LECTURE 1. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.1. Systems, structure, life cycle, examples.

PLAN:

1.Profile of specialist 2.About course 3. Illustrative example of system and life cycle

4.Role of mathematics (models, algorithms) 5.Life cycle and logistic curve

6.Russian engineering experience 7.Levels of system complexity

8.Simple examples of systems 9.Monitoring systems

Sept. 3, 2004

1.Profile of specialist

STRUCTURE:

A. Basic scientific disciplines
- 1.Mathematics
- 2.Physics, physical-chemical processes , etc.

B.Special engineering disciplines
- 1.Radioengineering, etc.

C. Information technology

D. Management / economics

E. System thinking

F. Creativity

G. Experience in applied domains

2.About course

A.Systems, multi-disciplinary systems
(airplane, machine, radar, team, plan, manufacturing systems, etc.)

B.Life cycle (life cycle engineering)

C.Design schemes (frameworks), support of life cycle

D.Structure of the course:
(1)lecture blocks
(schemes, models/methods, technological problems, applied examples)
(2)Assignment (simple preliminary works)
(3)Projects (realistic applied systems)

E.Neighbor courses:
*system engineering
*system design (e.g., architecture, mechanical engineering)
*technology management
*multicriteria decision making
*combinatorial optimization
*knowledge engineering
*applications (engineering, management, information technology)

3.Illustrative example of life cycle

A.Life cycle:

*preliminary research
*R&D
*manufacturing
*testing
*marketing
*utilization & maintenance
*recycling

B.A system (airplane):

*body
*engine
*electronics (control, communication, etc.)
*human environment

Additional support subsystem:

*maintenance
*training
*recycling subsystem
*etc.

4.Role of mathematics

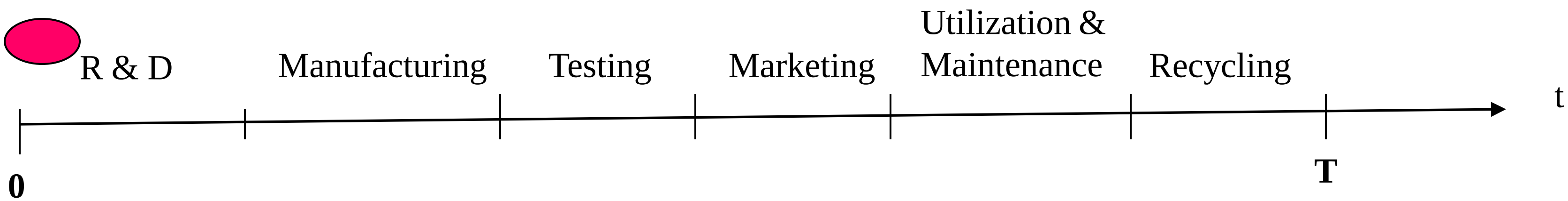
A.Models
- *structural models (e.g., graphs, networks)
- *optimization models
- *multicriteria decision making
- *differential equations (dynamics)
- *game theory
- *uncertain models (probability, fuzzy sets)

B.Algorithms

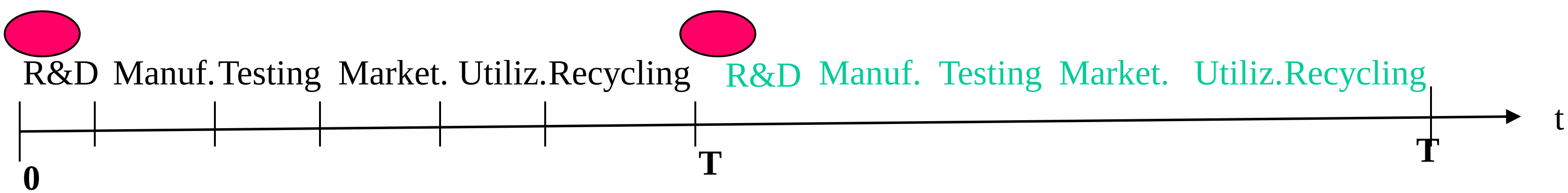
C.Solving schemes

Real New Application =>
new or modified models / algorithms

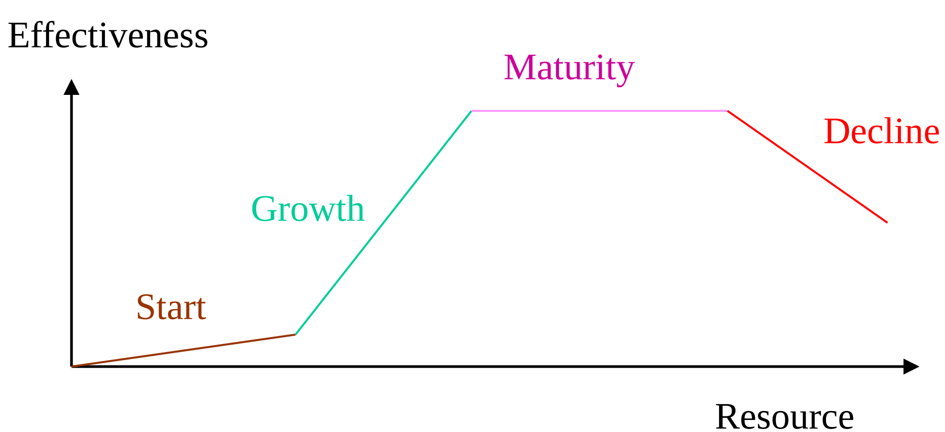
5.Life cycle and logistic curve



T: about 12 years (submarines, airplanes, nuclear technology, etc.) TENDENCY: increasing T (2 years, 6 months)



RESULT: need of specialists in system design & specialists in life cycle engineering



6.Russian engineering experience

Complex systems:

1. **Airplanes**
2. **Aerospace systems (stations, etc.)**
3. **Communication systems**
4. **Nuclear technology**
5. **Defense systems (radars, etc.)**
6. **etc.**

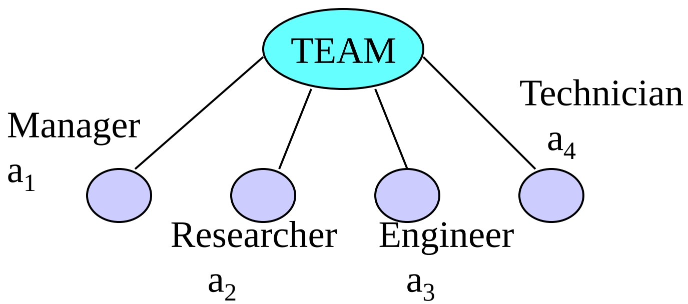
Factors:

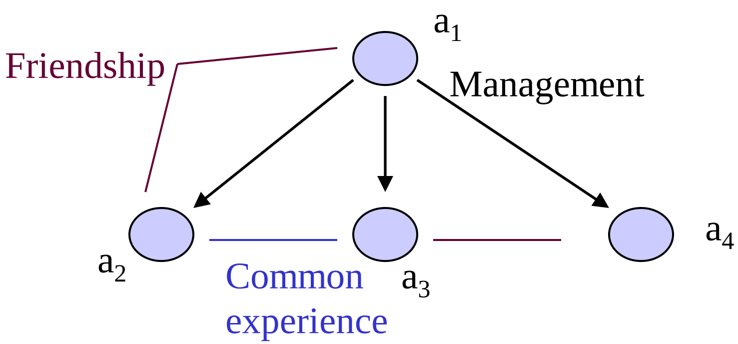
1. **Creative people**
2. **Educational system**
3. **Engineering traditions (in design of complex systems)**
4. **Complex problems**
   (very large territory, various environments, etc.)

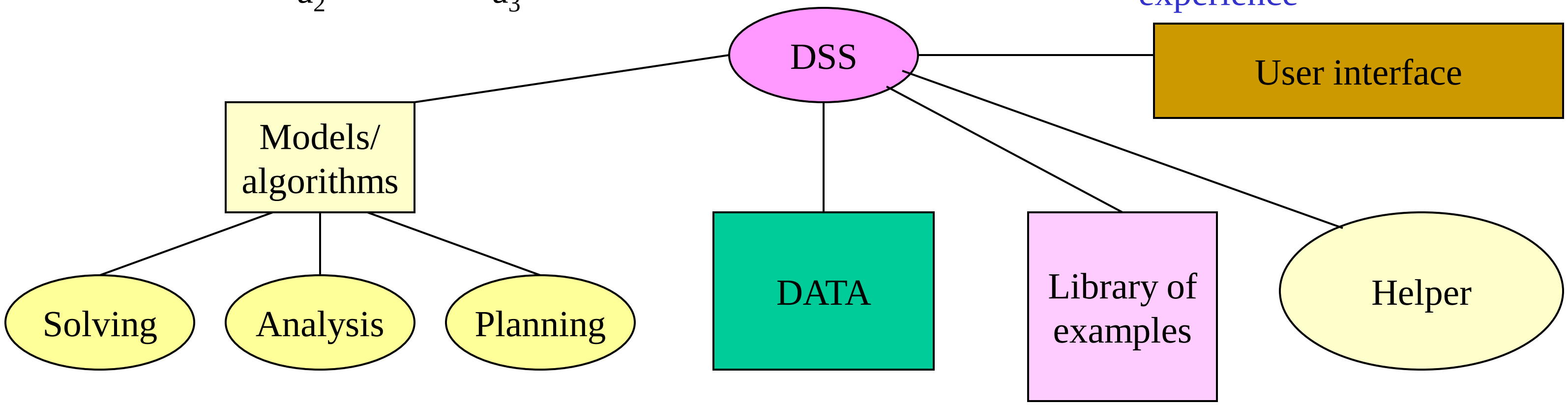
7.Levels of system complexity (A.Shenhar)

Level 1. Arrays (network of systems, e.g., network of radar defense systems)
Level 2. System (multiple functions; radar, defense system)
Level 3. Assembly (one function: TV)
Level 4. Component

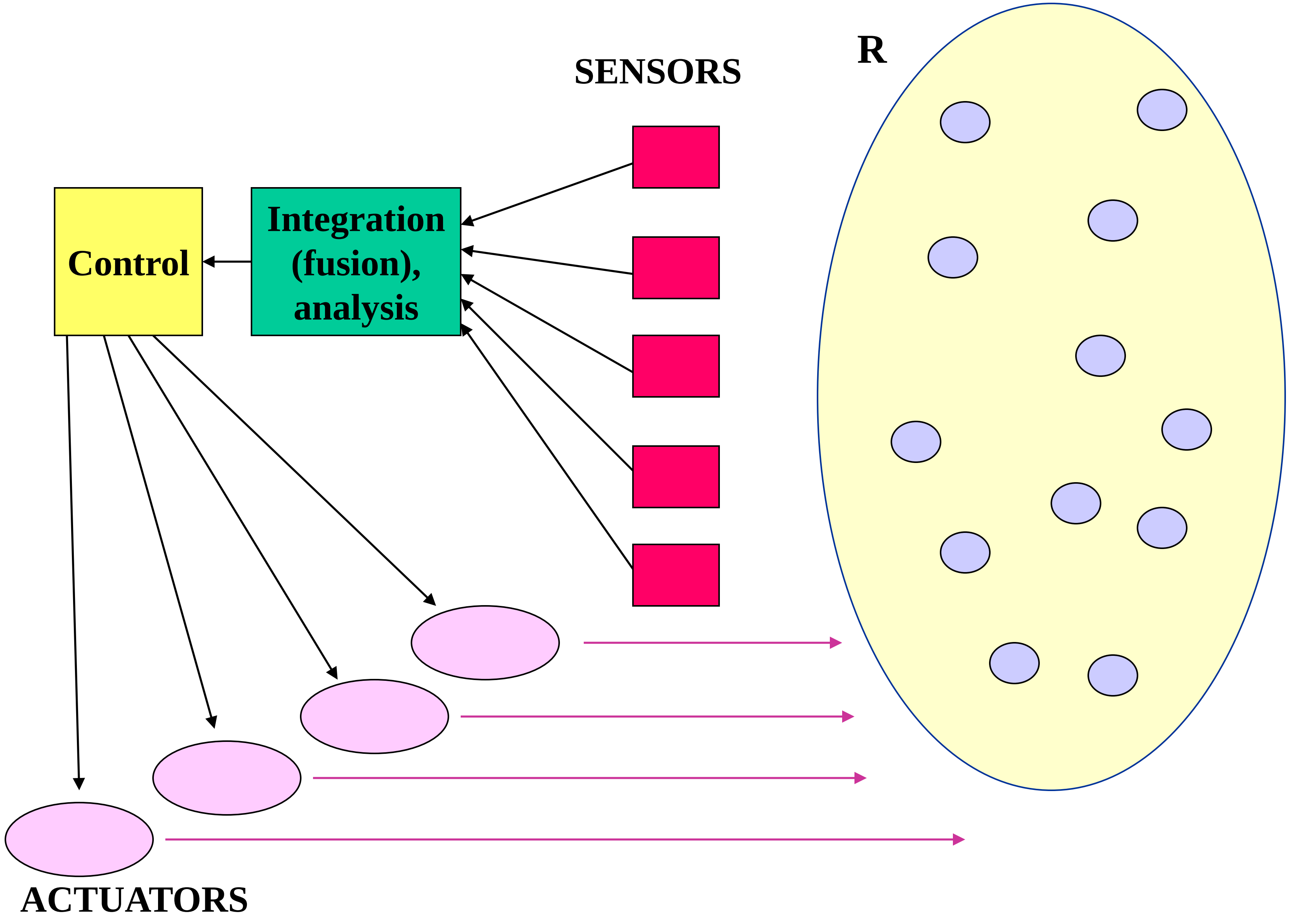
8.Simple examples of systems







9.Monitoring systems



LECTURE 2-3. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.2. Modularity, system decomposition (partitioning), example.

L.3. Structural models (graphs, networks, binary relations), examples.

PLAN:

1.Decomposition (partitioning) of systems *decomposition – partitioning; *illustrative examples; *approaches

2.Issues of modularity: *description and a basic linguistic analogue

*applied examples (mechanical engineering, , aerospace engineering, etc.) *goals and results

3. Structural models: *graphs (graphs, digraphs, sign graphs)

*simple structures (e.g., chains, trees, parallel-series graphs)

*problems on graphs (metric/proximity, optimization, advance models)

Sept. 4, 2004

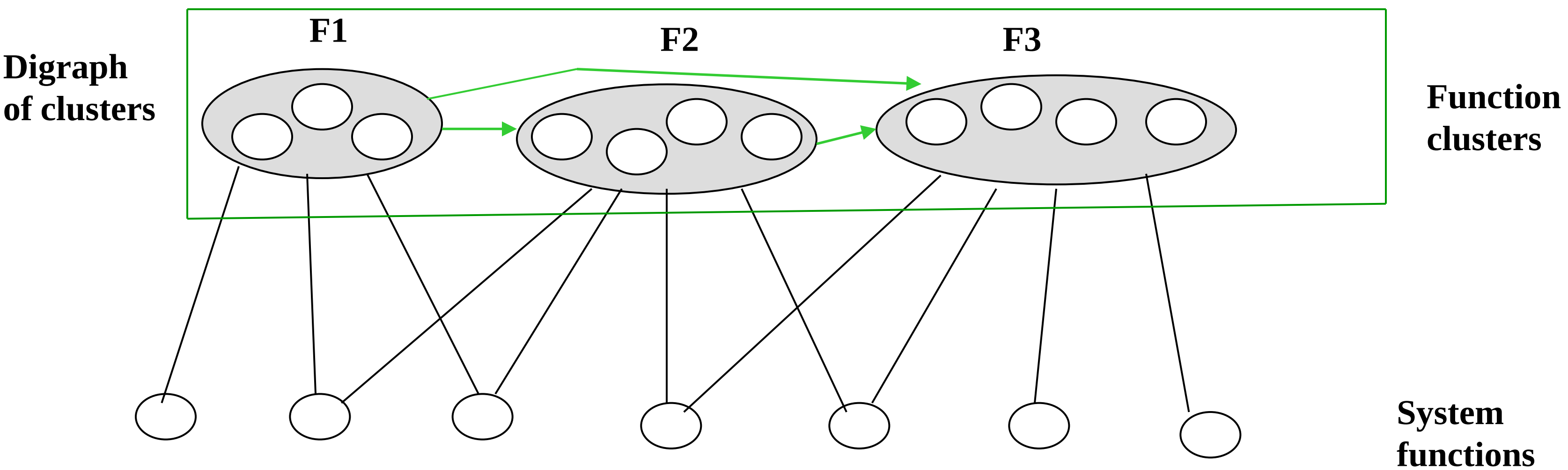
1.Decomposition / partitioning of systems

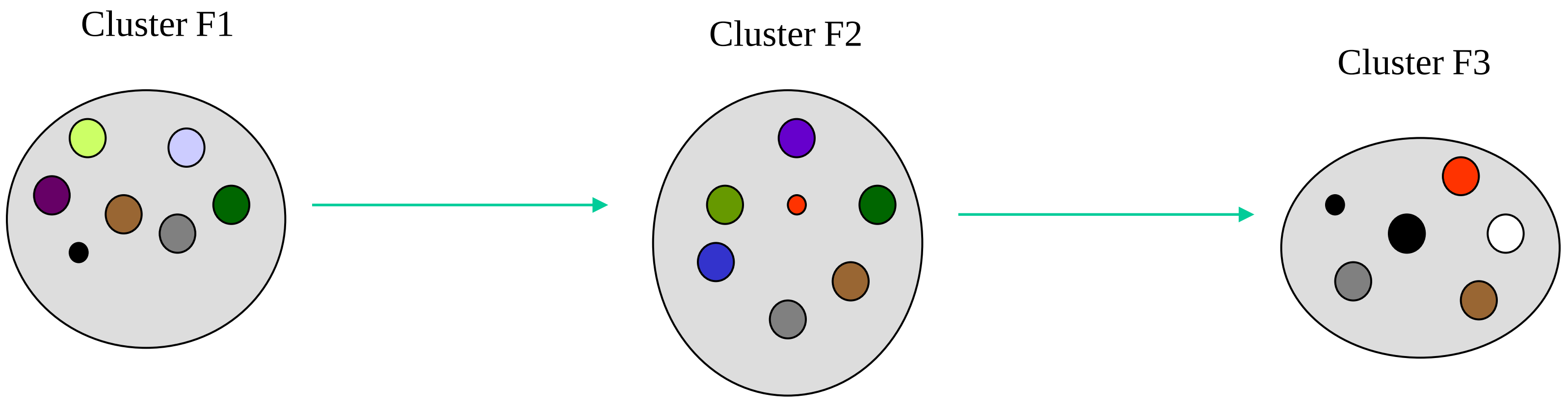
Decomposition: series process (e.g., dynamic programming)
Partitioning: parallel process / dividing (combinatorial synthesis)

Methods for partitioning:
*physical partitioning
*functional partitioning

Examples (for airplane, for human)
Examples for software:
1.Series information processing (input, solving, analysis, output)
2.Architecture:
data subsystem, solving process, user interface,
training subsystem, communication
3.Additional part: visualization (e.g., for data, for solving process)
4. Additional contemporary part (model management) as follows:
*analysis of an initial applied situation,
*library of models / methods,
*selection / design of models / methods,
*selection / design of multi-model solving strategy

1.Decomposition / partitioning of systems: Example for multifunction system testing

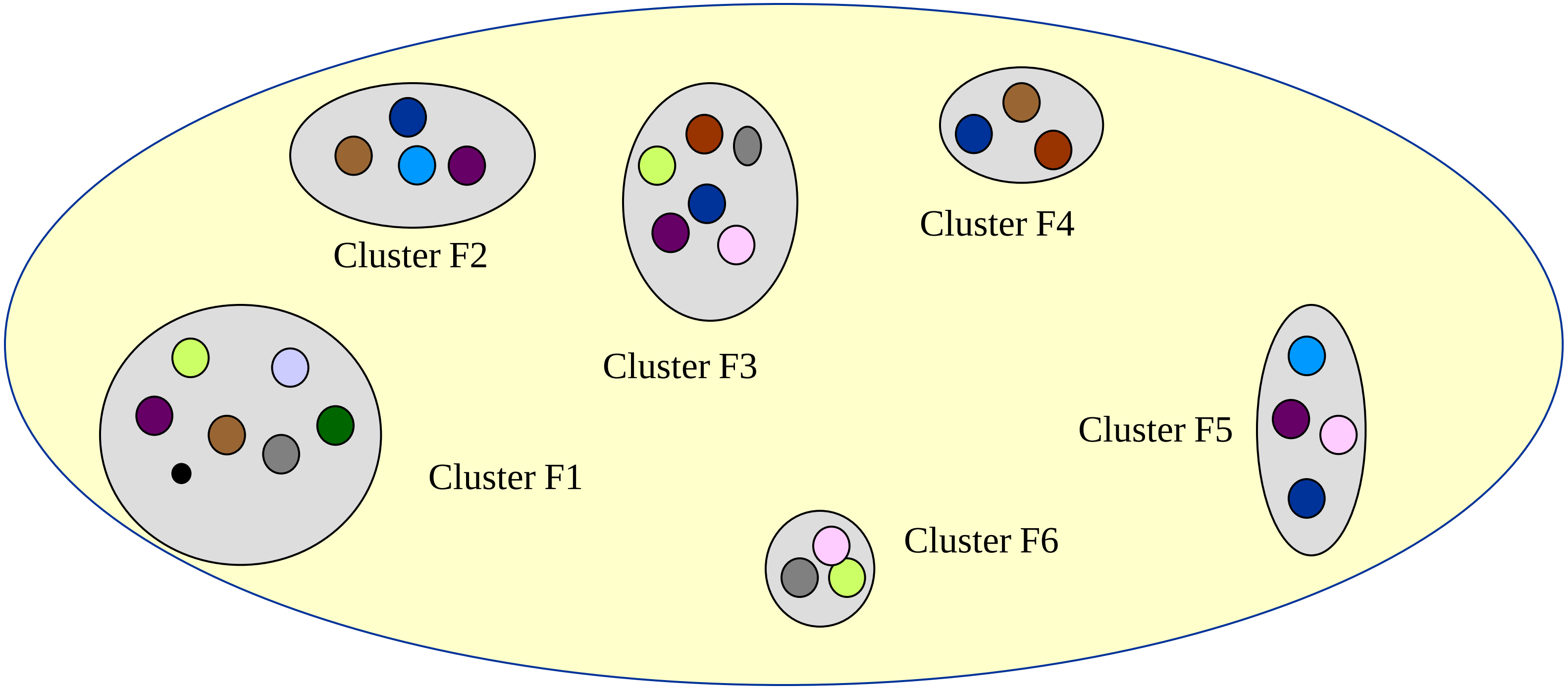




Main approaches to partitioning of systems

A.Content analysis and experience:
*by functions (basic functions, auxiliary functions)
*by system parts (physical partitioning)

B.Cluster analysis (clustering)



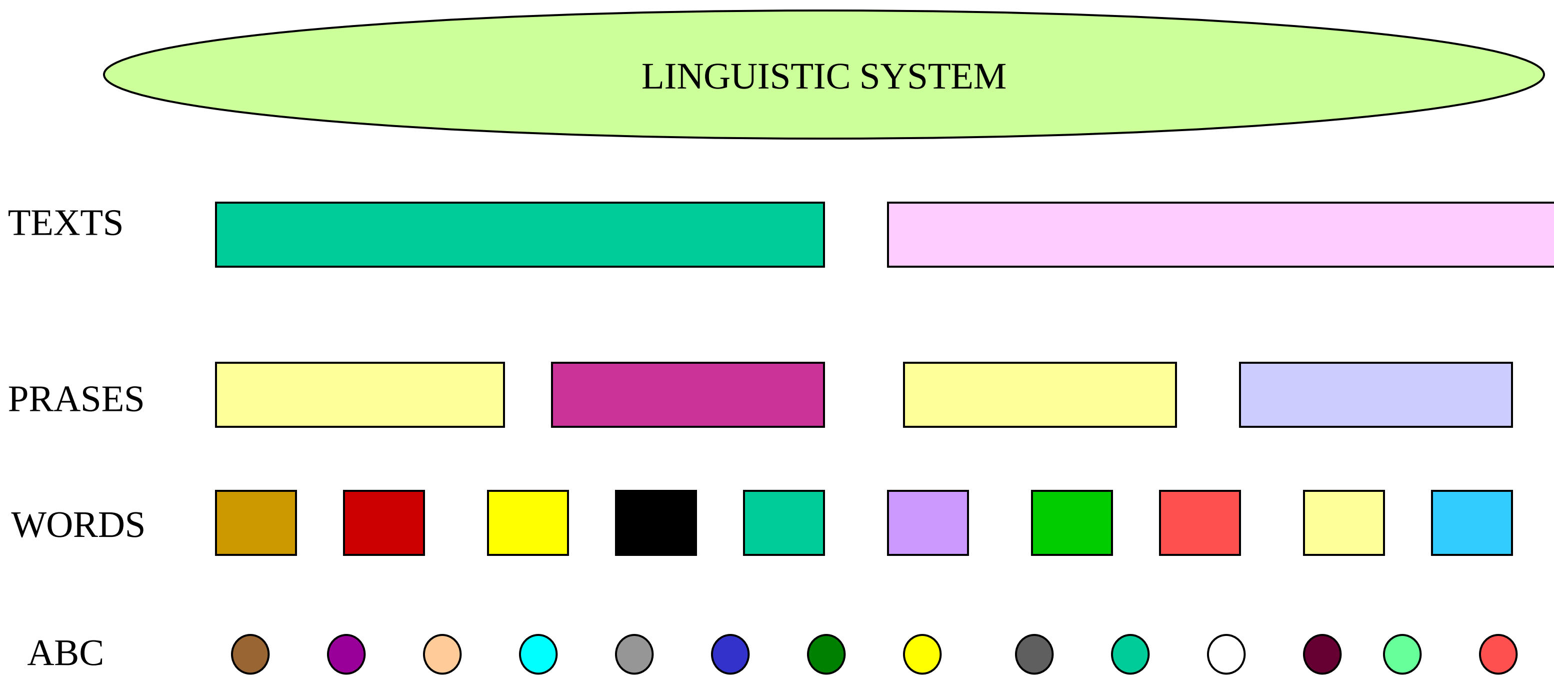
2.Issues of modularity

PRINCIPLES FOR MANAGEMENT OF COMPLEXITY :

*discrete pieces (modules)

*standard interfaces for module communication

Applications: *new technology design * organizational design



Applied examples for usage of modularity

1.Genetics

2.Reconfigurable manufacturing

3.Software libraries of standard modules

4.Combinatorial Chemistry:
  *molecular design in chemistry and biology
  *drug design
  *material engineering
  *etc.

5.Aerospace & mechanical engineering

6.Electronics

7.Civil engineering

Main goals of modularity and resume

Main goals:
1.Management of complexity
2.Parallel work
3.Accomodation of future uncertainty
4.Variety of resultant modular systems
5.Flexibility, adaptability, reconfigurability of resultant modular systems

Resume:
1.Simple design process & simple all phases of life cycle
2.Short life cycle of product, long life cycle of product modules
3.Reconfigurable systems (e.g., manufacturing systems):
long life cycle for system generation
4.Simple design and support of product families (airplanes, cars, etc.)
5.Simple design and support of different products
(on the basis of module libraries as reuse)

3.Structural models

A.GRAPHS
1.Graphs
2.Digraphs (directed graphs, oriented graphs - orgraphs)
3.Graphs / digraphs with weights (for vertices, for edges / arcs)
4.Simple graphs: chains, trees, parallel-series graphs, hierarchies
5.Sign graphs

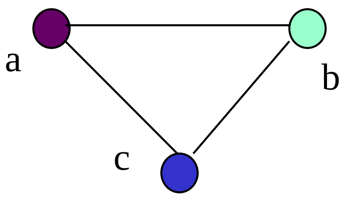
B.NETWORKS

C.AUTOMATA

D.BINARY RELATIONS

Illustration for graphs / digraphs

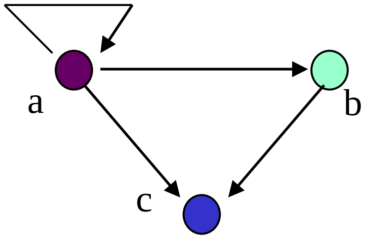
Graph: G = (A,E) where a set of nodes (vertices) A={1,…,n}
and a set of edges $E \subseteq A \times A$ (pairs of nodes)
Example: A={a, b, c}, E={(a, b), (b, c), (a, c)}



Matrix

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

Digraph (orgraph): G = (A,E) where a set of nodes (vertices) A={1,…,n}
and a set of arcs $E \subseteq A \times A$ (pairs of nodes)
Example: A={a, b, c}, E={(a, a), (a, b), (b, c), (a, c)}

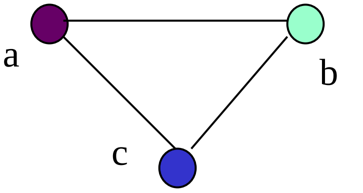


Matrix

	a	b	c
a	1	1	1
b			1
c			

Illustration for graphs with weights

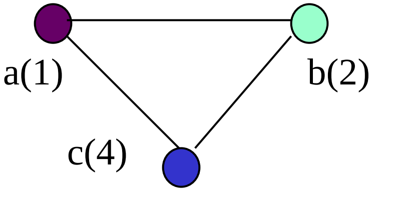
Graph (weights of edges): G = (A,E) where a set of nodes (vertices) A={1,…,n} and a set of edges $E \subseteq A \times A$ (pairs of nodes)
Example: A={a, b, c}, E={(a, b), (b, c), (a, c)}



Matrix

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

Graph (weights of edges & nodes): G = (A,E) where a set of nodes (vertices) A={1,…,n} and a set of edges $E \subseteq A \times A$ (pairs of nodes)
Example: A={a, b, c}, E={(a, b), (b, c), (a, c)}
(weights of nodes are pointed out in brackets)

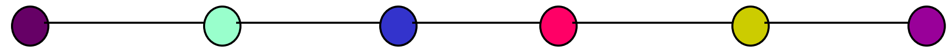


Matrix

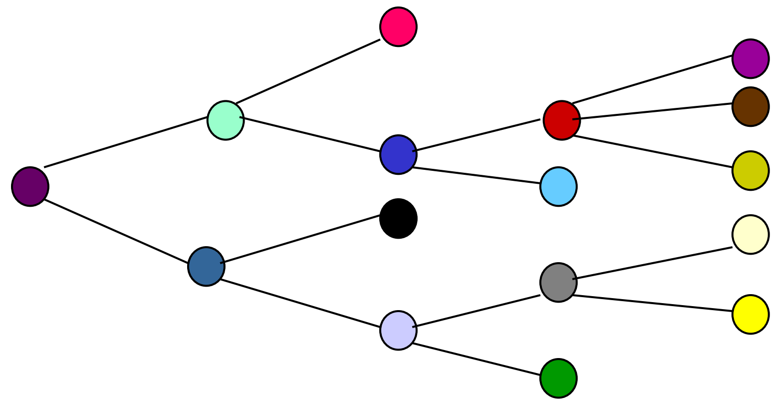
	a	b	c
a		2	5
b	2		3
c	5	3	

Simple structures (chains, trees, parallel-series graphs)

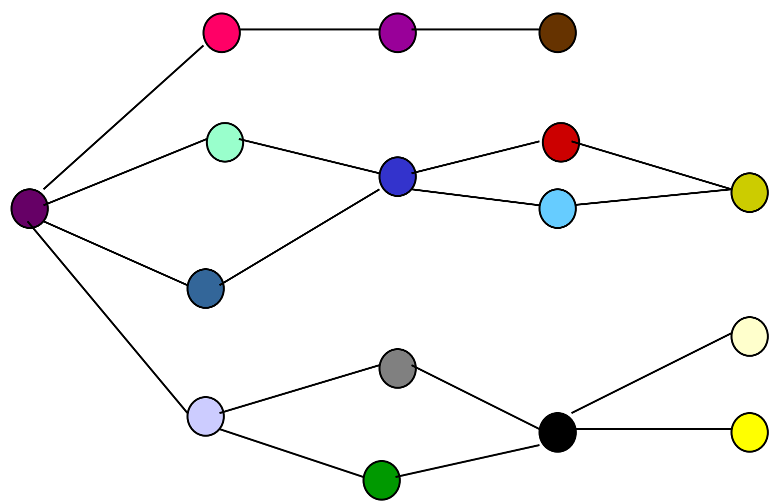
CHAIN



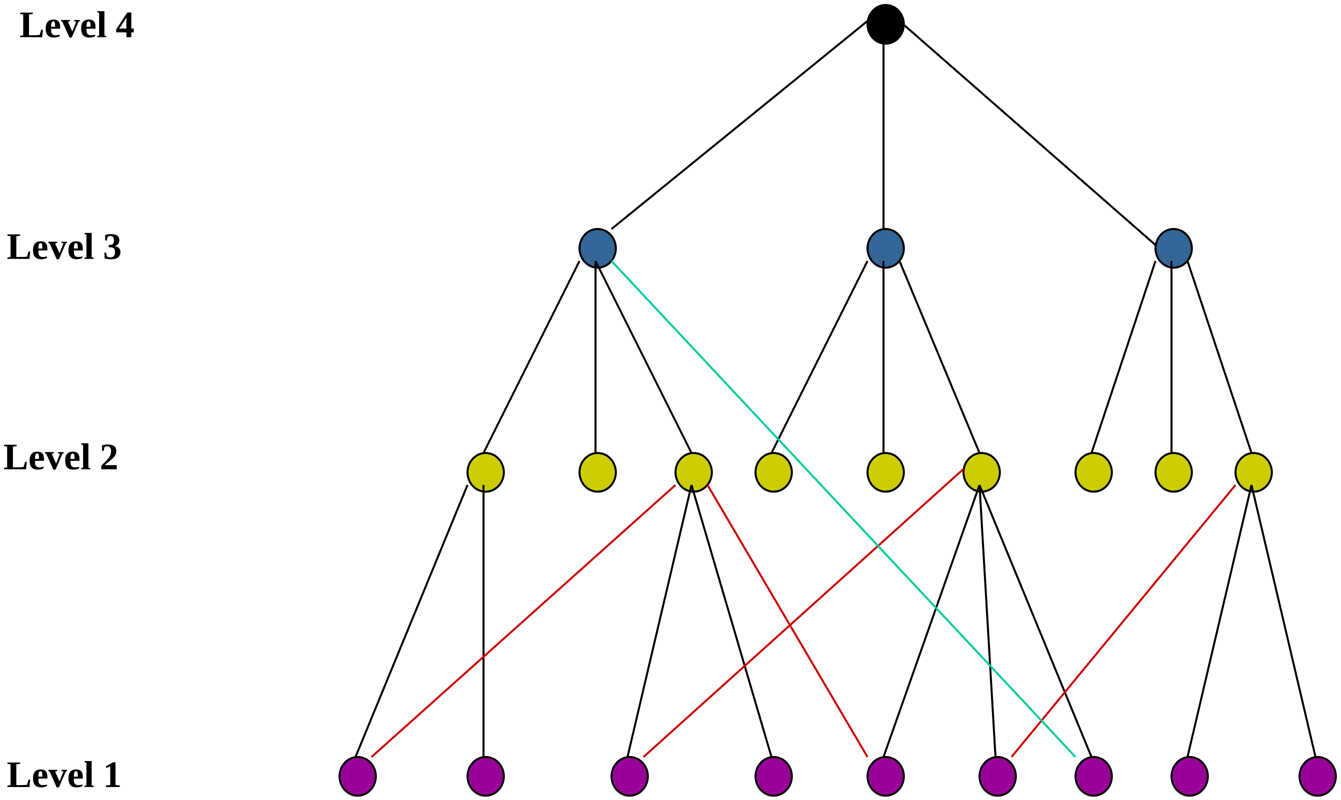
TREE



PARALLEL-SERIES GRAPH

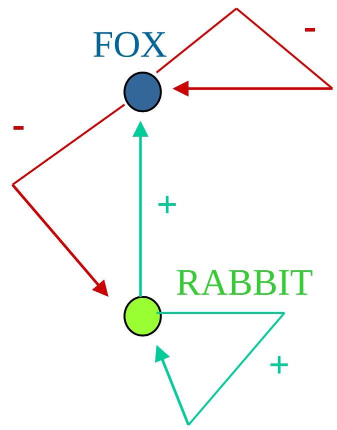


Simple structures (hierarchy)

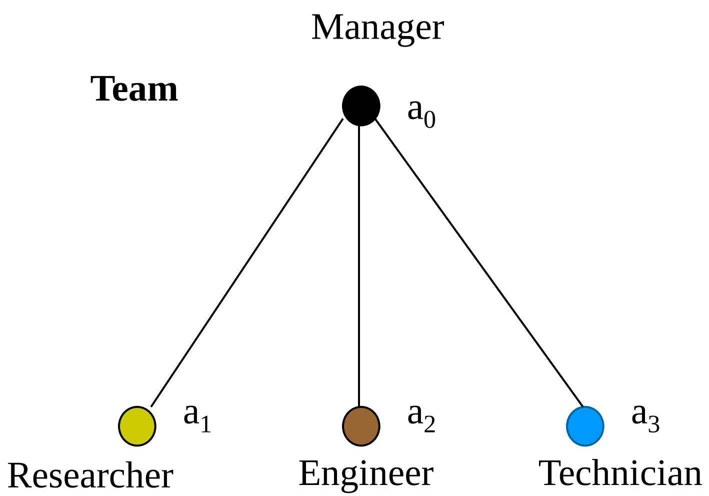


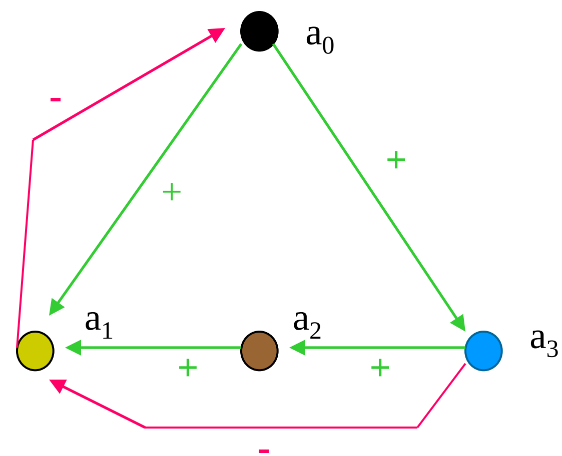
Sign graph: illustrative examples

Ecological system



Team





Some advanced structural models

1. Multigraphs

2. Graphs with versions for nodes (vertices)

3. Graphs with “vector weights”

4. Graphs with fuzzy weights

Problems on graphs

A.Metric / proximity (in graph between nodes, between graphs)
Proximity between graphs:
1.metrics, 2.edit distance (minimal “cost” transformation), 3.common part

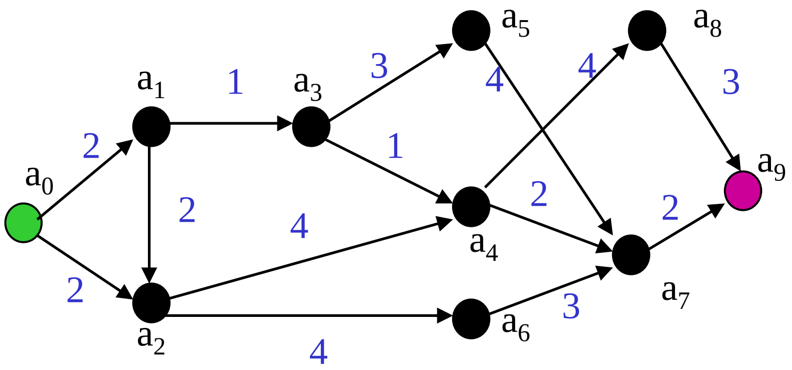
B.Optimization on graphs:
1.Shortest path
2.Spanning tree (& close approximation problems: spanning by other simple structures)
3.Traveling salesman problem
4.Minimal Steiner tree
5.Ordering of vertices
6.Alocation on graphs
7.Covering problems

C.Balance problem for sign graphs

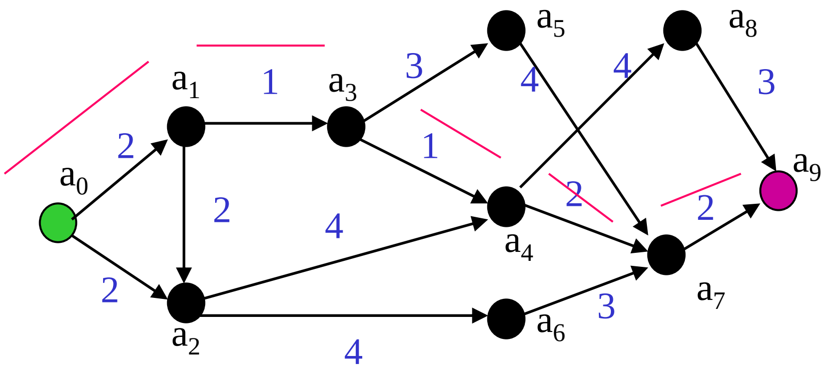
D.Clustering (dividing into interconnected groups)

Optimization problems on graphs: illustrations

BASIC GRAPH (DIGRAPH):
weights for arcs (or edges)

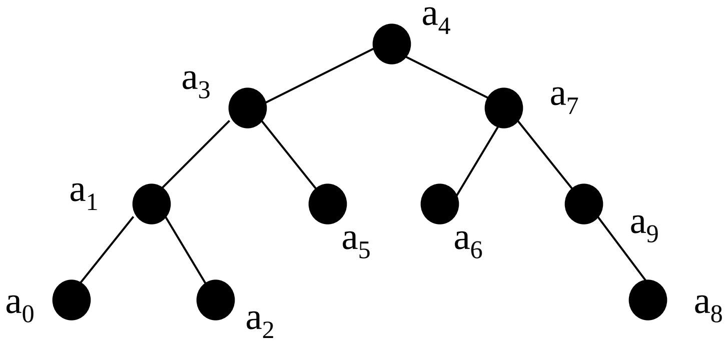


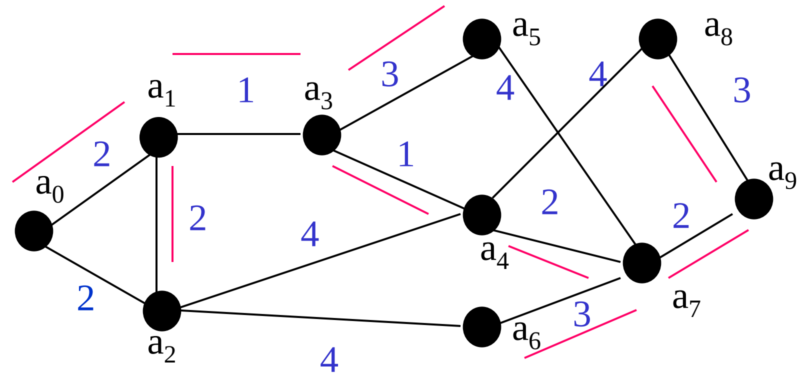
Shortest Path for $< a_0, a_9 >$:
$L = < a_0, a_1, a_2, a_3, a_4, a_7, a_9 >$
$2+1+1+2+2 = 8$



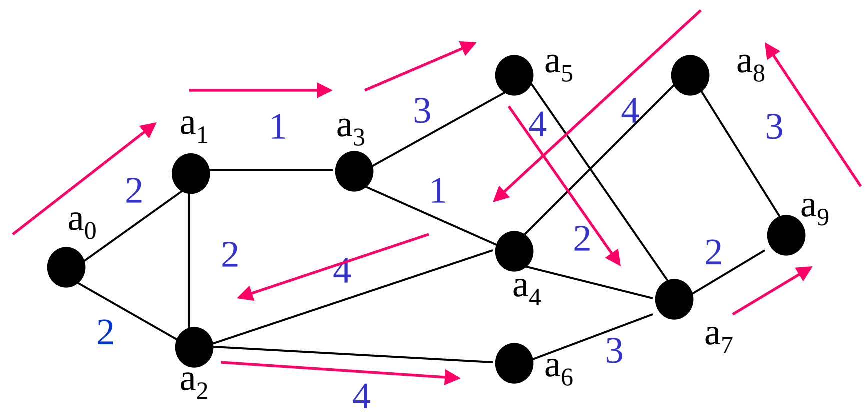
Optimization problems on graphs: illustrations

Spanning tree (length = 19):



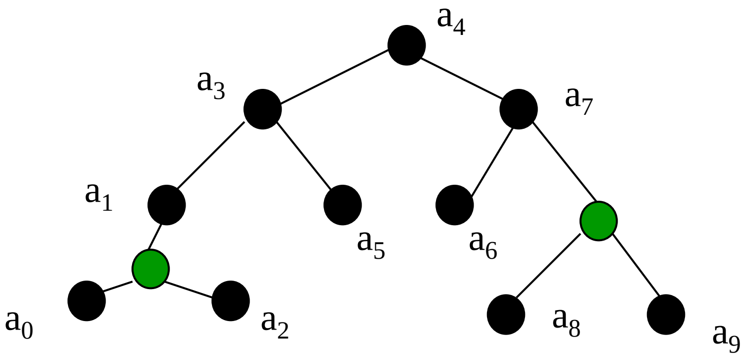


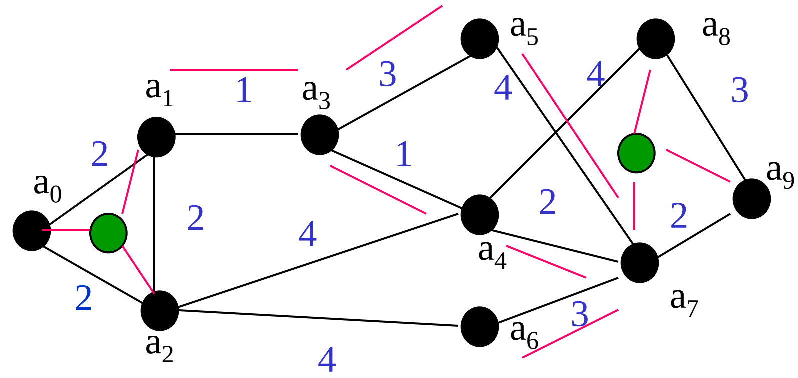
Traveling Salesman Problem :
L = < a_0,a_1,a_3,a_5,a_7,a_9,a_8,a_4,a_2,a_6>
2+1+3+4+2+2+3+4+4+4



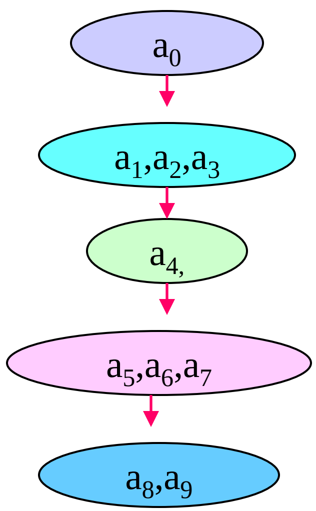
Optimization problems on graphs: illustrations

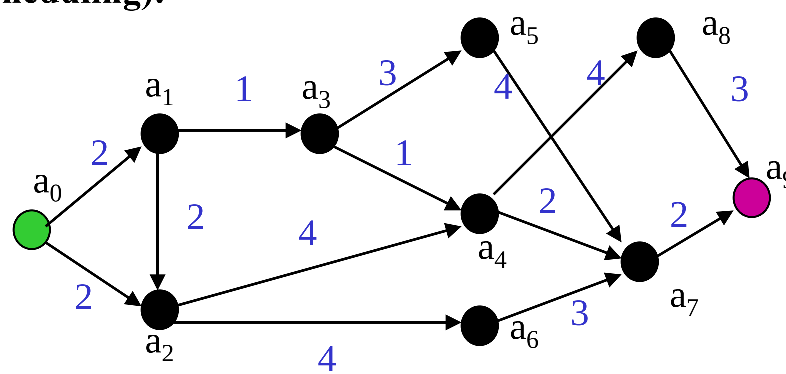
Steiner tree (example):





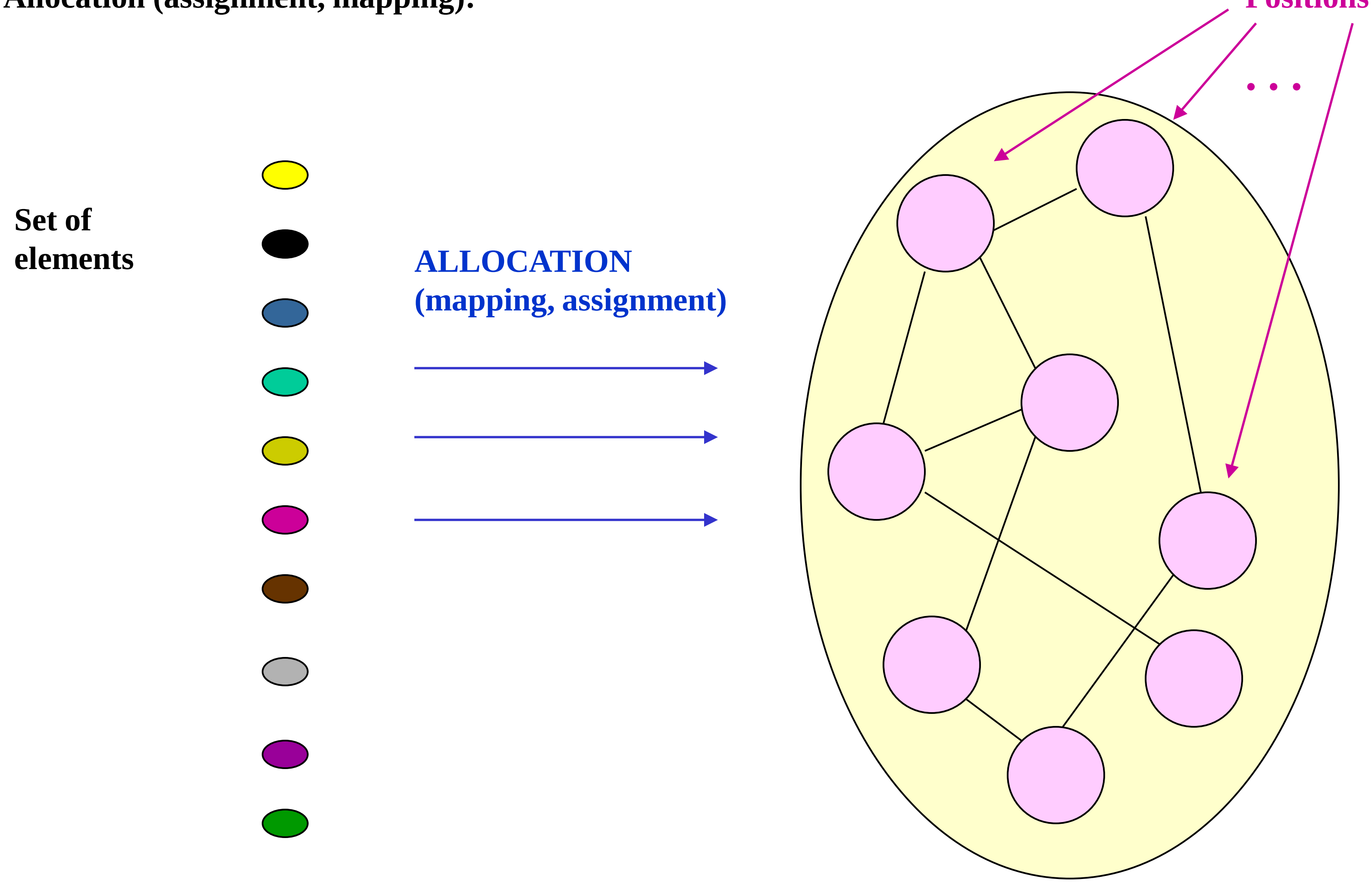
"Ordering" Problem (close problems: sequencing, scheduling):





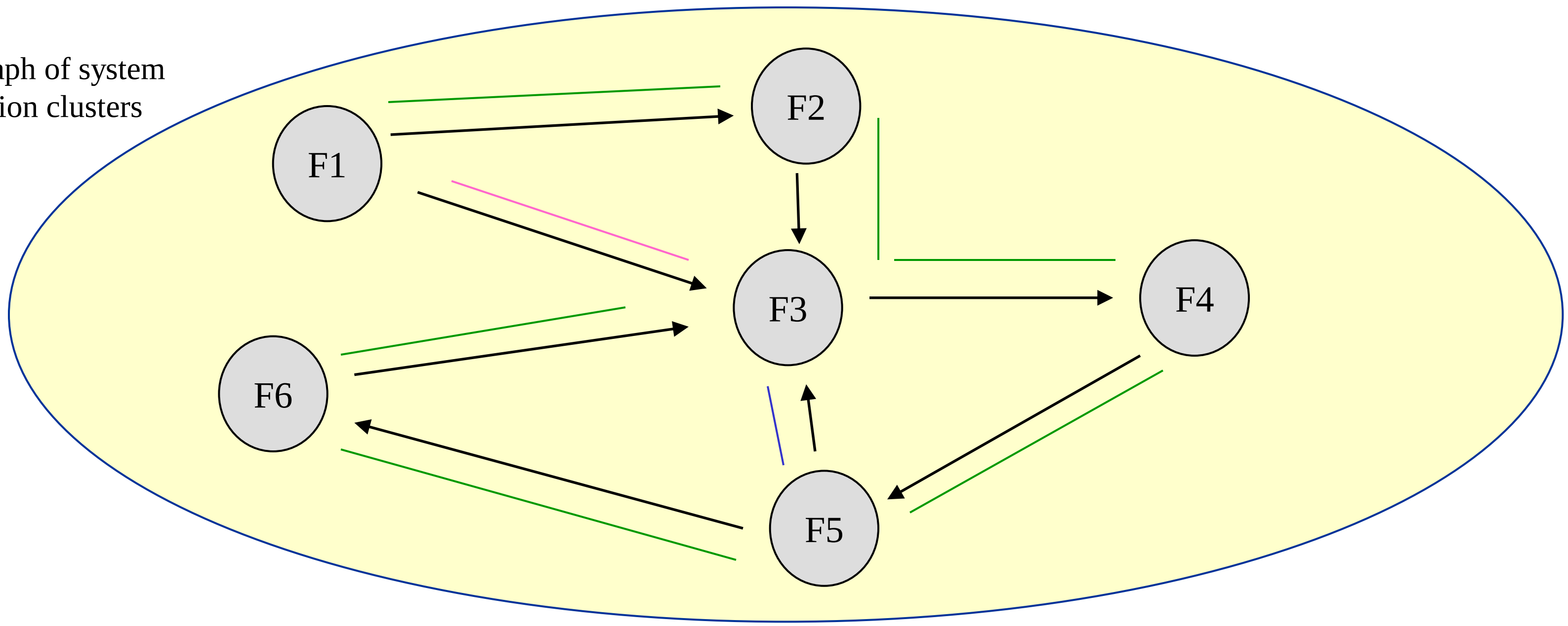
Optimization problems on graphs: illustrations

Allocation (assignment, mapping):



Example: system function clusters and covering by chains (covering of arcs)

Digraph of system function clusters



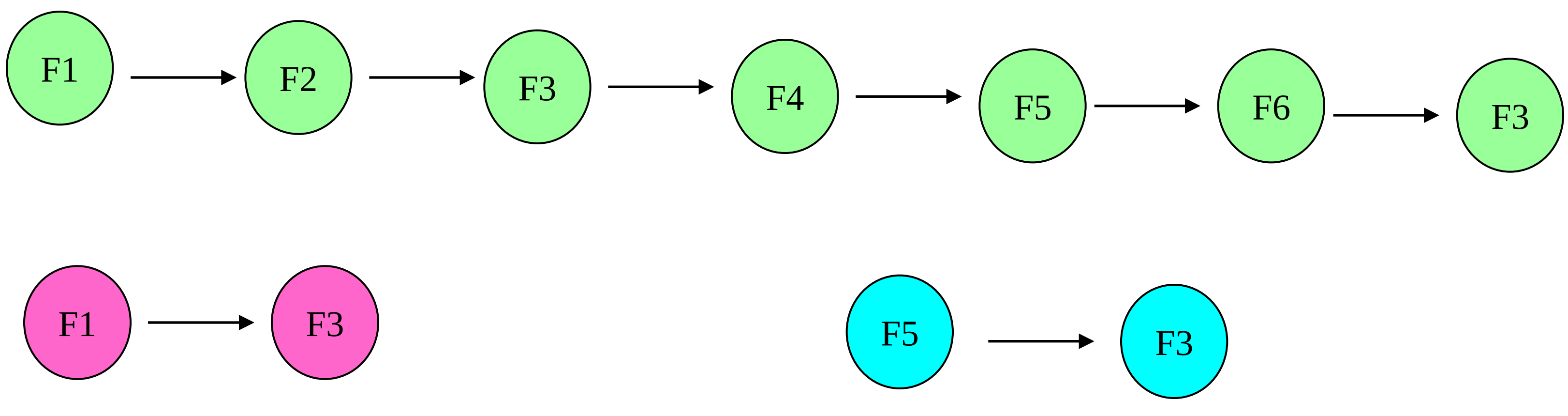
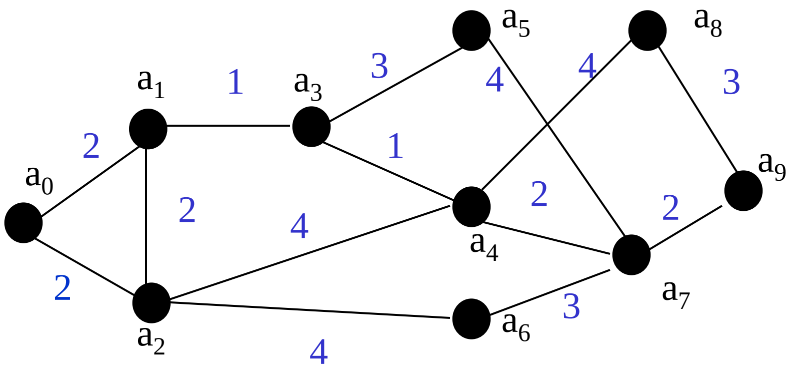
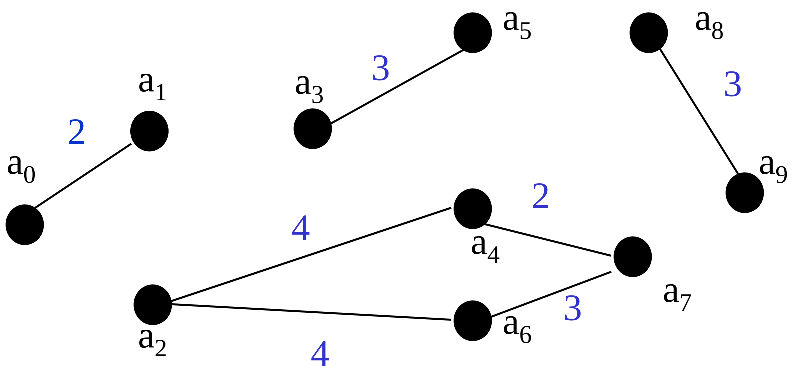


Illustration for clustering

Basic graph



Clusters (a version):
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 }



Initial set A = {1, 2, …, n}, B = A × A (∀ (x, y) such that x, y ∈ A)

Definition. Binary relation R is a subset of B

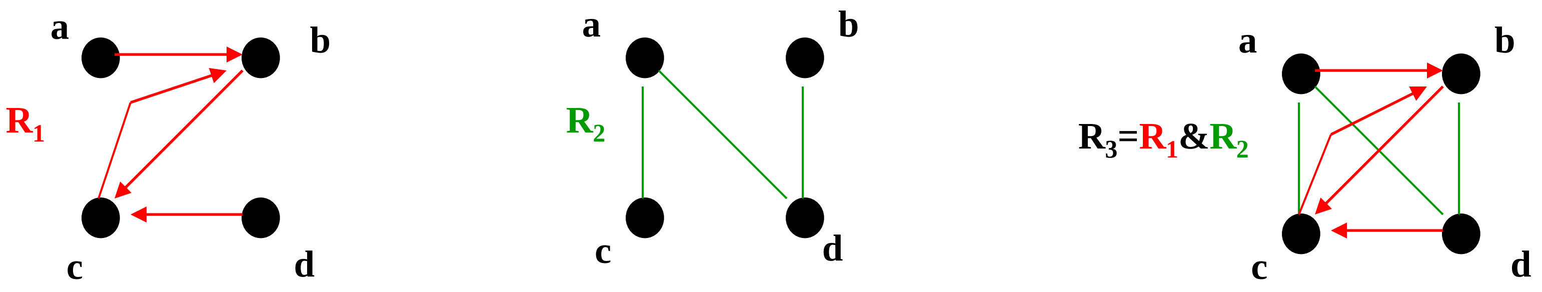
EXAMPLE: A={a, b, c, d}

B = {(a, a), (a, b), (a, c), (a, d), (b, a), (b, a), (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



Binary relations

Context Examples:

1.”Better” (dominance)
2.”Better & Equal” (dominance & equivalence)
3.”Equal” (equivalence)

SOME PROPERTIES:

1.Symmetry: $(x, y) \in R \Rightarrow (y, x) \in R \quad (\forall x \in R, \forall y \in R)$

2.Reflexivity: $(x, x) \in R \quad \forall x \in R$

3.Transitivity: $(x, y) \in R, \ (y, z) \in R \Rightarrow (x, z) \in R \quad (\forall x \in R, \forall y \in R, \forall z \in R)$

APPLICATIONS: *Friendship, *Partnership, *Similarity, *Etc.

Extended models:

1.Weighted binary relations (e.g., power of dominance)
2.K-relations

Prospective usage:

Combinatorial optimization problems on graphs with additional binary relations
(over node/vertices, over edges / arcs, over elements / positions)

LECTURE 4. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.4. Example: joint design of hierarchical system. Communication systems.

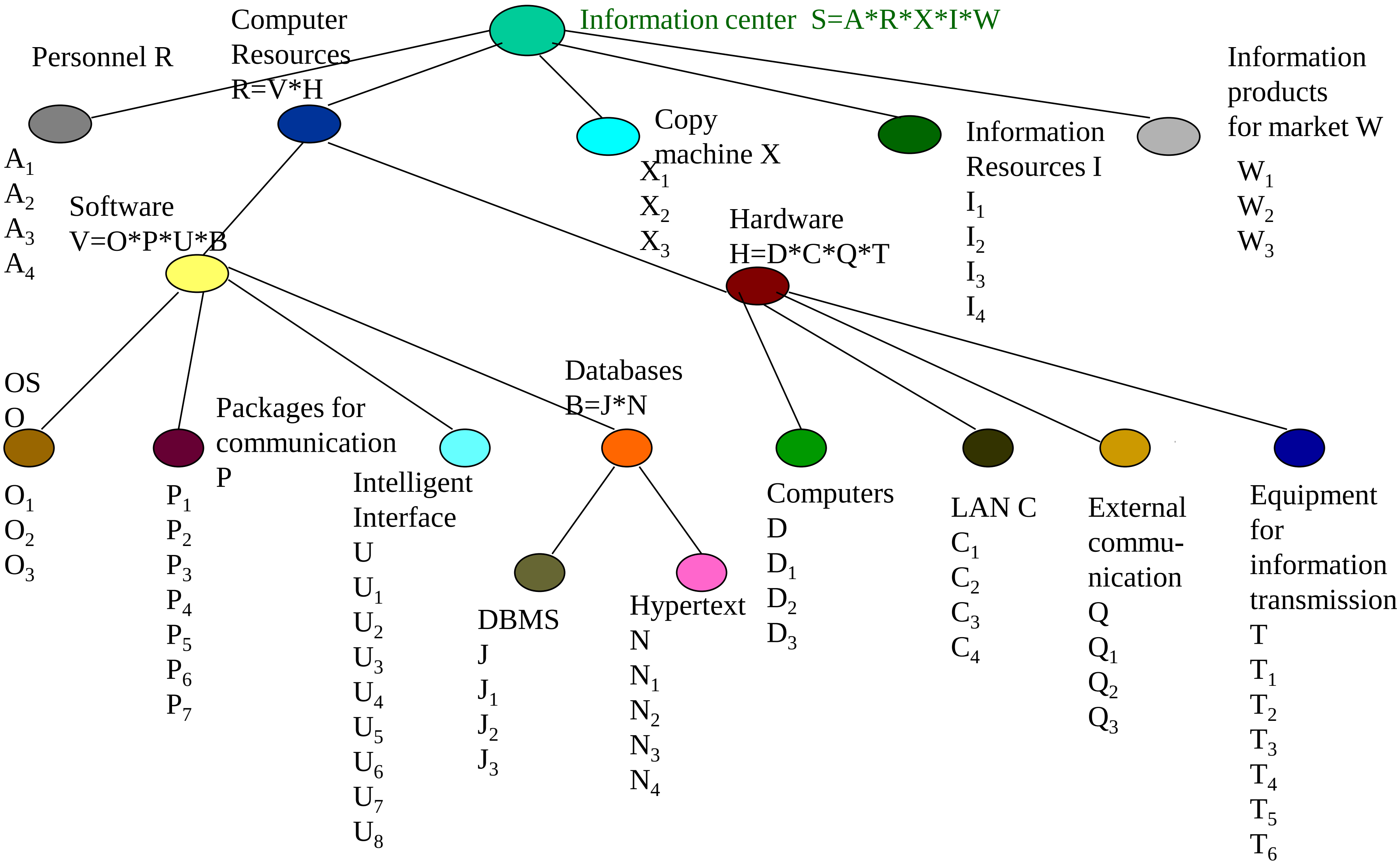
PLAN:

1.Joint design of a hierarchical system : *structural scheme *hierarchical model *generation of alternatives for system parts *generation of criteria for evaluation of alternatives for system parts *multicriteria selection of alternatives *synthesis of a composable system

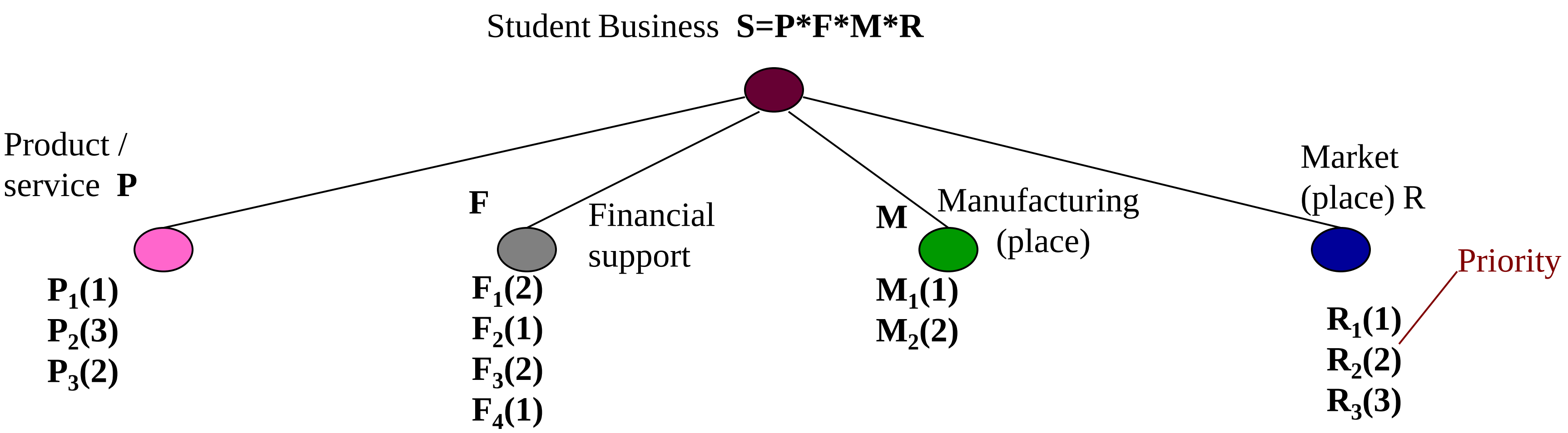
2.Discussion of prospective research directions in communication networks

Sept. 10, 2004

Example of a hierarchical system: information center



Joint design of a hierarchical system: student business



ALTERNATIVES:

P_1 a simple food
P_2 support service (help) for usage of home PC
P_3 special consulting service for searching for personnel in HighTech (for companies, for specialists)
F_1 self-financing
F_2 financing by relatives & friends
F_3 financing by a bank
F_4 financing by a company
M_1 in university
M_2 special office in Dolgoprudny
R_1 Dolgoprudny
R_2 Moscow
R_3 New York

Assessment of alternatives and total priority

	Cost of manufacturing (-)	Volume of to-day's market (+)	Perspectives(+)	Resultant grade	Total priority
P_1	1	5	3	7	**1**
P_2	3	5	2	4	**3**
P_3	4	4	5	5	**2**

	Possible volume(+)	Responsibility(-)	Resultant grade	Total priority
F_1	1	1	0	**2**
F_2	2	1	1	**1**
F_3	5	5	0	**2**
F_4	4	3	1	**1**

	Cost(-)	Usefulness(+)	Resultant grade	Total priority
M_1	1	5	4	**1**
M_2	3	5	2	**2**

	Volume (+)	Possible competition (-)	Distance(-)	Resultant grade	Total priority
R_1	1	0	0	1	**1**
R_2	3	2	1	0	**2**
R_3	3	3	5	-5	**3**

Assessment of compatibility between alternatives

	F_1	F_2	F_3	F_4	M_1	M_2	R_1	R_2	R_3
P_1	5	4	0	0	5	3	5	2	0
P_2	4	5	1	2	5	5	4	5	3
P_3	0	1	4	3	0	5	2	5	5
F_1					5	1	5	3	0
F_2					5	1	5	3	0
F_3					3	5	1	5	5
F_4					3	5	1	5	5
M_1							5	4	5
M_2							3	5	4

NOTE: **5** corresponds to the best level of compatibility
0 corresponds to incompatibility

Resultant composite decisions

$S_1 = P_1 * F_2 * M_1 * R_1$; vector of quality $N(S_1) = (4; 4,0,0)$;
Description: (a) all elements are at the top level (1); (b) grades of compatibility are (5,5,5,5,5,4)

Another possible prospective decision is:
$S_2 = P_3 * F_4 * M_1 * R_2$; vector of quality $N(S_1) = (4; 2,2,0)$;
Description: (a) levels of elements are (1,1,2,2); (b) grades of compatibility are (3,3,4,5,5,0)
Note: the decision is infeasible by compatibility

Bottlenecks (possible problems for improving the initial situation):

1. **compatibility (P_3,M_1) equals 0 (=> to increase)**
2. **compatibility (F_4,M_1) equals 3 (=> to increase)**
3. **compaibility (M_1,R_2) equals 3 (=> to increase)**
4. **priorityity of P_3 (=> to improve)**
5. **priority of R_2 (=> to improve)**

Prospective research directions in communication systems (the list is under extension)

1. General design of communication network

2. GRID-like network environment (GRID-computing, GRID-communication)

3. Extension of communication network:
   - 3a. Improvement of an existing communication network
   - 3b. Extension of an existing communication network via additional territory

4. Allocation of resources in communication networks
   (applied situations, problem, models, and approaches)

5. Frequence assignment / allocation in communication networks
   (applied situation, problems, models, and approaches)

6. Reliability issues for communication networks
   - 6a. Evaluation of reliability
   - 6b. Design of reliable communication networks

7. Routing problems

Prospective research directions in communication systems (the list is under extension)

8.System testing in communication networks:
probing problem

9.Maintenance in communication networks

10.Mobile communication
11a.Movement of users
11b.Movement of all components of the system

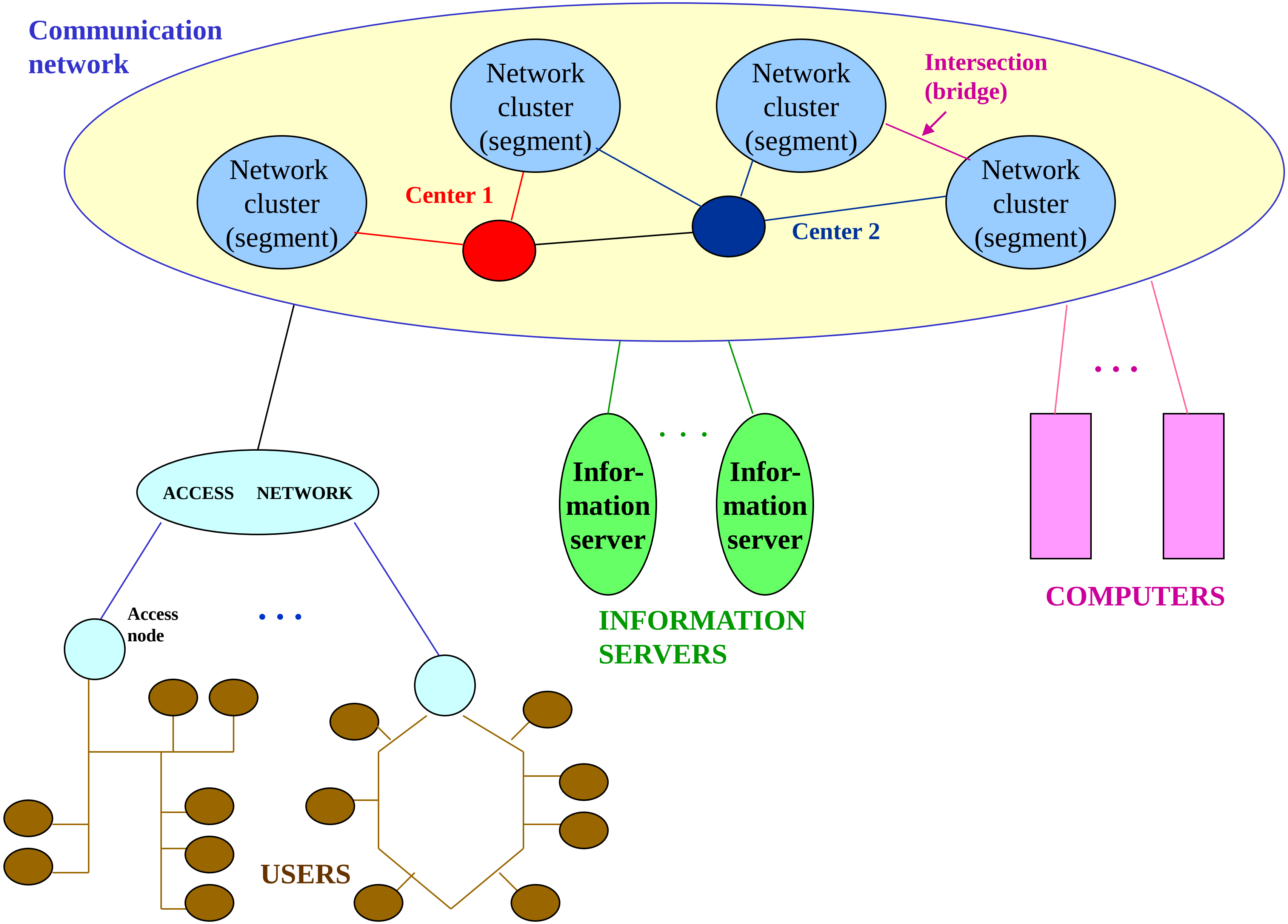
11.System for information compression (algorithm part)

12.Communication networks & free scale networks

13.Planning of access to information/computer resources
(information bases on servers or computers)
in distributed information/computer environment
(users, communication network, and
a set of information servers/computer)

14.Design of topology for communication networks

Communication system: a scheme



LECTURE 5-6. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.5. Information technology. Human participation.

