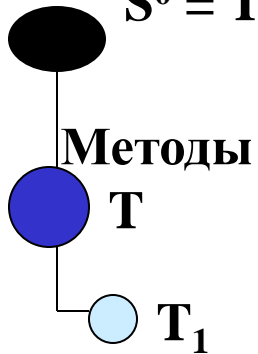
ПРОЕКТИРОВАНИЕ СОСТАВНОГО ПЛАНА УЛУЧШЕНИЯ :



ДИСКРЕТНОЕ ПРОСТРАНСТВО КАЧЕСТВА:
 $N(S) = (w(S); n1(S) , n2(S) , n3(S))$

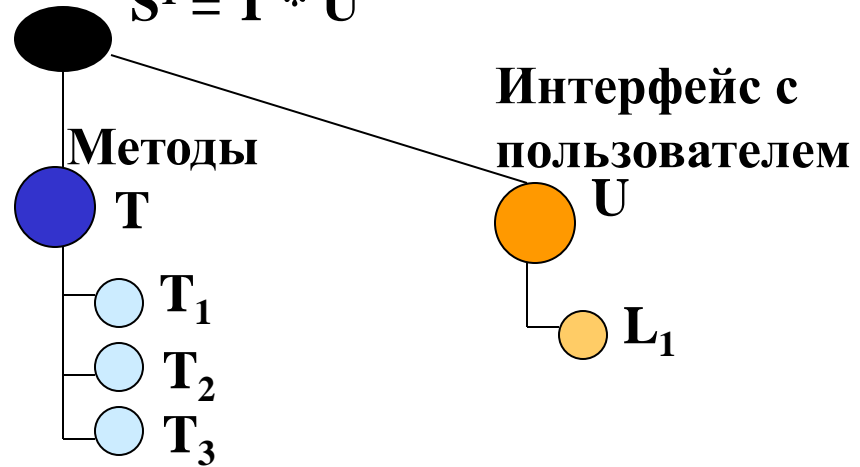
Система 0

$$S^0 = T$$



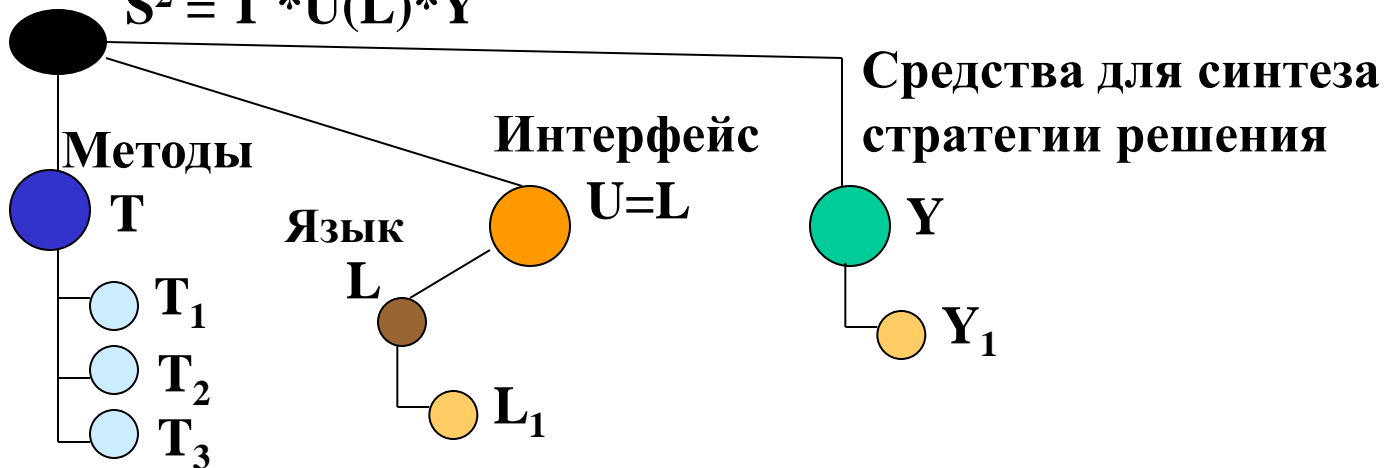
Система 1

$$S^1 = T * U$$



Система 2

$$S^2 = T * U(L) * Y$$

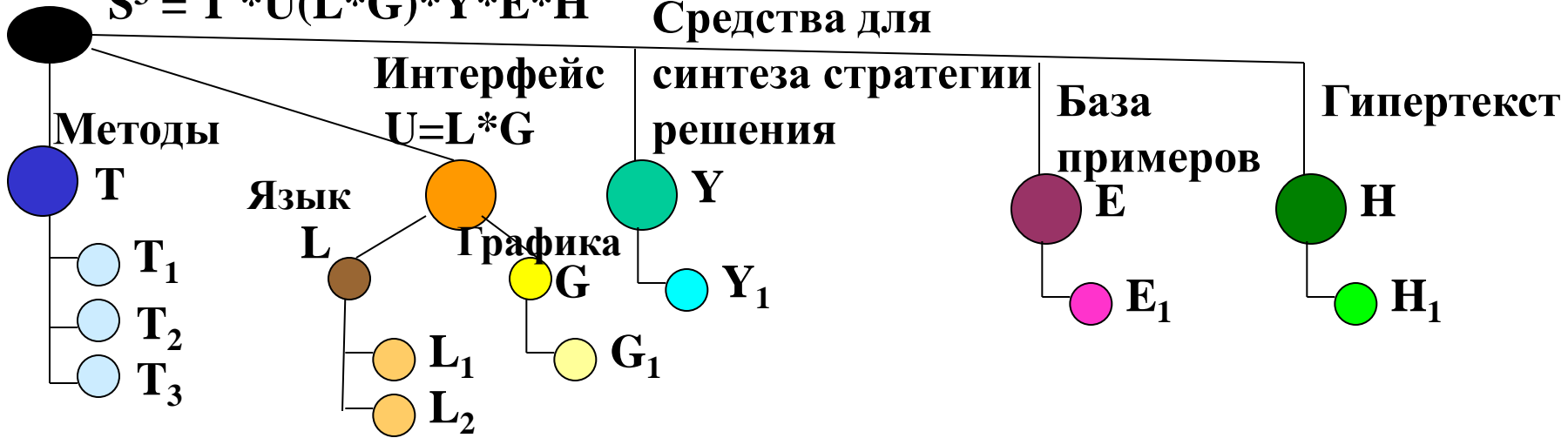


Система 3

$$S^3 = T * U(L * G) * Y * E * H$$

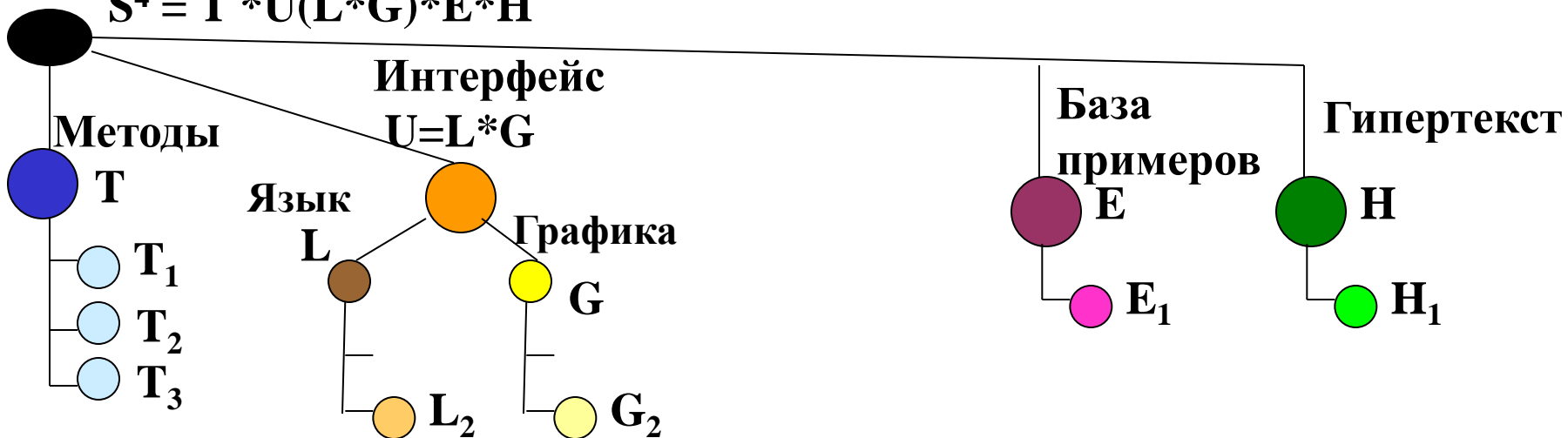
Средства для

синтеза стратегии
решения



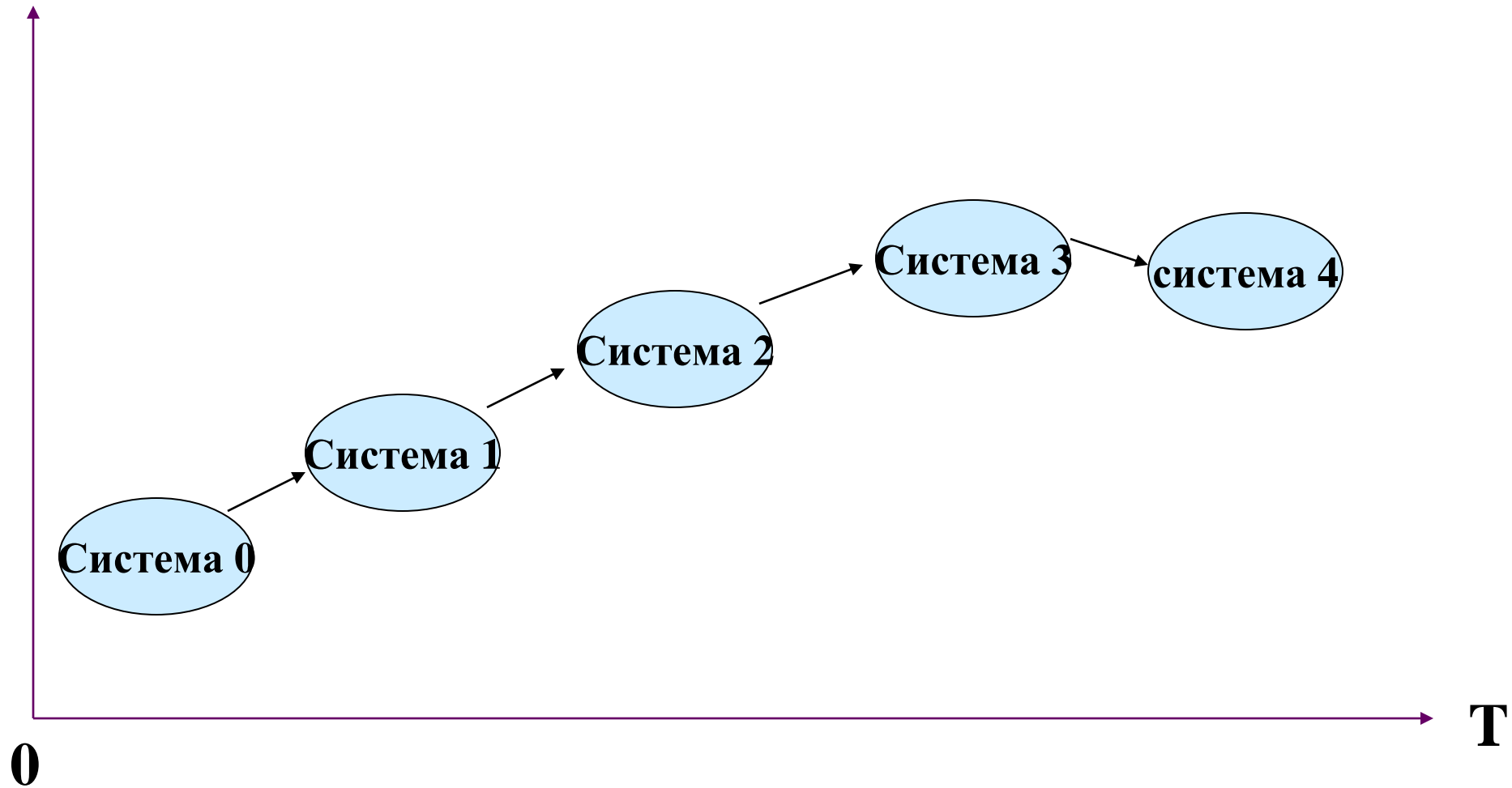
Система 4

$$S^4 = T * U(L * G) * E * H$$



Поколения системы DSS COMBI

Улучшение



ЛЕКЦИИ 23-24 (сокр. версия). Курс: “Проектирование систем: Структурный подход”
Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики
Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН
Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.23. Основные системные задачи. Пример для ноутбука.

Л.24. Система обработки сигналов. Процесс изменения системы.

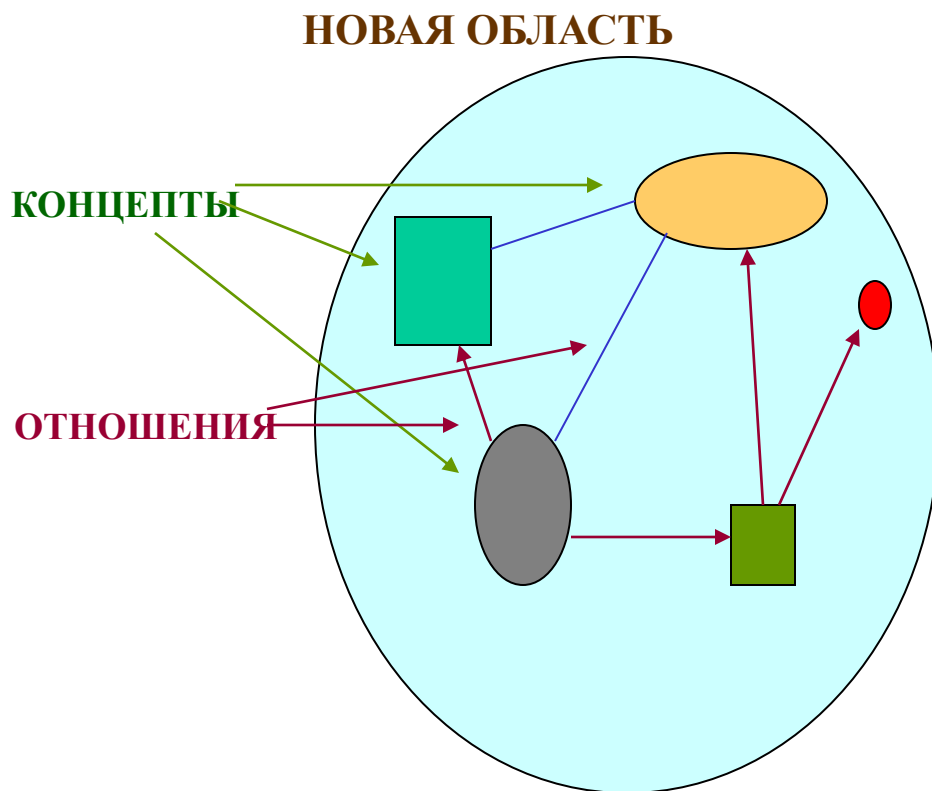
ПЛАН:

1. Анализ новой области
2. Иерархическое морфологическое проектирование: схема анализа и проектирования
(иллюстративный пример для ноутбука)
3. Уровни “систем”: *система, * требования, *стандарты
4. развитие / эволюция модульной системы (иллюстративные примеры): *ноутбук, *устройство для обработки сигнала
5. Типовые операции “системных изменений”
6. Базовые задачи комбинаторной оптимизации для улучшения / адаптации систем

Окт 23, 2004

ДВЕ СИТУАЦИИ:







1. Принципиально новая область
2. Новая область для исследователя



АЛГОРИТМИЧЕСКАЯ СХЕМА:

1. Выявление основных концептов (объекты, ресурсы, цели, участники)
2. Выявление основных отношений на указанных концептах
3. Постановка главных задач (т.е., назначение ресурсов, планирование/ составление расписаний)
4. разработка схем решения
5. Решение численных примеров
6. Исследование реальных приложений
7. Др.

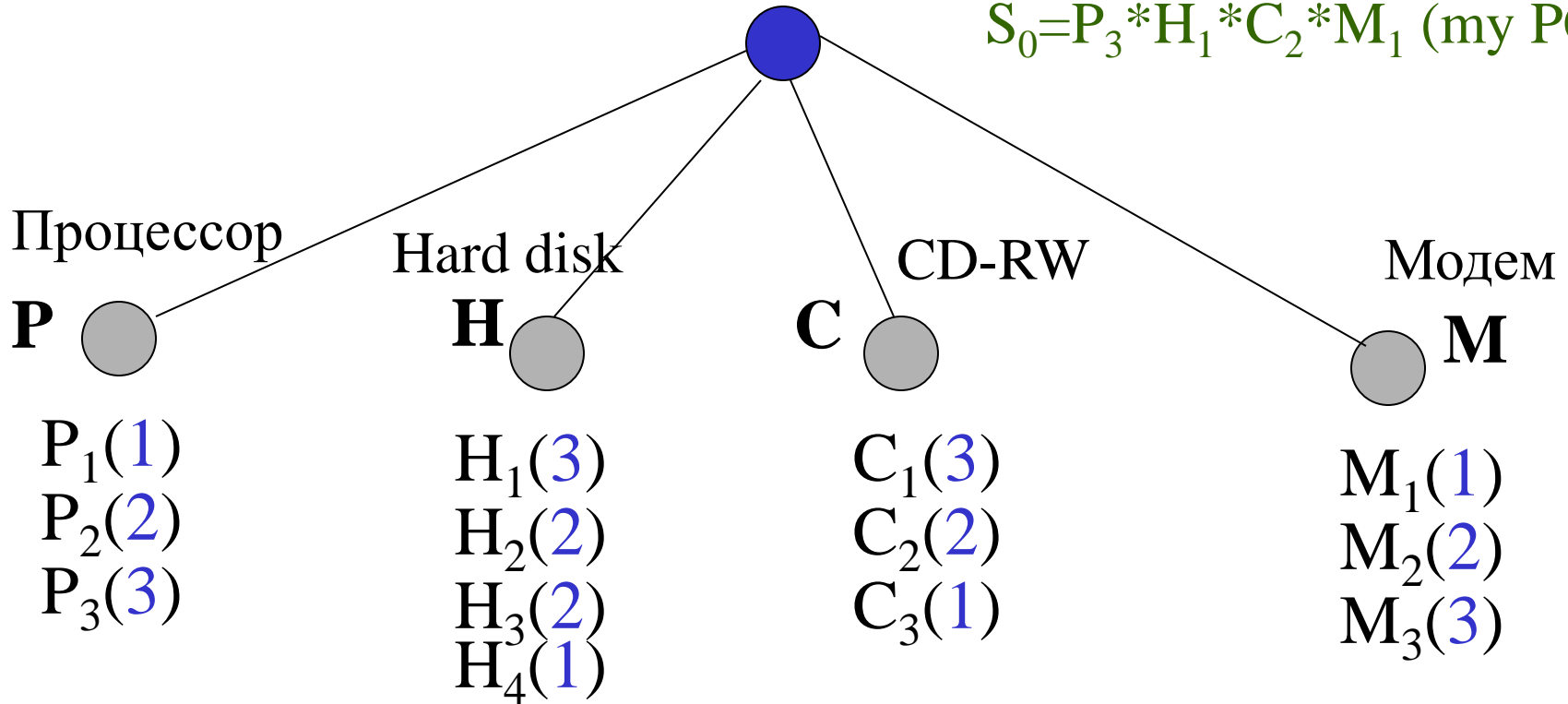
СИСТЕМНЫЙ АНАЛИЗ & ЗАДАЧА ПРОЕКТИРОВАНИЯ

1. Моделирование системы (структурная модель, 
например: И-ИЛИ граф)
2. Многокритериальное сравнение 
3. Выявление «узких мест» 
4. Иерархическое модульное проектирование 
5. Upgrade (улучшение, адаптация) 
6. Много-стадийное проектирование
7. Моделирование процесса развития / эволюции 
(поток поколений системы)
8. Прогнозирование систем

Пример: Ноутбук

Мой ноутбук: $S = P * H * C * M$

$S_0 = P_3 * H_1 * C_2 * M_1$ (my PC)



АЛЬТЕРНАТИВЫ:

P₁ Intel-4

P₂ Intel-3

P₃ Celeron

H₁ 20Gb

H₂ 40Gb

H₃ 60Gb

H₄ 80Gb

C₁ None

C₂ Read

C₃ Read & Write

M₁ 56 Kbit / sec

M₂ 48 Kbit / sec

M₃ 32 Kbit / sec

СРАВНЕНИЕ ноутбуков:

		Цена (-)	Надежность (+)	Техобсл. (+)	Удобство для upgrade(+)	Общий
1.Альт.	1	1300 (6)	5	3	5	2 (1)
2.Альт.	2	1250 (5)	4	3	4	3 (3)
3.Мой РС		900 (2)	4	4	5	1 (1)
4.Альт.	3	1200 (4)	5	3	4	3 (2)
5.Альт.	4	1200 (4)	5	3	3	3 (3)
6.Альт.	5	1100 (3)	4	4	4	2 (2)
7.Произ-тво в Москве		700 (1)	3	5	5	1 (1)
8.Альт.	6	1200 (4)	4	3	3	4 (4)

Веса критериев:

2 2 3 2

2 5 4 5

$$\text{СЕЙЧАС: } S_0 = P_3 * H_1 * C_2 * M_1$$

УЗКИЕ МЕСТА:

	Цена для upgrade (-)	Надежность (-)	Damage (+)	Общее
1.P ₃	100	5	2	3 (3)
2.H ₁	80	3	5	1 (1)
3.C ₂	200	4	1	4 (4)
4.M ₁	50	5	4	2 (2)

Веса критериев:

1 1 1
2 4 5

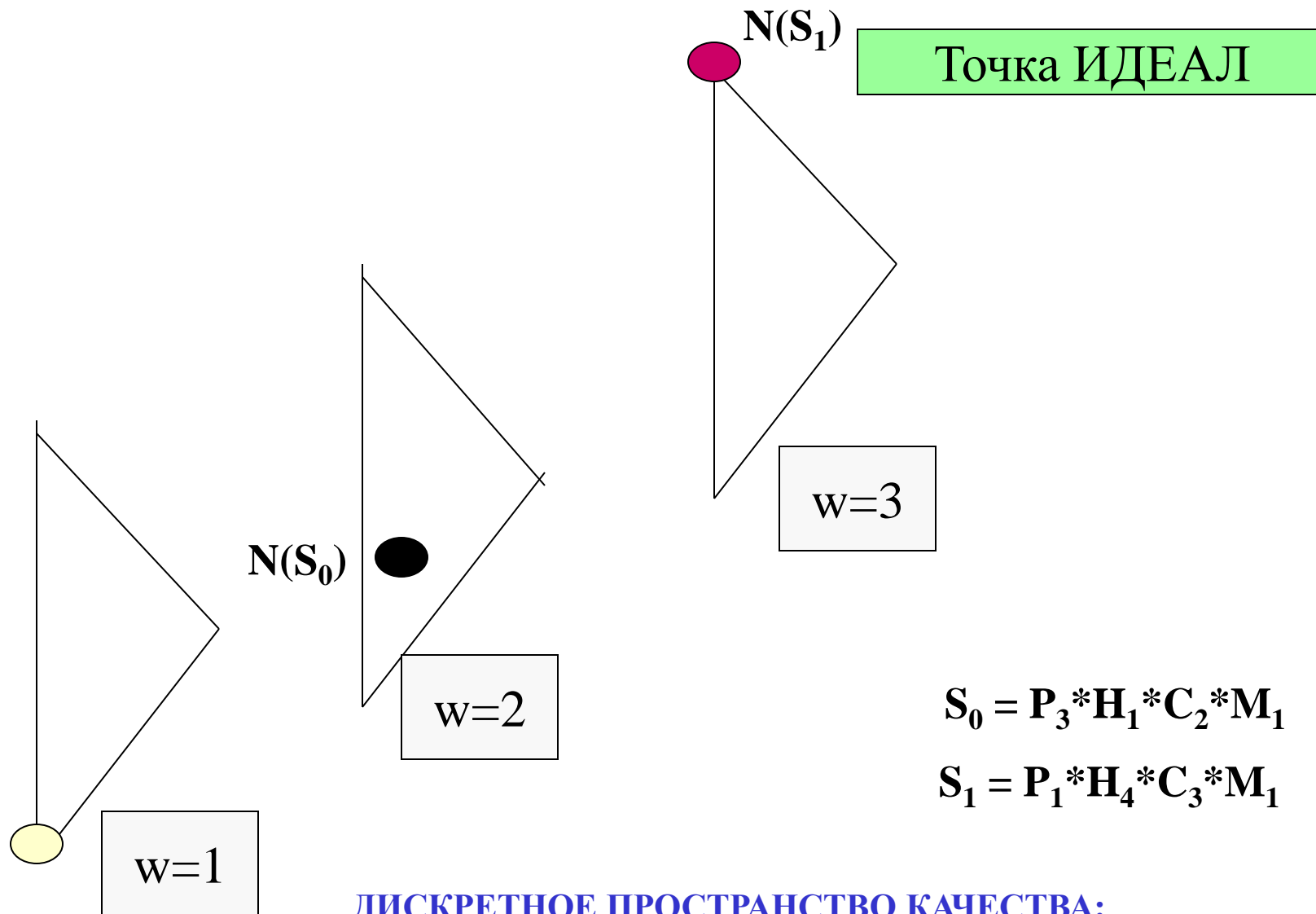
Оценивание совместимости между альтернативами (пример для ноутбука)

	H ₁	H ₂	H ₃	H ₄	C ₁	C ₂	C ₃	M ₁	M ₂	M ₃
P ₁	1	2	3	3	0	2	3	3	1	1
P ₂	3	2	1	1	1	3	3	2	3	1
P ₃	2	3	1	2	0	3	2	3	2	1
H ₁					1	3	1	3	3	2
H ₂					1	3	2	3	3	1
H ₃					1	2	3	3	2	1
H ₄					1	2	3	3	2	1
C ₁								3	2	2
C ₂								3	3	1
C ₃								3	3	1

ЗАМЕЧАНИЕ: 3 соответствует лучшему уровню совместимости
0 соответствует несовместимости

Лучшая комбинация : $S_1 = P_1 * H_4 * C_3 * M_1$ $N(S_1) = (3 ; 4,0,0)$

Upgrade (пример для ноутбука)



$$S_0 = P_3 * H_1 * C_2 * M_1$$

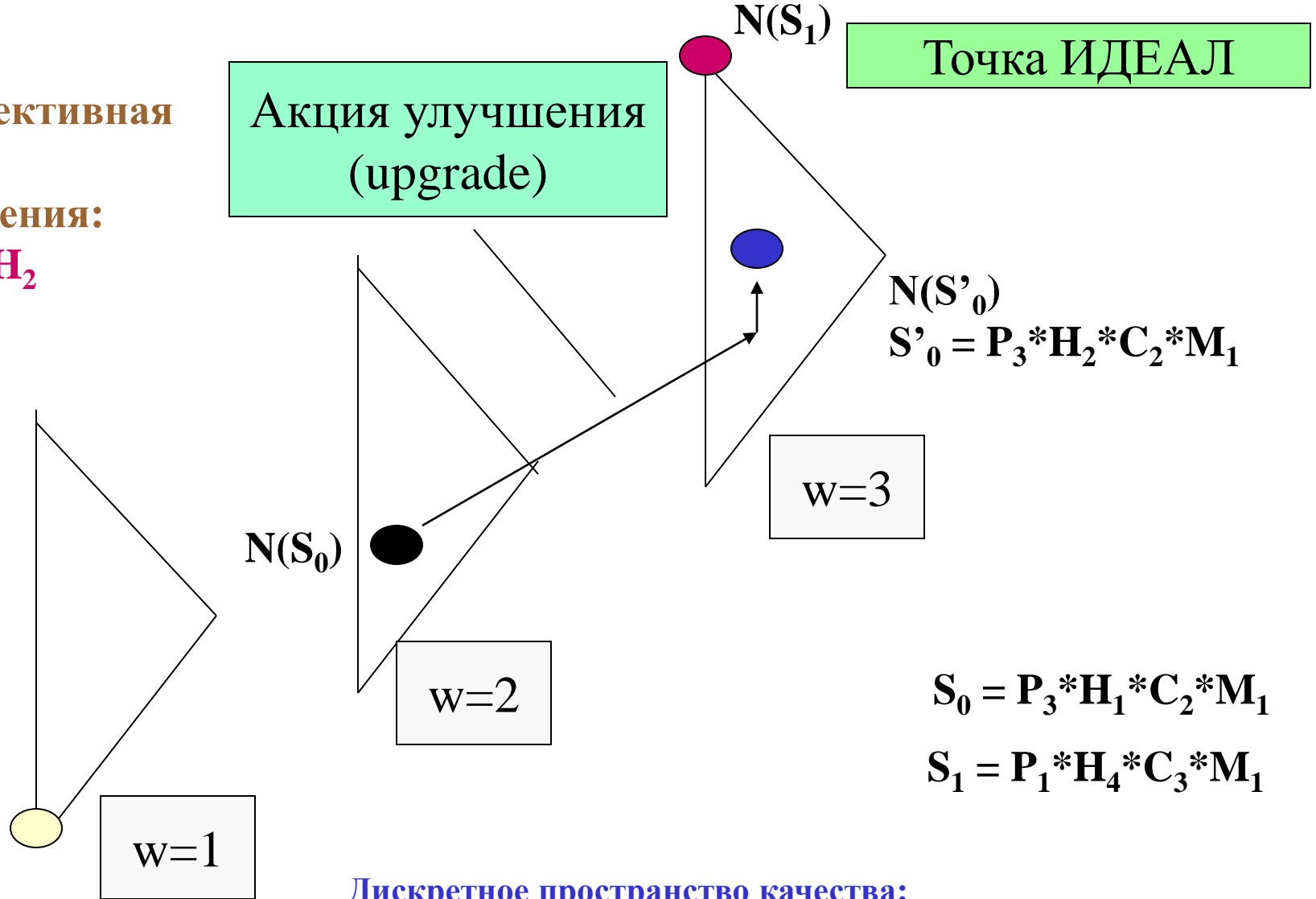
$$S_1 = P_1 * H_4 * C_3 * M_1$$

ДИСКРЕТНОЕ ПРОСТРАНСТВО КАЧЕСТВА:

$$N(S) = (w(S); n1(S) , n2(S) , n3(S))$$

Upgrade (пример для ноутбука)

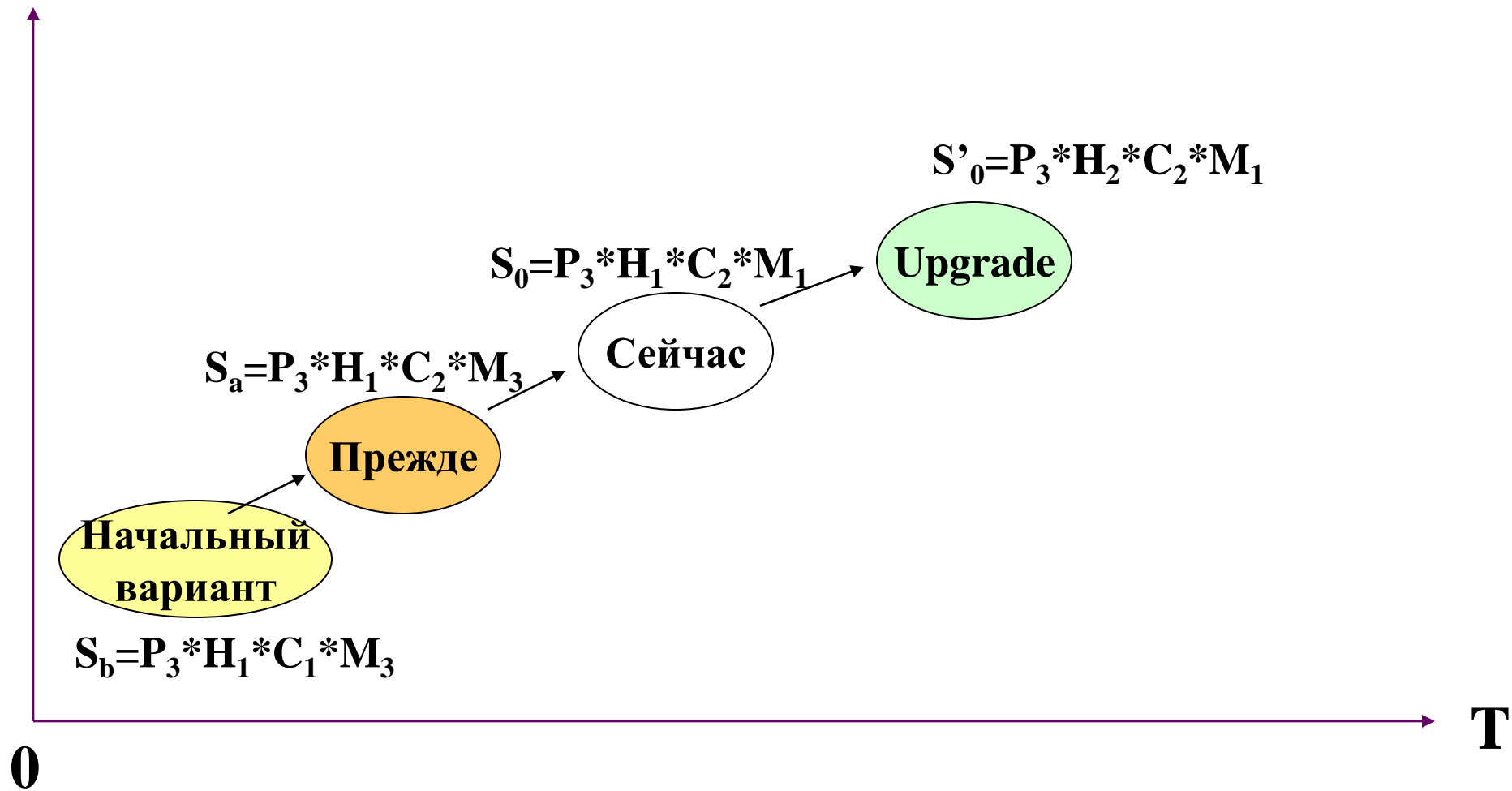
Перспективная
акция
улучшения:
 $H_1 \Rightarrow H_2$