L.6. Schemes of design processes, Design problems.

PLAN:

1.Information technology and its properties

2.Organizational-engineering systems. Human participation (in systems, in design)

3.Design frameworks (series process, cascade-like process). Close frameworks for information processing

4.Main design problems (design, redesign / upgrade process, multistage design,

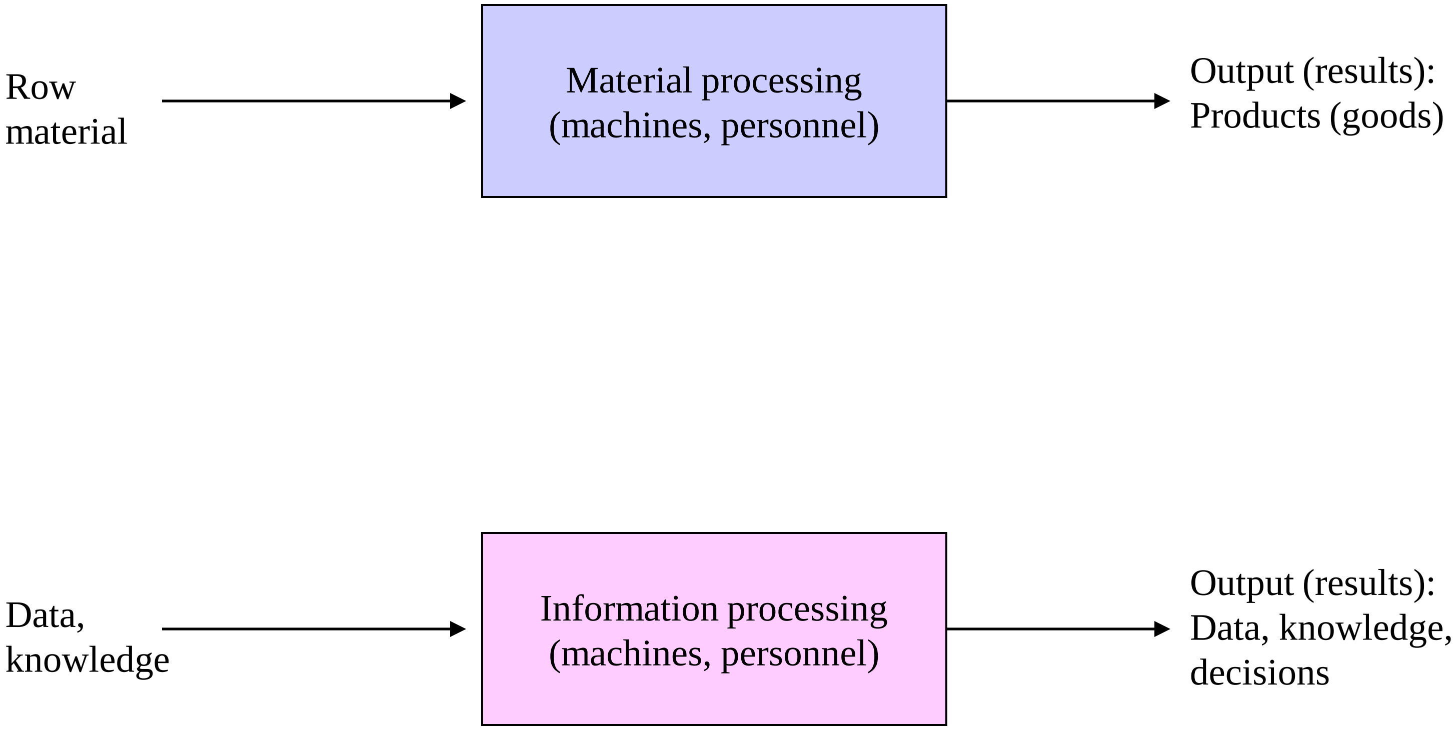
system evaluation, revelation of bottlenecks, system evolution / development)

Sept. 11, 2004

Information technology: structure

	R&D	Manufacturing	Testing	Marketing	Utilization/Maintenance	Recycling/reusing
HARDWARE PART *VLSI *computers *communication			●●●			●
SOFTWARE PART *oper. systems *DBMSs *communication soft.						
MATH./ALG. PART *math. models *algorithms	●●					
INFORMATION PART *data *knowledge	●	●			●	●●
ORG. PART *specialists *users *HCI *group work		●				●
APPL. SYSTEMS *MISs *DSSs & ESs *etc.					●	

“Processing” conveyor

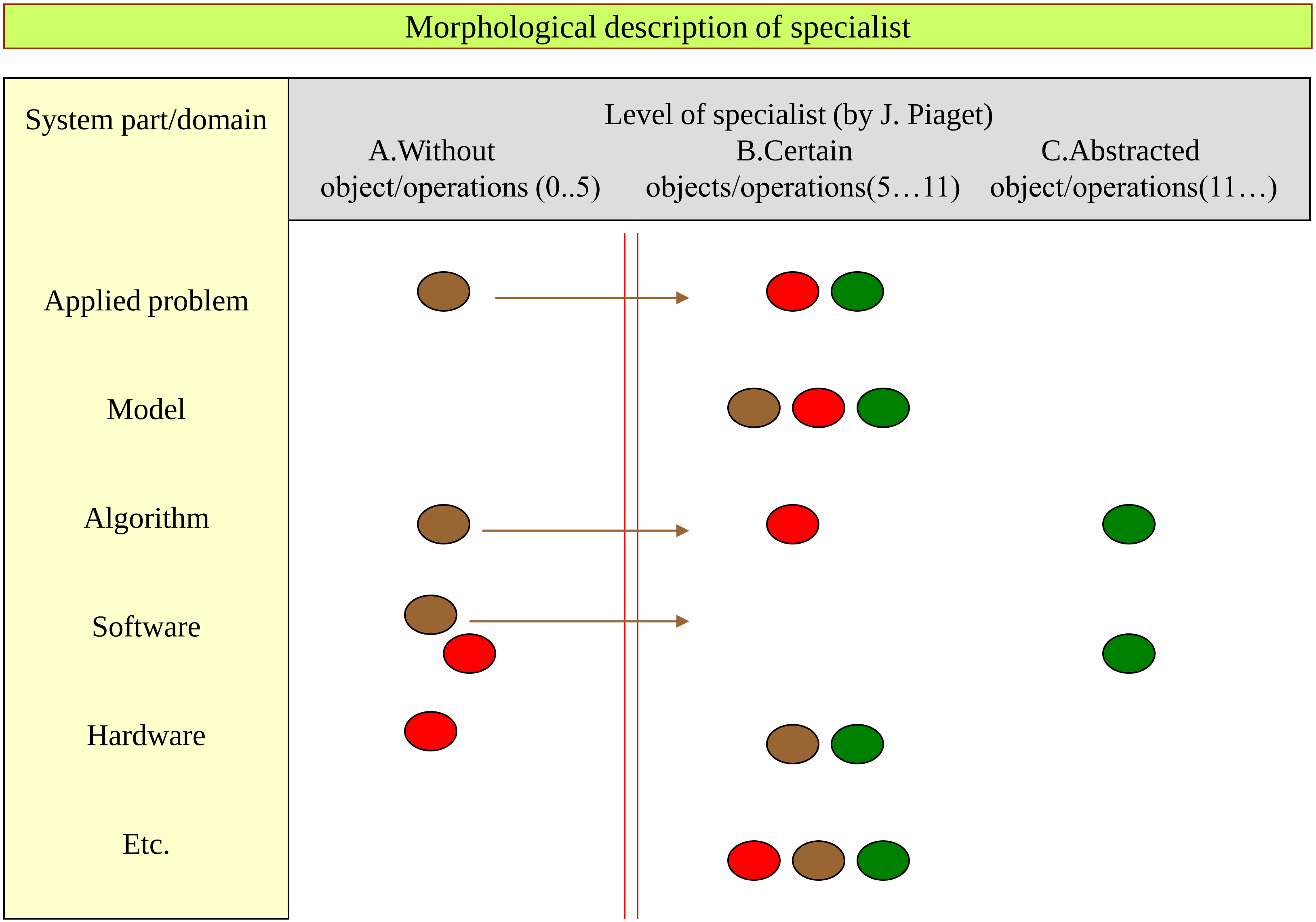


Comparison: material processing & information processing

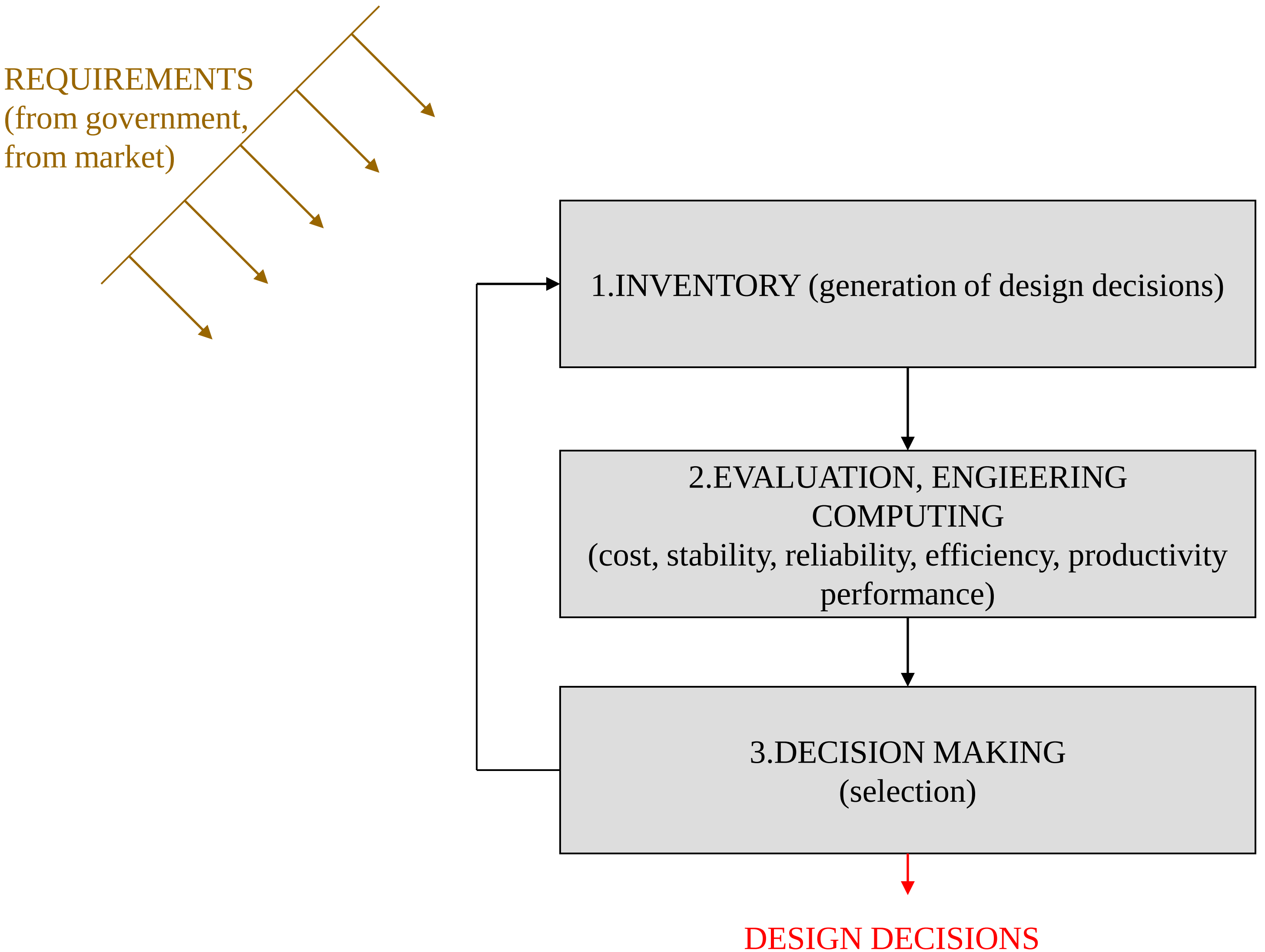
STAGES	Technology for trees	Information technology
Source of row materials	Forest	1.Books, news papers 2.Data & knowledge bases 3.People
Row materials	Woods	1.Data 2.Knowledge
Transportation	Cars, trains	Communication systems
Manufacturing *machines *personnel	Machines Engineers, Workers	Computers, software, communication 1.Specialists 2.Users
Output	Boards, etc.	1.Data 2.Knowledge 3.Decisions
Keeping	Depository	1.Data bases 2.Knowledge bases
Users	Firm for building, Private persons	1.Government 2.Firms 3.University & educational systems 4.Research Institutes & Universities 5.Private persons

Properties of information technology

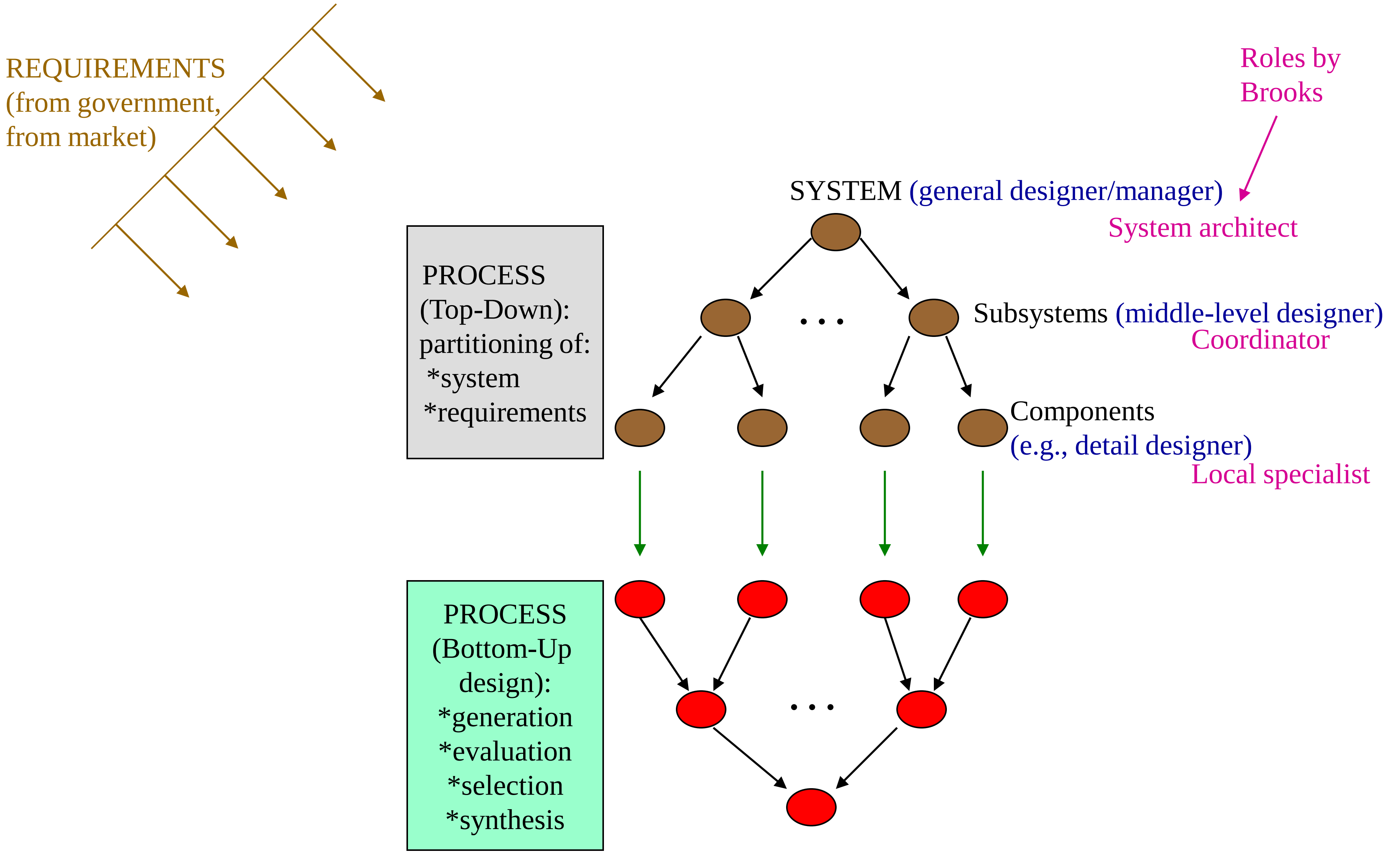
1.Various kinds of sources: *statistics, books, data bases *specialists, population
2.Preservation of initial information & possibility for re-processing
3.Possibility for parallel / concurrent processing
4.Possibility for usage of many different methods
5.Possibility to accumulate results (outputs)
6.High "ecologiability"
7.High requirements to professional skills
8.Unique role of human
9.High requirements to information presentation (e.g., visualization)
10.Integration:
 *exact science
 *engineering
 *psychology
 *education/training
 *art (e.g., TV, cinema)
11.Wide range of users:
 *science
 *industry
 *management, economics
 *education
 *art
 *private life


Morphological description of specialist
System part/domain
Level of specialist (by J. Piaget)
A.Without object/operations (0..5)
B.Certain objects/operations(5…11)
C.Abstracted object/operations(11…)
Applied problem
Model
Algorithm
Software
Hardware
Etc.

Series design flow (J.R. Dixon)



Cascade-like design flow



Levels of creativity (by G. Altshuller)

LEVEL 1. Usage of a well-known object (product, technology, decision, etc.)

LEVEL 2. Searching for & selection of the best object

LEVEL 3. Improvement (modification) of an object

LEVEL 4. Design of a new object

LEVEL 5. Design of a system of objects

Illustration of creativity levels for processing

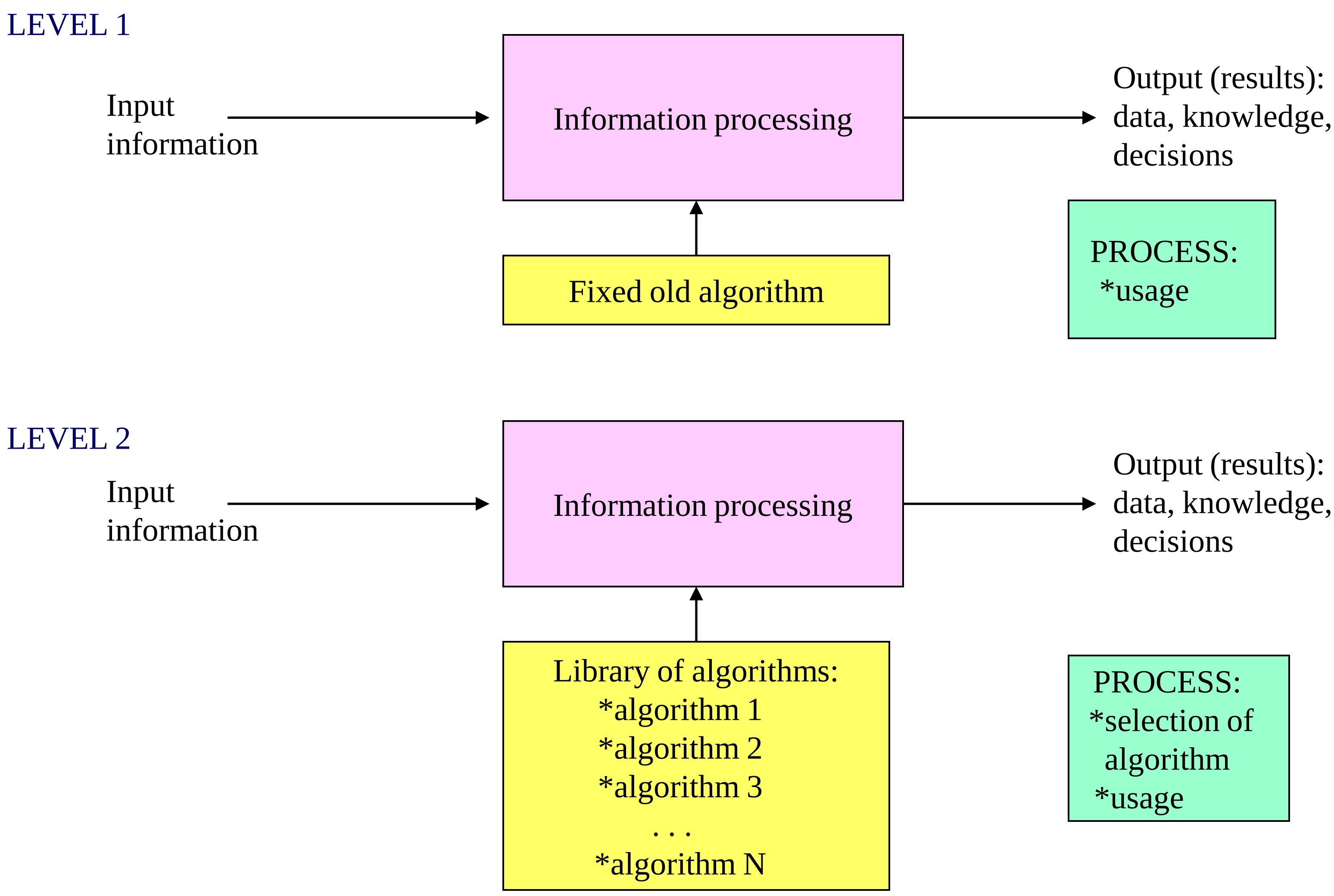


Illustration of creativity levels for processing

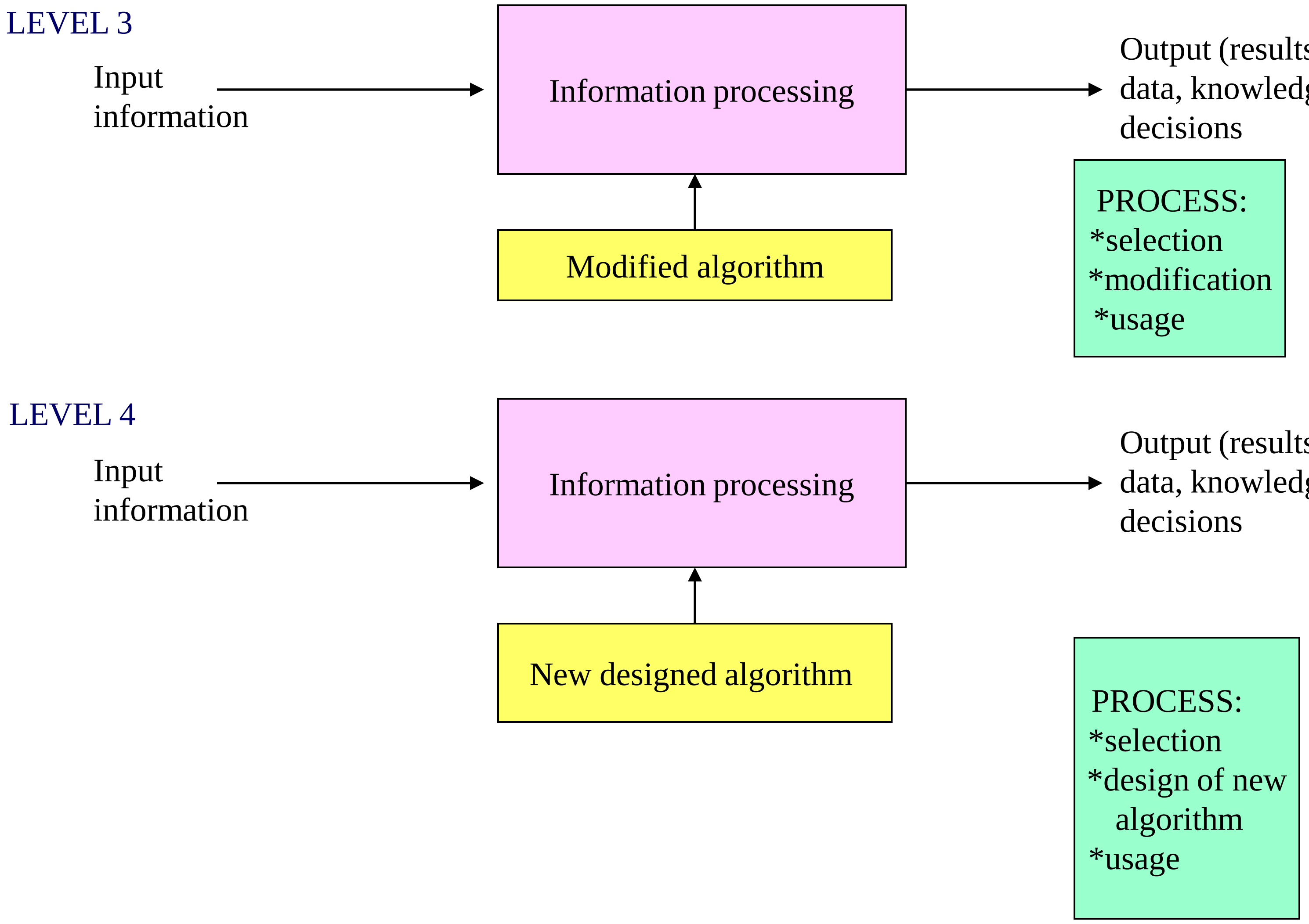
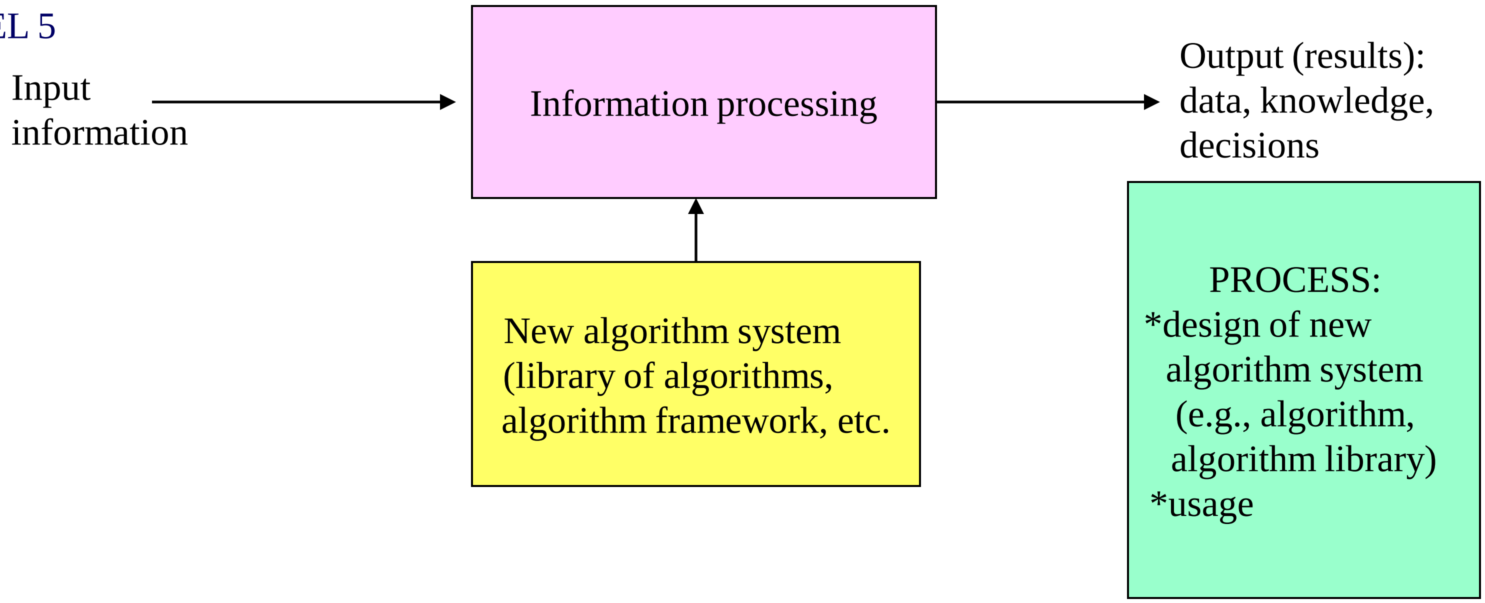


Illustration of creativity levels for processing

LEVEL 5



Design problems (technological problems)

1. Design

2. Redesign (improvement, upgrade process)

3. Multistage design

4. Evaluation

5. Revelation of bottlenecks

6. Modeling of system evolution /development (& forecasting)

LECTURE 7. Course: “Design of Systems: Structural Approach”

Dept. “Communication Networks &Systems”, Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.7.Concurrent engineering and life cycle. Traditional hierarchical system design.

PLAN:

1.Concurrent engineering (concurrent processes, concurrency in life cycle)

2.Hierarchical system design

3.Some examples of applied systems (e.g., manufacturing systems)

4.Some problems in mobile communication systems

Sept. 17, 2004

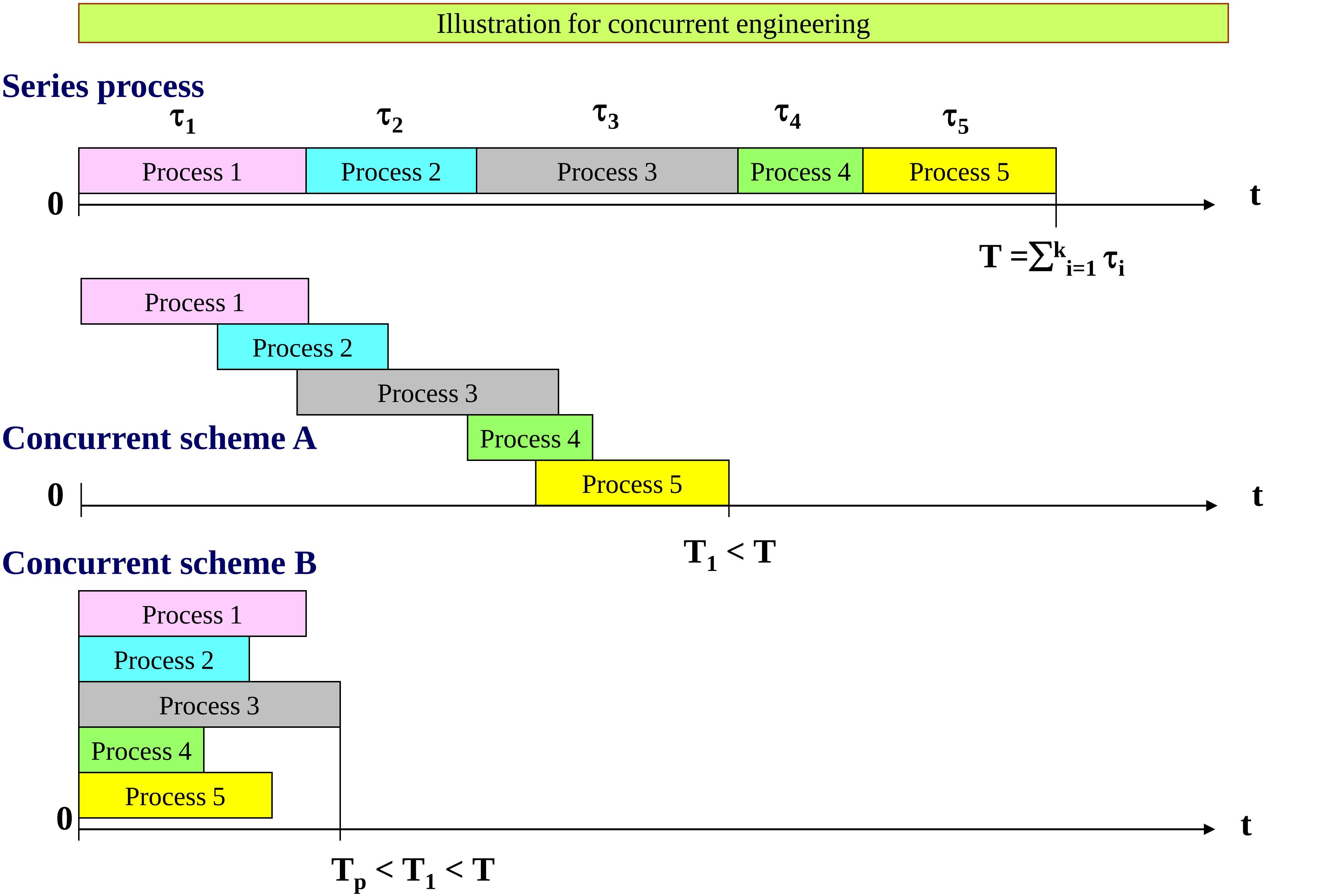
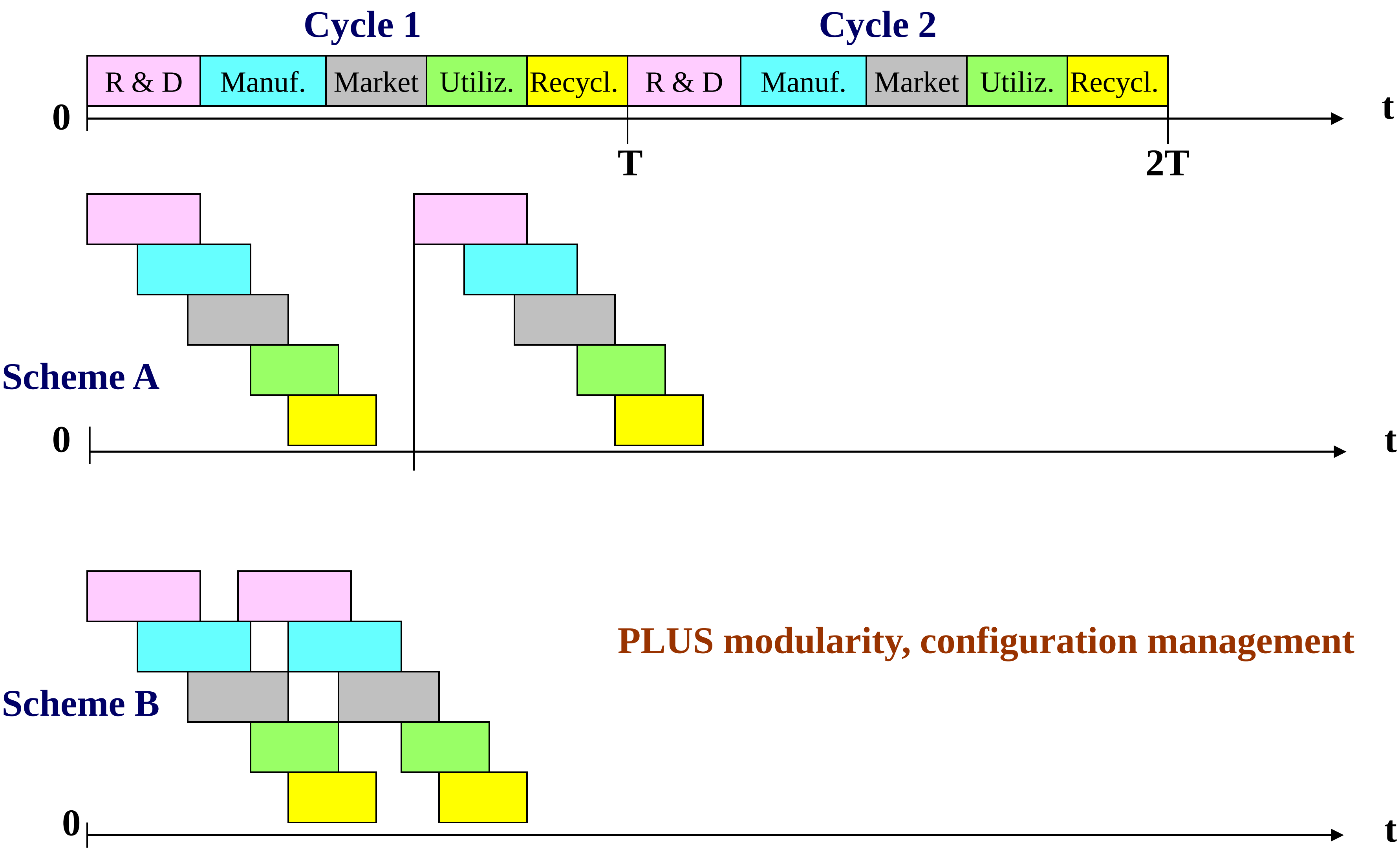


Illustration for concurrent engineering
Series process
τ1
τ2
τ3
τ4
τ5
Process 1
Process 2
Process 3
Process 4
Process 5
0
t
T =∑k i=1 τi
Process 1
Process 2
Process 3
Concurrent scheme A
Process 4
Process 5
0
t
T1 < T
Concurrent scheme B
Process 1
Process 2
Process 3
Process 4
Process 5
0
t
Tp < T1 < T

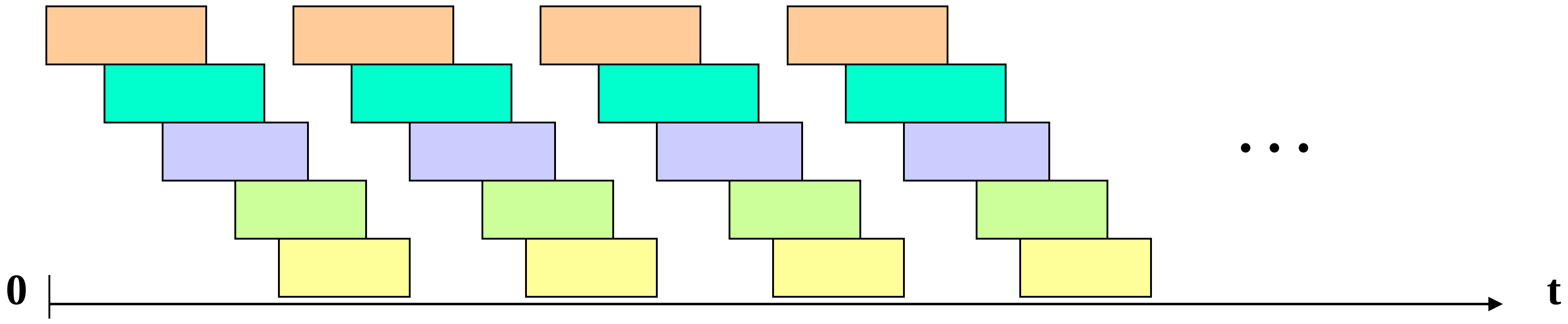
Concurrency in life cycle



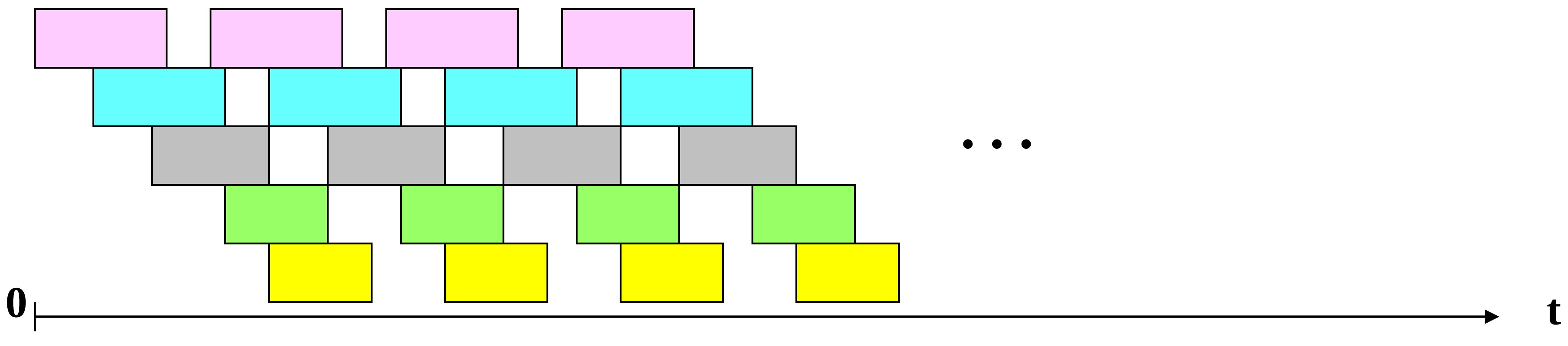
Concurrency in life cycle for several products

ORGANIZATIONAL DIRECTION:
concurrency, modularity, configuration management, and coordination for products A, B, etc.

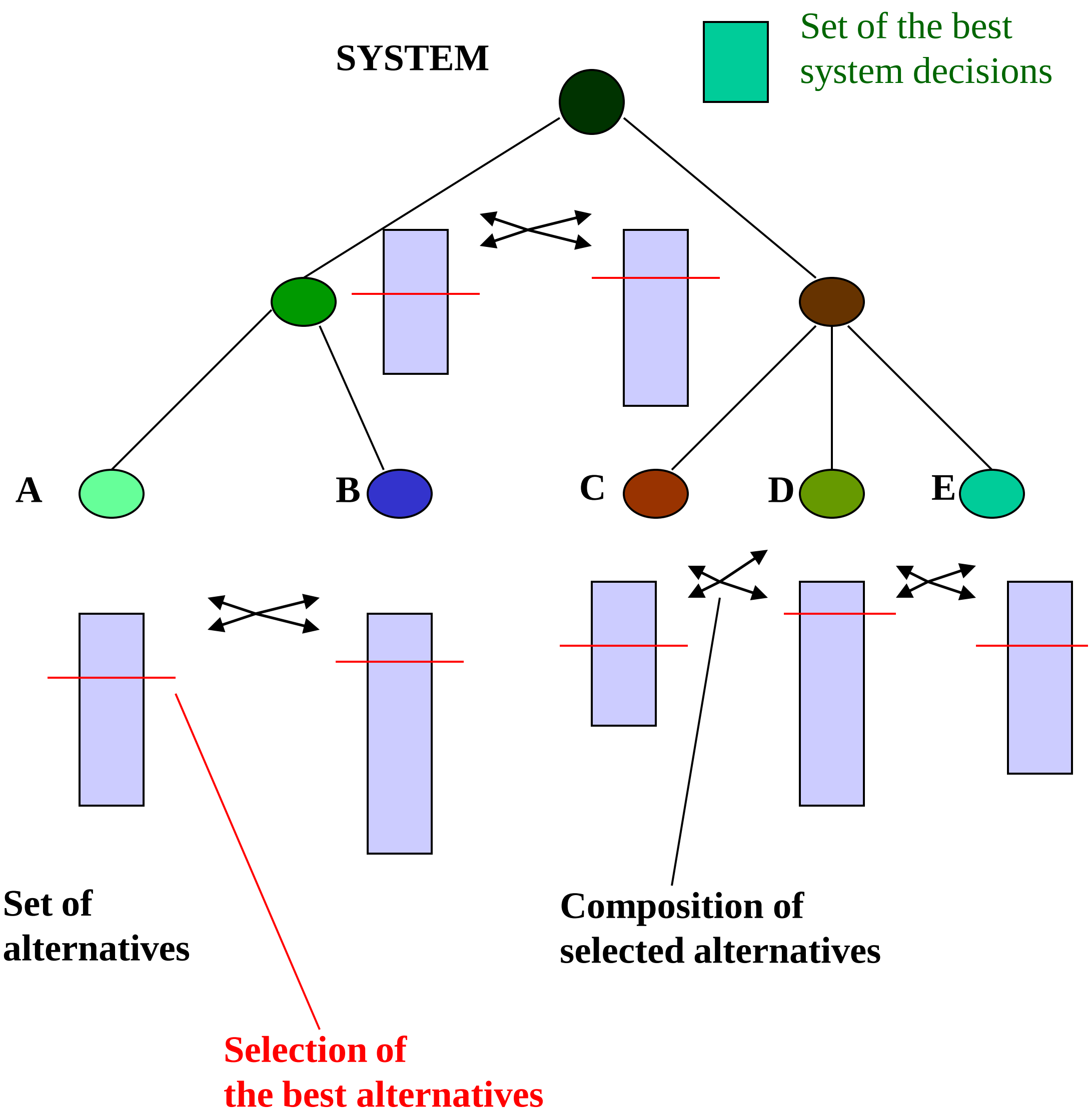
Product A



Product B



Hierarchical system design



BOTTOM-UP PROCESS

Selection process:

1.Contsraints
2.Multicriteria selection (ranking)
3.Optimization approach
4.Expert judgment

Main approaches to system design

1. Multidisciplinary optimization (e.g., aerospace engineering, structural engineering)
2. Mixed integer non-linear programming (e.g., chemical engineering, process, systems engineering)
3. Non-linear multicriteria (multiobjective) optimization including evolutionary multiobjective optimization
4. Formal methods in design (e.g., mechanical engineering)
5. Global optimization methods
6. Grammatical design (i.e., grammar description of decomposable systems)
7. Methods of artificial intelligence (knowledge-based systems, neural networks, genetic algorithms, etc.) (e.g., computer engineering, VLSI design)
8. Parameter Space Investigation PSI (e.g., mechanical engineering, nuclear engineering)
9. Hierarchical system design (traditional engineering organizational methods, modular design methods, combinatorial synthesis)
10. Hybrid methods

LECTURE 8-9. Course: “Design of Systems: Structural Approach”

Dept. “Communication Networks &Systems”, Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.8. Principles of system analysis. Paradigm of decision making. Basic DM problems.

L.9. Kinds of scales, Pareto-efficient decisions, system evaluation, hierarchy of requirements.

PLAN:

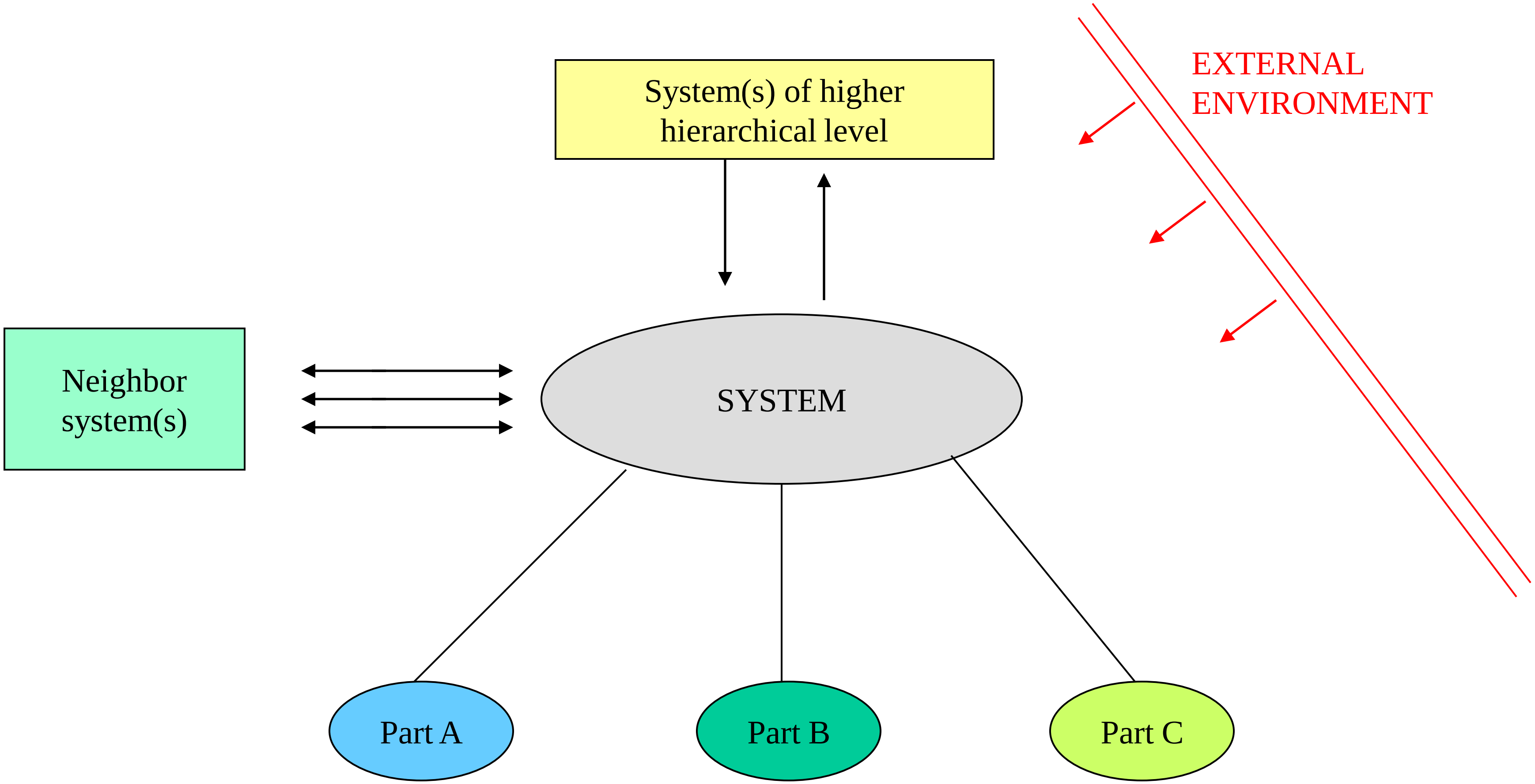
1.Priciples of system analysis. 2.Paradigm of decision making. 3.Basic decision making problems.

4.Kinds of scales. 5.Pareto-effective decisions 6.Evaluation of systems

7.Hierarchy of requirements / criteria 8.Roles in decision making process. Example

Sept. 18, 2004

A scheme for a system



Principles of system analysis

1. Examination of life cycle (e.g., R & D, manufacturing, testing, marketing, utilization & maintenance, recycling)
2. Examination of system evolution / development (i.e., dynamical aspects)
3. Examination of interconnection with environment (nature, community, other systems)
4. Examination of interconnection among system parts / components (physical parts, functions, information, energy, etc.)
5. Analysis of system changes (close to principle 2)
6. Revelation and study of main system parameters
7. Integration of various methods (decomposition, hierarchy, composition, etc.)
8. Investigation of main system contradictions (engineering, economics, ecology, politics, etc.)
9. Integration of various models and algorithms (e.g., physical experiments, mathematical modeling, heuristics, expert judgment)
10. Interaction among specialists from different professional domains and hierarchical levels (engineering, computer science, mathematics, management, social science, etc.)

Decision Making Paradigm (stages) by Herbert A. Simon

1.Analysis of an applied problem (to understand the problem: main contradictions, etc.)

2.Structuring the problem:

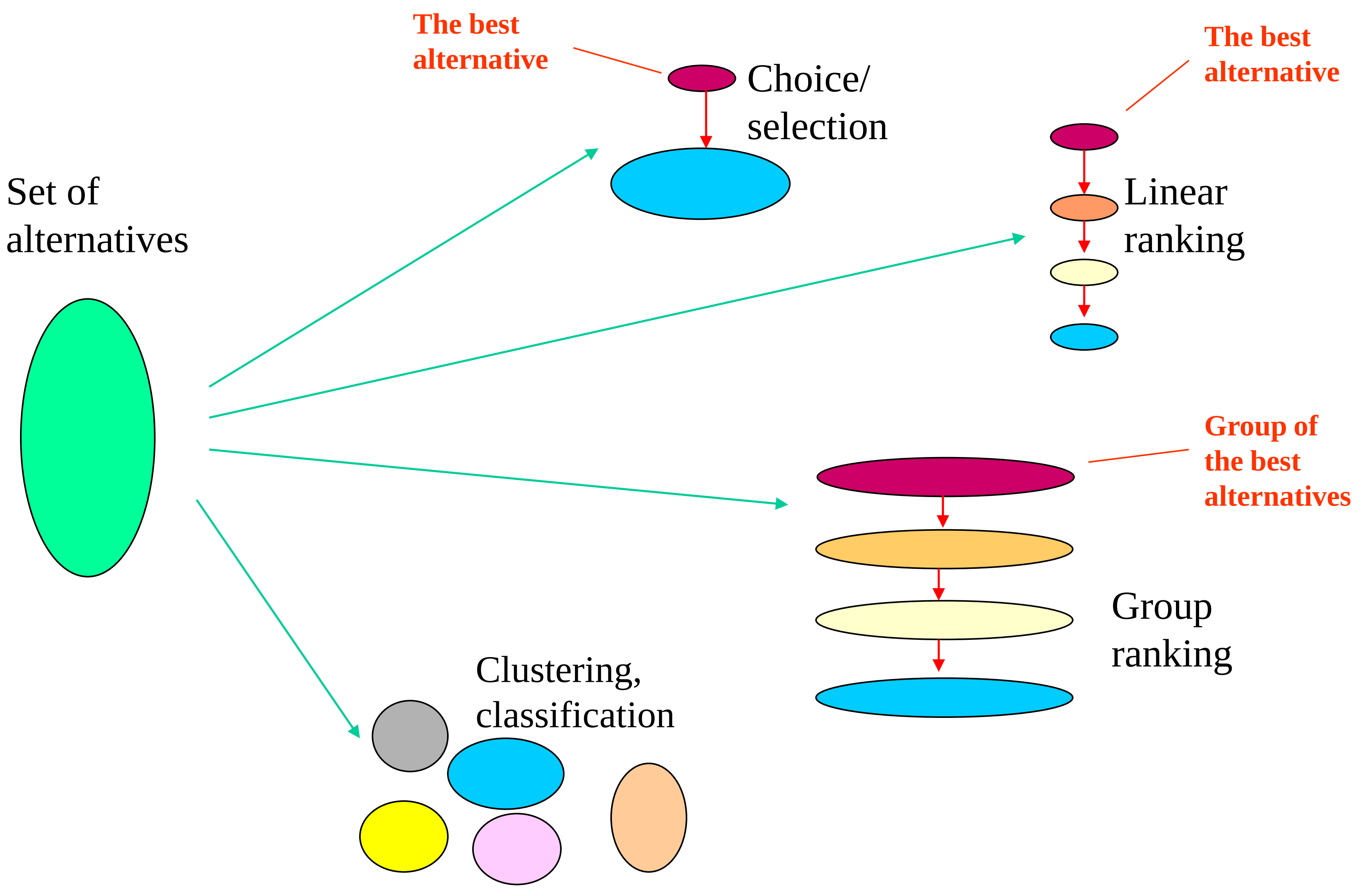
- **2.1.Generation of alternatives**
- **2.2.Design of criteria**
- **2.3.Design of scales for assessment of alternatives upon criteria**

3.Evaluation of alternatives upon criteria

4.Selection of the best alternative (s)

5.Analysis of results

Four Basic Decision Making Problems



Kinds of Problems by Herbert A. Simon

I.STANDARD PROBLEMS

II.FORMULIZED PROBLEMS
(models in mathematics as equations, optimization, etc.)

Decision Making Problems

III.ILL-STRUCTURED PROBLEMS
*human factors, information from expert(s) & decision maker(s)
*uncertainty

IV.FORECASTING (decisions for the future)

Applied Decision Making Problems

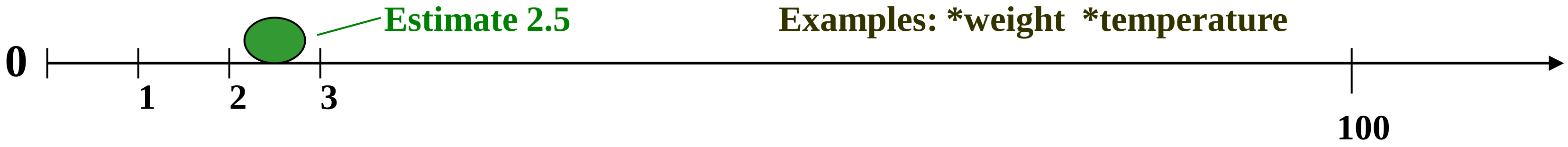
1.LEVEL OF GOVERNMENT:
***selection of research projects**
***investment into infrastructure (e.g., transport, communication, education)**
***selection of political decisions**

2.LEVEL OF COMPANY:
***selection of product**
***selection of market**
***selection of personnel**
***selection of partners**
***selection of place for new plants, etc.**

3.LEVEL OF PRIVATE LIFE:
***selection of apartment**
***selection of university / college**
***selection of car**
***selection of bank program**
***selection of place for vacation, etc.**

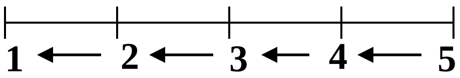
Kinds of Scales

1. Quantity: quantitative scale

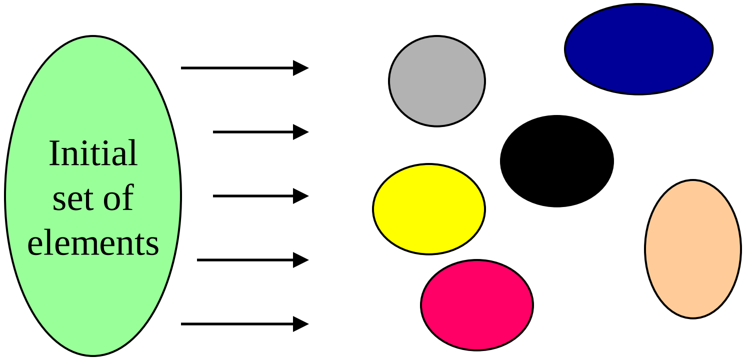


2.Quality: qualitative scales (levels, ordering, class)

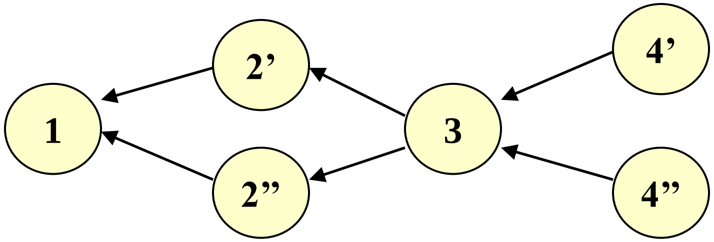
2a.Ordinal scale



2b.Nominal scale (for classes, clusters)



2c.Scale as partial order (generalization)



Description of decision making problem

Alternatives $A=(A_1, \dots, A_i, \dots, A_n)$ and criteria $C=(C_1, \dots, C_j, \dots, C_k)$,
$\forall A_i$ a vector of estimates $z_i = (z_{i1}, \dots, z_{ij}, \dots z_{ik})$

Matrix of estimates is:

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

Our goal is to get a "priority" for each alternatives: $P(A_i)$

Evaluation of $P(A_i)$ can be based on the following:

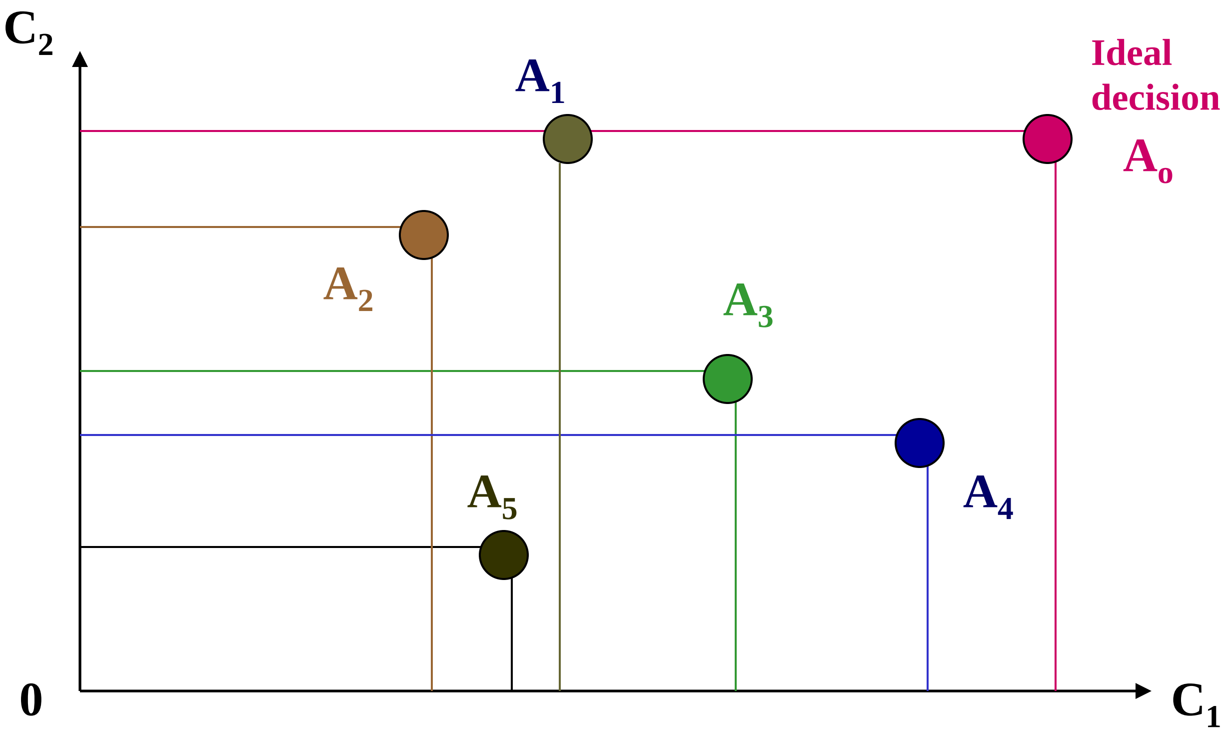
1. **Quantitative scale**
2. **Ordinal scale**
3. **Scale as partial order**

Pareto-effective (Pareto-optimal) decisions

PARETO RULE:

Alternative $X=(x_1, \dots, x_j, \dots, x_k)$ and alternative $Y=(y_1, \dots, y_j, \dots, y_k)$,

X is better than Y if $\forall j \quad x_j \geq y_j$ and $\exists i \ (1 \leq i \leq k)$ such that $x_i > y_i$



A_1 better A_2

A_3 better A_5

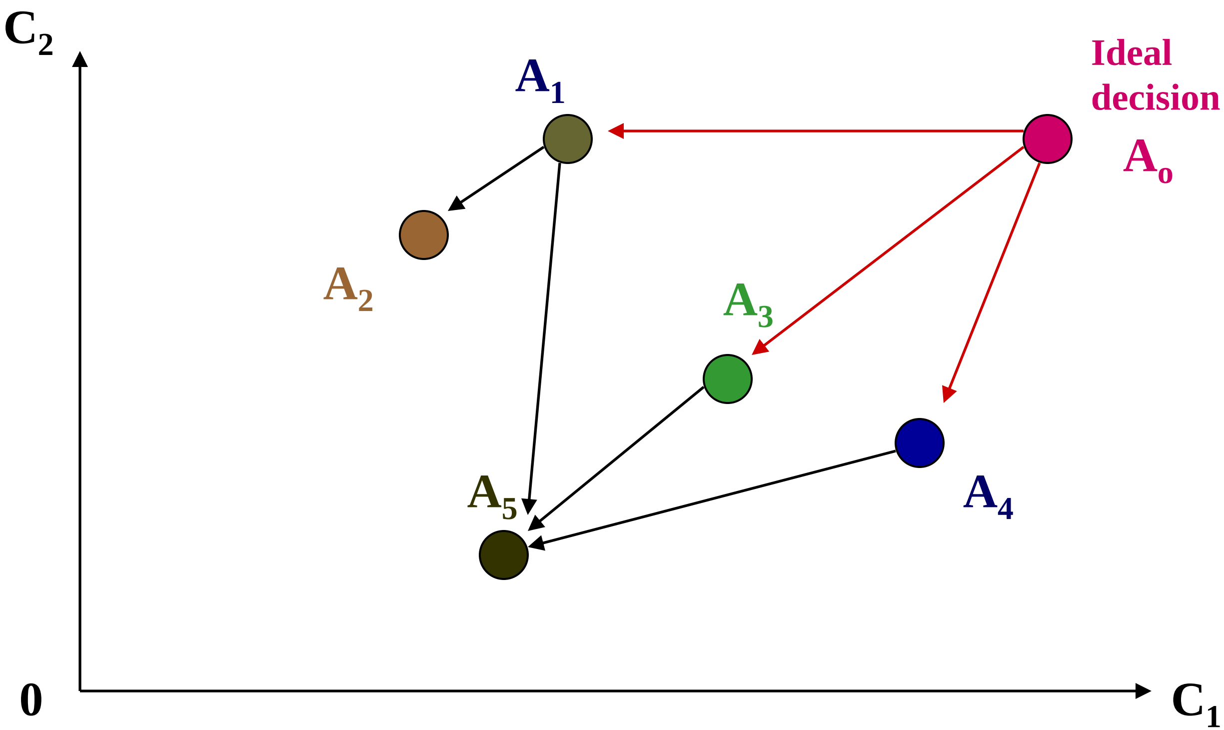
A_4 better A_5

A_1 better A_5

A_1, A_3, A_4 are incomparable and have no dominating elements (only A_o)

A_1, A_3, A_4 are Pareto-effective decisions for set $\{A_1, A_2, A_3, A_4, A_5\}$

Partial order on alternatives



A_1 better A_2
A_3 better A_5
A_4 better A_5
A_1 better A_5

A_1 , A_3, A_4 are incomparable and have no dominating elements (only A_o)

A_1 , A_3, A_4 are Pareto-effective decisions for set {A_1, A_2, A_3, A_4, A_5}

Evaluation of a complex system can be based on the following:

1.Quantitative scale

2.Ordinal scale

3.Scale as partial order (including special discrete spaces)

Hierarchy of requirements / criteria

1. **Ecology, politics**

2. **Economics, marketing**

3. **Technology (e.g., manufacturing issues, maintenance issues)**

4. **Engineering**

Main roles in decision making process

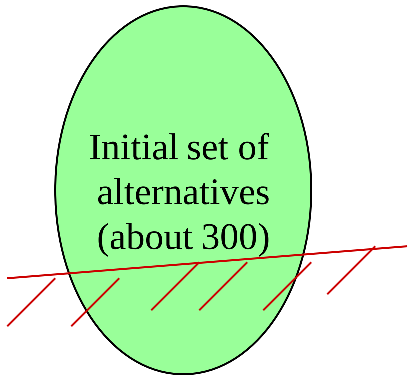
1.DECISION MAKER (DM)
(to make the resultant decision, to evaluate alternatives, etc.)

2.SUPPORT SPECIALIST
(to organize the decision making procedure including support of all stages)

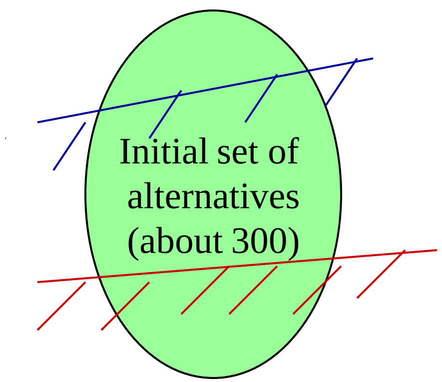
3.EXPERT(s) (to evaluate alternatives)

Phases of decision making process: Example for selection for the best company (P. Humphreys)

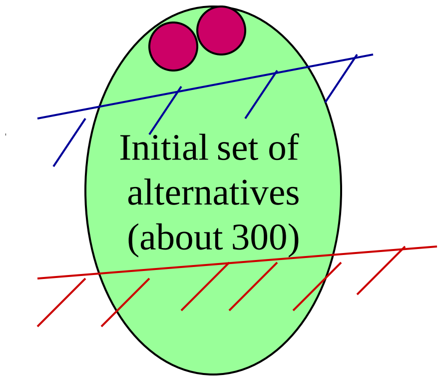
Phase 1. Analysis of initial requests (i.e., alternatives) & deletion of the worst material (about 1/3 of the requests)



Phase 2. Design of a special method for multicriteria selection, evaluation of alternatives upon criteria, selection of a group (a part) of the best alternatives (about 20…30) (group of experts)



Phase 3. Choice of the best alternative(s) (about 1…3): special procedure of expert judgment (group of Decision Makers)



LECTURE 10. Course: “Design of Systems: Structural Approach”

Dept. “Communication Networks &Systems”, Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.10. Multicriteria decision making.

PLAN:

1.Multicriteria decision making: *utility function, * method of pair comparison,

*method of incomparability (“equivalence”) levels, *outranking technique (ELECTRE), *AHP, etc.

2.Integration of results obtained on the basis of several methods (or subsystems of criteria)

Sept. 24, 2004

Examples of utility functions

Alternatives $A=(A_1, \dots, A_i, \dots, A_n)$ and criteria $C=(C_1, \dots, C_j, \dots, C_k)$,
$\forall A_i$ a vector of estimates $z_i = (z_{i1}, \dots, z_{ij}, \dots z_{ik})$, μ_j is a weight of criterion j

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

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

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

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

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

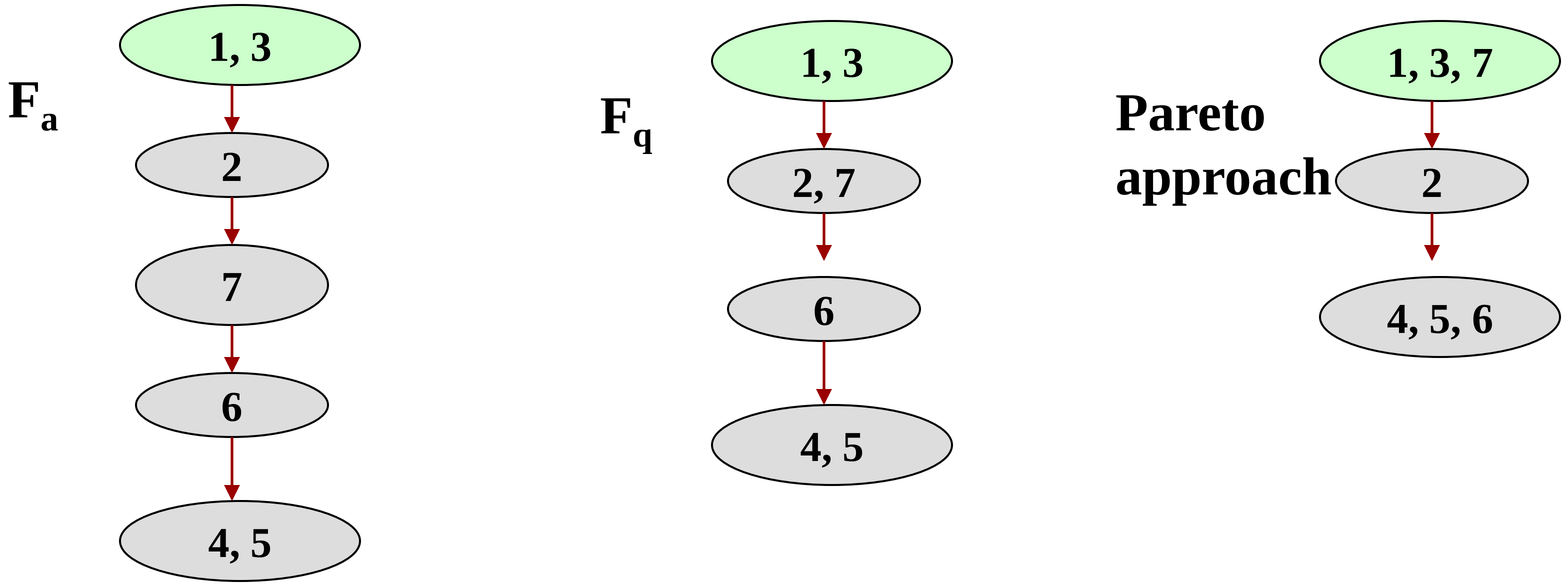
General $F_o = \sum_{j=1}^{k} \mu_j \; \varphi \; (z_j / z_{jb})$

where φ is a differential function,

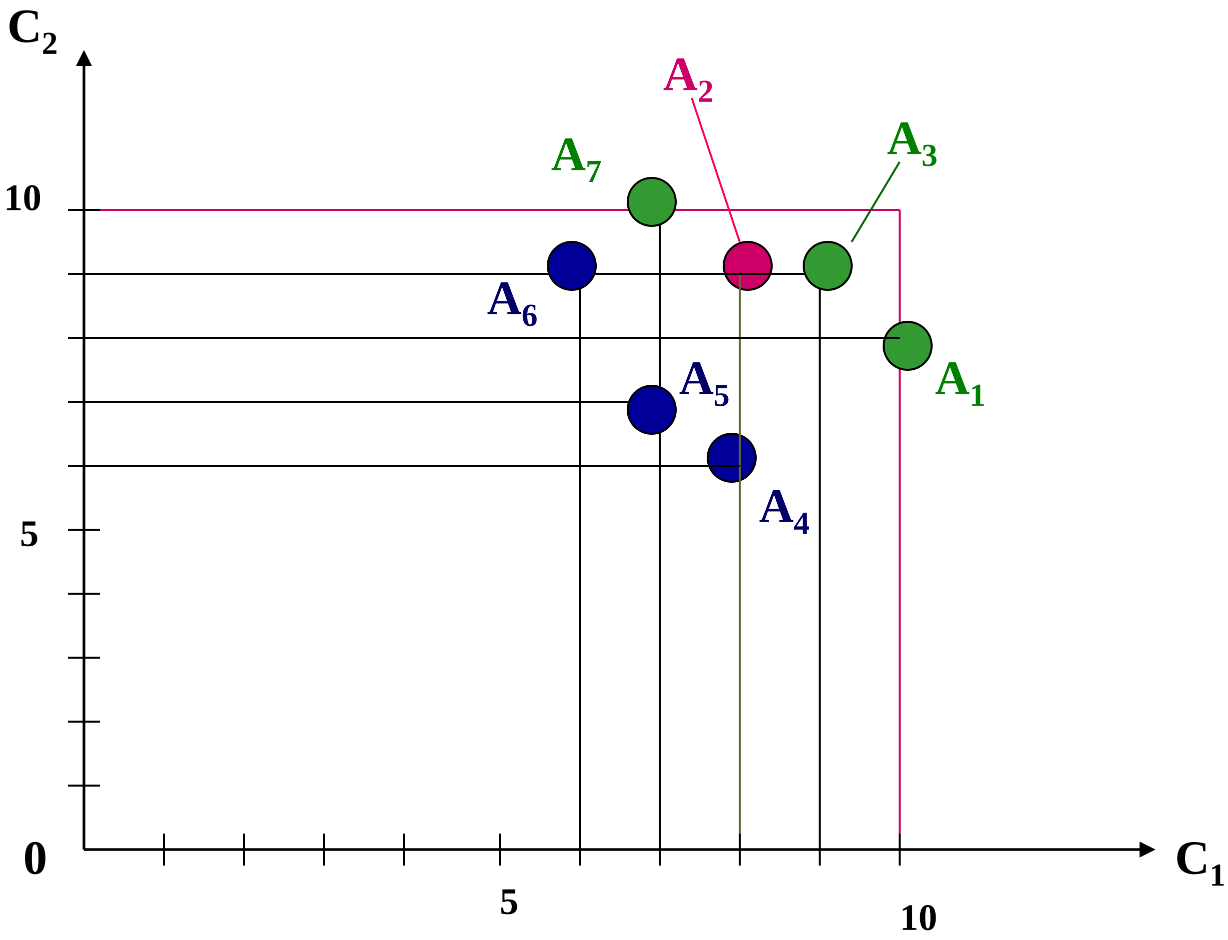
z_{jb} is an estimate-base

Illustrative numerical example for multicriteria ranking

	Mathematics C_1	Sport C_2	F_a	F_q	Pareto approach
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

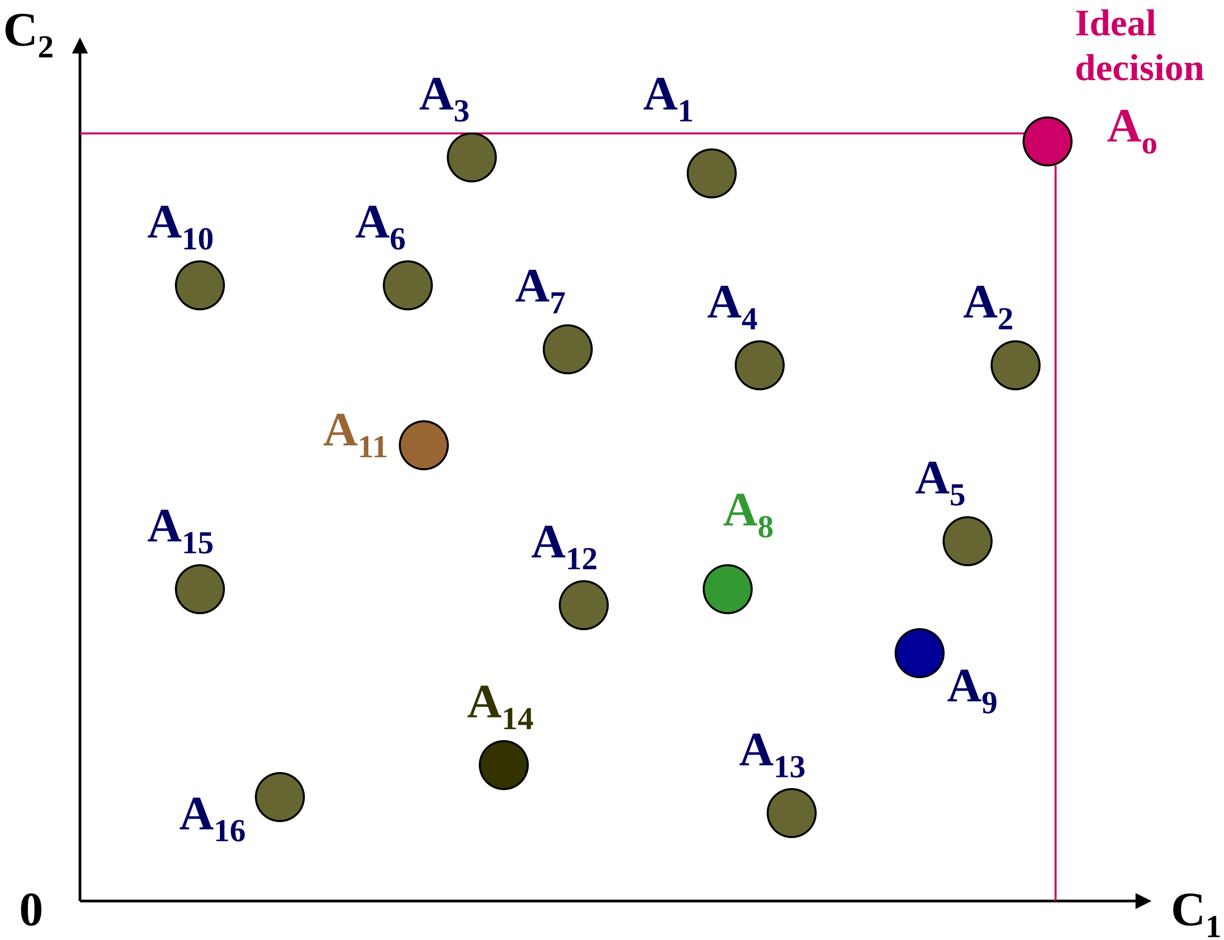


Pareto-approach for example above

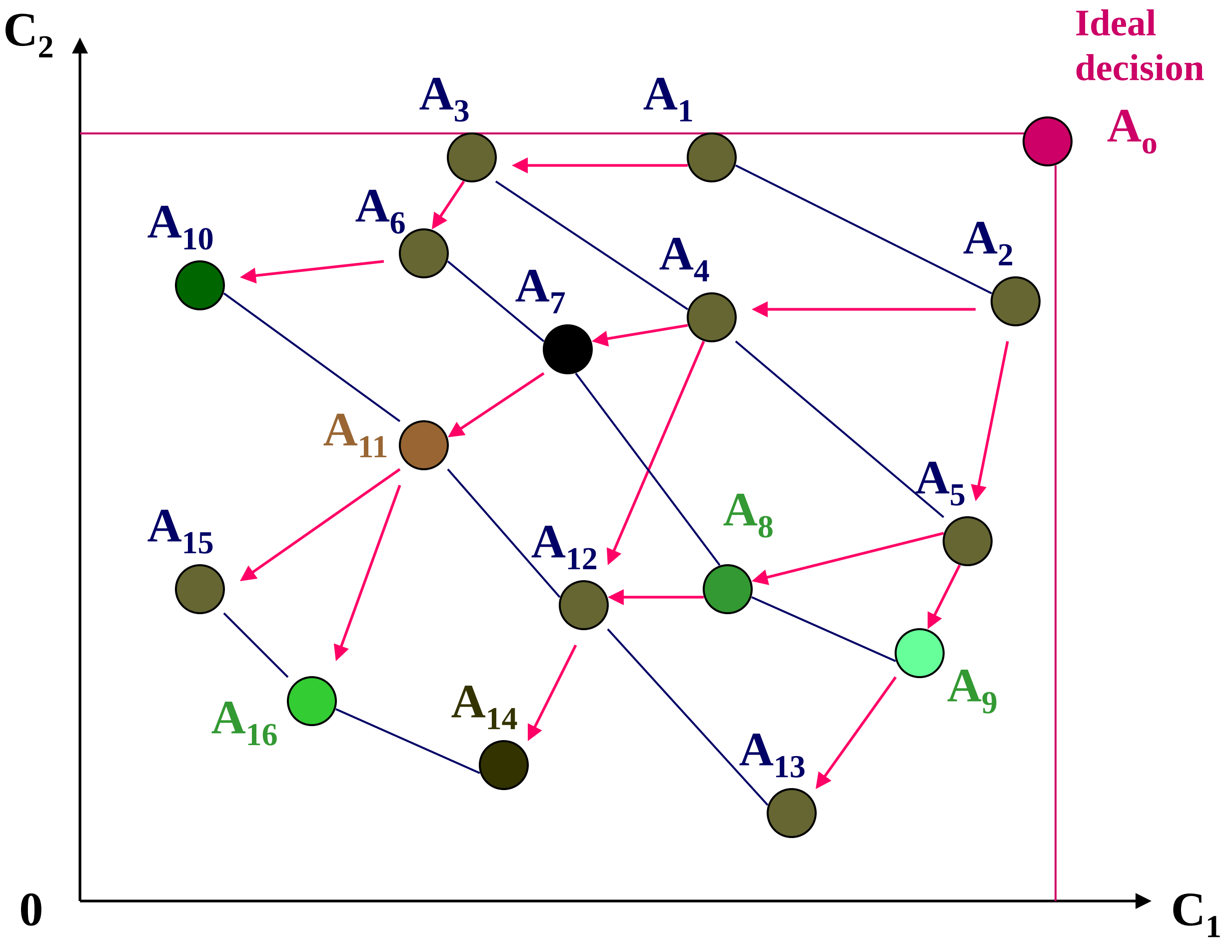


A_3 better A_2
A_3 better A_5
A_1 better A_4
A_7 better A_5
A_2 better A_4
A_2 better A_6
A_3 better A_6
A_3 better A_4
A_1 better A_5
A_7 better A_6
A_2 better A_5

Method of “equivalence” (incomparability) levels: Initial Alternatives



Method of "equivalence" (incomparability) levels: Pairwise Comparison

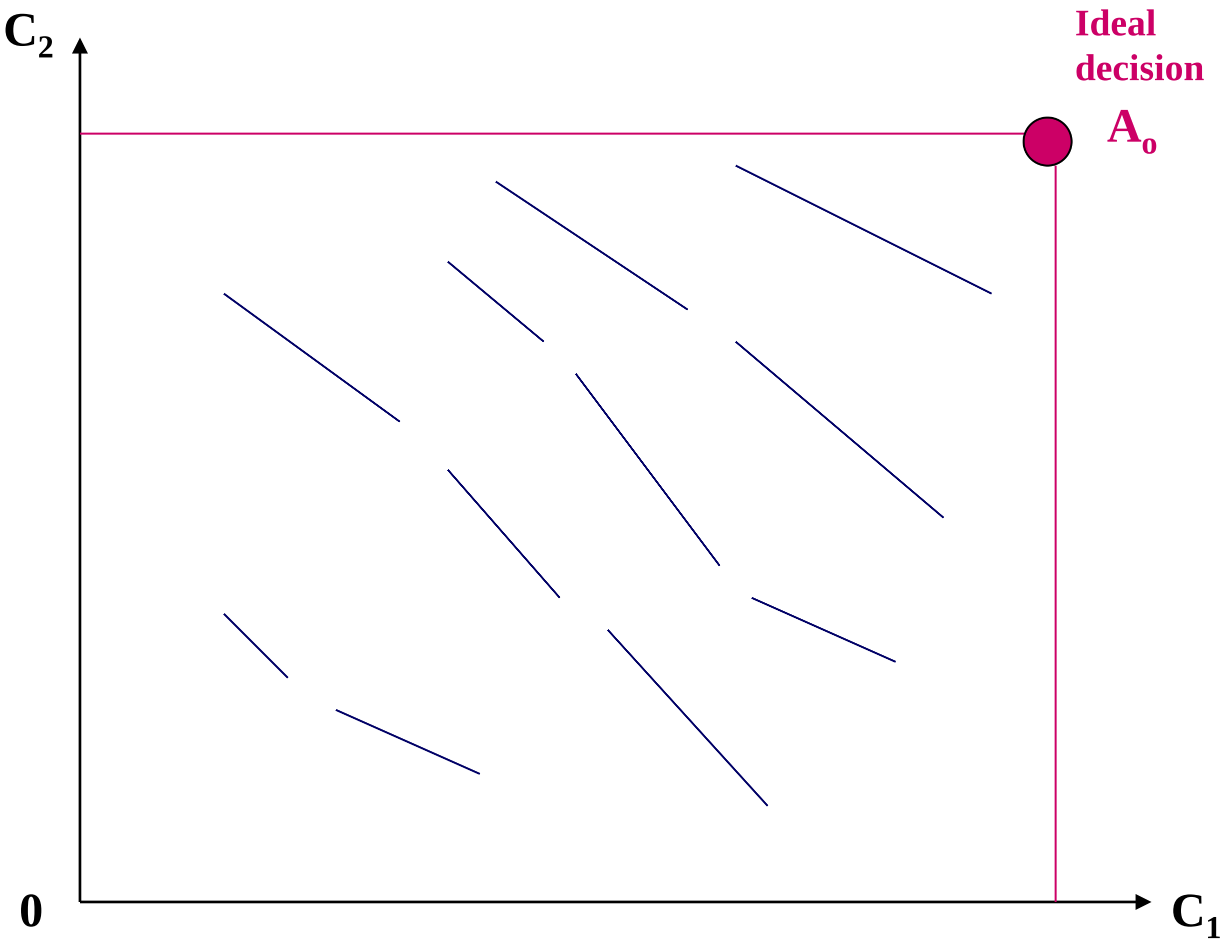


Pairwise Comparison:

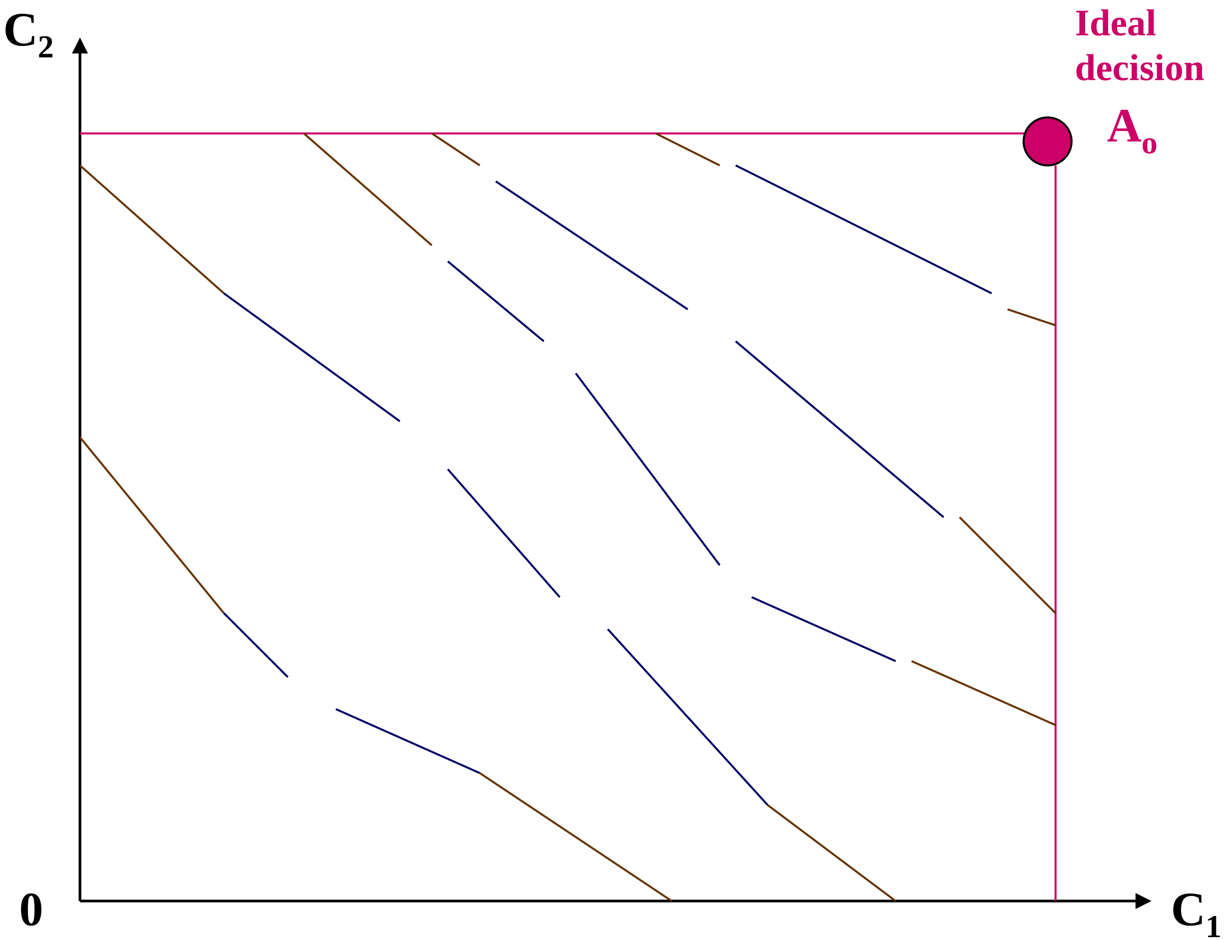
1.Dominance

2.Incomparability

Method of "equivalence" (incomparability) levels: basis or layers of incomparability



Method of "equivalence" (incomparability) levels: extended layers of incomparability



Method of "equivalence" (incomparability) levels: evaluation of new alternatives

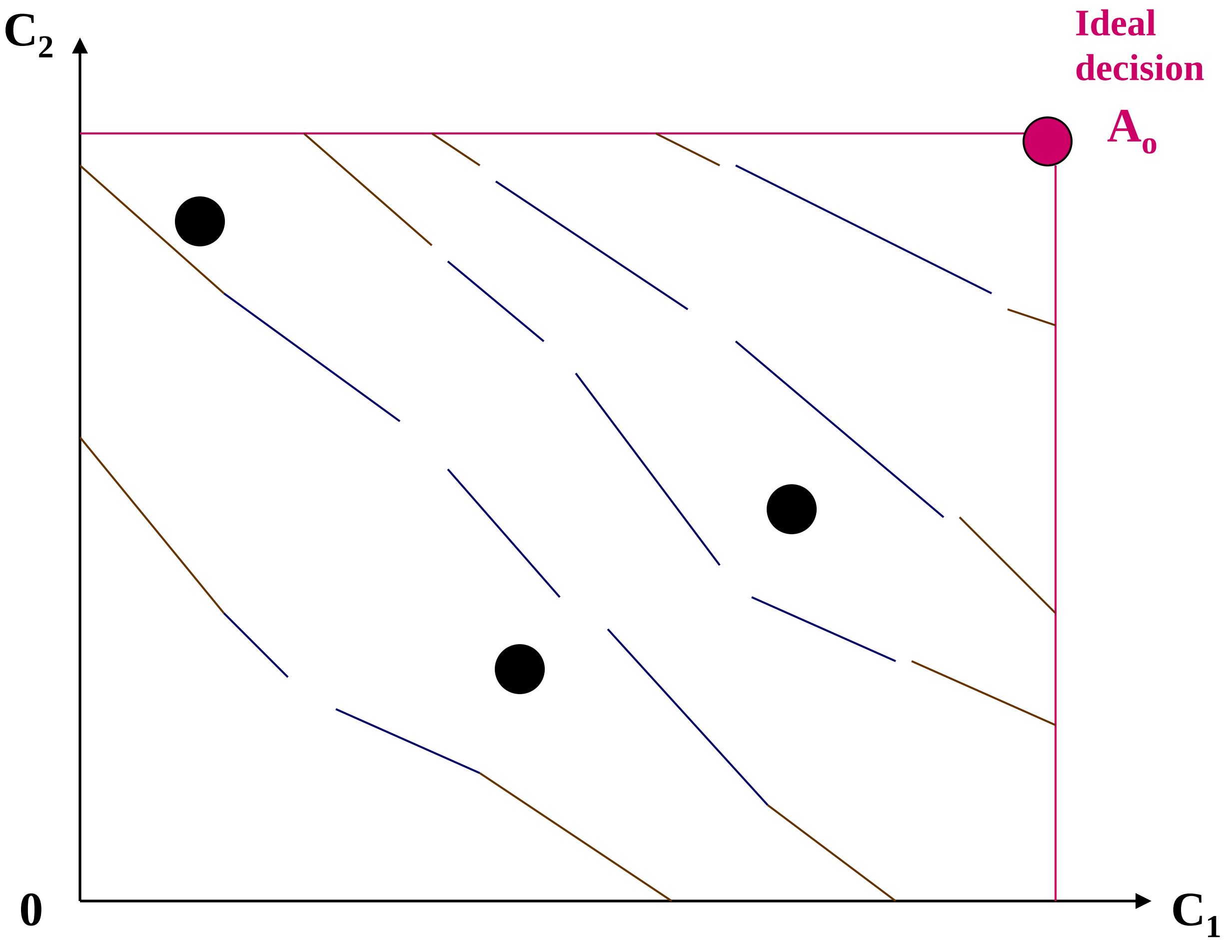


Illustration for arithmetic utility function: layers of incomparability

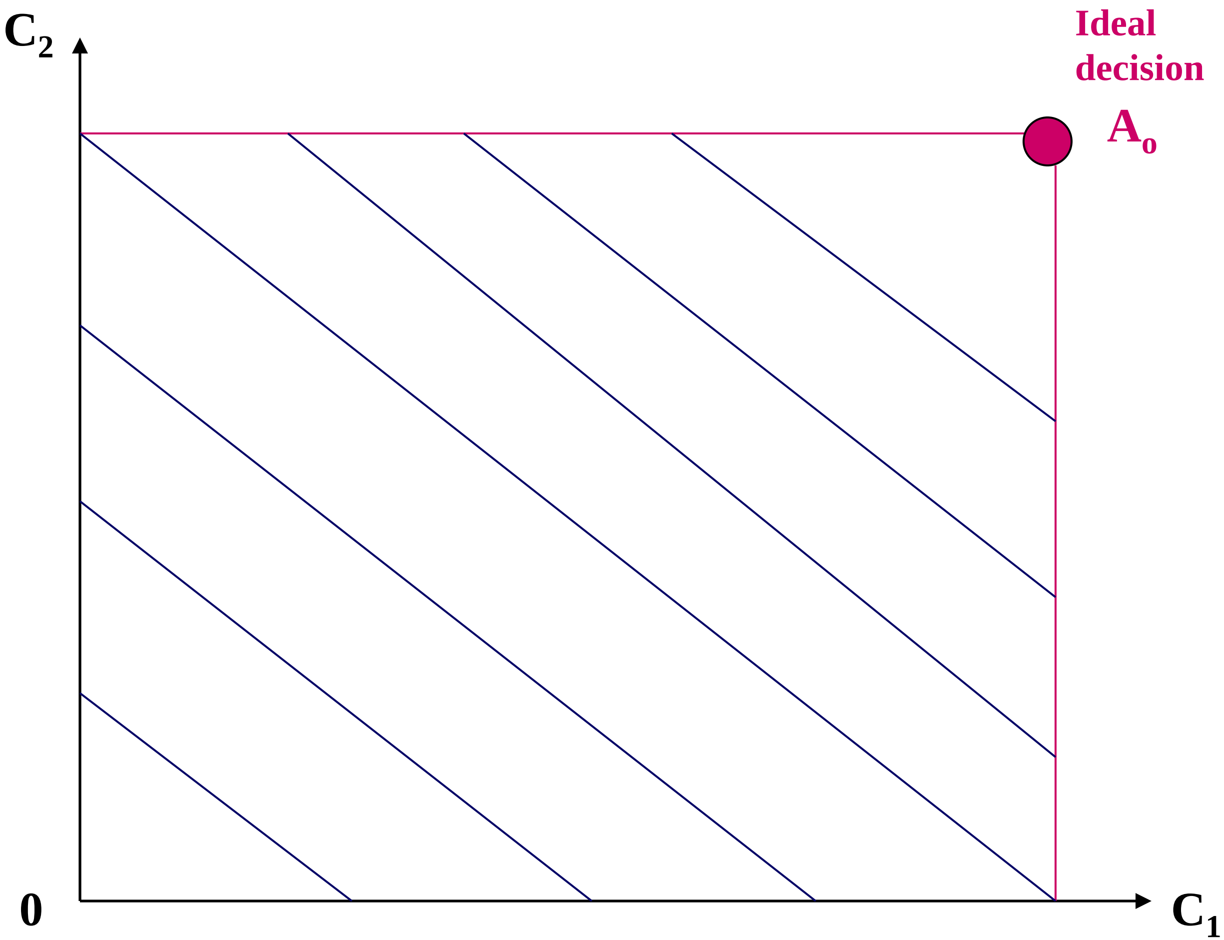


Illustration for quasi-quadratic utility function: layers of incomparability

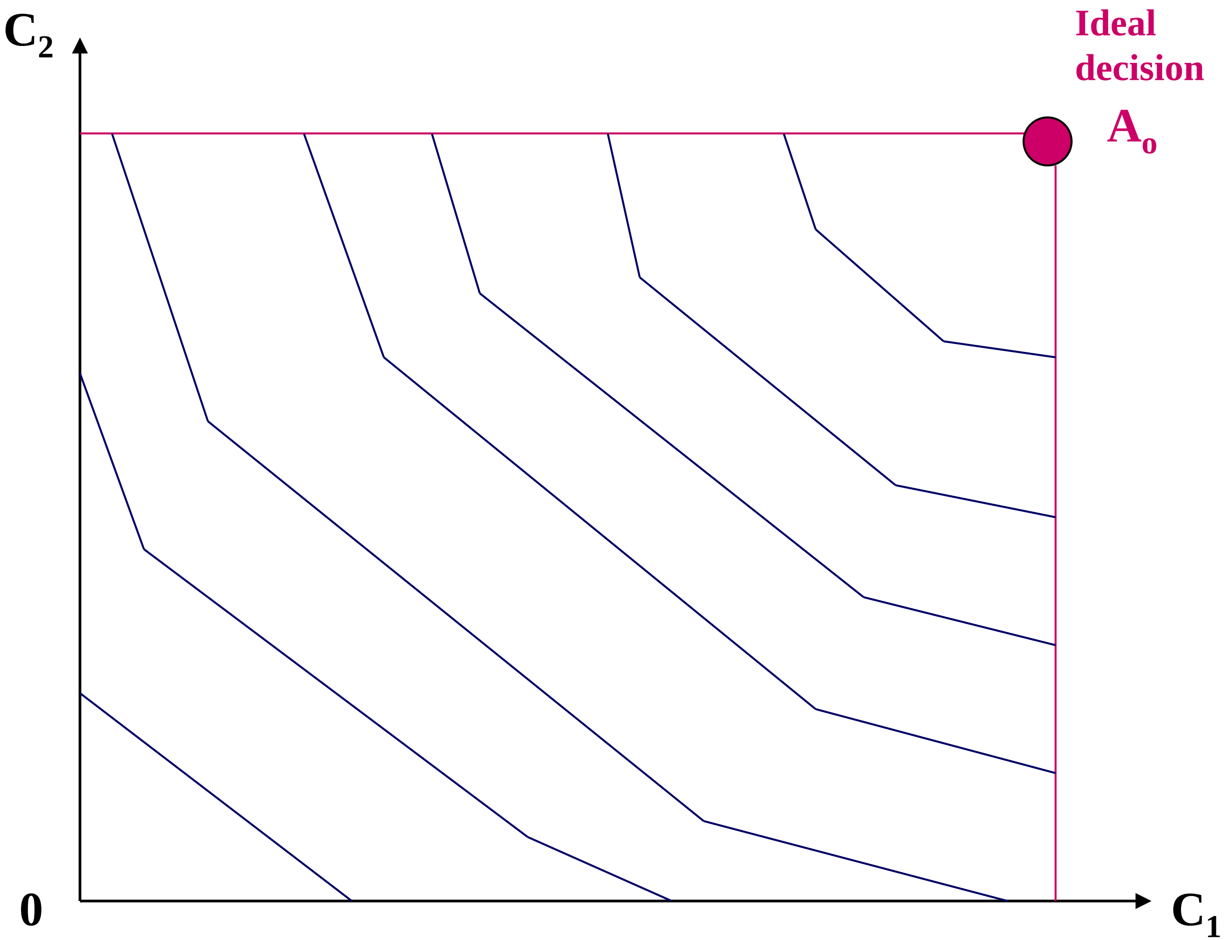
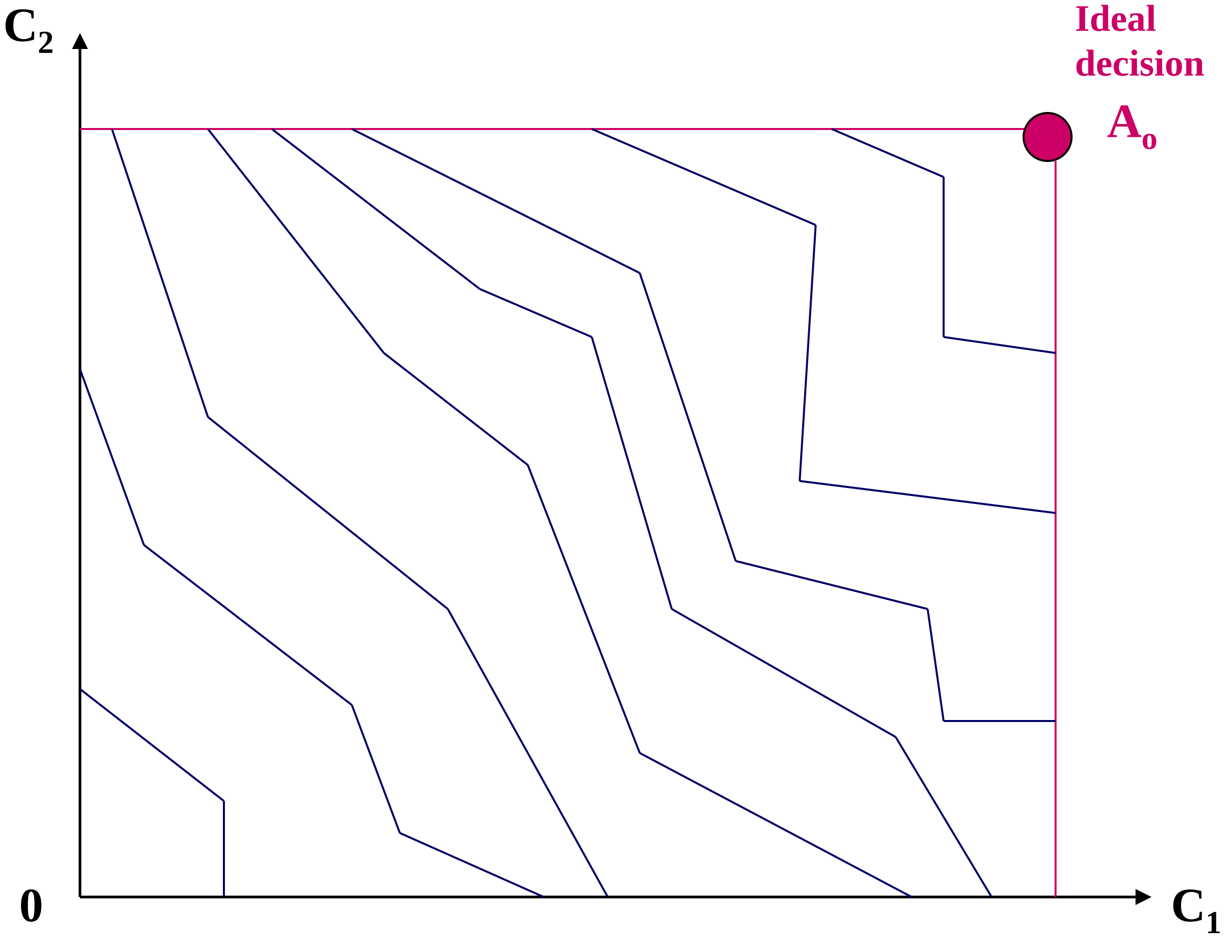


Illustration example for a "complex" situation of incomparability layers



Outranking technique (method ELECTRE) by B. Roy

Alternatives $A=(A_1, \dots, A_i, \dots, A_n)$ and criteria $C=(C_1, \dots, C_j, \dots, C_k)$,
$\forall A_i$ a vector of estimates $z_i = (z_{i1}, \dots, z_{ij}, \dots z_{ik})$, μ_j is a weight of criterion j

$\forall$ pair $A_u, A_v \in A$ to compute:

Coefficient of "concordance"

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

Coefficient of "discordance"

$$\beta_{uv} = 0 \quad \text{if} \quad |Y(u,v)| = 0 \qquad \text{else}$$

$$\max_j ((\mu_j \ |z_{uj} - z_{vj}|) \ / \ (d_j \sum_{j=1}^{k} \mu_j))$$

$X(uv) = \{ j \mid z_{uj} \geq z_{vj} \}$, $Y(uv) = \{ j \mid z_{uj} < z_{vj} \}$, d_j is scale size

RULE: A_u better A_v if $(\alpha_{uv} \geq p)$ & $(\beta_{uv} \leq q)$
where p, q are thresholds (e.g., $p = 0.9$ and $q = 0.2$)

Illustrative numerical example for multicriteria ranking

	C_1	C_2	C_3	C_4	C_5	Criteria {j}
	0.1	0.1	0.15	0.4	0.25	weight 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

A_1 ? A_3

X(1,3) = { 1,4,5 }
Y(1,3) = { 2,3 }

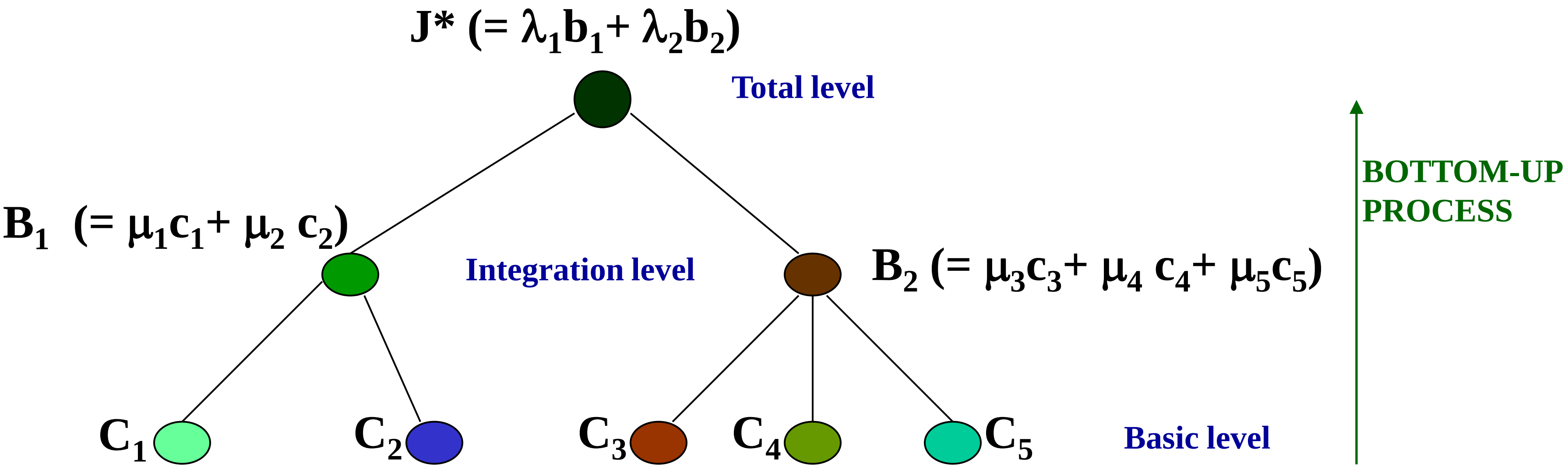
$$\alpha_{13} = (1/1)(0.1 + 0.4 + 0.25) = 0.75$$

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

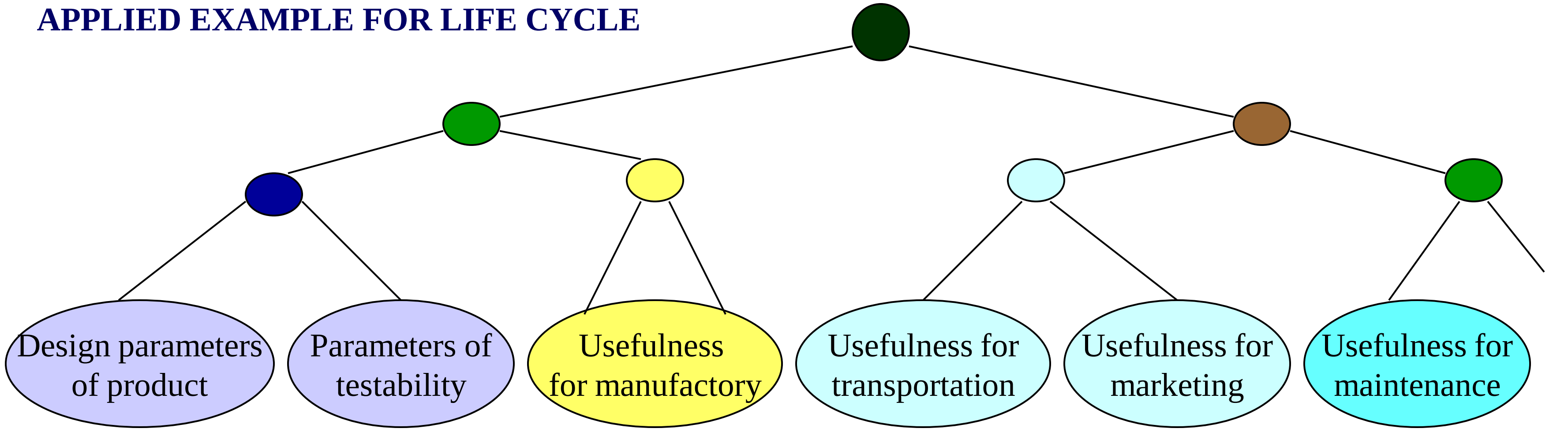
Version of Result 1: p = 0.7 q = 0.3 ⇒ A_1 better A_3

Version of Result 2: p = 0.8 q = 0.2 ⇒ incomparable ones

Analytic Hierarchy Process (T.L. Saaty)



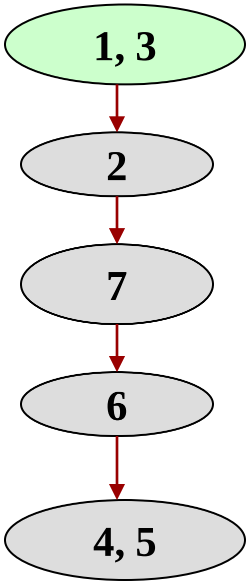
APPLIED EXAMPLE FOR LIFE CYCLE



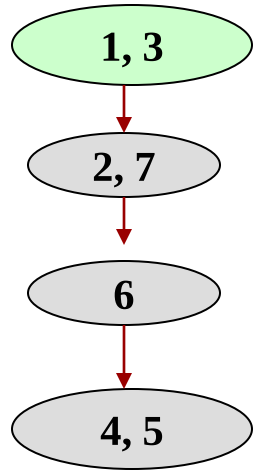
Integration (aggregation) of results

Previous example:

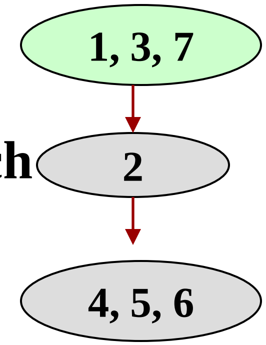
F_a



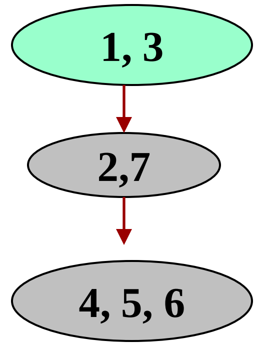
F_q



Pareto approach



Intuitive integration:



Integration (aggregation) approaches

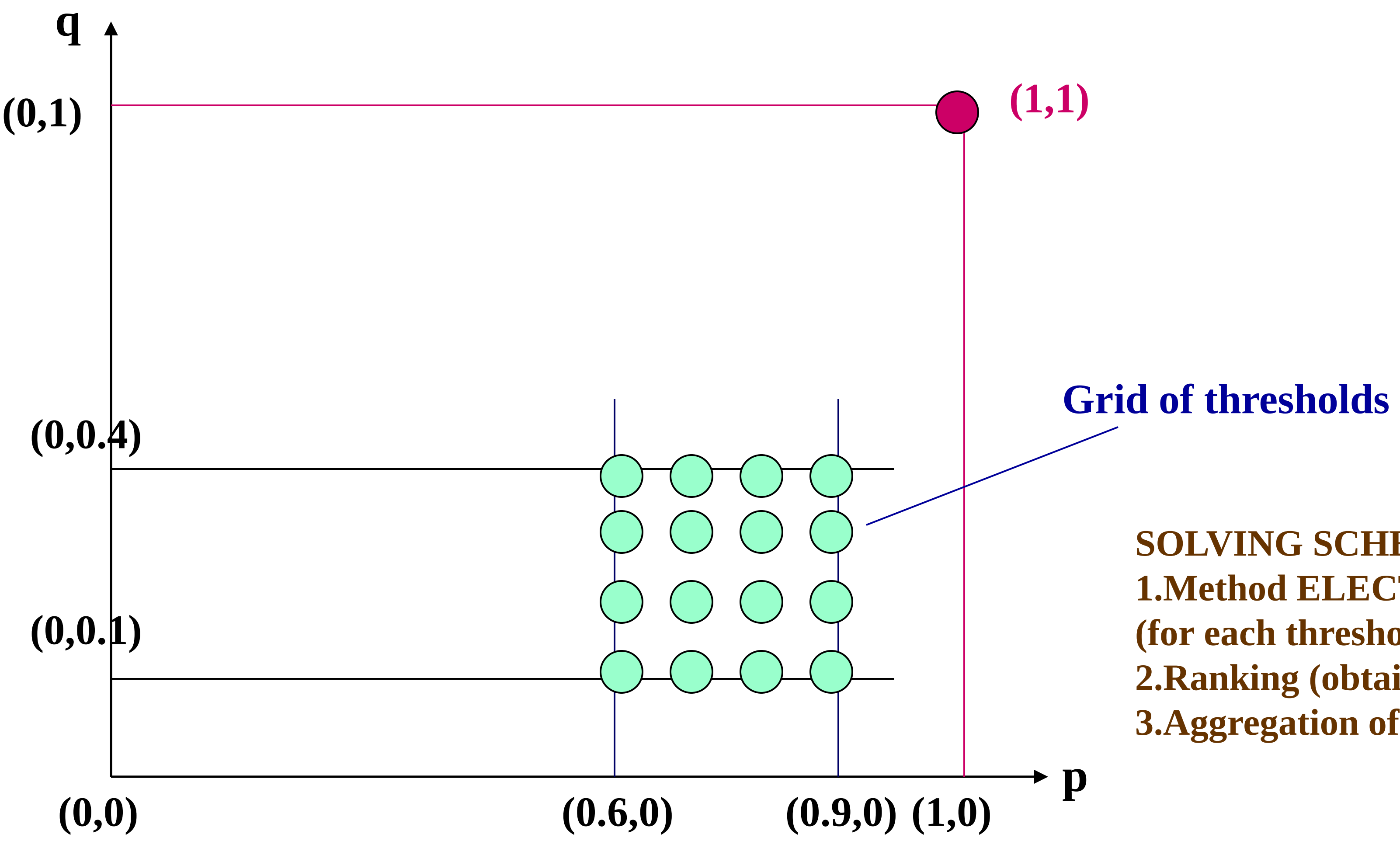
1.Election rules

2.Election rules & deletion of “margin results”

3.Multicriteria approaches above

4.Membership function (including fuzzy results)

Integration (aggregation) approach: Example for usage of ELECTRE (M.Sh. Levin, DSS COMBI)



SOLVING SCHEME:
1.Method ELECTRE
(for each threshold pair)
2.Ranking (obtaining layers)
3.Aggregation of results

LECTURE 11-12. Course: “Design of Systems: Structural Approach”

Dept. “Communication Networks &Systems”, Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.11. Framework of decision making, examples.

L.12. Function, mapping. Optimization models.

PLAN:

1.Framework of multicriteria decision making

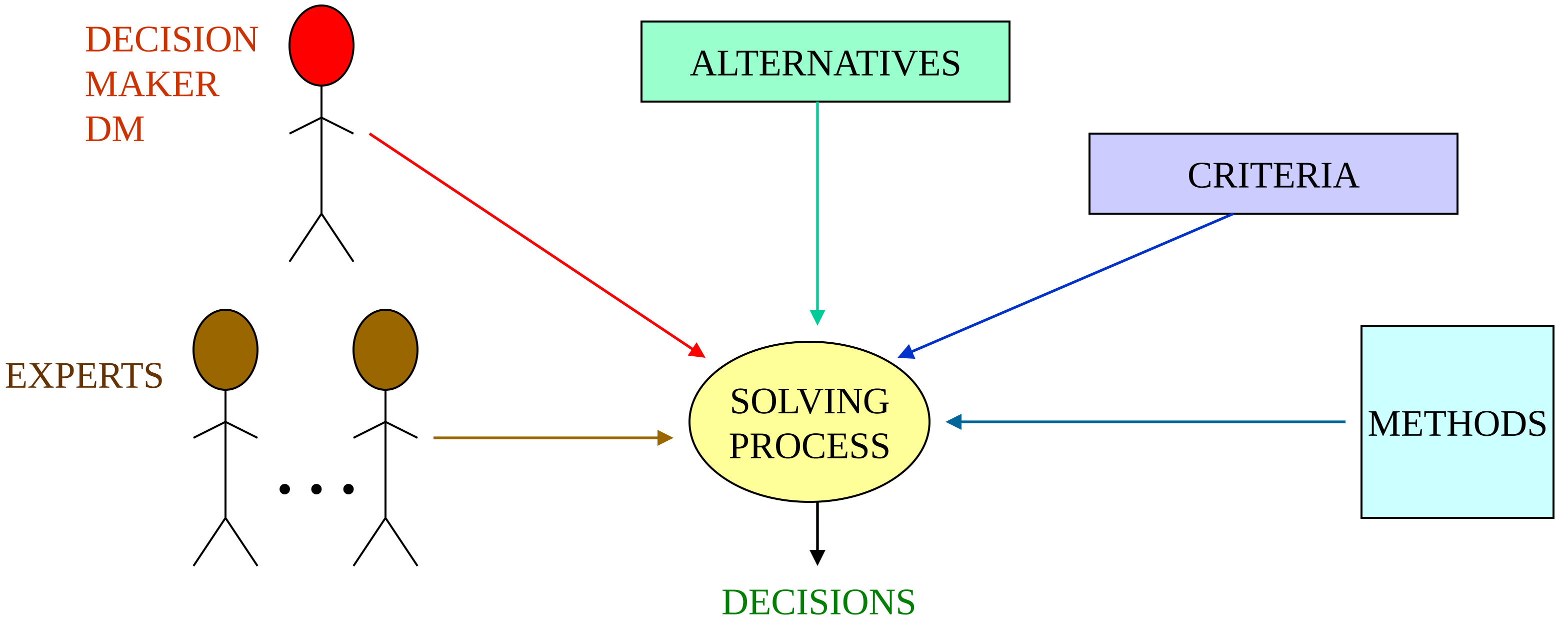
2.Partitioning the procedure of multicriteria decision making

3.Numerical examples: *partitioning of initial problem *aggregation of results

4.Mapping

Sept. 25, 2004

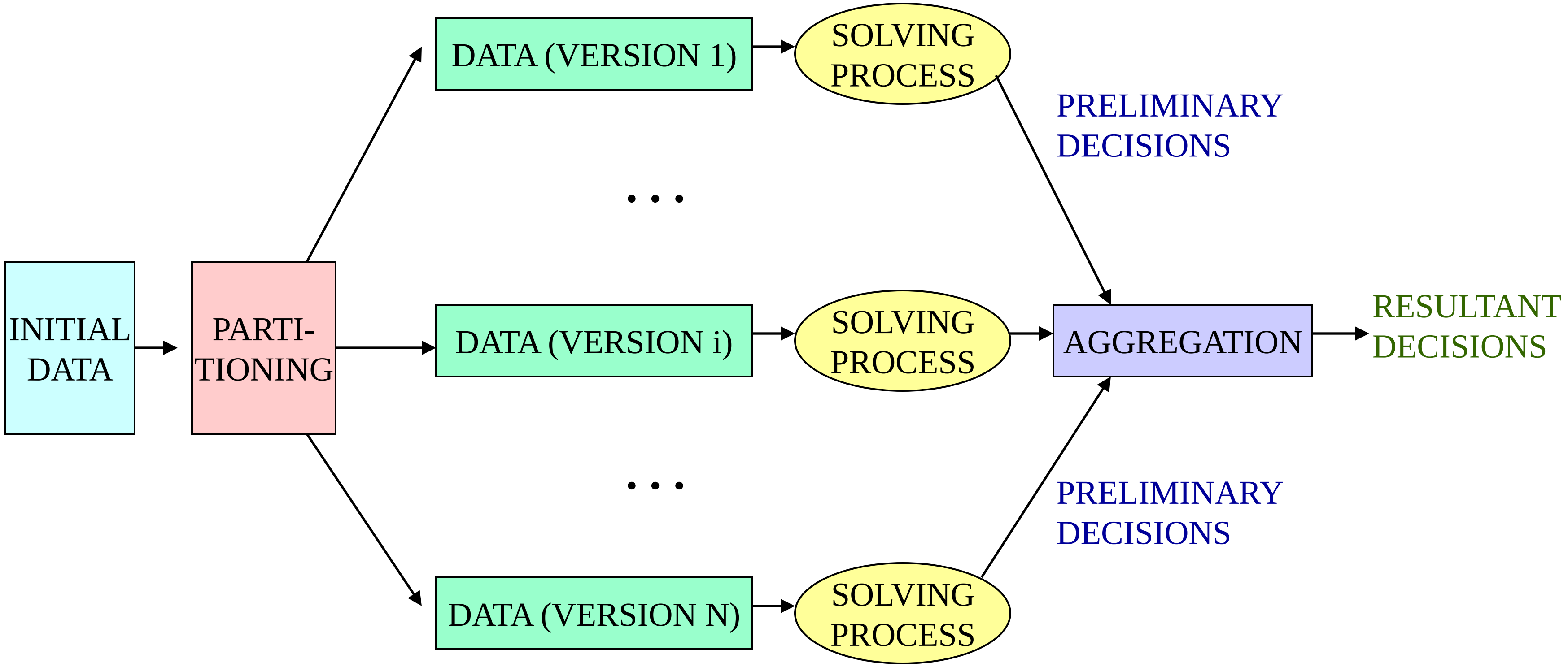
Framework of multicriteria decision making & its partitioning (parallelization)



APPROACHES FOR PARTITIONING OF DECISION MAKING FRAMEWORK:

1. **By criteria**
2. **By alternatives**
3. **By experts**
4. **By methods**
5. **Hybrid approaches**

Parallel Processing Scheme



Example: Partitioning by criteria groups

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	TOTAL
A_1	10	8	9	7	6	1	3	3
A_2	8	7	7	9	10	3	1	3
A_3	10	7	6	7	9	3	2	2
A_4	9	9	9	8	9	1	1	1
A_5	7	10	10	8	8	1	2	2

FINAL AGGREGATION

Example: Partitioning by alternative groups

PRELIMINARY STAGE

	C_1	C_2	C_3	1^{st} step
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

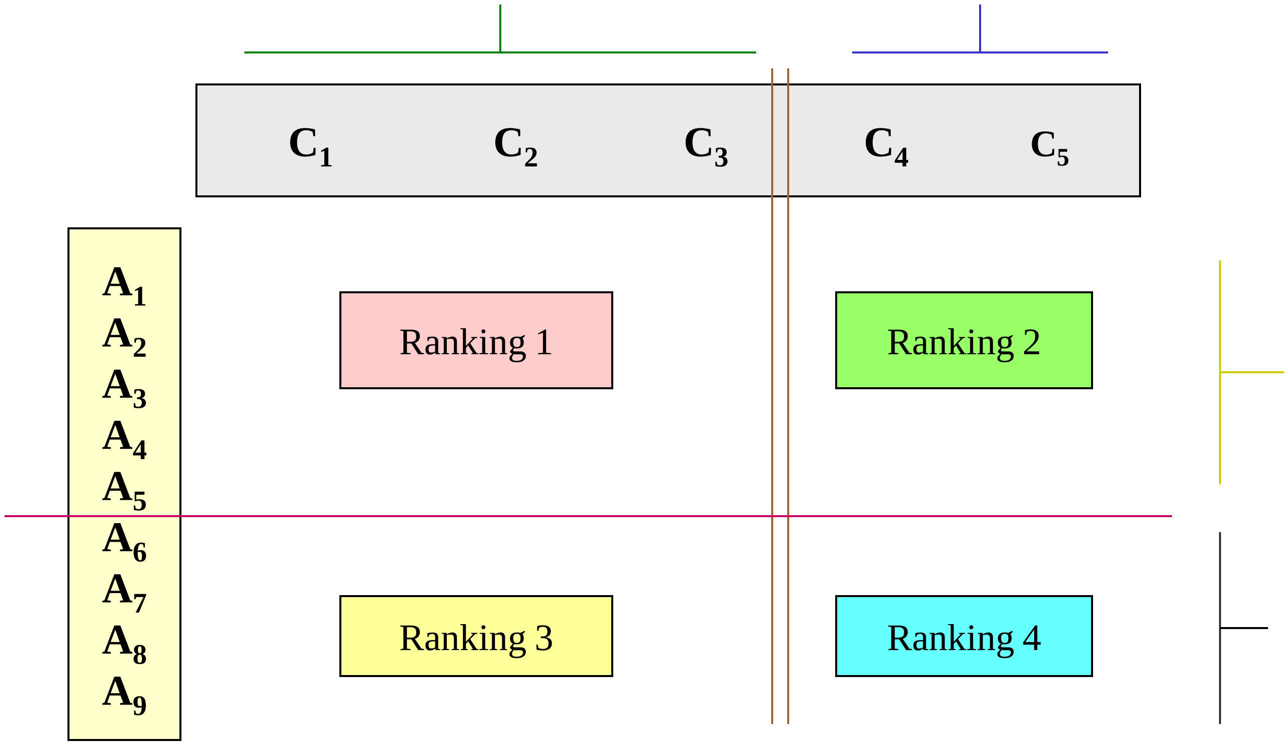
FINAL STAGE

	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

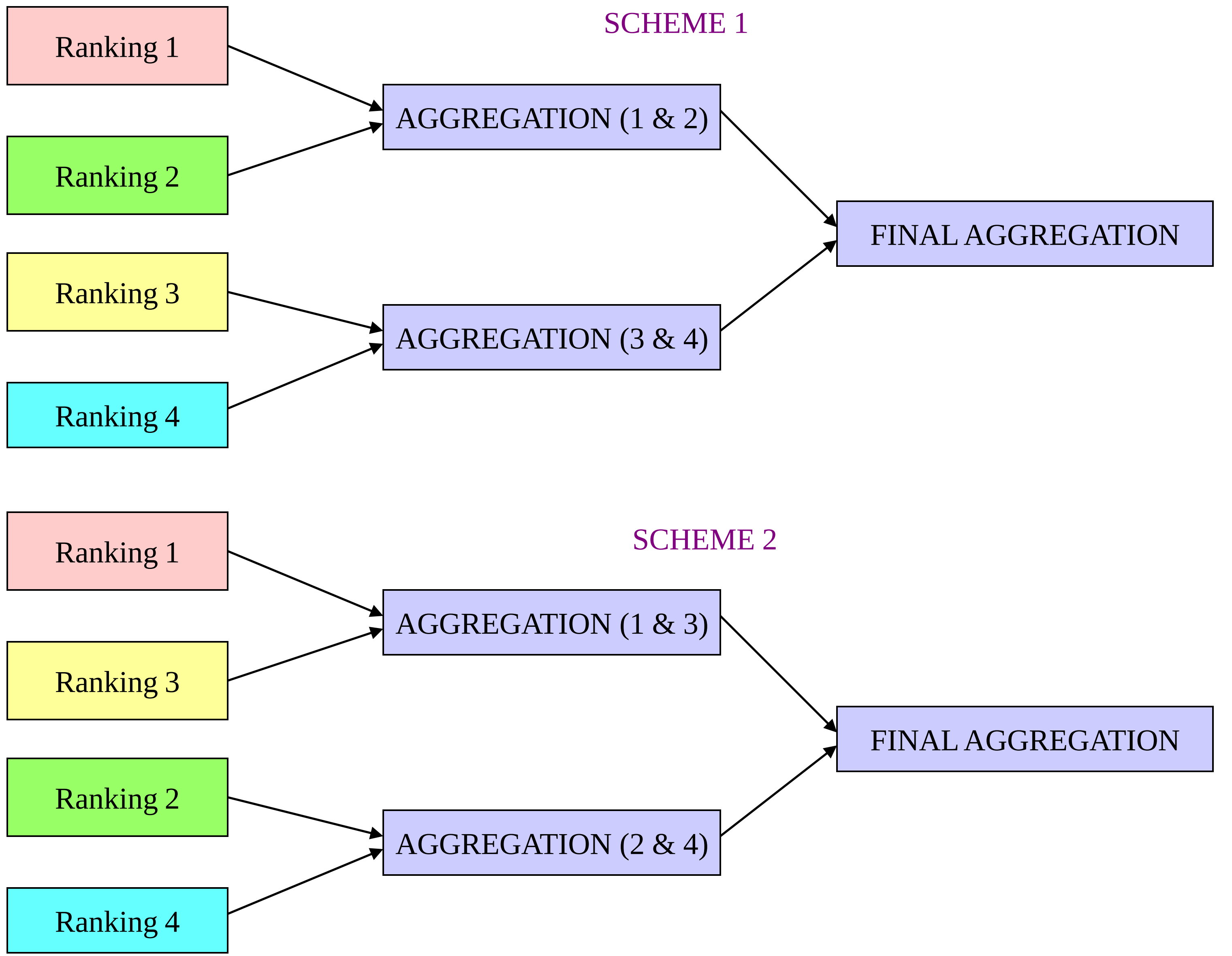
Example: Joint partitioning by criteria groups & alternative groups

	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

Possible schemes for joint partitioning by criteria groups & alternative groups

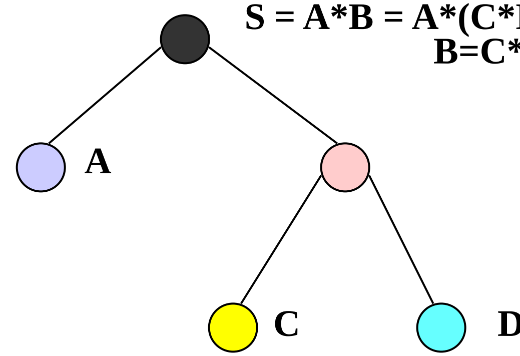


Possible schemes for joint partitioning by criteria groups & alternative groups



Example: Integration tables (Glotov & Paveljev)

S = A*B = A*(C*D)
B=C*D



Scale for S

B \ A	1	2	3	4
1	1	1	2	2
2	1	2	2	3
3	2	2	3	3
4	3	3	3	4

Scale for B

D \ C	1	2	3	4
1	1	1	2	3
2	2	2	3	3
3	3	3	4	4

Scale for A

1
2
3
4

Scale for C

1
2
3
4

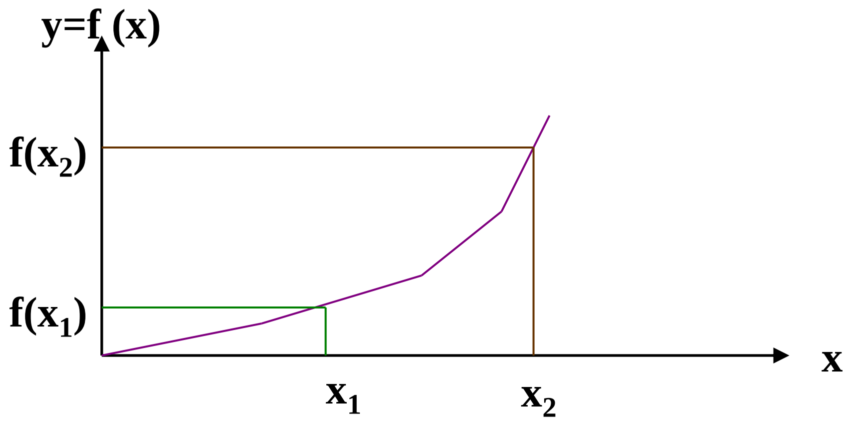
Scale for D

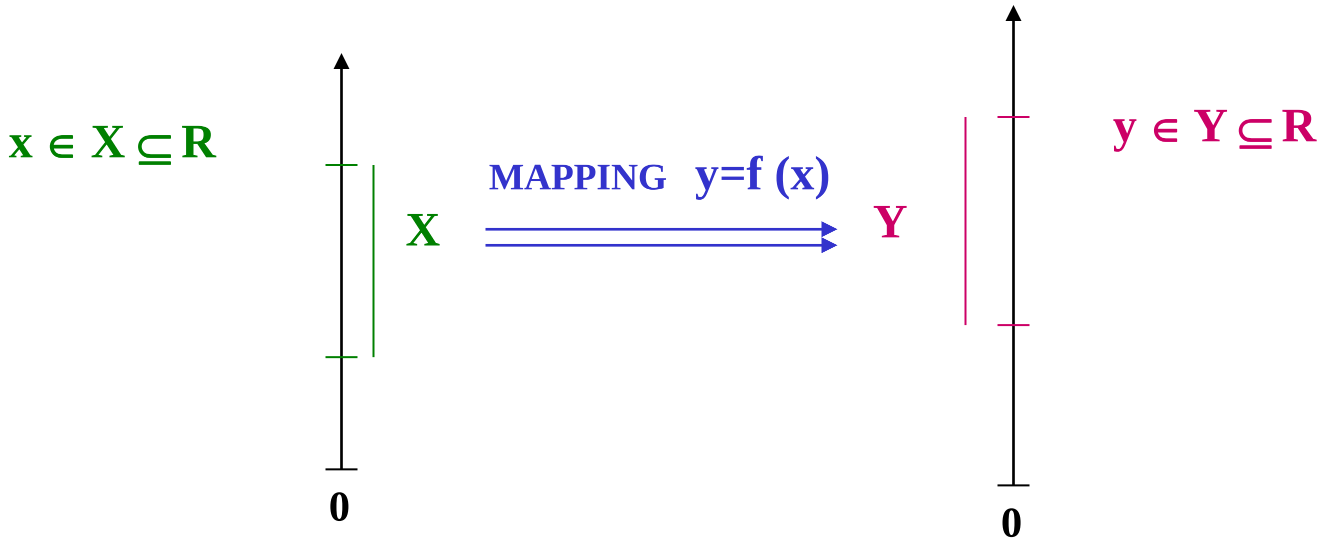
1
2
3

EXAMPLE: Basic estimates are the following: 4 for A, 3 for C, 1 for D;
intermediate estimate for B is 2;
resultant estimate for S is 3.

NOTE: multidimensional integration tables are possible (and usefu l) too.

$x \in R$

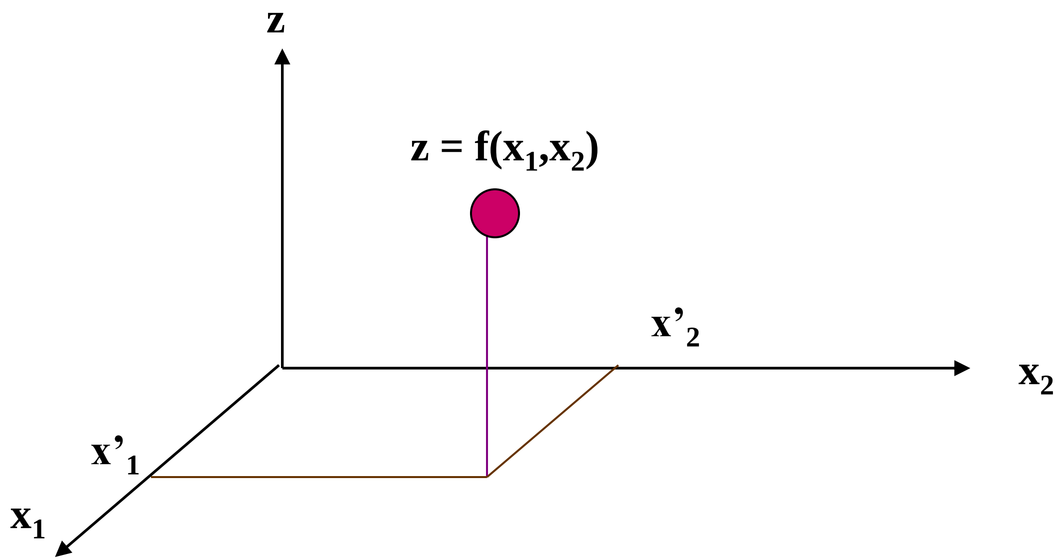




$z = f(x_1,x_2) \in R^2$

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

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



GENERALLY:

$(x_1, \ldots ,x_m) \in X \subseteq R^m$

$(y_1, \ldots ,y_n) \in Y \subseteq R^n$

Mapping & Optimization Models

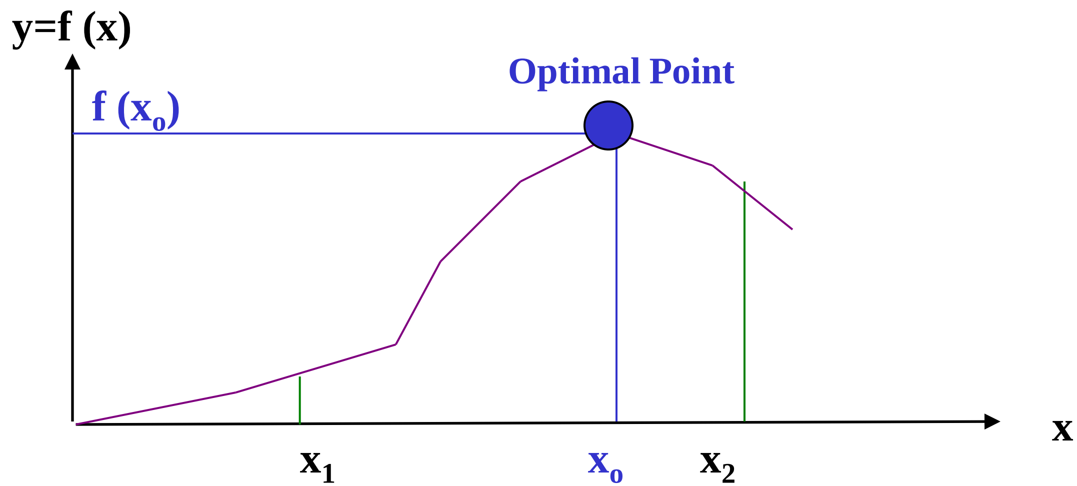
$x \in X \subseteq R$

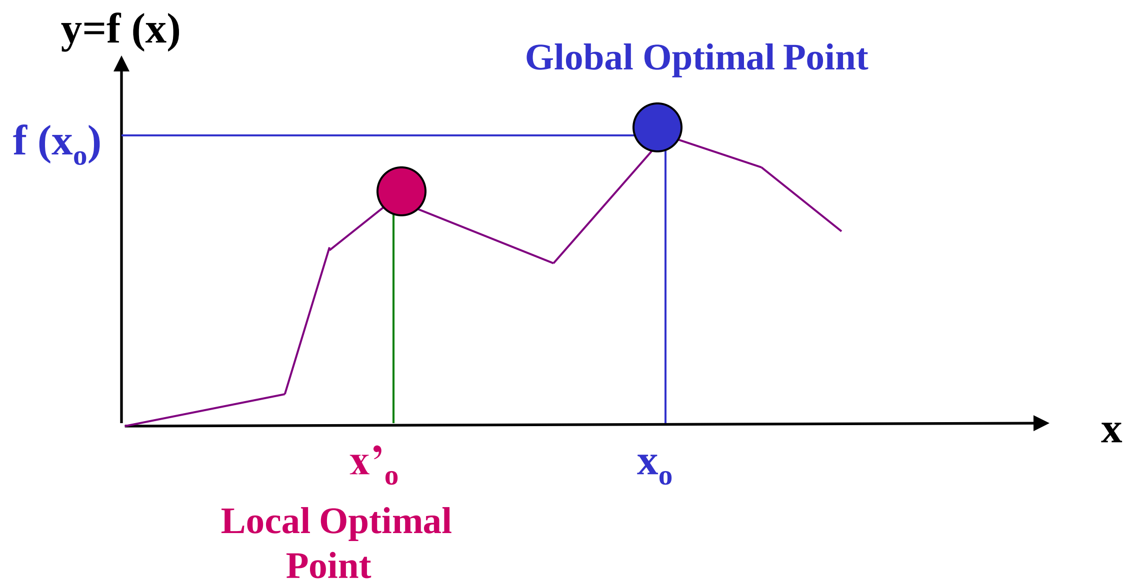
$X = [x_1, x_2]$ *(admissible domain)*

$f(x)$ *is objective function*

$$\left| \begin{array}{l} \max\ f(x) \\ \text{subject to} \\ \qquad x \in X \end{array} \right.$$

$$\left| \begin{array}{l} \max\ f(x) \\ \text{subject to} \\ \qquad x \geq x_1 \\ \qquad x \leq x_2 \end{array} \right.$$





GENERALLY:

$$\left| \begin{array}{l} \max\ f(x) \\ \text{subject to} \\ \qquad \varphi_1(x) \leq 0 \\ \qquad \dots \\ \qquad \varphi_k(x) \leq 0 \end{array} \right.$$

$\varphi_j(x)$ *is constraint function* $(1 \leq j \leq k)$

Illustration table for optimization models

Objective function f (x)	Constraint $\varphi_j(x)$	Type of model	Method
Linear	Linear	Linear	*simplex *ellipsoid method *method of Karmarkar
Quadratic	Linear	Quadratic	*simplex *ellipsoid
Convex	Linear	Convex	*gradient method *ellipsoid method
. . .	. . .	. . .	. . .