Дискретное пространство качества:
 $N(S) = (w(S); n1(S) , n2(S) , n3(S))$

Пример для ноутбука

Улучшение



СТАНДАРТЫ

ТРЕБОВАНИЯ / КРИТЕРИИ

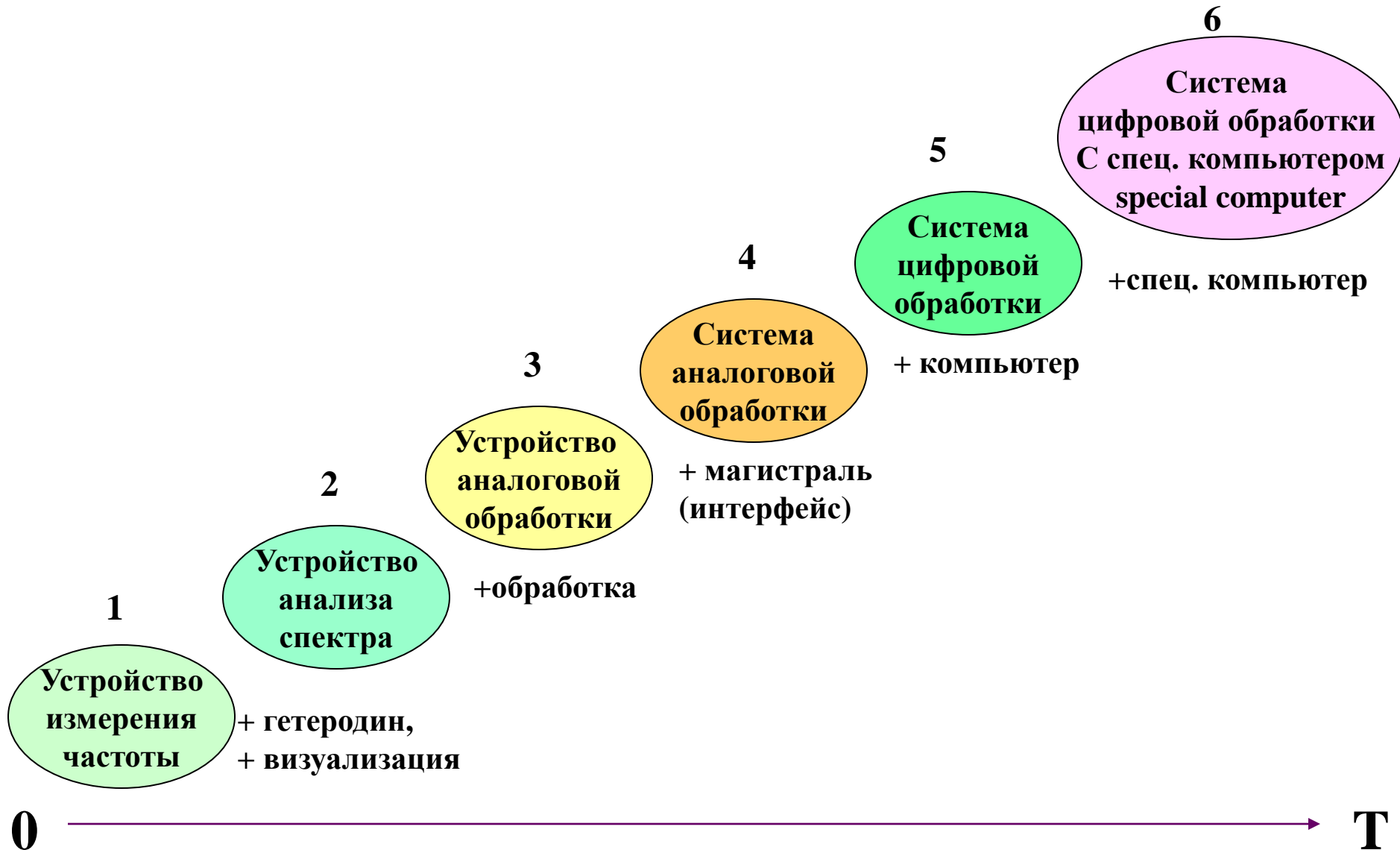
СИСТЕМА

0



T

Процесс макро-эволюции для устройств обработки сигнала (пример)



ОПЕРАЦИИ ИЗМЕНЕНИЯ:

I. Операции для DA's:

1.1. Изменение / улучшение DA's $O_1: A_i \Rightarrow A'_i$

1.2. Исключение DA O_2

1.3. Добавление DA O_3

1.4. Агрегирование DA's $O_4: \{ A_i \} \Rightarrow A^a = A_1 \& A_2 \& \dots$

1.5. Стандартизация DA's $O_5: \{ A_i \} \Rightarrow A^s$

II. Операции для подсистем:

2.1. Изменение / улучшение подсистемы O_6

2.2. Удаление подсистемы O_7

2.3. Добавление подсистемы O_8

2.4. Агрегирование системных частей O_9

I. Характеристики операция изменения:

- 1. Требуемые ресурсы**
- 2. Возможная полезность (прибыль)**
- 3. Др.**

II. Бинарные отношения на операциях изменения:

- 1. Ограничения предшествования ($O_i \Rightarrow O_j$)**
- 2. Эквивалентность**
- 3. Дополнительность**

ВОЗМОЖНЫЕ КОМБИНАТОРНЫЕ ПРОБЛЕМЫ:

- 1. Многокритериальное ранжирование**
- 2. Задача о рюкзаке**
- 3. Блочная задача о рюкзаке**
- 4. Многокритериальная задача о рюкзаке**
- 5. Многокритериальная блочная задача о рюкзаке**
- 6. Задачи составления расписаний**
- 7. Комбинаторный синтез (модульное проектирование)**
- 8. Многостадийное проектирование**

ЛЕКЦИЯ 25. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационных сетей и систем”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.25. Проектирование жизненного цикла. Система с общими модулями

ПЛАН:

1. Проектирование жизненного цикла: иллюстративный пример (морфологический комбинаторный подход)

2. Морфологический комбинаторный подход к много-продуктовым системам: общие модули:

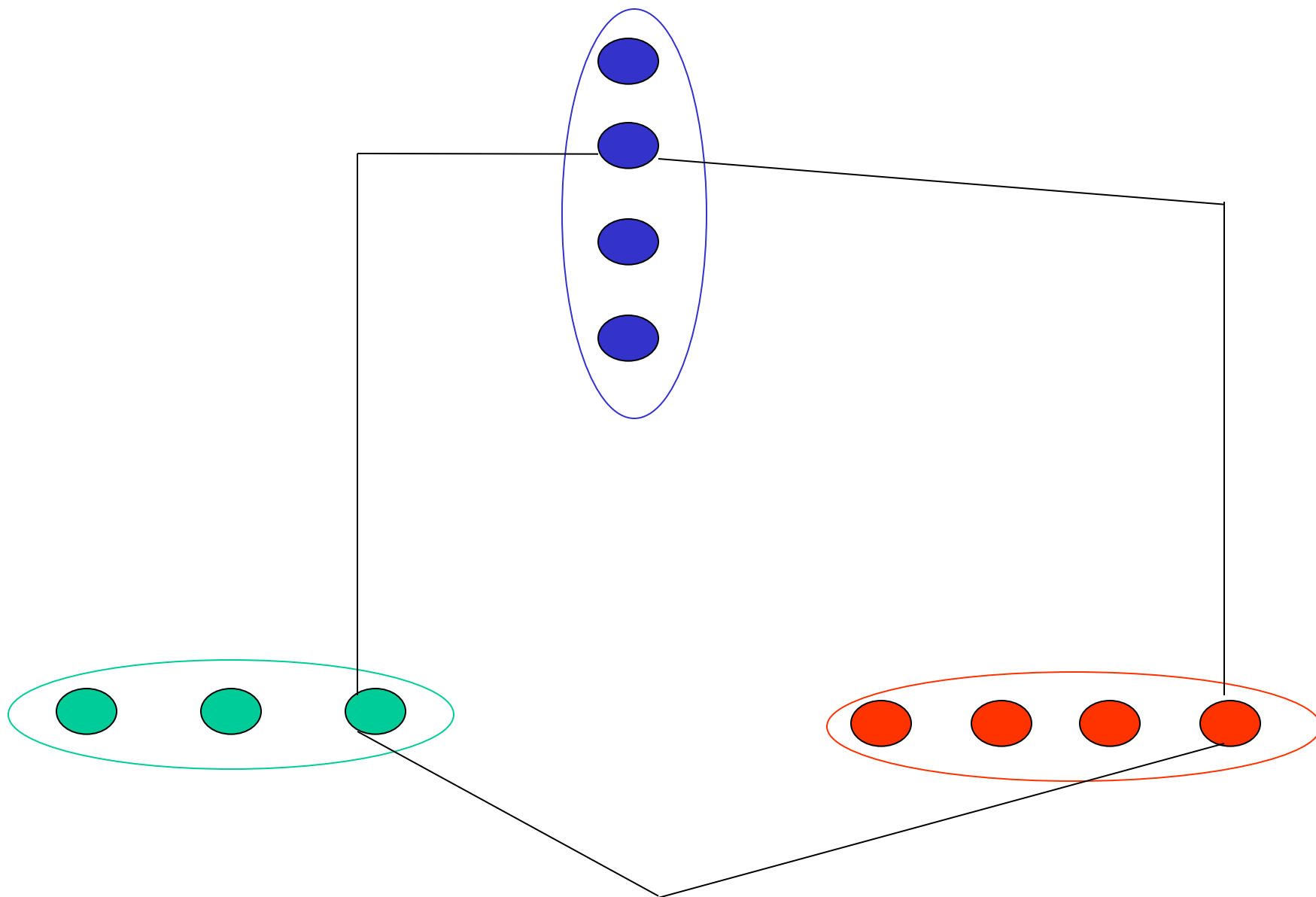
*система из 2-х продуктов (1 общий модуль, k общих модулей)

*система из m продуктов (1 общий модуль, k общих модулей)

3. Библиография

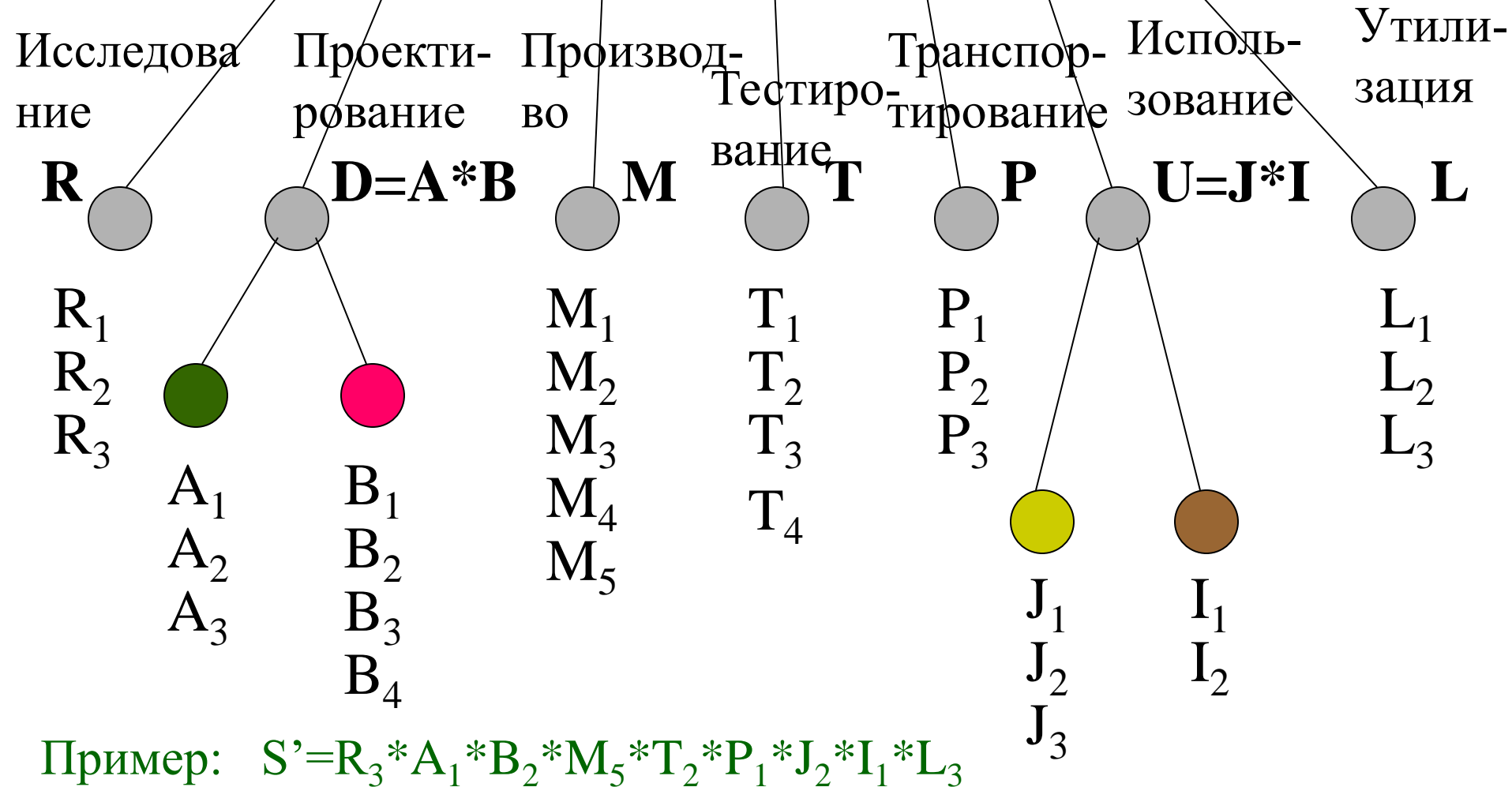
Oct. 29, 2004

Морфологическая клика



Проектирование (планирование) жизненного цикла

Жизненный цикл S



Жизненный цикл S

Пример 1: Проектирование & Производство $S_1 = D^1(A_2 * B_1) * M_3$

Пример 2: Проектирование & Тестирование $S_2 = D^2(A_1 * B_3) * T_3$

Пример 3: Исследование & Проектирование & Производство
 $S_3 = R_2 * D'(A_3 * B_1) * M_4$

Пример 4: Проектирование & Производство & Транспортирование
 $S_1 = D''(A_3 * B_3) * M_4 * T_3$

Жизненный цикл S

**Проектирование (Планирование)
жизненного цикла**

Управление жизненным циклом

Технология жизненного цикла

Поддержка жизненного цикла

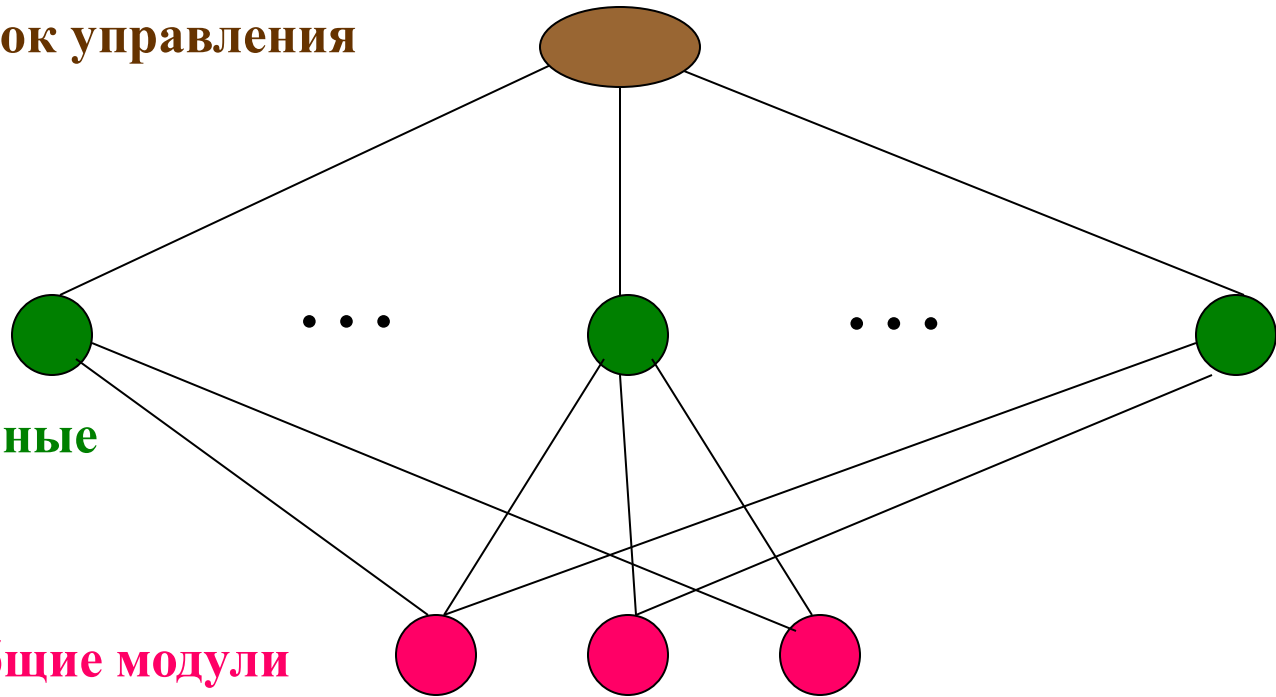
Техобслуживание жизненного цикла

Структура модульного программного пакета: 3 уровня

Уровень 1: Блок управления

**Уровень 2:
Функциональные
блоки**

Уровень 3: Общие модули



Общая структура модульного программного пакета

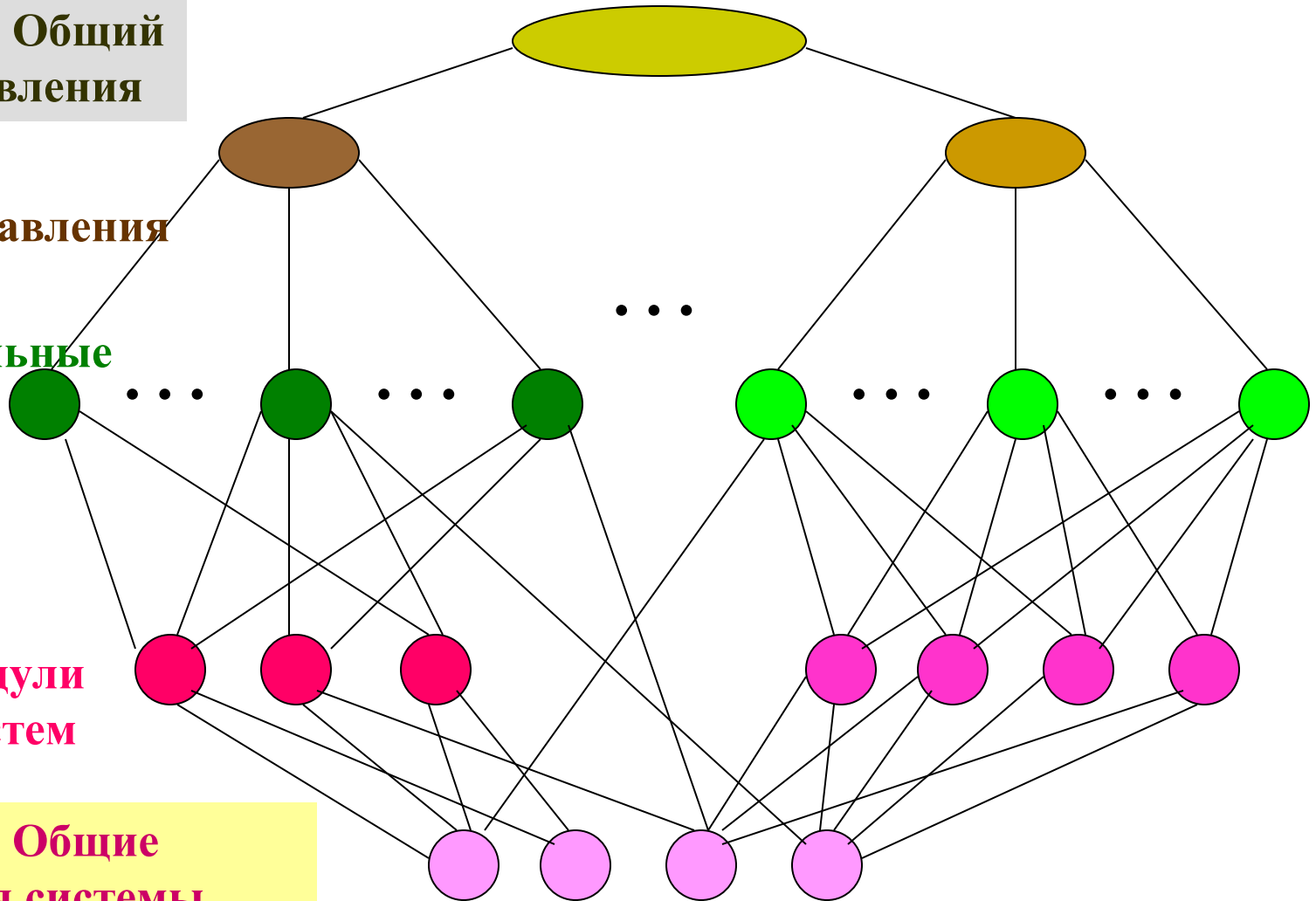
**Уровень 0: Общий
Блок управления**

**Уровень 1:
Блоки управления**

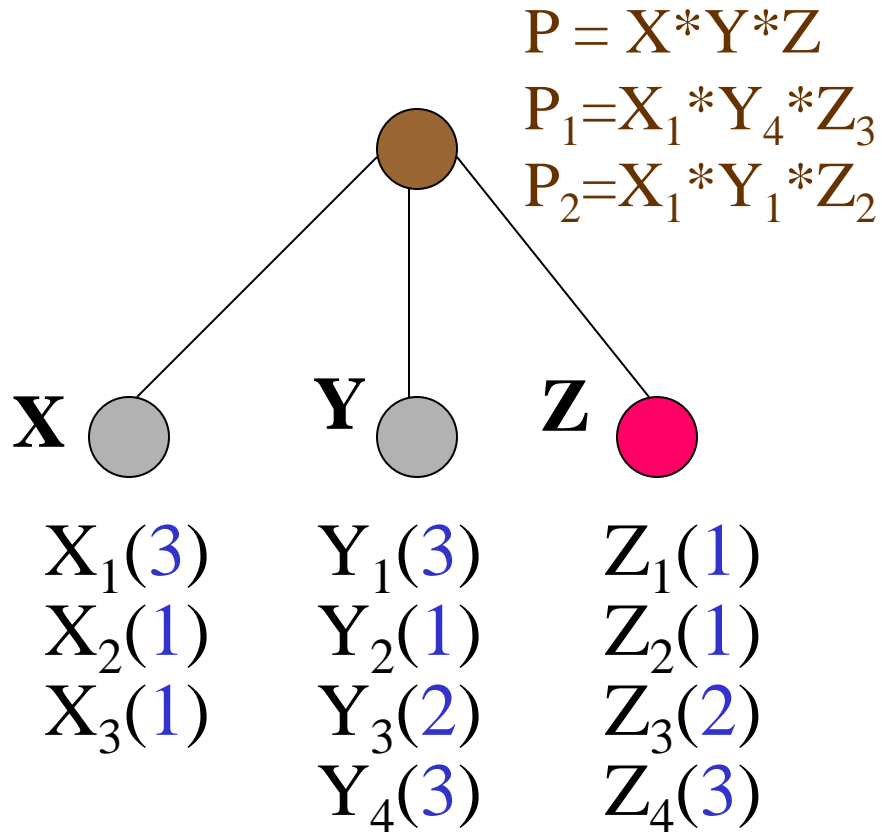
**Уровень 2:
Функциональные
блоки**

**Уровень 3:
Общие модули
для подсистем**

**Уровень 4: Общие
модули для системы**

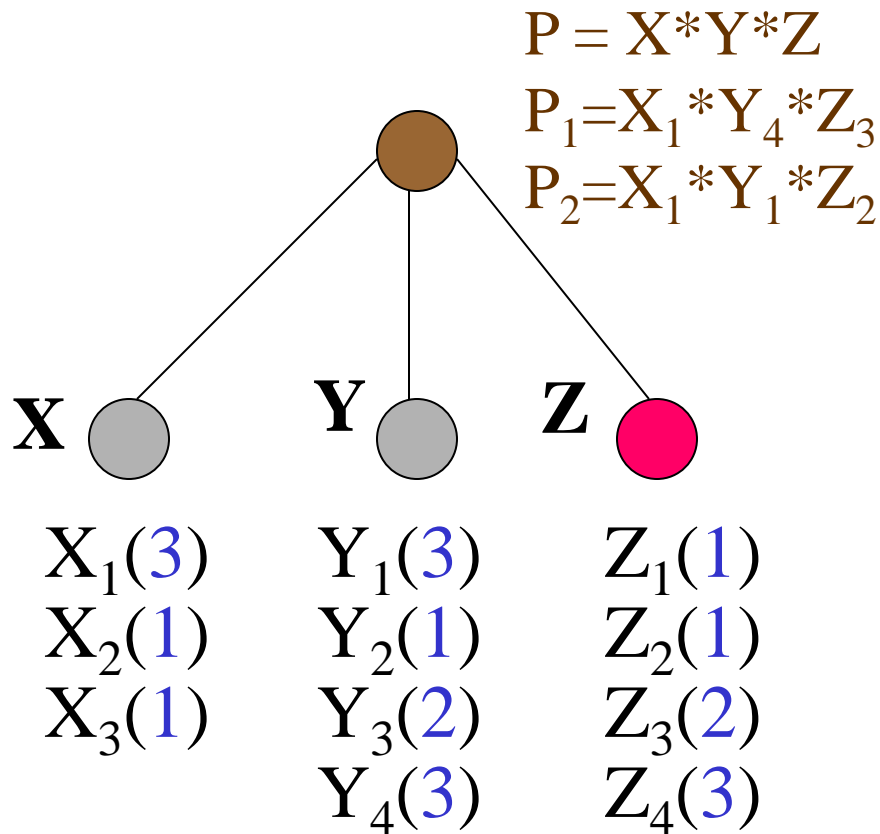


Продукт 1

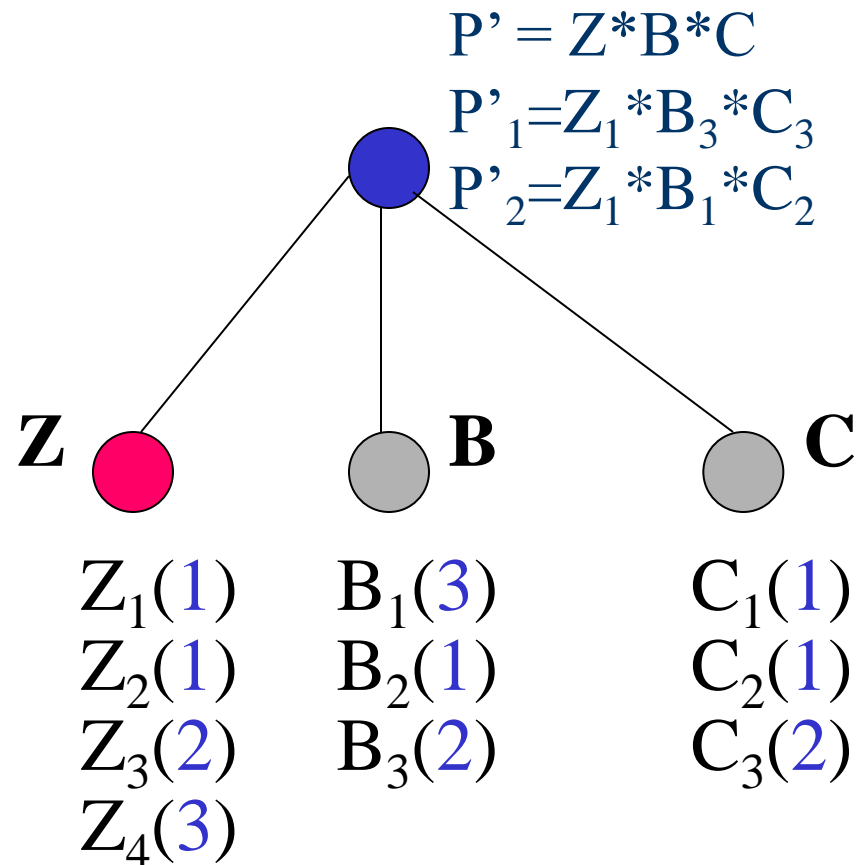


Система из 2-х продуктов

Продукт 1



Продукт 2



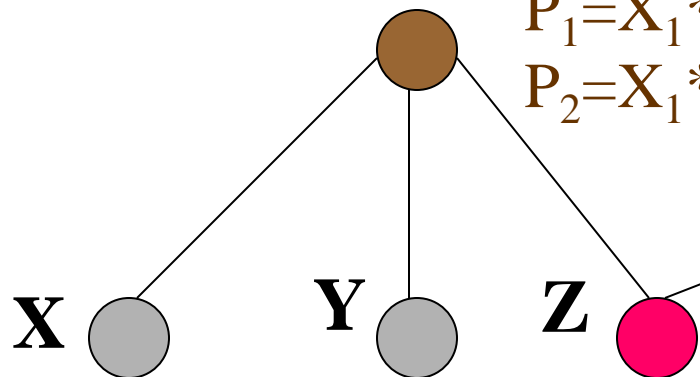
Система из 2-х продуктов (1 общий модуль)

Продукт 1

$$P = X * Y * Z$$

$$P_1 = X_1 * Y_4 * Z_3$$

$$P_2 = X_1 * Y_1 * Z_2$$



$X_1(3)$
 $X_2(1)$
 $X_3(1)$

$Y_1(3)$
 $Y_2(1)$
 $Y_3(2)$
 $Y_4(3)$

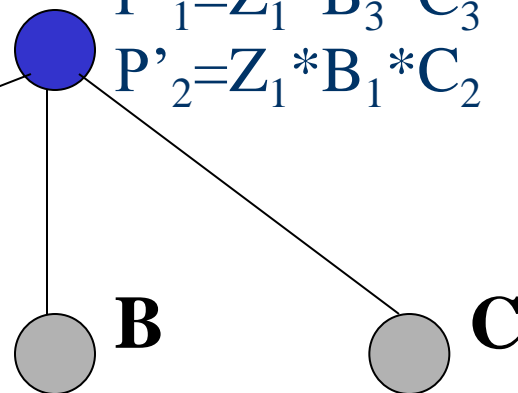
$Z_1(1)$
 $Z_2(1)$
 $Z_3(2)$
 $Z_4(3)$

Продукт 2

$$P' = Z * B * C$$

$$P'_1 = Z_1 * B_3 * C_3$$

$$P'_2 = Z_1 * B_1 * C_2$$



$B_1(3)$
 $B_2(1)$
 $B_3(2)$

$C_1(1)$
 $C_2(1)$
 $C_3(2)$

Система из 2-х продуктов (1 общий модуль)

$$S = P * P'$$

Продукт 1

$$P = X * Y * Z$$

$$P_1 = X_1 * Y_4 * Z_3$$

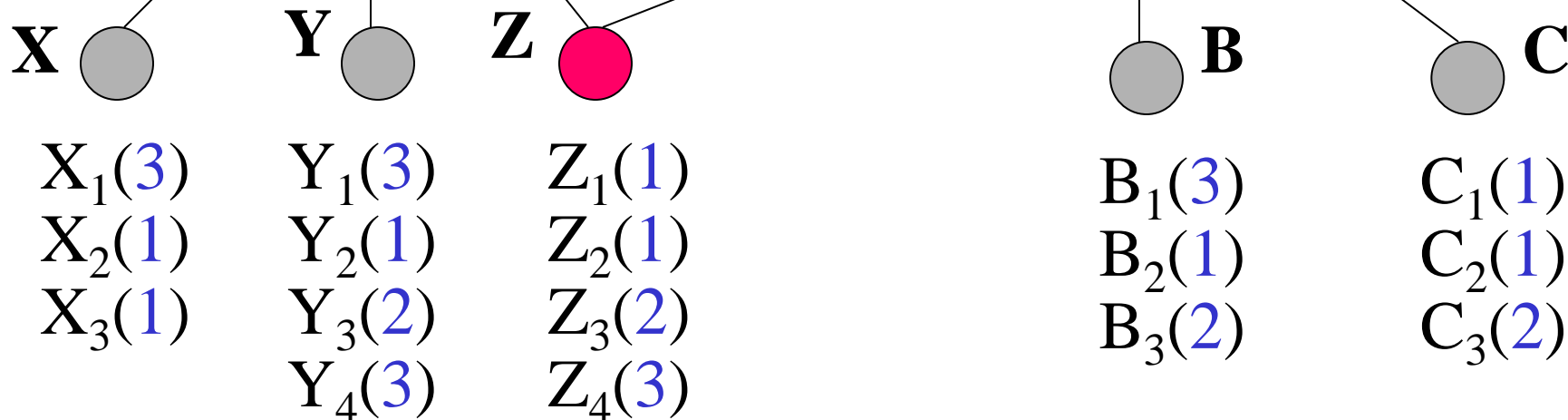
$$P_2 = X_1 * Y_1 * Z_2$$

Продукт 2

$$P' = Z * B * C$$

$$P'_1 = Z_1 * B_3 * C_3$$

$$P'_2 = Z_1 * B_1 * C_2$$



Система из 2-х продуктов (2 общих модулей)

$$S = P * P'$$

Продукт 1

$$P = X * Y * Z$$

$$P_1 = X_1 * Y_4 * Z_3$$

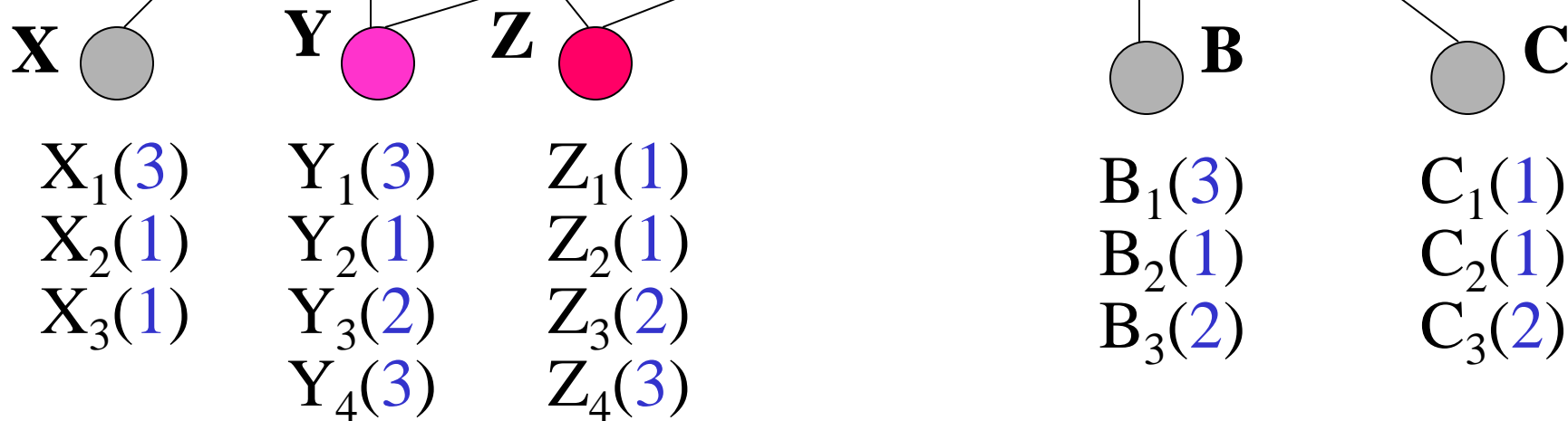
$$P_2 = X_1 * Y_1 * Z_2$$

Продукт 2

$$P' = Y * Z * B * C$$

$$P'_1 = Y_1 * Z_1 * B_3 * C_3$$

$$P'_2 = Y_2 * Z_1 * B_1 * C_2$$



- 1. B. Agard, A. Kusiak, Data-mining-based methodology for the design of product family. Int. J. of Prod. Res., 42(15), 2955-2969, 2004.**
- 2. C.Y. Baldwin, K.B. Clark, Design Rules: The Power of Modularity. MIT Press, 2000.**
- 3. J. Dahmus, J.P. Gonzalez-Zugasti, K.N. Otto, Modular product architecture, Design Studies 22(5), 409-424, 2001.**
- 4. G. Dobrescu, Y. Reich, Progressive sharing of modules among product variants. Computer-Aided Design 35(9), 791-806, 2003.**
- 5. X. Du, J. Jiao, M.M. Tseng, Architecture of product family: Fundamentals and methodology. Concurrent Eng.: Res. and Appl. 9(4), 309-325, 2001.**
- 6. J.K. Gershenson, G.J. Prasad, S. Allamneni, Modular product design: A life-cycle view. Trans. of the SDPS 3(4), 13-26, 1999.**
- 7. J.P. Gonzalez-Zugasti, K.N. Otto, J.D. Baker, A method for architecting product platform. Res. in Eng. Des. 12(2), 61-72, 2000.**
- 8. T.K.P. Holmqvist, M.L. Person, Analysis and improvement of product modularization methods: Their ability to deal with complex products. Systems Engineering 6(3), 195-209, 2003.**
- 9. C.C. Huang, A. Kusiak, Modularity in design of products and systems. IEEE Trans. on Syst., Man and Cybern. - Part A, 28(1), 66-77, 1998.**
- 10. M.Sh. Levin, Modular system synthesis: Example for packaged composite software, IEEE Tr. on SMC-Part C 35(4), 544-553 2005.**
- 11. M.Sh. Levin, Combinatorial design of multiproduct system: common modules. Elsevier Server of Preprints in CS, 2003.**