Integration (aggregation) approaches

NOTES:

1.Objective function can be examined as vector-like one too (multi-objective optimization)

2.Constraints can be examined as binary relations too

3. In discrete optimization discrete spaces are examined

4.In stochastic optimization all parameters / functions can be stochastic ones

5.It is possible to take into account uncertainty by two ways:
***stochastic parameters / functions**
***parameters / functions on the basis of fuzzy sets**

LECTURE 13. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.13. Basic models of combinatorial optimization I.

PLAN:

1.Basic combinatorial optimization problems:

*knapsack problem, *solving schemes for multicriteria knapsack problem, *multiple choice problem.

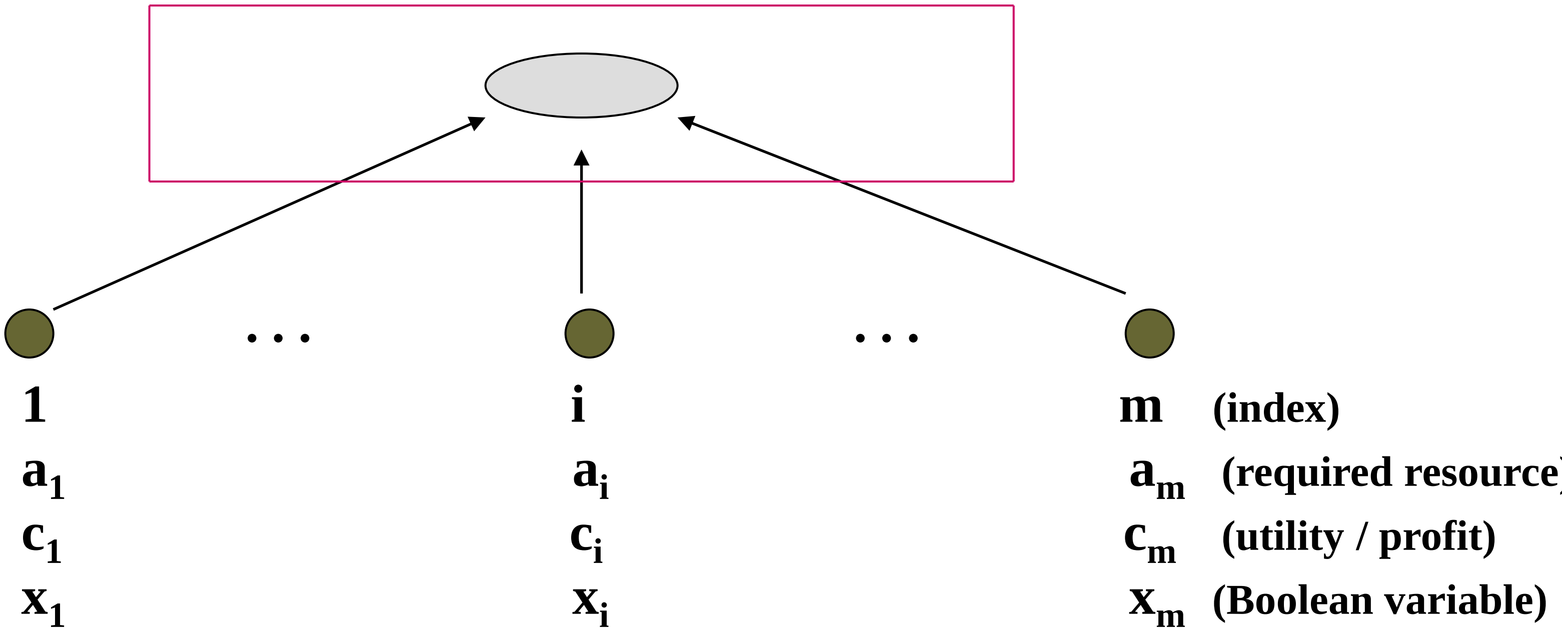
2.Algorithms: *types of solutions (exact, approximate), *types of algorithms (polynomial and enumerative algorithms),

3.Complexity of problems.

4.Global approaches and local techniques

Oct. 1, 2004

Knapsack problem



$$\textit{max} \quad \sum_{i=1}^{m} c_i \; x_i$$

$$\textit{s.t.} \quad \sum_{i=1}^{m} a_i \; x_i \le b$$

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

possible additional constraints

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

Algorithms for knapsack problem

1. ***Ordering by decreasing of*** c_i / a_i ***(algorithm by Danzig, heuristic)***

2. ***Branch-And-Bound*** **method**

3. ***Dynamic programming (exact solution)***

4. ***Dynamic programming (approximate solving scheme)***

5. ***Probabilistic methods***

6. ***Hybrid schemes***

1. $c_i = c_o$ *(equal utilities)*

2. $a_i = a_o$ *(equal required resources)*

Polynomial algorithm:

1. *ordering by non-decreasing of* a_i

2. *ordering by non-increasing of* c_i

Extended versions of knapsack problem

1.Knapsack problem with objective function as min

2.Knapsack problem with several “knapsacks”

3.Knapsack problem with additional structural (logical) constraints over elements (e.g., some kinds of trees)

4.Multi-objective knapsack problem

5.Knapsack problem with fuzzy parameters

Heuristic solving scheme for multicriteria (multiple objective) versions of knapsack problem

ALGORITM SCHEME (case of *linear ranking*):

STEP 1.Multicriteria ranking of elements (to obtain *linear ranking*)

STEP 2.Series selection of elements

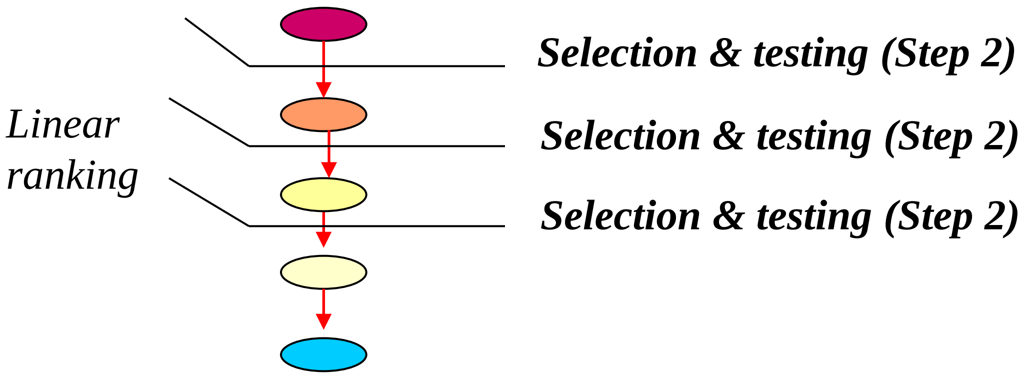
(the best element, the next element, etc.)

After each selection: testing the resource constraint ($\leq$ b).

***If the constraint is not right it is necessary to delete the last selected element and to* STOP.**

***Else: to* STEP 2.**

STOP.



Heuristic solving scheme for multicriteria (multiple objective) versions of knapsack problem

ALGORITM SCHEME (case of *group ranking*):

STEP 1.Multicriteria ranking of elements (to obtain *group ranking*)

STEP 2.Series selection of elements

(elements of the best group, elements of the next group, etc.)

After each selection: testing the resource constraint ($\leq$ b).

***If the constraint is not right it is necessary to go to* STEP 3.**

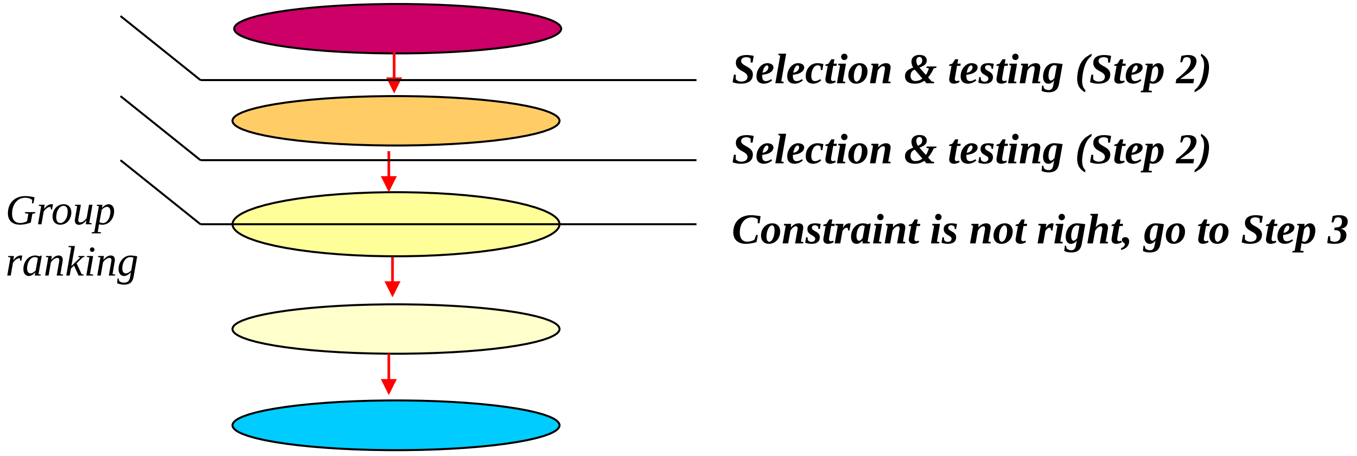
***Else: to* STEP 2.**

STEP 3. Solving for the last analyzed element group the special case of knapsack problem (with equal utilities) as series selection of elements from the list (non-increasing by a_i).

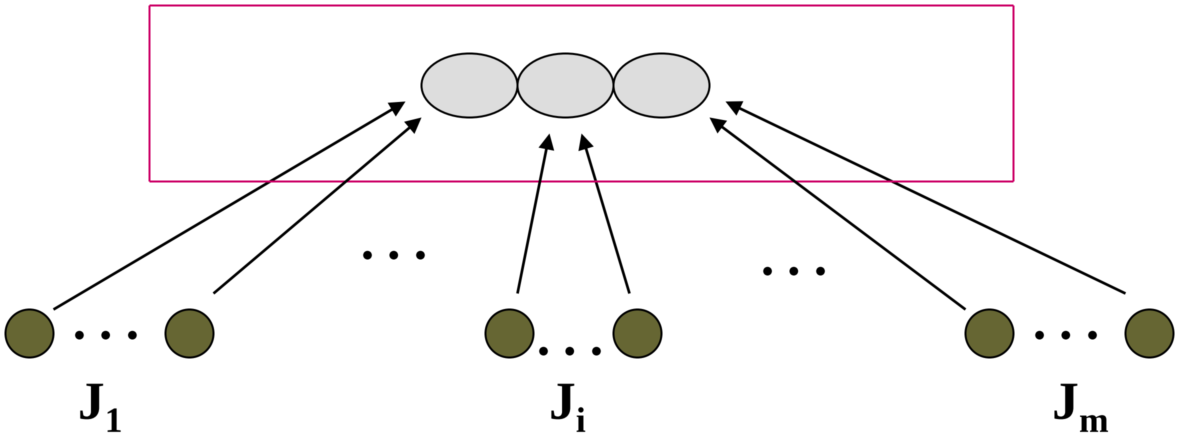
Here constraint is the following: $\leq b - \sum_{(i \in Q)} a_i$

(where Q is a set of selected elements from the previous groups)

STOP.



Multiple choice problem



$$\forall\ i \quad |J_i| = qi\,,\ \ j = 1, \ldots, qi$$

$$\begin{aligned}
&\textbf{\textit{max}} \quad \sum^{m}_{i=1} \sum^{qi}_{j=1} c_{ij}\ x_{ij} \\
&\textbf{\textit{s.t.}} \quad \sum^{m}_{i=1} \sum^{qi}_{j=1} a_{ij}\ x_{ij} \le b \\
&\qquad\quad \sum^{qi}_{j=1} x_{ij} \le 1\,, \quad i = 1, \ldots, m \\
&x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, m\,, \quad j = 1, \ldots, qi
\end{aligned}$$

Algorithms for multiple choice problem (as for knapsack problem)

1. ***Ordering by decreasing of*** $\mathbf{c_{ij} / a_{ij}}$ ***(heuristic)***

2. ***Branch-And-Bound*** **method**

3. ***Dynamic programming (exact solution)***

4. ***Dynamic programming (approximate solving scheme)***

5. ***Probabilistic methods***

6. ***Hybrid schemes***

Illustration for dynamic programming

Series Design of a Solution:

1.From START point to END point

2.From END point to START point

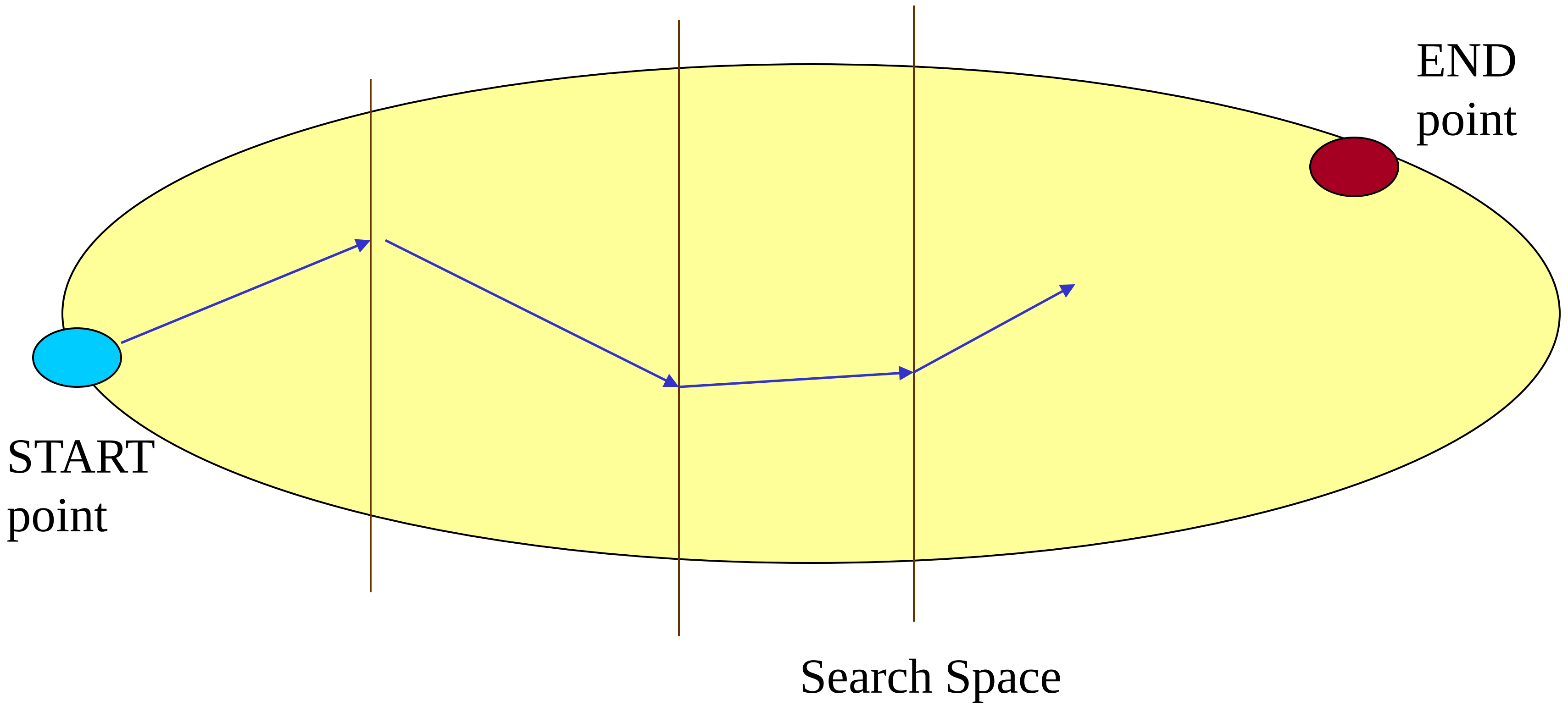
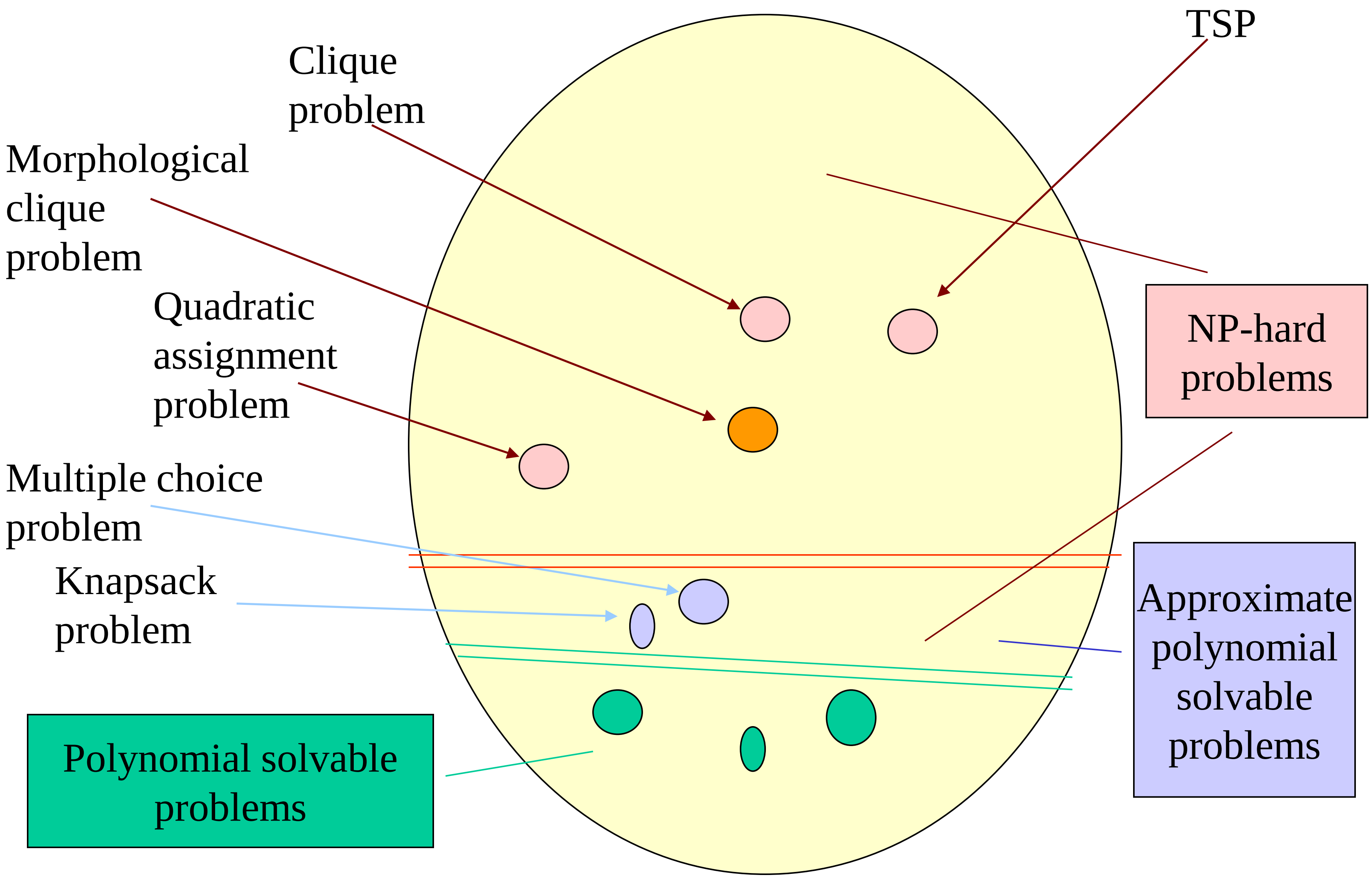


Illustration for complexity of combinatorial optimization problems



Classification of algorithms

BY EXACTNES OF RESULT (solution):

1. **Exact solution**
2. **Approximate solution (for worst case):**
   ***limited error (absolute error) *limited error (relative error) *other situations**
3. **Approximate solution (statistically)**
4. **Heurstic (without an estimate of exactness)**

BY COMPLEXITY OF SOLVING PROCESS (e.g., number of steps):

1. **Polynomial algorithms (of length of input, for example:**
   $O(n \log n)$), $O(n)$, $O(1)$, $O(n^2)$)
2. **Polynomial approximate schemes (for a specified exactness / limited error, for example: $O(n^2/\varepsilon)$ where $\varepsilon \in [0,1]$ is a relative error for objective function)**
3. **Statistically good algorithms (statistically polynomial ones)**
4. **Enumerative algorithms**

. . .

BASIC ALGORITHM RESOURCES:

1. **Number of steps (computing operations)**
2. **Required volume of memory**
3. **Required number of interaction with specialists (oracle)**
   (to get additional information)
4. **Required communication between processors (for multi-processor algorithms)**

Global approaches and local techniques

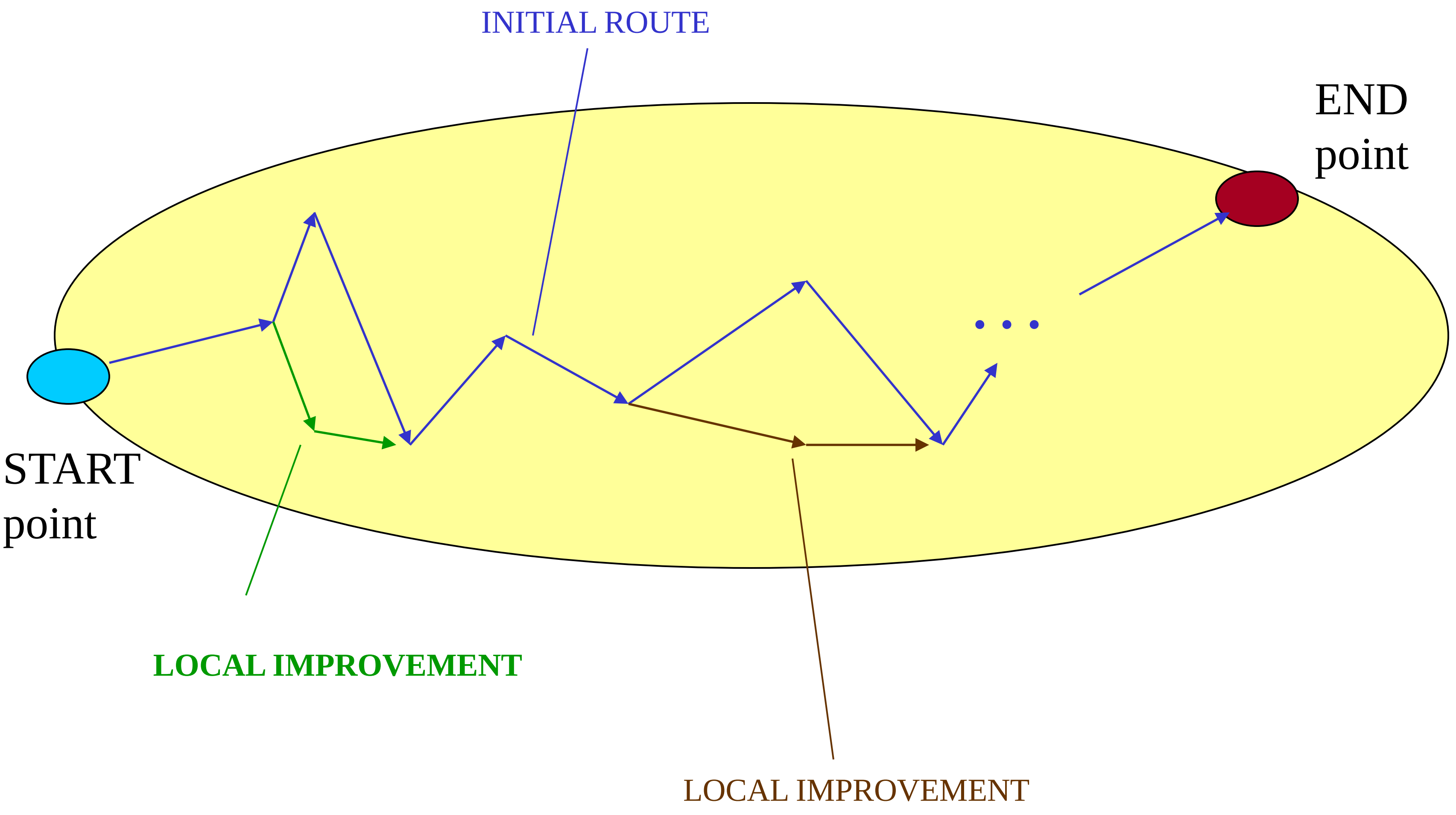
GLOBAL APPROACHES:

1.Partitioning into subproblems
2.Decomposition (extension of an obtained "good" local solutions)
(examples: dynamic programming, Branch-And-Bound)
3.Grid method with deletion of "bad points"
4.Approximation approach (i.e., approximation of initial problem or its part(s) by more simple construction(s))

LOCAL TECHNIQUES:

1.Local optimization as improvement of a solution or its part
2.Probabilsitic steps
3.Greedy approach (selection of the "simple" / "close" / etc. step)
4.Recursion

Illustration for improvement of a solution (local optimization)



LECTURE 14-15. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Institute of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.14. Basic models of combinatorial optimization II.

L.15. Scheme of multicriteria design PSI

PLAN:

1.Basic combinatorial optimization problems:

*integer nonlinear programming (special formulation & applied example),

*packing problems & bin-packing problem (illustration),

*scheduling problems (problem and algorithm for assembly process, 3 examples for one-machine scheduling,

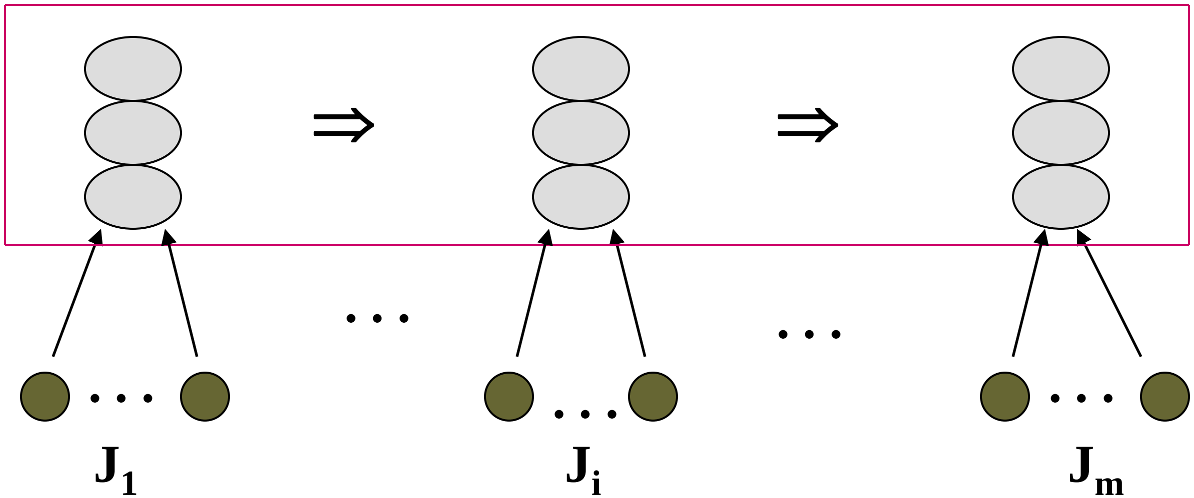
*maximal clique problem (illustration)

2.Scheme of multicriteria design (PSI – parameter space investigation)

Oct. 2, 2004

Integer Nonlinear programming (modular design of series system from the viewpoint of reliability)

(by Berman & Ashrafi)



$$\forall i \quad |J_i| = qi , j = 1, \ldots , qi$$

$$\textit{max} \quad \prod_{i=1}^{m} \left(1 - \prod_{j=1}^{qi} (1 - p_{ij} \; x_{ij})\right)$$

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

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

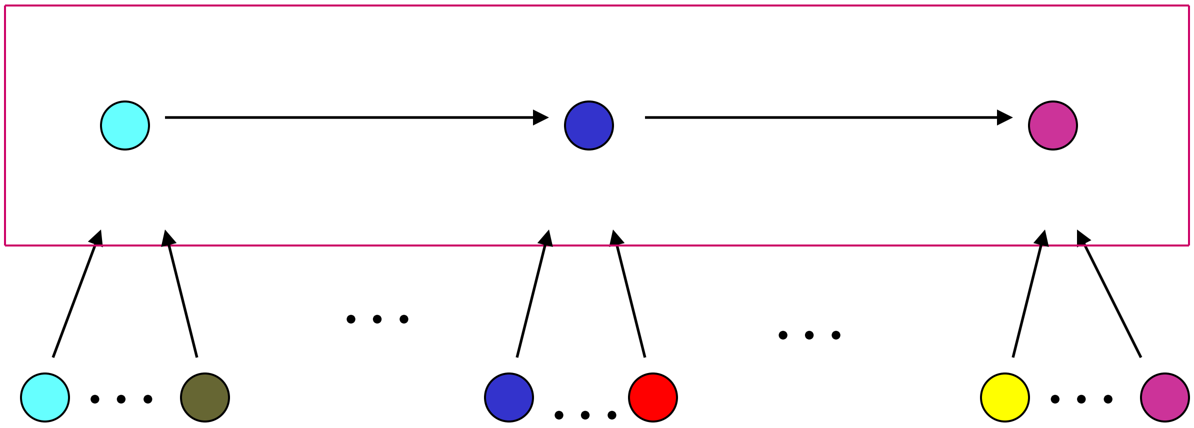
$$x_{ij} \in \{0, 1\}, \quad i = 1, \ldots , m , \quad j = 1, \ldots , qi$$

p_{ij} *is reliability* , d_{ij} *is cost*

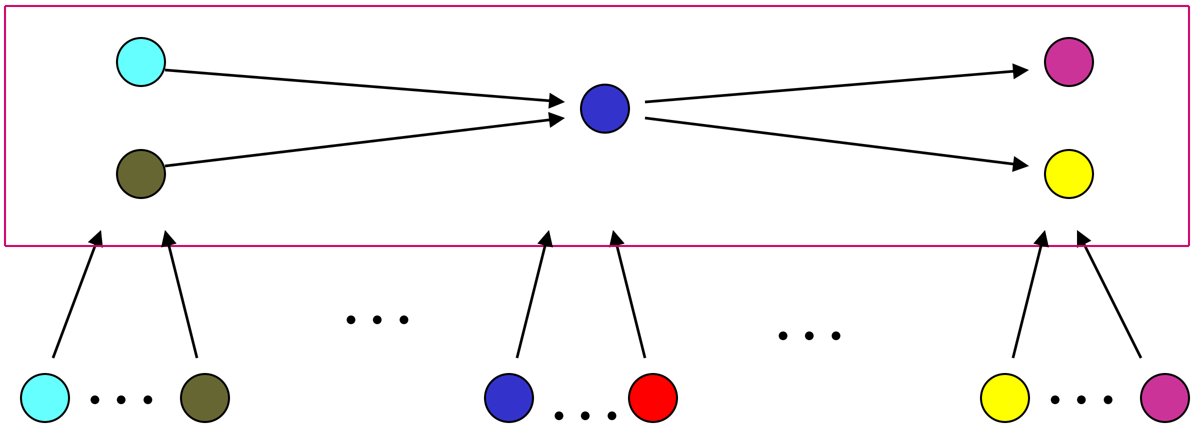
Integer Nonlinear programming (modular design of series system from the viewpoint of reliability)

(by Berman & Ashrafi)

EXAMPLE 1
(series scheme)



EXAMPLE 2
(parallel-series scheme)



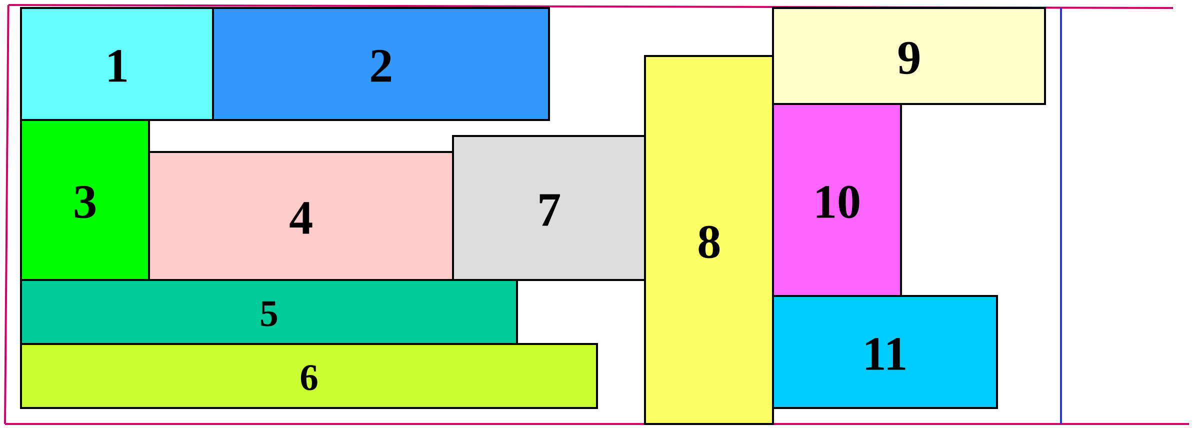
1.*Branch-And-Bound* method

2.*Dynamic programming*

4.*Heuristics (e.g., reducing the problem to a continuous one)*

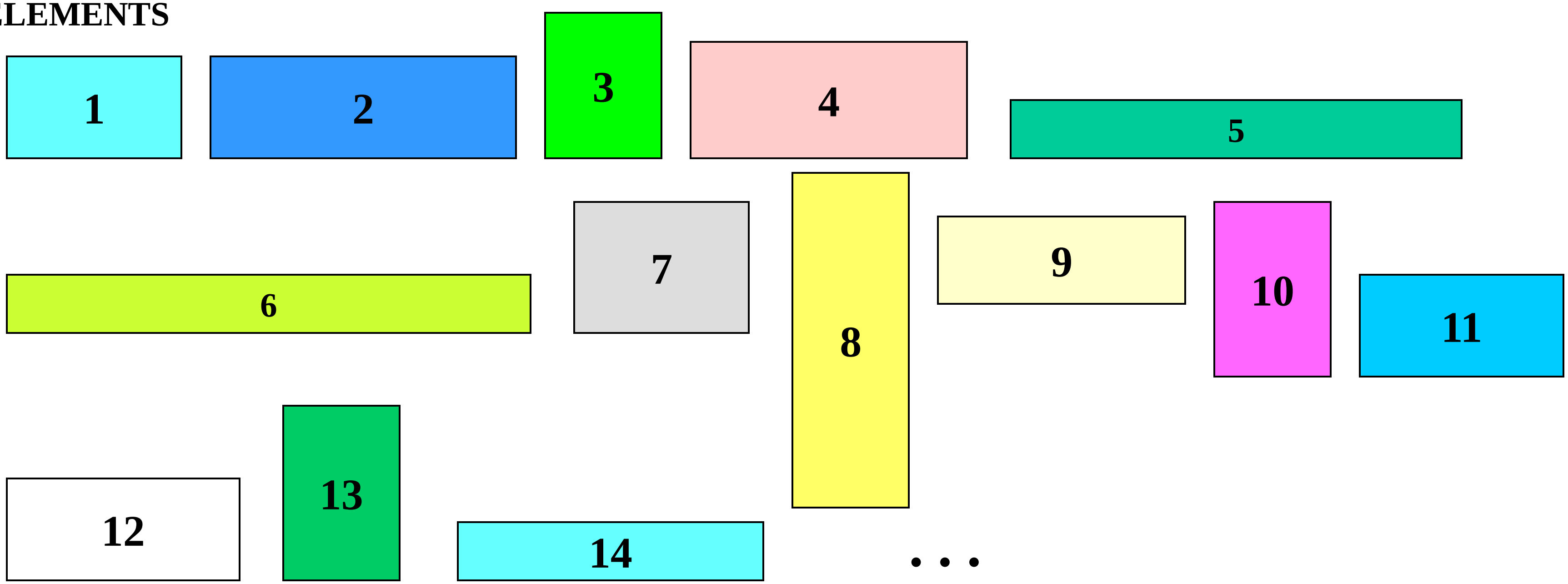
Packing problem (illustration)

REGION FOR PACKING

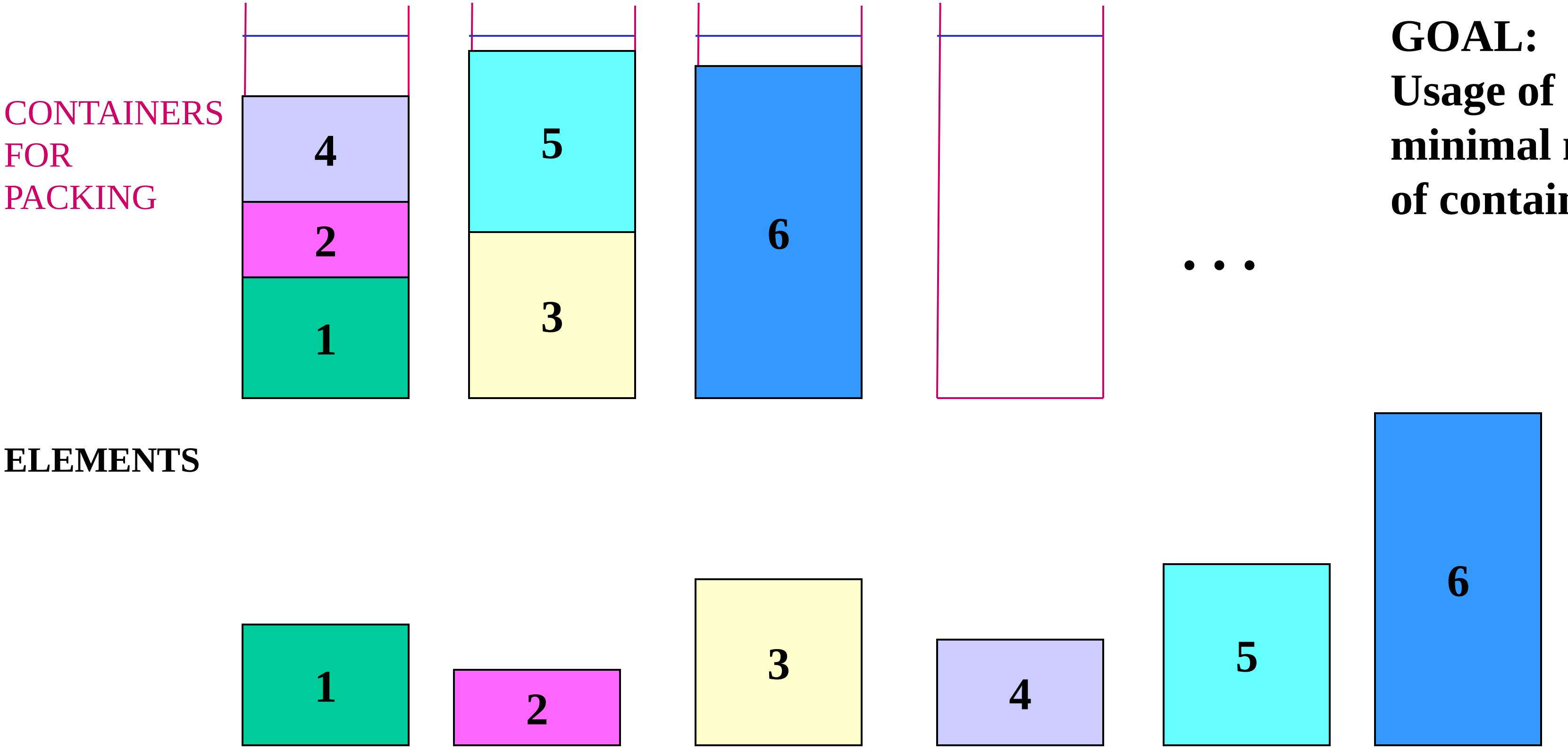


GOALS:
***Maximum of packed elements**
***Minimum of free space**

ELEMENTS

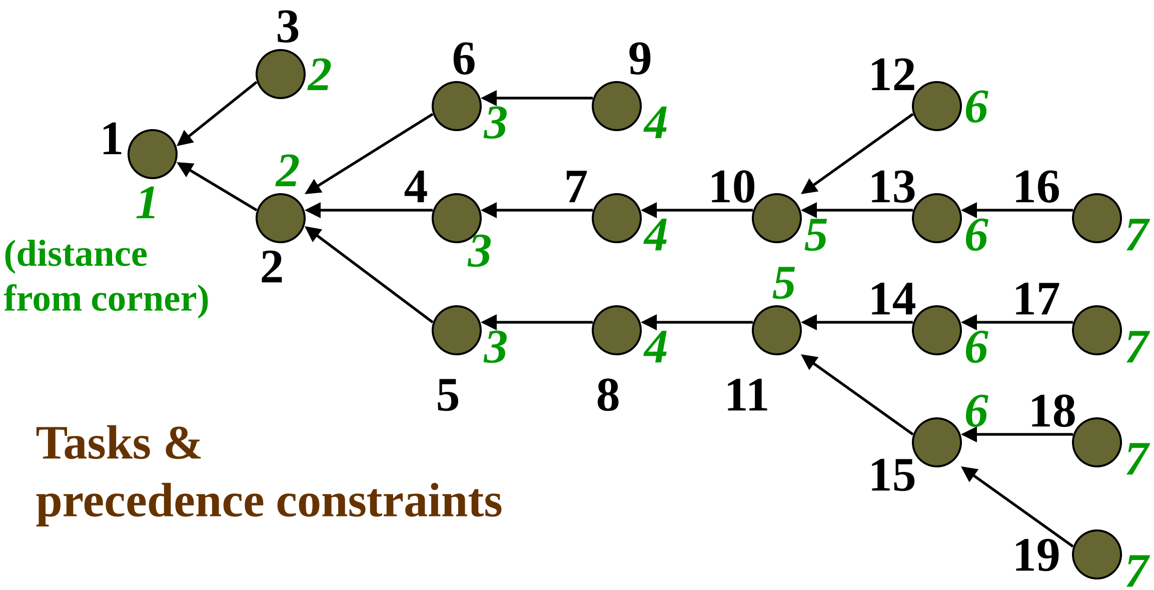


Bin-packing problem (illustration)



GOAL:
Usage of minimal number of containers

Scheduling problems: illustrative example for assembly process (algorithm of longest tails)



Tasks &
precedence constraints

GOAL:
Minimal total complete time

3 processors:

1	17	12	13	10	7			
2	18	16	15	11	8	4		
3	19	14	9	6	3	5	2	1

0 … 8 → t

Simple scheduling problems for one machine (processor): Problem Formulation

Initial set of elements: $R = \{ 1 , \dots , i , \dots , n \}$

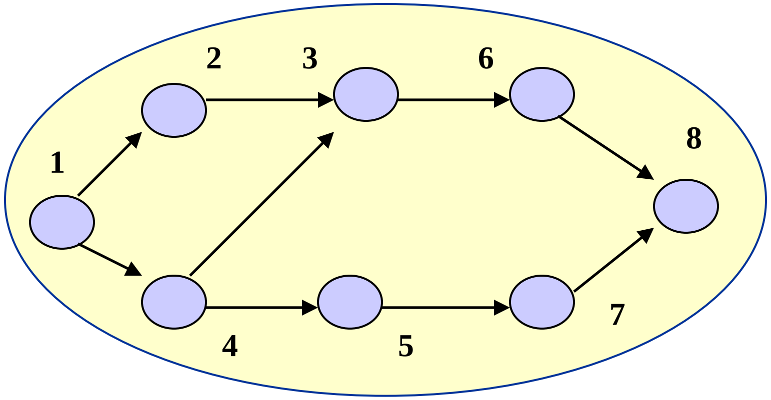
Schedule (linear ordering): $S = < s[i] , \dots , s[i] , \dots, s[n] >$

s{i} is the element number on position i in schedule S

f(S) is a real-value positive objective function

Problem is:

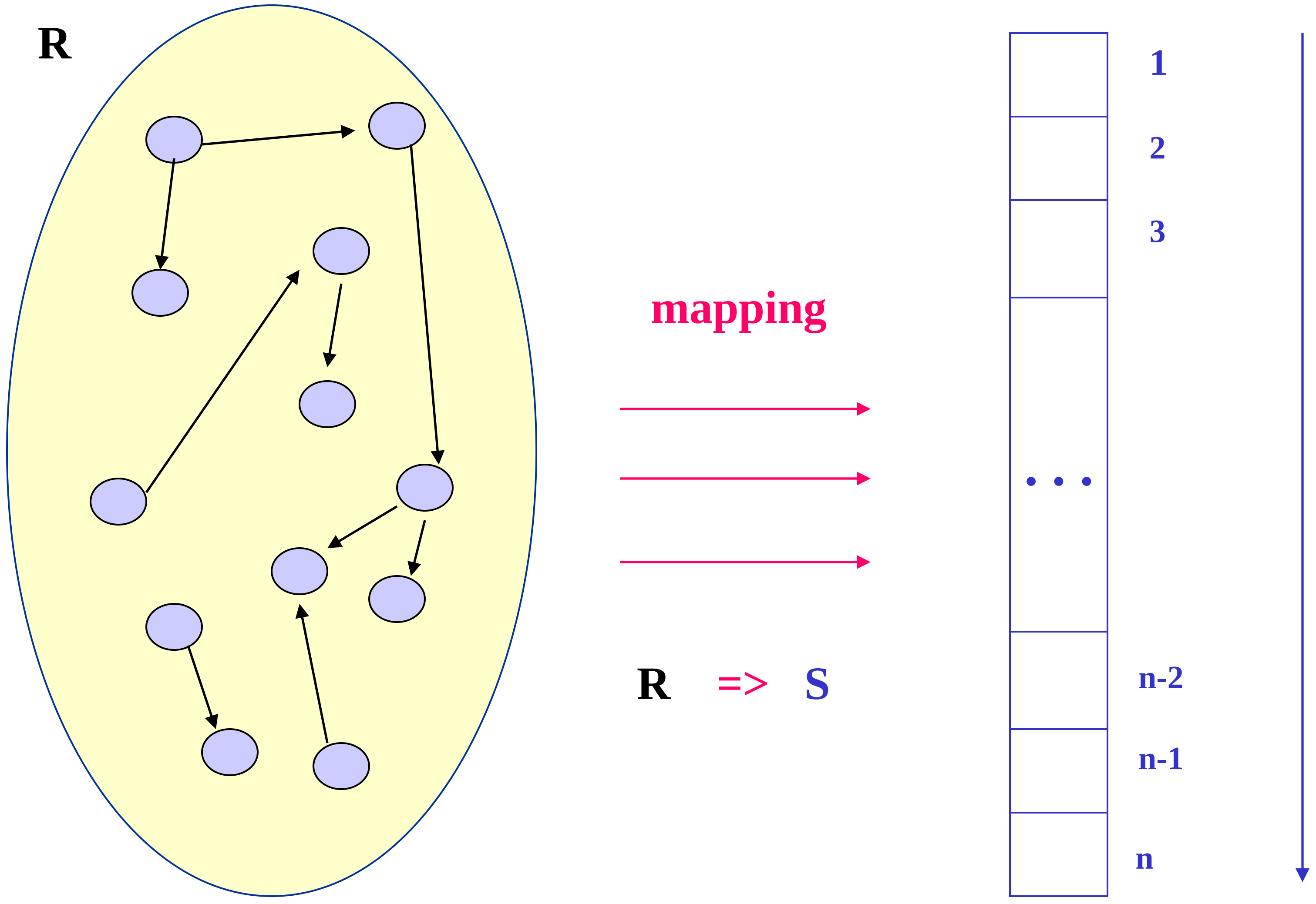
Find optimal schedule S*: $f(S^*) = \min f(S) \quad \forall S$



Precedence constraints: G = (R,E)
(usually: free-cycle)

E is mapping R into R

Basic Illustrative Figure



Basic Schedule Problem [Hardy, Littlewood, Polya, 1930]

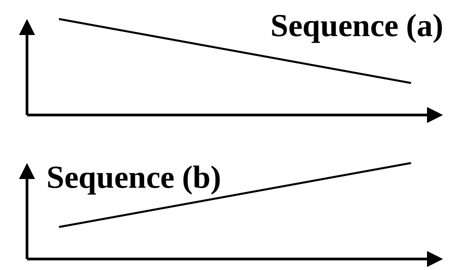
Sequence (a) = $\{ a_1, a_2, \dots , a_n \}$

Sequence (b) = $\{ b_1, b_2, \dots , b_n \}$

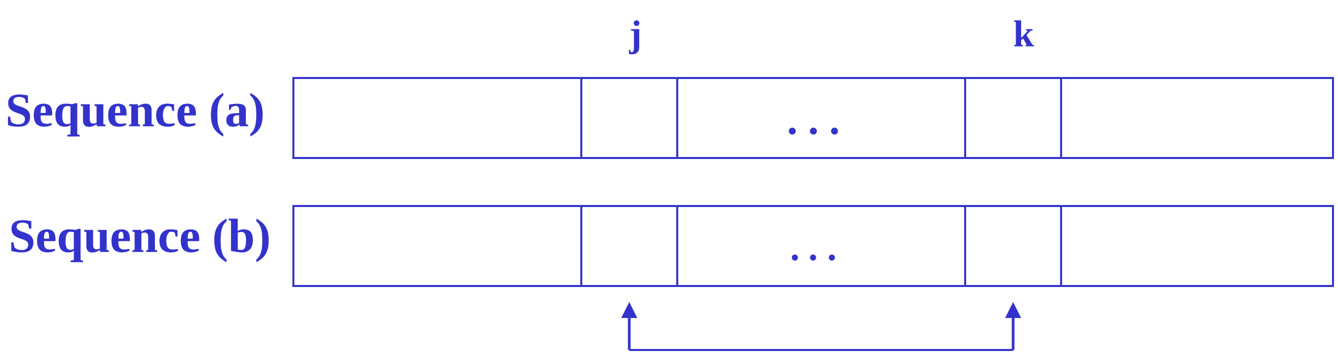
=>

$S = < b(s[1]), b(s[2], \dots , b(s[n]))>$
$f(S) = \sum^{n}_{i=1} a_i b(s[i])$ => min

LEMMA: S* is an optimal schedule if
(1) sequence (a) is ordered under non-increasing
(2) sequence (b) is ordered under non-decreasing



PROOF: Let $a_1 > a_2 > \dots > a_n$ & $b_1 < b_2 < \dots < b_n$
Let exist j and k that $a_j > a_k$ & $b_j > b_k$ $(j < k)$
Than we interchange b_j and b_k in sequence (b) and get:
$\Delta f = a_j b_k + a_k b_j - (a_j b_j + a_k b_k) = (a_j - a_k)(b_k - b_j) \leq 0.$



Basic Schedule Problem by Smith (1956)

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

$S = < s[1], \dots, s[i], \dots, s[n] >$

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

C_i is a completion time for job (task) i

$f_i(C_i) = a_i C_i + b_i$ (penalty function, $a_i > 0$)

THEOREM 1: S^* is an optimal schedule ($f(S^*) \leq f(S) \ \forall \ S$) if

1. exists a real value function $g(i,j)$ such that $g(i,j)<g(j,i) \Rightarrow f(S)<f(S^*)$

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

This is Problem 1 (P1)

Basic Schedule Problem by Tanaev (1966)

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

$S = < s[1], \ldots , s[i], \ldots , s[n] >$

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

C_i is a completion time for job (task) i

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

This is Problem 2 (P2)

Algorithm for Problems 1 and 2: non-decreasing of the indicator

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

Problem 2 P2: $\omega(i) = a_i \exp(\lambda \tau_i) / (1 - \exp \lambda \tau_i)$

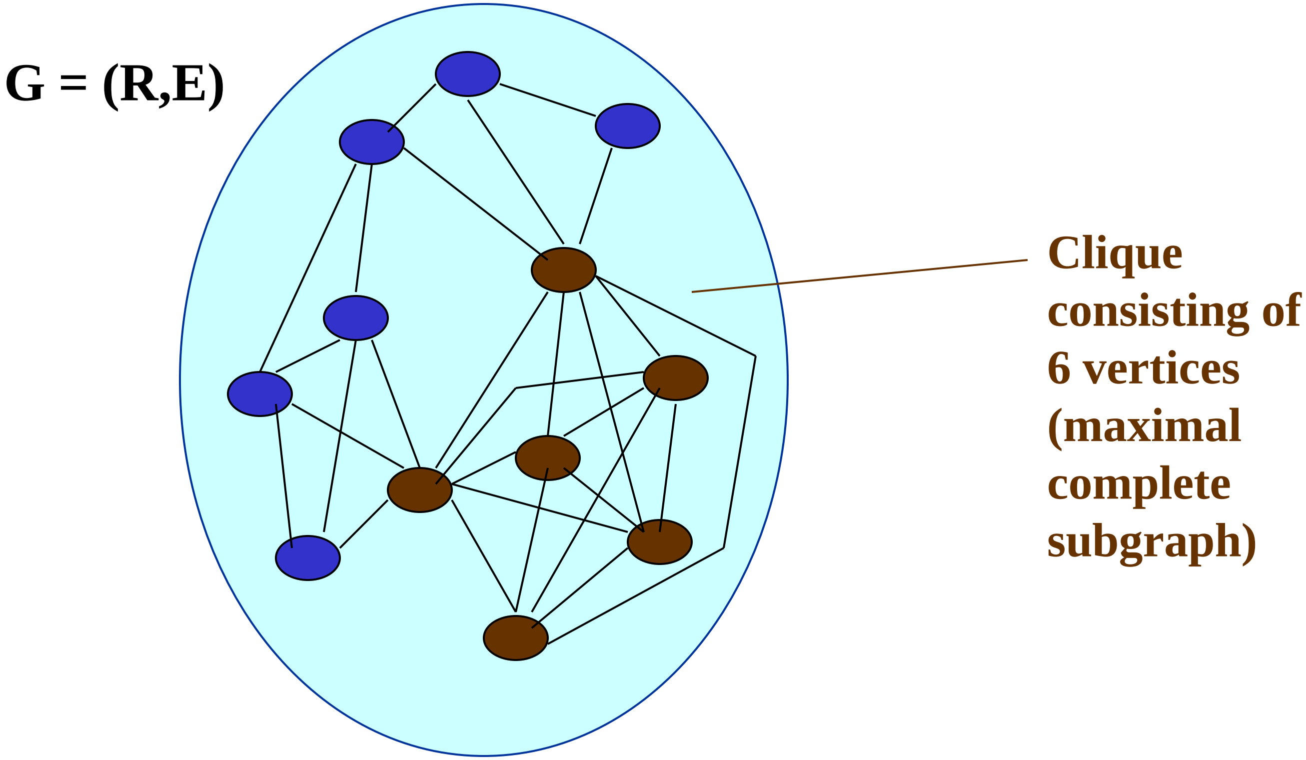
NOTE:

Considered algorithms (on the basis of ordering) can be used (an extended version) in the case of precedence constraints as tree or parallel-series graph

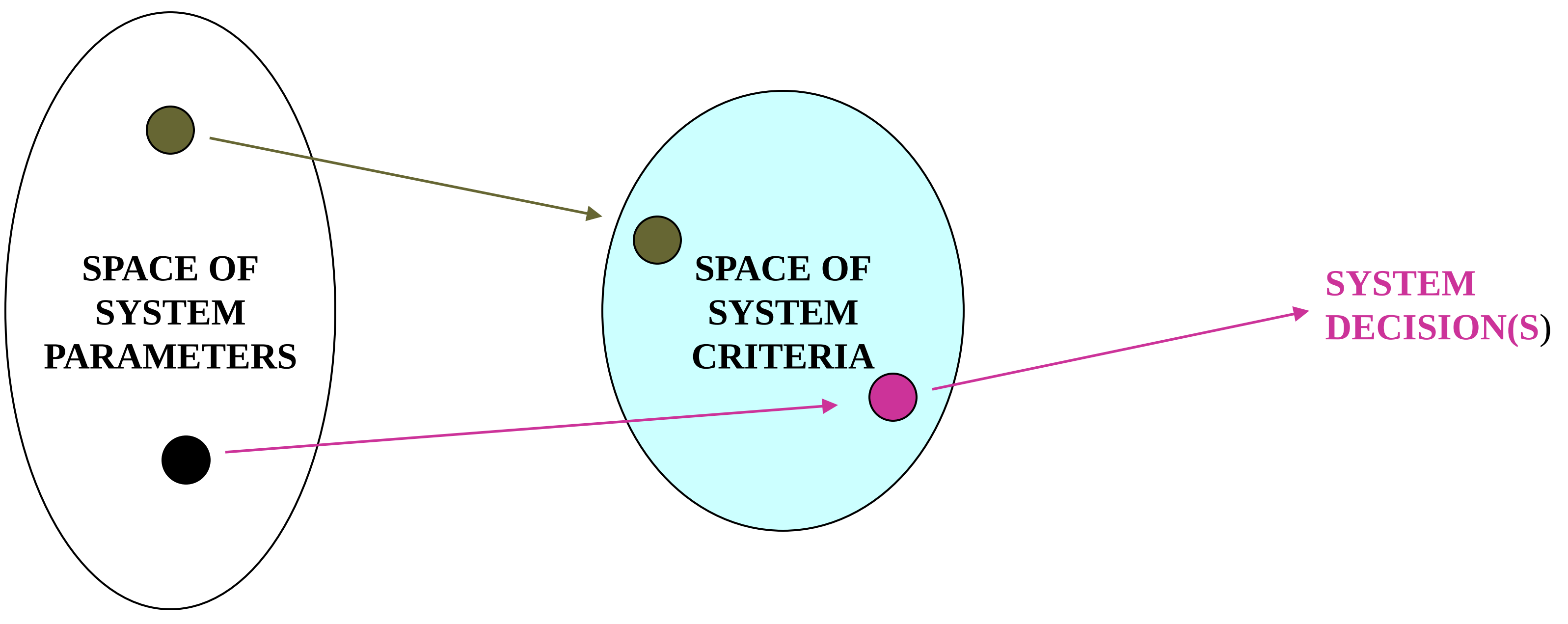
Maximal clique problem (illustration)

Initial graph G = (R, E), R is set of vertices, E is set of edges

Problem is: ***Find the maximal (by number of vertices) clique (i.e., complete subgraph)***



Scheme of multicriteria system design



Hierarchy of requirements / criteria

1.Ecology, politics

2.Economics, marketing

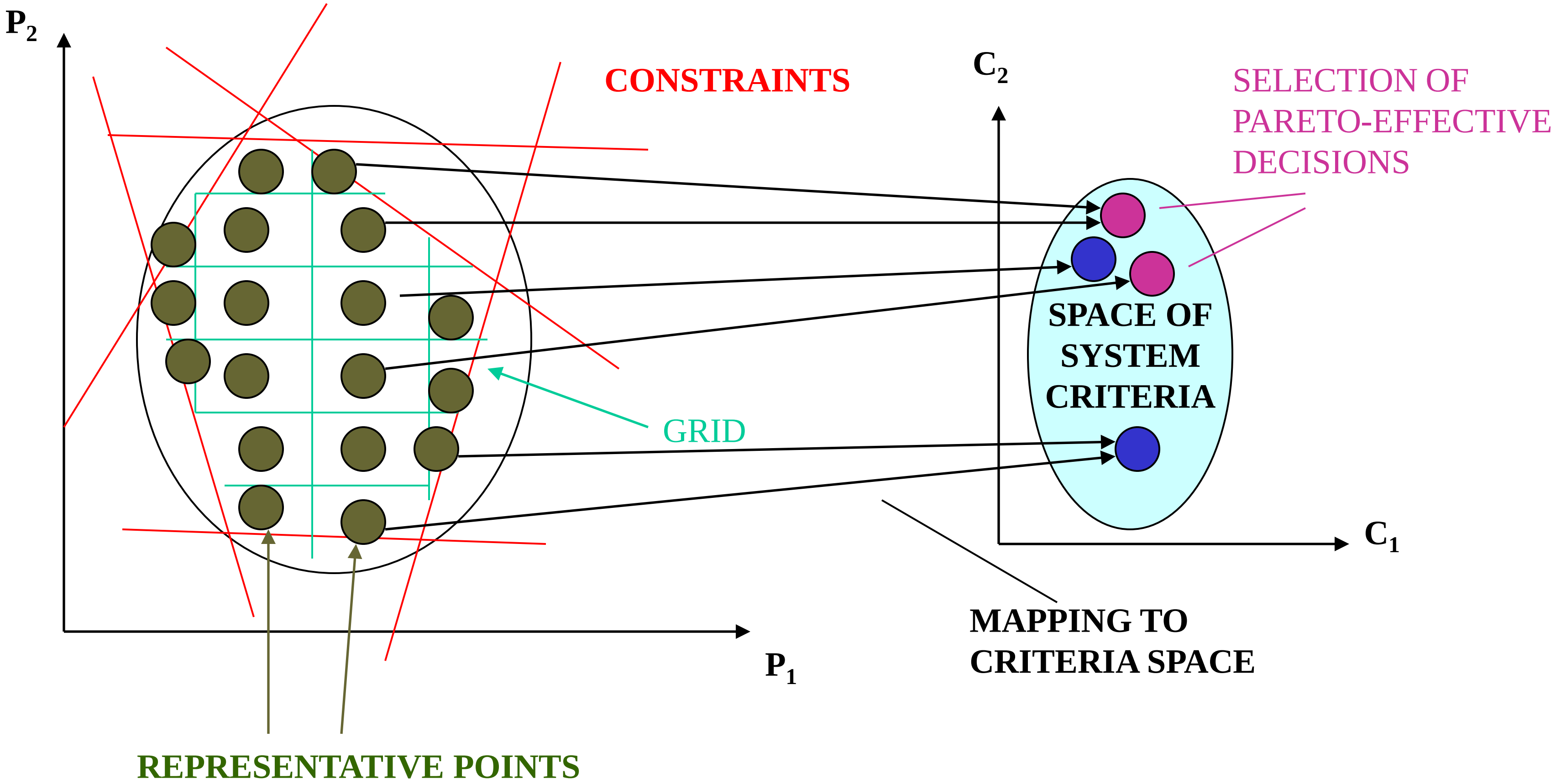
2.Technology (e.g., manufacturing issues, maintenance issues)

3.Engineering

SPACE OF SYTEM CRITERIA

SPACE OF SYTEM PARAMETERS

Scheme of multicriteria design PSI (parameters space investigation) (Sobol & Statnikov)



LECTURE 16. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Institute of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.16. Interchange techniques, genetic algorithm, etc.

PLAN:

1.Technical documentation (Russian experience as a set of basic documents)

2.Types of interchange techniques

3.Genetic algorithms

4.Multi-objective evolutionary optimization

5.Multidisciplinary optimization

6.Mixed Integer Nonlinear Programming

Oct. 8, 2004

Technical documentation (Russian experience)

BASIC VERSION

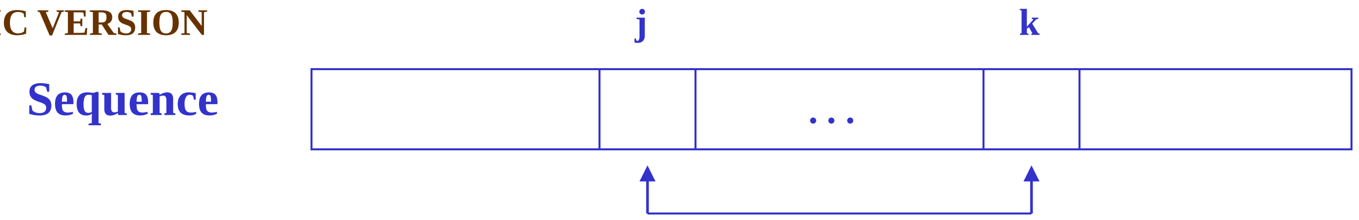
1. **Preliminary “avan”-project**
2. **”Avan”-project**
3. **Technical suggestion (proposal)**
4. **Technical project**
5. **Work-project**
6. **Report on the 1st-stage of utilization (including suggestion on system improvement)**
7. **Resultant report on utilization (including suggestion on system improvement)**

COMPESSED VERSION

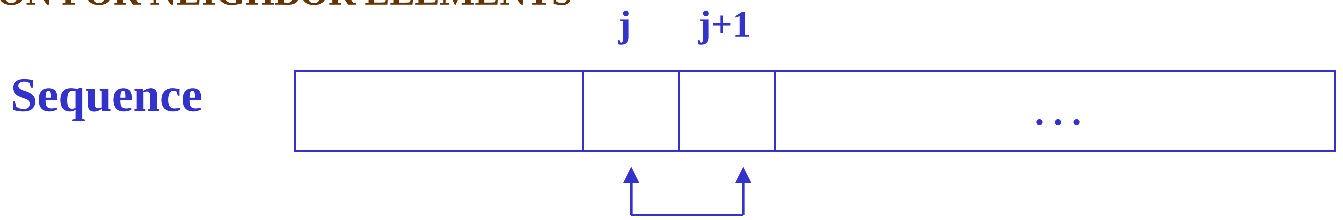
1. **Technical suggestion (proposal)**
2. **Technical-work project**
3. **Report on utilization**

Types of 2-Exchange Technique (Illustration)

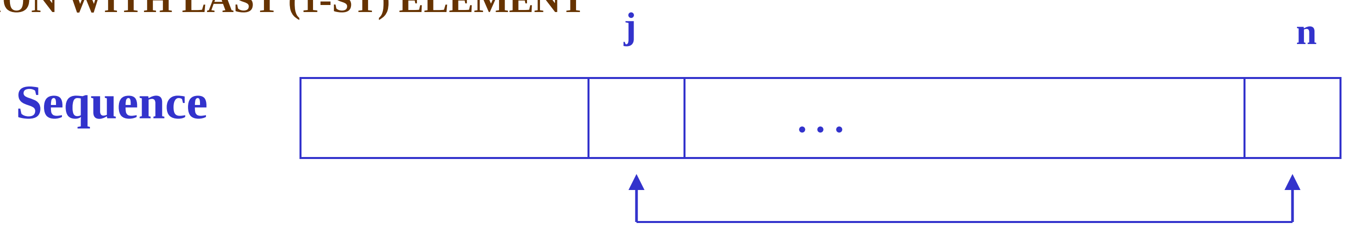
BASIC VERSION



VERSION FOR NEIGHBOR ELEMENTS

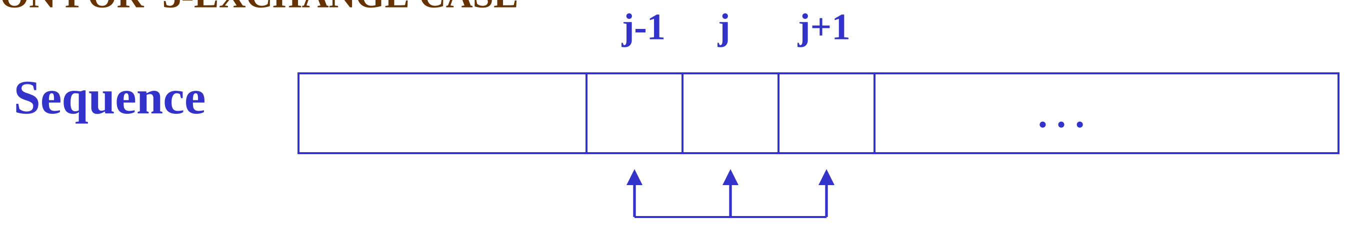


VERSION WITH LAST (1-ST) ELEMENT

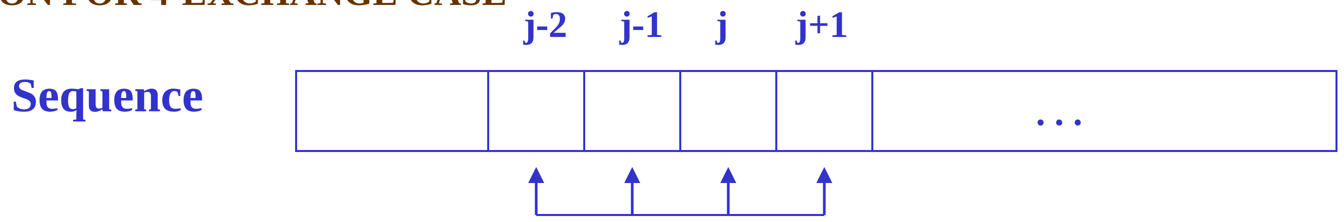


3-Exchange Technique & 4-Exchange Technique (Illustration)

VERSION FOR 3-EXCHANGE CASE



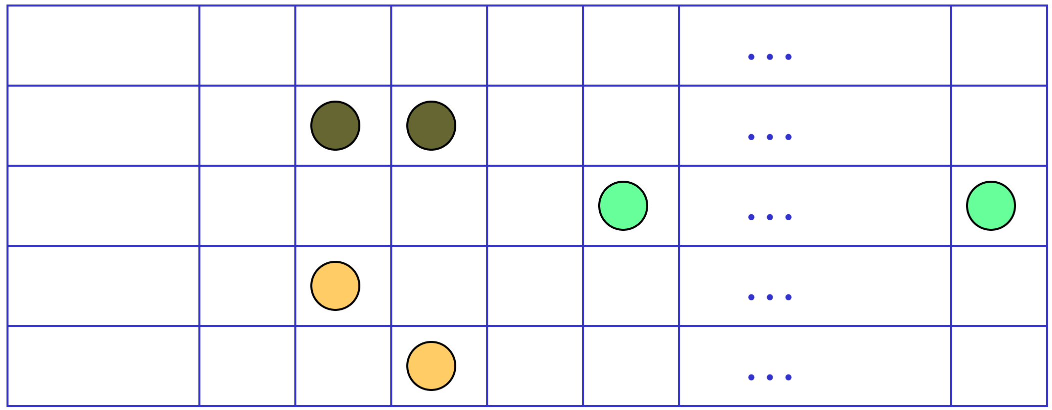
VERSION FOR 4-EXCHANGE CASE



Exchange Technique: Two-dimensional Case (Illustration)

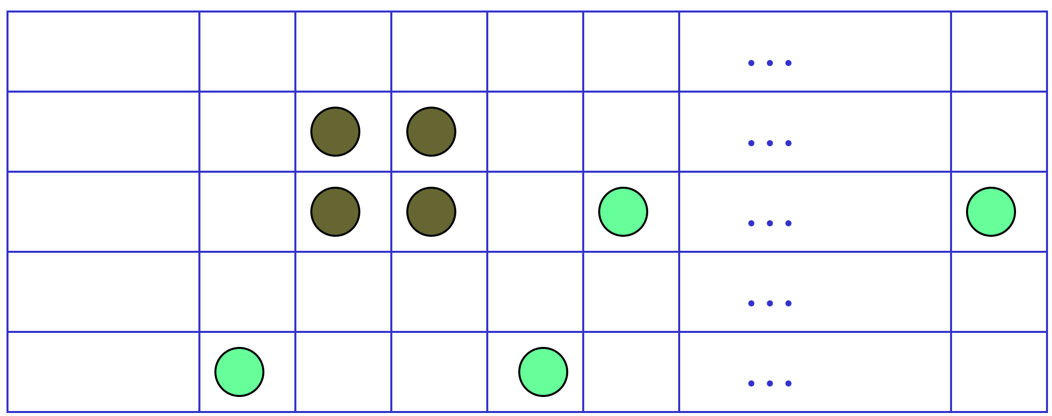
2-EXCHANGE CASE

Array



4-EXCHANGE CASE

Array



Basic knapsack problem is:

$$\textit{max} \quad \sum_{i=1}^{m} c_i \; x_i$$

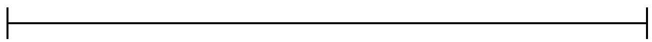
$$\textit{s.t.} \quad \sum_{i=1}^{m} a_i \; x_i \leq b$$

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

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

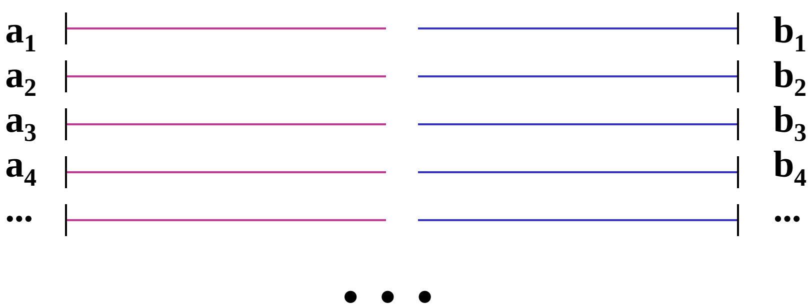
Genetic algorithms (illustration for knapsack problem)

STEP 1. AN INITIAL SOLUTION x_0

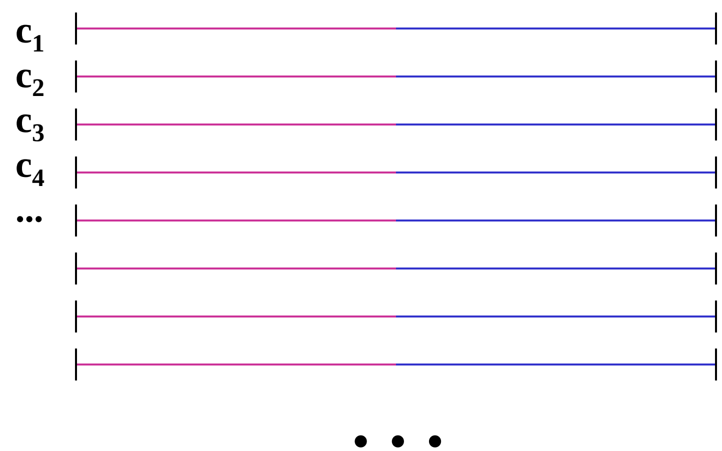


STEP 2. DIVIDING OF x_0 INTO 2 PARTS

STEP 3. MUTATION (GENERATION OF VERSIONS): *exchange of elements, *change of elements, etc.