Recent English References

12. **M. Kokkolaras, R. Fellini, H.M. Kim, N. Michelena, and P. Papalambros, Extension of the target cascading formulation to the design of product family, Structural and Multidisciplinary Optimization, 24(4), 293-301, 2002.**
13. **A. Kusiak, Integrated product and process Design: a modularity perspective. J. of Eng. Des., 13(3), 223-231, 2002.**
14. **A. Messac, M.P. Martinez, T.W. Simpson, Effective product family design using physical programming. Engineering Optimization 34(3), 245-261, 2002.**
15. **M.H. Meyer, A.P. Lehnerd, The Power of Product Platforms, The Free Press, New York, 1997.**
16. **J.H. Mikkola, O. Gassmann, Managing modularity of product architectures: Toward an integrated theory. IEEE Trans. on Eng. Manag. 18(3), 204-218, 2003.**
17. **D. Robertson, K. Ulrich, Planning for product platforms, Sloan Manag. Review 39(4), 19-34, 1998.**
18. **M.S. Sawhney, Leverage high-variety strategies: From portfolio thinking to platform thinking. J. of the Academy of Marketing Science 26(1), 54-61, 1998.**
19. **D.M. Sharman, A.A. Yassine, Charactrizing complex product architecture. Systems Engineering 7(1), 35-60, 2004.**
20. **Z. Siddique, D.W. Rosen, On combinatorial design spaces for the configuration design of product family. AI EDAM 15(2) 91-108, 2001.**
21. **T.W. Simpson, J.R.A. Maier, F. Mistree, Product platform design: methods and application. Res. in Eng. Des. 13(1), 2-22,2001.**

ЛЕКЦИЯ 26. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марок Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.26. Тестирование систем

ПЛАН:

1. Тестирование систем (основные подходы): *white-box тестирование (системы с известной структурой),

*black-box тестирование как проверка модели (model checking), *black-box тестирование как много-функциональное тестирование

2. Много-функциональное тестирование: базовые комбинаторные задачи:

*предварительный анализ системы, *композиция тестовых входов (test cases),

*проектирование цепочки тестовых входов (chain of test cases), *покрытие орграфа кластеров функций цепочками

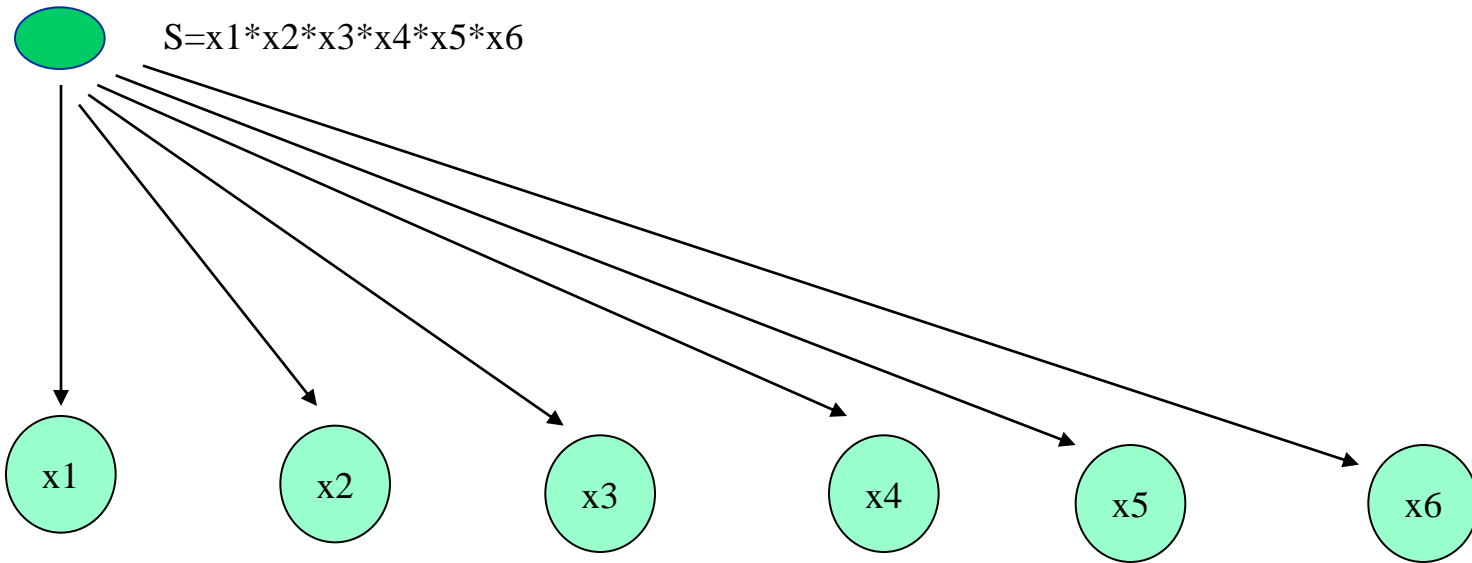
3. Иллюстративный пример много-функционального тестирования

Ноябрь 5, 2004

Система как черный ящик (Black-box system)



Пример структуры тестового входа (test case)



- | | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 0: x1='Any' (2) | 0: x2='Any' (2) | 0: x3='Any' (4) | 0: x4='Any' (4) | 0: x5='Any' (2) | 0: x6='Any' (4) |
| 1: x1=1(3) | ● 1: x2=1(3) | 1: x3=1(1) | ● 1: x4=1(1) | ● 1: x5=1(3) | 1: x6=1(2) |
| ● 2: x1=2(3) | 2: x2=2(3) | 2: x3=2(1) | 2: x4=2(1) | 2: x5=2(3) | 2: x6=2(2) |
| | | ● 3: x3=3(3) | | | 3: x6=3(2) |
| | | | | | ● 4: x6=4(3) |
| | | | | | 5: x6=5(1) |

ТЕСТОВЫЙ ВХОД (TEST CASE)



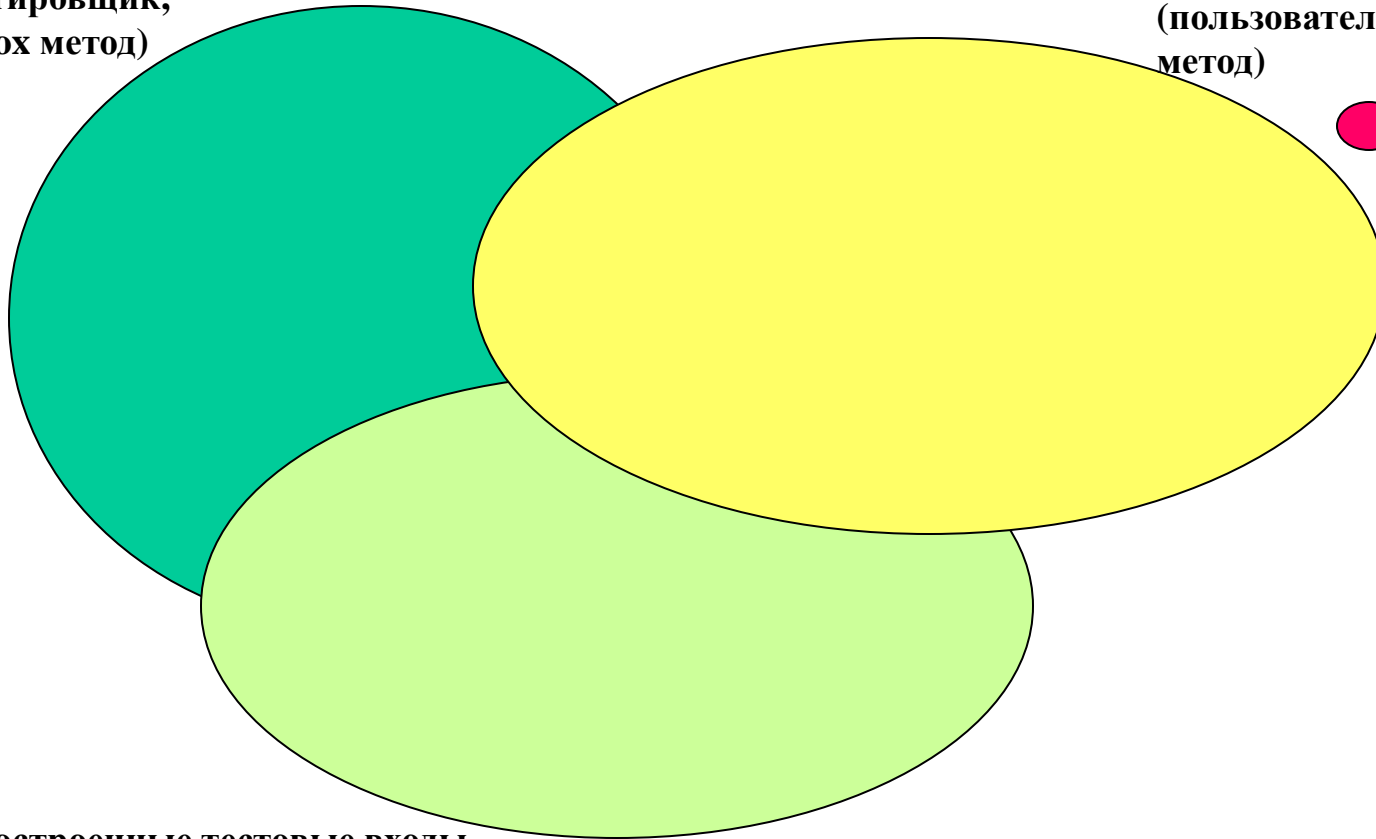
Основные алгоритмы:

- 1.Сужение (Reducing) исходного множества (по эквивалентности)**
- 2.Построение множества тестовых входов (test cases), которые покрывают суженное множество тестовых входов**

Дополнительная “размерность”

Спецификации
основанные на входе
(проектировщик,
white box метод)

Реальное поведение системы
(пользователь, black-box
метод)



**Мы
находимся
здесь**

Построенные тестовые входы
(test cases)
(специалист по тестированию, black-box метод,
Проверка модели - Model checking и др.)

**ПЛЮС: Динамика как развитие /
эволюция системы**

1. Предварительные сведения: роли и ответственность (пример: системное тестирование)

СИСТЕМНЫЙ СПЕЦИАЛИСТ:

- * производительность системы
- * безопасность системы
- * жизненный цикл системы
- * новые требования
- * новые поколения системы
- * новые стандарты

ПОЛЬЗОВАТЕЛЬ:

- * тестирование функций системы (functional test)

ПРОЕКТИРОВЩИК:

- * тестирование блоков (частей)
- * интегрирующие тест (integration test)

СПЕЦИАЛИСТ ПО ТЕСТИРОВАНИЮ:

- * проверка моделей системы (model checking)
- * др.

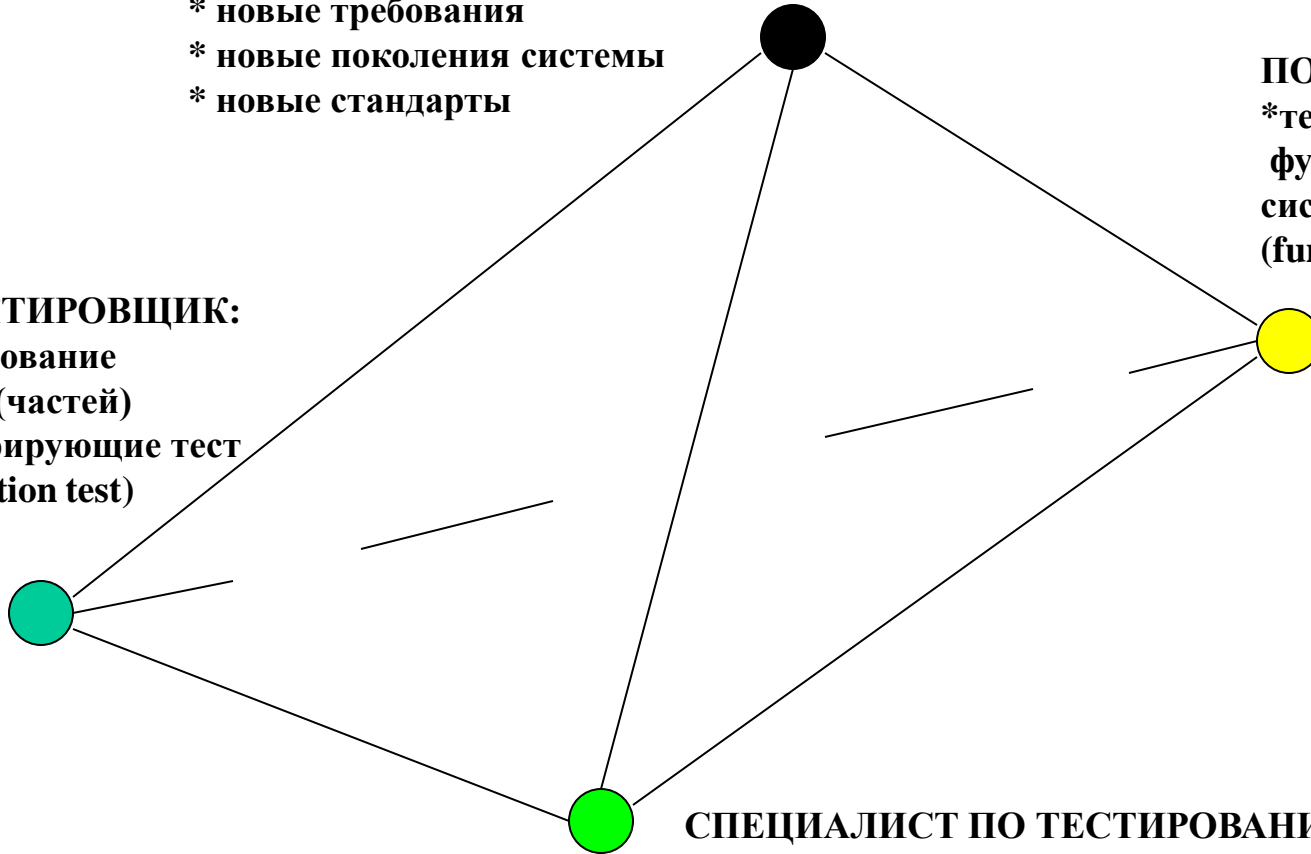
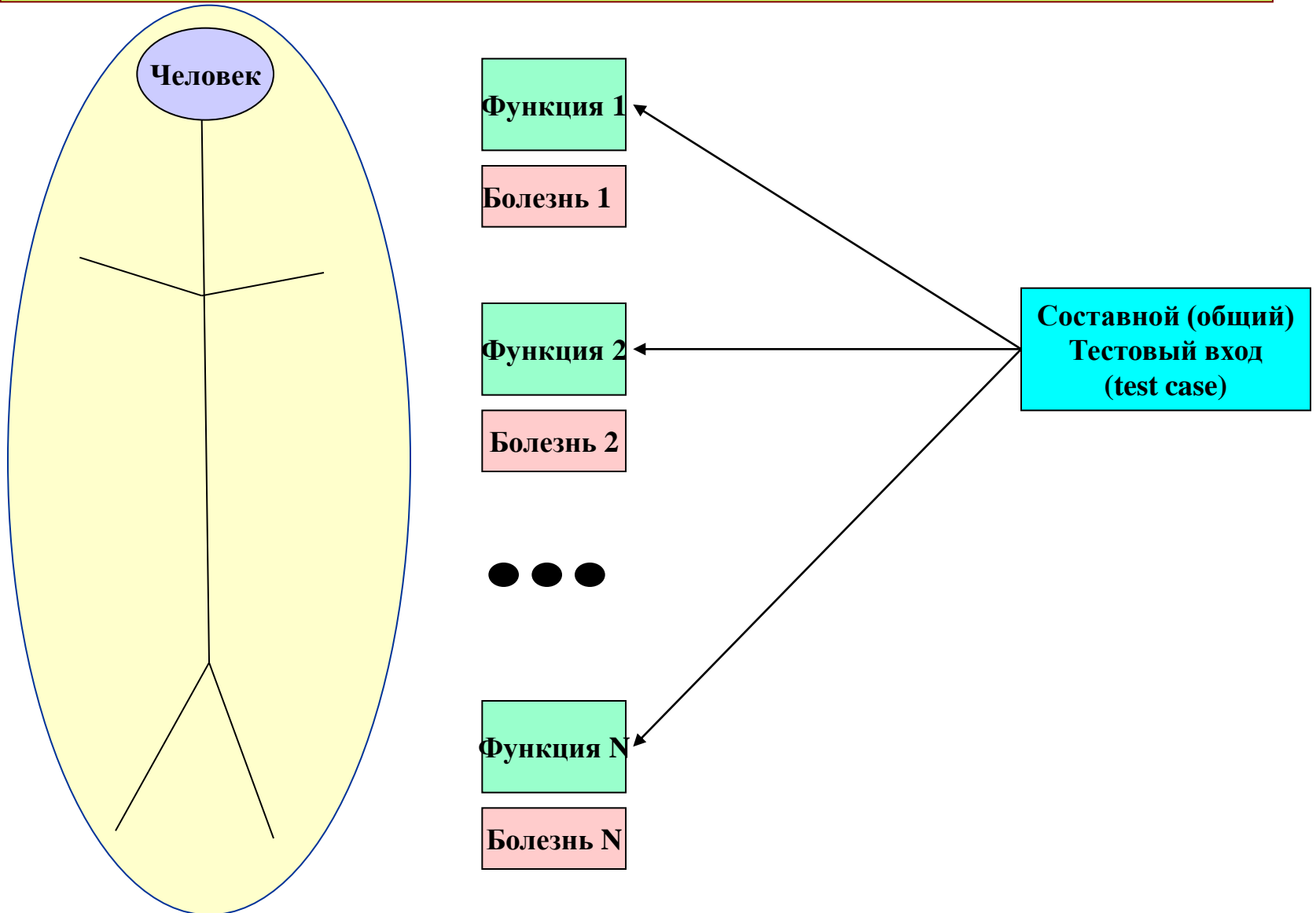
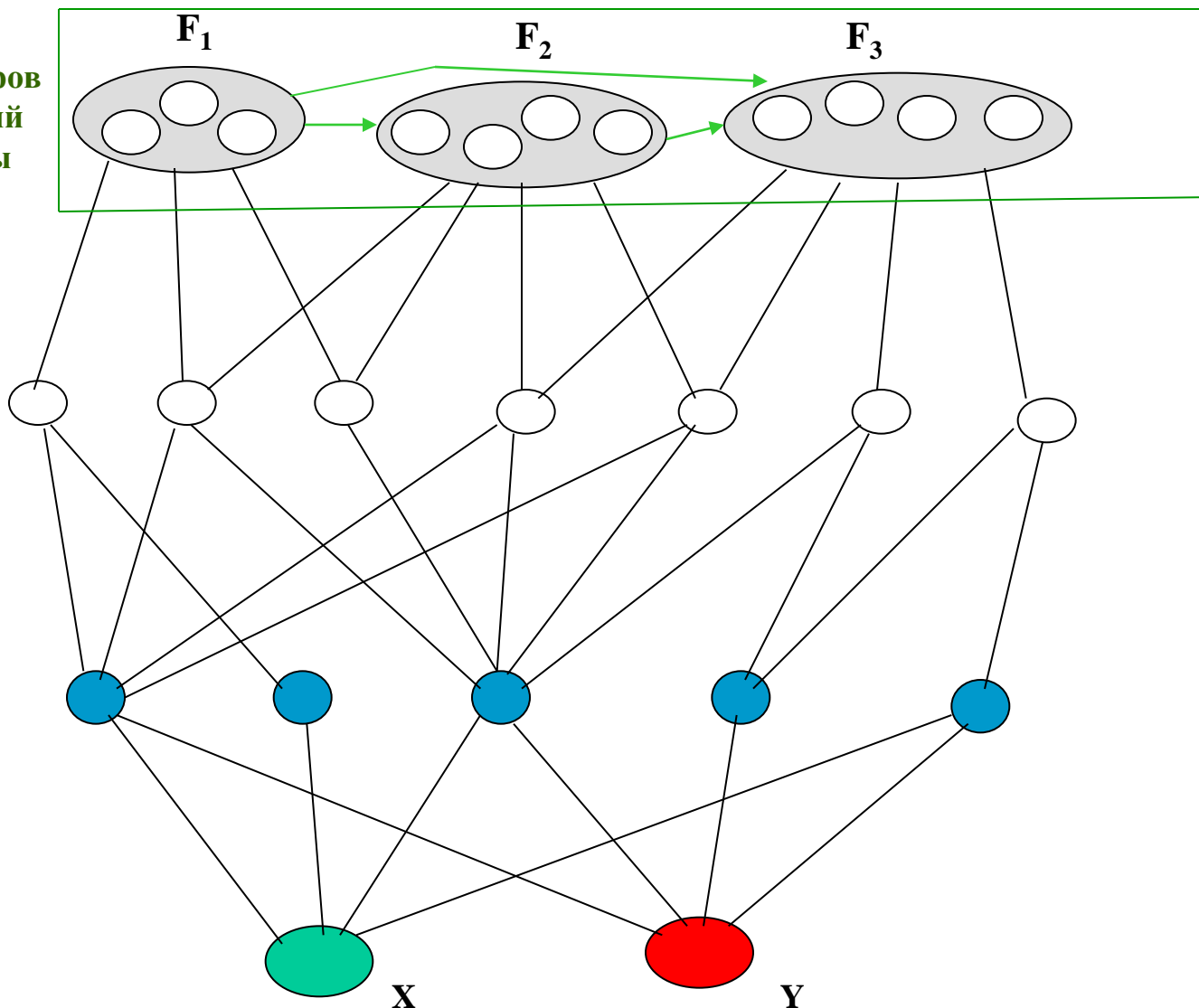


Иллюстрация много-функционального тестирования на примере человека: N функций и составной тестовой вход



Иерархия характеристик системы при много-функциональном тестировании

Орграф
Кластеров
Функций
системы



Кластеры
Функций
системы

Функции
системы

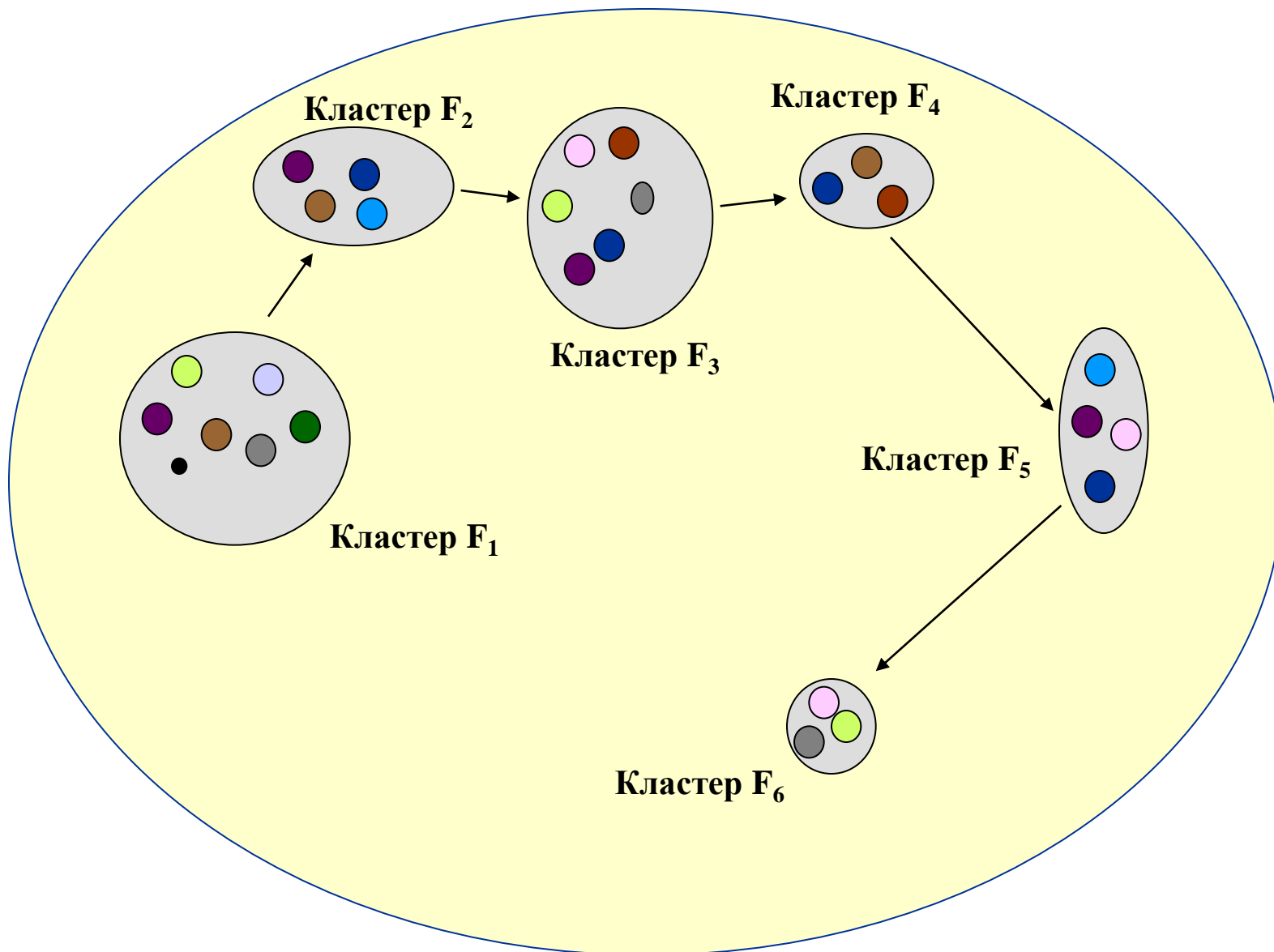
Отношения
Вход & Выход

Вход & Выход

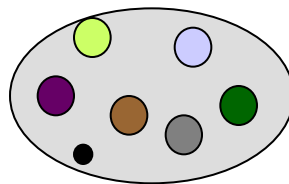
Уровни процесса тестирования и проблемы



«Пространство» функций системы и кластеров функций



Тестовый вход (Test case) для кластера функций и цепочка (последовательность) тестовых входов для цепи кластеров функций



Кластер системы



Тестовый вход (Test case) 1

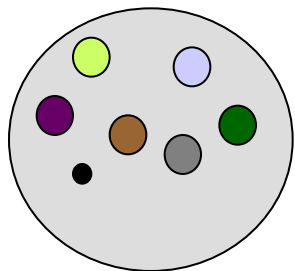


Тестовый вход (Test case) 2

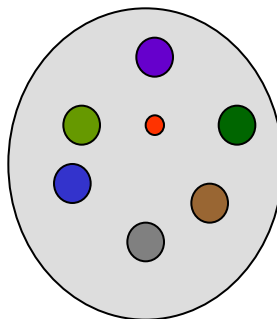


Тестовый вход (Test case) 3

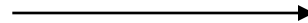
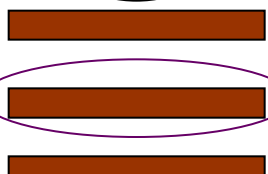
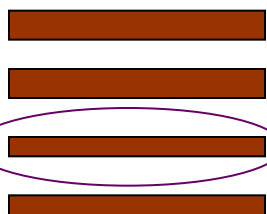
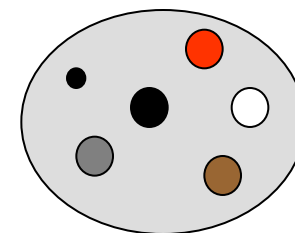
Кластер F_1



Кластер F_2

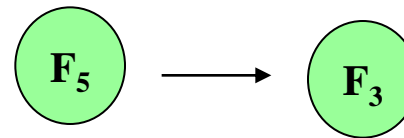
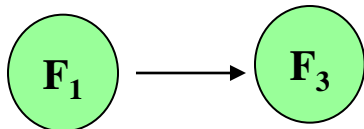
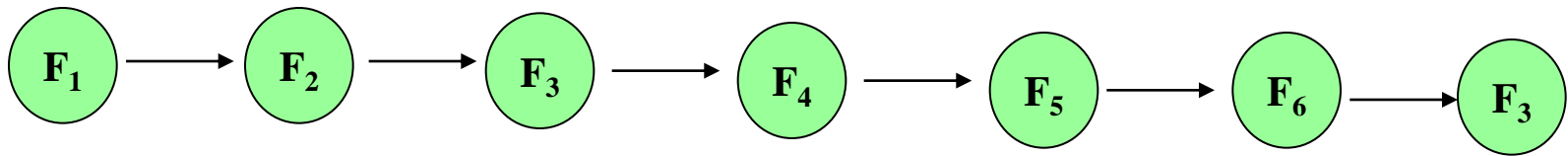
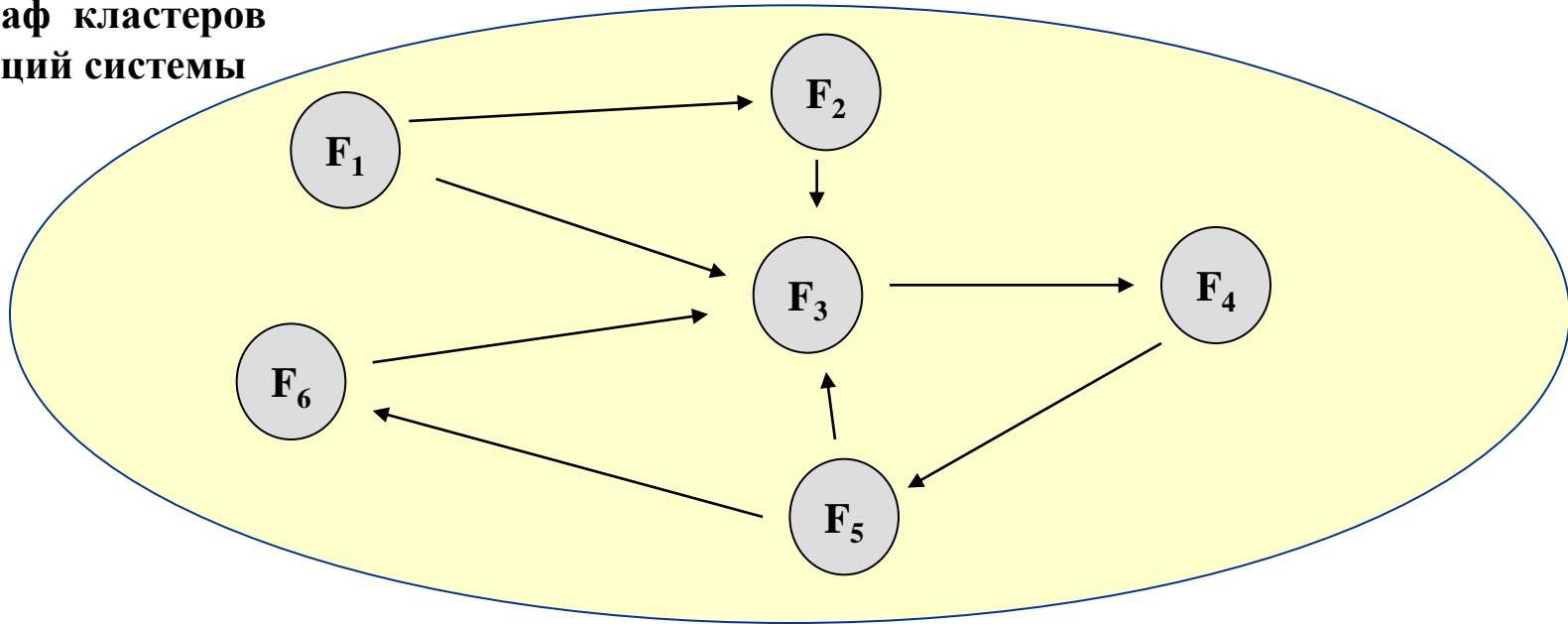


Кластер F_3

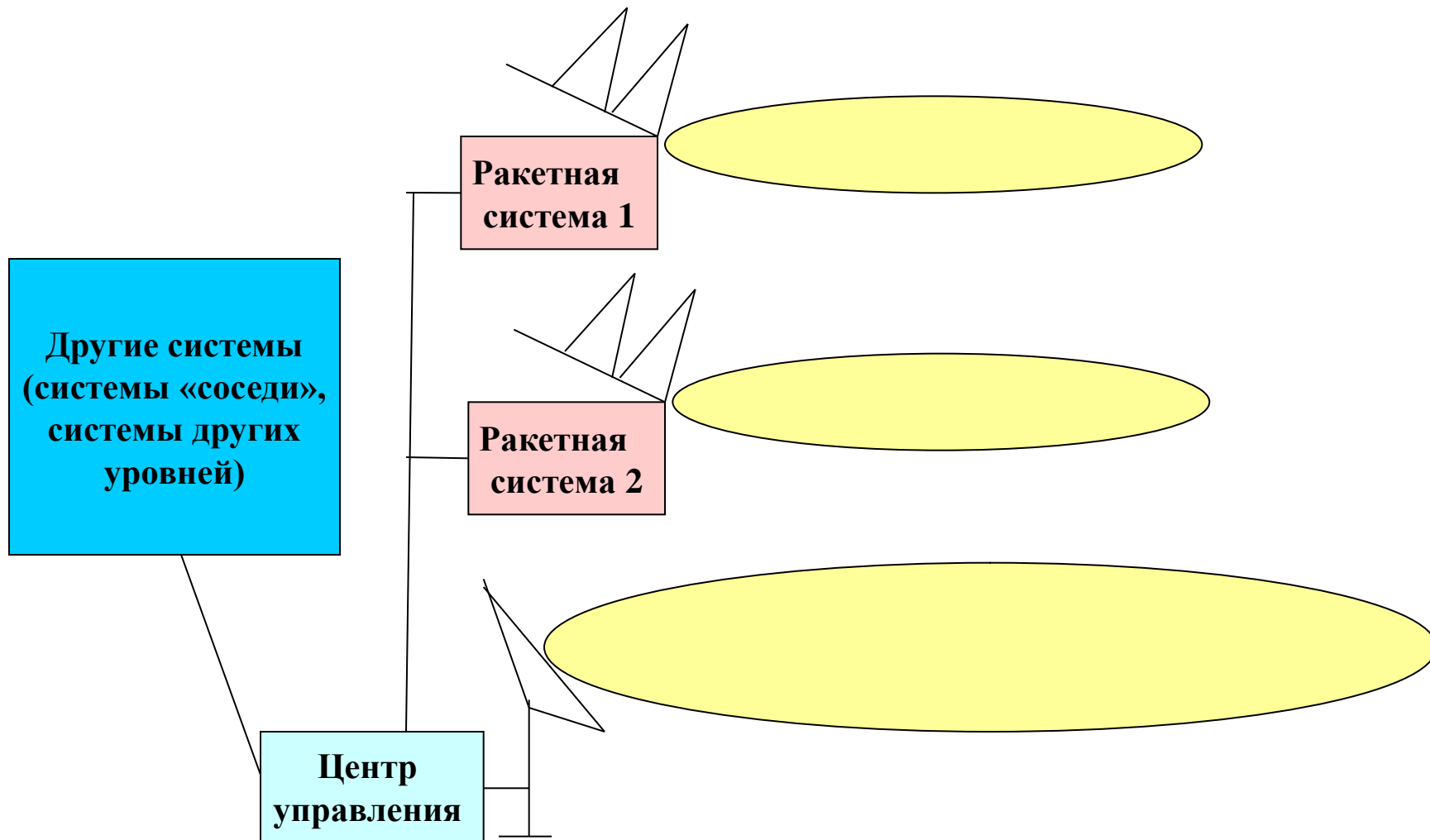


Цепи кластеров функций и покрытие

Орграф кластеров функций системы



Прикладная оборонная система (против ракет / самолетов)



Функции и кластеры функций

Функции:

1. Сканирование исследуемой области f_1

2. Захват f_2

3. Идентификация f_3

4. Сопровождение целей f_4

5. Multi-target multi-track
assignment f_5

6. Управление огнем (назначение
Ракет по целям) f_6

7. Удаление неопасных целей f_7

8. Получение данных от др. систем f_8

9. Отправки данных к др. системам f_9

Кластеры функций

$F_1: f_1$

$F_2: f_1, f_4$

$F_3: f_2, f_3, f_4$

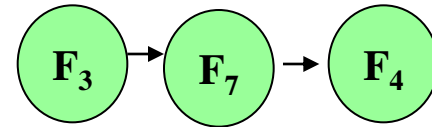
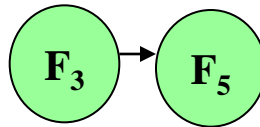
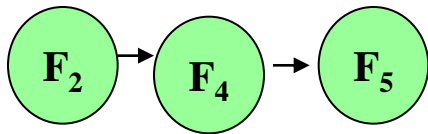
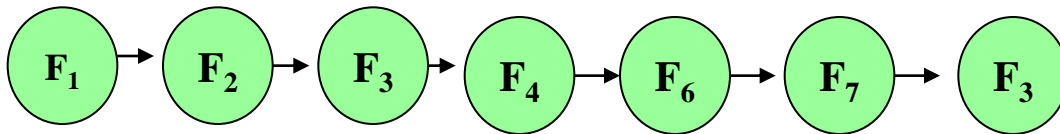
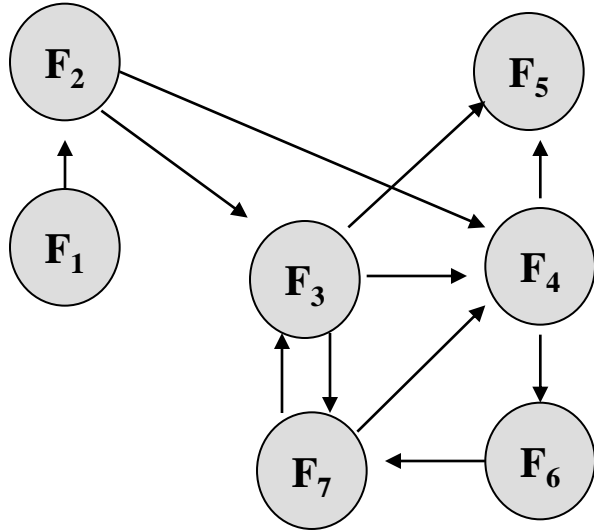
$F_4: f_4, f_5$

$F_5: f_4, f_7$

$F_6: f_5, f_6$

$F_7: f_5, f_8, f_9$

Орграф кластеров функций



1. M. Sh. Levin, M. Last, *Multi-Function System Testing: Composition of Test Sets*. 8th IEEE Int. Conf. HASE 2004, Tampa, FL, 99-108, 2004.
2. M. Sh. Levin, M. Last, *Test Case Sequences in System Testing: Selection of Test Cases for a Chain (Sequence) of Function Clusters*. 17th Int. Conf. IEA/AIE, Ottawa, LNCS 3029, Springer, 895-904, 2004.
3. M. Sh. Levin, M. Last, *Collection of Test Case Sequences Covering of Function Cluster Digraph*. IASTED Int. Conf. "AI and Applications", Innsbruck, 806-810, Febr. 2004.