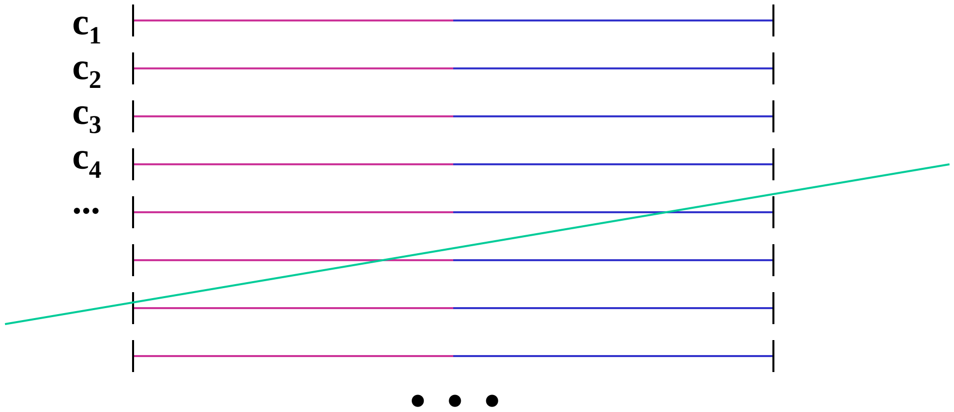


STEP 4. GENERATION OF NEW SOLUTIONS BY PAIRS (a_i, b_j)

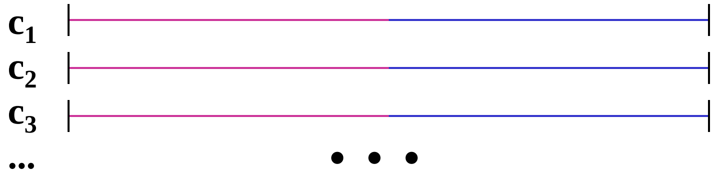


Genetic algorithms (illustration for knapsack problem)

STEP 5. Deletion of some solutions by resource constraint (≤ b)



STEP 6. Selection of the best solution(s)



Selection by two ways:

1.Selection by utility function

2.Selection of Pareto-effective solutions. This is

Multi-Objective Evolutionary Optimization

STEP 7. Repetion of steps 2, 3, 4, 5, and 6 for selected solutions

General optimization model
with constraints corresponding to certain disciplines
(e.g., weight, reliability, etc.):

max **f (x)** **(** ***or*** **extr f(x))**

subject to

$\varphi_1 (x) \leq W$ ***weight***

$B_1 \leq \varphi_2 (x) \leq B_2$ ***height***

$C_1 \leq \varphi_3 (x) \leq C_2$ ***temperature***

$D_1 \leq \varphi_4 (x) \leq D_2$ ***reliability***

. . .

$\varphi_k (x) \leq 0$

$\varphi_j (x)$ ***is a constraint function*** $(1 \leq j \leq k)$

Int. Society for Structural and Multidisciplinary Optimization
(civil engineering, ship engineering, marine engineering, aerospace engineering)
//www.issmo.org
Optimal design of structures (including issues of fluids)

Generalized optimization model that involves integer & continuous variables:

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

subject to

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

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

where **x** ***is vector of binary variables (selection of subsystems)***

y ***is vector of continuous variables / parameters (e.g., size)***

***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. ***Branch-And-Bound method***

2. ***Combinatorial hybrid techniques***

3. ***Gradient method***

4. ***Interior point method***

LECTURE 17-18. Course: “Design of Systems: Structural Approach”

Dept. “Communication Networks &Systems”, Faculty of Radioengineering & Cybernetics

Moscow Institute of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.17. Basic models of combinatorial optimization III.

L.18. Basic models of combinatorial optimization IV.

PLAN:

1.Spanning (illustration): *spanning tree, *minimal Steiner problem, *2-connected graph

2.TSP, assignment (formulations)

3.Multple matching (illustration) , usage in processing of experimental data

4.Graph coloring problem, covering problems (illustration and applications)

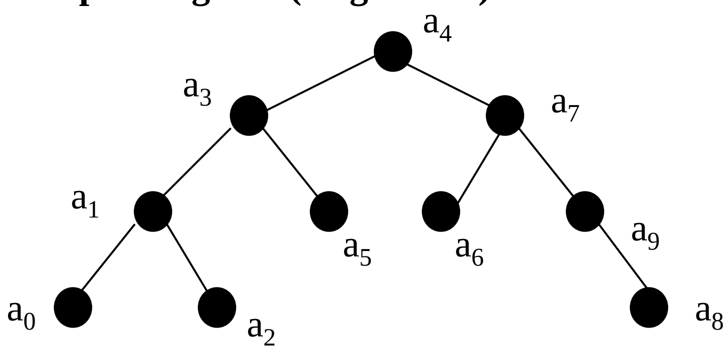
5.Alignment, maximal substructure, minimal superstructure (illustration, applications)

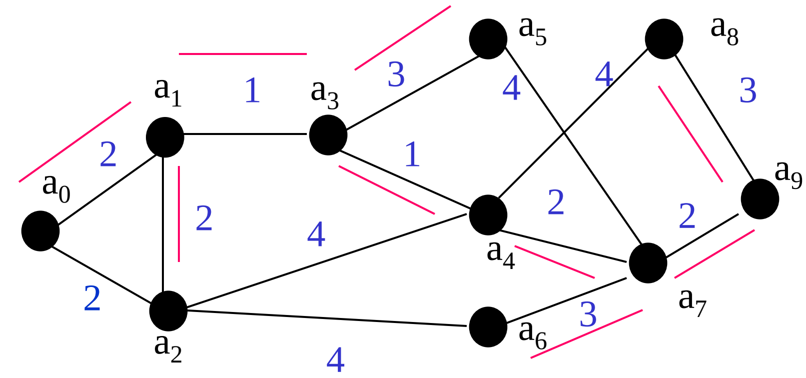
6.Timetabling

Oct. 9, 2004

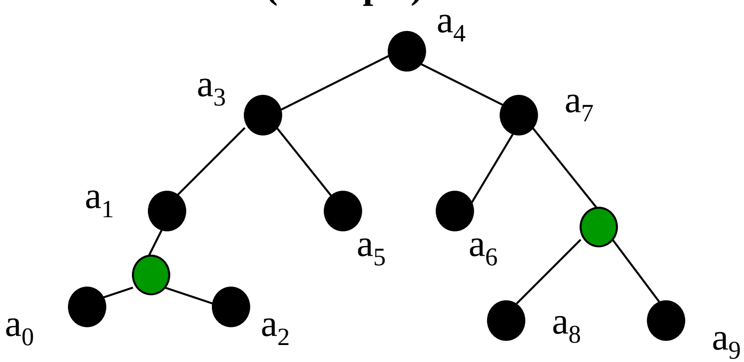
Spanning (illustration): 1-connected graph

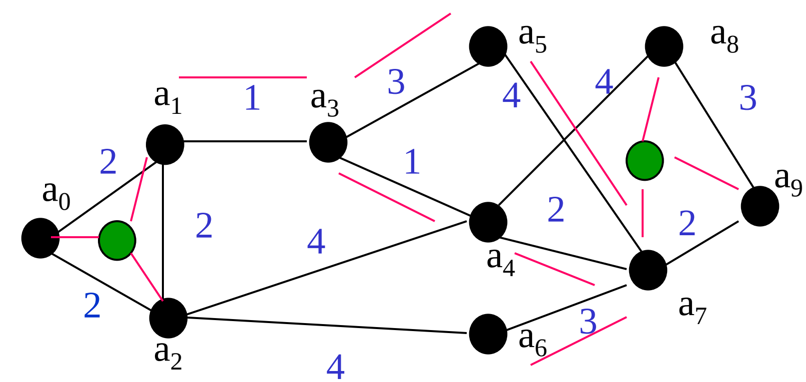
Spanning tree (length = 19):



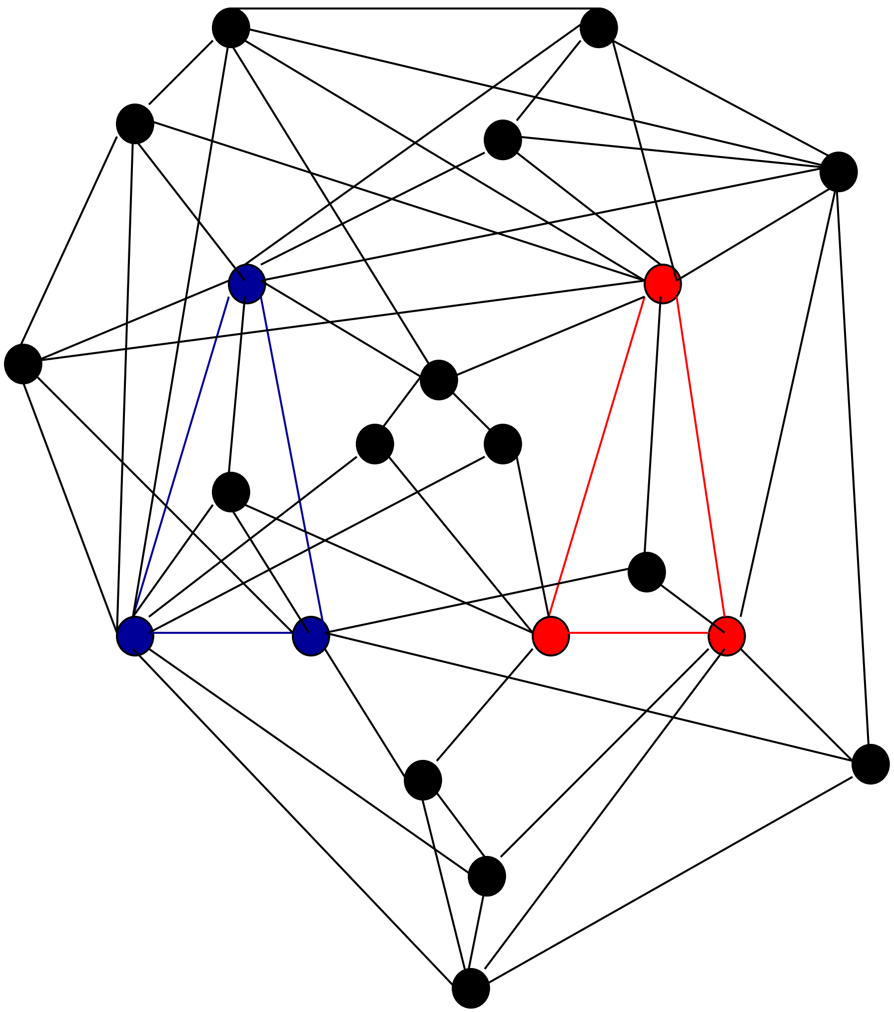


Steiner tree (example):



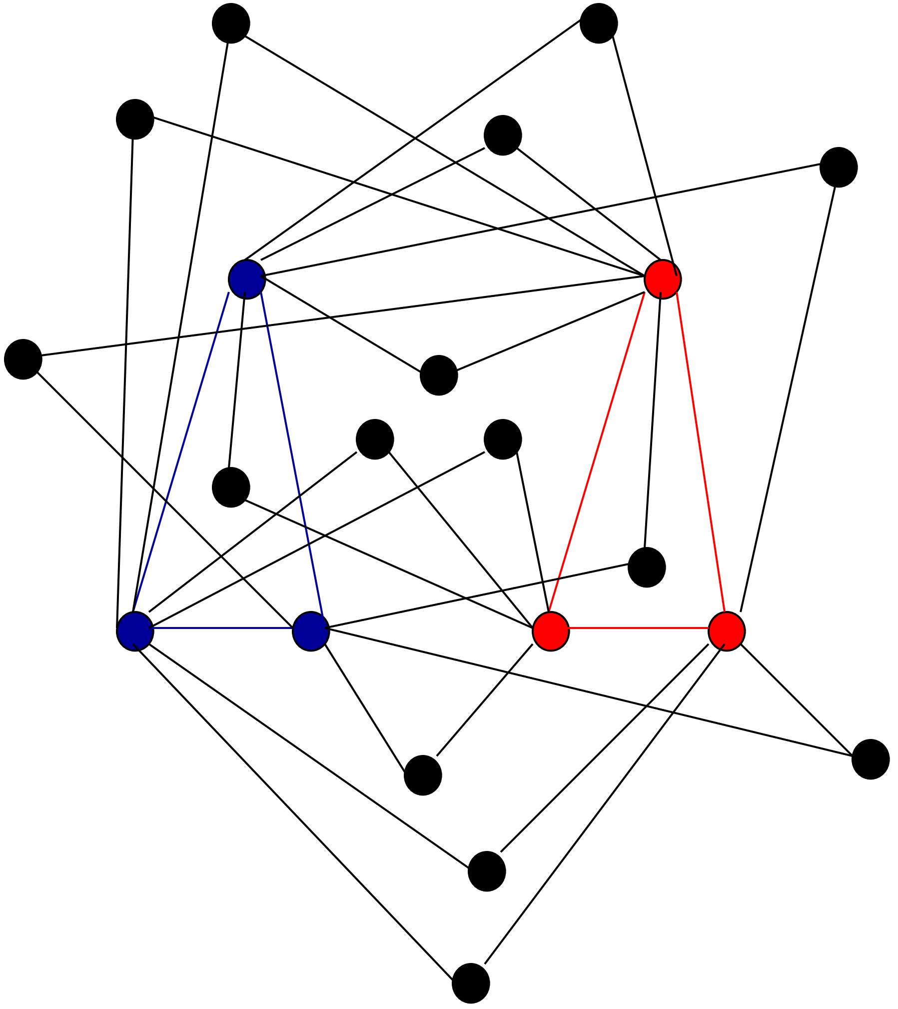


Spanning by two-connected graph:
Revelation of two 3-node cliques (centers)

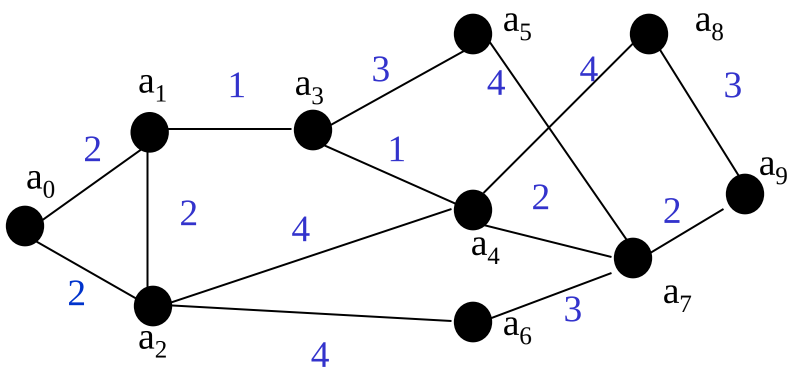


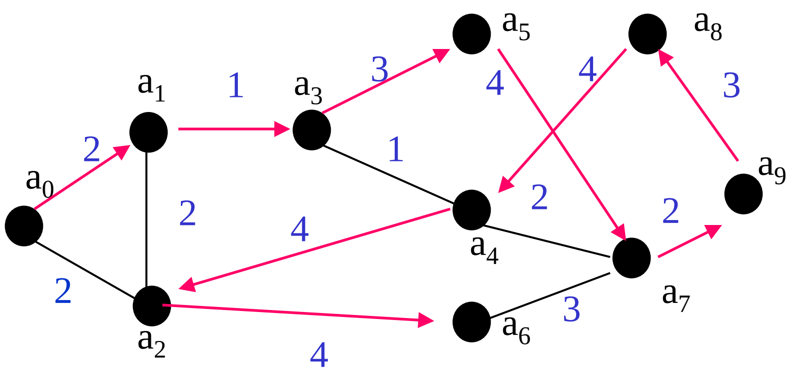
Spanning by two-connected graph:

Connection of each other node
with the two centers



Traveling salesman problem





L = < $a_0,a_1,a_3,a_5,a_7,a_9,a_8,a_4,a_2,a_6$>

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

FORMULATION:

Set of cities: $A = \{ a_1 , \ldots , a_i , \ldots , a_n \}$

Distance between cities i and j : $\rho (a_i , a_j)$

$\prod$ is set of permutations of elements of A, permutation

$s^* = < a(s^*[1]), \ldots ,a(s^*[i]), \ldots ,a(s^*[n]) >$

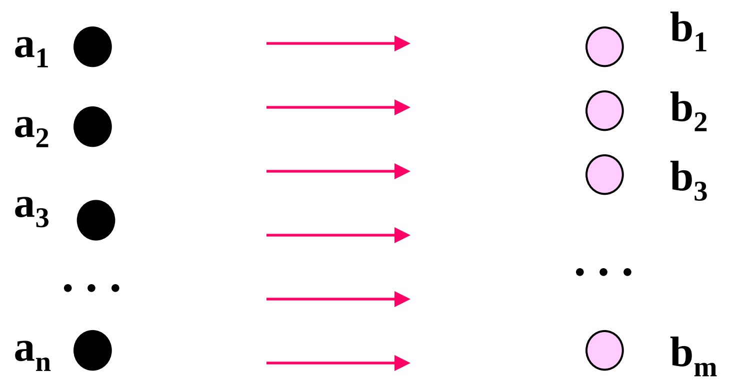
$$\min_{(s \in \prod)} f(s)=f(s^*)$$

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

ALGORITMS:
1.Greedy algorithm
2.On the basis of minimal spanning tree
3.Branch-And-Bound
Etc.

VERSIONS (many):
1.Cycle or None
2.m-salesmen
3.asymetric one (*i.e., distances* $\rho(a_i, a_j)$ *and* $\rho(a_j, a_i)$ *are different ones*)
4.Various spaces (metrical space, etc.)
5.Multicriteria problems
Etc.

Assignment problem



FORMULATION:
Set of elements: A = { a_1 , … , a_i , … , a_n }
Set of positions B = { b_1 , … , b_j , … . b_m } (now let n = m)
Effectiveness of pair i and j is: z (a_i , b_j)
∏ = {s} is set of permutations (assignment) of elements of A
into position set B:
s* = < (s*[1]), … ,(s*[i]), … ,(s*[n]) > , i.e., element a_i into position s[i] in B
The goal is:

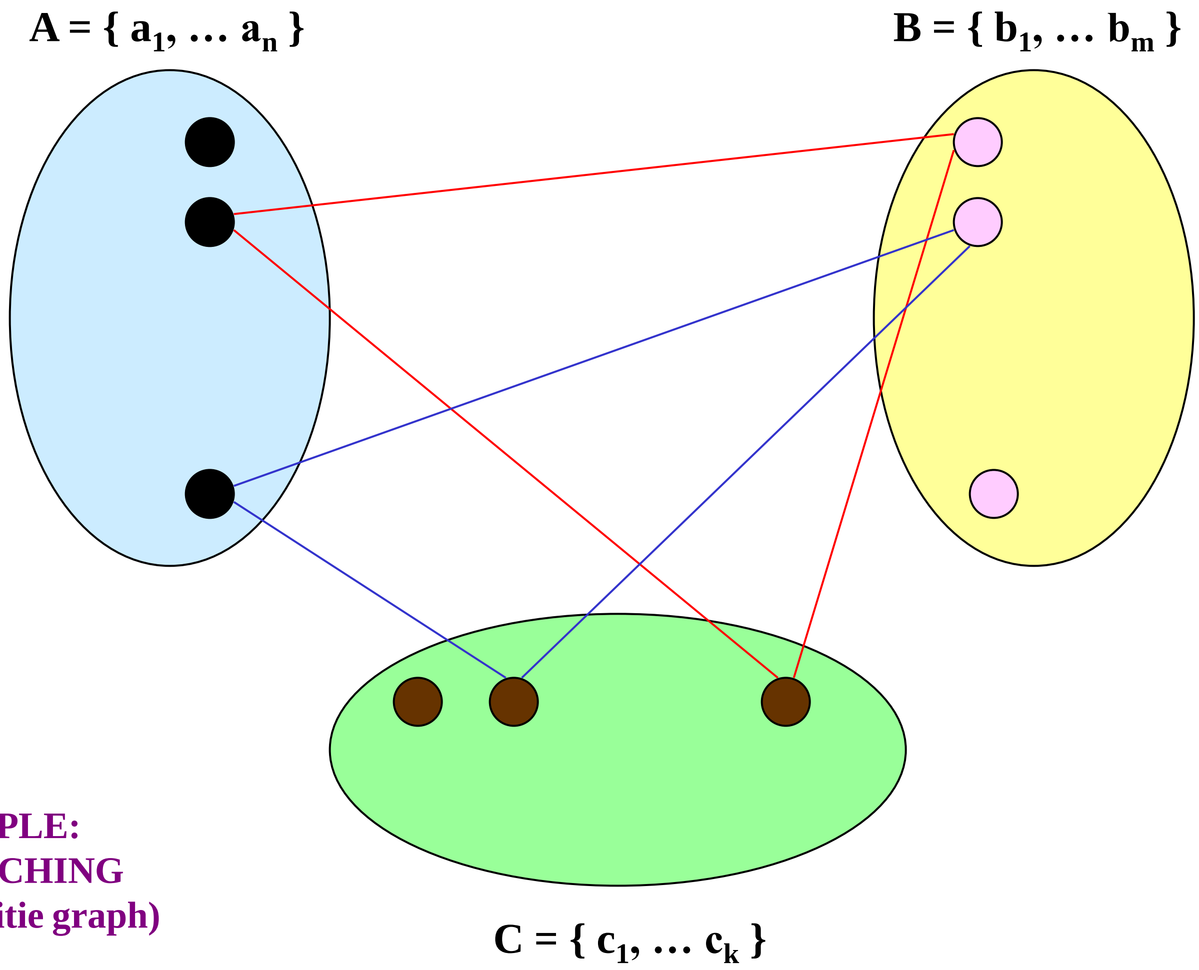
max $\sum^{n}_{i=1}$ z (i, s[i])

Assignment problem

ALGORITMS:
1.Polynomial algorithm ($O(n^3)$)

VERSIONS:
1.*Min max* problem
2.Multicriteria problems
Etc.

Multiple matching problem



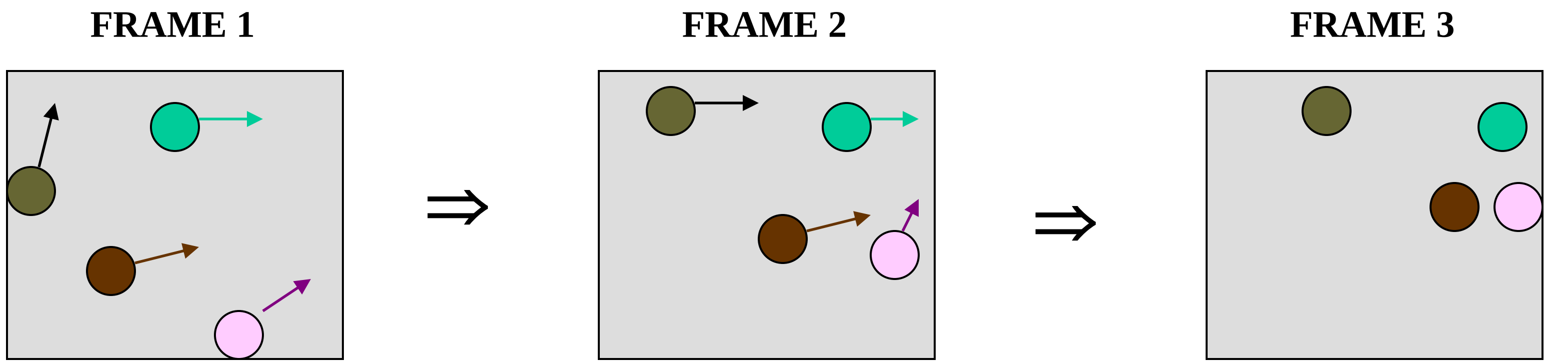
EXAMPLE:
3-MATCHING
(3-partitie graph)

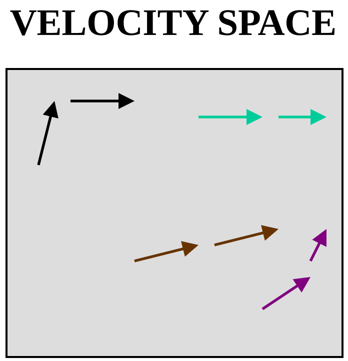
Multiple matching problem

ALGORITMS:
1.Heursitcs
(e.g., greedy algorithms, local optimization, hybrid heuristics)
2.Enumerative algorithms (e.g., Branch-And-Bound method)
3.Morphological approach

VERSIONS:
1.Dynamical problem (multiple track assignment)
2.Problem with errors
4.Problem with uncertainty (probabilistic estimates, fuzzy sets)
Etc.

Recent applied example: usage of assignment problem(s) to define velocity of particles





Recent applied example: usage of assignment problem(s) to define velocity of particles

MODELS & ALGORITMS:
1.Correlation functions (from radioengineering: signal processing)
2.Assignment problem between two neighbor frames
(algorithm schemes: genetic algorithms, other algorithms for assignment problems, hybrid schemes)
3.Multistage assignment problem (e.g., examination of 3 frames, etc.)
(algorithm schemes: genetic algorithms, other algorithms for assignment problems, hybrid schemes)

VERSIONS :
1.Basic problem
2.With errors
3.Under uncertainty
Etc.

Recent applied example: usage of assignment problem(s) to define velocity of particles

APPLICATIONS (air/ water environments):

1. **Physical experiments**
2. **Climat science**
3. **Chemical processes**
4. **Biotechnological processes**

Contemporary sources:

1. **PIV systems (laser/optical systems)**
2. **Sattelite photos**
3. **Electronic microscope**

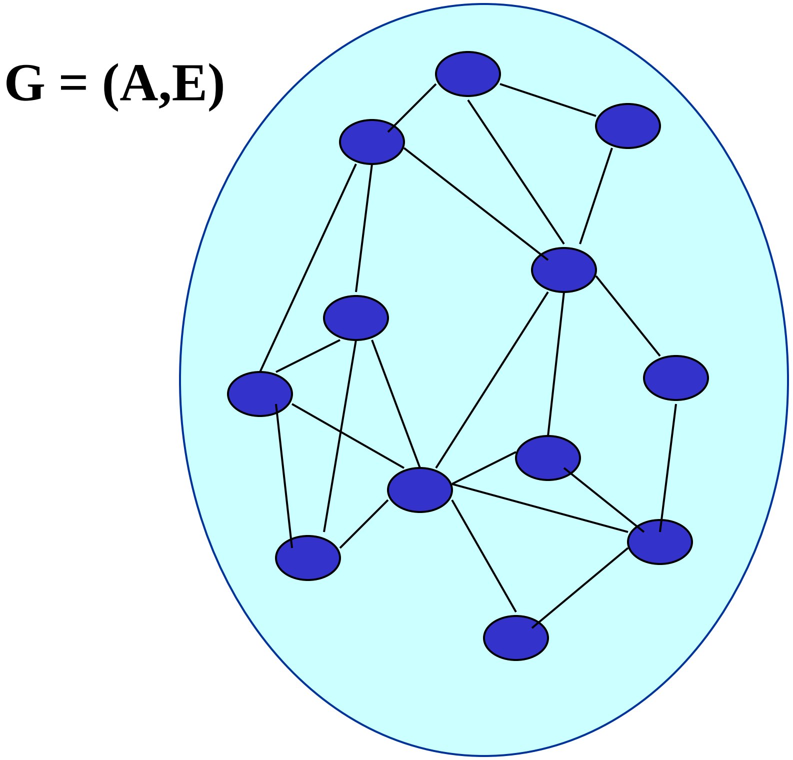
Etc.

Graph coloring problem (illustration)

Initial graph G = (A, E), A is set of vertices, E is set of edges

Problem is:

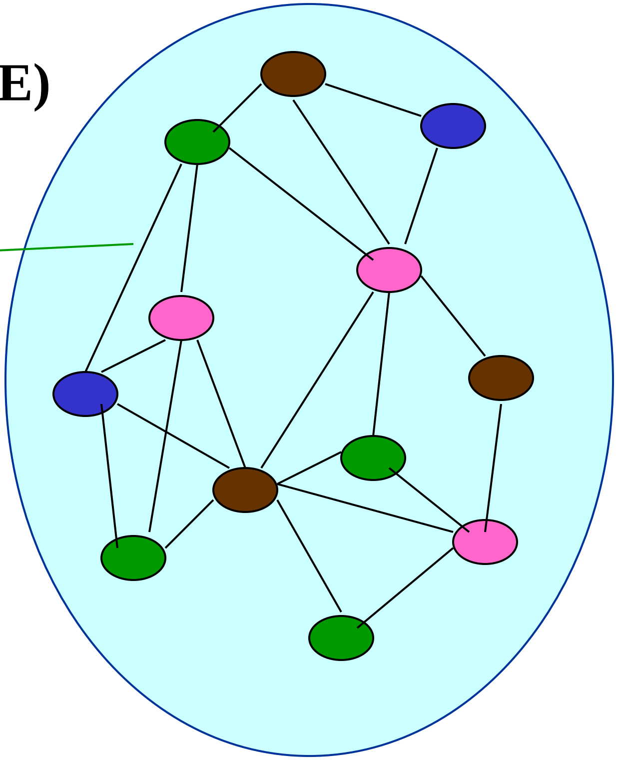
Assign a color for each vertex with minimal number of colors under constraint: neighbor vertices have to have different colors



Graph coloring problem (illustration)

G = (A,E)

Right coloring



Graph coloring problem (illustration)

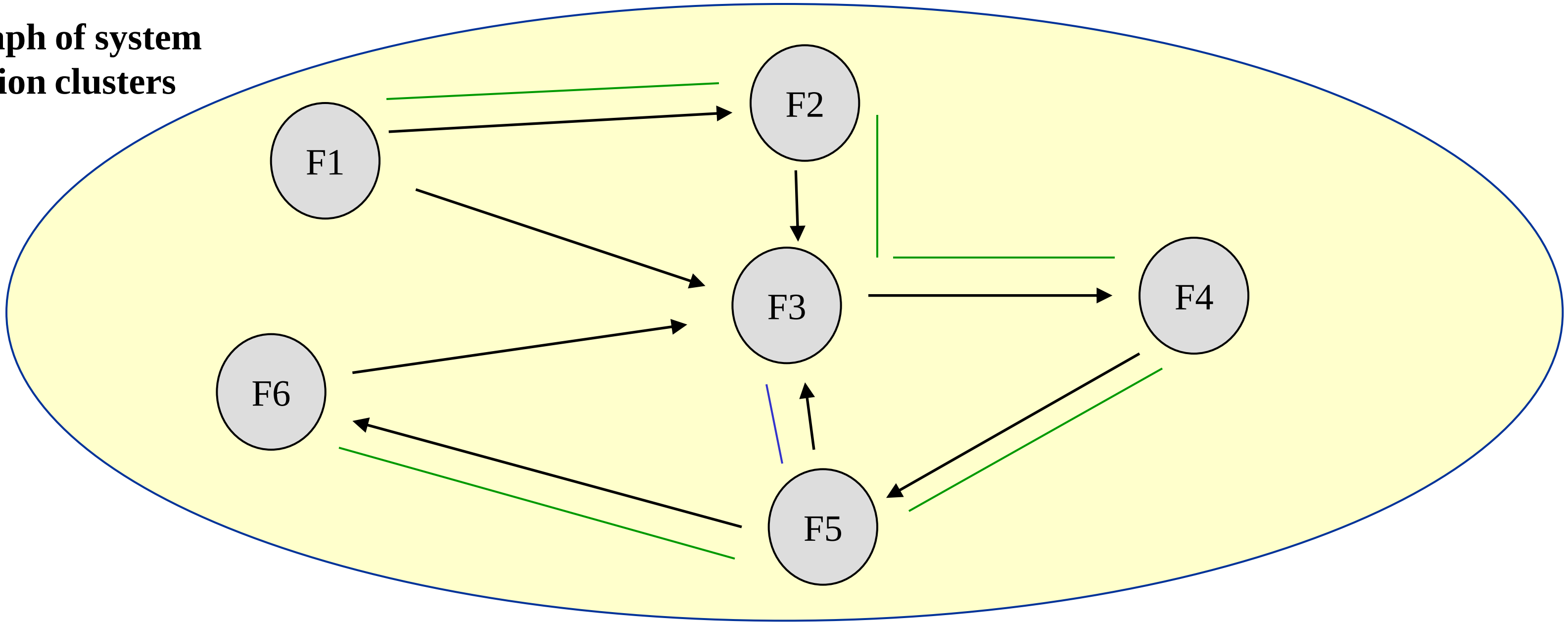
APPLICATIONS:

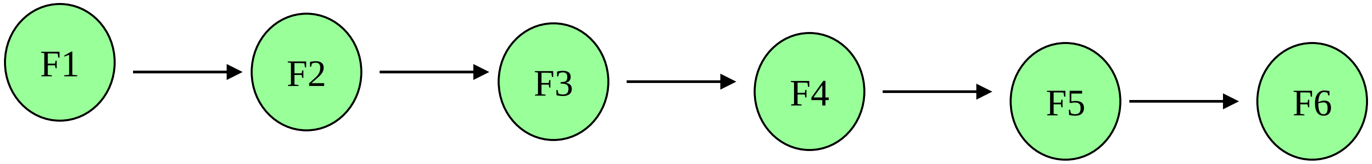
1. **Assignment of registers in compilation process (A.P. Ershov, 1959)**
2. **Frequency allocation / channel assignment**
   (static problem, dynamic problem, etc.)
3. **VLSI design**

   etc.

Example: system function clusters and covering by chains (covering of vertices)

Digraph of system function clusters



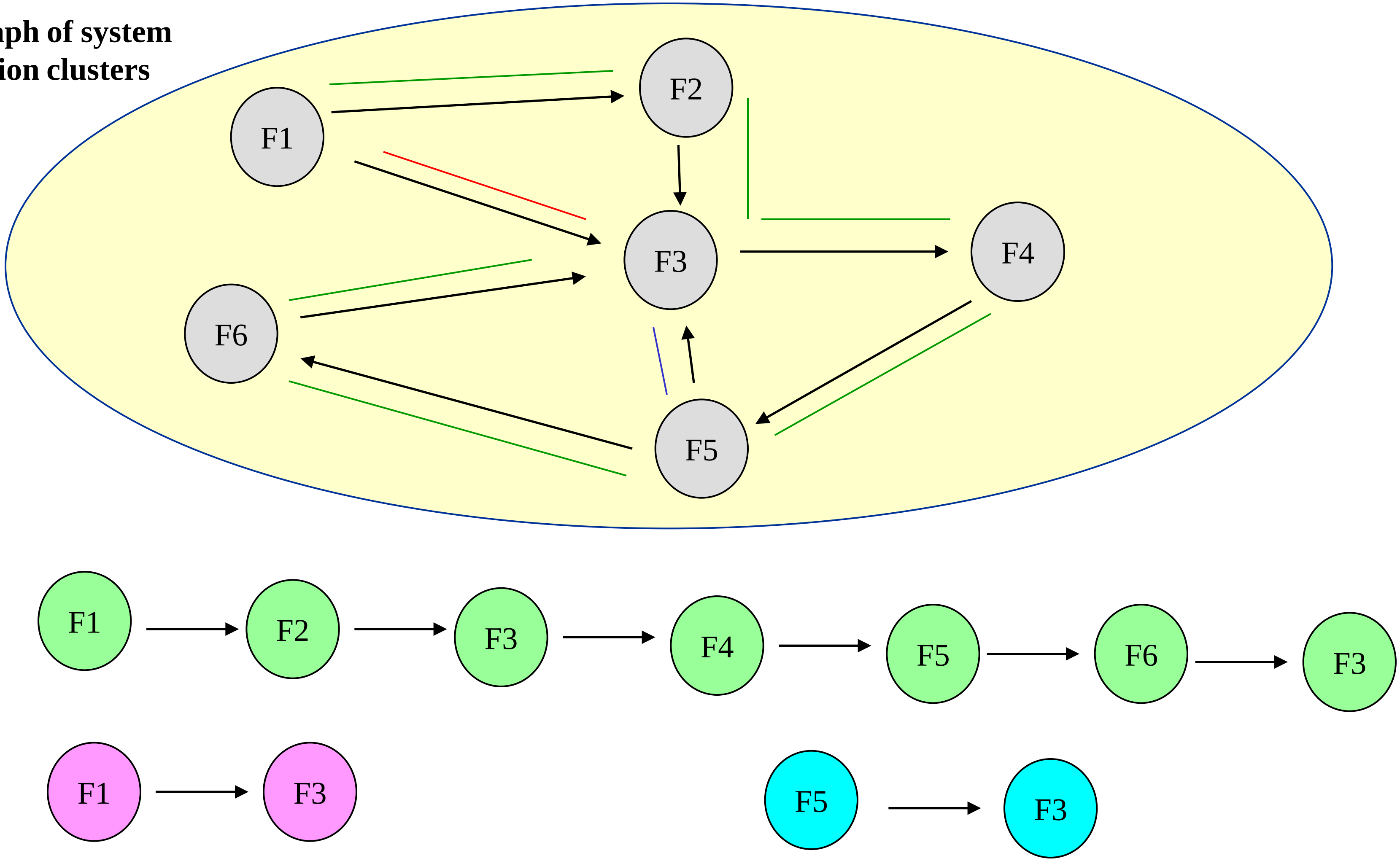


THE LONGEST PATH

Application: system testing

Example: system function clusters and covering by chains (covering of arcs)

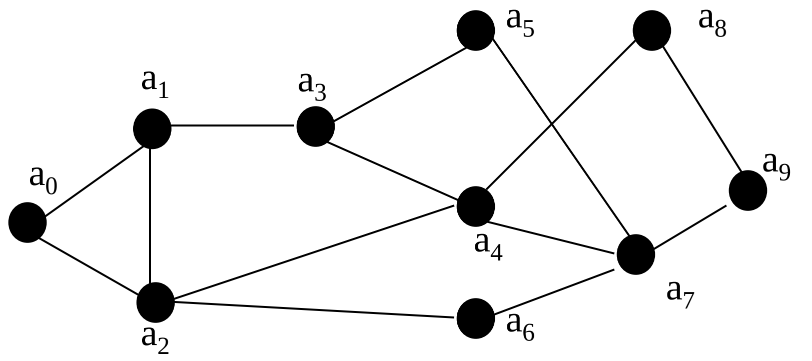
Digraph of system function clusters



APPLICATION: TESTING OF "CHANGES"

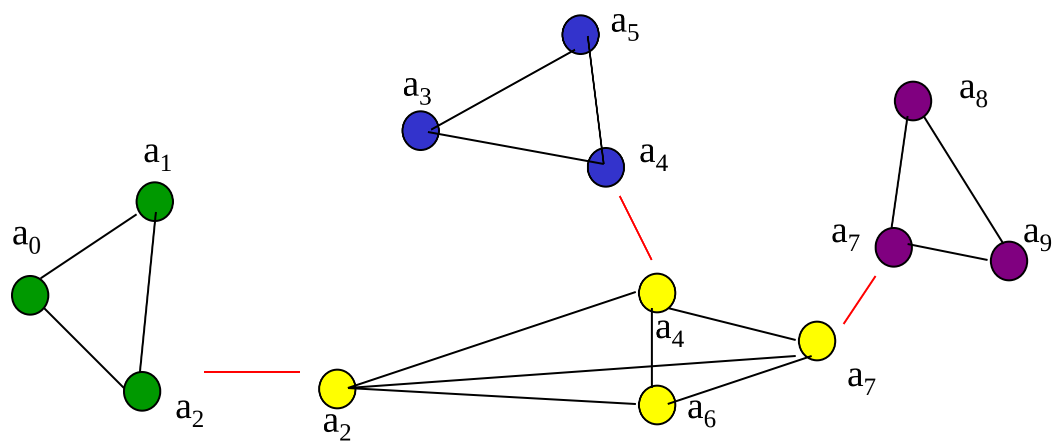
Illustration: covering by cliques

Basic graph



Cliques (a version):

$C_1 = \{ a_0 , a_1 , a_2 \}$
$C_2 = \{ a_3 , a_5 , a_4 \}$
$C_3 = \{ a_7 , a_8 , a_9 \}$
$C_4 = \{ a_2 , a_4 , a_6 , a_7 \}$



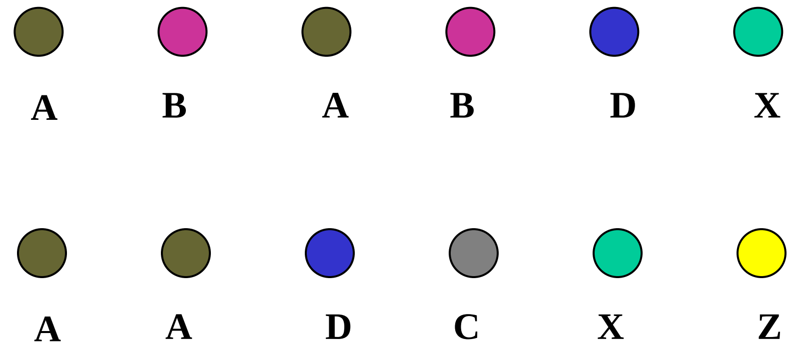
APPLICATION: ALLOCATION OF SERVICE (e.g., communication centers)

Alignment (illustration)

CASE OF 2 WORDS:

Word 1

Word 2



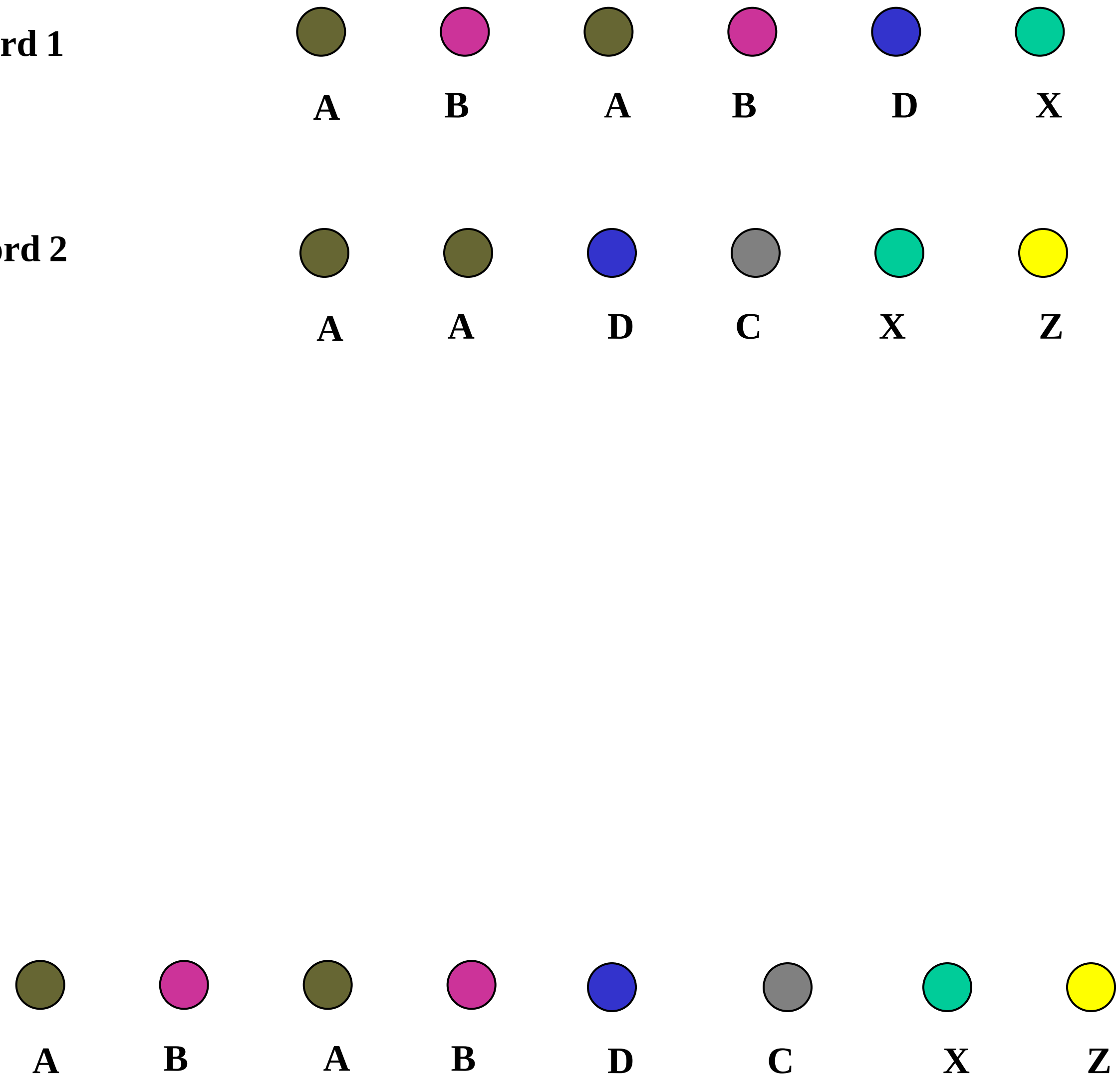
ALIGNMENT PROBLEM: ***minimal additional elements***

Alignment (illustration)

CASE OF 2 WORDS:

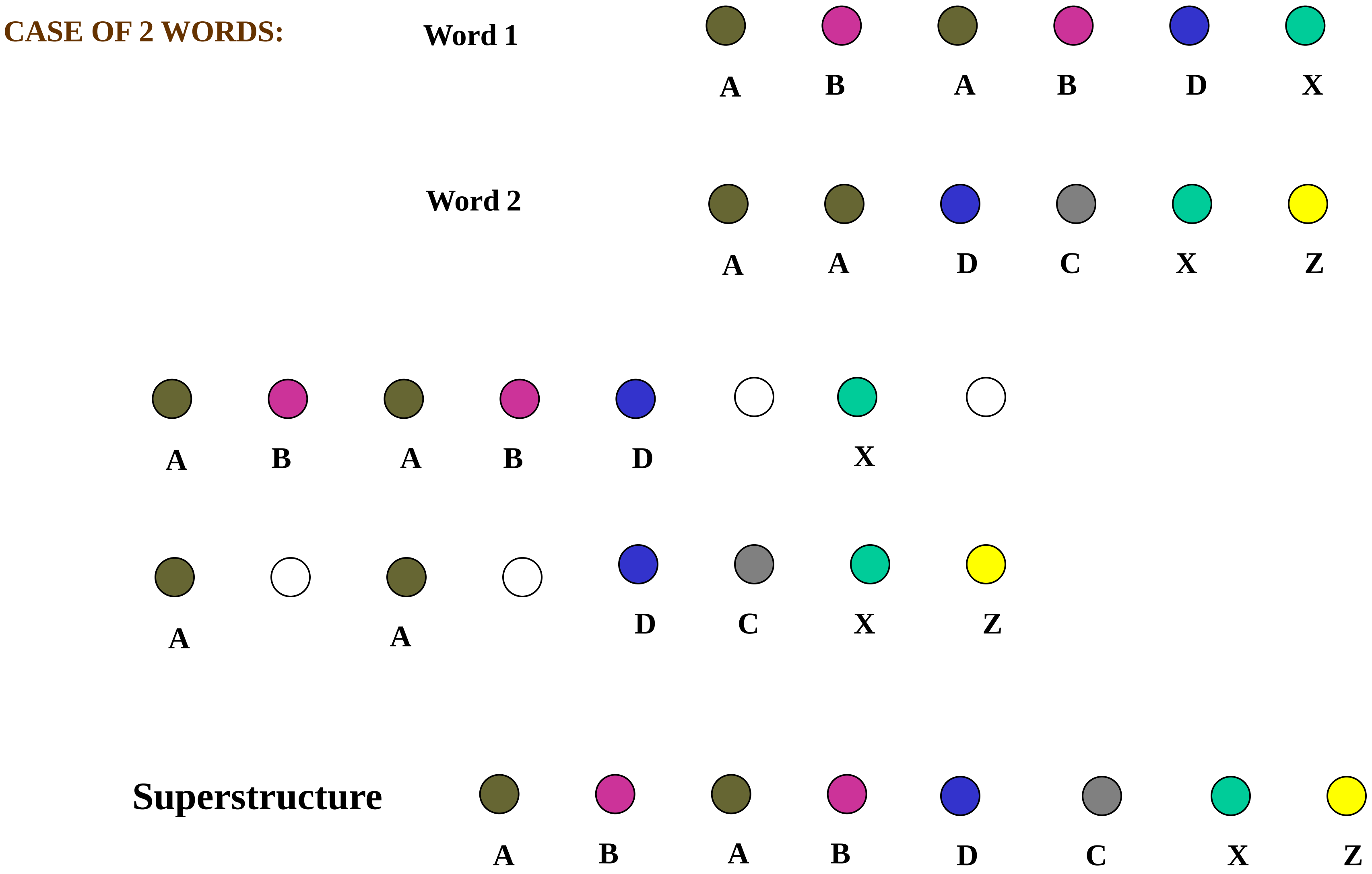
Word 1

Word 2



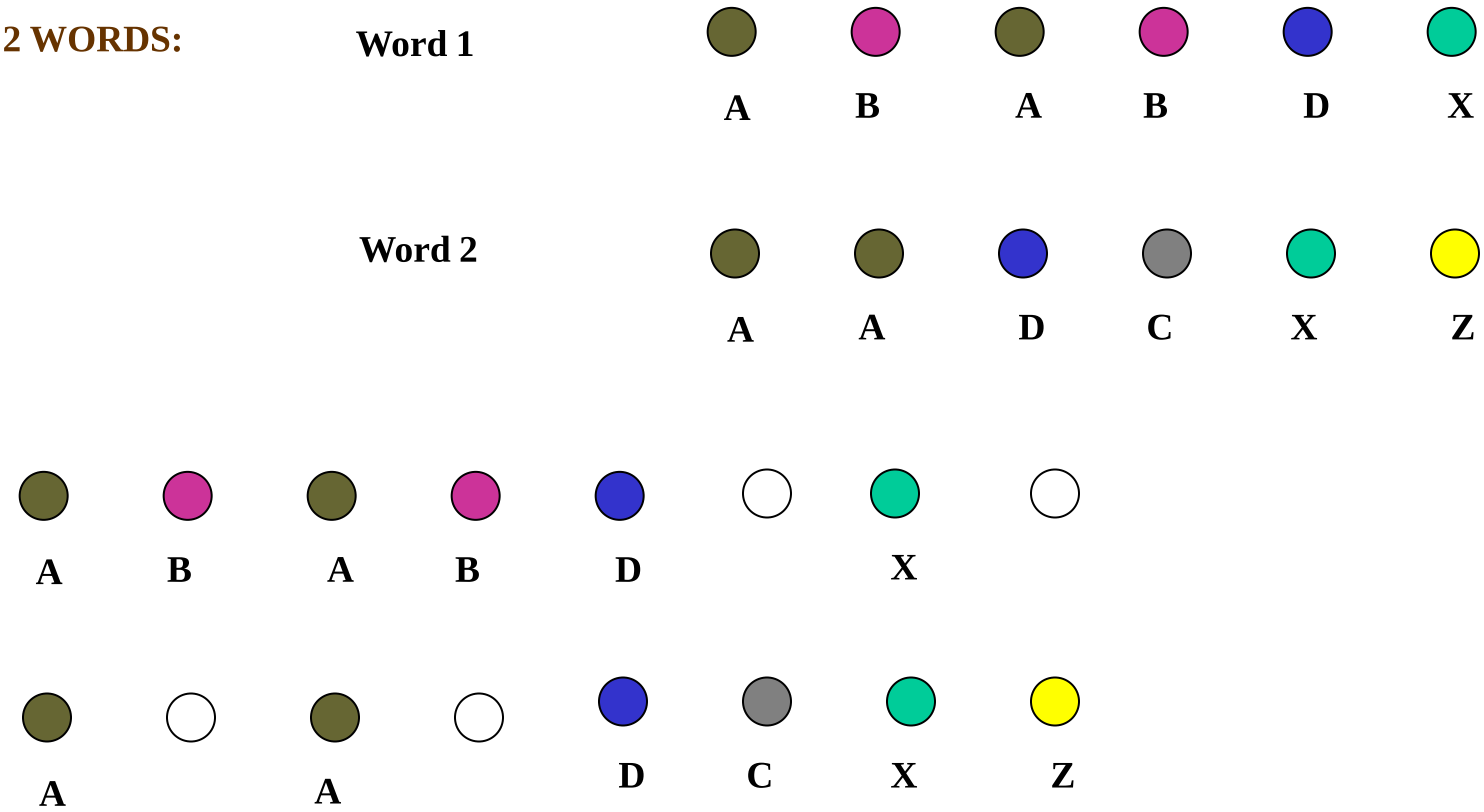
Minimal **Superstructure**

Alignment (illustration)



Alignment (illustration)

CASE OF 2 WORDS:



APPLICATIONS:
1. Linguistics
2. Bioinformatics (gene analysis, etc.)
3. Processing of frame sequences (image processing)
4. Modeling of conveyor-like manufacturing systems

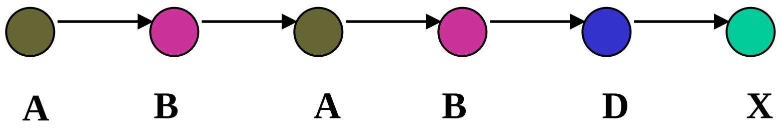
OTHER VERSIONS OF PROBLEM:

- ***CASE OF N WORDS**
- ***CASE OF 2 ARAYS**
- ***CASE OF N ARRAYS**
- ***M-DIMENSIONAL CASES**
- ***ETC.**

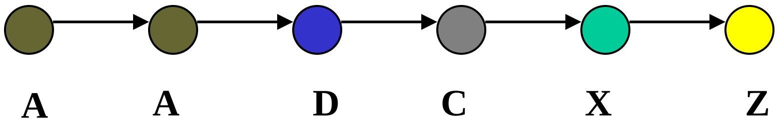
Substructure and superstructure (illustration)

CASE OF 2 CHAINS:

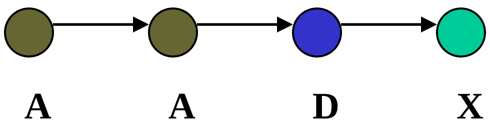
Chain 1



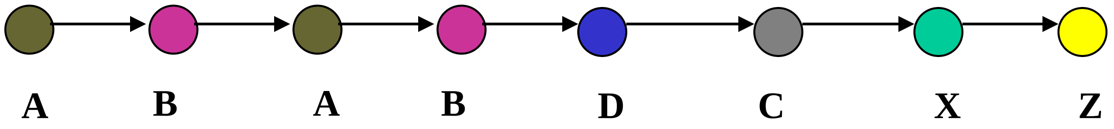
Chain 2



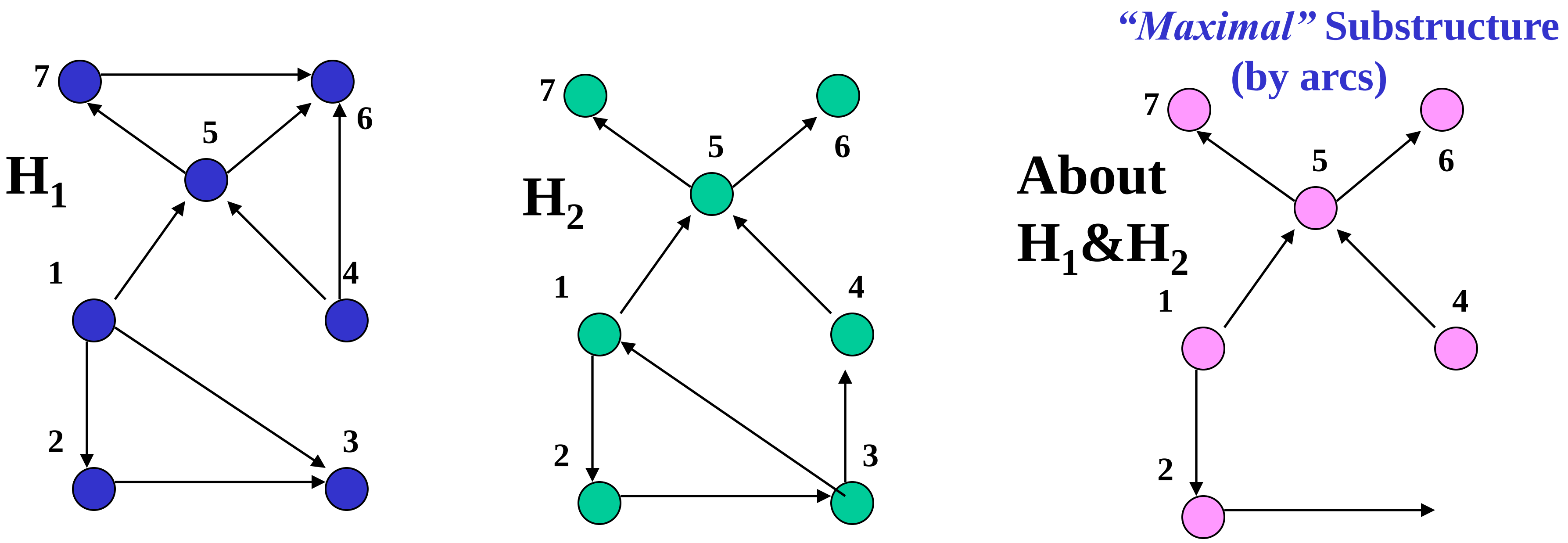
Problem 1: *Maximal* Substructure



Problem 2: *Minimal* Superstructure

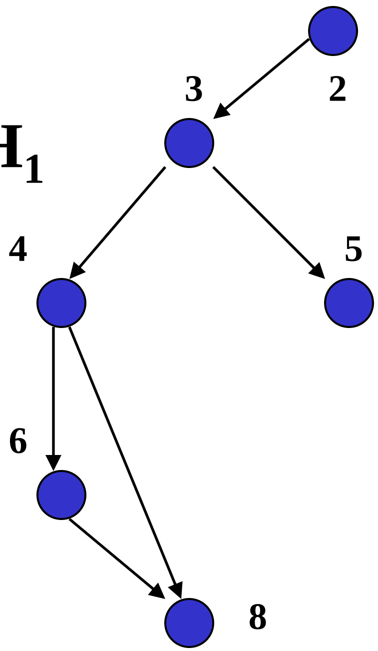


Substructure and superstructure (illustration): case of 2 orgraphs

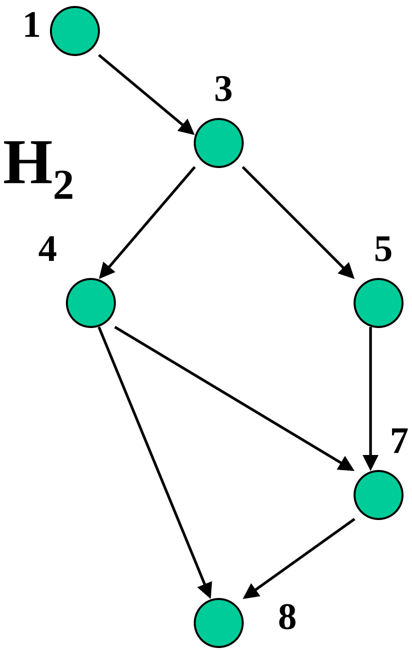


Substructure and superstructure (illustration): case of 2 orgraphs

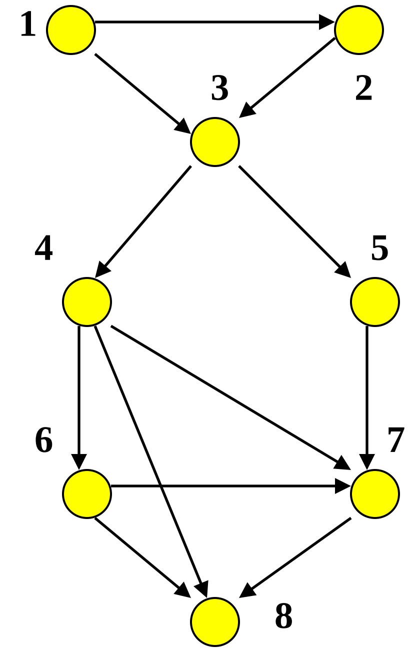
H_1



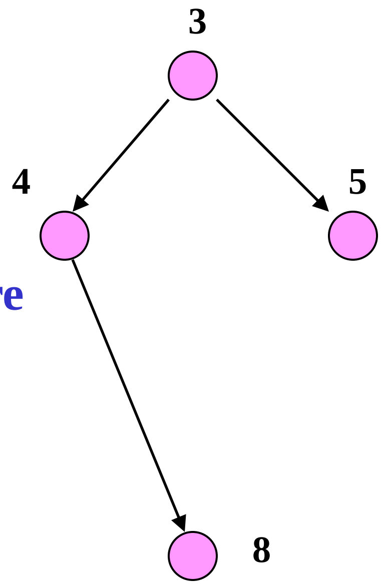
H_2



***Minimal* Superstructure**



***Maximal* Substructure**



APPLICATIONS:

1. **Decision making / Expert judgment (relation of dominance)**
2. **Information structures (data bases)**
3. **Information structures (knowledge bases)**
4. **Bioinformatics**
5. **Chemical structures**
6. **Network-like systems (e.g., social networks, software)**
7. **Graph-based patterns**
8. **Images (graph models of images)**
9. **Linguistics**
10. **Organizational structures**
11. **Engineering systems**
12. **Architecture**
13. **Information retrieval**
14. **Pattern recognition**
15. **Proximity for graph-like systems**

etc.

Substructures and superstructures (illustration)

OTHER VERSIONS OF PROBLEM:

- ***CASE OF N GRAPHS (BINARY RELATIONS)**
- ***CASE OF WEIGHTED GRAPHS**
- ***CASES UNDER SPECIAL CONSTRAINTS**
- ***ETC.**

APPLICATIONS: **1.Scheduling in educational institutions (universities, schools)**
2.Scheduling in hospitals
3.Scheduling in sport (e.g., basketball)
ETC.

COMPOSITE ALGORITM SCHEMES on the basis of model combination:
1.Graph coloring
2.Assignment / Allocation
3.Combinatorial design
4.Basic scheduling
Etc.

LECTURE 19. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Institute of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.19. Morphological synthes.