ЛЕКЦИЯ 27. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационных сетей и систем”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.27. Диагностика систем, оценивание, улучшение

ПЛАН:

1. Иерархический подход к диагностике сложных систем

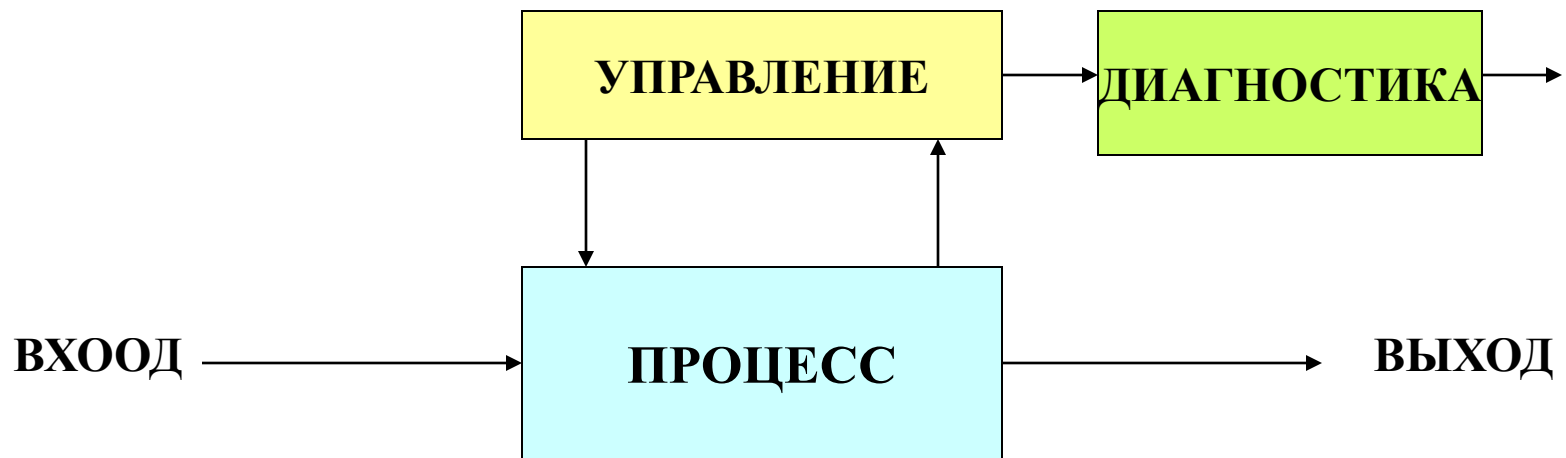
2. Иерархическое оценивание составной системы: пример для здания:

*модель здания и шкалы оценки для частей здания, *метод интегрирующих таблиц,

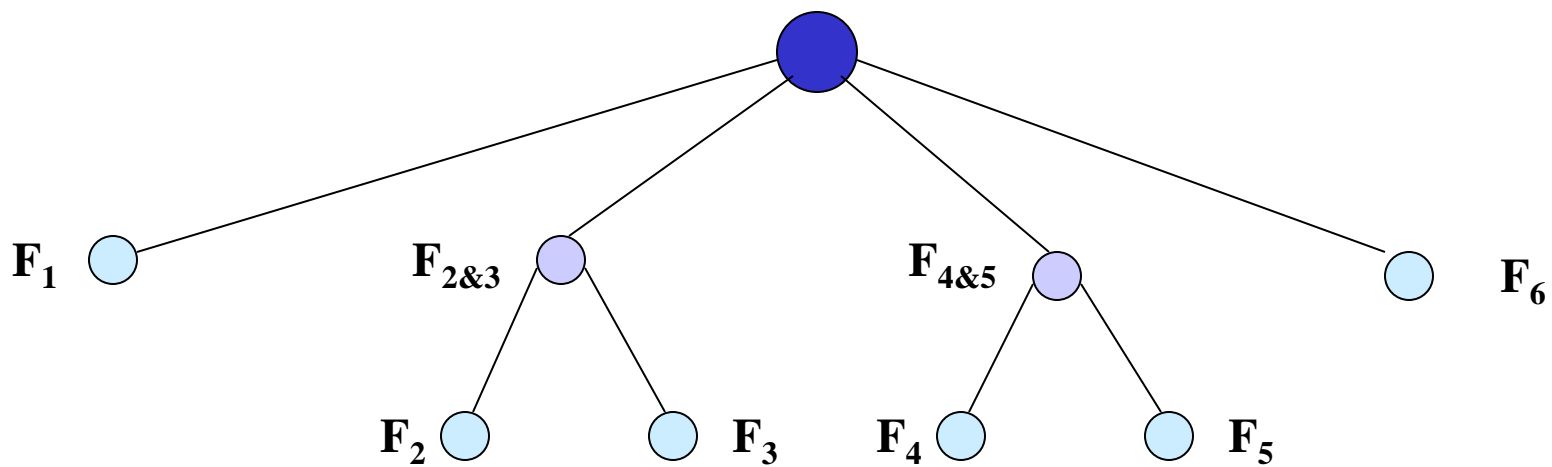
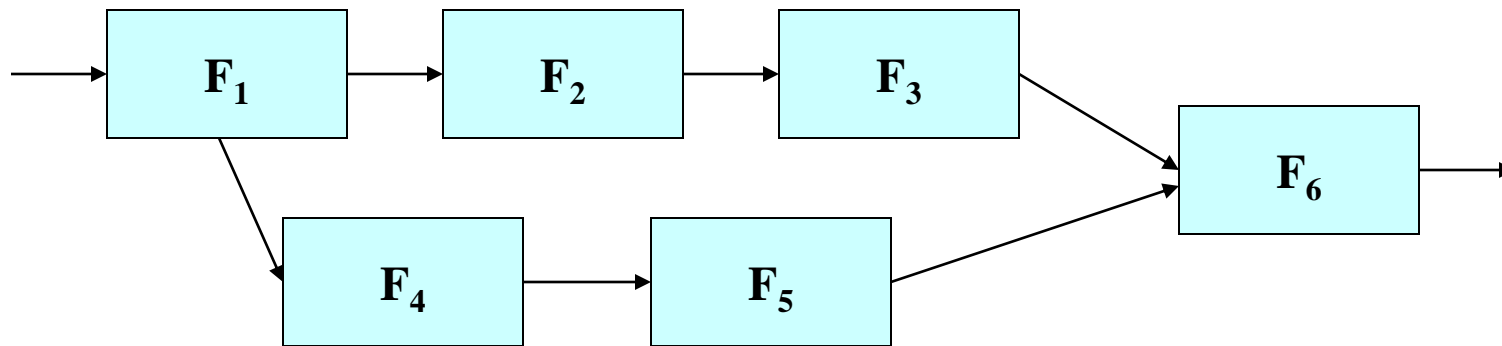
*иерархический комбинаторный синтез, *операции изменения и планирование процесса upgrade

Ноябрь 12, 2004

Много-уровневая диагностика сложной (составной) системы



ПРОЦЕСС



Много-уровневая диагностика сложной (составной) системы

ШКАЛА

1

2

3

4



РАЗРУШЕНИЕ

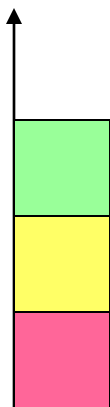
ПЛОХО

ХОРОШО

ОТЛИЧНО



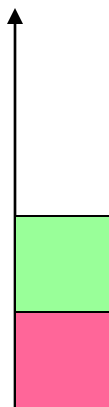
F₁



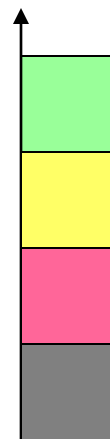
F₂



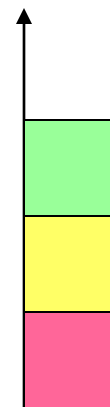
F₃



F₄

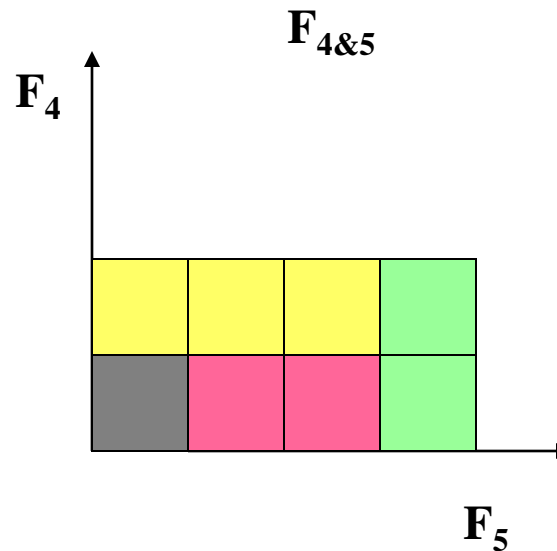
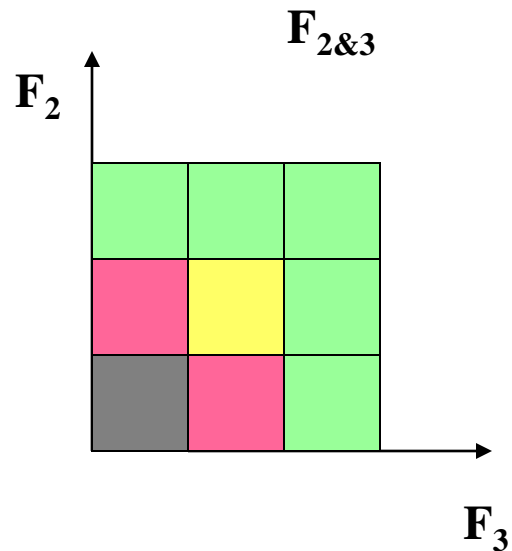


F₅

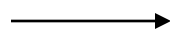


F₆

Много-уровневая диагностика сложной (составной) системы

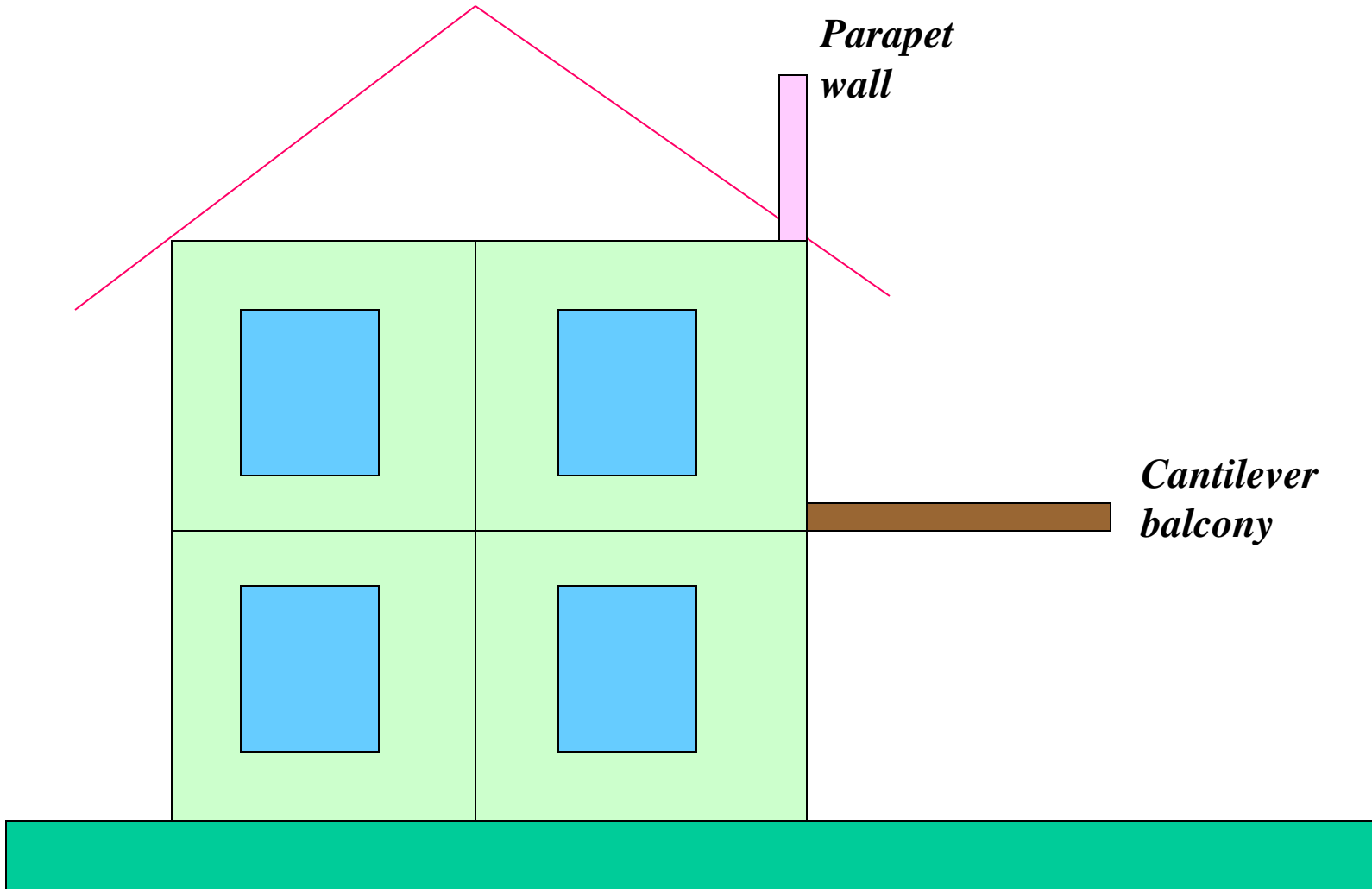


F_1 и $F_{2\&3}$ и $F_{4\&5}$ и F_6

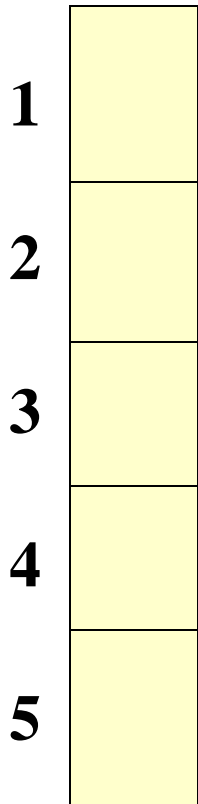


РЕЗУЛЬТИРУЮЩАЯ ОЦЕНКА

Example of building (evaluation from the viewpoint of earthquake engineering)



Generalized ordinal scale for damage



1. Distriction (global)

2. Distriction (local)

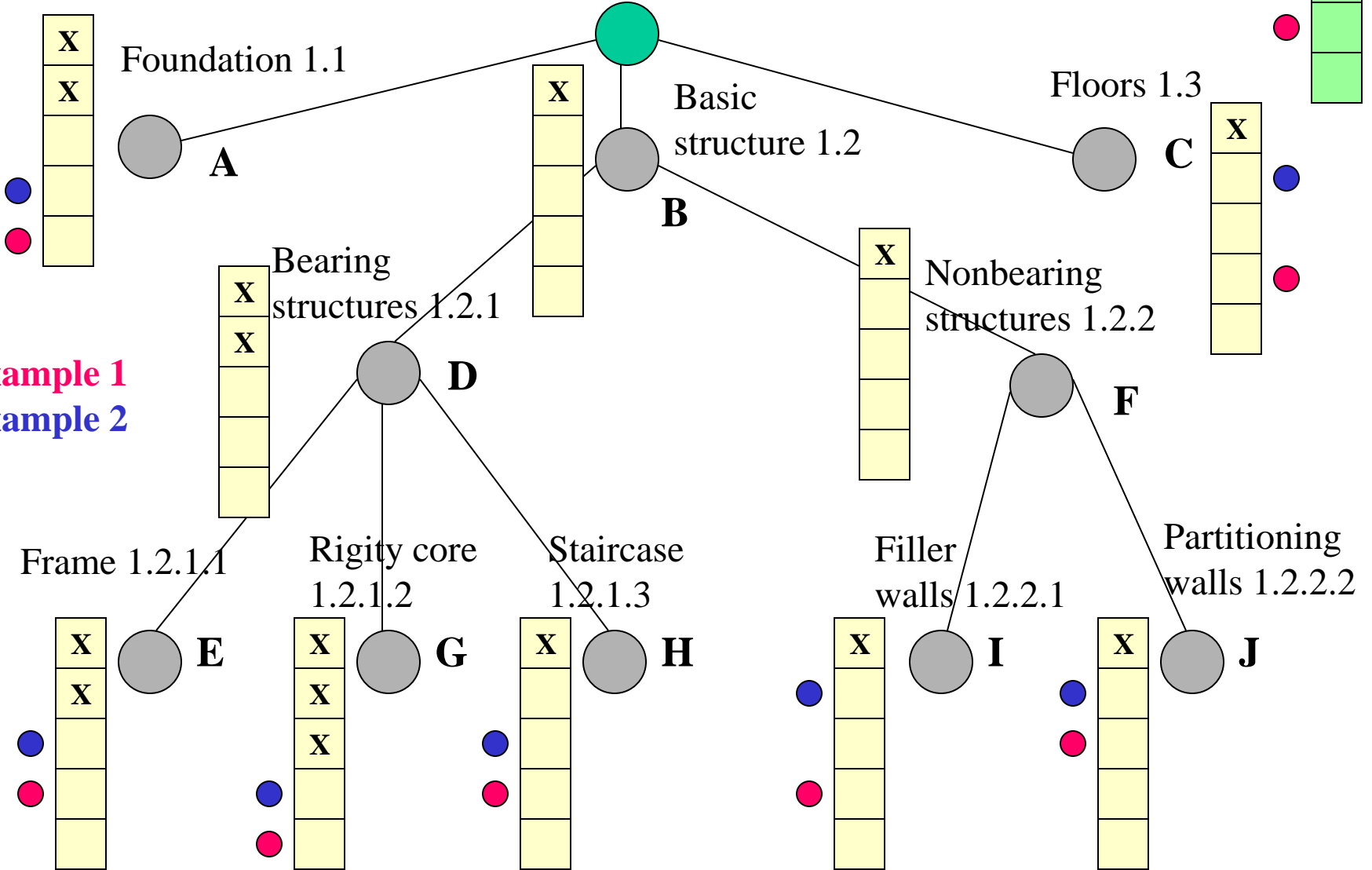
3. Chinks

4. Small chinks (hair like)

5. Without damage

Hierarchical model of building and corresponding scales

Building: $S = A * B * C$



Example 1
Example 2

Method 1: integration tables

Bearing structures D (1.2.1), scale [3,4,5]

E	G	H	D
3	4	3	3
3	4	4	3
3	4	5	-
3	5	3	3
3	5	4	3
3	5	5	-
5	4	3	3
5	4	4	4
5	4	5	4
5	5	3	4
5	5	4	4
5	5	5	5
4	4	3	3
4	4	4	4
4	4	5	-
4	5	3	3
4	5	4	4
4	5	5	4

Method 1: integration tables

Nonbearing structures **F** (1.2.2), scale [2,3,4,5]

2	2	-	-	2	I
3	3	-	-	3	
3	3	4	-	4	
-	4	4	5	5	
2	3	4	5		

J

Method 1: integration tables

Basic structure **B** (1.2), scale [2,3,4,5]

2	3	-	-	3	D
3	4	4	-	4	
-	4	5	5	5	
2	3	4	5		

F

Method 1: integration tables

Building S, scale [2,3,4,5]

A	B	C	S	A	B	C	S	A	B	C	S
3	2	2	2	5	2	2	2	4	2	2	2
3	2	3	-	5	2	3	-	4	2	3	-
3	2	4	-	5	2	4	-	4	2	4	-
3	2	5	-	5	2	5	-	4	2	5	-
3	3	2	2	5	3	2	-	4	3	2	-
3	3	3	3	5	3	3	-	4	3	3	3
3	3	4	3	5	3	4	3	4	3	4	3
3	3	5	-	5	3	5	3	4	3	5	-
3	4	2	-	5	4	2	-	4	4	2	-
3	4	3	-	5	4	3	-	4	4	3	-
3	4	4	-	5	4	4	4	4	4	4	4
3	4	5	-	5	4	5	4	4	4	5	4
3	5	2	-	5	5	2	-	4	5	2	-
3	5	3	-	5	5	3	-	4	5	3	-
3	5	4	-	5	5	4	-	4	5	4	-
3	5	5	-	5	5	5	5	4	5	5	-

Method 2: Hierarchical morphological design (combinatorial synthesis)

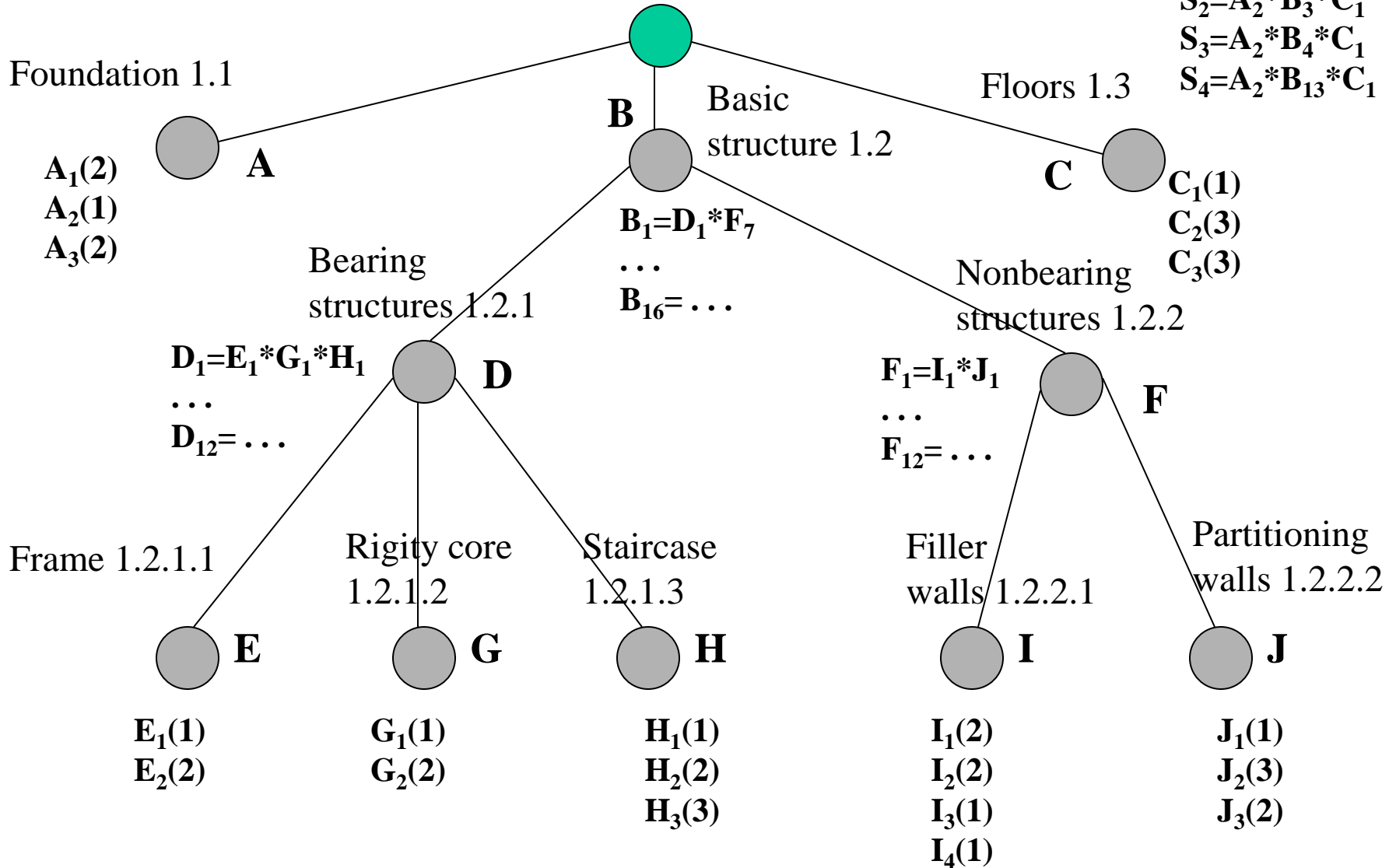
Building: $S = A * B * C$

$$S_1 = A_2 * B_1 * C_1$$

$$S_2 = A_2 * B_3 * C_1$$

$$S_3 = A_2 * B_4 * C_1$$

$$S_4 = A_2 * B_{13} * C_1$$



Design Alternatives for Building

Foundation A : A_1 (strip foundation), A_2 (bedplate foundation), A_3 (isolated parts)

Frame E : E_1 (monolith frame), E_2 (precast frame)

Rigidity core G : G_1 (monolith rigid core), G_2 (precast rigid core)

Stair case H : H_1 (monolith staircase), H_2 (precast staircase), H_3 (composite staircase)

**Filler walls I : I_1 (small elements), I_2 (curtain panel walls),
 I_3 (precast enclose panel walls), I_4 (frame walls)**

Partitioning walls J : J_1 (precast panel walls), J_2 (small elements), J_3 (frame walls)

Floors C : C_1 (monolith slabs), C_2 (composite slabs), C_3 (precast slabs)

Method 2: Hierarchical morphological design (combinatorial synthesis)

Compatibility

	G₁	G₂	H₁	H₂	H₃
E₁	3	2	3	1	2
E₂	2	1	2	1	2
G₁			3	2	1
G₂			2	1	1

	I₁	I₂	I₃	I₄
J₁	1	2	3	3
J₂	1	1	1	1
J₃	1	2	3	3

NOTE: 3 corresponds to the best level of compatibility
0 corresponds to incompatibility

Method 2: Hierarchical morphological design (combinatorial synthesis)

Compatibility

	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	F ₇	F ₈	F ₉	F ₁₀	F ₁₁	F ₁₂
D ₁	3	3	3	2	2	2	2	2	2	2	2	3
D ₂	2	2	2	2	2	2	2	2	2	2	2	2
D ₃	2	2	2	2	2	2	2	2	2	2	2	2
D ₄	2	2	2	2	2	2	2	2	2	2	2	2
D ₅	2	2	2	2	2	2	2	2	2	2	2	2
D ₆	2	2	2	2	2	2	2	2	2	2	2	2
D ₇	2	2	2	2	2	2	2	2	2	2	2	2
D ₈	2	2	2	2	2	2	2	2	2	2	2	2
D ₉	2	2	2	2	2	2	2	2	2	2	2	2
D ₁₀	2	2	2	2	2	2	2	2	2	2	2	2
D ₁₁	1	1	1	3	2	2	3	2	2	2	2	2
D ₁₂	2	2	2	2	2	2	2	2	2	2	2	2

NOTE: 3 corresponds to the best level of compatibility
0 corresponds to incompatibility

Method 2: Hierarchical morphological design (combinatorial synthesis)

Compatibility

	C₁	C₂	C₃	B₁	B₃	B₄	B₁₃
A₁	2	2	2	2	2	2	1
A₂	3	2	2	3	3	3	2
A₃	2	2	2	2	2	2	1
C₁				3	3	3	2
C₂				3	3	3	2
C₃				2	2	2	3

NOTE: 3 corresponds to the best level of compatibility
0 corresponds to incompatibility

Examples for building :

$$S^i = A_1 * (E_1 * G_1 * H_1) * (I_3 * J_1) * C_1 \quad \text{estimate 2} \quad (\text{Pareto-layer})$$

$$S^{ii} = A_2 * (E_2 * G_2 * H_2) * (I_3 * J_1) * C_1 \quad \text{estimate 2} \quad (\text{Pareto-layer})$$

$$S^{iii} = A_1 * (E_2 * G_2 * H_2) * (I_3 * J_1) * C_3 \quad \text{estimate 3}$$

$$S^{iv} = A_2 * (E_2 * G_2 * H_2) * (I_3 * J_1) * C_3 \quad \text{estimate 3}$$

$$S^v = A_1 * (E_2 * G_1 * H_1) * (I_3 * J_3) * C_3 \quad \text{estimate 4}$$

Operation group I (frames):

- O₁ increasing a geometrical dimension and active reinforcement**
- O₂ increasing of active reinforcement**

Operation group II (joints):

- O₃ increasing a level for fixing a longitudinal active reinforcement in zone of joints**
- O₄ decreasing the step of reinforced cross rods in zone of joint**

Operation group III (cantilever and cantilever balcony):

- O₅ decreasing the projection cantilever**
- O₆ supplementary supporting the cantilever**

Operation group IV (fronton and parapet wall):

- O₇ fixing a bottom part**
- O₈ designing a 3D structure (special)**

Operation group V (connection between frame and filler walls):

- O₉ design of shear keys**
- O₁₀ design of mesh reinforcement**
- O₁₁ partition of filler walls by auxiliary frame**

BINARY RELATIONS OVER IMPROVEMENT OPERATIONS

Binary relation “equivalence” and “nonequivalence”

Binary relation “complementarity” and “noncomplementarity”

Binary relation “precedence”

CRITERIA FOR IMPROVEMENT OPERATIONS

Group 1. Improvement of earthquake resistance

Group 2. Quality of architecture and plan decisions

Group 4. Utilization properties

Group 4. Expenditure

COMBINATORIAL MODELS FOR PLANNING OF IMPROVEMENT

Model 1: Knapsack

Model 2: Multiple choice problem

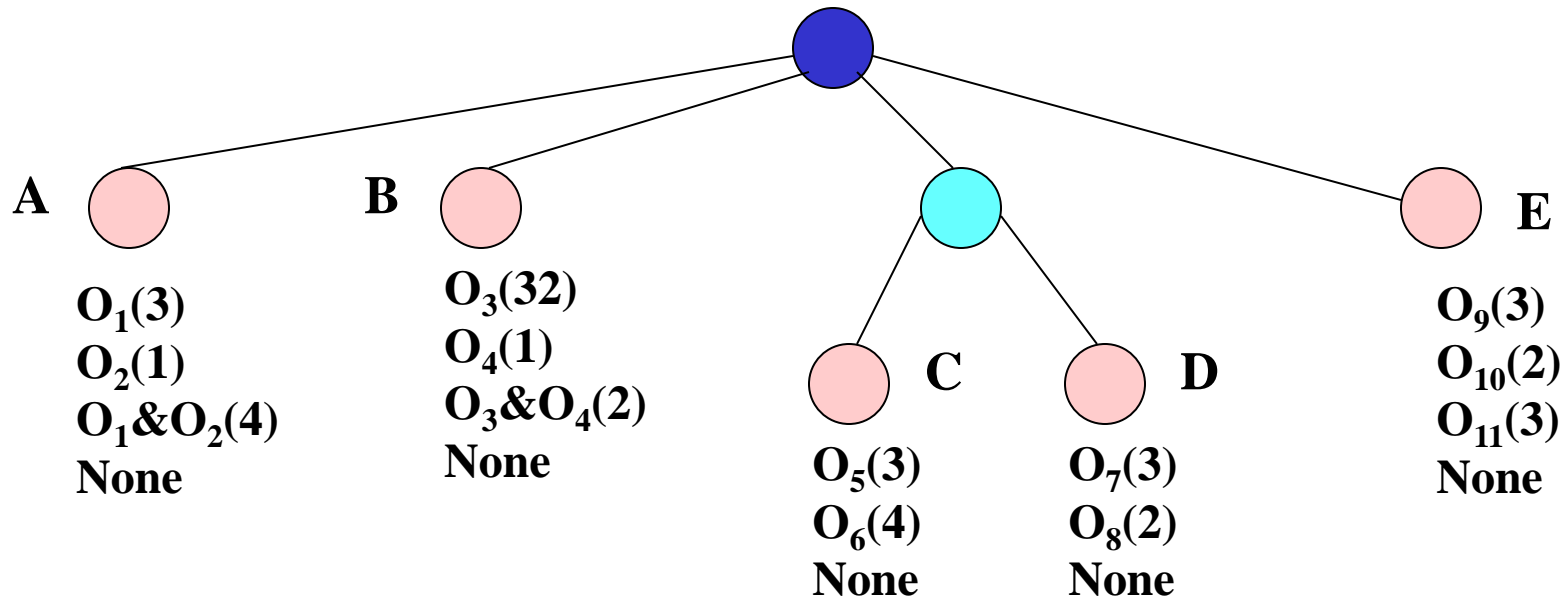
Model 3: Multiple criteria ranking

Model 4: Morphological clique problem

Model 5: Scheduling

ETC.

Improvement : $S = A * B * (C * D) * E$



Strategy: $O_2 \Rightarrow O_4 \Rightarrow O_5 \& O_7(4) \Rightarrow O_{10}$

ЛЕКЦИЯ 28. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.28. Техническое обслуживание

ПЛАН:

1. Предварительная информация: жизненный цикл, системы, эксплуатация, персонал, техобслуживание, ролинг

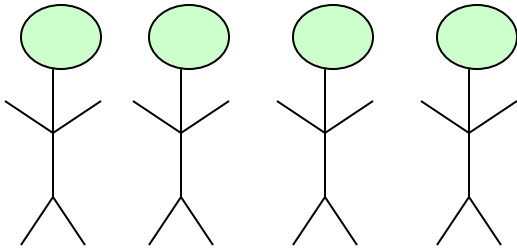
2. Схема процесса техобслуживания: *общая схема, *обслуживаемые системы,

*операции техобслуживания (инспекция как тестирование / анализ / диагностика, ремонт, *замена), *др.

3. Иллюстрации: базовый аналог (мониторинговая система), траектории неисправностей

Ноябрь 19, 2004

1.Предварительная информация



ПЕРСОНАЛ



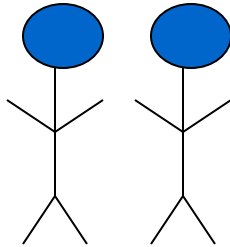
СИСТЕМА

1.Предварительная информация

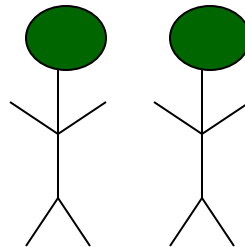


ЭКСПЛУАТАЦИЯ

(например:
шофер,
летчик,
оператор)



СИСТЕМА



ТЕХОБСЛУЖИВАНИЕ
(поддержка системы &
персонала)

АКЦИИ

ТЕХОБСЛУЖИВАНИЯ:

- *инсталляция системы
- *обучение персонала
- *тестирование персонала
- *планирование операций техобслуживания
- *выполнение операций (инспекция, ремонт, замена)
- *анализ систем, сбор данных, прогнозирование

ПРИМЕРЫ:

СИСТЕМА

ЭКСПЛУАТАЦИЯ

ТЕХОБСЛУЖИВАНИЕ

1.Автомобиль

Шофер, пассажиры

Персонал
техобслуживания

2.Самолет

Пилот, пассажиры

Персонал
техобслуживания

3.Человек

Человек

Врач, человек

4.Дом

Хозяин

Персонал
техобслуживания,
хозяин

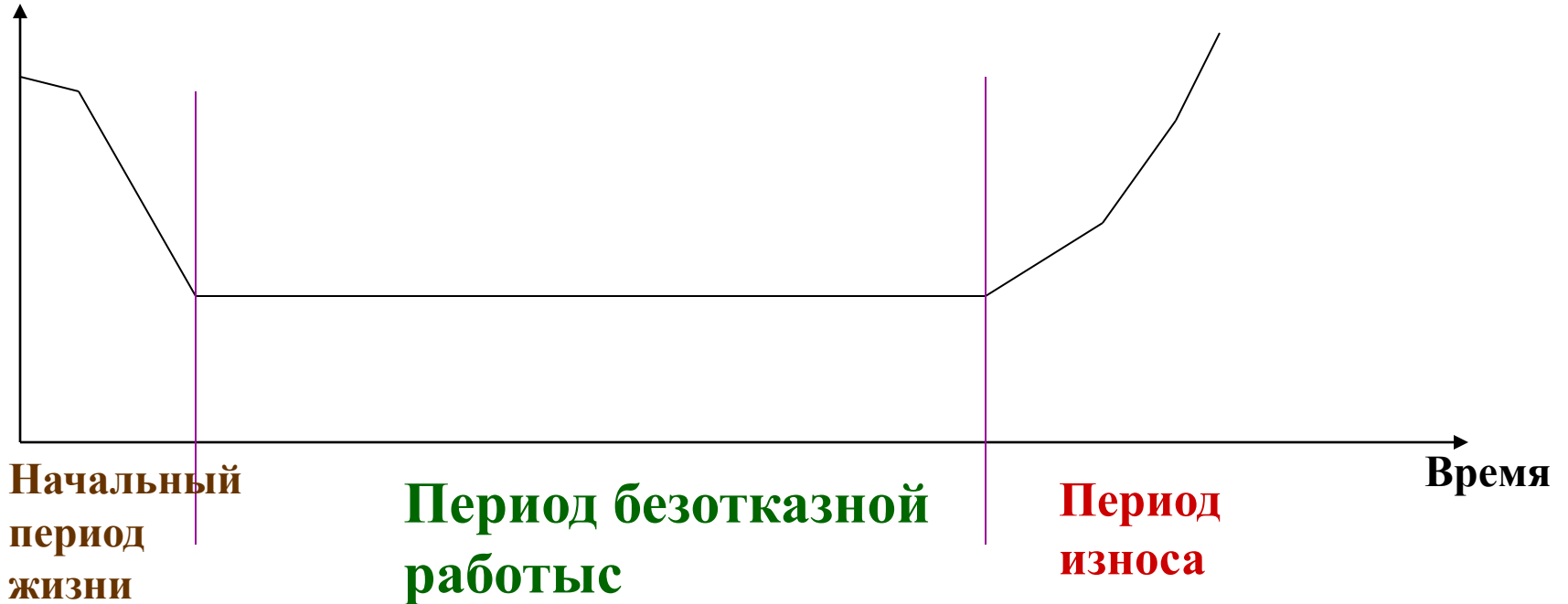
5.Компьютер

Пользователь

Персонал
техобслуживания,
специальные тестовые
программы, пользователь

ТИПОВАЯ КРИВАЯ ИНТЕНСИВНОСТИ ОТКАЗОВ

**Интенсивность
отказов**



ПОРЯДКОВАЯ ШКАЛА ДЛЯ ОТКАЗОВ:

1.Нормальная работа

2.Незначительный отказ

3.Существенный отказ

4.Разрушение

ОБСЛУЖИВАЕМЫЕ ОБЪЕКТЫ:

1.Система И/ИЛИ часть системы (компонент, блок)

2.Состояние системы

**3.Функция системы или кластер функций
(как группа взаимосвязанных функций)**

СХЕМА ТЕХОБСЛУЖИВАНИЯ (основные задачи):

1.Анализ / оценивание системы (системы и ее частей)

2.Предсказание поведение системы

3.Операционный менеджмент / профилактическое техобслуживание:

***тестирование**

***оценивание**

***дополнительная информация**

***ремонт / замена**

4.Проектирование моделей системы / ее частей

5.Системные стратегии:

***выбор объектов для техобслуживания**

***выбор операций (инспекция, ремонт, замена)**

***назначение времени проведения операций**

***выполнение операций**

2.Общая Схема

**Планиро-
вание**

**Модели
для
системы,
отказов,
прогнозов**

СИСТЕМА

**Операции техобслуживания
(сканирование, ремонт, замена)**

**Анализ,
сбор данных,
обработка,
интеграция
(синтез),
распределение
информации**

МОДЕЛИ:

- *выбор, *рюкзак, *маршрутизация, *назначение/размещение и др.**
- *вероятностные модели, процессы Маркова**
- *оценивание / анализ надежности**
- *анализ безопасности**
- *имитационное моделирование**
- *др.**

ВИДЫ ОБСЛУЖИВАЕМЫЕ СИСТЕМЫ:

1. «Целостная» система

2. Многокомпонентная (модульная) система:

***одно-уровневая модульная система**

***иерархическая (много-уровневая) система**

***много-уровневая модульная система**

с комплексными взаимосвязями модулей

(включая связи между различными уровнями и частями)

3. Развивающиеся системы (upgrade для компонентов, структуры, взаимосвязей)

4. Изменение внешней среды

ВИДЫ ТЕХОБСЛУЖИВАНИЯ :

1.БАЗОВОЕ ТЕХОБСЛУЖИВАНИЕ:

отказ => операция

2.ПРОФИЛАКТИЧЕСКОЕ ТЕХОБСЛУЖИВАНИЕ:

предсказание отказов =>

предварительная операция техобслуживания

3.АВТОМАТИЧЕСКОЕ ТЕХОБСЛУЖИВАНИЕ

ХАРАКТЕРИСТИКИ СИСТЕМ:

1.Надежность (устойчивость и др.)

2.Безопасность

3.Живучесть

4.Запас прочности, способность к восстановлению

4.Производительность

ЗАДАЧИ & МОДЕЛИ ПРОФИЛАКТИЧЕСКОГО ТЕХОБСЛУЖИВАНИЯ:

1.ВЫЯВЛЕНИЕ (РАЗМЕЩЕНИЕ) ТОЧЕК ТЕСТИРОВАНИЯ

Модели: многокритериальный выбор, задачи рюкзачного вида, задачи размещения и др.

2.ПЛАНИРОВАНИЕ ОПЕРАЦИЙ ТЕСТИРОВАНИЯ

Модели: многокритериальное ранжирование, задачи рюкзачного вида, планирование расписаний и др.

МОДЕЛИРОВАНИЕ&ОБРАБОТКА ИНФОРМАЦИИ:

- 1.Моделирование отказов**
- 2.Диагностика отказов**
- 3.Мониторинг отказов**
- 4.«Сопровождение» отказов**
- 5.Интеграция (синтез) информации о локальных отказах**
- 6.Распределение информации об отказах между различными информационными системами и специалистами**



СИСТЕМА

БАЗОВЫЕ АКЦИИ:

- 1.Сканирование (тестирование) элементов**
- 2.Несущественный ремонт / замена**
- 3.Существенный ремонт / замена**

СИСТЕМНЫЕ УРОВНИ

- 1.Система**
- 2.Части системы**
(группы состояний,
группы функций)
- 3.Компоненты**
(блоки, состояния,
функции)

3.Базовый аналог: Мониторинговая система

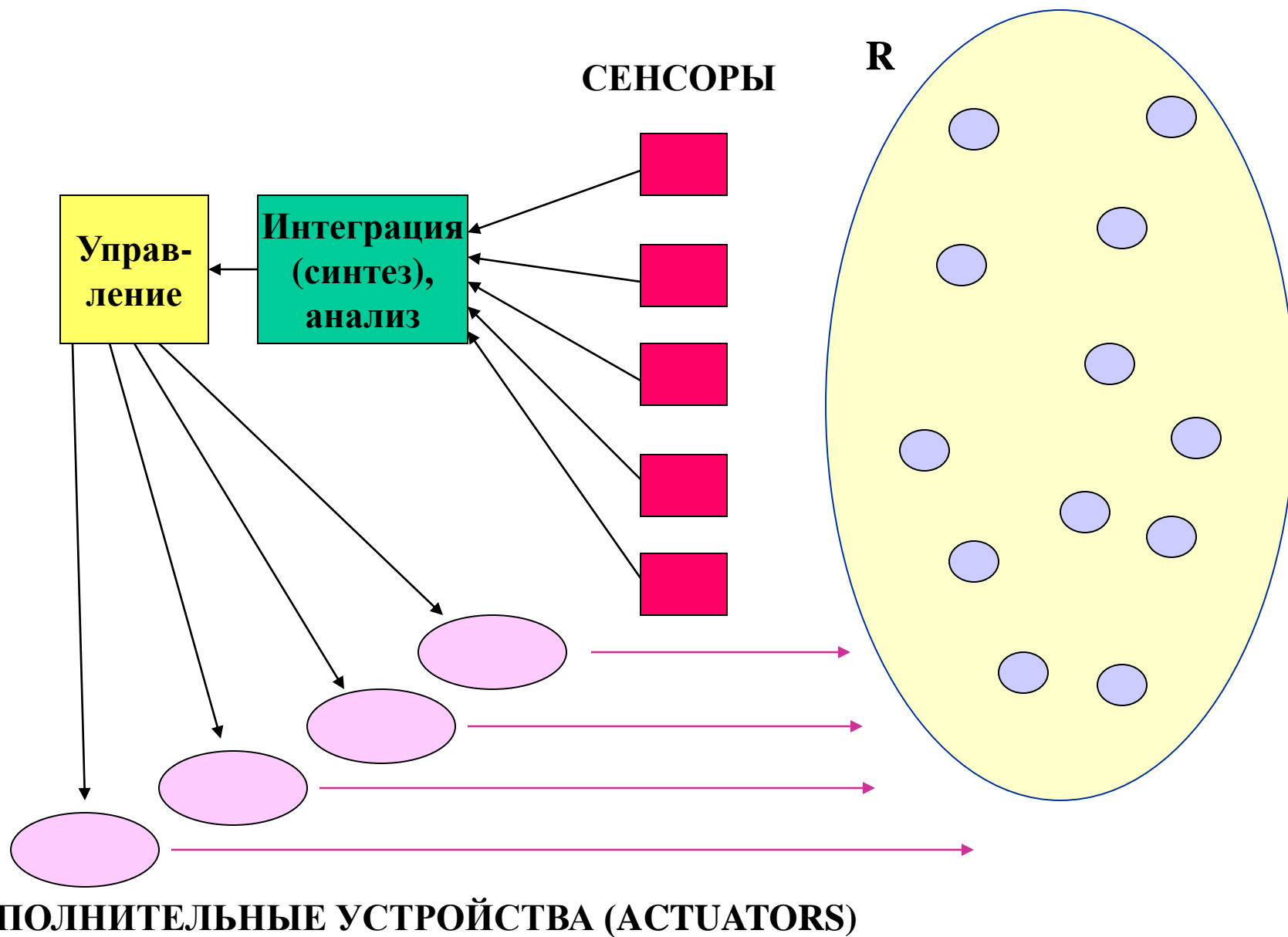
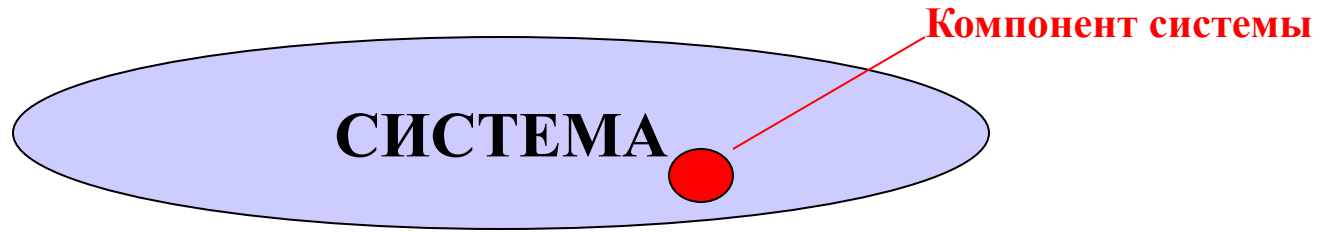
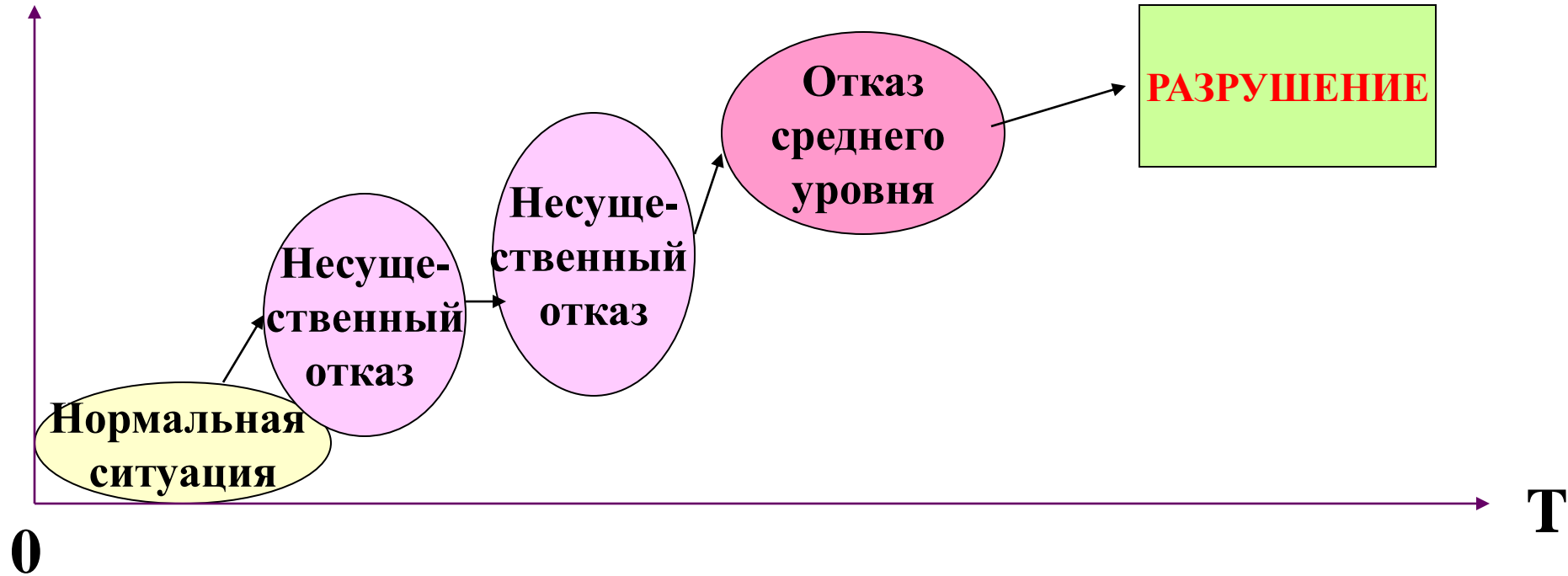


Illustration: Accumulation of information



Траектория отказов (для компонента)



ЛЕКЦИЯ 29. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.29. Технология требований (requirements engineering)

PLAN:

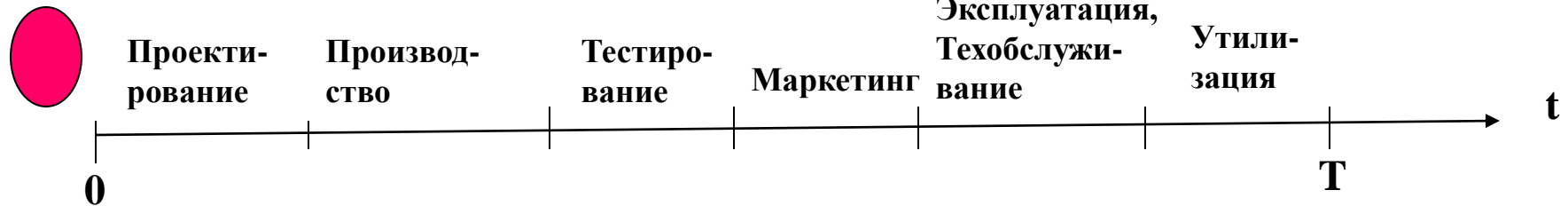
1. Разработка требований к системе (Requirements engineering): предварительные сведения
 2. Типы требований
 2. Дополнения
3. Анализируемые системы
4. Модели

Ноябрь 26, 2004

1.Предварительные сведения



1.Предварительные сведения



ОПЫТ В РОССИИ: NB!!!

БАЗОВЫЕ ПУБЛИКАЦИИ:

1. Ralph R. Young, *The Requirements Engineering Handbook*, Artech House, 2004 (Carnegie Mellon Univ.)
2. S. Robertson, J. Robertson, *Mastering the Requirements Process*. Addison-Wesley, 1999.
3. K.E. Wieggers, *Software Requirements*. 2nd ed., Microsoft Press 2003.

ЖУРНАЛЫ: “Requirement Engineering” (Springer), etc.

КОНФЕРЕНЦИИ:
IEEE Requirement Engineering Conference, etc.

1.Предварительные сведения: уровни (продукты, требования, стандарты)



1.Стандарты

2.Требования к системе

3.Система (продукт, семейство продуктов, платформа)

1.Предварительные сведения



Соседние дисциплины:

- 1.Системный инжиниринг
(Systems engineering, & life cycle engineering)
- 2.Стратегический менеджмент
- 3.Маркетинг
- 4.Прогнозирование
- 5.Технологии знаний (выявление опыта)

1. Предварительные сведения: роли и ответственность (пример: системное тестирование)

СИСТЕМНЫЙ СПЕЦИАЛИСТ:

- * производительность системы
- * безопасность системы
- * жизненный цикл системы
- * новые требования
- * новые поколения системы
- * новые стандарты

ПОЛЬЗОВАТЕЛЬ:

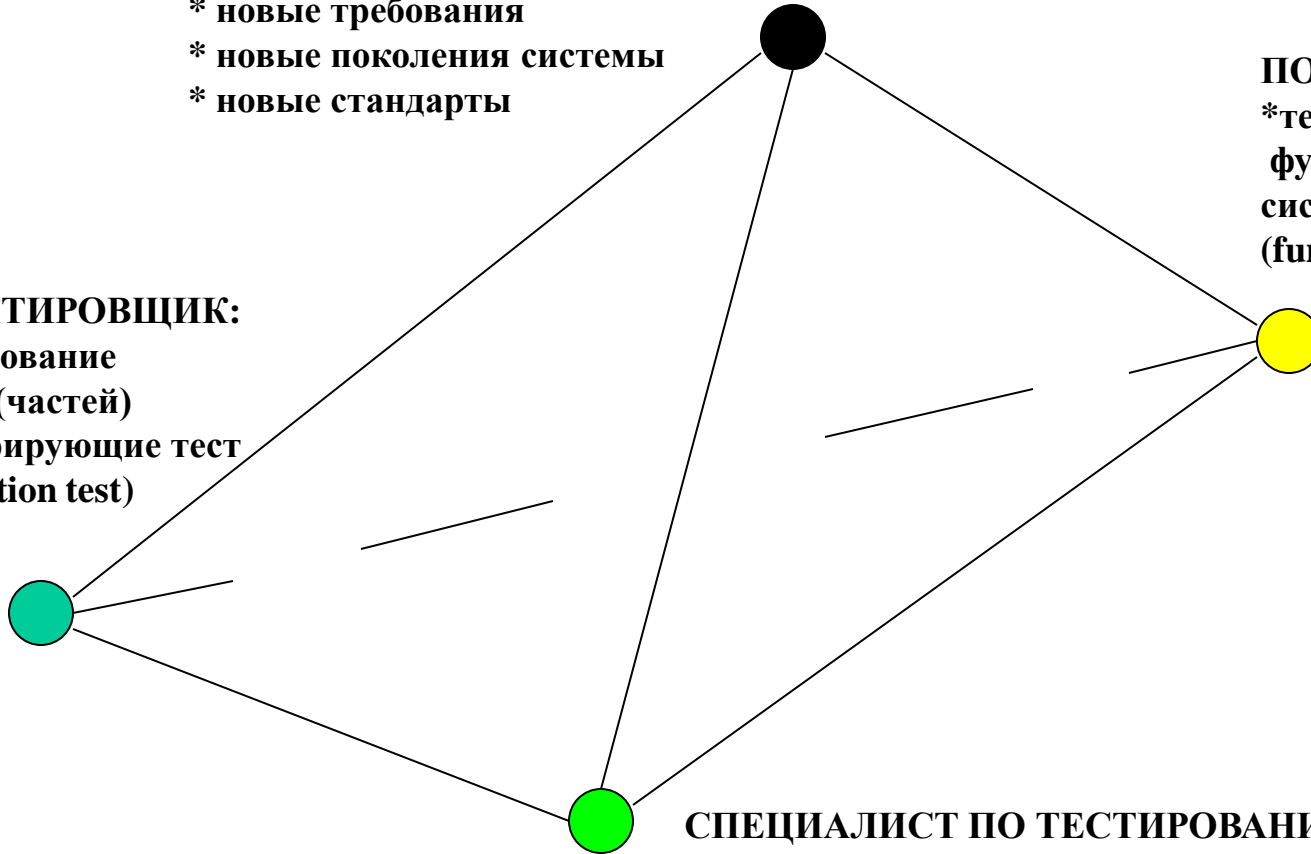
- * тестирование функций системы (functional test)

ПРОЕКТИРОВЩИК:

- * тестирование блоков (частей)
- * интегрирующие тест (integration test)

СПЕЦИАЛИСТ ПО ТЕСТИРОВАНИЮ:

- * проверка моделей системы (model checking)
- * др.



1.Предварительные сведения



Проектировщик
Пользователь
Заказчик
Системный специалист

Спецификация требований

СИСТЕМА

1.Предварительные сведения



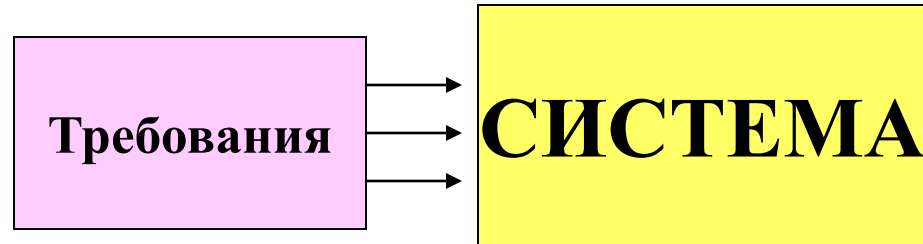
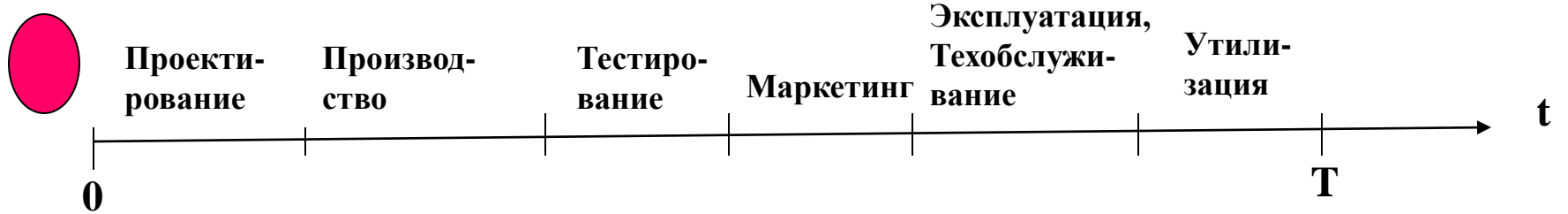
Проектировщик
Пользователь
Заказчик
Системный специалист

Спецификация требований

Персонал с высоким уровнем мастерства (ГРАМОТНОСТЬ)

СИСТЕМА

1.Предварительные сведения



1.Предварительные сведения

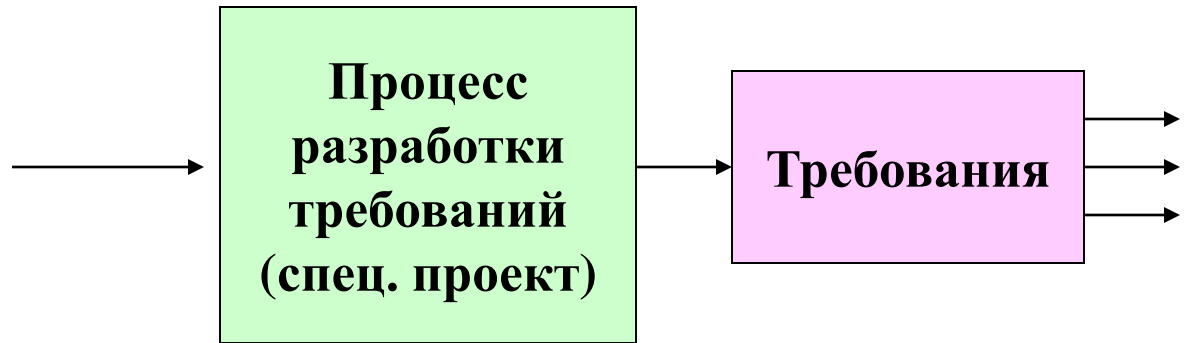


1.Предварительные сведения



ИСТОЧНИКИ:

- 1.Информация от пользователя
- 2.Прежний опыт (проектирование, Производство и т.п.)
- 3.Системы-аналоги
- 4.Случая из истории техники (Use Cases)



1.Предварительные сведения



ИСТОЧНИКИ:

- 1.Информация от пользователя
- 2.Прежний опыт (проектирование, Производство и т.п.)
- 3.Системы-аналоги
- 4.Случая из истории техники (Use Cases)

Процесс разработки требований (спец. проект)

Требования

ПРОБЛЕМЫ & МЕТОДЫ:

- 1.Системный анализ
- 2.Открытия
- 3.Выявление знаний, опыта, мастерства
- 4.Структуризация & интеграция
- 5.Моделирование
- 6.Анализ динамики
- 6.Тестирование
- 7.Прогнозирование

ТИПЫ ТРЕБОВАНИЙ:

- 1. Требования бизнеса (Business requirements)**
 - 2. Требования пользователя (User's requirements)**
 - 3. Требования системы со стороны более высокого уровня системной иерархии (High-level or system requirements)**
 - 4. Функциональные требования (система должна выполнять функции ...)**
 - 5. Нефункциональные требования система должна иметь характеристики ...)**
 - 6. Проектные требования / ограничения**
 - 7. Ограничения технологии производства**
 - 8. Требования по производительности (Performance requirements)**
 - 9. Требования к интерфейсу (с др. системами)**
 - 10. Требования по квалификации персонала (Qualification requirements)**
 - 11. Логистические требования (Logistics requirements)**
 - 12. Требования среды (например, экологические)**
 - 13. Требования к системе, подсистеме и компонентам**
 - 14. Повторное использование требований (Reusing)**
- ЕТС.**

ДОПОЛНЕНИЯ:

1. Критерии для оценивания требований

2. Метод прототипов

3. Сценарии

**4. Повторное использование требований
(reusing)**

ОБЪЕКТ & ИЕРАРХИЯ:

1. Система И / ИЛИ ее часть (компонент, блок)

**2. Состояние системы, группа состояний,
блок-диаграмма состояний**

**3. Функция системы, кластер функций ,
орграф кластеров функций**

ОБЪЕКТ & ИЕРАРХИЯ:

1. Система И / ИЛИ ее часть (компонент, блок)

**2. Состояние системы, группа состояний,
блок-диаграмма состояний**

**3. Функция системы, кластер функций ,
орграф кластеров функций**

**1.СИСТЕМЫ / ПРОДУКТЫ
2.СЕМЕЙСТВА ПРОДУКТОВ
3.ПЛАТФОРМА**

Сценарии:

1.Структура (цепи, деревья и др.) для системных состояний, функций

2.Количественные сценарии

3.Интеграция базовых случаев (use cases) & прогнозирование

Сценарии:

1.Структура (цепи, деревья и др.) для системных состояний, функций

2.Количественные сценарии

3.Интеграция базовых случаев (use cases) & прогнозирование

МОДЕЛИ:

1.Отношения на сущностях (Entity relationship)

2.Модель переходов на множестве состояний

3.Entity relationship &

Диаграмма переходов на множество состояний

Модели:

I. ИЕРАРХИЯ ТРЕБОВАНИЙ

- 1. Иерархия информации**
- 2. Интеграция информации (синтез)**

II. СЦЕНАРИИ

- 1. Концептуальные схемы (conceptual maps)**
- 2. Графы и сети**

III. МОДЕЛИРОВАНИЕ ДИНАМИКИ

- 1. Имитационное моделирование**
- 2. Тестирование и др.**

ЛЕКЦИЯ 30 (сокр. версия) Курс: “Проектирование систем: Структурный подход”
Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики
Московский физико-технический институт (университет)

Марк Ш. ЛЕВИН
Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.30. Задачи назначения / размещения

ПЛАН:

1. Задача размещения (формулировки как назначения, сопоставление, размещение):

*задача о назначении (паросчетаниях), *квадратичная задача о назначении (QAP),

*обобщенная задача о назначении (GAP), *сопоставление строк (иллюстрация), *multiple matching (иллюстрация)

2. Список базовых алгоритмических подходов

3. Схема эволюции задач типы размещения

4. Список прикладных областей

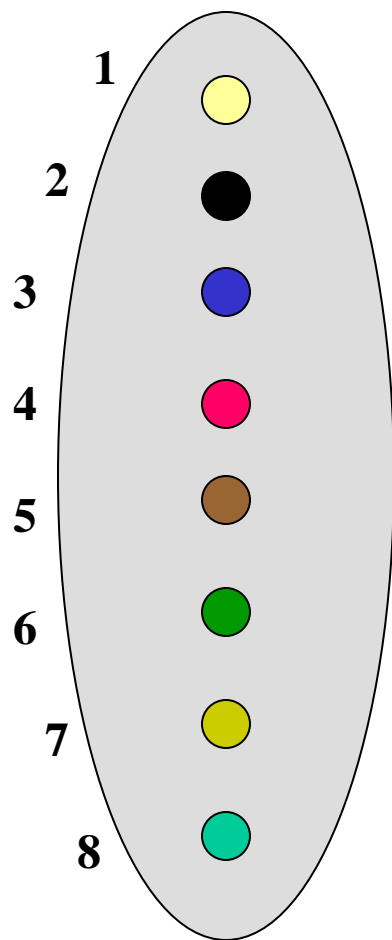
5. Базовая библиография (книги и сайты)

Дек. 3, 2004

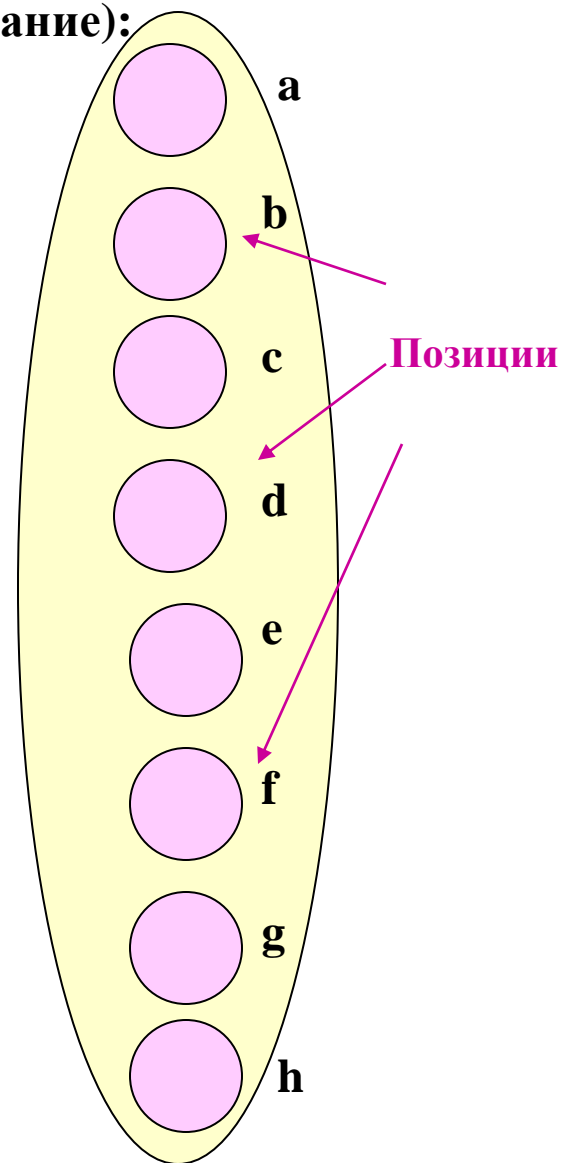
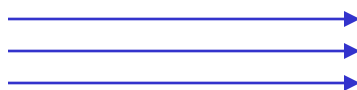
Задача размещения (Allocation problem)

Размещение (назначение, сопоставление, позиционирование):

Множество элементов
(персонал, оборудование и др.)