PLAN:

1.Morphological analysis

2.Hierarchical Morphological Multicriteria Design (HMMD)

(morphological synthesis, combinatorial synthesis): Fundamentals

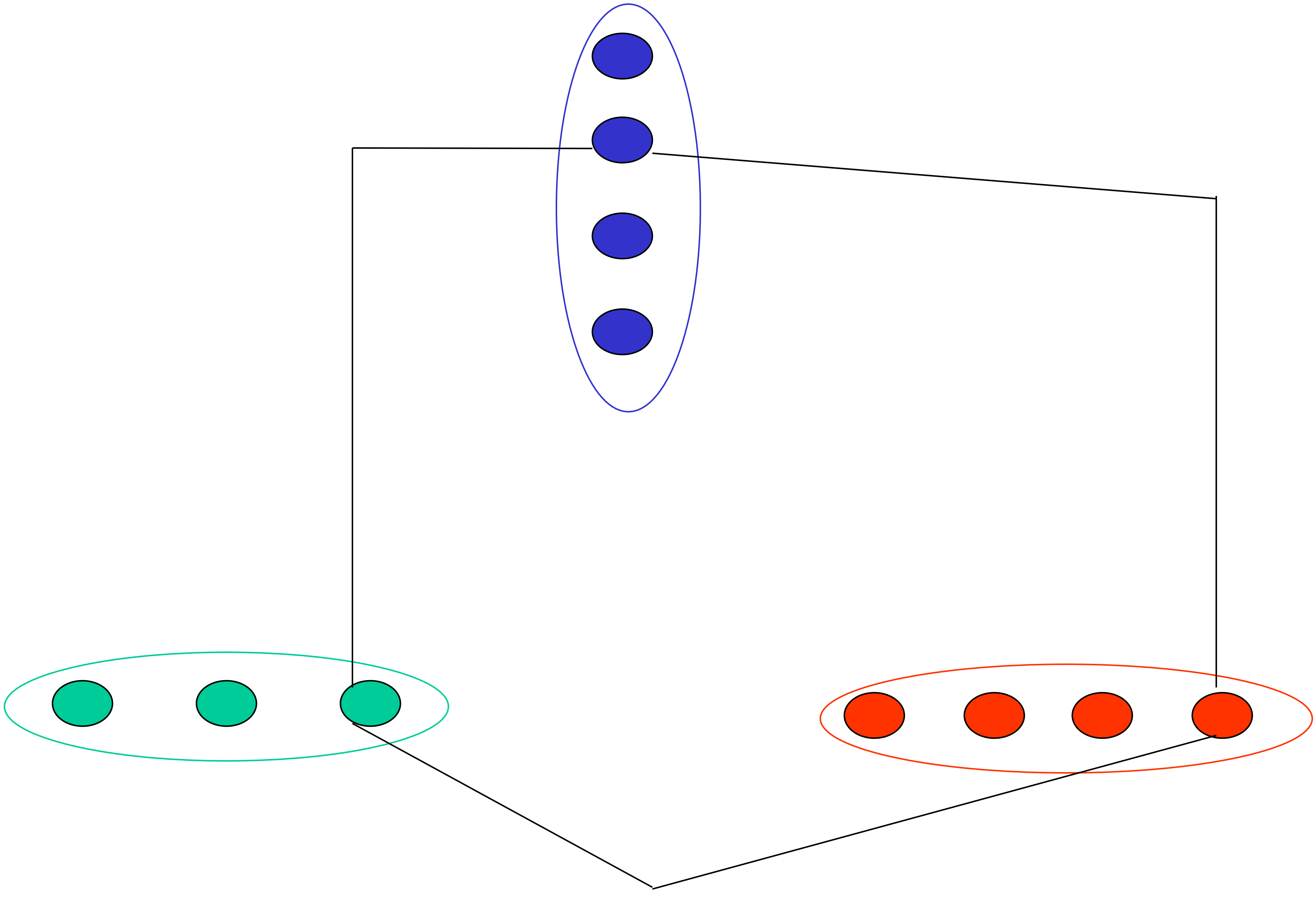
3.Five development phases of morphological analysis

4.Preliminary phases (1, 2, 3, 4)

5.HMMD: *formulation, *solving schemes, *examples

Oct. 15, 2004

Morphological clique



Fundamentals of Hierarchical Morphological Design

1. Morphological analysis (F. Zwicky, 1943)

2. Paradigm of multicriteria decision making (H. Simon)

3. Dynamic programming (R. Bellman)

4. Engineering practice in hierarchical design of complex multidisciplinary systems

5. Combinatorial optimization

6. Knowledge engineering (multidisciplinary information): extraction, organization, usage

Development Phases of Morphological Analysis

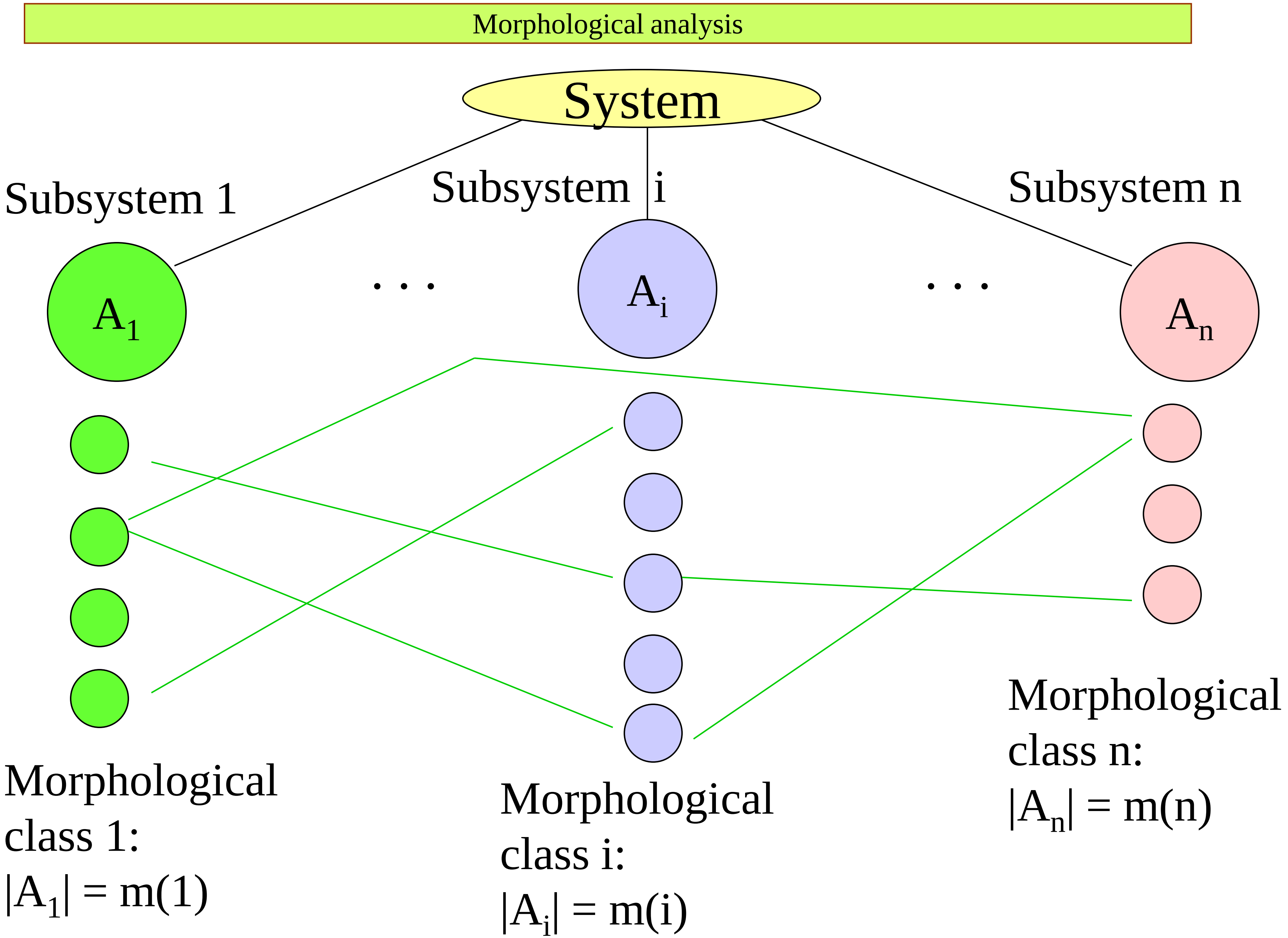
1.Morphological analysis [F. Zwicky]

2.Closeness of feasible combination to "IDEAL" [Ayres,1969; Iakimets, Inst. for System Analysis, Russian Academy of Sci. (RAS), 1977]

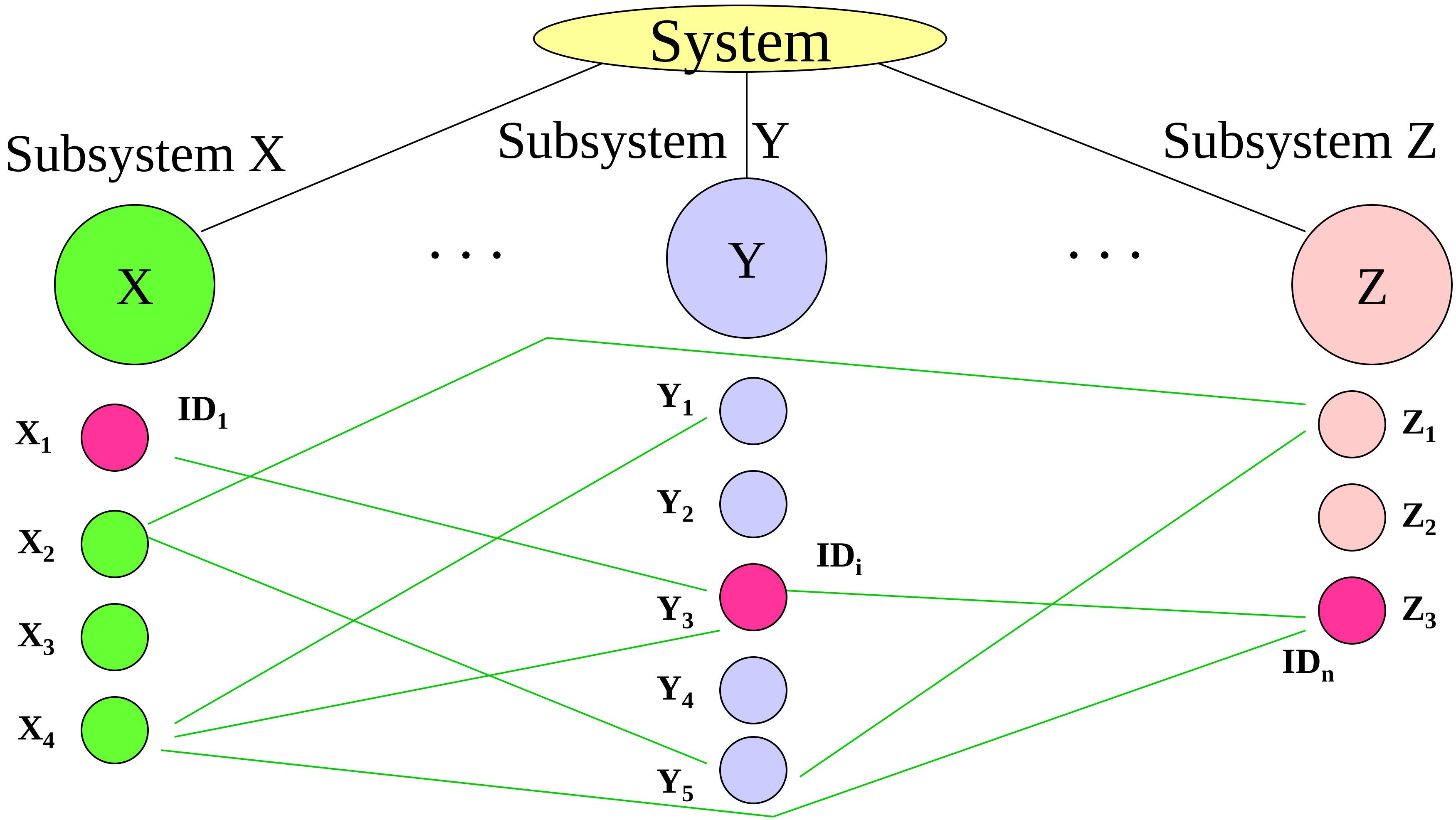
3.Multicriteria evaluation of feasible combinations and selection of Pareto-effective decisions [Inst. for Control Problems (IPU), Inst. for System Analysis (ISA), Inst. for Machine-engineering (IMASH); RAS, 1972/82]

4.Hierarchical design (composition of local Pareto-effective solutions) [Comp. Center, RAS, Krasnoshekov et al.,1979…]

5. Hierarchical Morphological Multicriteria Design (HMMD) (combinatorial morphological synthesis) [Levin, 1994…]


Morphological analysis
System
Subsystem 1
Subsystem i
Subsystem n
A_1
A_i
A_n
. . .
. . .
Morphological class 1: $|A_1| = m(1)$
Morphological class i: $|A_i| = m(i)$
Morphological class n: $|A_n| = m(n)$

Morphological analysis: Closeness to “IDEAL” (Ayres, 1969)



“IDEAL” : $X_1 * Y_3 * Z_3$
(infeasible combination)

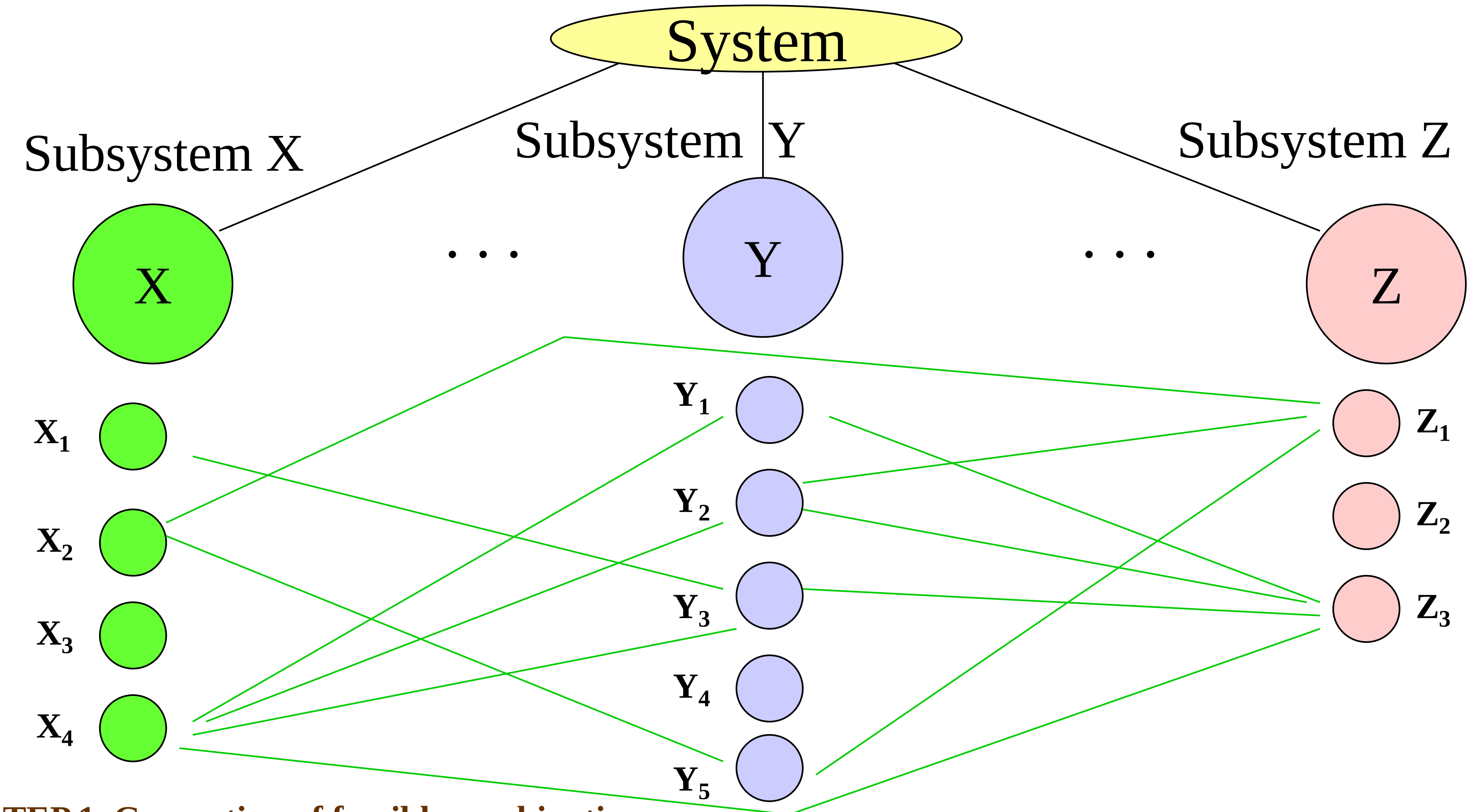
$S_1 = X_4 * Y_3 * Z_3$

$S_2 = X_2 * Y_5 * Z_1$

ρ (“IDEAL” , S_1) < ρ (“IDEAL”, S_2)

ρ *is a closeness (proximity)*

Multicriteria evaluation of feasible combinations & selection of Pareto-effective points (1972..1982)



STEP 1. Generation of feasible combinations:

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

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

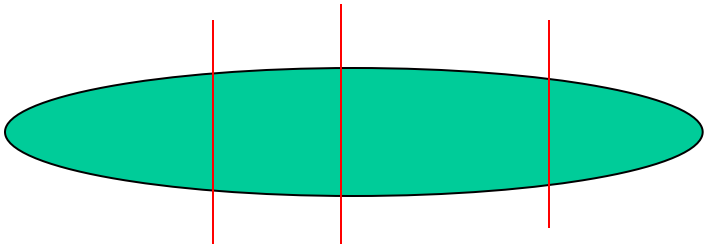
STEP 2. Evaluation upon criteria

STEP 3. Selection of Pareto-effective solutions

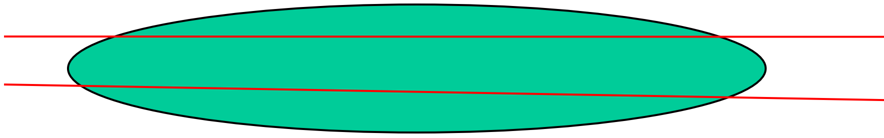
Complexity: m(1)*…*m(i)*…*m(n)

Decreasing of complexity:

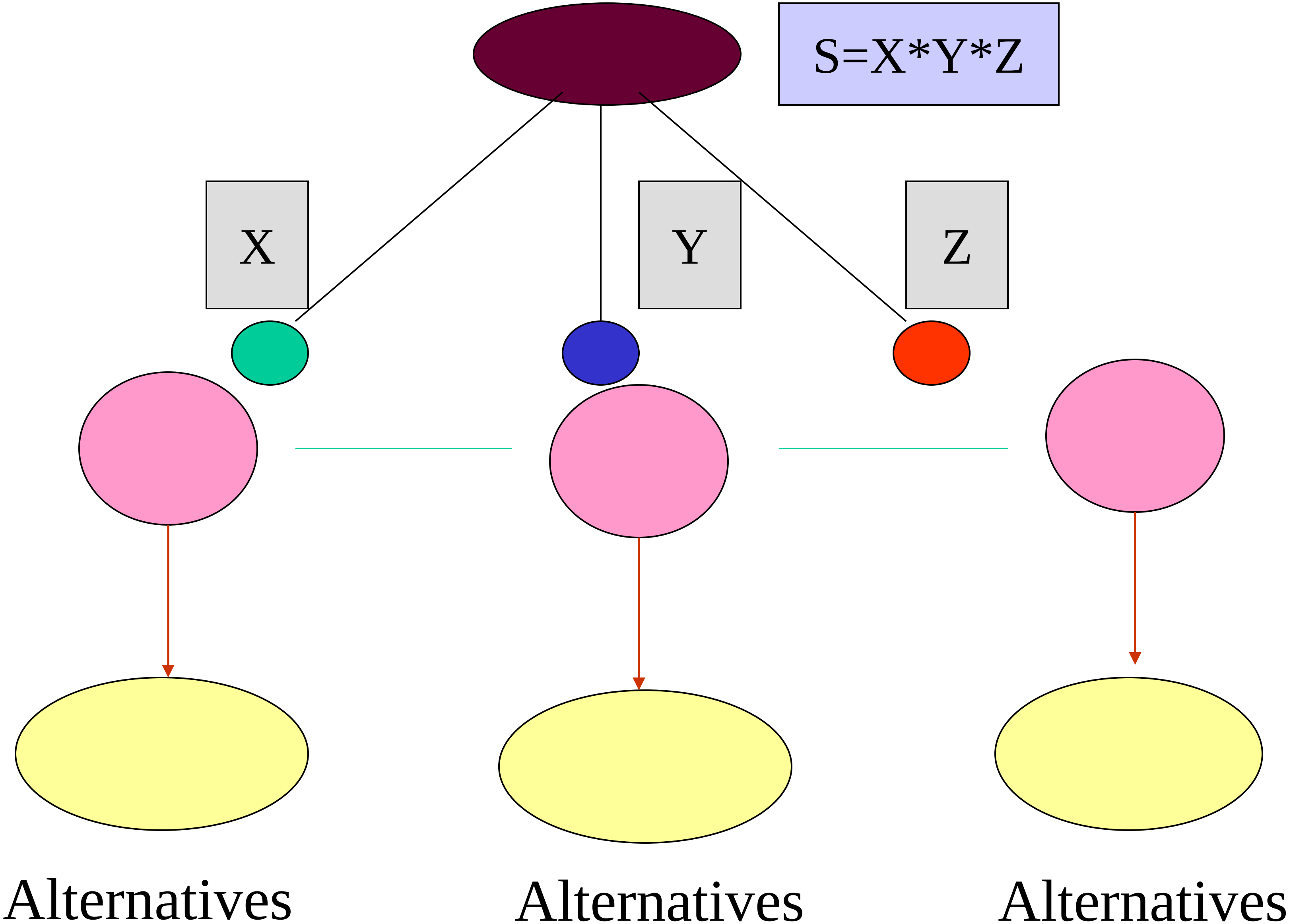
Horizontal partitioning



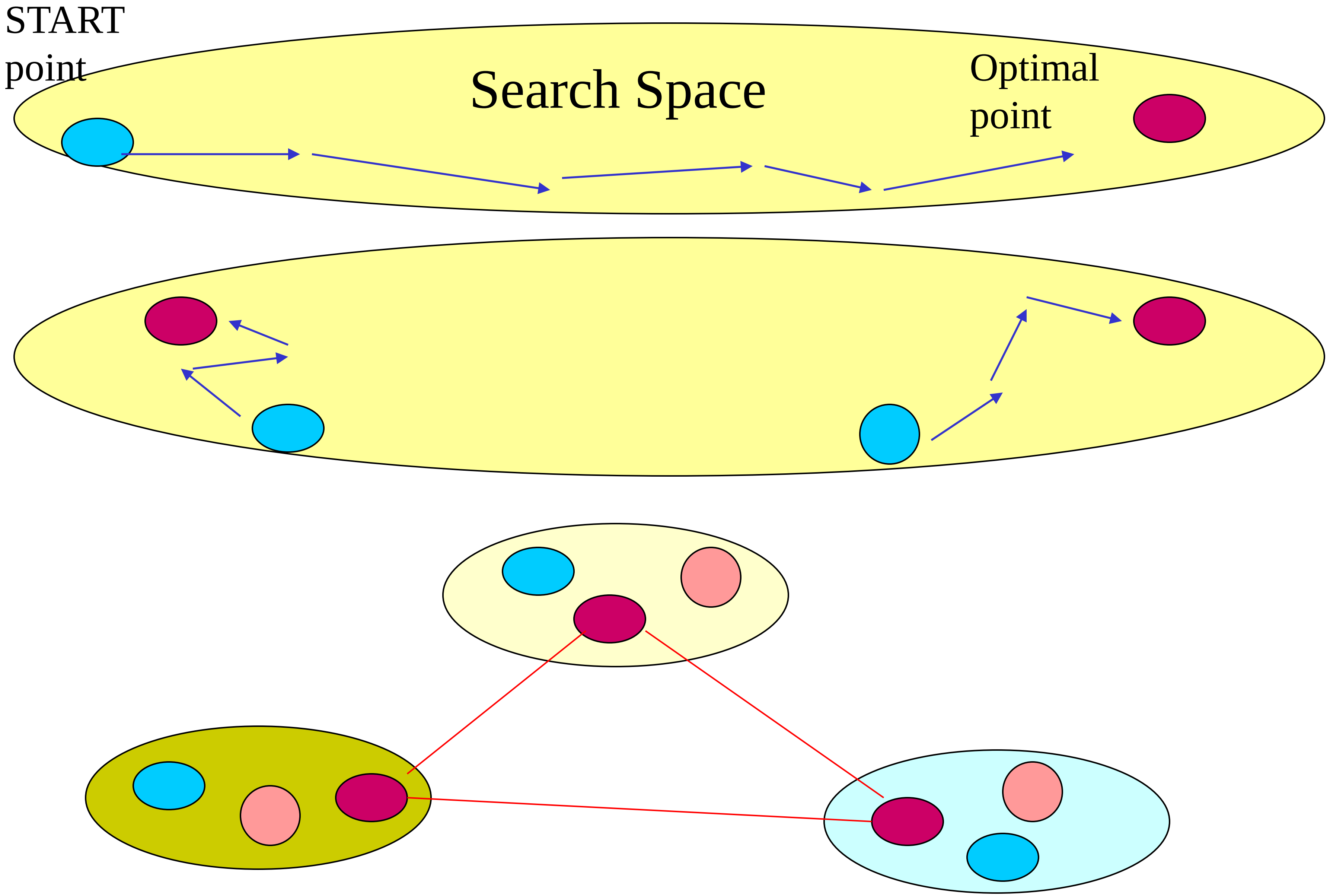
Vertical partitioning



Hierarchical morphological design (engineering experience, phases 3, 4)



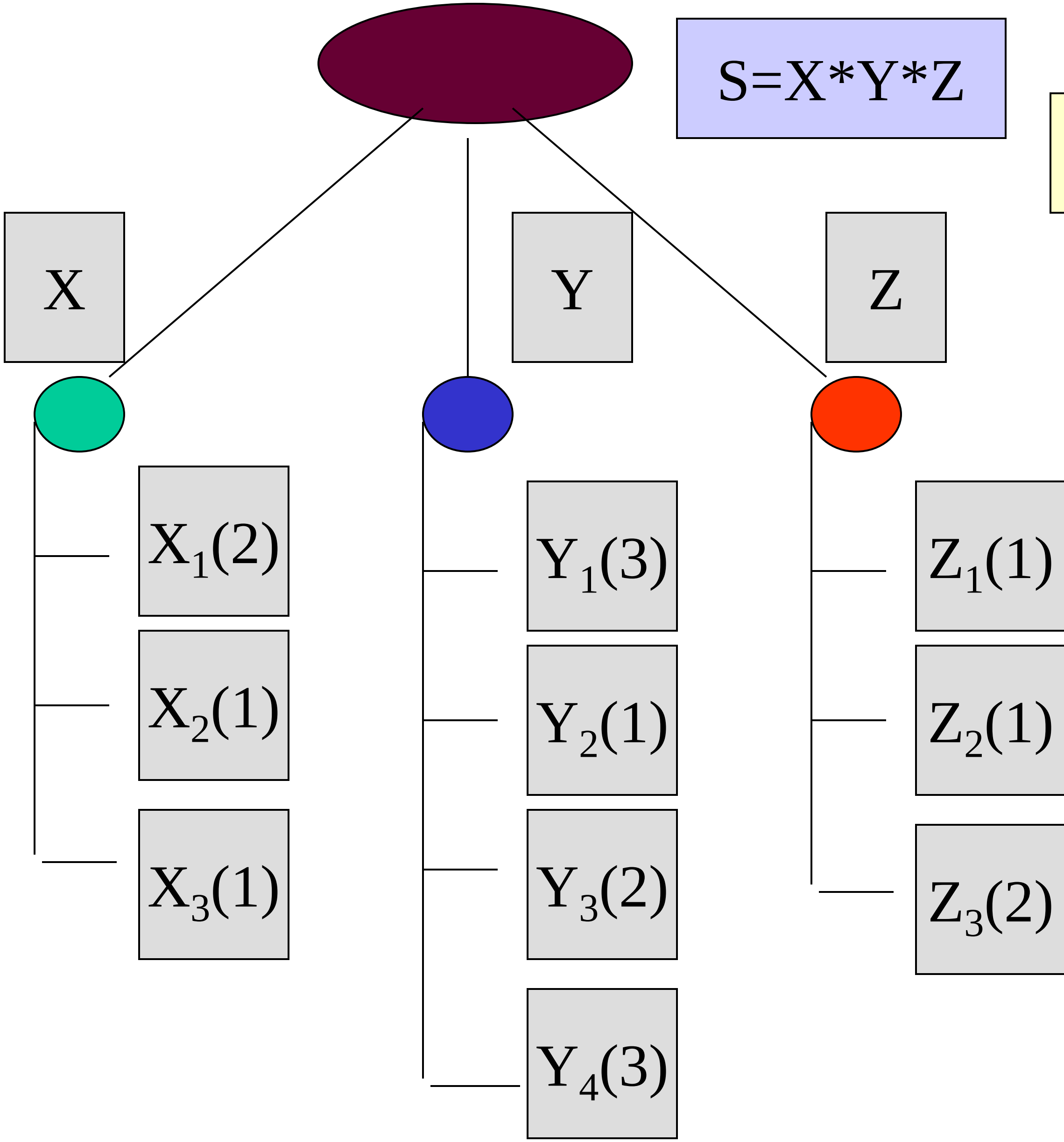
Search strategies



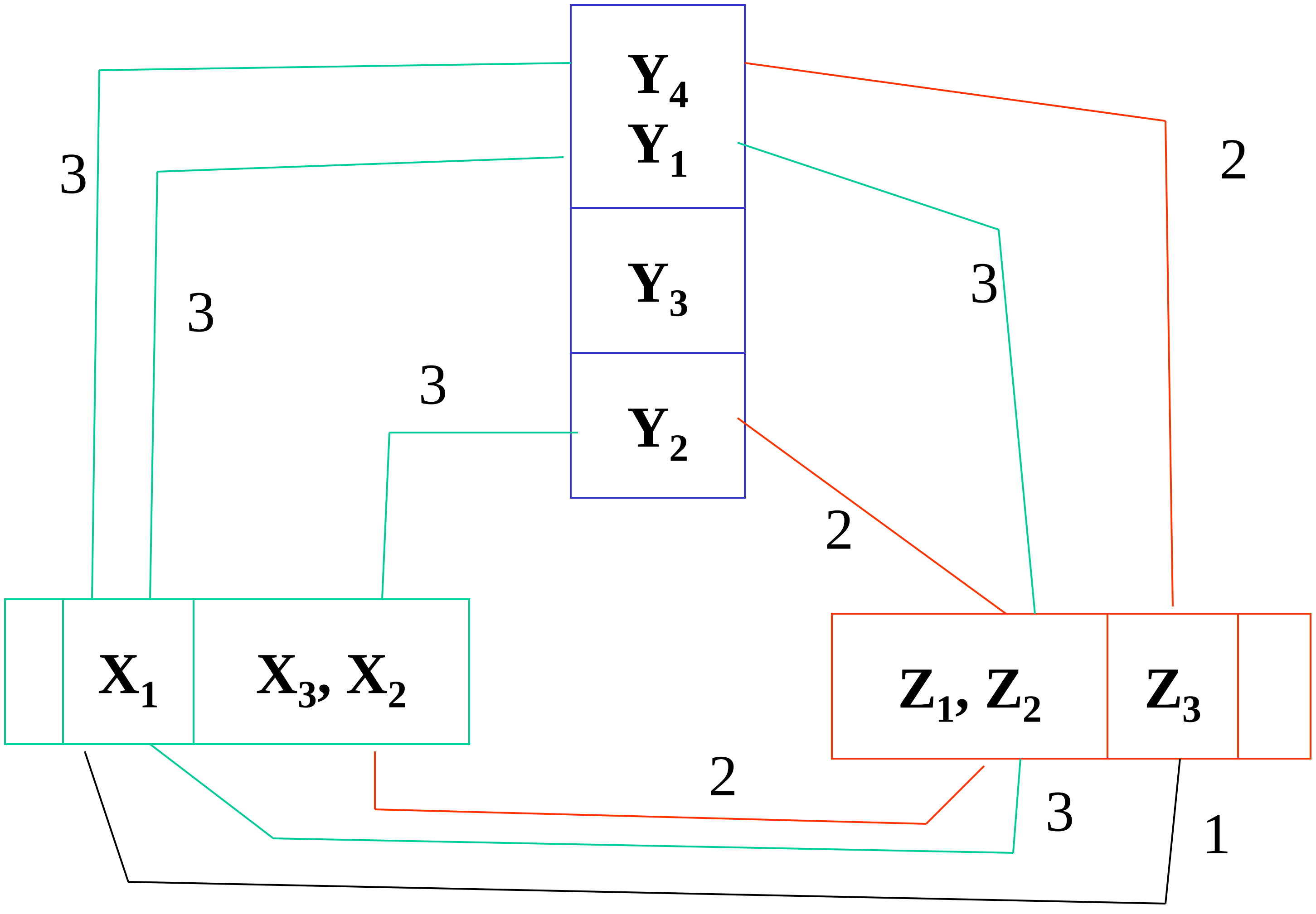
Morphological combinatorial synthesis: example

$S = X * Y * Z$

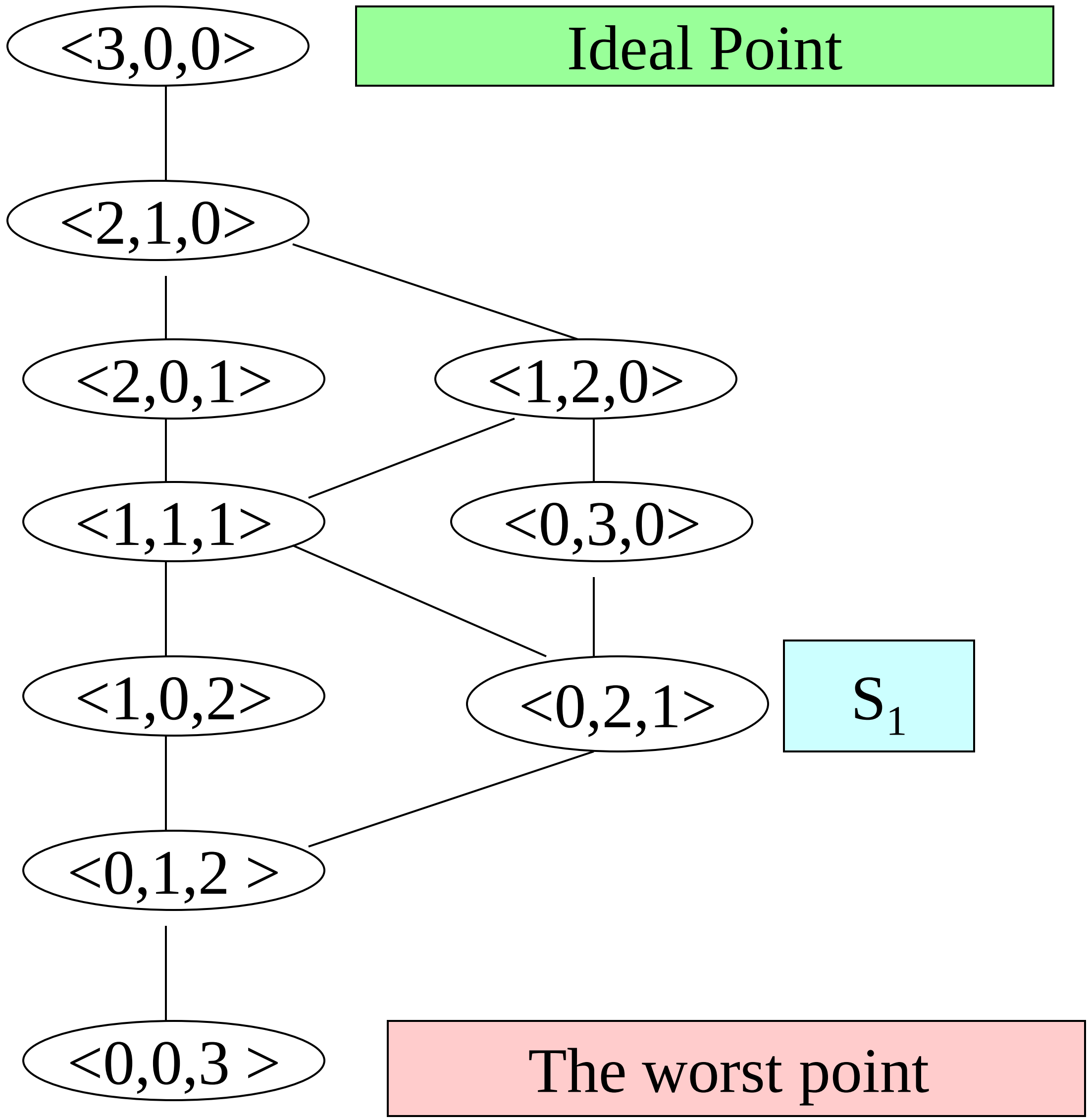
$S_1 = X_1 * Y_4 * Z_3$



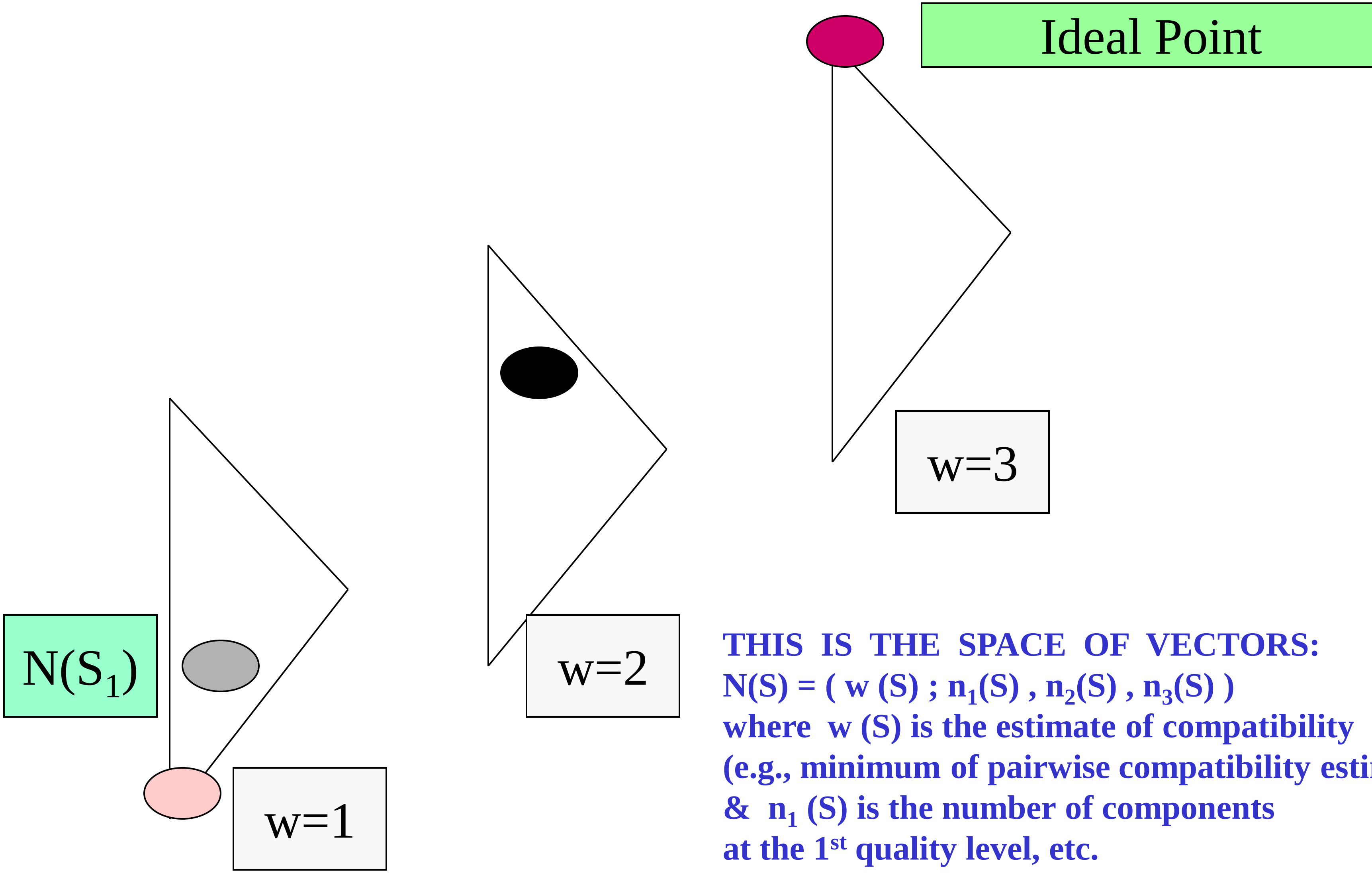
Concentric presentation of morphological clique with estimates of compatibility



Discrete space of quality (by components)

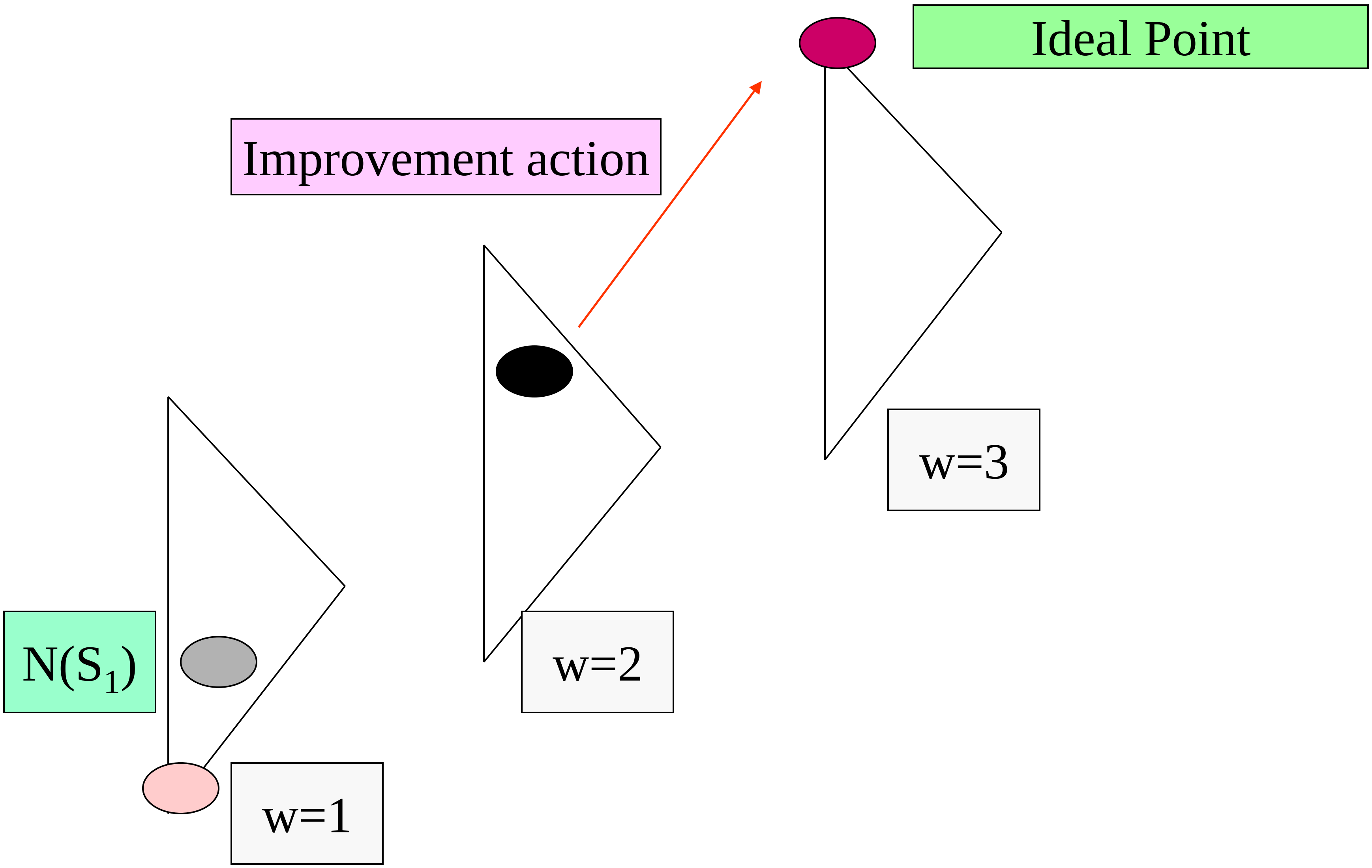


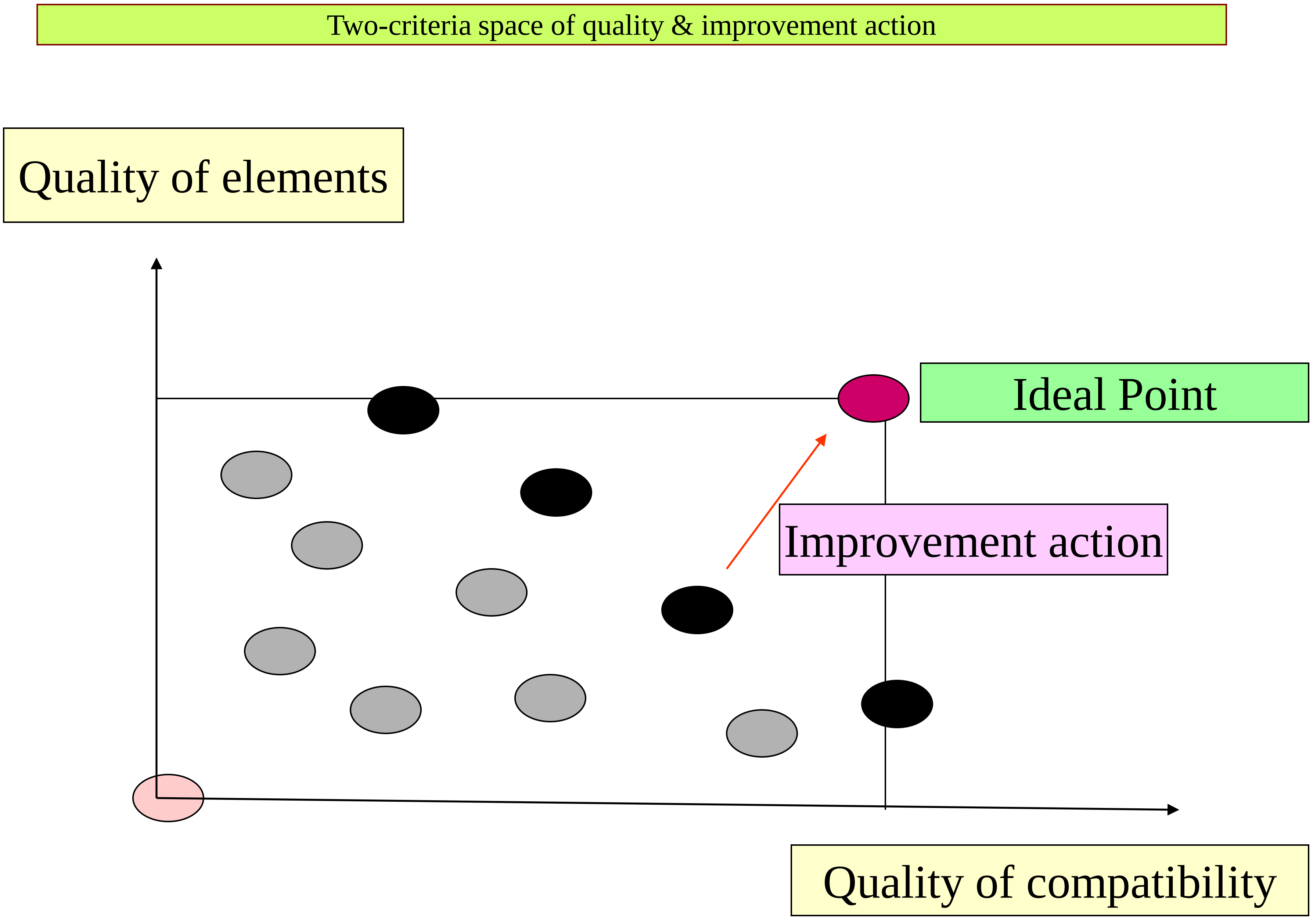
Discrete space of quality (by components, by compatibility)



THIS IS THE SPACE OF VECTORS:
$N(S) = (w (S) ; n_1(S) , n_2(S) , n_3(S))$
where w (S) is the estimate of compatibility
(e.g., minimum of pairwise compatibility estimates)
& n_1 (S) is the number of components
at the 1st quality level, etc.

Discrete space of quality (by components, by compatibility) & improvement



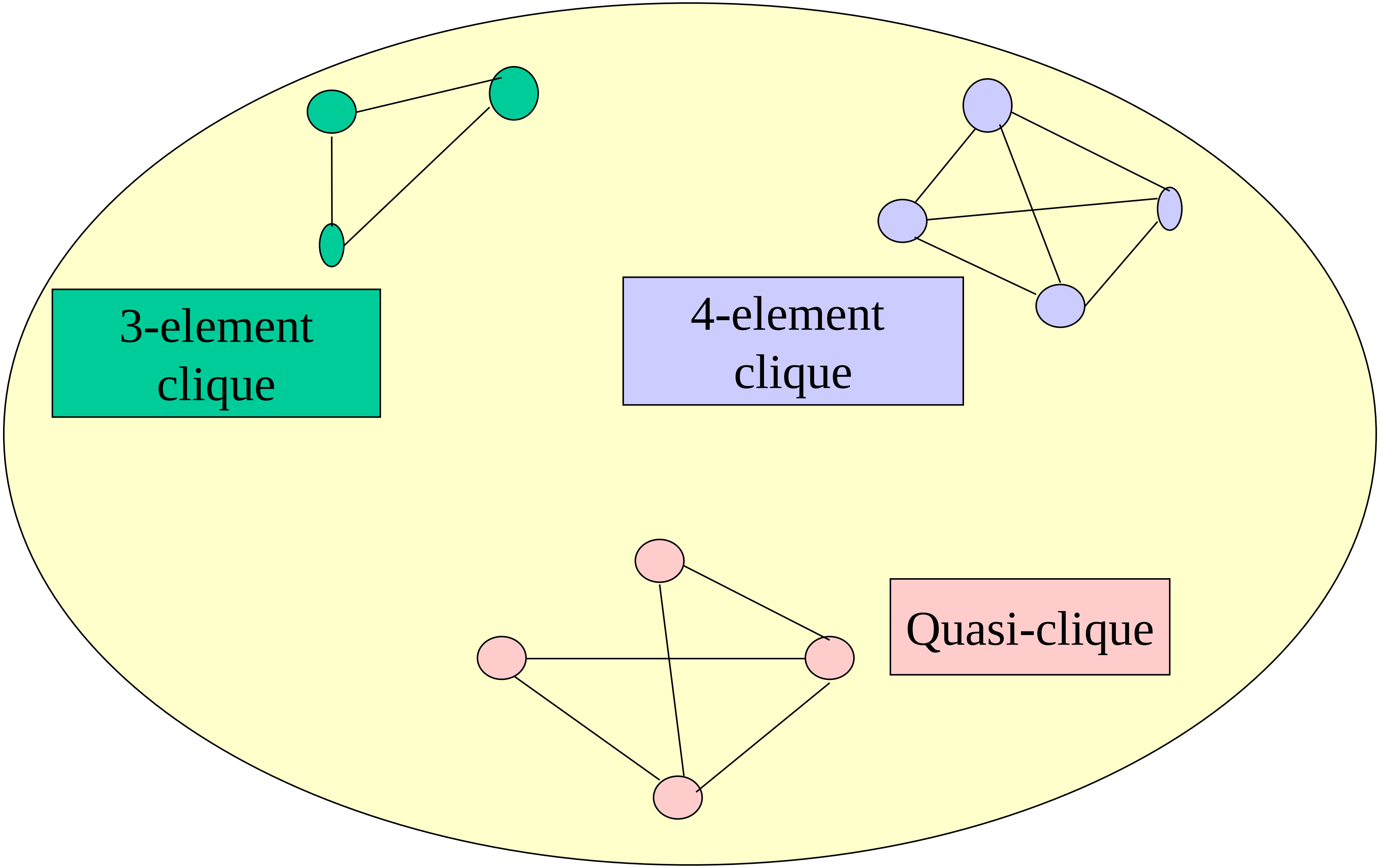

Two-criteria space of quality & improvement action
Quality of elements
Ideal Point
Improvement action
Quality of compatibility

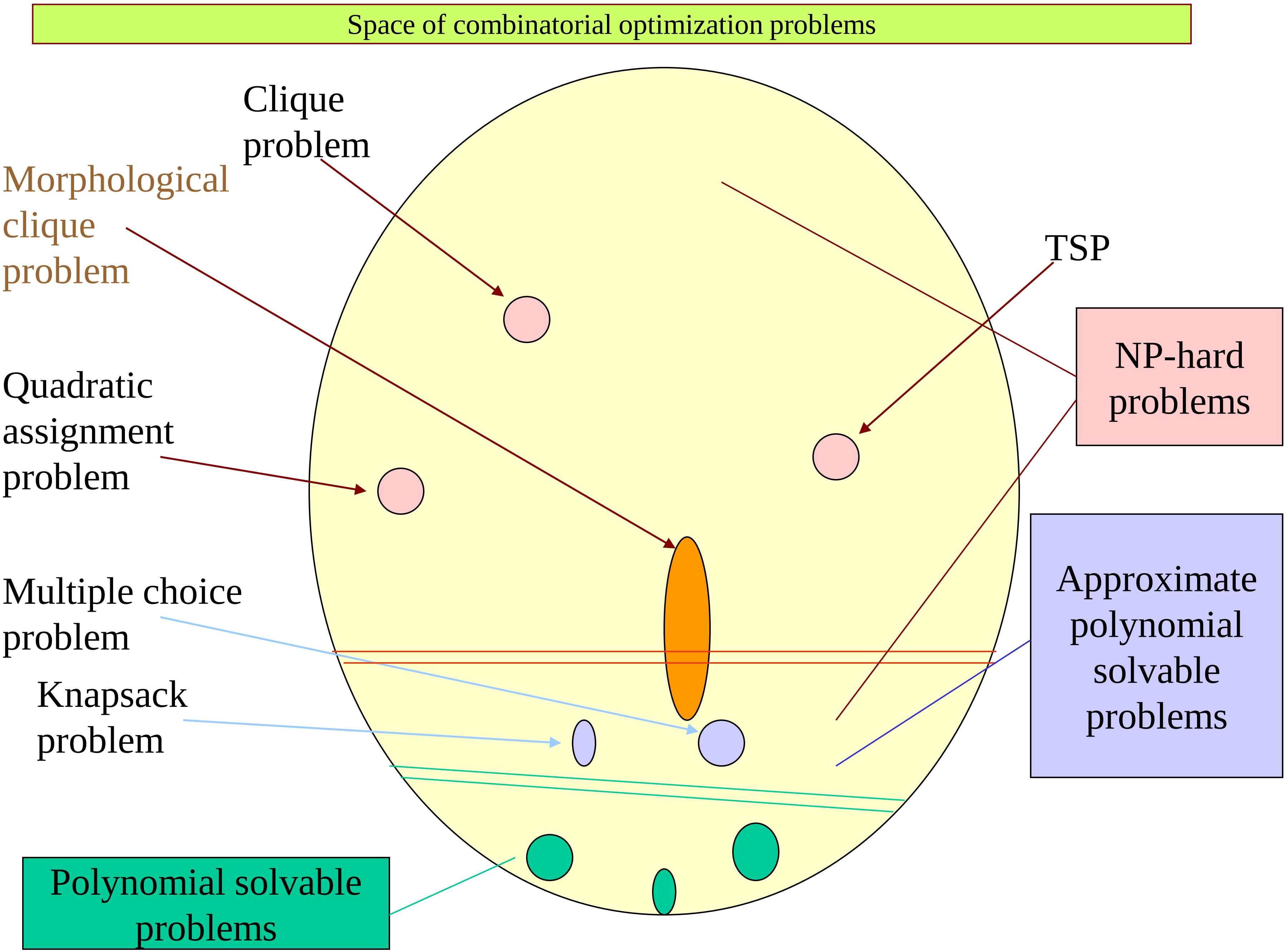
Two types of solving schemes

1. Enumerative directed heuristic:
   analysis and checking
   since the best point.

2. Dynamic programming methods:
   extension of the method for
   knapsack problem or
   multiple choice problem.

Examples of clique & quasi-clique




Space of combinatorial optimization problems
Clique problem
Morphological clique problem
TSP
NP-hard problems
Quadratic assignment problem
Multiple choice problem
Approximate polynomial solvable problems
Knapsack problem
Polynomial solvable problems

"Sequence" of close problems

1. Knapsack problem

2. Multiple choice problem

3. Quadratic assignment problem

4. Integer non-linear programming

5. Mixed non-linear programming

LECTURE 20-21. Course: “Design of Systems: Structural Approach”

Dept. “Communication Networks &Systems”, Faculty of Radioengineering & Cybernetics

Moscow Institute of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.20. Application of morphological synthesis

L.21. System bottlenecks, improvement, multistage design

PLAN:

1Hierarchical Morphological Multicriteria Design (HMMD).

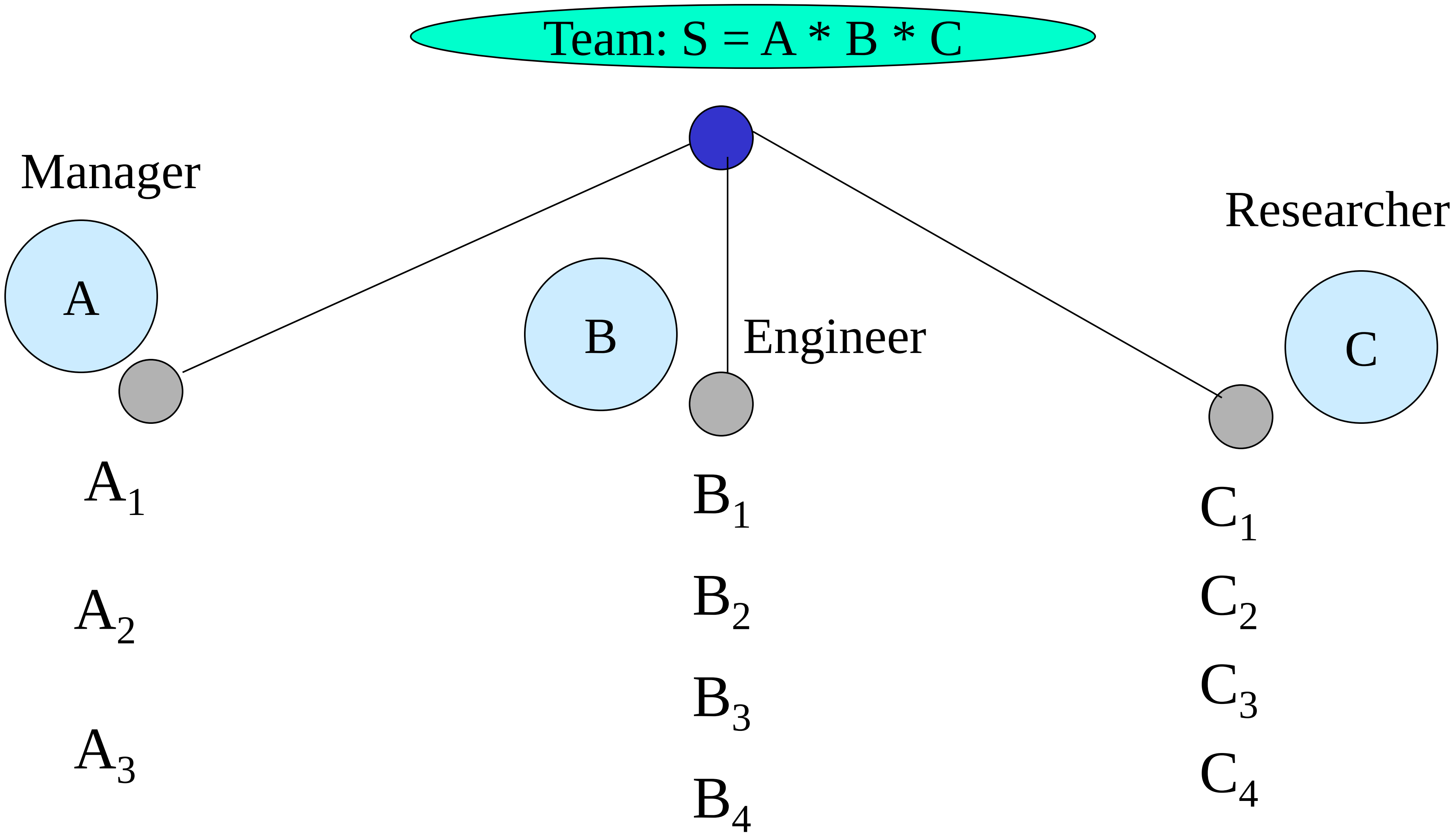
Examples: *design of team *design of solving strategy *

2.Approaches to revelation of bottlenecks

3.Multistage design

Oct. 16, 2004

Application example 1: Design of team



Application example 1: Design of team (assessment of alternatives for Manager)

	Experience 3	Market 4	Salary -1	Total rank
A_1	5	3	10	3
A_2	15	5	20	1
A_3	10	4	15	2

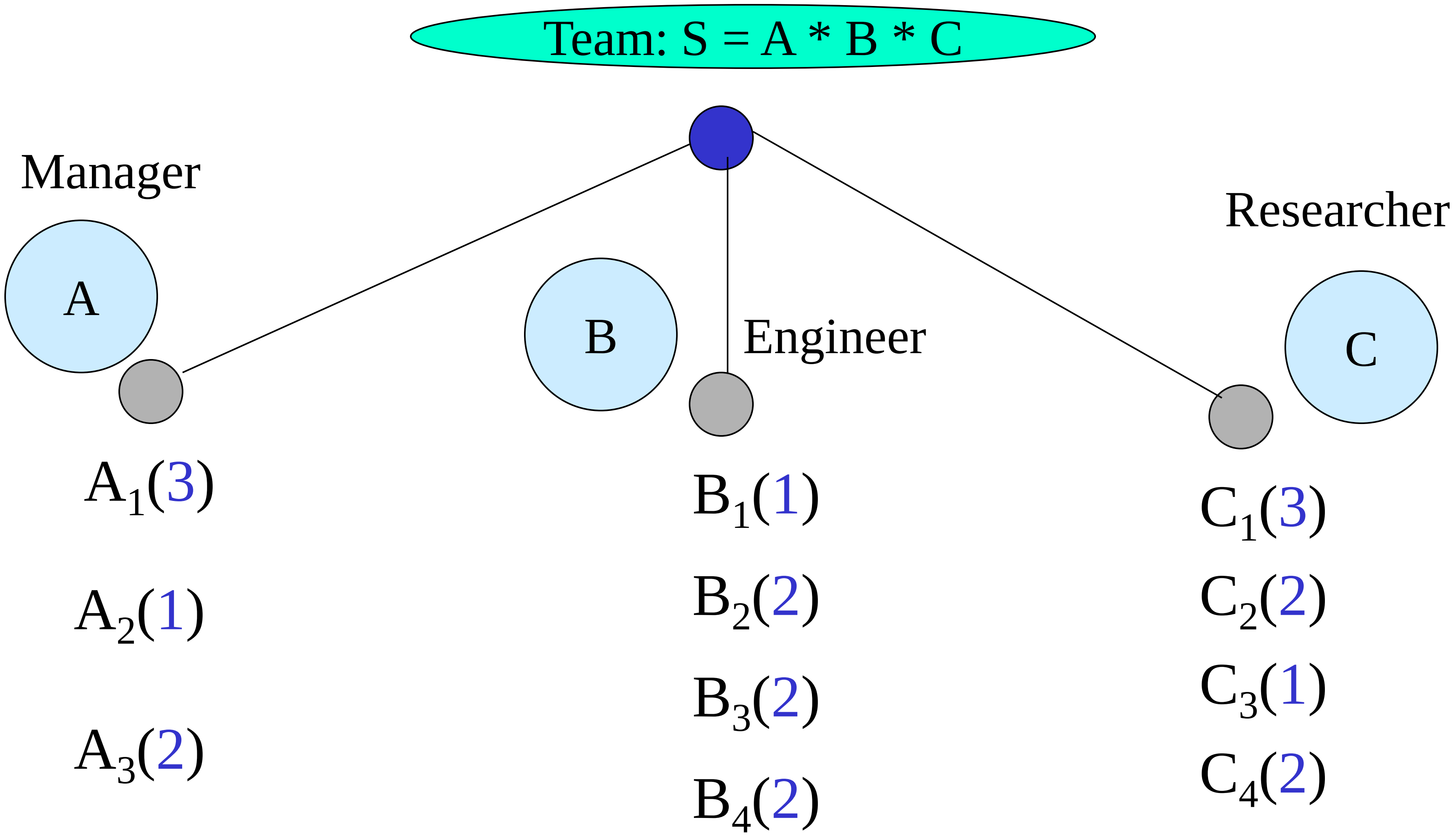
Application example 1: Design of team (assessment of alternatives for Engineer)

	Experience 3	West Experience 2	Salary -3	Total rank
B_1	15	5	7	1
B_2	6	6	3	2
B_3	10	0	9	2
B_4	3	0	3	2

Application example 1: Design of team (assessment of alternatives for Researcher)

	Experience 2	Creativity 4	Salary -3	Total rank
C_1	15	3	10	3
C_2	5	5	7	2
C_3	4	7	4	1
C_4	6	4	6	2

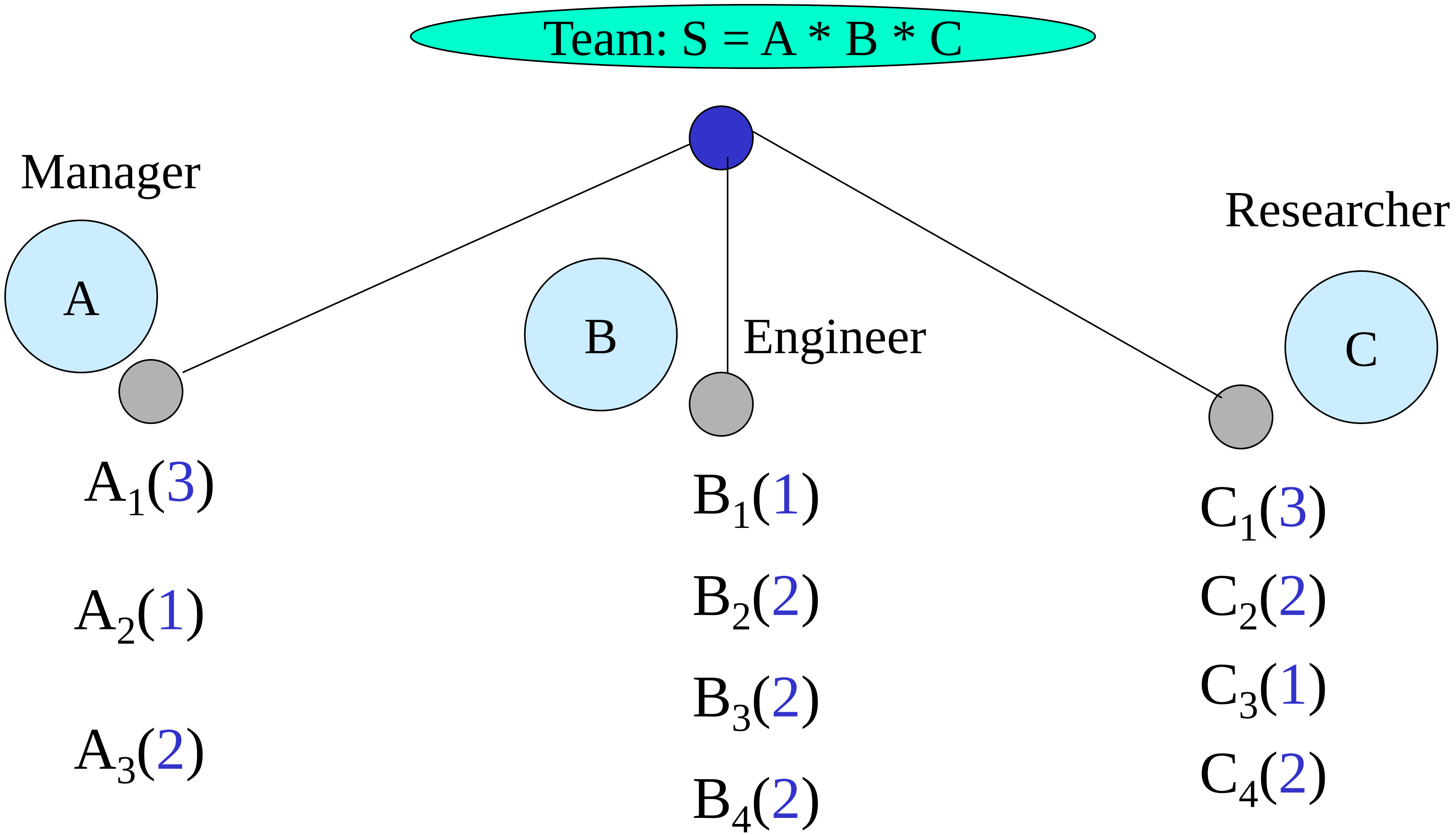
Application example 1: Design of team



Application example 1: Design of team (estimates of compatibility)

	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

Application example 1: Design of team

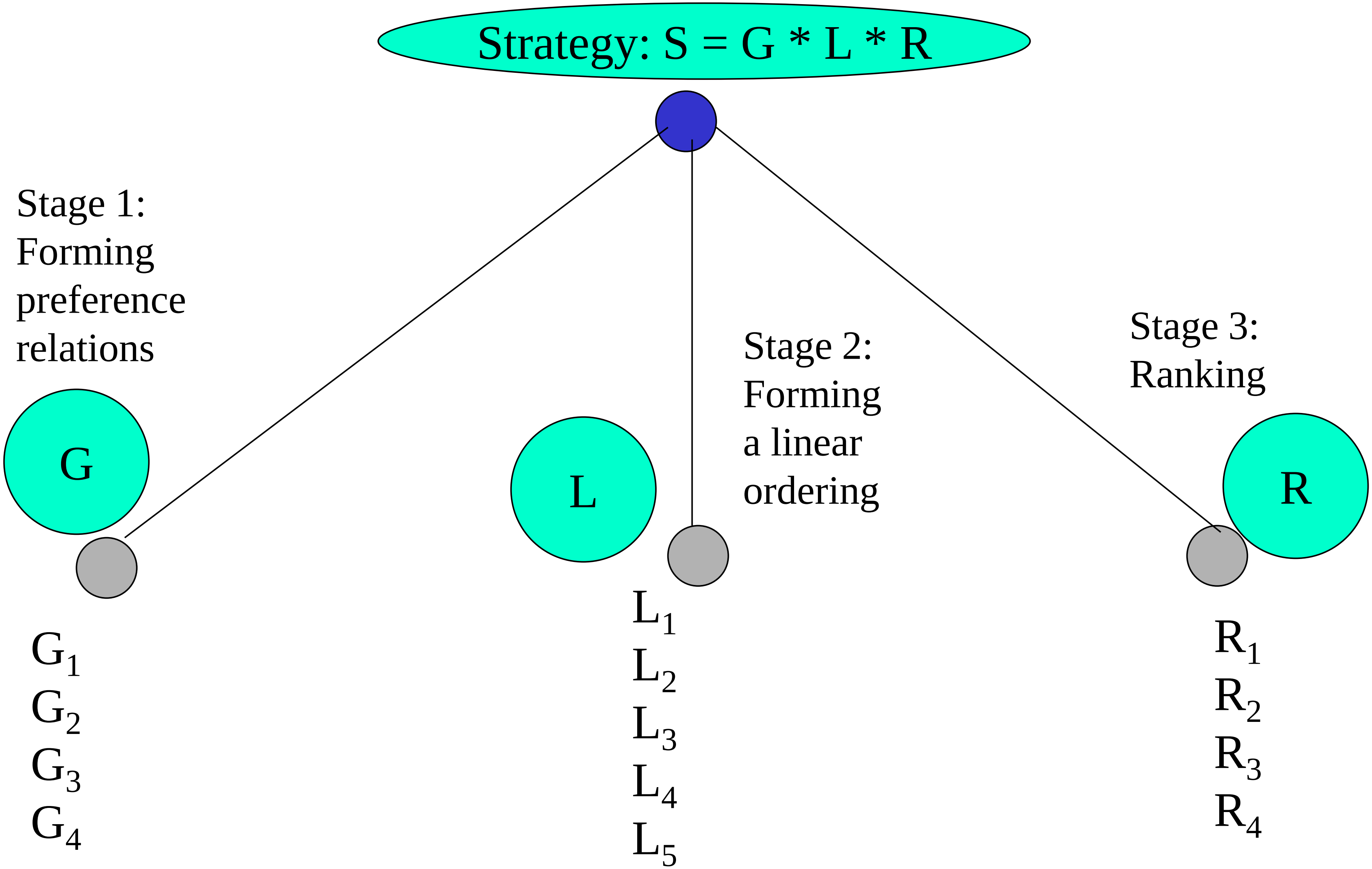


Application Example 1: Design of Team (resultant decisions)

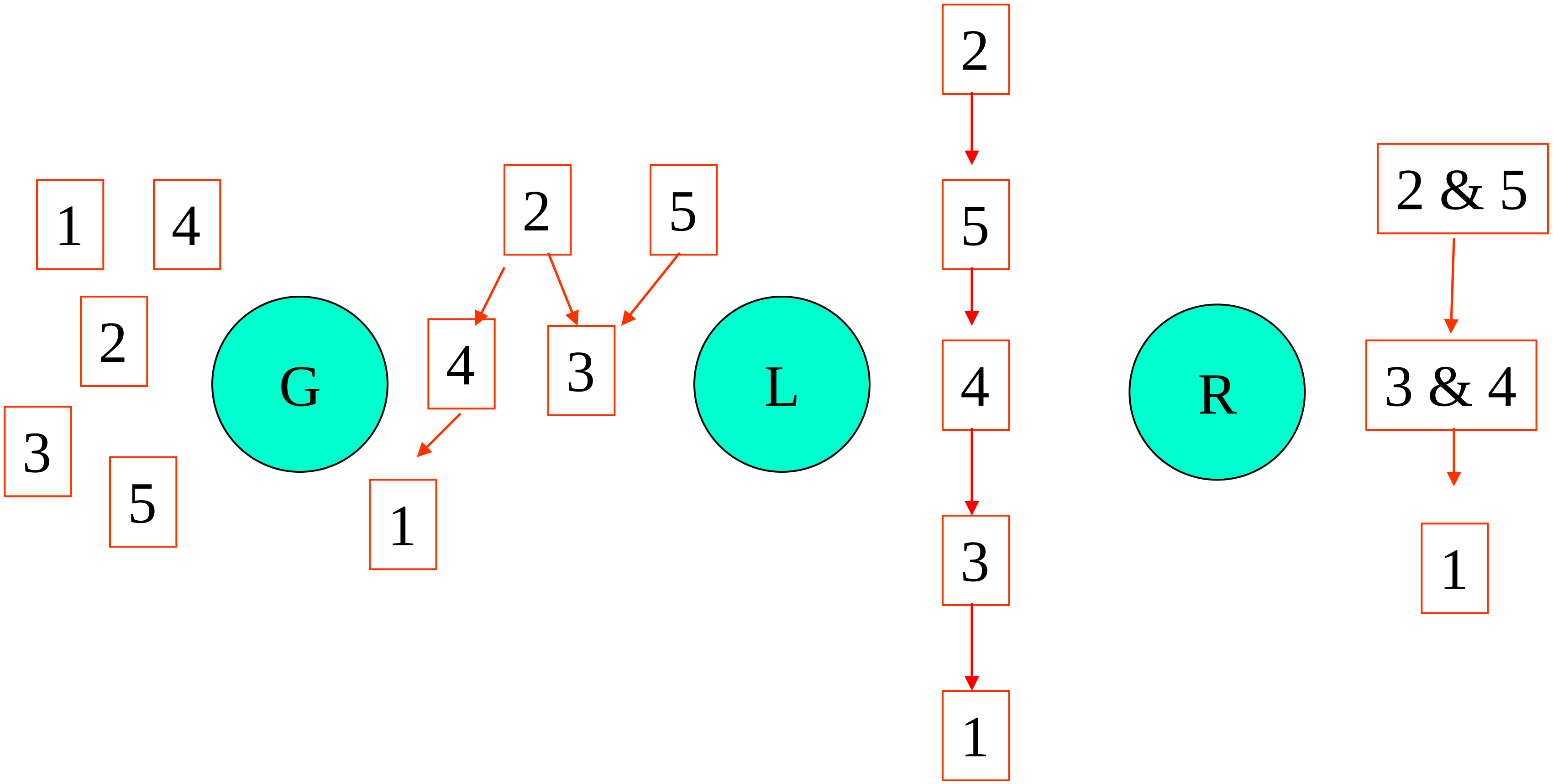
$S_1=A_2*B_1*C_2$ $N(S_1)=(3;2,1,0)$

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

Application Example 2: Strategy for multicriteria ranking



Application Example 2: Strategy for multicriteria ranking

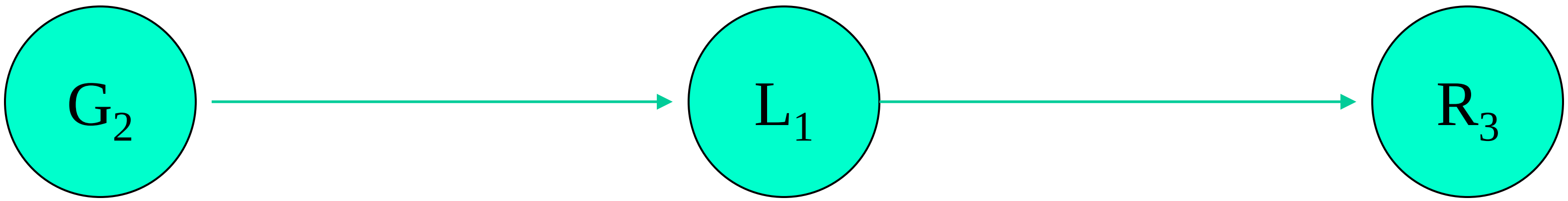


LOCAL ALTERNATIVES (COMBI-PC, 1989…):

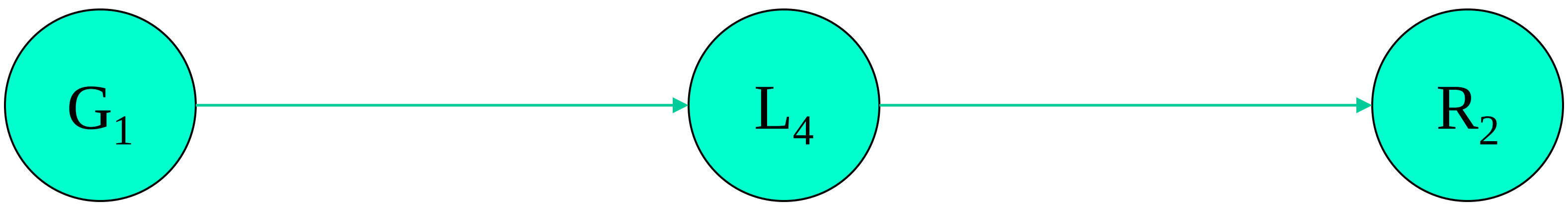
- **G_1 Pairwise comparison**
- **G_2 ELECTRE-like technique**
- **G_3 Additive utility function method**
- **G_4 Expert stratification**
- **L_1 Line element sum of preference matrix**
- **L_2 Additive function**
- **L_3 Series revelation of "max" element**
- **L_4 Series revelation of Pareto-elements**
- **L_5 Expert stratification**
- **R_1 Series revelation pf "max" element**
- **R_2 Series revelation of Pareto elements**
- **R_3 Dividing the linear ordering**
- **R_4 Expert stratification**

EXAMPLES OF STRATEGIES:

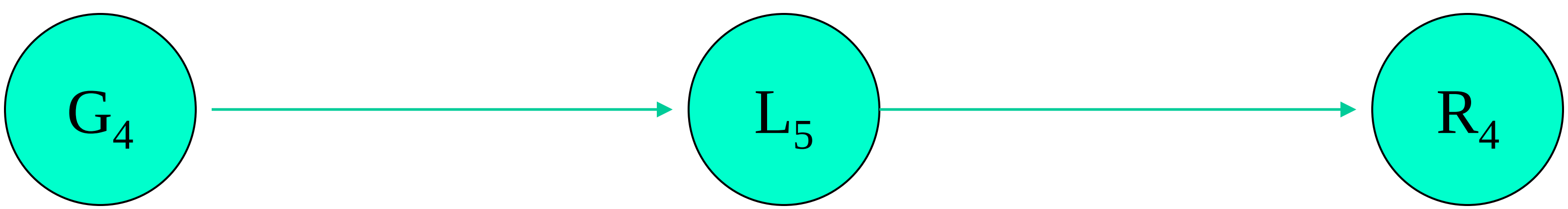
Strategy 1



Strategy 2

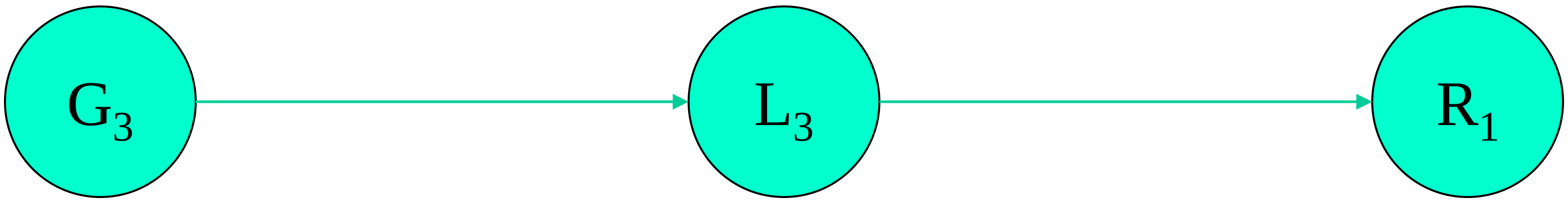


Strategy 3

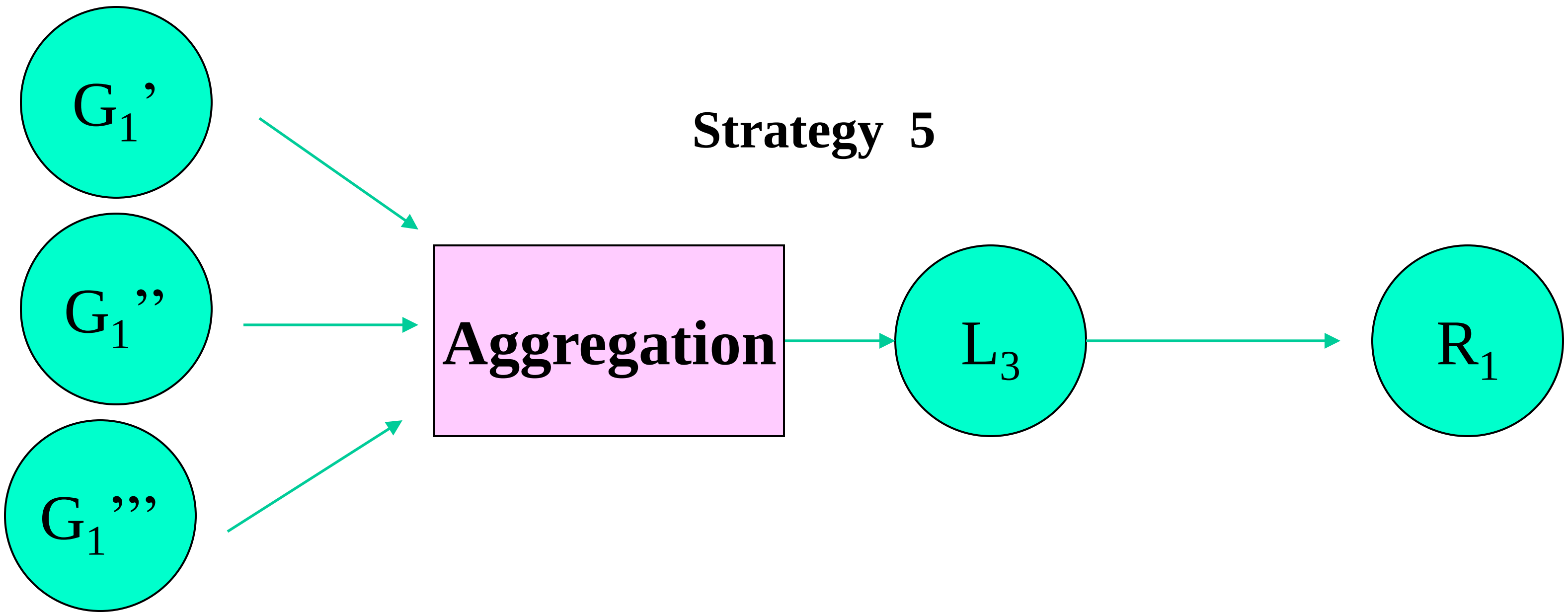


EXAMPLES OF STRATEGIES:

Strategy 4



Strategy 5



Approaches to revelation of system bottlenecks

Approach 1.

Engineering analysis (expert judgment)

Approach 2.

Design of system structure, assessment of reliability of system components, and selection of the most non-reliable ones ("Pareto approach" from Japanese system of quality management)

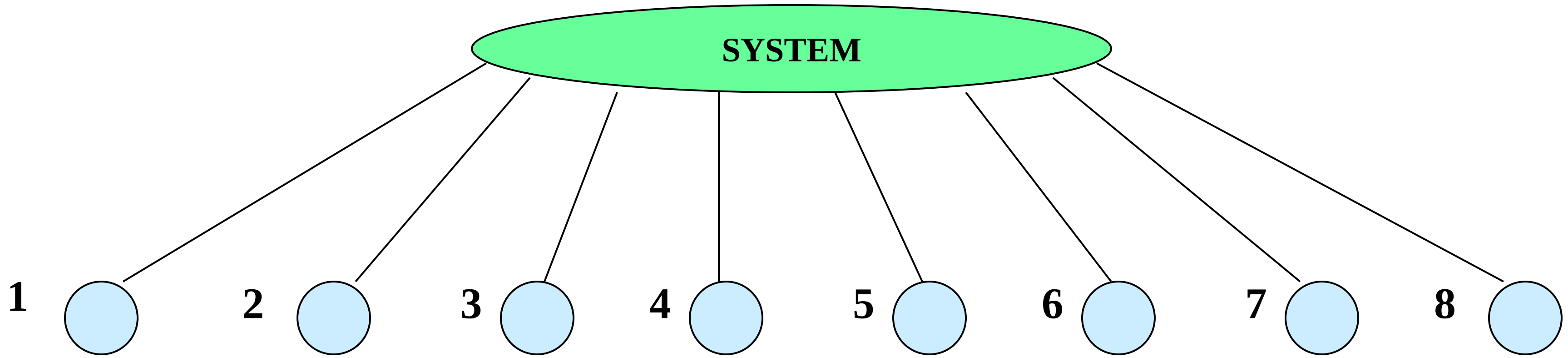
Approach 3.

Design of system structure, assessment of system components / parts, and multicriteria ranking of the system part (to reveal the most important parts)

Approach 4.

Analysis of the total vector estimate of the composable system decision S: $N(S) = (w(S); n_1(S), n_2(S),\dots)$

Approaches to revelation of bottlenecks



1.1
1.2
1.3
1.4

2.1

CRITERIA:
C_1
C_2
C_3
C_4
C_5
C_6

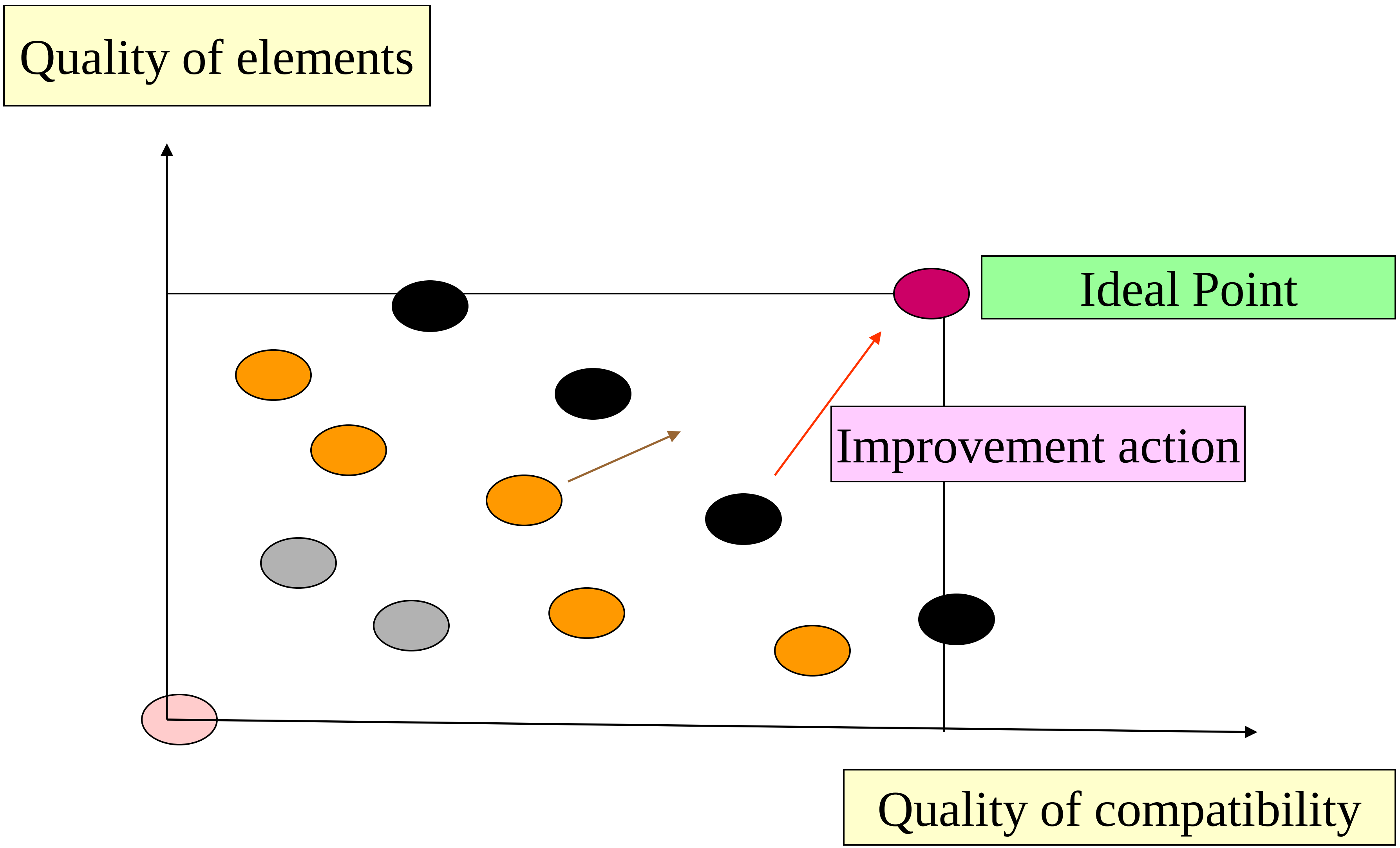
5.1
5.2
5.3

6.1
6.2
6.3
6.4
6.5
6.6
6.7
6.8
6.9

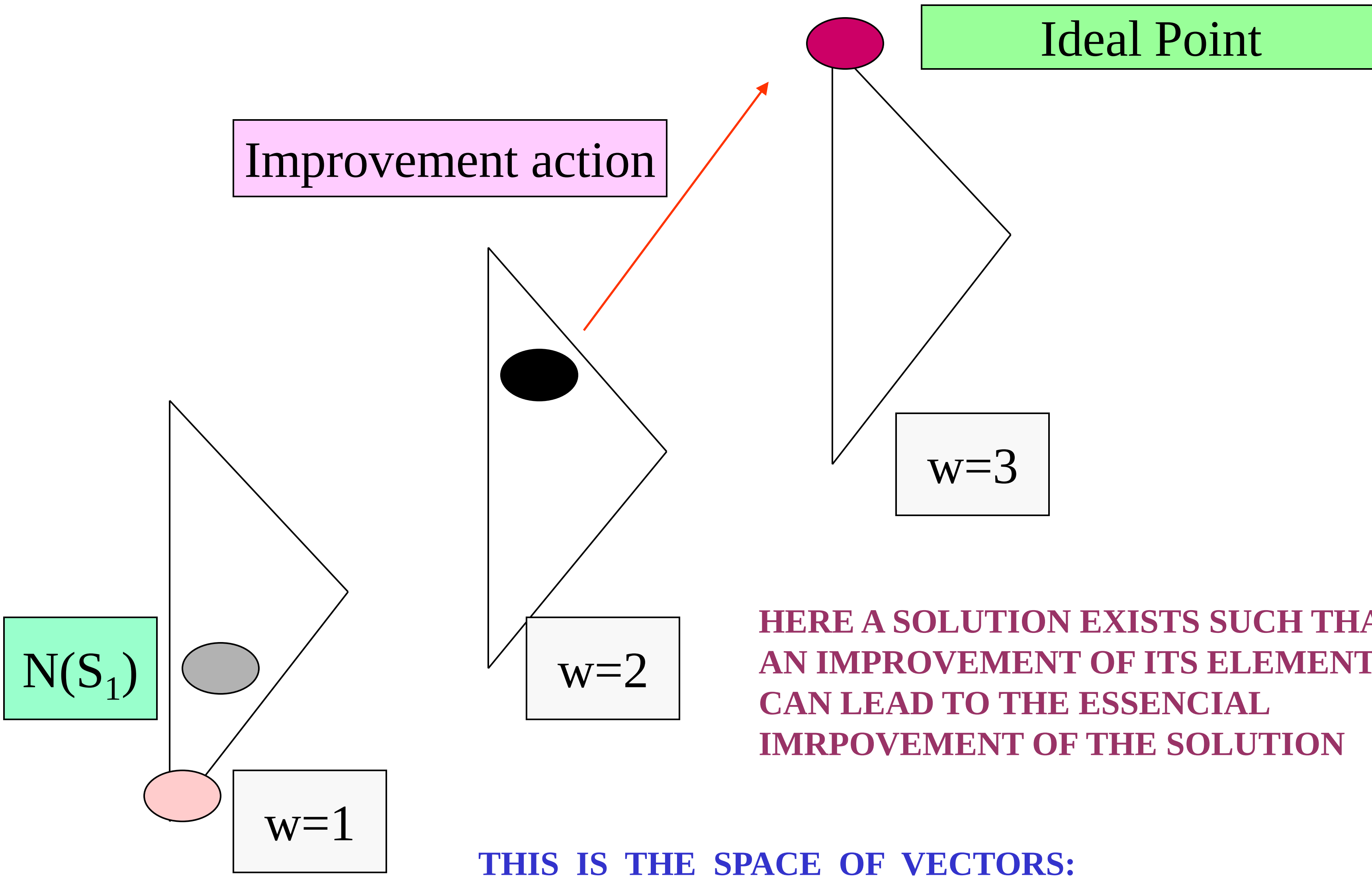
7.1
7.2
7.3
7.4
7.5
7.6
7.7
7.8
7.9
7.10
7.11

8.1
8.2
8.3
8.4

Pareto-effective & near Pareto-effective solutions



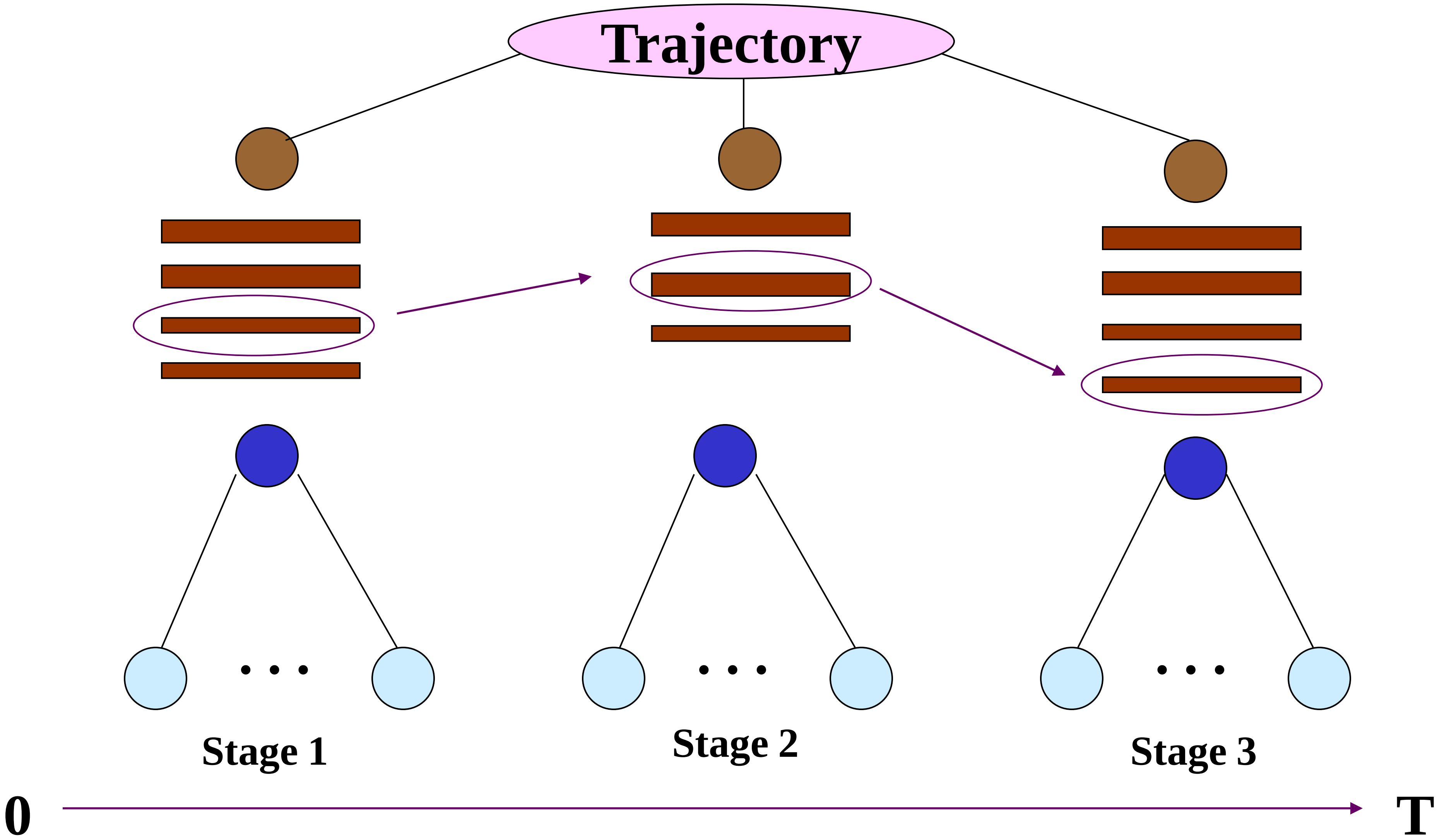
Revelation of bottlenecks via analysis of N(S)



HERE A SOLUTION EXISTS SUCH THAT AN IMPROVEMENT OF ITS ELEMENT CAN LEAD TO THE ESSENCIAL IMRPOVEMENT OF THE SOLUTION

THIS IS THE SPACE OF VECTORS:
N(S) = (w(S); n1(S) , n2(S) , n3(S))

Application Example 3: Multistage design



LECTURE 22. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.22. System improvement (examples), system evolution (example)

PLAN:

1Hierarchical Morphological Multicriteria Design (HMMD):

Application to system improvement (transformation, upgrade, adaptation)

2.System evolution / development:

Example for several generations of software DSS COMBI-PC

3.Example for complex system analysis and design

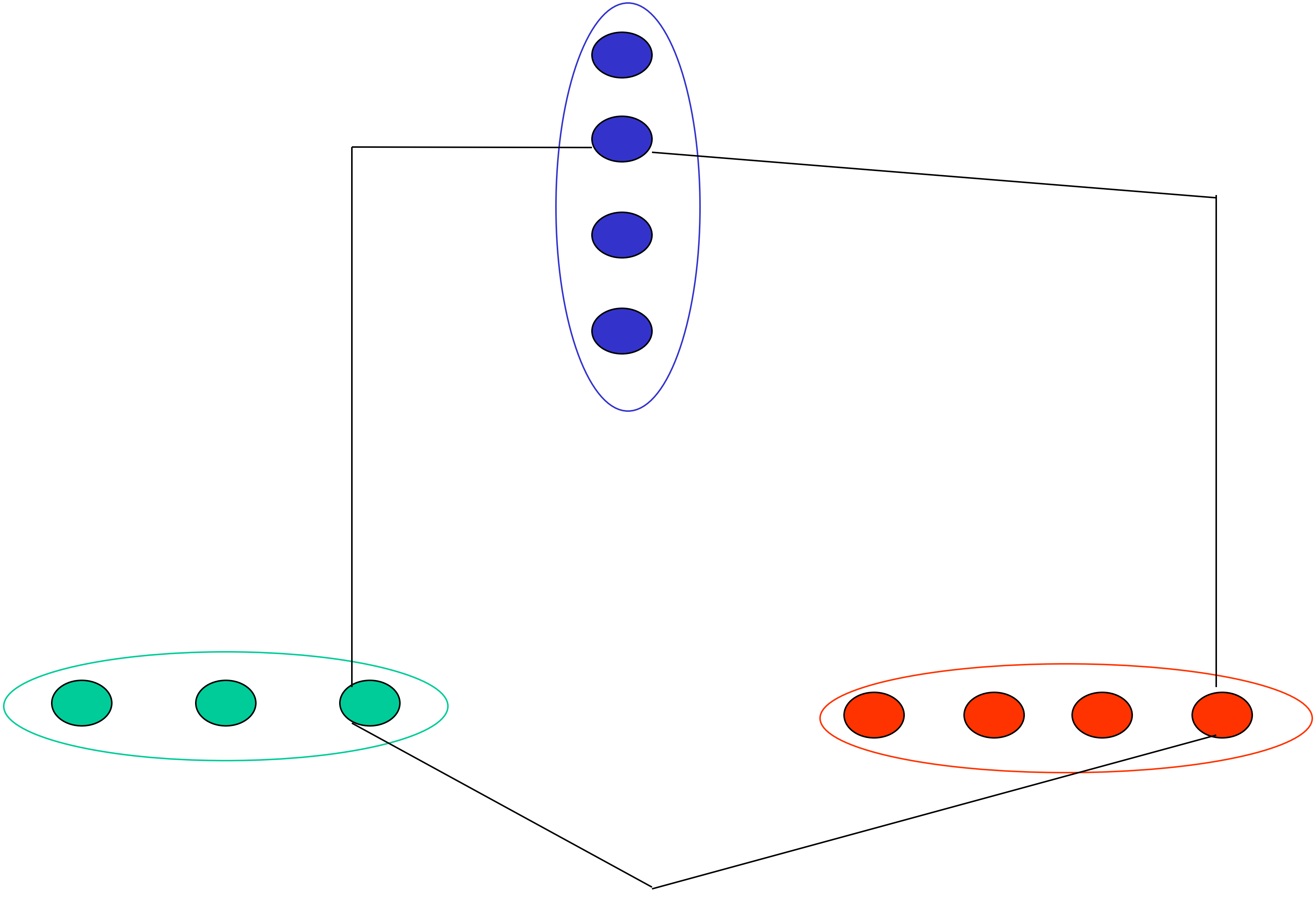
(modeling, analysis, & design of modular system):

(a) system modeling, (b) system comparison, (c) revelation of bottlenecks,

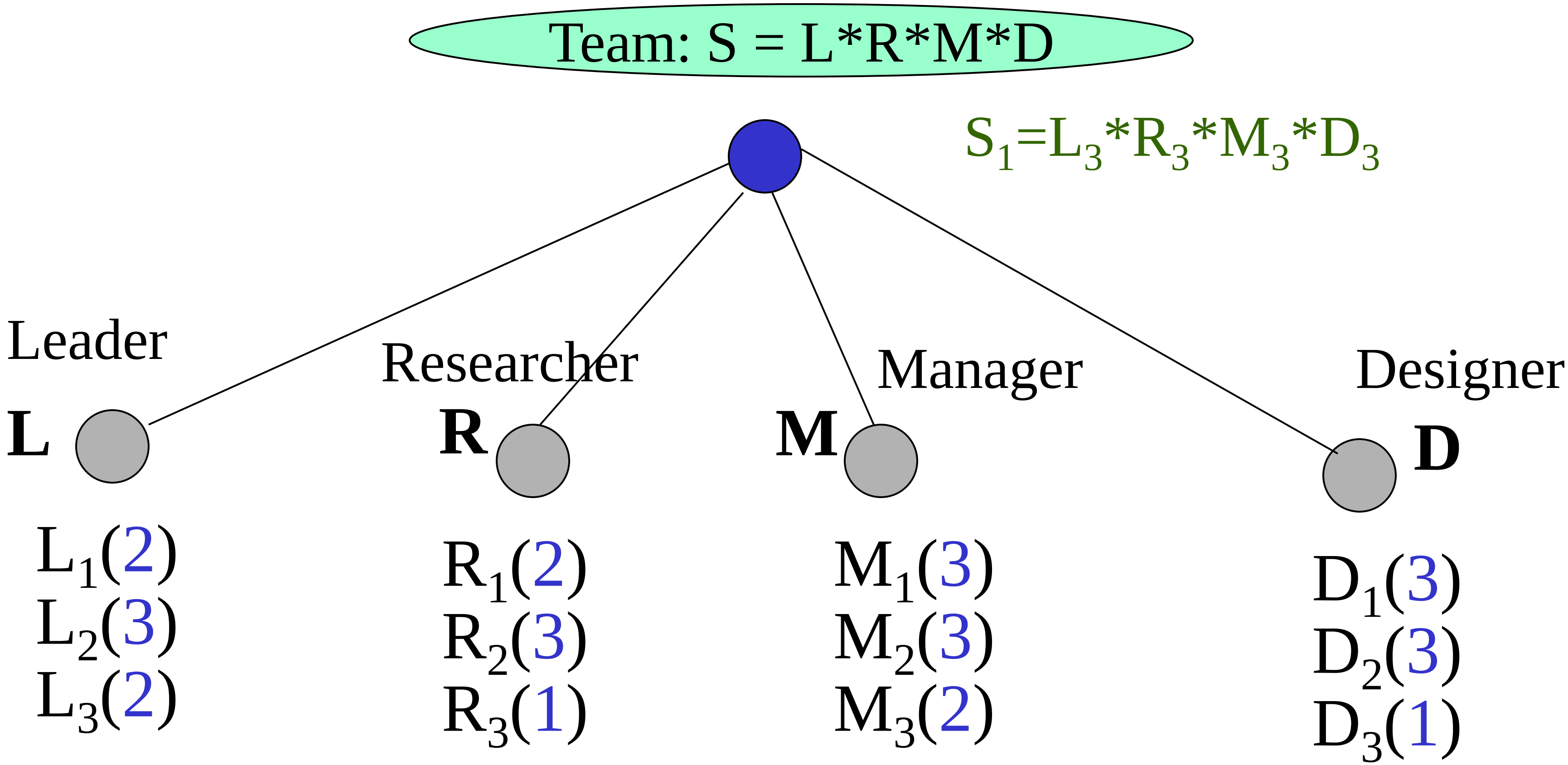
(d) system design, (e) system upgrade, (f) modeling of system generations.

Oct. 22, 2004

Morphological clique



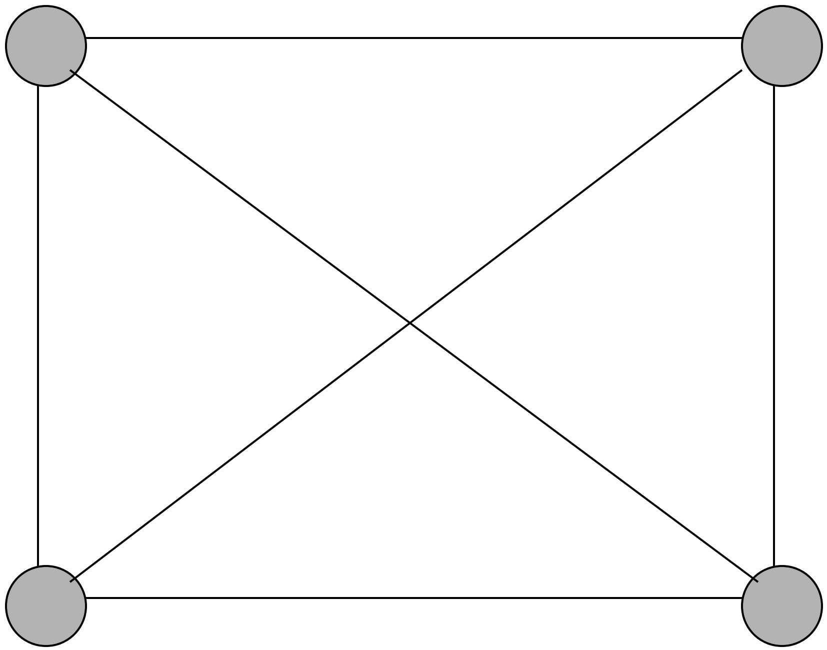
Application example 1: Design of team



Application example 1: Design of team

Team: S = L*R*M*D

$S_1=L_3*R_3*M_3*D_3$



IMPROVEMENT ACTIONS:

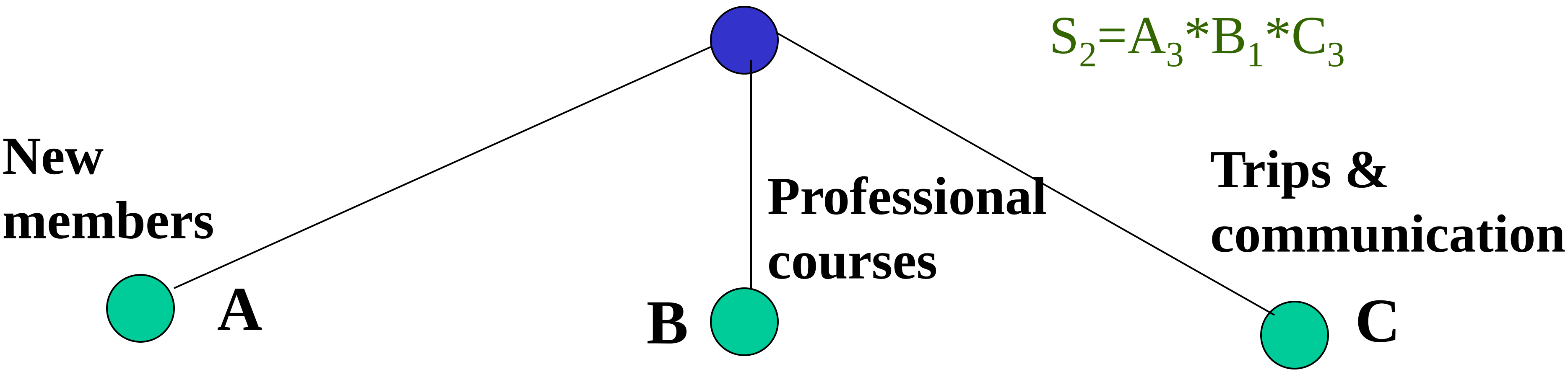
1. **New member**
2. **Improvement of member**
3. **Improvement of compatibility**
4. **New structure**

Application example: Improvement of team

Plan for improvement S = A * B * C

$S_1=A_3*B_1*C_2$

$S_2=A_3*B_1*C_3$



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

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

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

Design alternatives for improvement of a team

A_1 new leader

A_2 new manager

$A_3 = A_1$ & A_2

B_1 course on advances in science & engineering

B_2 course on foreign language

B_3 course on system analysis

B_4 course on creativity

$B_5 = B_3$ & B_4

$B_6 = B_1$ & B_4

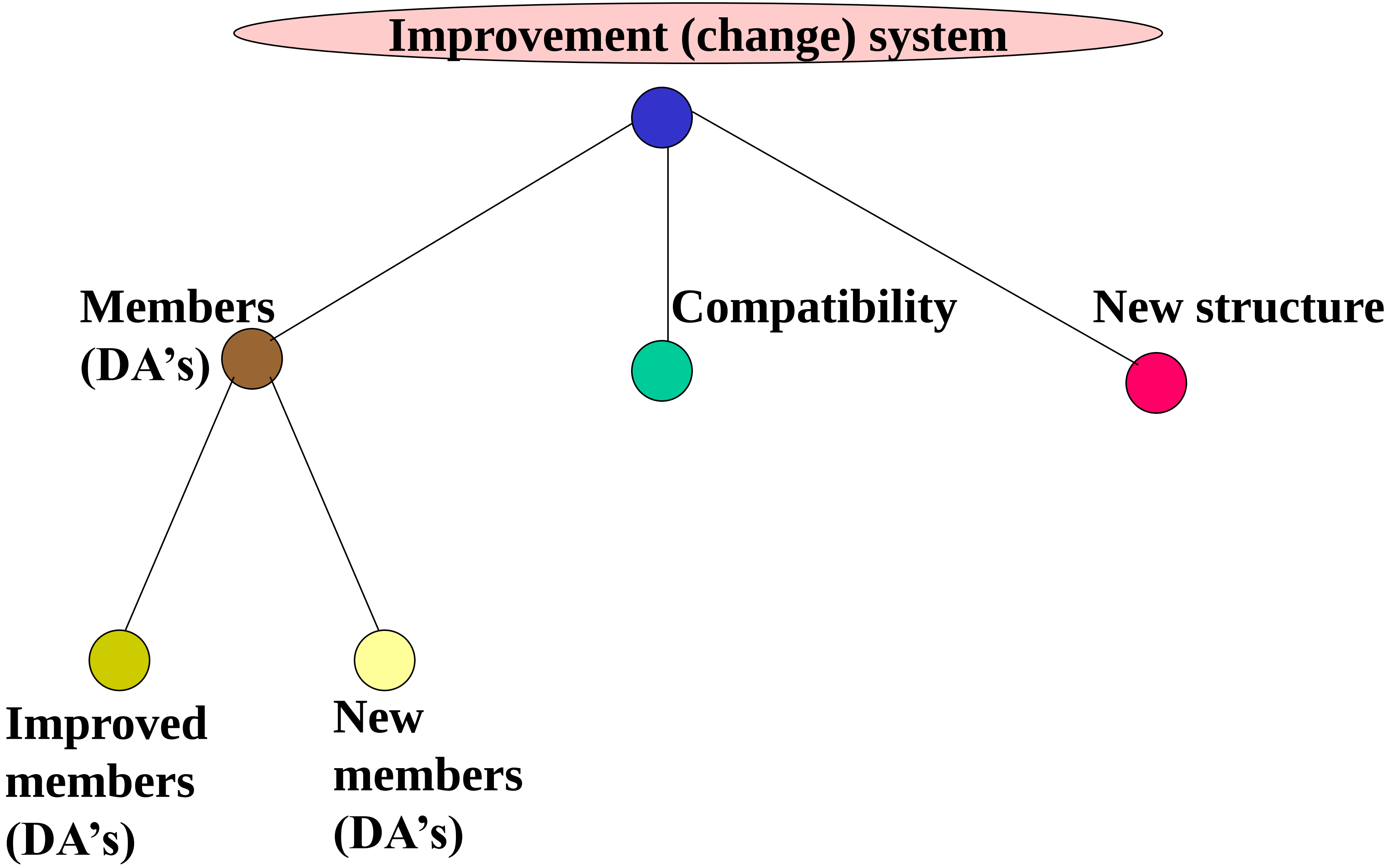
$B_7 = B_1$ & B_2 & B_4

C_1 course on human relation

C_2 joint trip to rest-home

C_3 joint participation in professional conference

$C_4 = C_1$ & C_2


Improvement (change) system
Members (DA's)
Compatibility
New structure
Improved members (DA's)
New members (DA's)

SYSTEM IMPROVEMENT PROCESS:

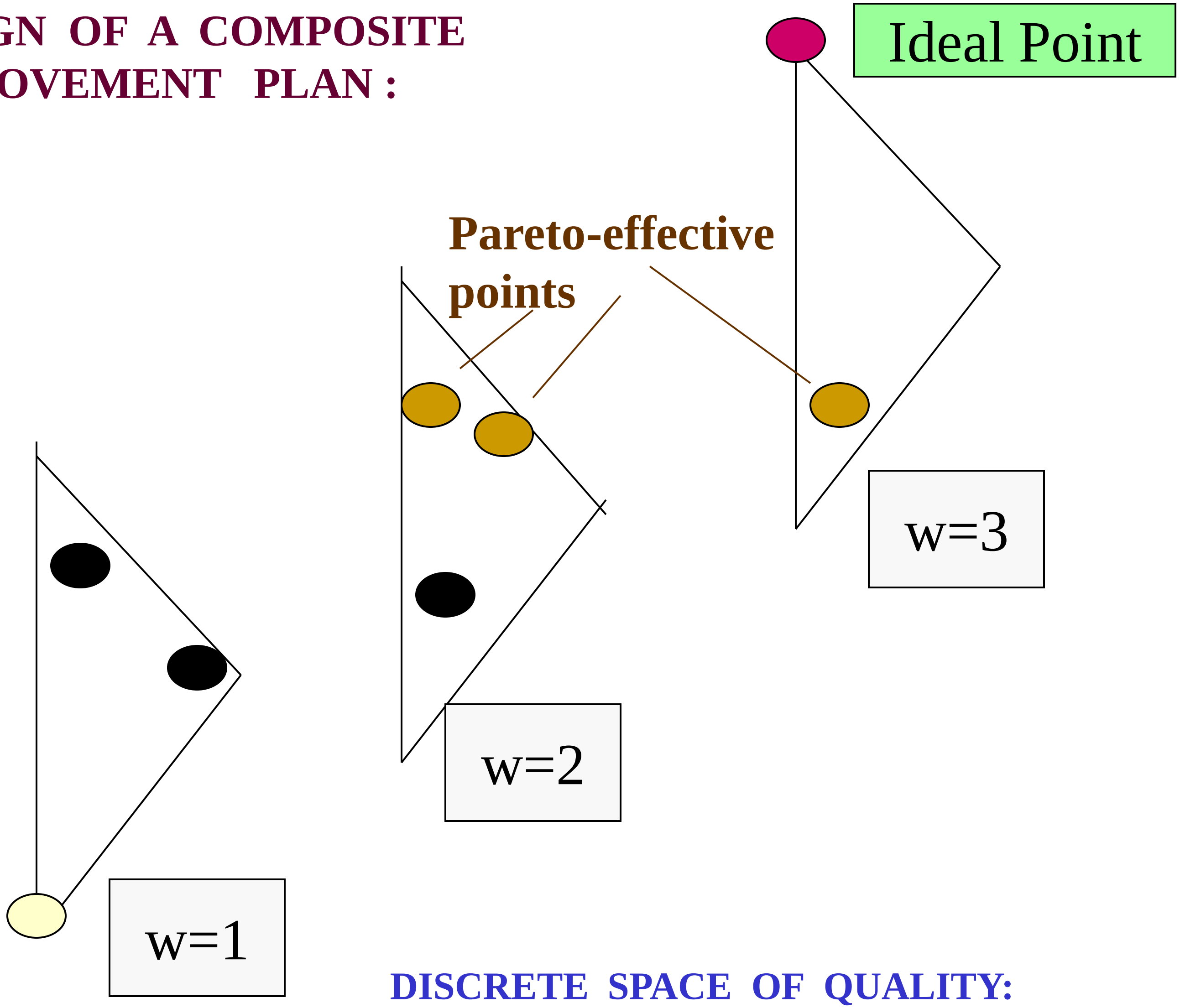
1.Improvement / Upgrade

2.System change (e.g., development / evolution)

3.System transformation

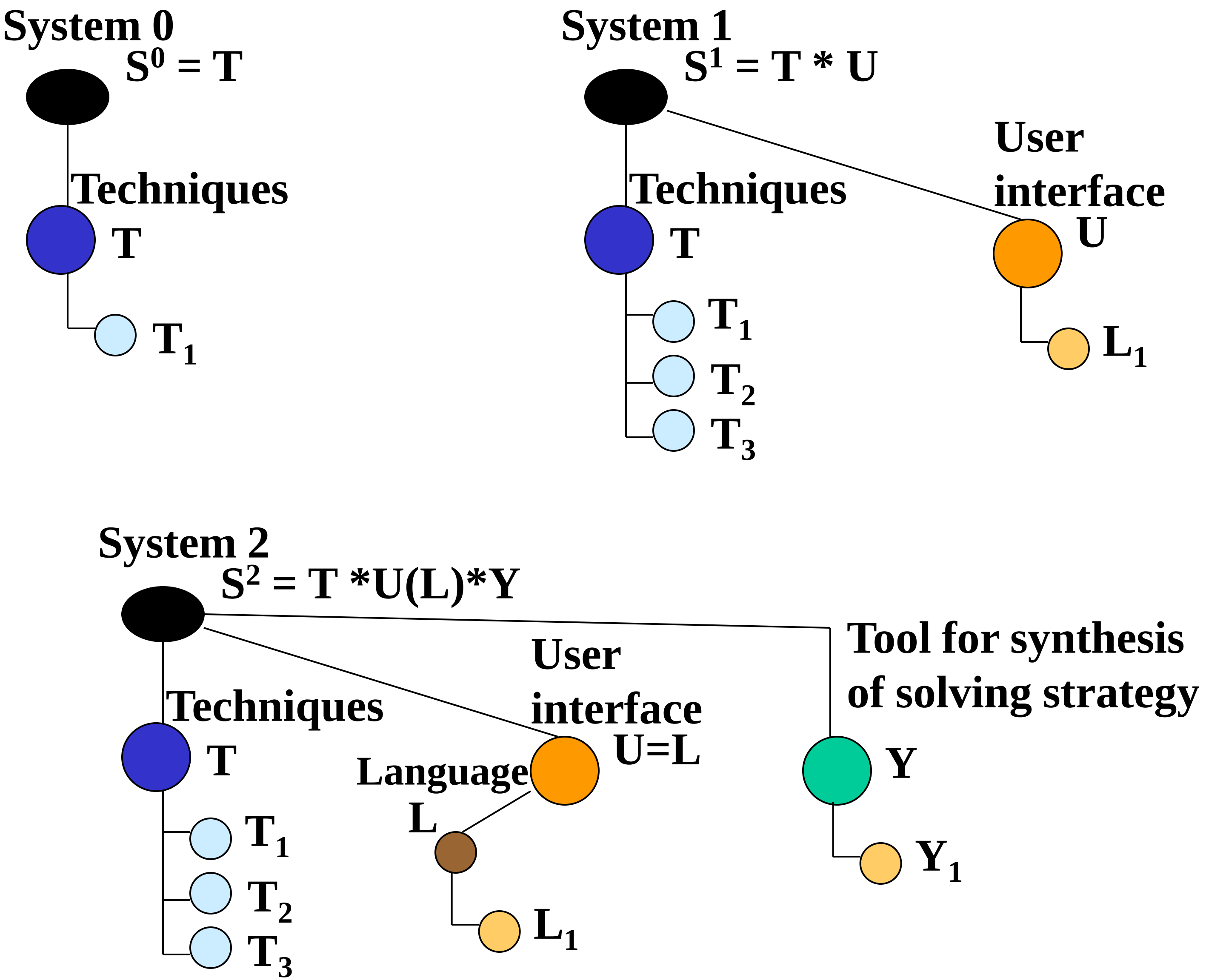
4.Adaptation (including on-line adaptation)

DESIGN OF A COMPOSITE IMPROVEMENT PLAN :

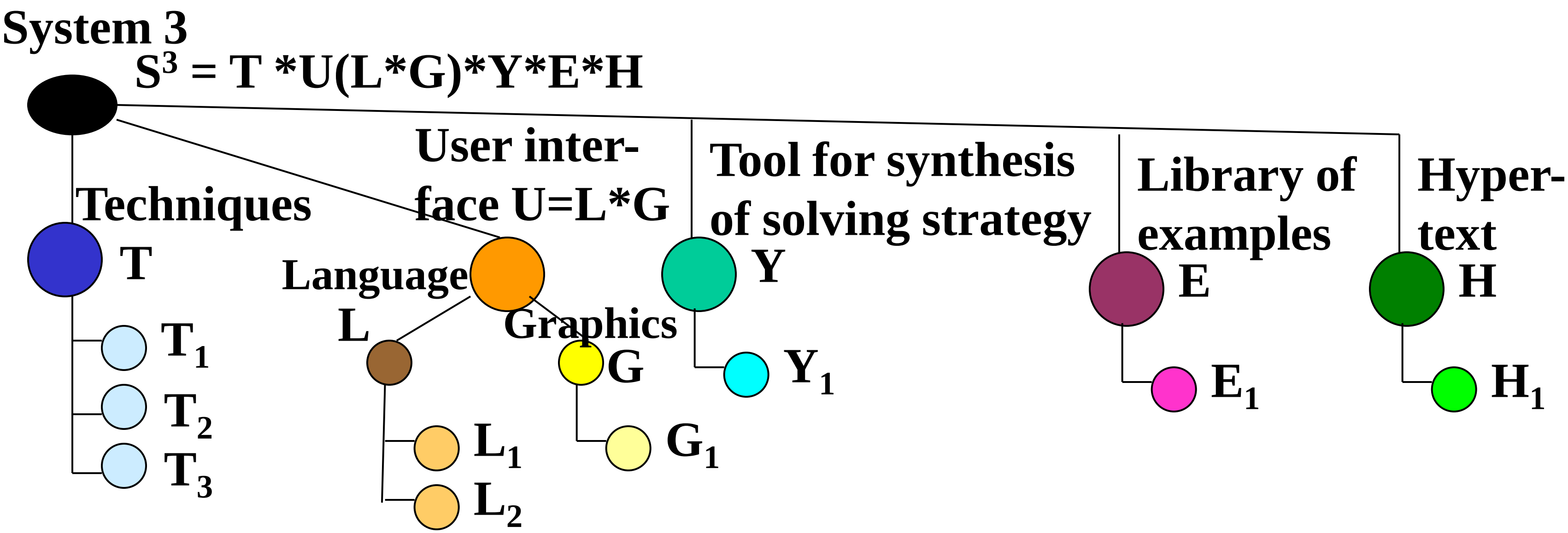


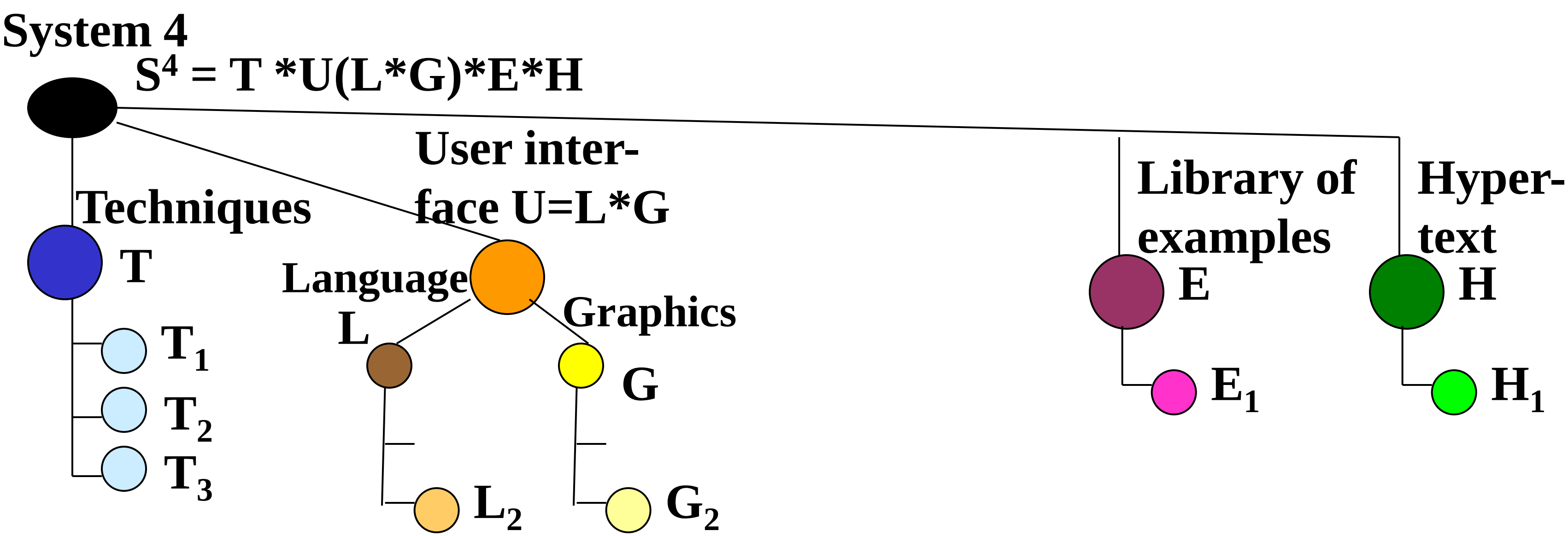
DISCRETE SPACE OF QUALITY:
N(S) = (w(S); n1(S) , n2(S) , n3(S))

Generations of software DSS COMBI-PC

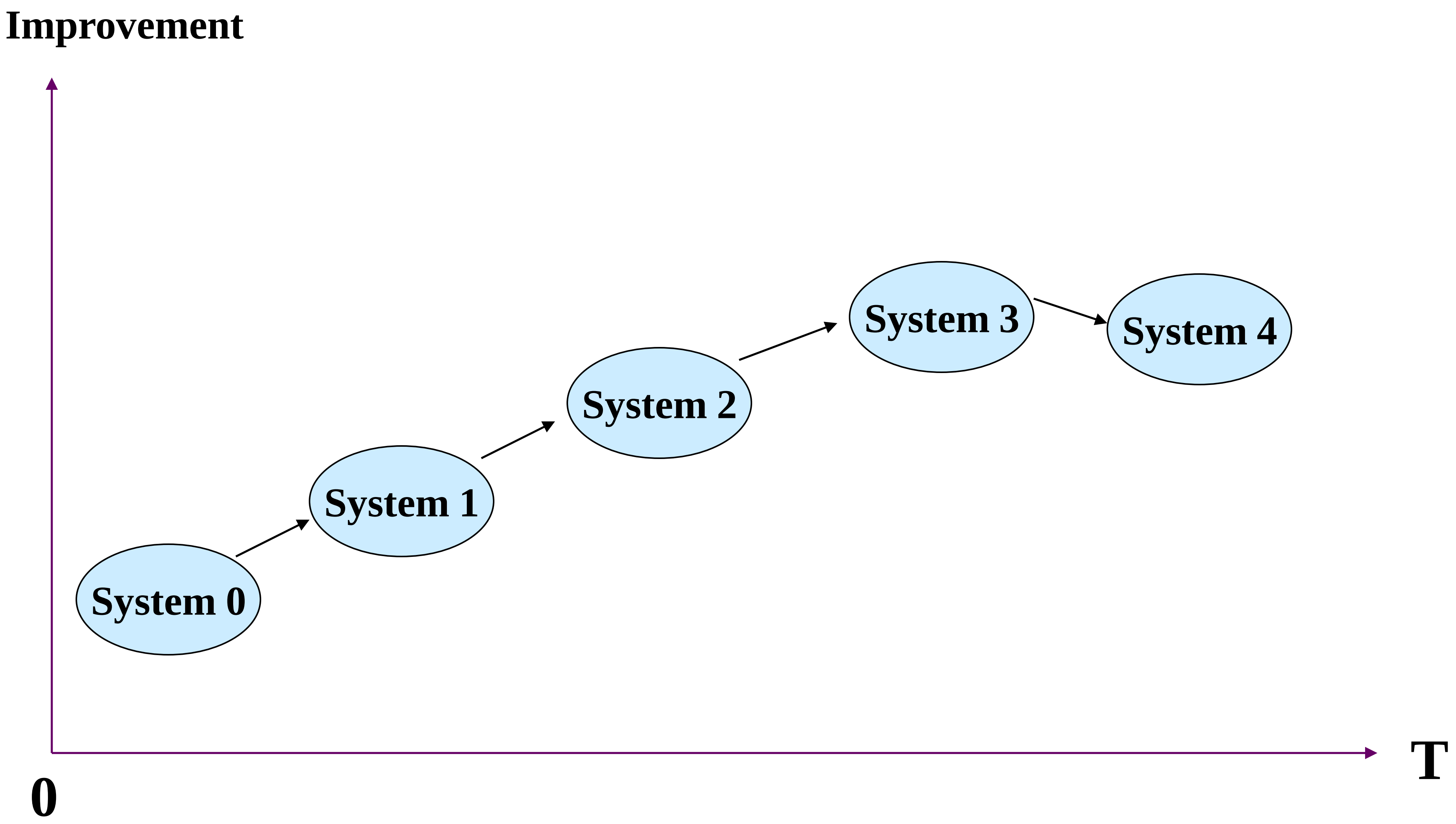


Generations of software DSS COMBI-PC





Generations of software DSS COMBI



LECTURE 23-24 (compressed version). Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.23. Basic systems problems. Example for notebook.

L.24. Systems for signal processing, system change process.

PLAN:

1.Analysis of a new domain: construction of a new world

2.Hierarchical Morphological Multicriteria Design (HMMD): framework of analysis & design problems.

3.Illustrative example for note book 4.Layers of "systems": *system, * requirements, *standards

5.Development / evolution of modular systems: illustrative examples: *notebook, *device for signal processing

6.Standard "system change" operations

7.Basic combinatorial optimization problems for system improvement / adaptation / upgrade / etc.:

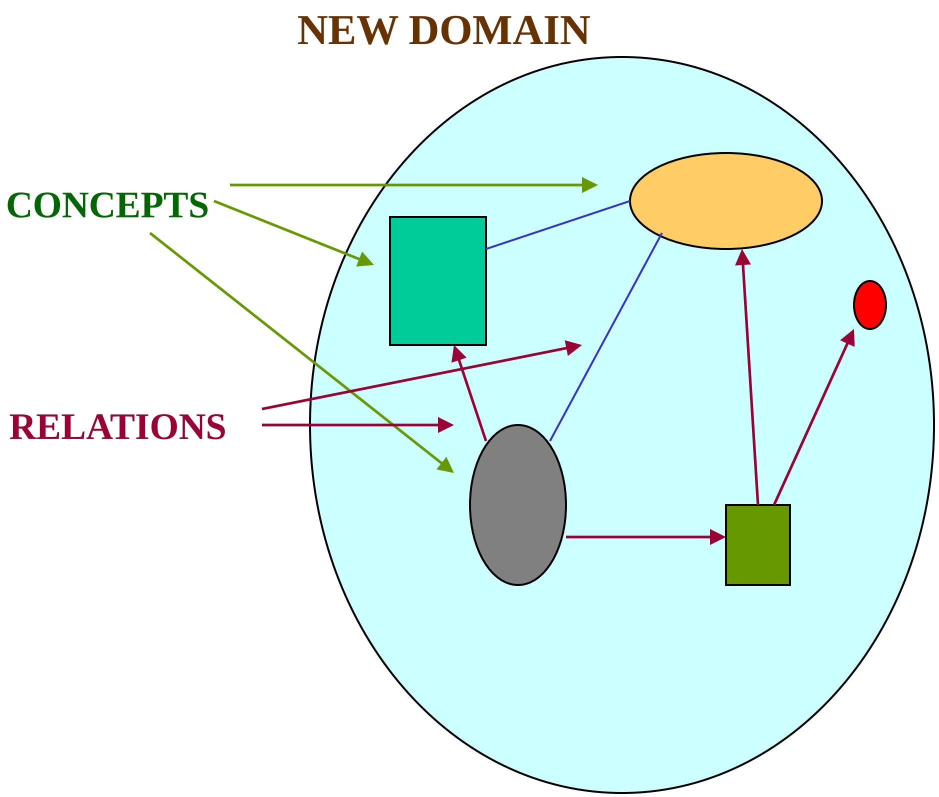
*knapsack, *multicriteria ranking, *multiple choice problem, *multicrtieria knapsack,

*multicrtieria multiple choice problem, *scheduling, *combinatorial synthesis, *multi-stage design

Oct. 23, 2004

Analysis of a new domain: construction of a new world

TWO SITUATIONS:
1.Principally New Domain
2.New Domain for Researcher



ALGORITHM SCHEME:
1.Revelation of basic concepts (objects, resources, goals, participants)
2.Revelation of basic relations over the above-mentioned concepts
3.Formulation of main problems (e.g., resource assignment, planning/ scheduling)
4.Design of solving schemes
5.Solving of numerical examples
6.Study of real (realistic) applications
7.Etc.

Our basic system design problems

SYSTEM ANALYSIS & DESIGN PROBLEMS

1.Modeling of system (structural model, e.g., and-or graph)

2.Multicriteria comparison

3.Revelation of bottlenecks

4.Hierarchical modular design

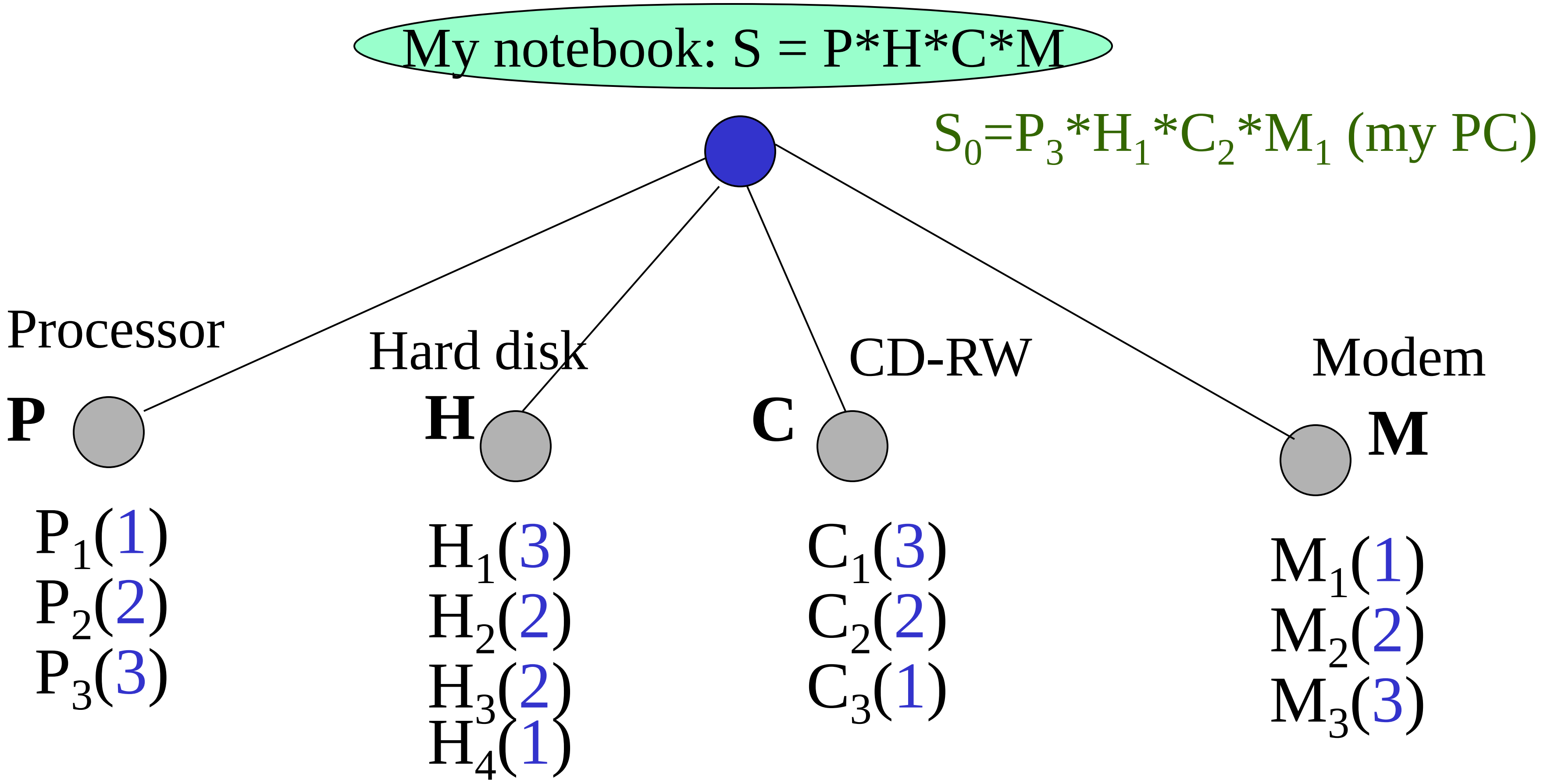
5.Upgrade (improvement, adaptation)

6.Multi-stage design

7.Modeling of development process (flow of system generations)

8.System forecasting

Example: Notebook



ALTERNATIVES:

P_1 Intel-4

P_2 Intel-3

P_3 Celeron

H_1 20Gb

H_2 40Gb

H_3 60Gb

H_4 80Gb

C_1 None

C_2 Read

C_3 Read & Write

M_1 56 Kbit / sec

M_2 48 Kbit / sec

M_3 32 Kbit / sec

Example: Notebook

COMPARISON of note books:

	Cost (-)	Reliability (+)	Maintenance-ability(+)	Upgrade-ability(+)	Total
1.Alternative 1	1300 (6)	5	3	5	2 (1)
2.Alternative 2	1250 (5)	4	3	4	3 (3)
3.My note book	900 (2)	4	4	5	1 (1)
4.Alternative 3	1200 (4)	5	3	4	3 (2)
5.Alternative 4	1200 (4)	5	3	3	3 (3)
6.Alternative 5	1100 (3)	4	4	4	2 (2)
7.Moscow’s PC	700 (1)	3	5	5	1 (1)
8.Alternative 6	1200 (4)	4	3	3	4 (4)

Weights of criteria:

2 2 3 2

2 5 4 5

Example: Notebook

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

BOTTLENECKS:

	Cost of upgrade (-)	Reliability (-)	Damage (+)	Total
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)

Weights of criteria:

1 1 1

2 4 5

Assessment of compatibility between alternatives (example for notebook)

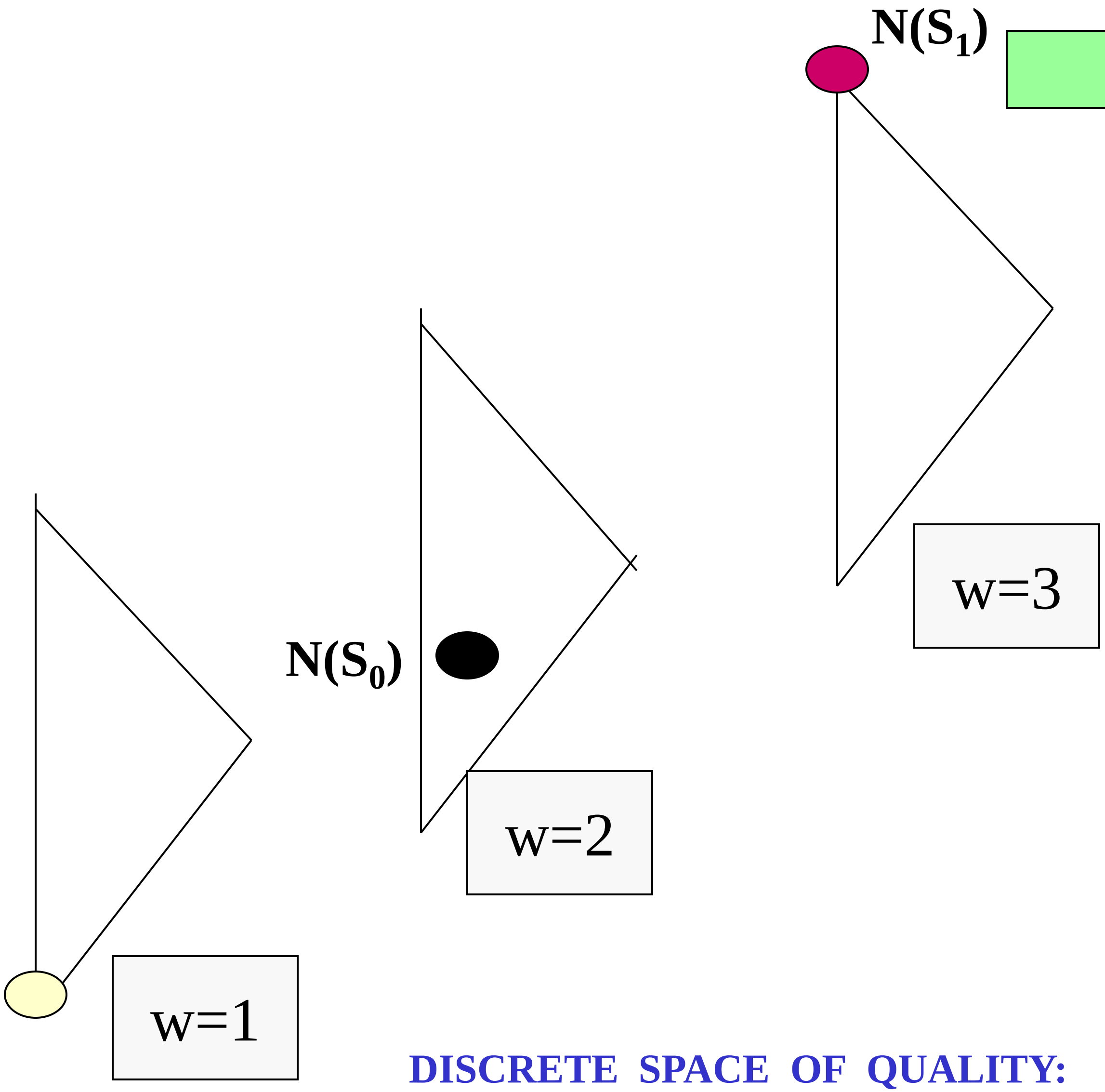
	H_1	H_2	H_3	H_4	C_1	C_2	C_3	M_1	M_2	M_3
P_1	1	2	3	3	0	2	3	3	1	1
P_2	3	2	1	1	1	3	3	2	3	1
P_3	2	3	1	2	0	3	2	3	2	1
H_1					1	3	1	3	3	2
H_2					1	3	2	3	3	1
H_3					1	2	3	3	2	1
H_4					1	2	3	3	2	1
C_1								3	2	2
C_2								3	3	1
C_3								3	3	1

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

The best combination is : $S_1 = P_1 * H_4 * C_3 * M_1$ $N(S_1) = (3 ; 4,0,0)$

Upgrade (example for notebook)

Ideal Point



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

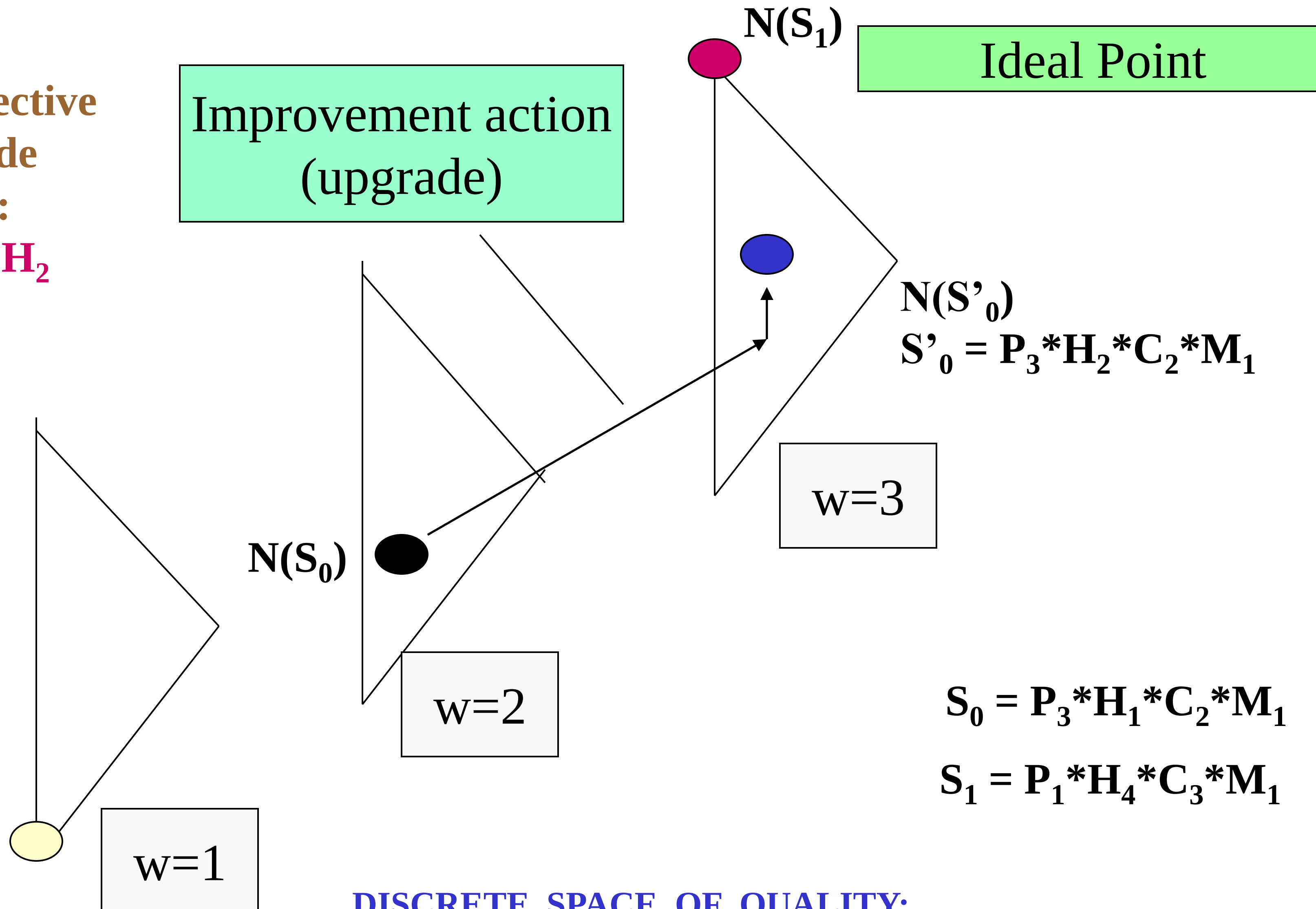
$$S_1 = P_1*H_4*C_3*M_1$$

DISCRETE SPACE OF QUALITY:
N(S) = (w(S); n1(S) , n2(S) , n3(S))

Upgrade (example for notebook)

Prospective upgrade action:

$H_1 \Rightarrow H_2$



$S'_0 = P_3*H_2*C_2*M_1$

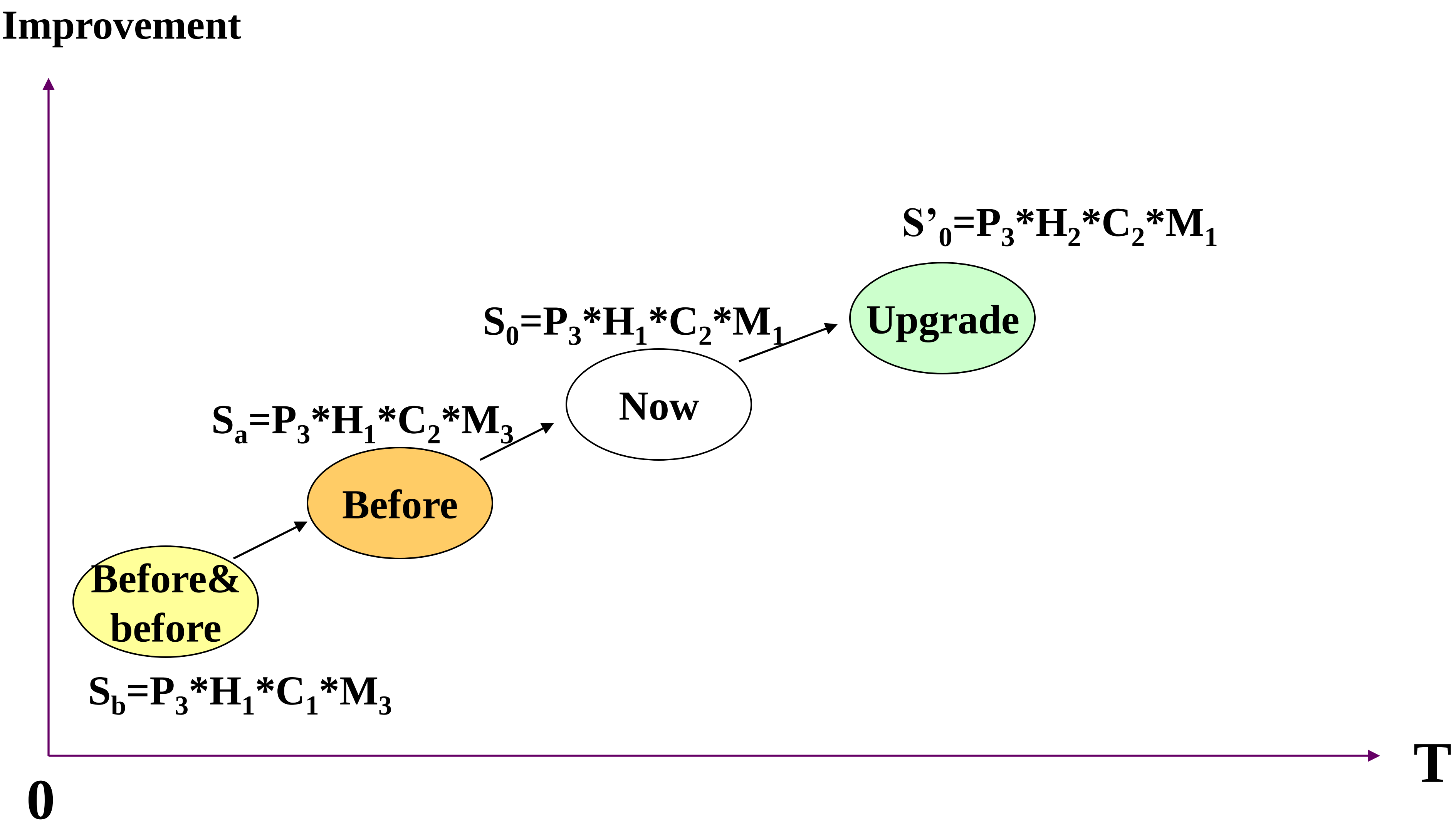
$S_0 = P_3*H_1*C_2*M_1$

$S_1 = P_1*H_4*C_3*M_1$

DISCRETE SPACE OF QUALITY:

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

Example for notebook



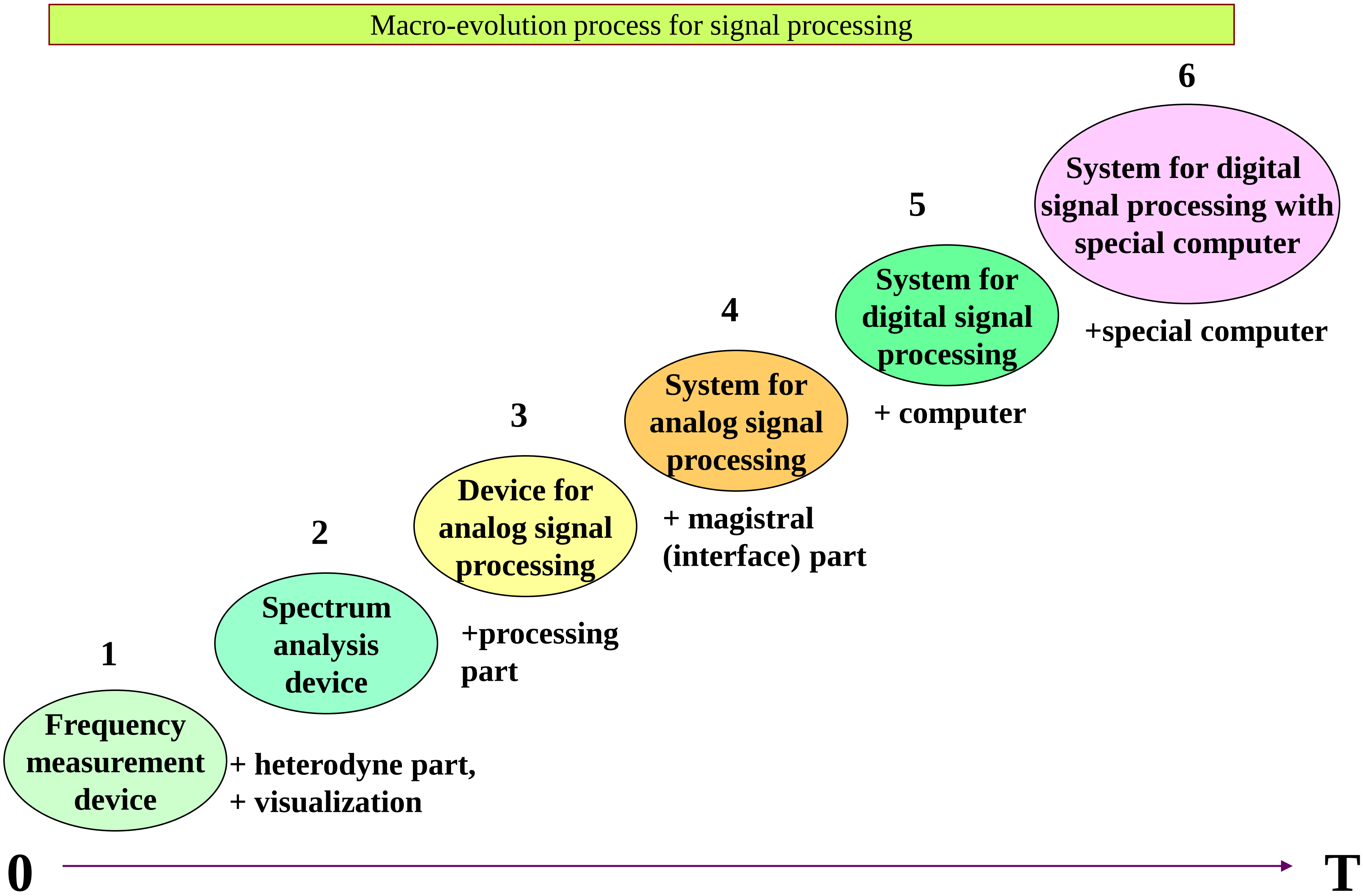
Layers of system development / evolution

STANDARDS to system

REQUREMENTS / CRITERIA to system

SYSTEM

0 → **T**


Macro-evolution process for signal processing
1
Frequency measurement device
+ heterodyne part,
+ visualization
2
Spectrum analysis device
+processing part
3
Device for analog signal processing
+ magistral (interface) part
4
System for analog signal processing
+ computer
5
System for digital signal processing
+special computer
6
System for digital signal processing with special computer
0
T

Macro-evolution process: Operations (general case)

CHANGE OPERATIONS:

I.Operations for DA's:

1.1.Change / improvement of DA's O_1: $A_i \Rightarrow A'_i$
1.2.Deletion of DA O_2
1.3.Addition of DA O_3
1.4.Aggregation of DA's O_4: $\{A_i\} \Rightarrow A^a = A_1 \& A_2 \& \ldots$
1.5.Standartization of DA's O_5: $\{A_i\} \Rightarrow A^s$

II.Operations for subsystems (parts, components):

2.1.Change / improvement of a system part O_6
2.2.Deletion of system part O_7
2.3.Addition of system part O_8
2.4.Aggregation of system part O_9

Design / planning of system change process

I.Characteristics over change operations:

1.Required resource
2.Possible profit
3.Etc.