ОТОБРАЖЕНИЕ

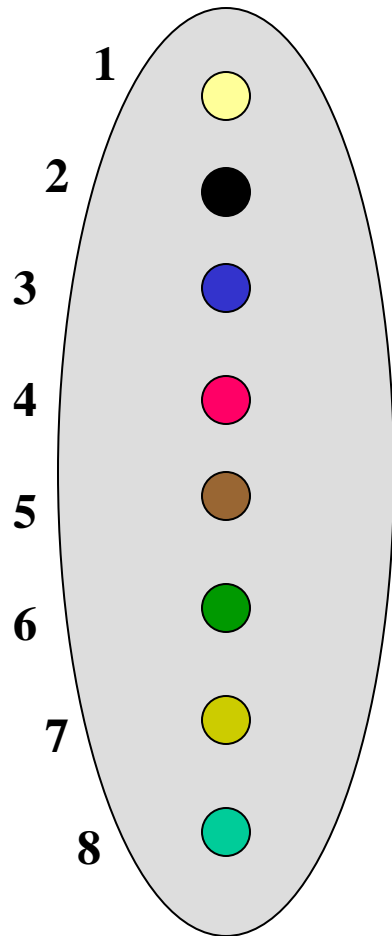


Двухдольный граф

Задача размещения (Allocation problem)

Размещение (назначение, сопоставление, позиционирование):

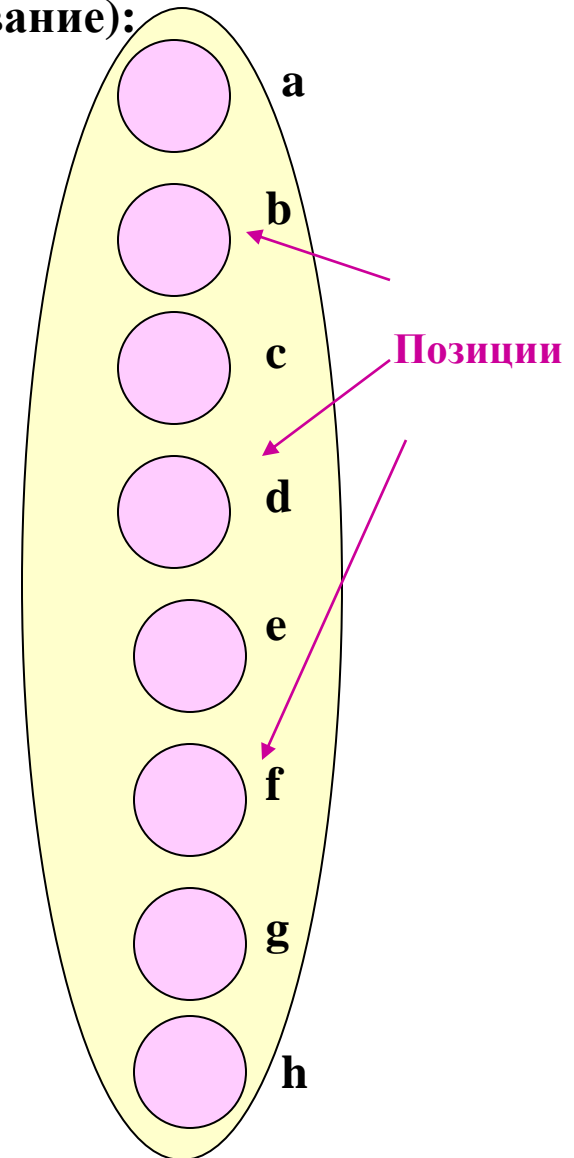
Множество элементов
(персонал, оборудование и др.)



Матрица весов c_{ij}

	a	b	c	d	e	f	g	h
1
2
3
4
5
6
7
8

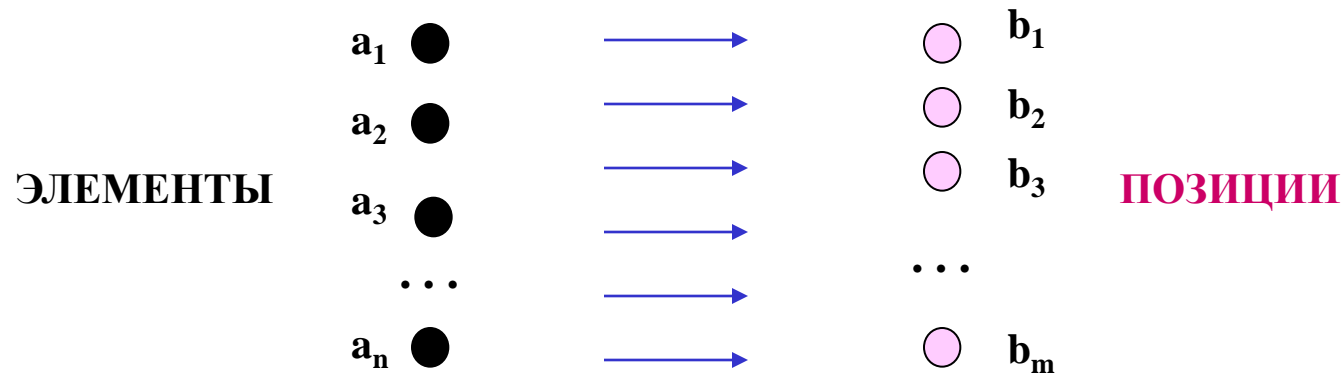
Двухдольный граф



1. Мальчики -- *Девочки* (задача о женитьбах)
2. Рабочие – *Рабочие позиции*
3. Оборудование – *Позиции в производственных системах*
(размещение оборудования)
4. Задания -- *Процессоры в многопроцессорных системах*
5. Анти-ракеты – *Цели в оборонных системах*
6. Массивы -- *Базы данных в распределенных информационных системах*

Др..

Задача о назначении (Assignment problem AP)



ФОРМУЛИРОВКА (комбинаторная):

Элементы (персонал, оборудование, задания): $A = \{ a_1, \dots, a_i, \dots, a_n \}$

Позиции (позиции, процессоры и др.) $B = \{ b_1, \dots, b_j, \dots, b_m \}$

(здесь $n = m$)

Эффективность пары a_i и b_j : $c(a_i, b_j)$

$\Pi = \{s\}$ - множество перестановок (назначений) элементов A

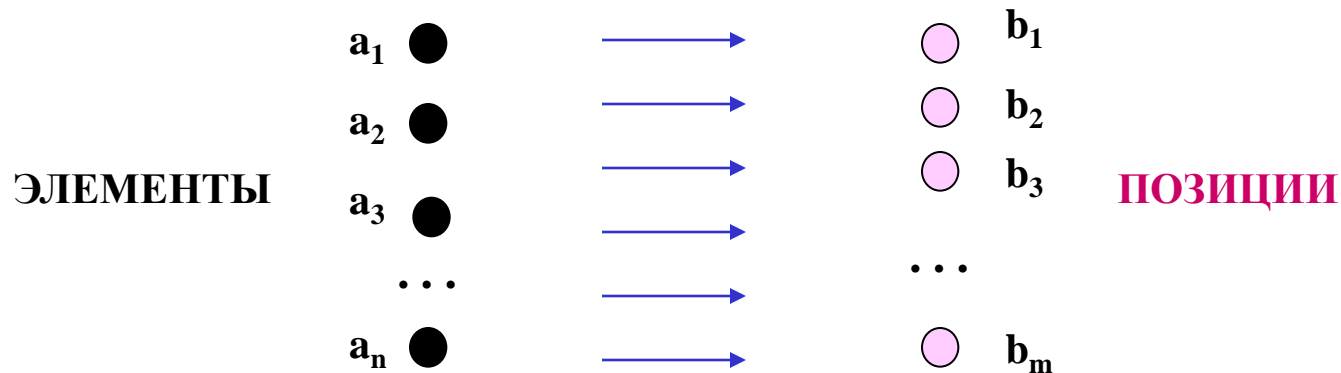
на позиции из множества B :

$s^* = \langle (s^*[1]), \dots, (s^*[i]), \dots, (s^*[n]) \rangle$, т.е., элемент a_i на позиции $s[i]$

Целевая функция: $\max \sum_{i=1}^n c(i, s[i])$

Максимум (минимум) сопоставление с весами в взвешенном двухдольном графе

Задача о назначении (Assignment problem AP)



ДРУГАЯ ФОРМУЛИРОВКА (алгебраическая):

Элементы (персонал, оборудование, задания): $A = \{ a_1, \dots, a_i, \dots, a_n \}$

Позиции (позиции, процессоры и др.) $B = \{ b_1, \dots, b_j, \dots, b_m \}$

(здесь $n = m$)

Эффективность пары a_i и b_j is: $c(a_i, b_j)$

$x_{ij} = 1$ если a_i размещено на позиции b_j и 0 в противном случае ($x_{ij} \in \{ 0, 1 \}$)

Задача: $\max \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$

s.t. $\sum_{i=1}^n x_{ij} = 1 \quad \forall j$
 $\sum_{j=1}^n x_{ij} = 1 \quad \forall i$

Задача о назначении (Assignment problem AP)

АЛГОРИТМЫ:

1. Полиномиальный точный алгоритм ($O(n^3)$)

2. Венгерский метод

Др.

ДРУГИЕ ВЕРСИИ:

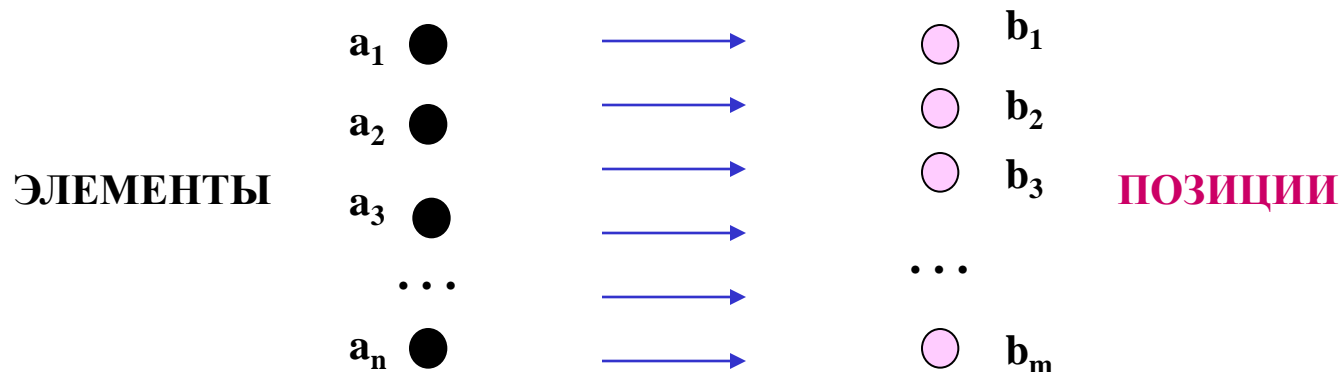
1. "*Minimum*" задача

2. "*Min max*" задача

3. Многокритериальная задача

Др.

Квадратичная задача о назначении (Quadratic Assignment Problem QAP)



матрица весов (“поток”) c_{ij}

	b_1	...	b_j	...	b_n
a_1	.		.		.
...	.		.		.
a_i	.		c_{ij}		.
...	.		.		.
a_n	.		.		.

матрица расстояний d_{ij}

	b_1	...	b_j	...	b_n
b_1	.		.		.
...	.		.		.
b_i	.		d_{ij}		.
...	.		.		.
b_n	.		.		.

Базовая (“потокковая”) Формулировка Квадратичной Задачи о Назначении

Назначение элементов $\{ i \mid i \in \{1, \dots, n\} \}$ на позиции $\{ p(i) \}$,
где p - перестановка номеров $\{1, \dots, n\}$,

Множество всех возможных перестановок - $\Pi = \{ p \}$.

Рассматриваются две матрицы n на n :

(i) матрица “потока” (или “полезности”) C , где элемент (i, j)
представляет поток между элементами (например, оборудованием) i и j ,

(ii) матрица расстояния D , где элемент (i, j) ($p(i), p(j)$) представляет
Расстояние между позициями $p(i)$ и $p(j)$.

Таким образом, задача QAP записывается так:

$$\max_{p \in \Pi} \sum_{i=1}^n \sum_{j=1}^n c_{ij} d_{p(i)p(j)}$$

АЛГОРИТМЫ:

- 1.Метод ветвей и границ**
- 2.Методы релаксации (сведения к непрерывным моделям)
(Relaxation approach)**
- 3.Жадные алгоритмы**
- 4.Генетические алгоритмы (эволюционные вычисления)**
- 5.Мета-эвристики (локальная оптимизация, гибридные схемы и др.)**

БАЗОВЫЕ КНИГИ:

- 1.P.M. Pardalos, H. Wolkowicz, (Ed.), Quadratic Assignment and Related Problems. American Mathematical Society, 1994.**
- 2.E. Cela, The Quadratic Assignment Problem. Kluwer, 1998.**

САЙТЫ:

- 1.Quadratic assignment Problem Library: <http://www.opt.math.tu-graz.ac.at/qaplib/>**

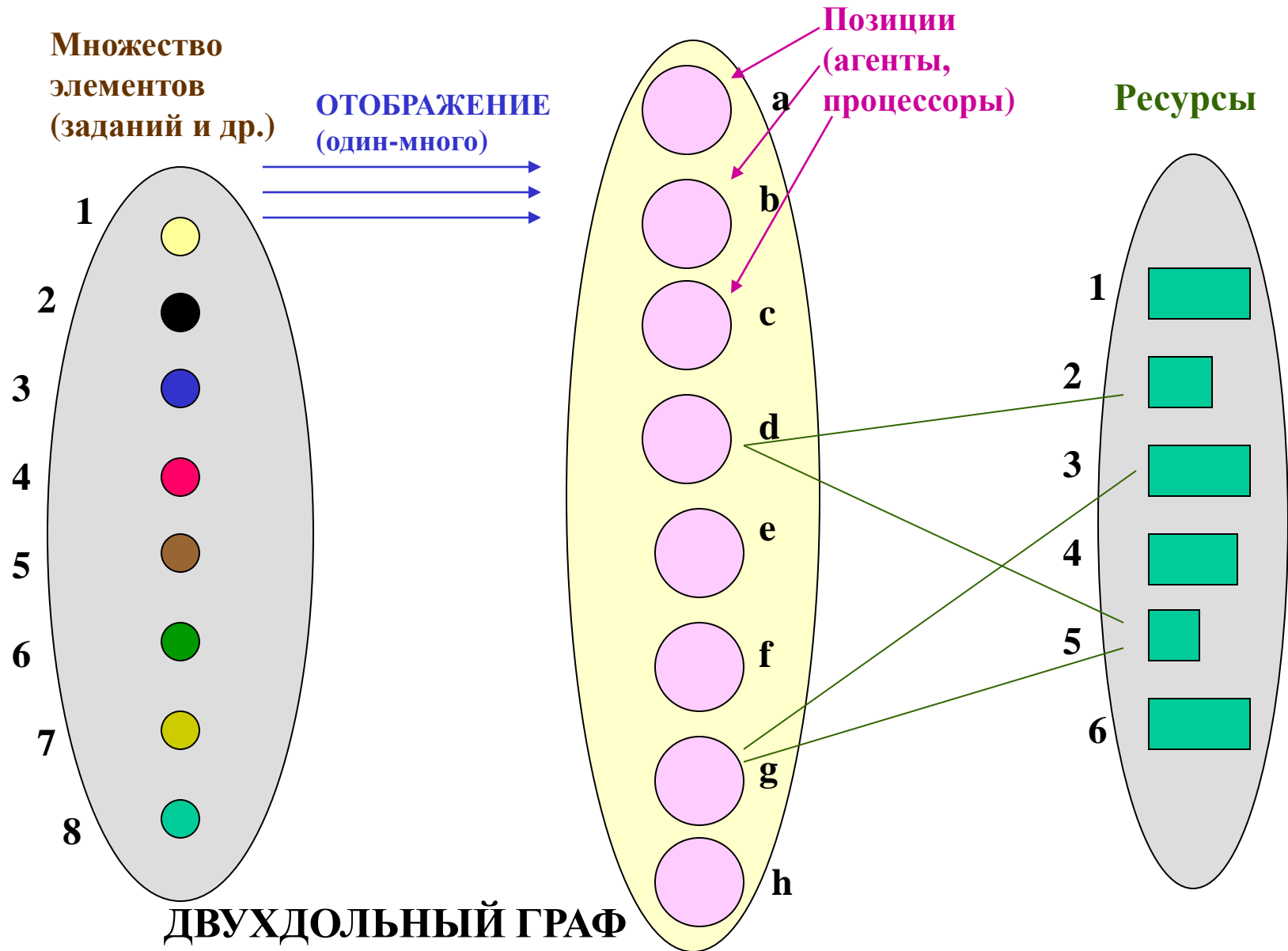
**Multi-objective (multicriteria) Assignment Problem
(Quadratic Assignment Problem):**

Элементы в матрице весов (“потoki”): вектора

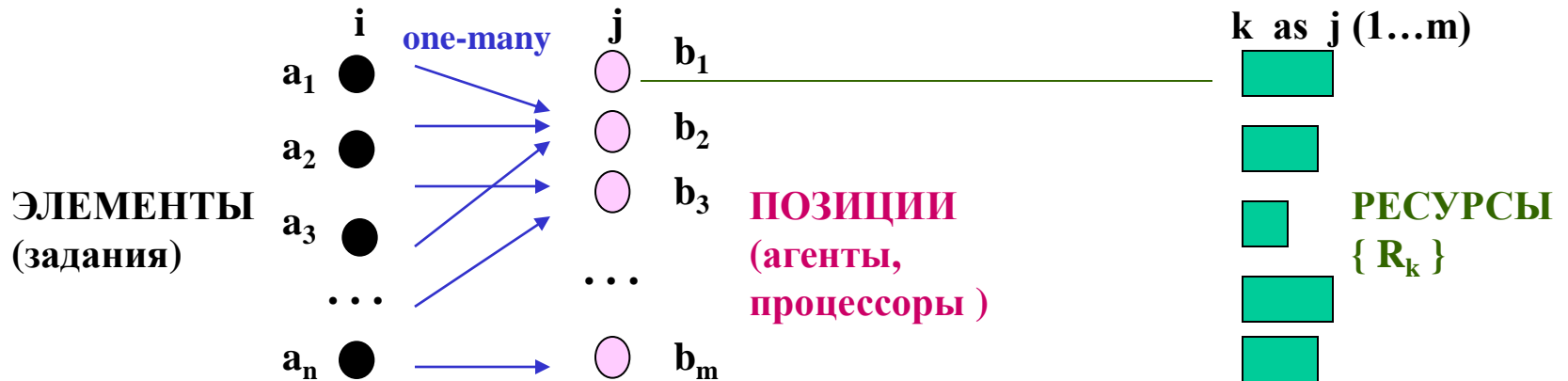
АЛГОРИТМЫ:

- 1.Метод ветвей и границ**
- 2.Метод релаксации (Relaxation approach)**
- 3.Жадные алгоритмы**
- 4.Мета-эвристики (локальная оптимизация, гибридные схемы и др.)**
- 5.Многокритериальная эволюционная оптимизация
(Multi-objective evolutionary optimization)**

Обобщенная задача о назначениях (Generalized Assignment Problem GAP)



Обобщенная задача о назначении (Generalized Assignment Problem GAP)



ФОРМУЛИРОВКА (алгебраическая):

Элементы (персонал, оборудование, задания): $A = \{ a_1, \dots, a_i, \dots, a_n \}$

Позиции (позиции, процессоры и др.) $B = \{ b_1, \dots, b_j, \dots, b_m \}$

(здесь $n = m$)

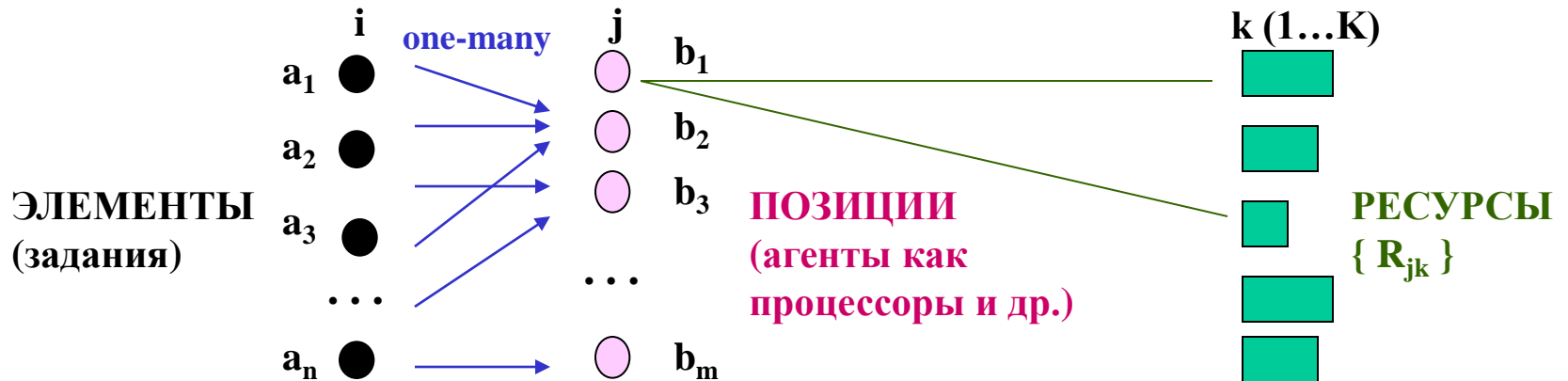
Эффективность пары a_i и b_j : $c(a_i, b_j)$

$x_{ij} = 1$ если a_i размещено на позиции b_j и 0 в противном случае ($x_{ij} \in \{0, 1\}$)

Задача: $\max \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$

s.t. $\sum_{i=1}^n r_{ik} x_{ij} \leq R_j \quad \forall j$ (R_k - это ресурс агента k)
 $\sum_{j=1}^n x_{ij} = 1 \quad \forall i$

Обобщенная задача о назначении (GAP) (много общих ресурсов)



ФОРМУЛИРОВКА (алгебраическая):

Элементы (персонал, оборудование, задания): $A = \{ a_1, \dots, a_i, \dots, a_n \}$

Позиции (позиции, процессоры и др.) $B = \{ b_1, \dots, b_j, \dots, b_m \}$

(здесь $n = m$)

Эффективность пары a_i и b_j : $c(a_i, b_j)$

$x_{ij} = 1$ если a_i размещено на позиции b_j и 0 в противном случае ($x_{ij} \in \{ 0, 1 \}$)

Задача: $\max \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$

s.t. $\sum_{j=1}^m \sum_{i=1}^n r_{ik} x_{ij} \leq R_k \quad \forall k \text{ (K общих ресурсов)}$
 $\sum_{j=1}^m x_{ij} = 1 \quad \forall i$

АЛГОРИТМЫ:

- 1.Метод ветвей и границ**
- 2.Метод релаксации**
- 3.Эвристики (жадные алгоритмы и др.)**
- 4.Генетические алгоритмы (эволюционные вычисления)**
- 5.Мета-эвристики (локальная оптимизация, гибридные схемы и др.)**

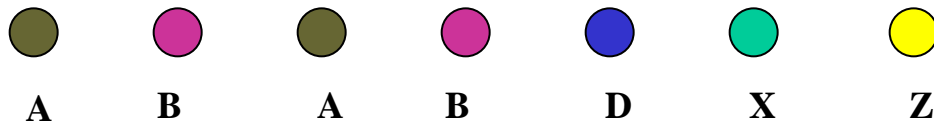
РАСШИРЕНИЯ:

- 1.Многокритериальные случаи**
- 2.Неопределенность**
- 3.Др. (зависимость между частями задачи, Зависимость от времени и др.)**

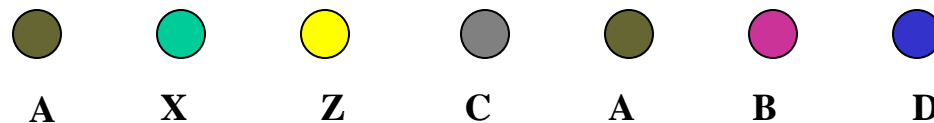
Сопоставление строк (String matching)

СЛУЧАЙ 2-Х СЛОВ:

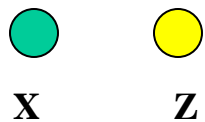
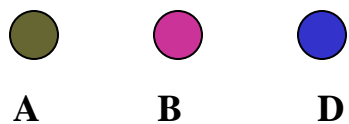
Строка (слово) 1



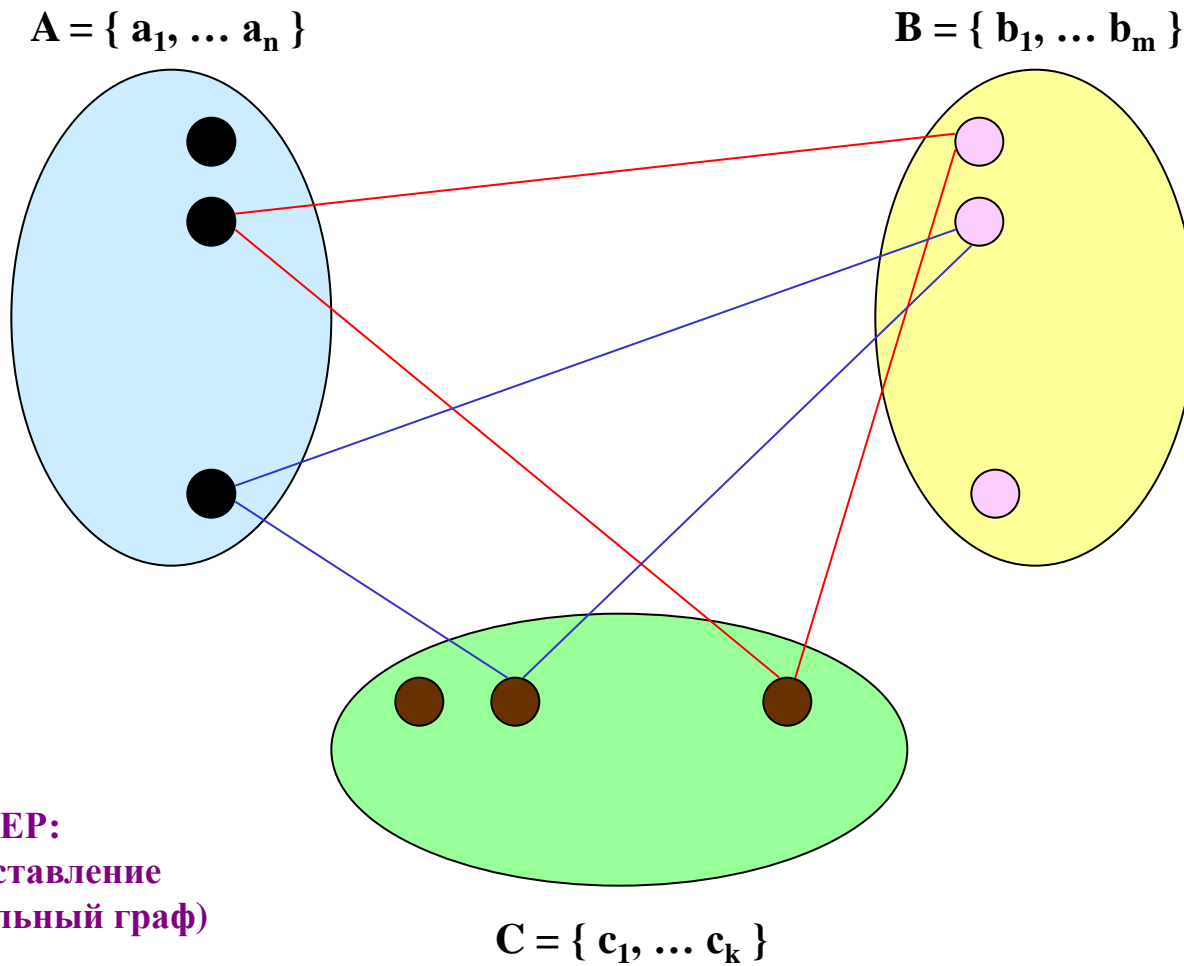
Строка (слово) 2



**Образы
(PATTERNS):**



Задача сопоставления на k-дольном графе (Multiple matching problem, Lecture 17-18)



ПРИМЕР:
3-сопоставление
(3-х дольный граф)

АЛГОРИТМЫ:

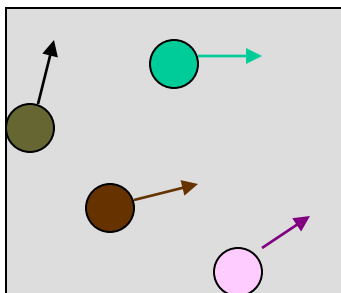
- 1.Переборные алгоритмы (метод ветвей и границ и др.)**
- 2.Эвристики (e.g., жадные алгоритмы, различные методы локальной оптимизации и др.)**
- 3.Мета-эвристики включая гибридные схемы**
- 4.Морфологический подход**

ВЕРСИИ:

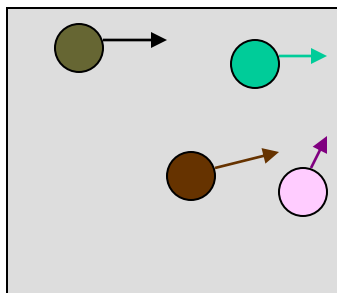
- 1.Динамическая (много-стадийная) задача (сопоставление траекторий / трасс целей)**
 - 2.Задача с ошибками в данных**
 - 3.Задача с неопределенностью (вероятностные оценки, размытые множества)**
- Др.**

Использование задачи о назначении для определения скоростей частиц (Lecture 17-18)

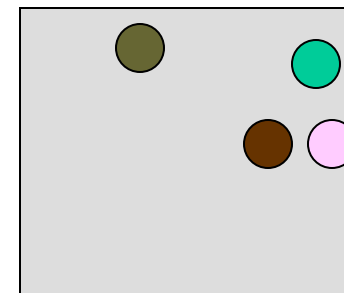
КАДР 1



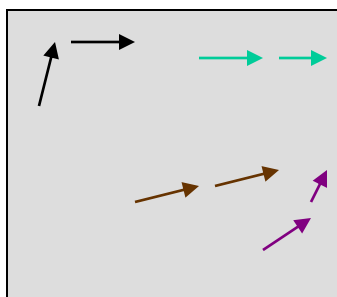
КАДР 2



КАДР 3



ПРОСТРАНСТВО СКОРОСТЕЙ



ПРИЛОЖЕНИЯ (среда газа/жидкости):

1. Физические эксперименты
2. Наука о климате (анализ облаков и др.)
3. Химические процессы
4. Биотехнология

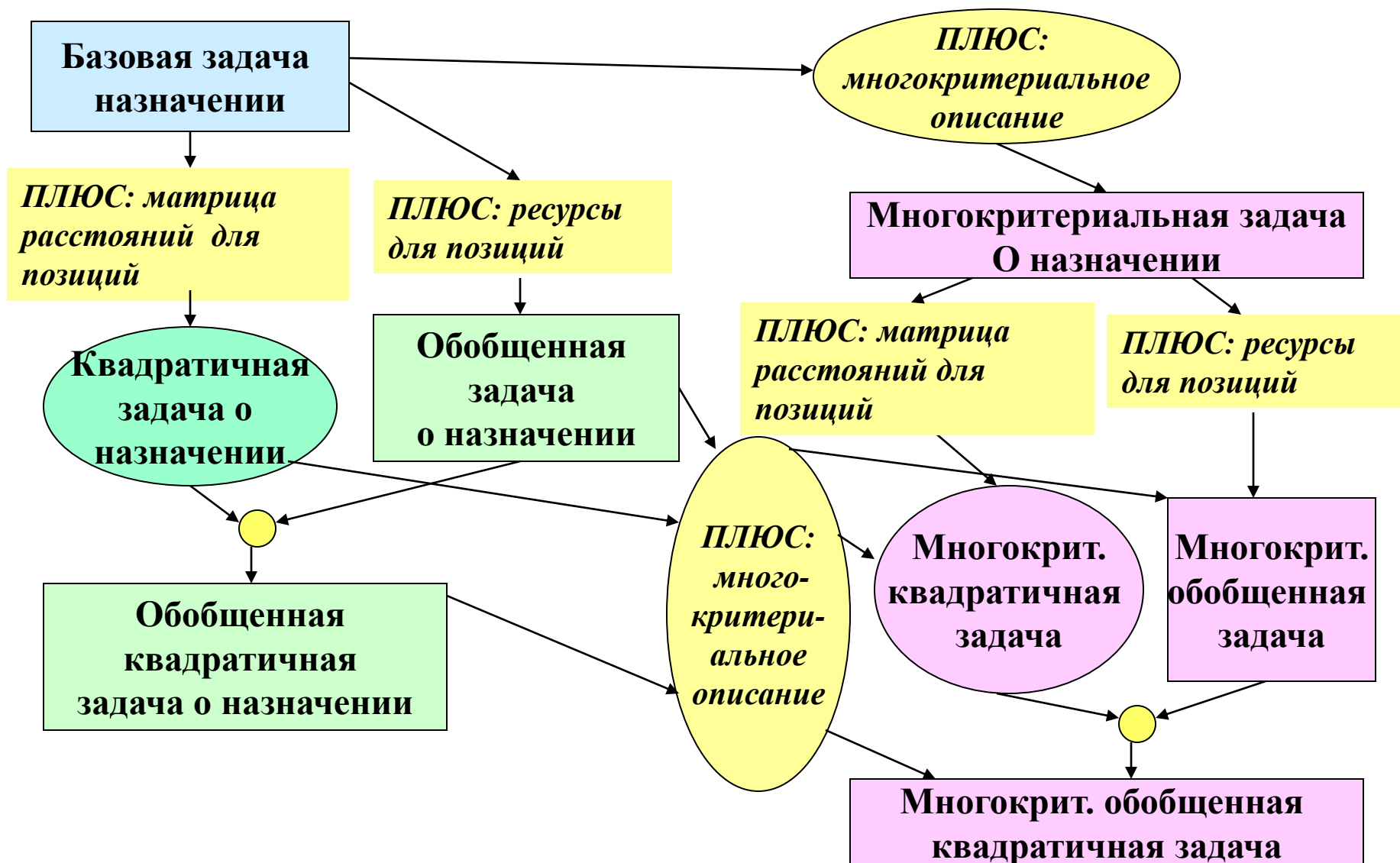
Источники:

1. PIV системы
2. Фотографии со спутников
3. Электронные микроскопы и др.

Список алгоритмов для задач типа назначения / размещения

1. полиномиальные алгоритмы (венгерский метод, методы на основе потоковых подходов и др.)
2. Методы на основе линейного программирования
3. epsilon-приближенные полиномиальные методы
4. корректирующие алгоритмы
5. Метод ветвей и границ
6. метод динамического программирования
7. эволюционные и генетические алгоритмы
8. Схема декомпозиции Бендерса
9. Методы локальной оптимизации как различные эвристики (Tabu-поиск, метод отжига, гибридные схемы и др.)
10. Методы многокритериального анализа (функция полезности, Метод Аналитических , методы порогов сравнимости)
11. методы на основе имитационного моделирования
12. полихедральные методы (polyhedral methods)
13. Иерархические подходы включая морфологический подход
14. метод на основе задачи о выполнимости ограничений ("constraint satisfaction problem«)
15. подходы на основе размытых множеств
16. алгоритмические подходы на основе баз знаний
20. подходы на основе нейронных сетей

Блок-диаграмма задач типа размещения



Прикладные области для задач типа размещения

1. размещение оборудования (layout) в производственных системах
2. размещение ресурсов в инвестиционных мероприятиях
3. размещение оборудования в управлении сетью поставок (supply chain management)
4. размещение задач маршрутизации
5. размещение при проектировании больших интегральных схем (layout in VLSI design)
6. размещение при пространственном проектировании зданий
7. размещение в архитектурном планировании
8. размещение служб быстрого реагирования в чрезвычайных ситуациях
9. размещение буферов накопления в производственных системах
10. размещение операций техобслуживания
11. размещение операций контроля (тестирования)
12. размещение операций балансировки в сборочных линиях
13. размещение (распределение) человеко-машинных функций
14. формирование (оптимизация) портфеля финансовых инструментов (акций и др.)
15. назначение проектных задач

Прикладные области для задач типа размещения

16. распределение ресурсов в больших коллективах
17. размещение ресурсов дорожной полиции
18. распределение крови в донорской системе
19. распределение статей между рецензентами
20. назначение в спорте
21. динамическое распределение памяти в вычислительных системах
22. размещение заданий в распределенных информационных / компьютерных системах
23. размещение задач в виртуальных организациях
24. размещение информации (документы, массивы) в информационных системах
25. распределение пропускных способностей в коммуникационных сетях
26. размещение ресурсов в Интернет
27. размещение ресурсов в инфраструктурах электронного бизнеса
(e-business infrastructure)
28. проектирование стандартов
29. сопоставление траекторий (трасс) целей (в много-радарных системах)
30. размещение надежности между компонентами системы
31. задачи назначения в экспериментальной физике высоких энергий
32. locomotive assignment to train-segments

Прикладные области для задач типа размещения

- 33.задачи назначения / размещения в университетах (назначение студентов на экзамены, назначение преподавателей на курсы, назначение аудиторий для лекций, и др.)**
- 34.назначение частиц / точек (для сопоставление частиц) в системах particle image velocimetry (PIV) Для измерений в механике потоков**
- 35.управление персоналом (размещение / назначение персонала, размещение заданий, распределение ролей)**
- 36.распределение комнат между людьми**
- 37.назначение прав (в социальных сетях (social networks for social choice and welfare), для участников электронного финансового рынка (electronic financial markets))**
- 38.размещение дискретных ресурсов**
- 39.назначение частот / каналов в мобильных радио системах**
- 40.назначение сот (cells) к коммутаторам (switches) в сотовых мобильных сетях (cellular mobile networks)**
- 41.размещение допусков в производственных системах**
- 42.размещение wavelength на деревьях колец (rings) в оптических коммуникационных сетях**

43. размещение в медицинских организациях:

- (1) Размещение хирургов по операционным, (2) allocation of inpatient resources,**
- (3) размещение персонала, (4) размещение оборудование в госпитале,**
- (5) размещение кроватей, (6) размещение органов для трансплантации,**
- (7) размещение пациентов и др.**

44. размещение узких мест в производственных системах

45. размещение трафика в транспортных и коммуникационных/компьютерных сетях

46. иерархическое размещение задач в сетях

47. динамическое размещение ресурсов в много-проектных исследования

48. размещение экологического оборудования

49. иерархическое размещение банковского оборудования

BASIC BOOKS:

- 1.M.S. Daskin, Networks and Discrete Location. Models, Algorithms, and Applications. Wiley, 1995.**
- 2.G.Y. Handler, P.B. Mirchandrani, Location on Networks: Theory and Algorithms. MIT Press, 1979.**
- 3.E. Minieka, Optimization Algorithms for Networks and Graphs. Marcel Dekker, 1978.**
- 4.P.B. Mirchandrani, R.L. Francis, (Eds.), Discrete Location Theory, Wiley, 1990.**
- 5.P.M. Pardalos, H. Wolkowicz, (Ed.), Quadratic Assignment and Related Problems. American Mathematical Society, 1994.**
- 6.E. Cela, The Quadratic Assignment Problem. Kluwer, 1998.**
- 7.M.I. Rubinshtein, Optimal Grouping of Interconnected Objects, Nauka, (in Russian), 1989.**
- 8.D. Gusfield, R.W. Irwing, The Stable Marriage Problem: Structure and Algorithms, The MIT Press, 1989.**
- 9.J. Aoe, (Ed.), Computer Algorithms: String Pattern Matching Strategies. IEEE CS Press, 1994.**
- 10.A.I. Barros, Discrete and Fractional Programming Techniques for Location Models. Kluwer, 1998.**

SITES:

- 1.Dictionary of Algorithms and Data Structure (NIST): <http://www.nist.gov/dads/>**
- 2.OR-Library by J.E. Beasley: <http://www.brunel.ac.uk/depts/research/jeb/info.html>**
- 3.Quadratic Assignment Problem Library: <http://www.opt.math.tu-graz.ac.at/qaplib/>**

ЛЕКЦИЯ 31. Курс: “Проектирование систем: Структурный подход”

Каф. “Коммуникационные сети и системы”, Факультет радиотехники и кибернетики

Московский физико-технический институт (университет)

Марк Ш. Левин

Институт проблем передачи информации, РАН

Email: mslevin@acm.org / mslevin@iitp.ru

Л.31. задача выполнимости. 6 базовых «трудных» задач. Timetabling.

PLAN:

1. Задача выполнимости: *формулировка и иллюстрации, *модификация, *приложения
 2. Базовые задачи комбинаторной оптимизации (выполнимость, 3-выполнимость, вершинное покрытие, 3-сопоставление (matching), клика, Гамильтонов цикл, разбиение)
 3. Задачи составления календарных планов (Timetabling problems)
- *формулировка и иллюстрации, *приложения, *алгоритмы и схемы решения, *библиография (статьи, сайты)

Дек. 10, 2004

Задача выполнимости

БУЛЕВЫ ПЕРЕМЕННЫЕ:

$U = x_1, x_2, x_3, \dots$ (0 or 1)

ЛОГИЧЕСКИЕ ОПЕРАЦИИ (AND, OR, NOT):

*ОТРИЦАНИЕ (НЕТ) NOT *ПЕРЕСЕЧЕНИЕ (и) conjunction AND,
*СКОБКИ ДЛЯ ГРУППИРОВКИ

МНОЖЕСТВО ЛОГИЧЕСКИХ ВЫРАЖЕНИЙ C:

Примеры: $C_1 = (\text{NOT } x_1) \text{ OR } x_2$

$C_2 = x_1 \text{ OR } (\text{NOT } x_3)$

$C_3 = (\text{NOT } x_5) \text{ OR } (\text{NOT } x_7) \text{ OR } x_9$

$C_4 = x_1 \text{ OR } x_{10} \text{ OR } (\text{NOT } x_{11}) \text{ OR } x_{22}$

**ВСЕ TRUE (1)
(ВЕРНЫ)**

Задача выполнимости является базовой NP-полной проблемой (задачей)

Применима в следующих областях:

- *выполнение ограничений,**
- *логика,**
- *автоматическое доказательство теорем**

Кроме того,

эта задача важна с точки зрения теории как «корневая» проблема для доказательства NP-полноты для других задач

ПРИЛОЖЕНИЯ:

1.Верификация программ (Software Verification)

**2.Проектирование и верификация в электронике
(Electronic Design Automation and Verification)**

3.Анализ моделей (Model Analysis)

4.Проверка моделей (Model Checking)

5.Доказательство теорем (Theorem Prover)

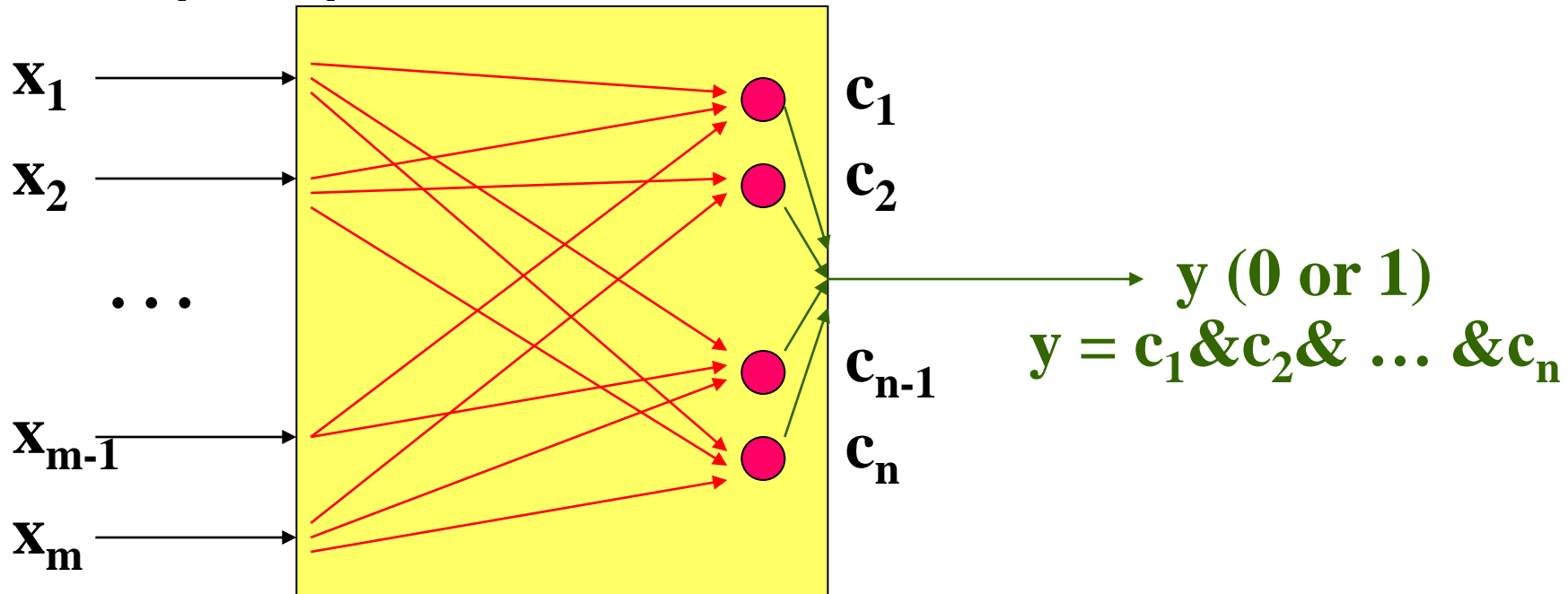
6.Планирование в искусственном интеллекте (AI Planning)

Задача выполнимости: иллюстрация для применения в программировании и электронике

ЗАДАЧА: Существует ли $x_0=(x_1, \dots, x_n)$ такой что $y(x_0) = 1$
ИЛИ нет

Литерал: x_i / $\text{not } x_i$

СИСТЕМА



Пример:

$$c_1 = \text{not } x_1 \text{ OR } x_2 \text{ OR } x_4 \text{ OR } \text{not } x_5 \text{ OR } x_7$$

$$c_2 = x_1 \text{ OR } \text{not } x_2 \text{ OR } \text{not } x_3 \text{ OR } x_5 \text{ OR } x_7$$

$$c_3 = \text{not } x_1 \text{ OR } \text{not } x_2 \text{ OR } x_3 \text{ OR } \text{not } x_5 \text{ OR } \text{not } x_n$$

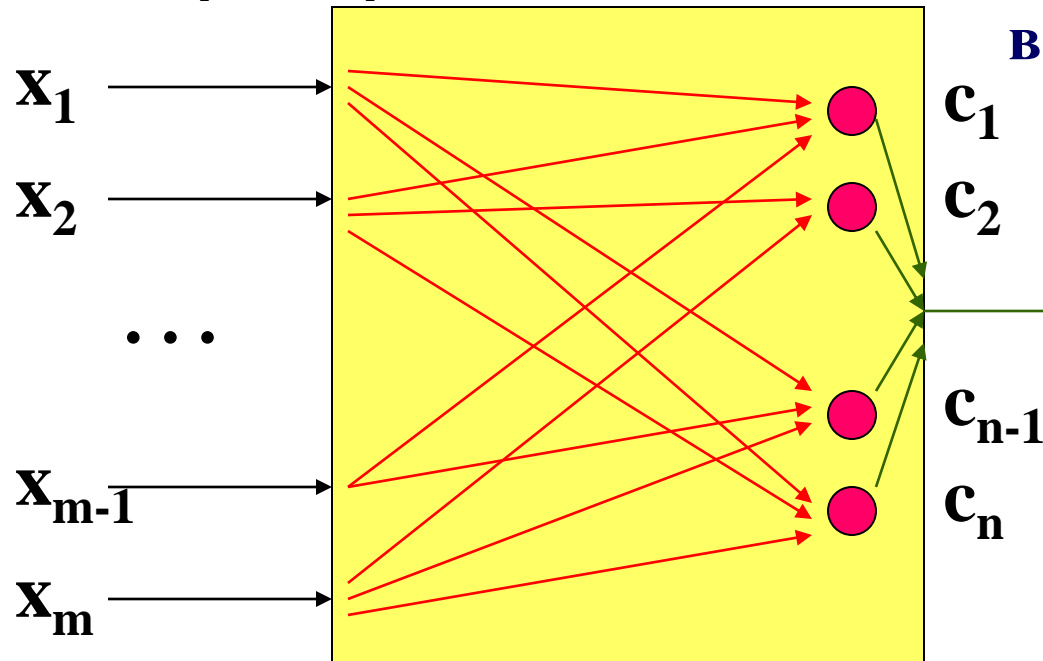
$$c_4 = \text{not } x_2 \text{ OR } x_3 \text{ OR } x_7 \text{ OR } x_{n-2} \text{ OR } x_{n-1}$$

...

Задача 3-выполнимости

Литерал: x_i / $\text{not } x_i$

СИСТЕМА



Задача 3-выполнимости:
в каждом c_j - 3 литерала

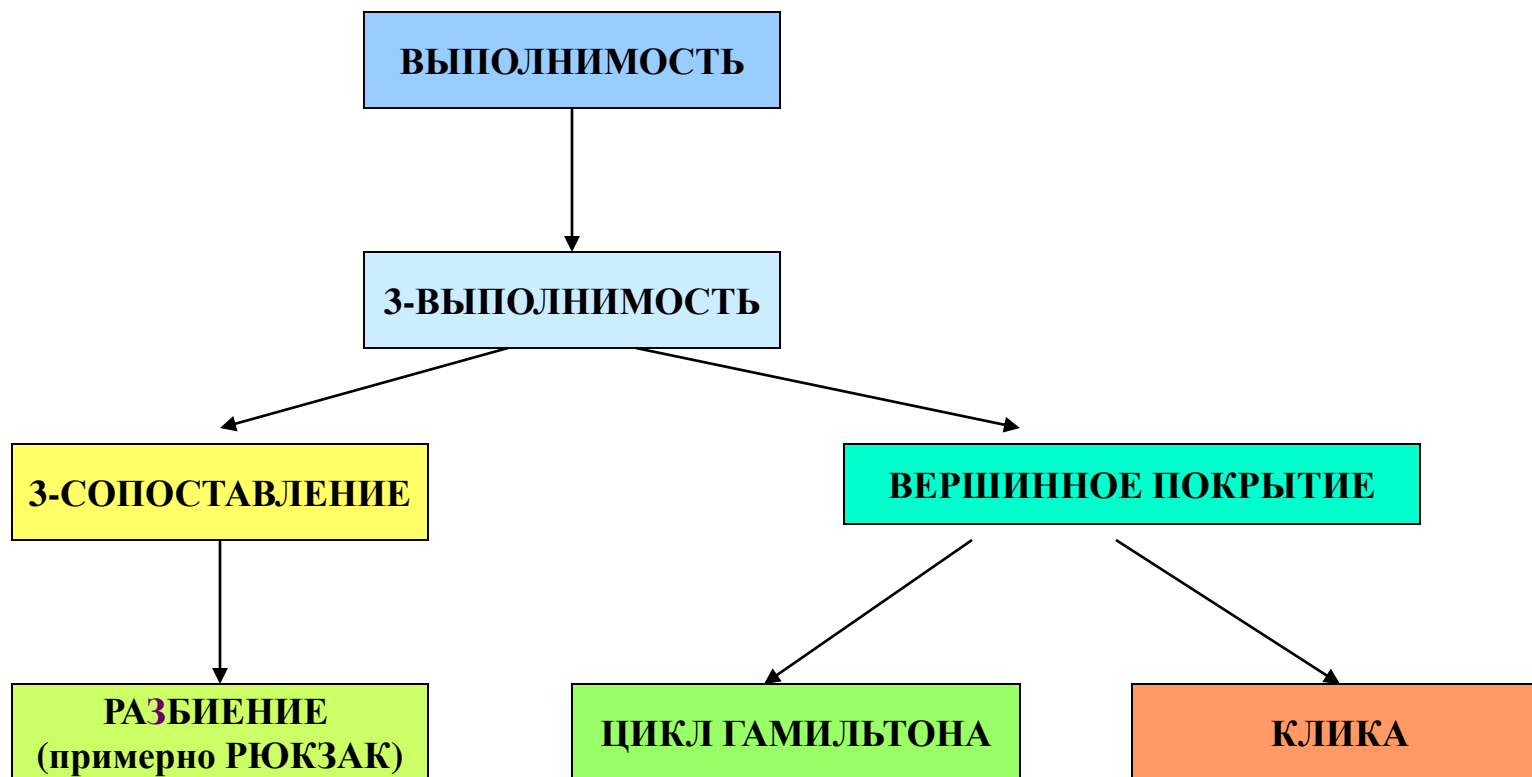
y (0 or 1)

$$y = c_1 \& c_2 \& \dots \& c_n$$

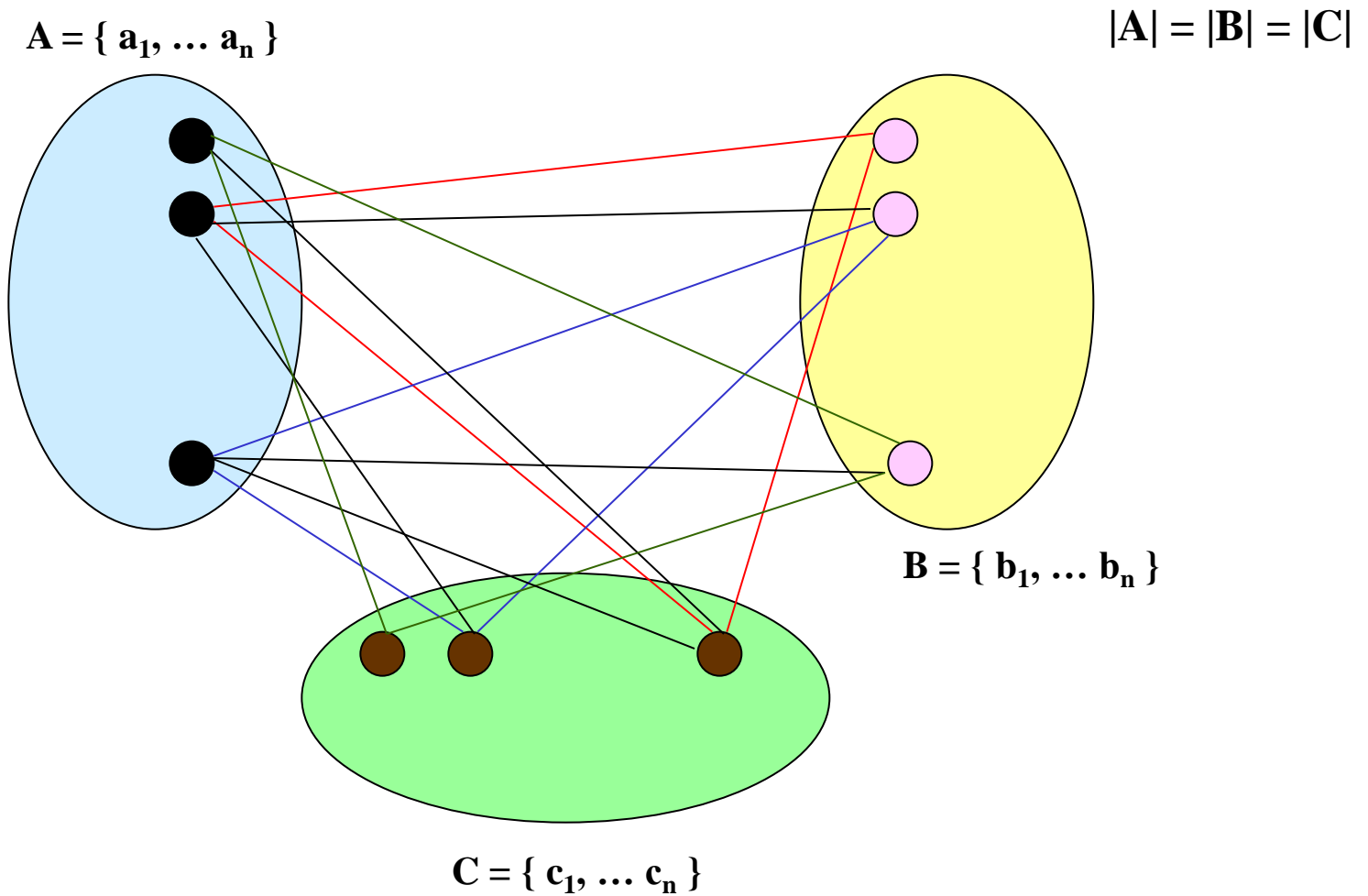
Задача:

$$\begin{aligned} c_1 &= \text{not } x_1 \text{ OR } x_2 \text{ OR } x_4 \\ c_2 &= x_2 \text{ OR } \text{not } x_3 \text{ OR } x_7 \\ c_3 &= \text{not } x_1 \text{ OR } \text{not } x_5 \text{ OR } \text{not } x_n \\ c_4 &= \text{not } x_2 \text{ OR } x_{n-2} \text{ OR } x_{n-1} \\ &\dots \end{aligned}$$

6 БАЗОВЫХ NP-ПОЛНЫХ ЗАДАЧ



3-СОПОСТАВЛЕНИЕ



ЗАДАЧА: Существует ли покрытие вершин тройками (без пересечений)
ИЛИ нет

РАЗБИЕНИЕ

**Множество элементов $A = \{ 1, \dots, i, \dots, n \}$
“веса” элементов $\{ b_1, \dots, b_i, \dots, b_n \}$**

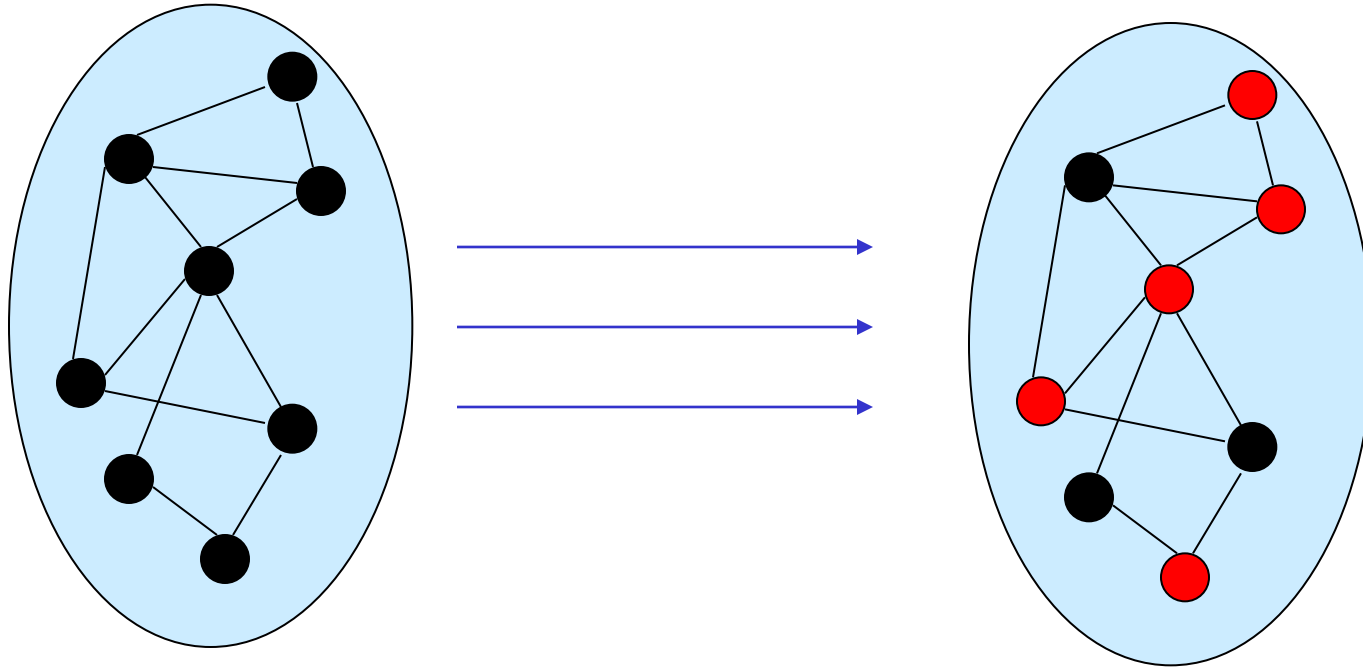
РАЗБИЕНИЕ: Существует ли $A' \subset A$ такое что

$$\sum_{i \in A'} b_i = \sum_{j \in A \setminus A'} b_j$$

ИЛИ нет

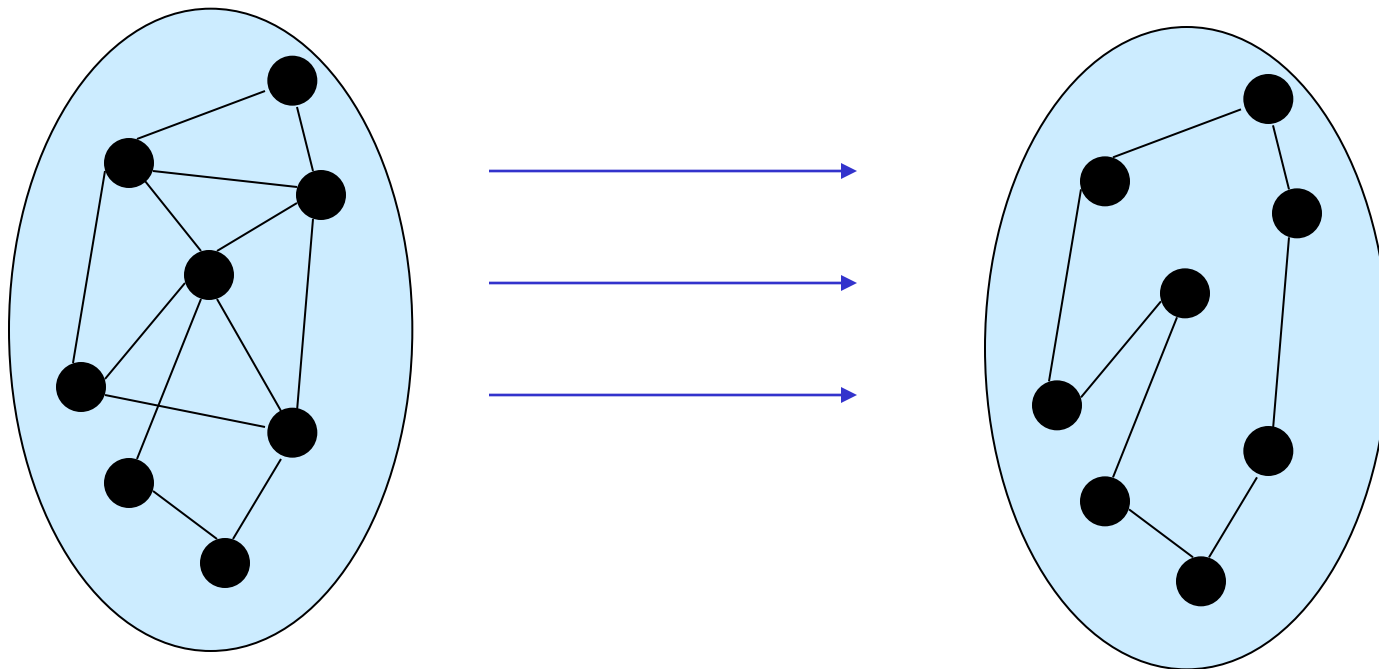
ВЕРШИНОЕ ПОКРЫТИЕ

Множество вершин $A = \{ a_1, \dots, a_n \}$, множество ребер $E = \{ e_1, \dots, e_k \}$, граф $G = (A, E)$



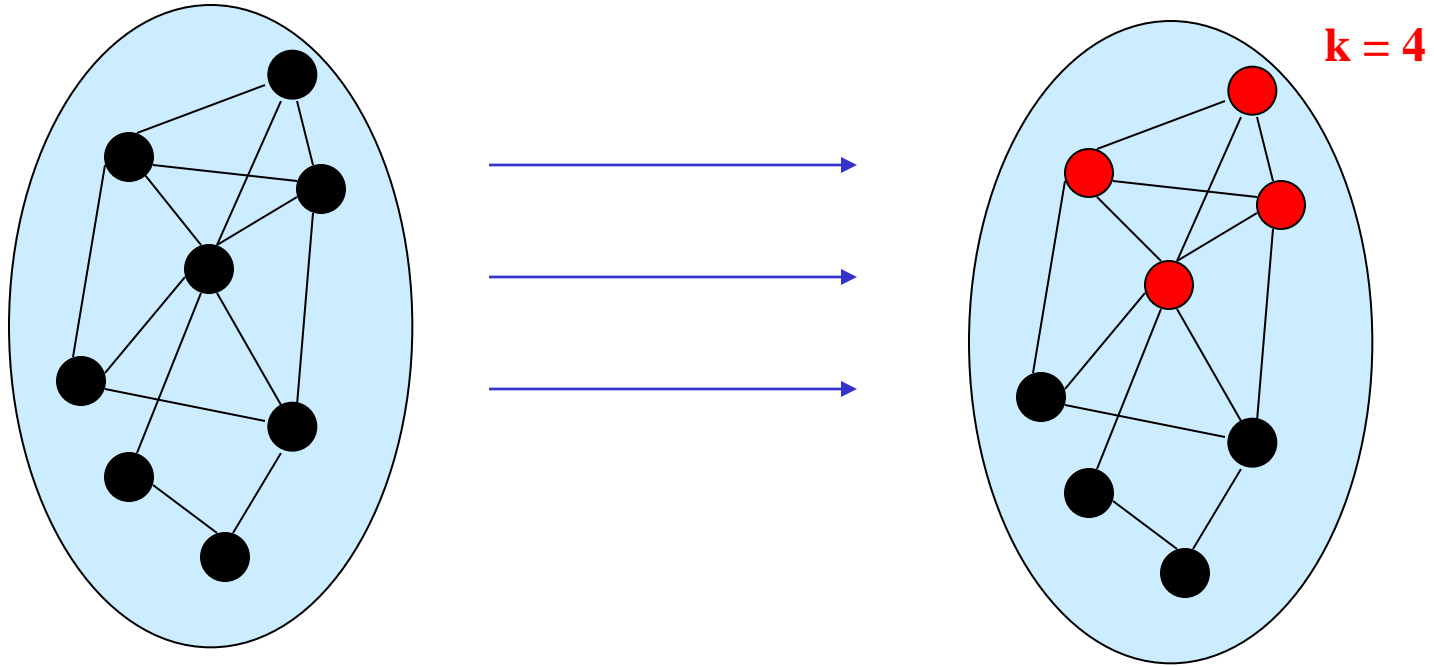
ЗАДАЧА: Найти вершинное покрытие ($A' \subseteq A$)
которое покрывает $\forall e \in E$

ЦИКЛ ГАМИЛЬТОНА



ЗАДАЧА: Существует ли цикл Гамильтона ИЛИ нет

КЛИКА



ЗАДАЧА: Существует ли подграф-клика с числом вершин k ИЛИ нет

БАЗОВЫЕ (& БЛИЗКИЕ) КОМБИНАТОРНЫЕ МОДЕЛИ:

1.Раскраска графа

2.Назначение/ Размещение / сопоставление

3.Задача о женитьбах

4.Задача теории расписаний

5.Циклические задачи теории расписаний

6.Клика

7.Планирование эксперимента (Combinatorial design, например, латинские квадраты - Latin square и до.)

Др.

АЛГОРИТМЫ И АЛГОРИТМИЧЕСКИЕ СХЕМЫ:

1. ПЕРЕБОРНЫЕ МЕТОДЫ

2. МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЕКТА:

экспертные системы (на основе правил – продукций и др.)

алгоритмы на основе баз знаний

нейронные сети

3. МЕТОДЫ НА ОСНОВЕ ПРОВЕРКИ ОГРАНИЧЕНИЙ (CONSTRAINED PROGRAMMING)

4. МЕТА-ЭВРИСТИКИ (различные методы локальной оптимизации):

оптимизация на основе биологической аналогии (ant colony optimization,)

итеративный локальный поиск (iterated local search)

метод отжига

Tabu поиск

генетические алгоритмы

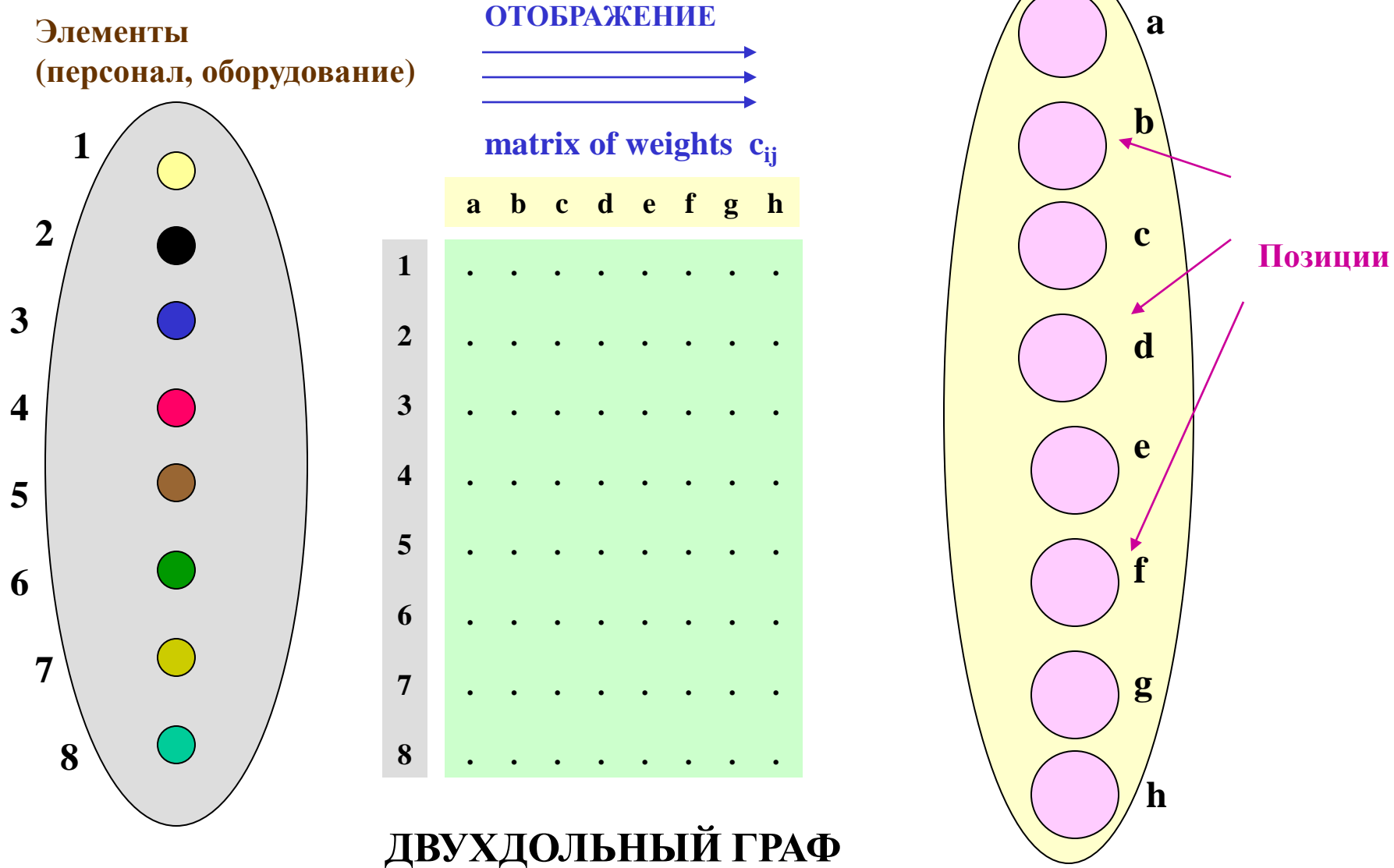
5. ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТА (COMBINATORIAL DESIGN)

6. ЭВОЛЮЦИОННЫЕ ВЫЧИСЛЕНИЯ

(включая multi-objective evolutionary optimization)

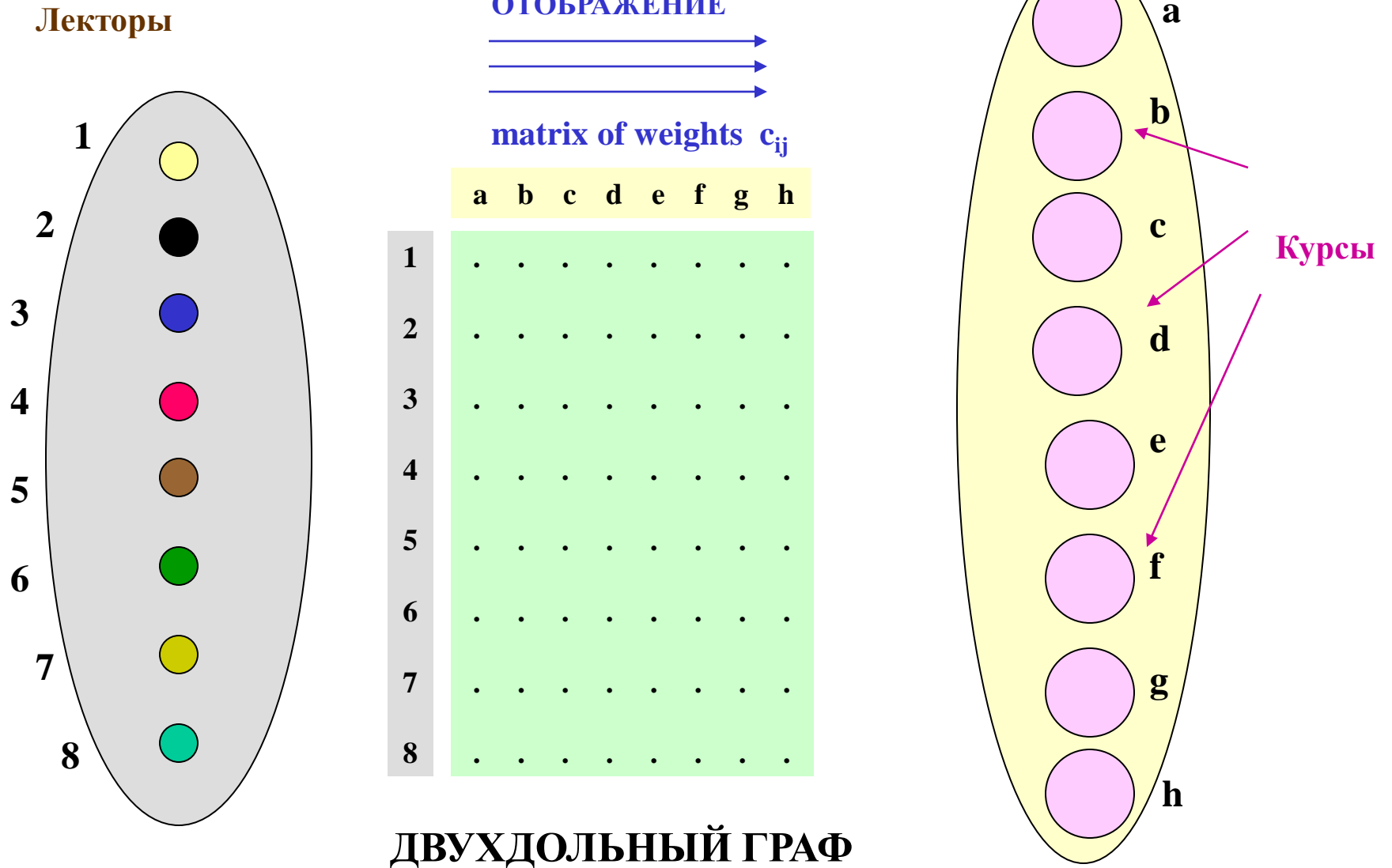
7. ГИБРИДНЫЕ СХЕМЫ РЕШЕНИЯ

Размещение (назначение, сопоставление, позиционирование):

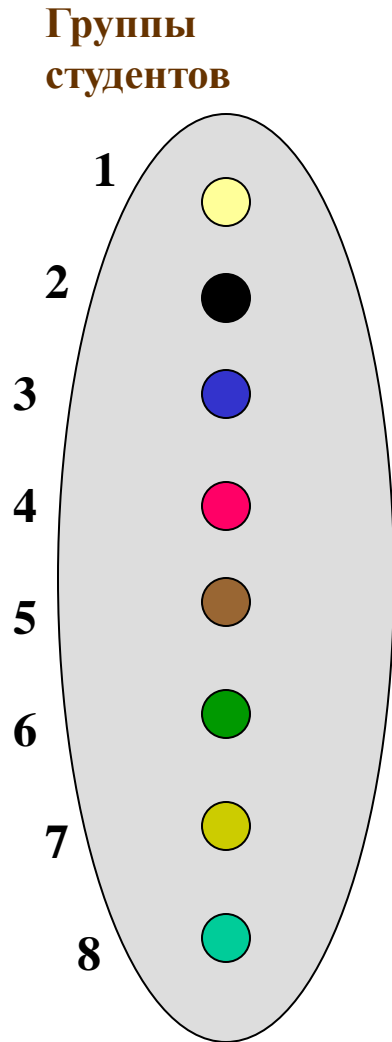


Задача размещения (Лекция 30)

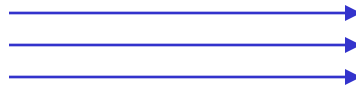
Размещение (назначение, сопоставление, позиционирование):



Размещение (назначение, сопоставление, позиционирование):



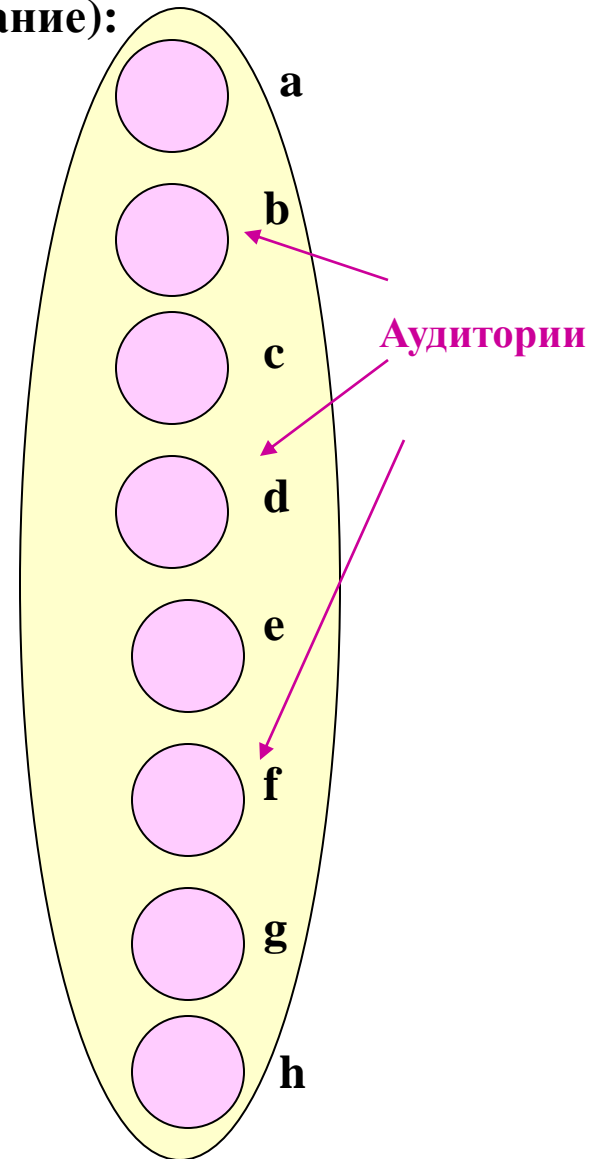
ОТОБРАЖЕНИЕ



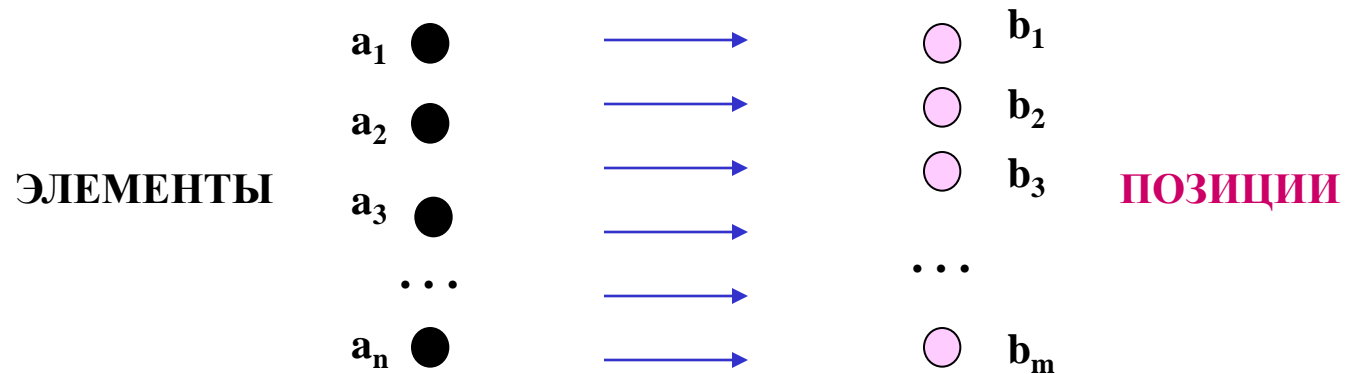
matrix of weights c_{ij}

	a	b	c	d	e	f	g	h
1
2
3
4
5
6
7
8

ДВУХДОЛЬНЫЙ ГРАФ



Задача о назначении AP (Лекция 30)



ФОРМУЛИРОВКА (алгебраическая):

Элементы (персонал, оборудование, задания): $A = \{ a_1, \dots, a_i, \dots, a_n \}$

Позиции (позиции, процессоры) $B = \{ b_1, \dots, b_j, \dots, b_m \}$

(здесь $n = m$)

Эффективность пары a_i и b_j is: $c(a_i, b_j)$

$x_{ij} = 1$ если a_i размещен на позиции b_j и 0 в противном случае ($x_{ij} \in \{0, 1\}$)

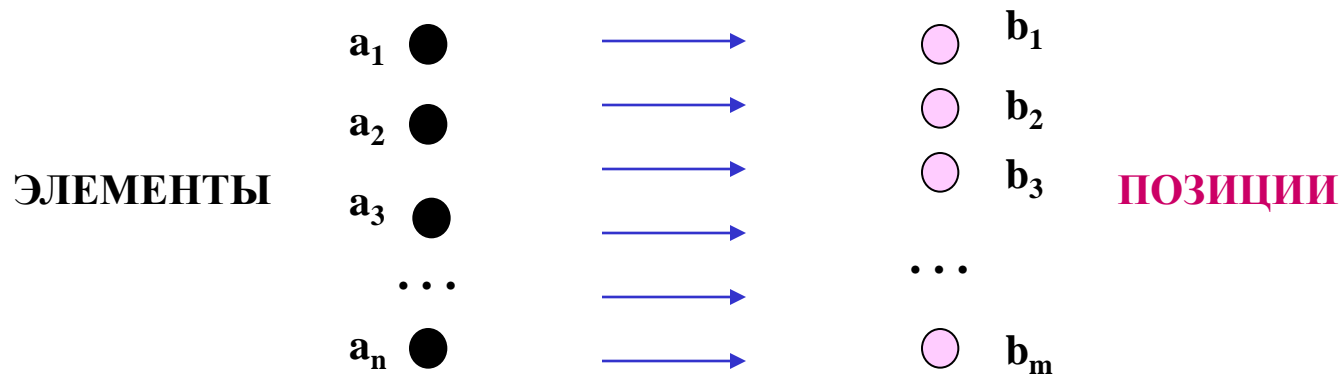
Задача: $\max \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$

s.t.

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i$$

Квадратичная задача о назначении (Quadratic Assignment Problem QAP) (Лекция 30)



матрица весов («потoki») c_{ij}

	b_1	...	b_j	...	b_n
a_1	.		.		.
...	.		.		.
a_i	.		c_{ij}		.
...	.		.		.
a_n	.		.		.

матрица «расстояний» d_{ij}

	b_1	...	b_j	...	b_n
b_1	.		.		.
...	.		.		.
b_i	.		d_{ij}		.
...	.		.		.
b_n	.		.		.

Базовая (“потоковая”) математическая формулировка

Назначение элементов $\{ i \mid i \in \{1, \dots, n\} \}$ на позиции $\{ p(i) \}$,
где p – перестановка номеров $\{1, \dots, n\}$,

Множество возможных перестановок: $\Pi = \{ p \}$.

Рассмотрим две n на n матрицы:

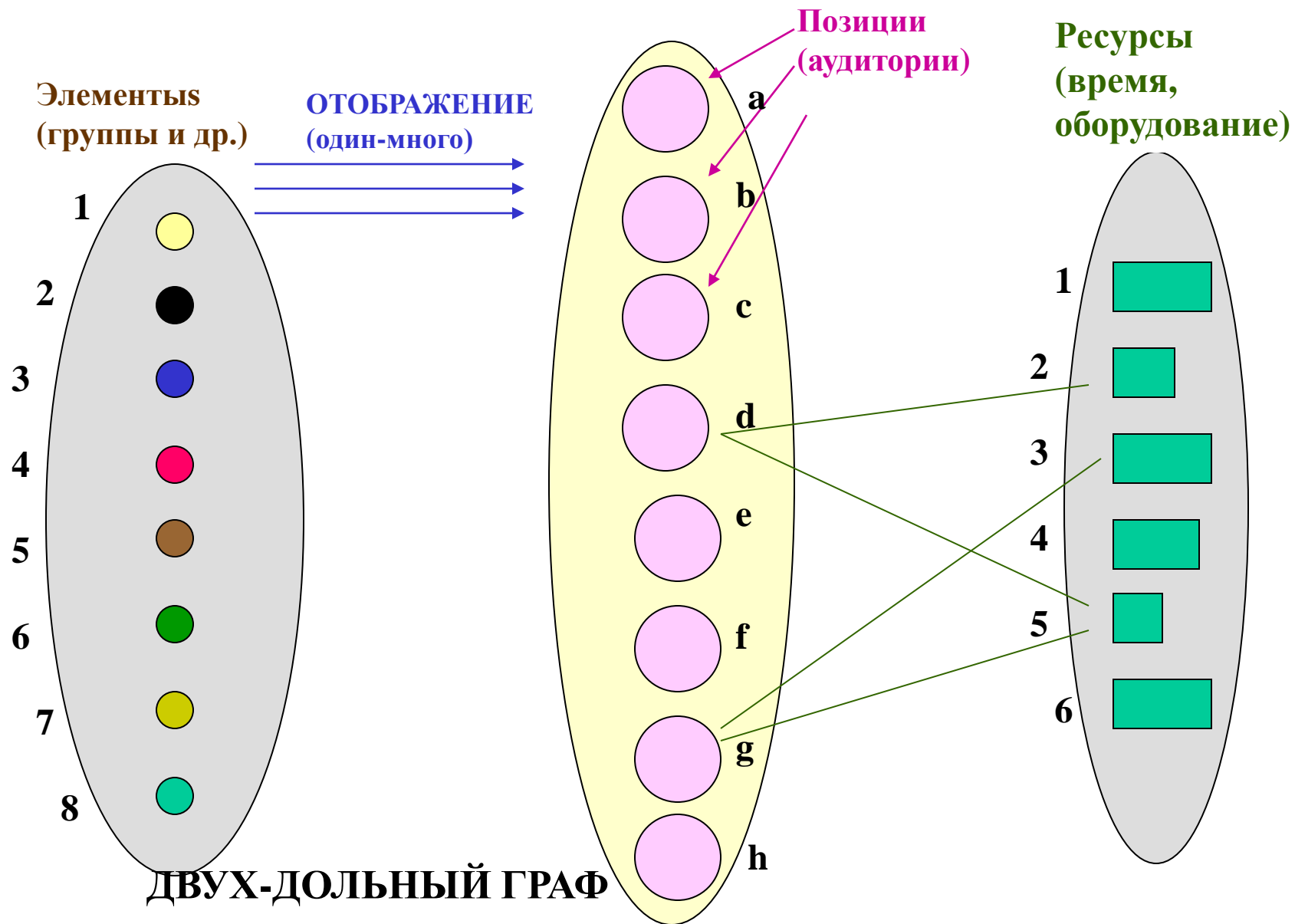
(i) матрица “потока” (или “полезности”) C где элемент (i, j) представляет поток между элементами (например, оборудованием) i и j ,

(ii) матрица расстояний D где (i, j) (элемент $p(i), p(j)$) представляет Расстояние между позициями $p(i)$ and $p(j)$.

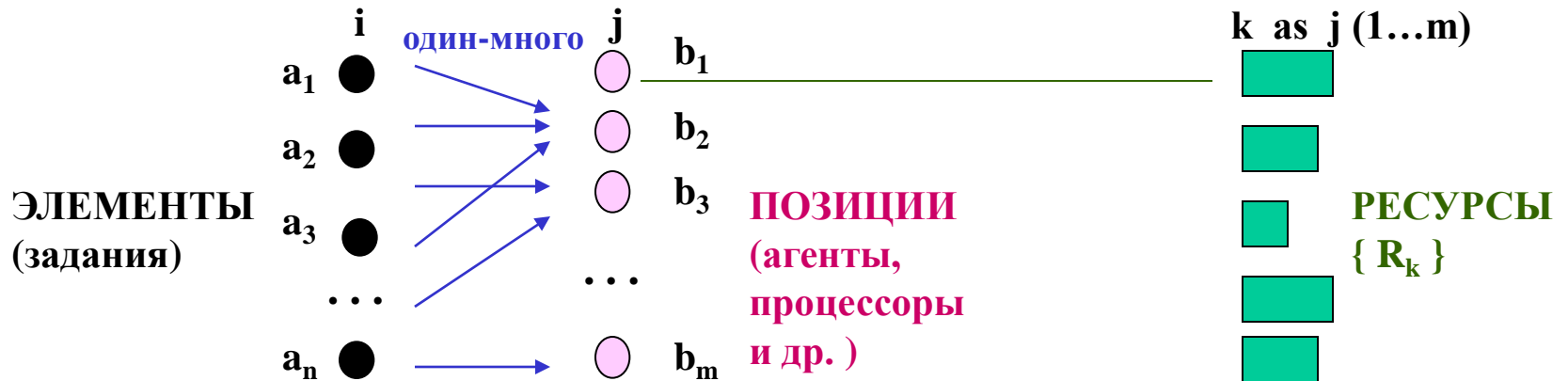
Таким образом, задача QAP имеет вид

$$\max_{p \in \Pi} \sum_{i=1}^n \sum_{j=1}^n c_{ij} d_{p(i)p(j)}$$

Квадратичная задача о назначении (Quadratic Assignment Problem QAP)



Обобщенная задача о назначении GAP (Лекция 30)



ФОРМУЛИРОВКА (алгебраическая):

Элементы (персонал, оборудование, задания): $A = \{ a_1, \dots, a_i, \dots, a_n \}$

Позиции (процессоры и др.) $B = \{ b_1, \dots, b_j, \dots, b_m \}$

(здесь $n = m$)

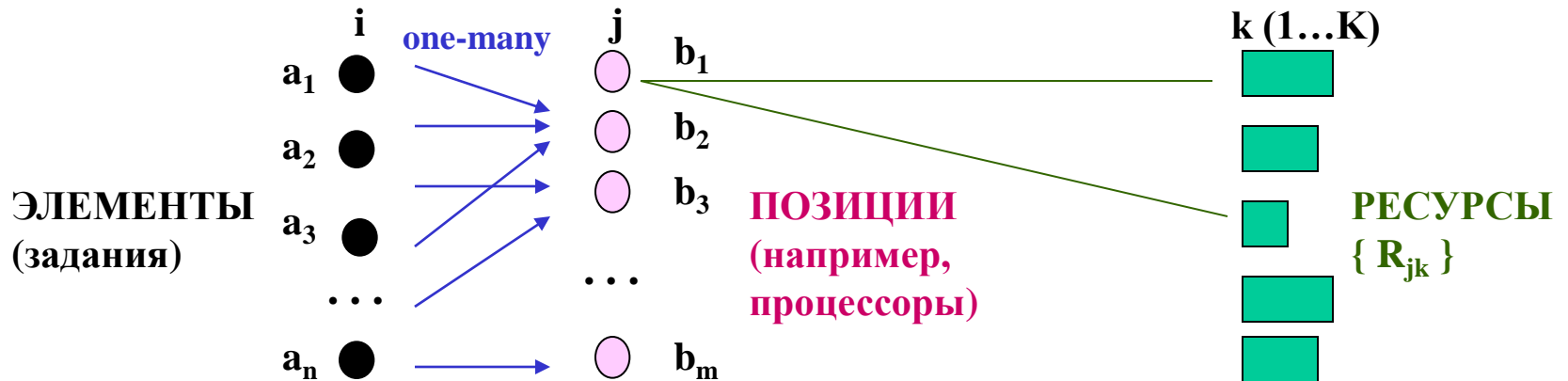
Эффективность пары a_i и b_j is: $c(a_i, b_j)$

$x_{ij} = 1$ если a_i размещено на позиции b_j и 0 в противном случае ($x_{ij} \in \{ 0, 1 \}$)

Задача 1 : $\max \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$

s.t. $\sum_{i=1}^n r_{ik} x_{ij} \leq R_j \quad \forall j$ (R_k – это ресурс агента k)
 $\sum_{j=1}^n x_{ij} = 1 \quad \forall i$

Обобщенная задача о назначении GAP (общие ресурсы) (Лекция 30)



ФОРМУЛИРОВКА (алгебраическая):

Элементы (e.g., personnel, facilities, tasks): $A = \{ a_1, \dots, a_i, \dots, a_n \}$

Позиции (процессоры и др.) $B = \{ b_1, \dots, b_j, \dots, b_m \}$

(здесь $n = m$)

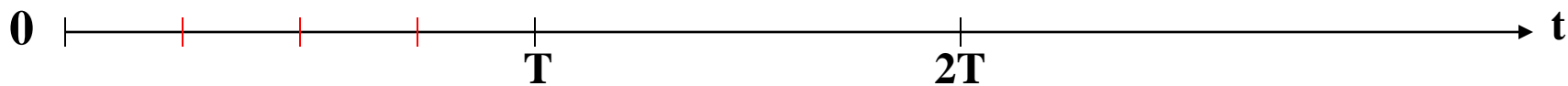
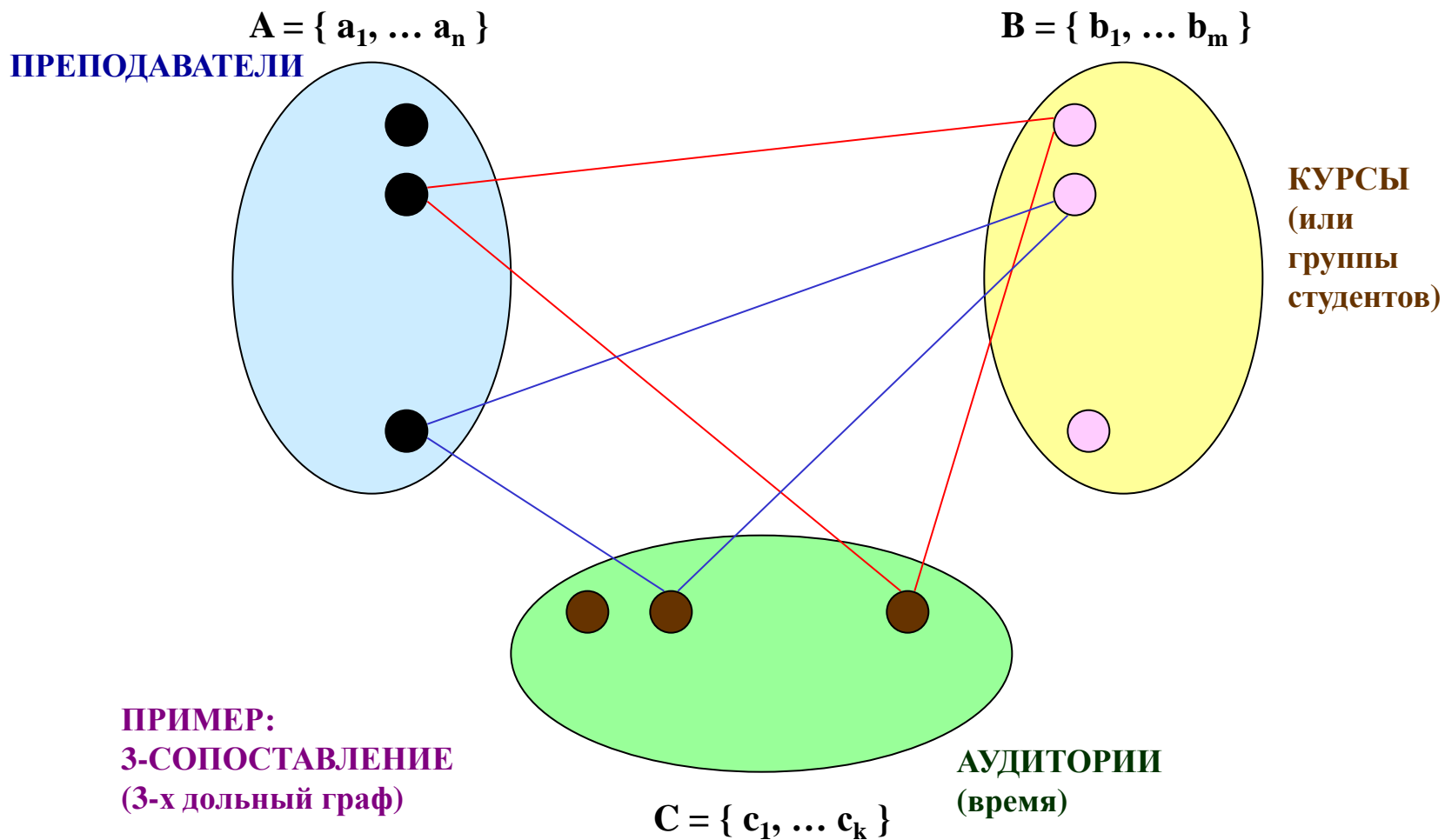
Эффективность пары a_i и b_j : $c(a_i, b_j)$

$x_{ij} = 1$ если a_i размещен на позицию b_j и 0 в противном случае ($x_{ij} \in \{0, 1\}$)

Задача 2 : $\max \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$

s.t. $\sum_{j=1}^m \sum_{i=1}^n r_{ik} x_{ij} \leq R_k \quad \forall k$ (K общих ресурса)
 $\sum_{j=1}^m x_{ij} = 1 \quad \forall i$

Специальный «класс» задач календарного планирования (Timetabling) ПЛЮС время (циклы)



ОГРАНИЧЕНИЯ (примеры):

ДЛЯ ЛЕКТОРОВ:

Лектор 2 может преподавать только в Понедельник и Пятницу (lecturers - time)

Лектор 11 должен быть после Лектора 12 (по лекциям)

Lecturers 5 and 7 can teach only in auditoriums 9 or 10 (lecturers-auditoriums)

ДЛЯ ГРУПП СТУДЕНТОВ:

Группе 1 требуется аудитория 5 в Понедельник утром (группы - время)

Группы 7, 8, 9 должны иметь того же Лектора 1 (этот же курс)

(группы - лекторы) * (группы - группы)

Группа 4 предпочитает Лектора 10 (группы - лекторы)

Группа 5 требует Лекторов 5 & 8 как соседей по времени (группы - лекторы)

ДЛЯ АУДИТОРИЙ:

Аудитория 1 открыта в Среду (аудитории - время)

Аудитория 4 соответствует только группам 5, 7, 8, 9 (аудитории - группы)

ТИПЫ ОГРАНИЧЕНИЙ (по элементам):

Лекторы – лекторы
Лекторы - группы
Лекторы - аудитории
Лекторы - время

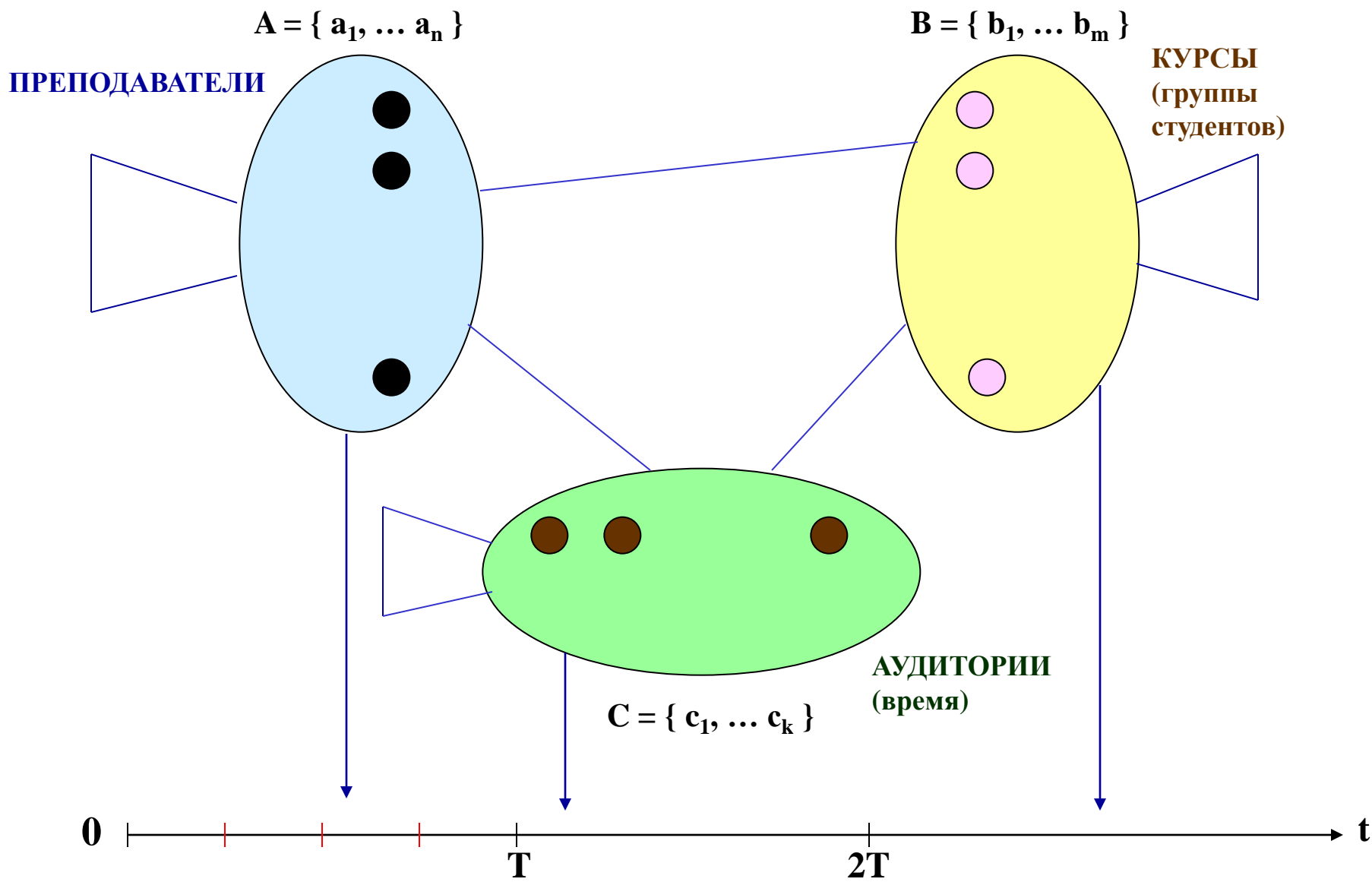
Группы - группы
Группы - аудитории
Группы - время

Аудитории – аудитории
Аудитории - время

ТИПЫ ОГРАНИЧЕНИЙ (по типам):

Логические (бинарные отношения и др.)
Количественные (ограничения по ресурсам и др.)

Специальный «класс» задач календарного планирования (Timetabling): ограничения



**1.ЗАДАЧИ ТЕОРИИ РАСПИСАНИЙ В КОММУНИКАЦИОННЫХ СИСТЕМАХ
(COMMUNICATION TIMETABLING)**

**2.ЗАДАЧИ ТЕОРИИ РАСПИСАНИЯ В МОНИТОРИНГОВЫХ СИСТЕМАХ
(MONITORING TIMETBLING)**

G.L. Nemhauser, M.A. Trick, Scheduling a major college basketball conference. Operations Research, 46(1), 1-8, 1998.

ФОРМУЛИРОВКА, ДАННЫЕ & ОГРАНИЧЕНИЯ:

- 1. Базовые объекты:** команда, slots (weekday slots - слот в середине недели, слот в конце недели - weekend slots),
Каждая команда играет дважды в неделю, 8 home slots – слотов дома, 8 away slots - слотов на выезде
- 2. Ограничения для команд и слотов:** home slots, away slots, цепочка: home-away-и т.д.
- 3. «Patterns» и ограничения:** ≤ 2 away games (игр), ≤ 2 home games (игр)
- 4. Ограничение для пар команд:** пары кандидатов (из предыдущего расписания)

СХЕМА РЕШЕНИЯ:

ШАГ 1. A pattern – это строка H (home), A (away), B (bye).

Примеры: HAA, HBW

Найти множество pattern, пример: HHA, AHA, HAH, AAH (команды a, b, c, d)
(enumeration & integer programming)

ШАГ 2. Назначить игры на patterns. Это дает timetabling
(integer programming)

ШАГ 3. Назначить команды на patterns. Это дает календарный план
(quadratic assignment problem)

ВЫЧИСЛЕНИЕ: 24 часов