II.Binary relations over change operations:

1.Precedence constraints ($O_i \Rightarrow O_j$)
2.Equivalence
3.Complementarity

POSSIBLE COMBINATORIAL PROBLEMS:

1.Multicriteria ranking
2.Knapsack problem
3.Multiple choice problem
4.Multicriteria knapsack problem
5.Multicriteria multiple choice problem
6.Scheduling
7.Combinatorial synthesis (modular design)
8.Multi-stage design

LECTURE 25. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.25. Design of life cycle. Systems with common modules.

PLAN:

1.Design of life cycle: illustrative example (morphological combinatorial approach)

2.Morphological combinatorial approach to multi-product system: common modules:

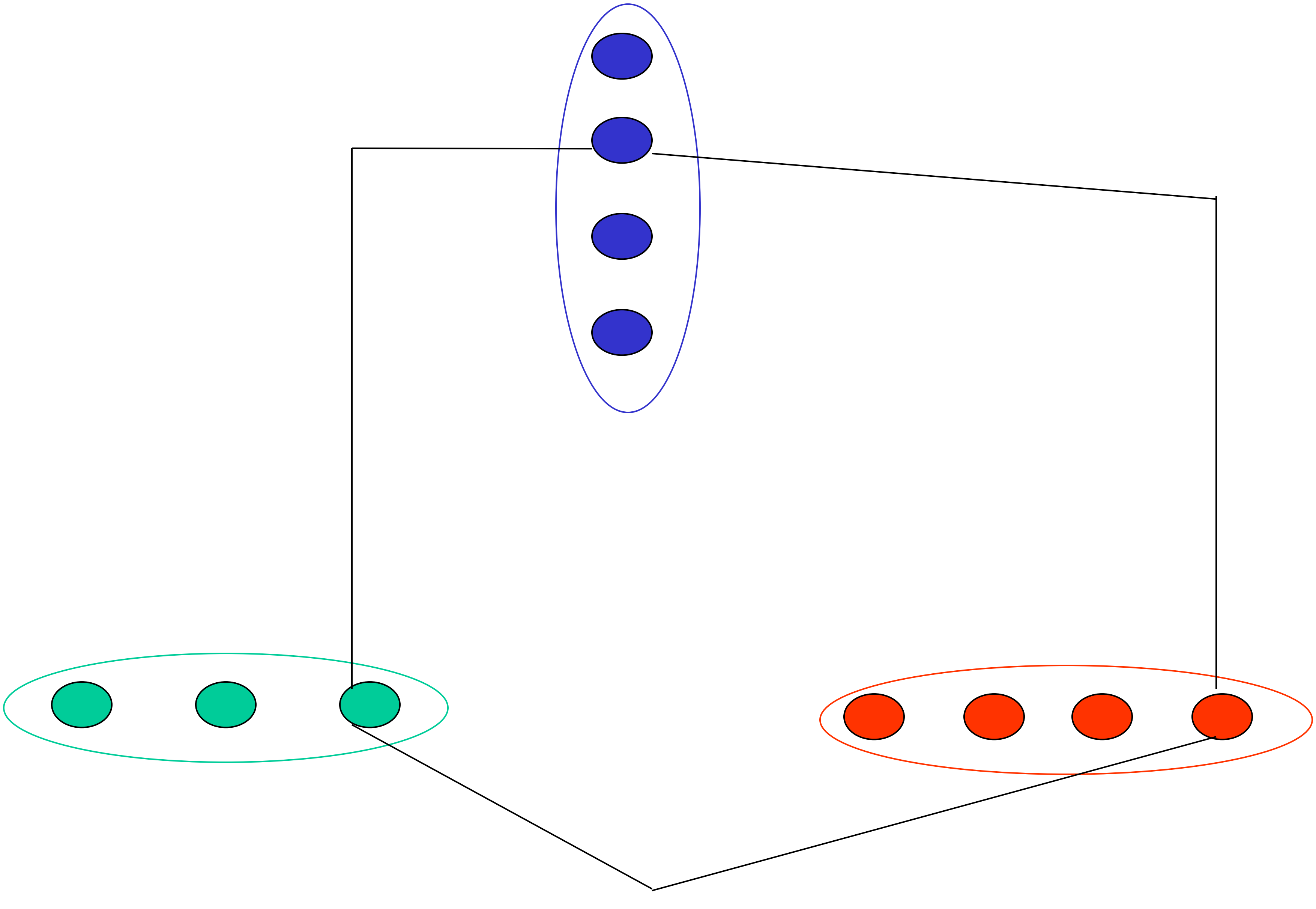
*2-product system (one common module, k common modules)

*m-product system (one common module, k common modules)

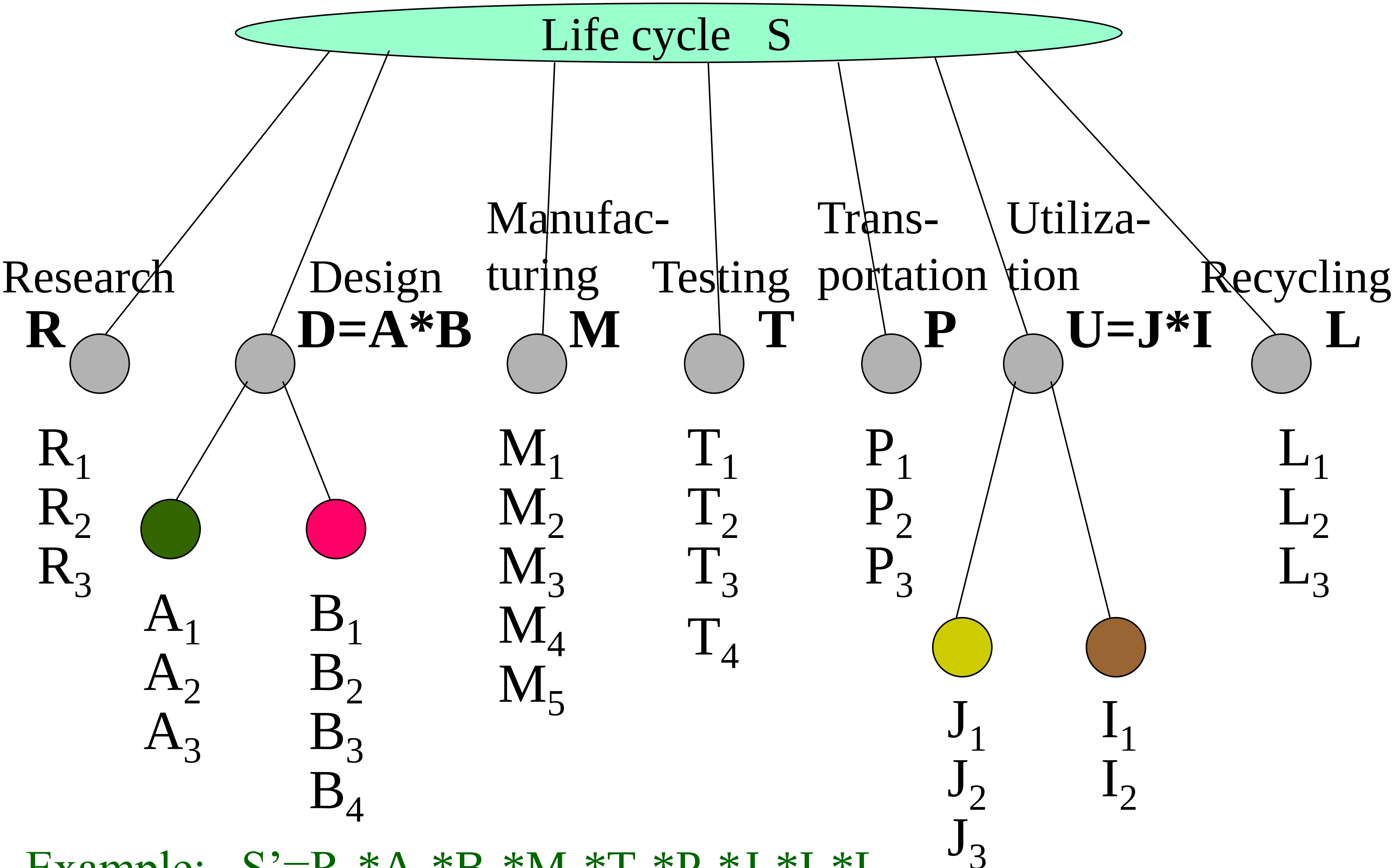
3.Recent references

Oct. 29, 2004

Morphological clique



Design (planning) of life cycle



Example: $S' = R_3 * A_1 * B_2 * M_5 * T_2 * P_1 * J_2 * I_1 * L_3$

Life cycle S

Example 1: Design & Manufacturing $S_1=D^1(A_2*B_1)*M_3$

Example 2: Design & Testing $S_2=D^2(A_1*B_3)*T_3$

Example 3: Research & Design & Manufacturing
$S_3=R_2*D'(A_3*B_1)*M_4$

Example 4: Design & Manufacturing & Transportation
$S_1=D''(A_3*B_3)*M_4*T_3$

Life cycle S

Design (Planning) of Life Cycle

Life Cycle Management

Life Cycle Engineering

Support of Life Cycle

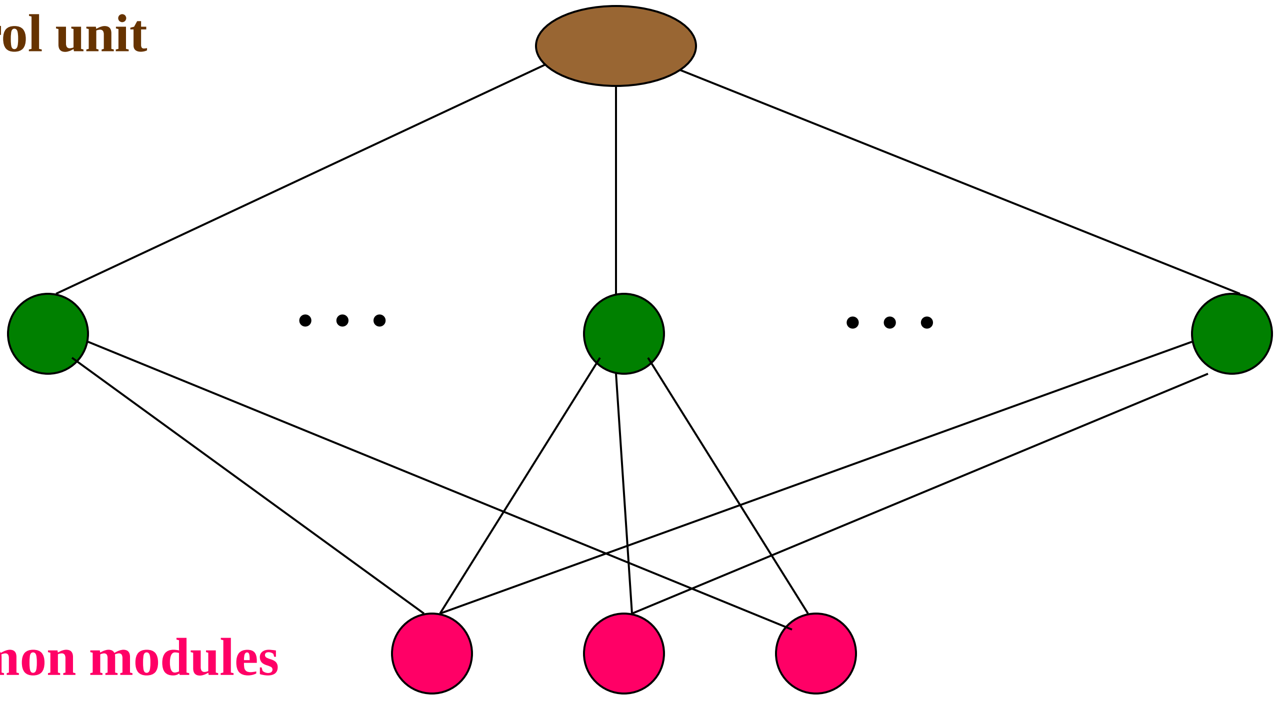
Maintenance of Life Cycle

Structure for Modular Software Package: 3 layers

Layer 1: Control unit

Layer 2: Function units

Layer 3: Common modules



General Structure for Modular Software Package

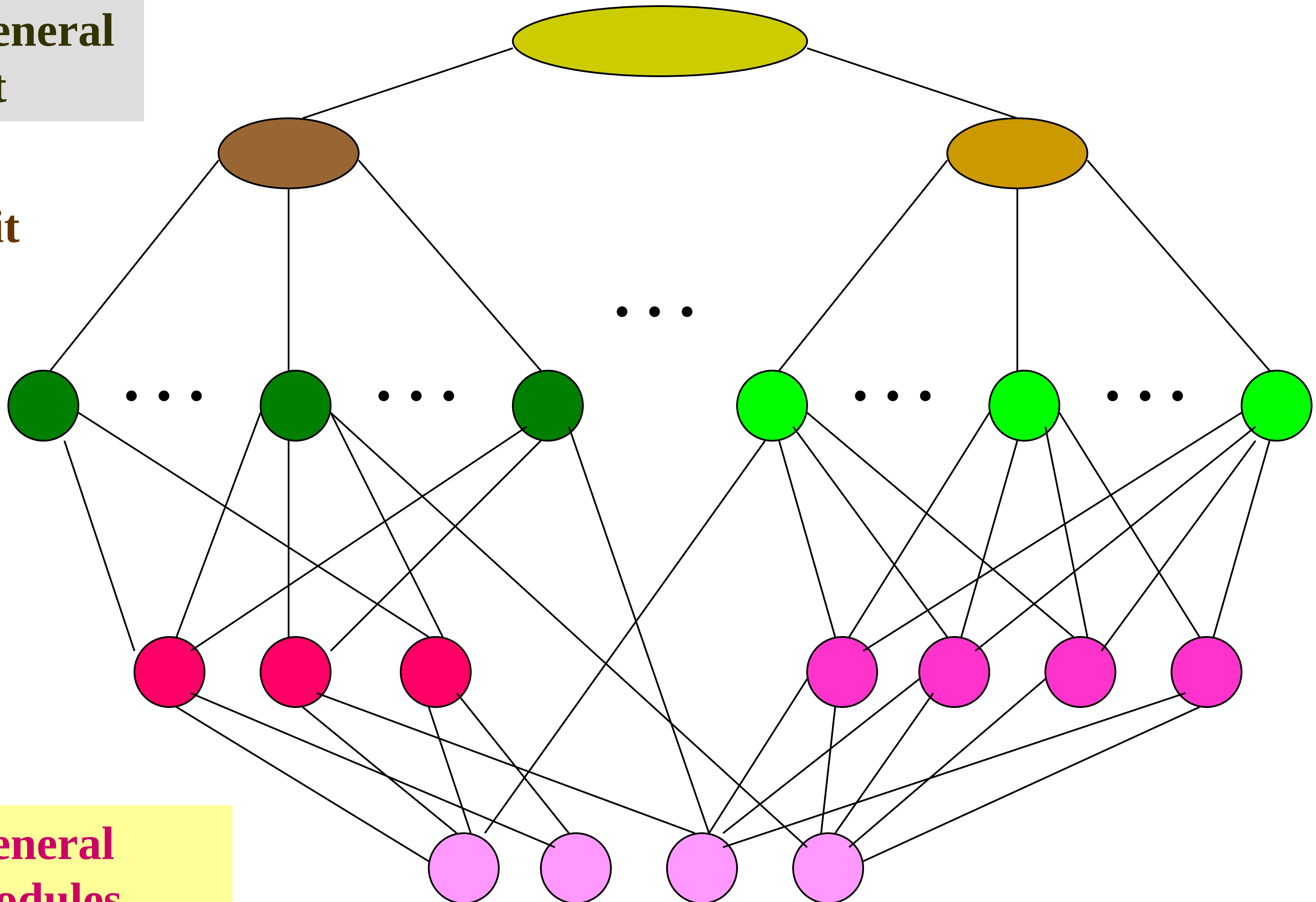
Layer 0: General control unit

Layer 1: Control unit

Layer 2: Function units

Layer 3: Common modules

Layer 4: General common modules



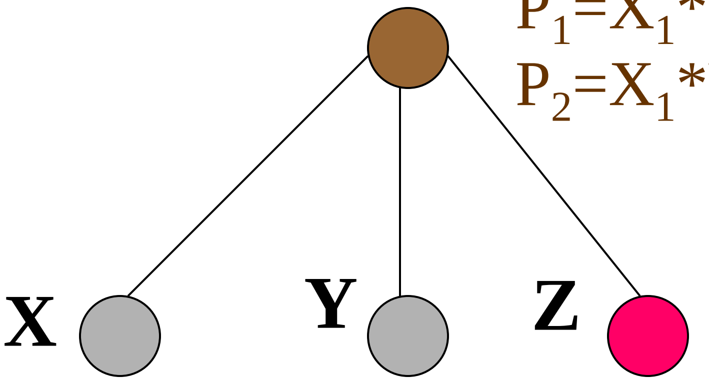
2-product system

The 1st Product

$$P = X*Y*Z$$

$$P_1 = X_1*Y_4*Z_3$$

$$P_2 = X_1*Y_1*Z_2$$



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

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

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

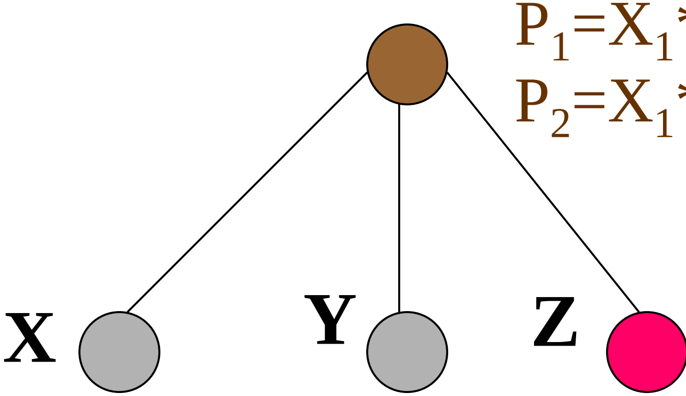
2-product system

The 1st Product

$P = X*Y*Z$

$P_1 = X_1*Y_4*Z_3$

$P_2 = X_1*Y_1*Z_2$



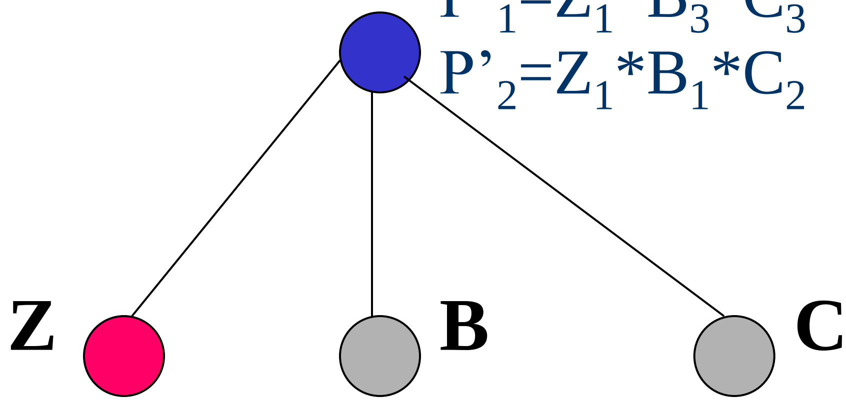
X	Y	Z
$X_1(3)$	$Y_1(3)$	$Z_1(1)$
$X_2(1)$	$Y_2(1)$	$Z_2(1)$
$X_3(1)$	$Y_3(2)$	$Z_3(2)$
	$Y_4(3)$	$Z_4(3)$

The 2nd Product

$P' = Z*B*C$

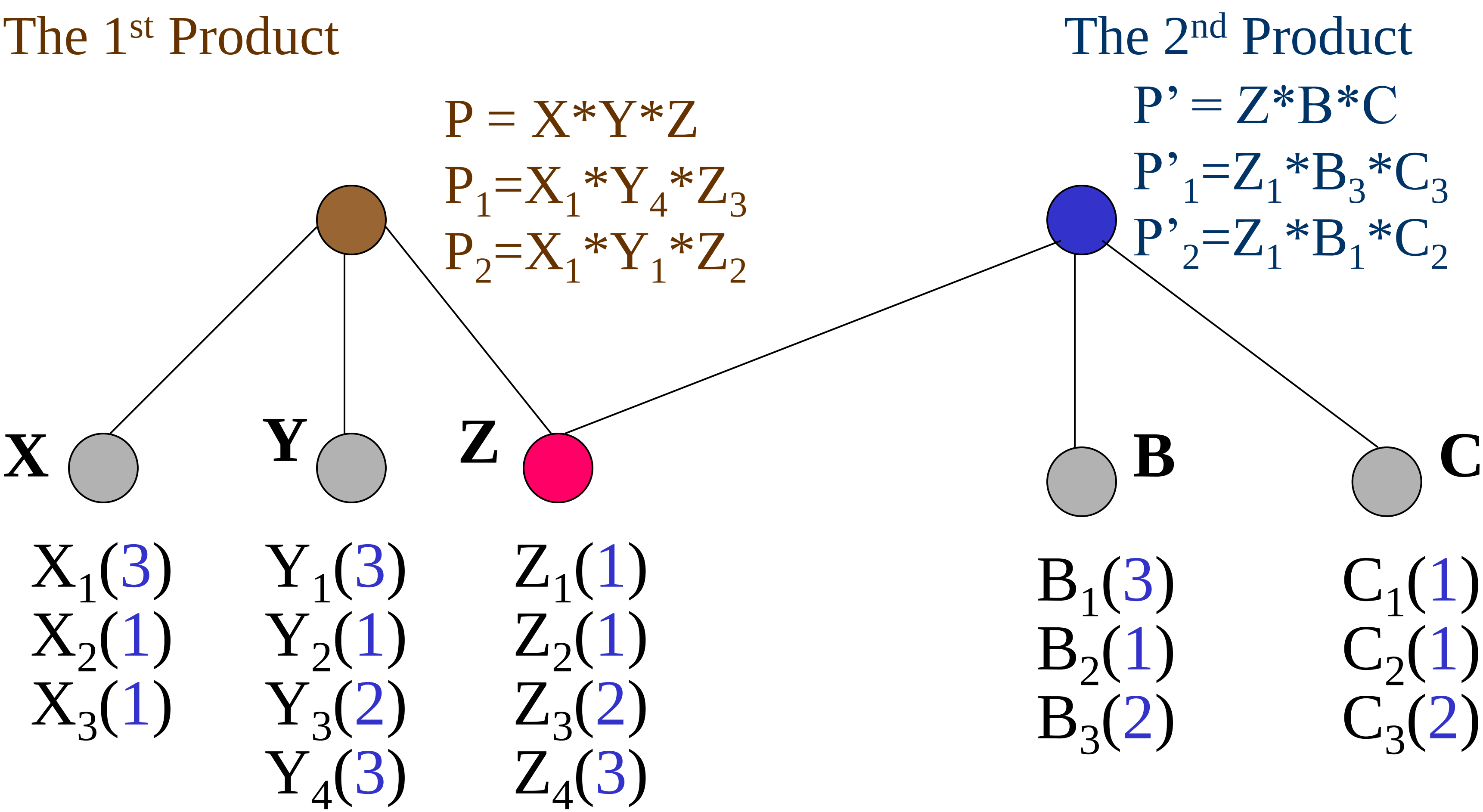
$P'_1 = Z_1*B_3*C_3$

$P'_2 = Z_1*B_1*C_2$

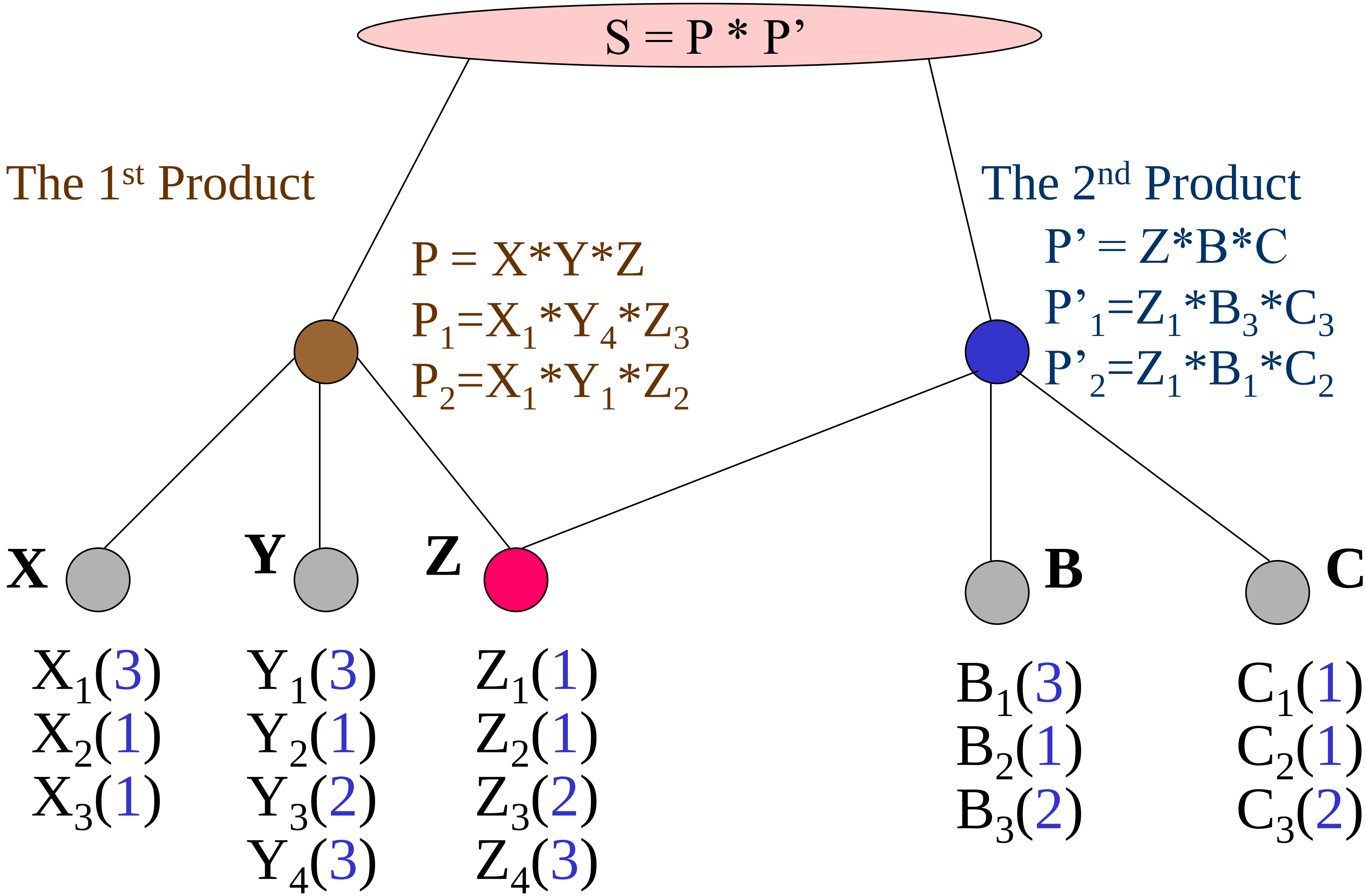


Z	B	C
$Z_1(1)$	$B_1(3)$	$C_1(1)$
$Z_2(1)$	$B_2(1)$	$C_2(1)$
$Z_3(2)$	$B_3(2)$	$C_3(2)$
$Z_4(3)$		

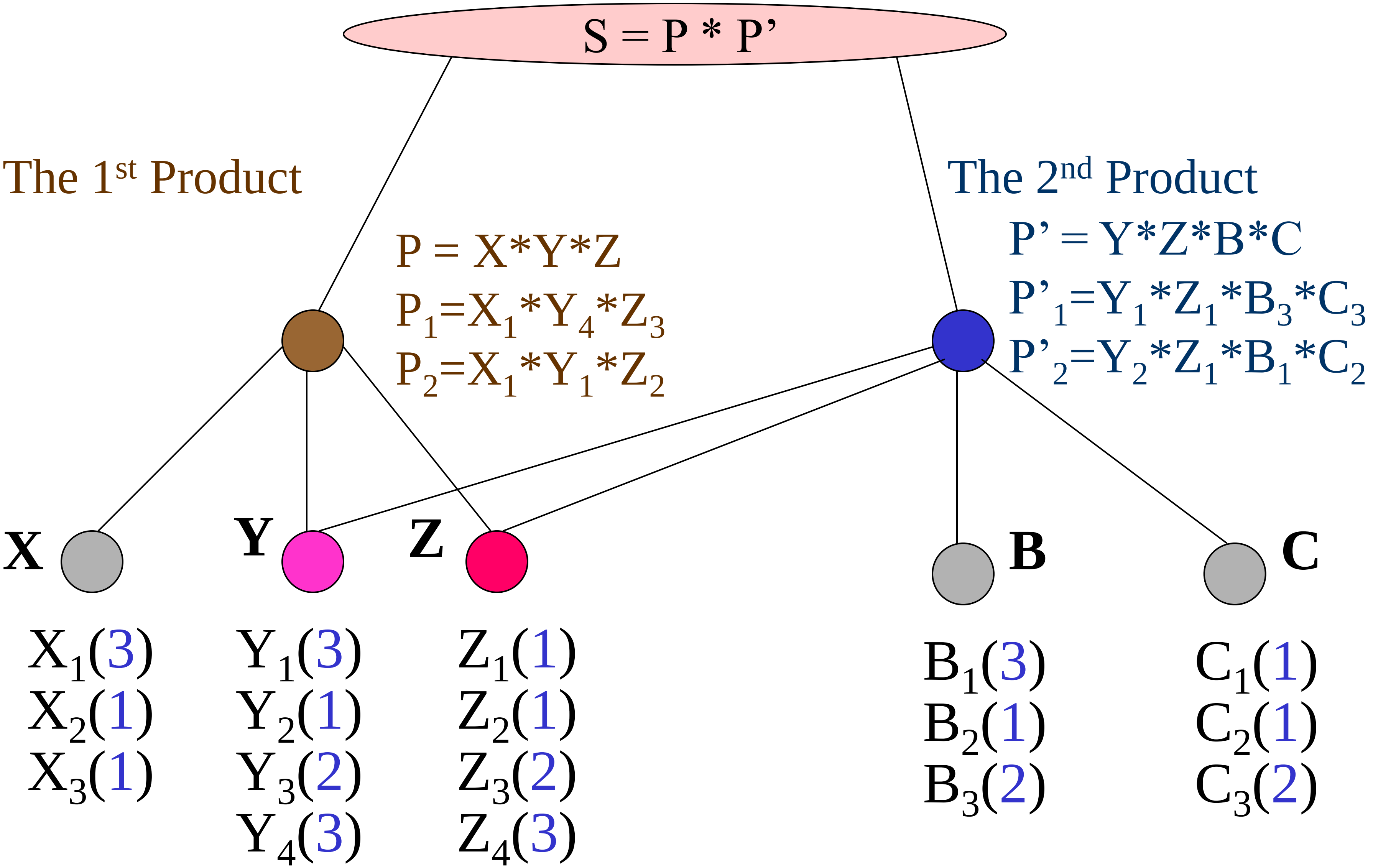
2-product system (one common module)



2-product system (one common module)



2-product system (two common modules)



Recent English References

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.

LECTURE 26. Course: “Design of Systems: Structural Approach”

Dept. “Communication Networks &Systems”, Faculty of Radioengineering & Cybernetics

Moscow Institute of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.26. System testing

PLAN:

1.System Testing: main approaches: *white-box testing (system structure is well-known)

*black-box testing as model checking *black-box testing as multi-function testing

2.Multi-function system testing: basic combinatorial problems:

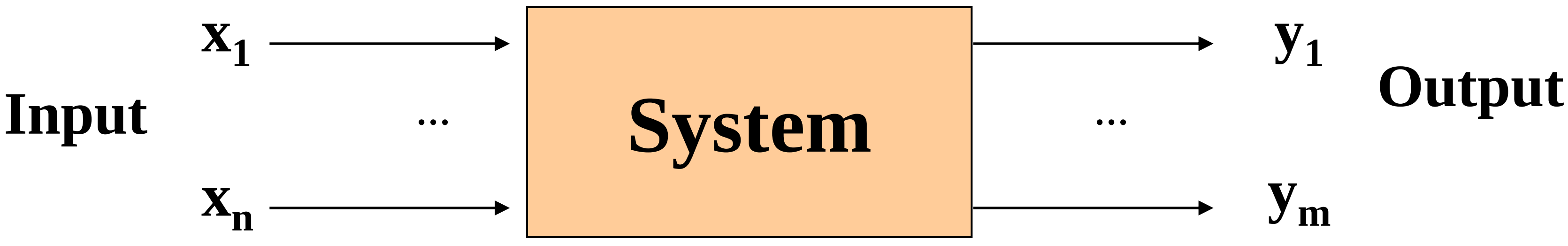
*preliminary analysis of system, *composition of test cases

*design of chain of test cases, *covering of digraph of function clusters by chains

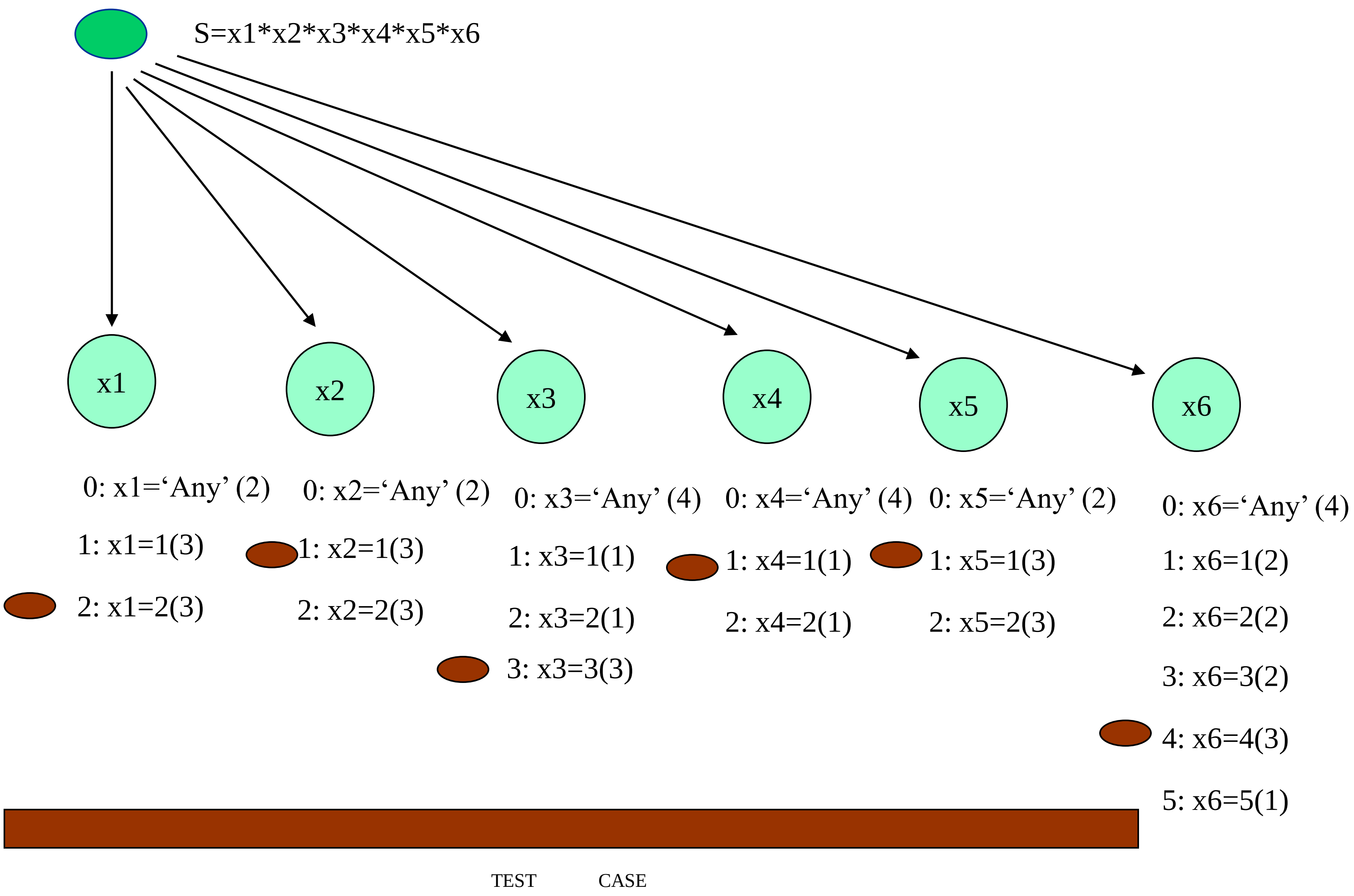
3.Illustrative example for multi-function system testing

Nov. 5 2004

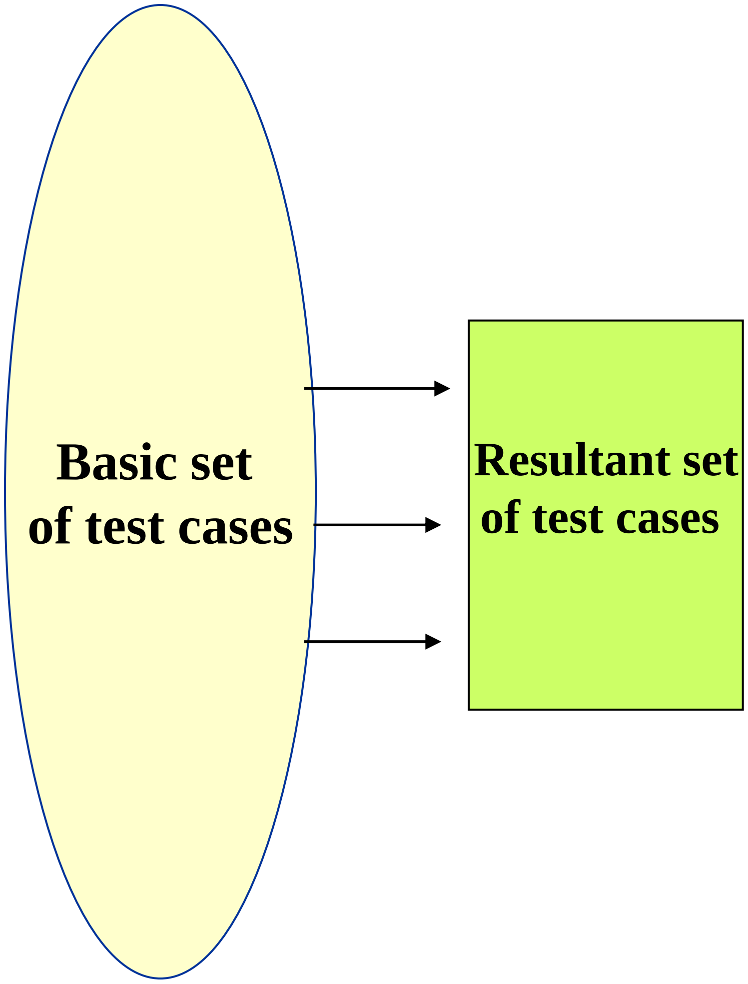
Black-box system



Example of test case structure (cluster: functions 1, 3, and 4)



Selection of test cases (basic approaches)

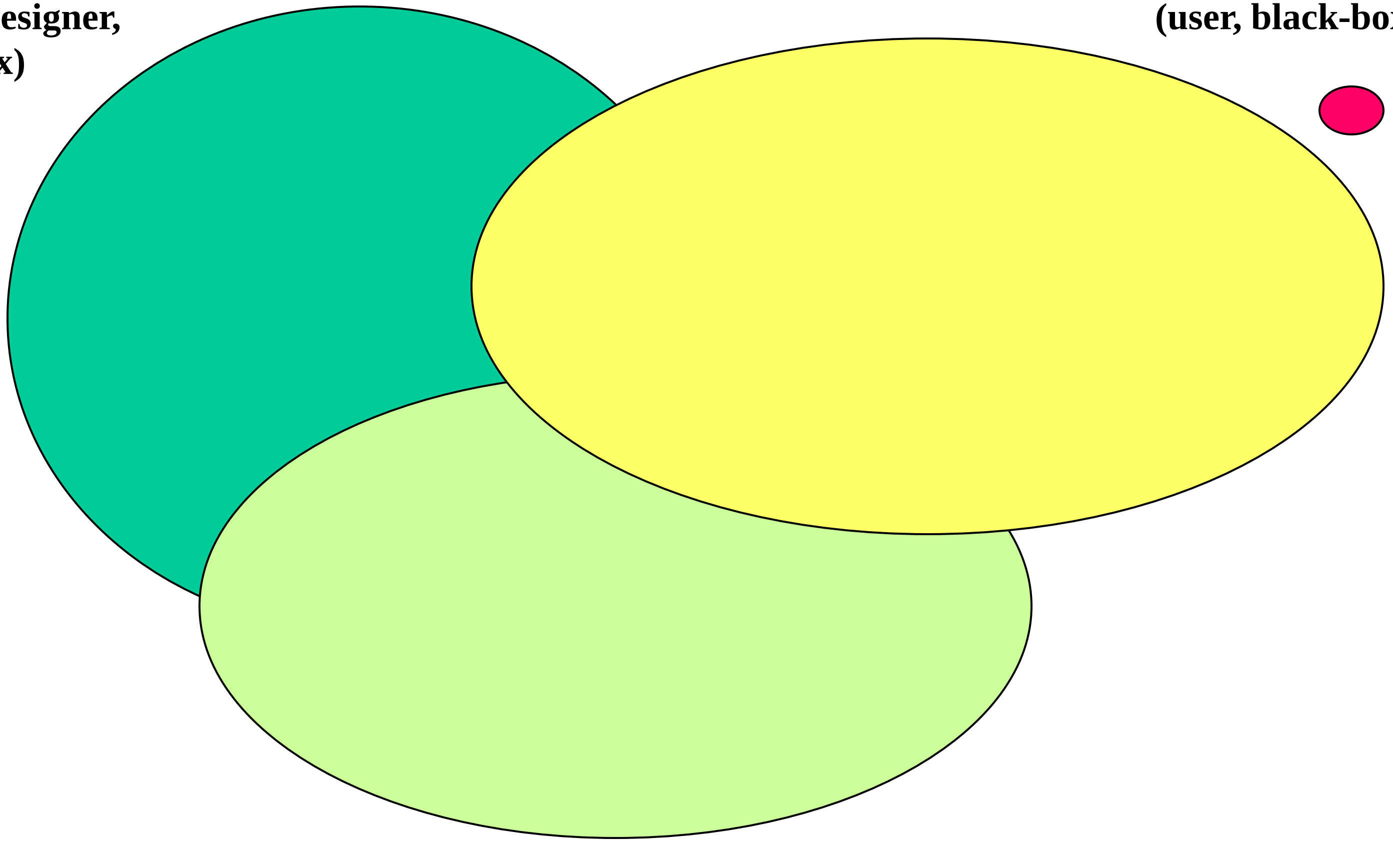


Basic algorithms:

1.Reducing of the initial space (by equivalence)

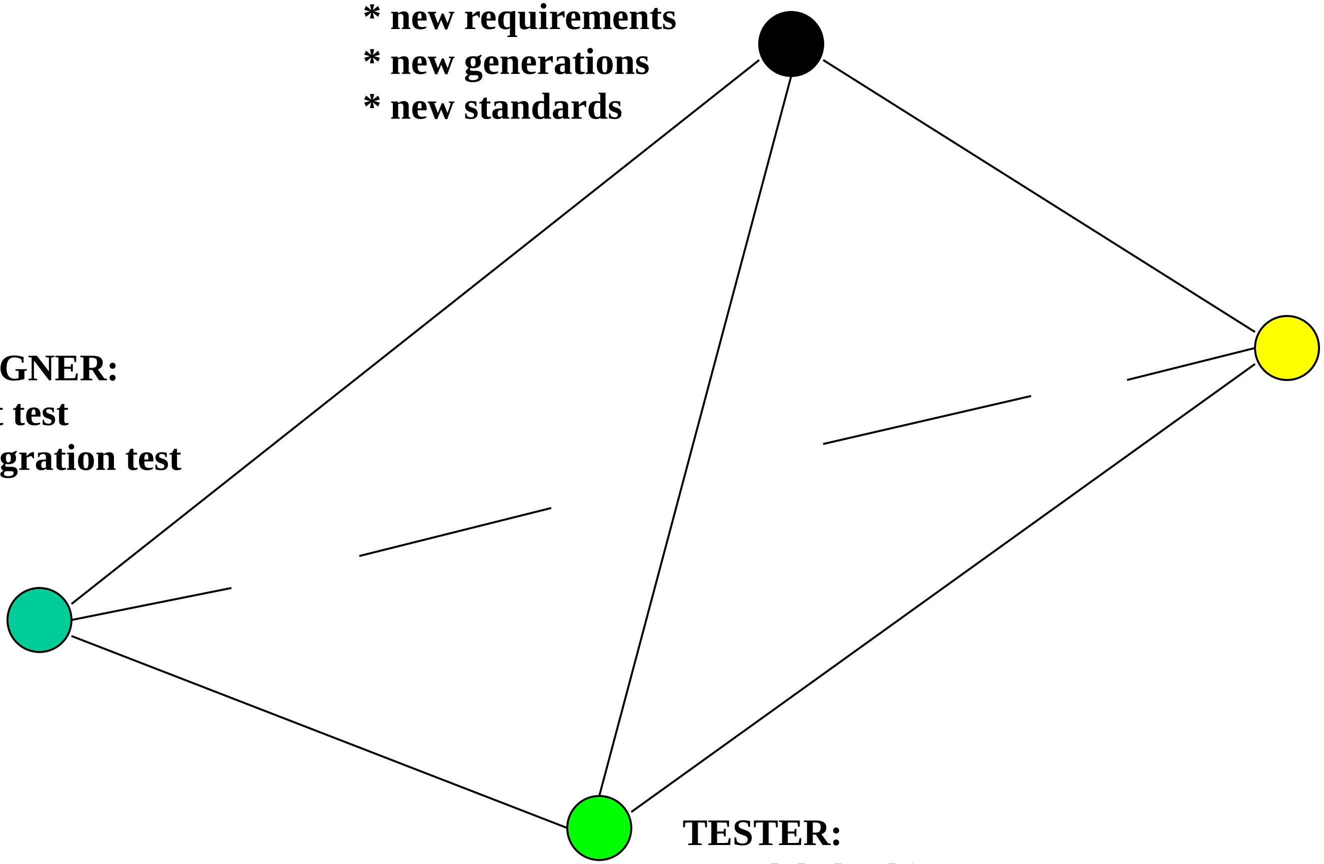
2.Design of a test cases set which covers the reduced set

Additional "dimension"

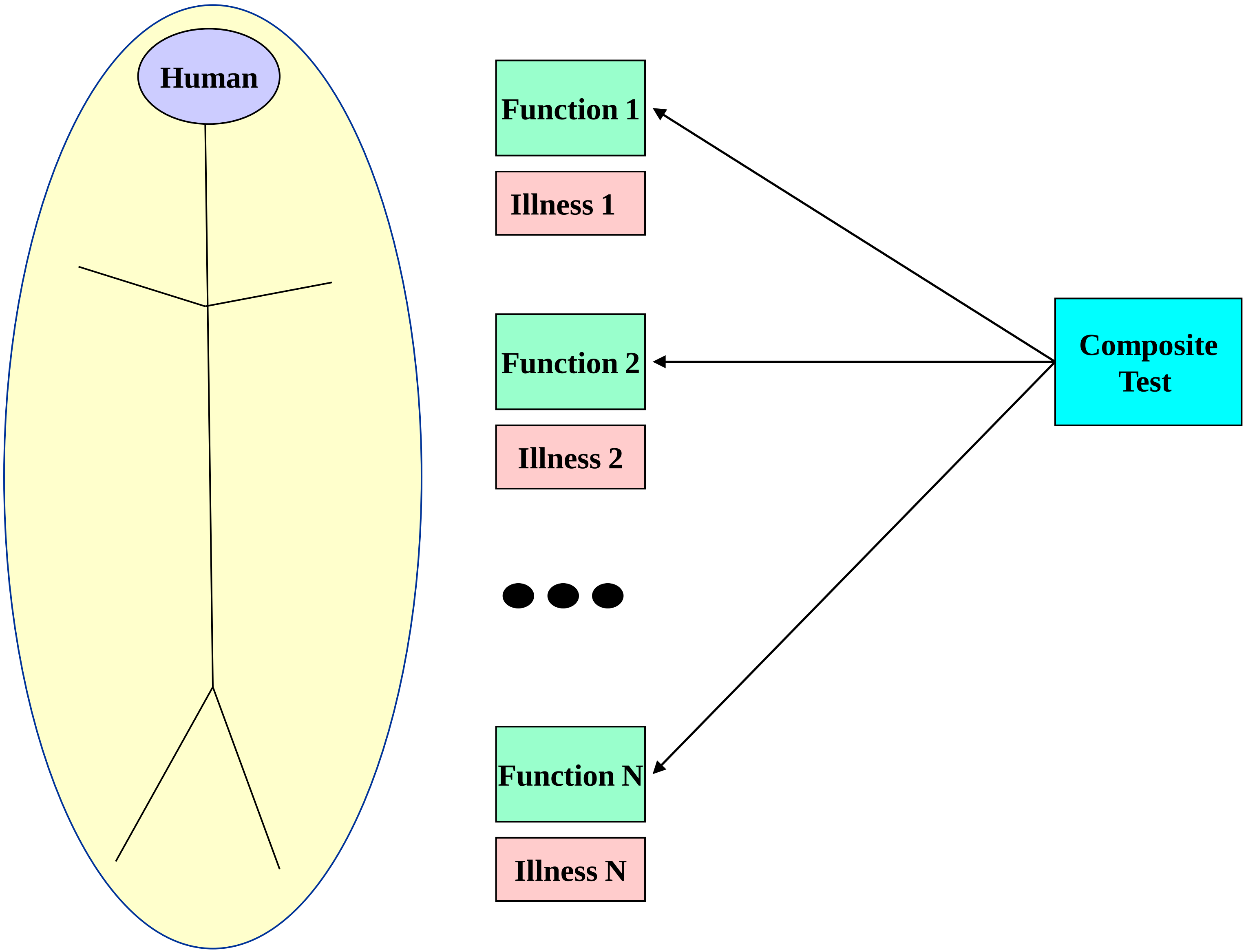


PLUS: Dynamics as system development / evolution

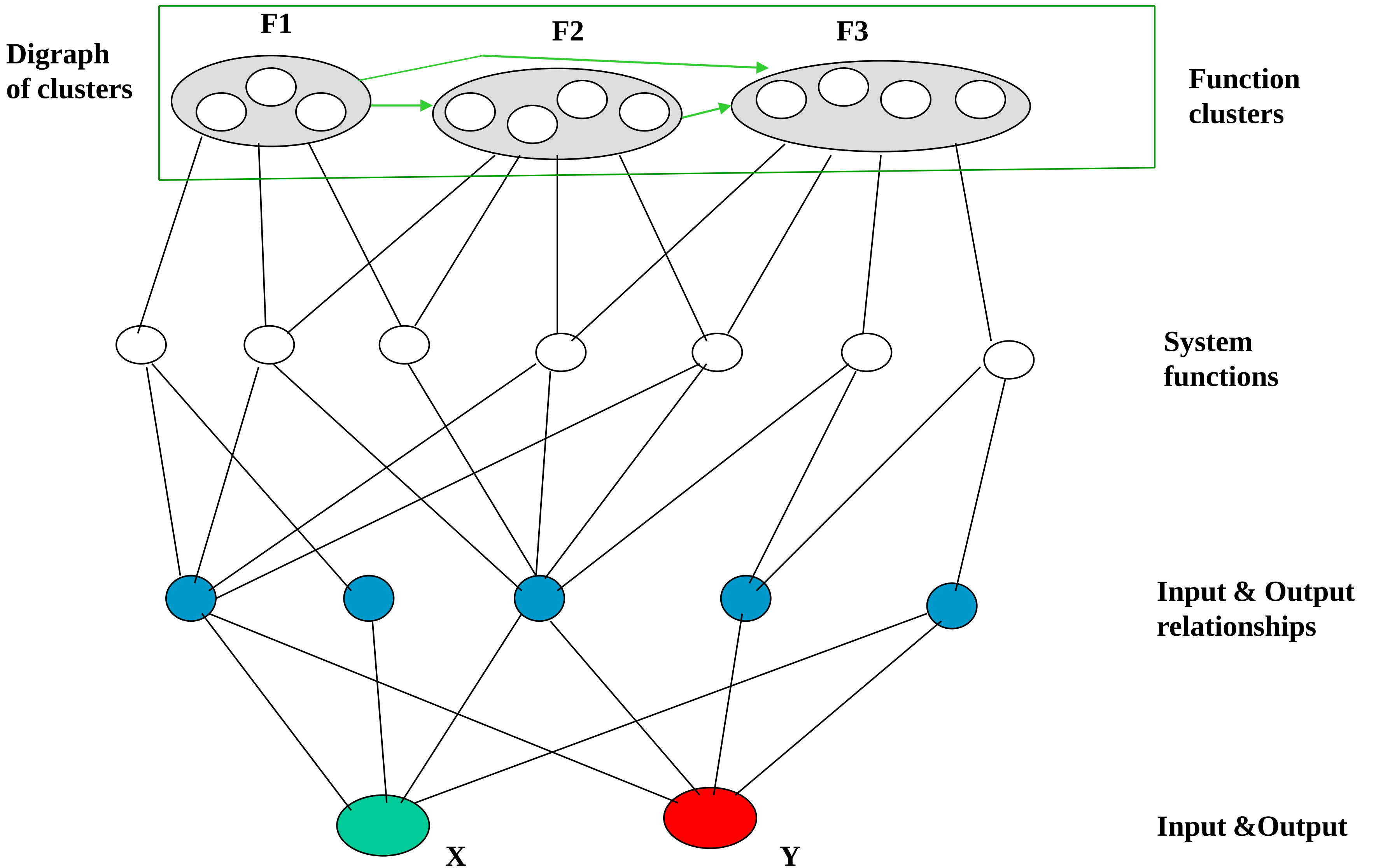
1.Preliminaries: Main Roles and Responsibility (system testing)



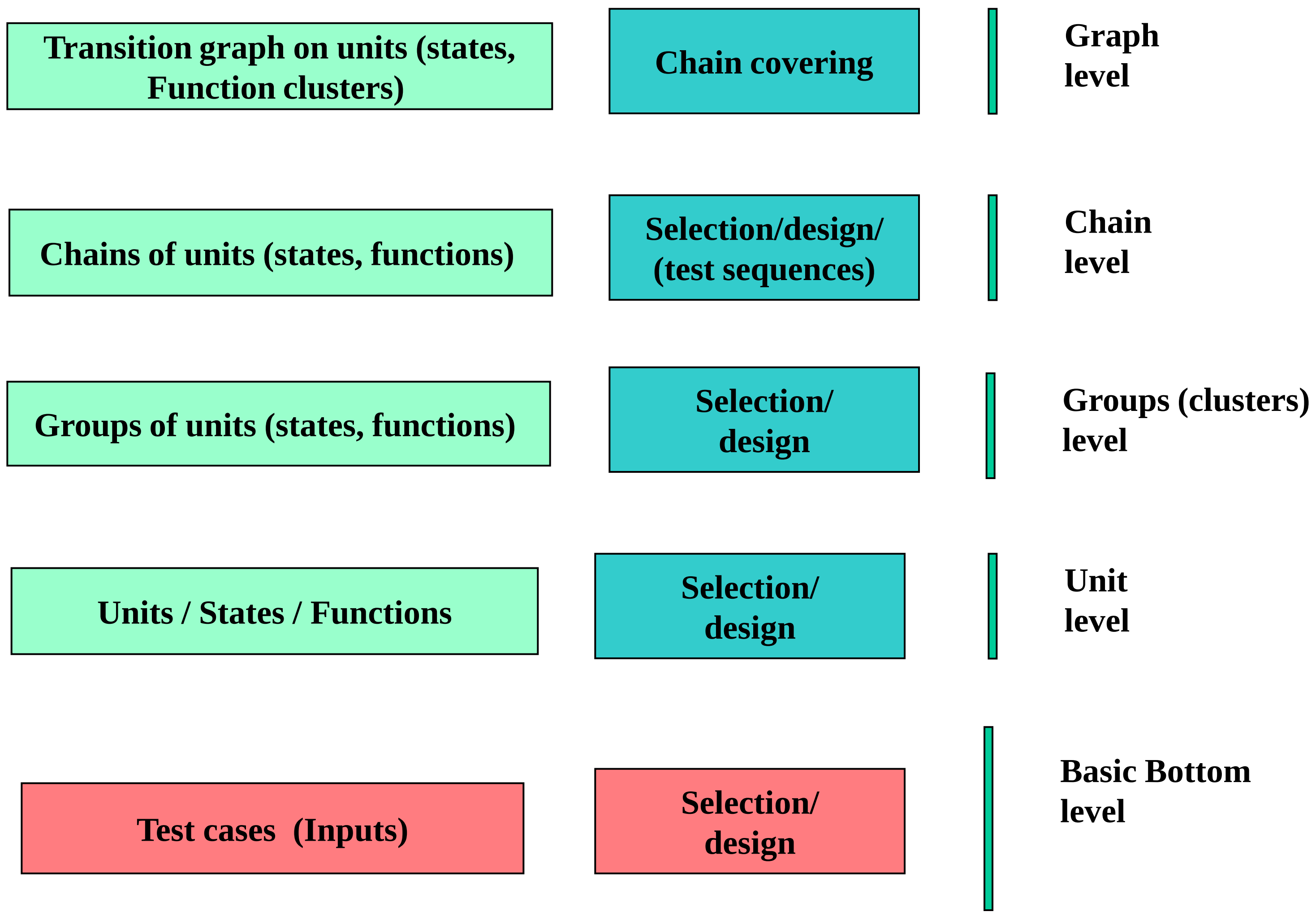
Human-based illustration for multi-function testing: N functions & composite test



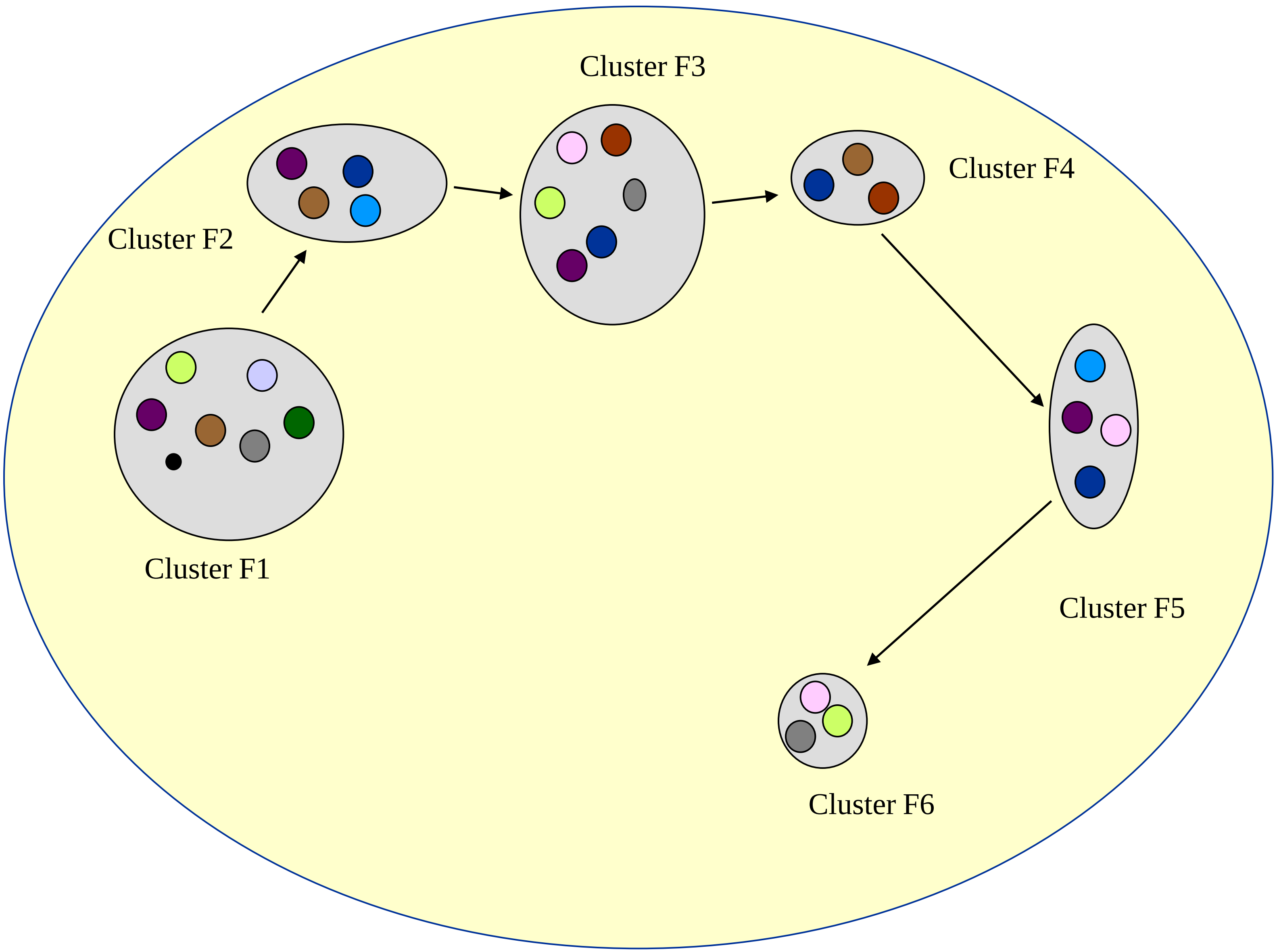
Hierarchy of system characteristics in multi-function system testing 4



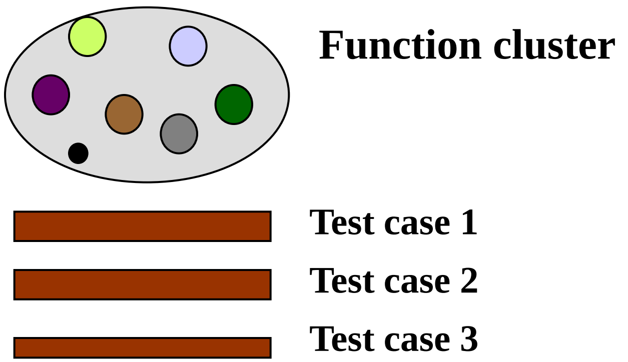
Levels of testing process and problems

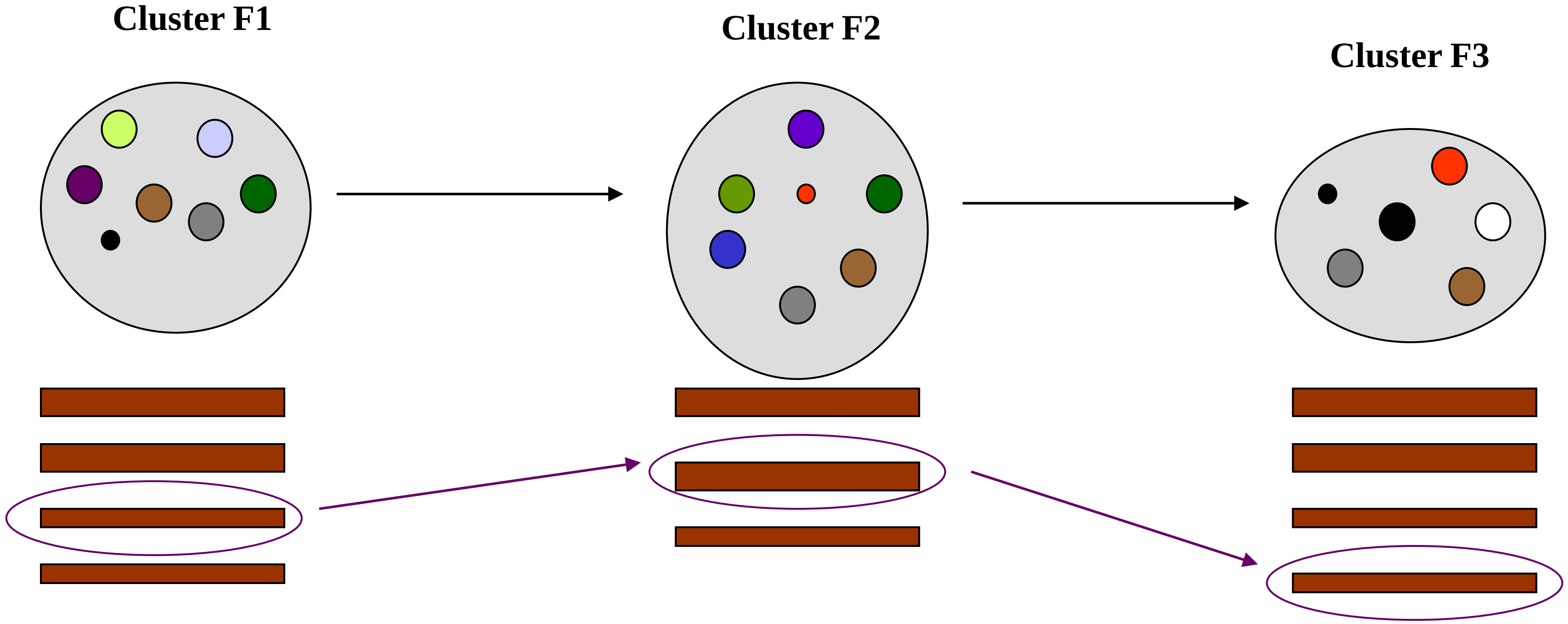


Space of system functions and function clusters



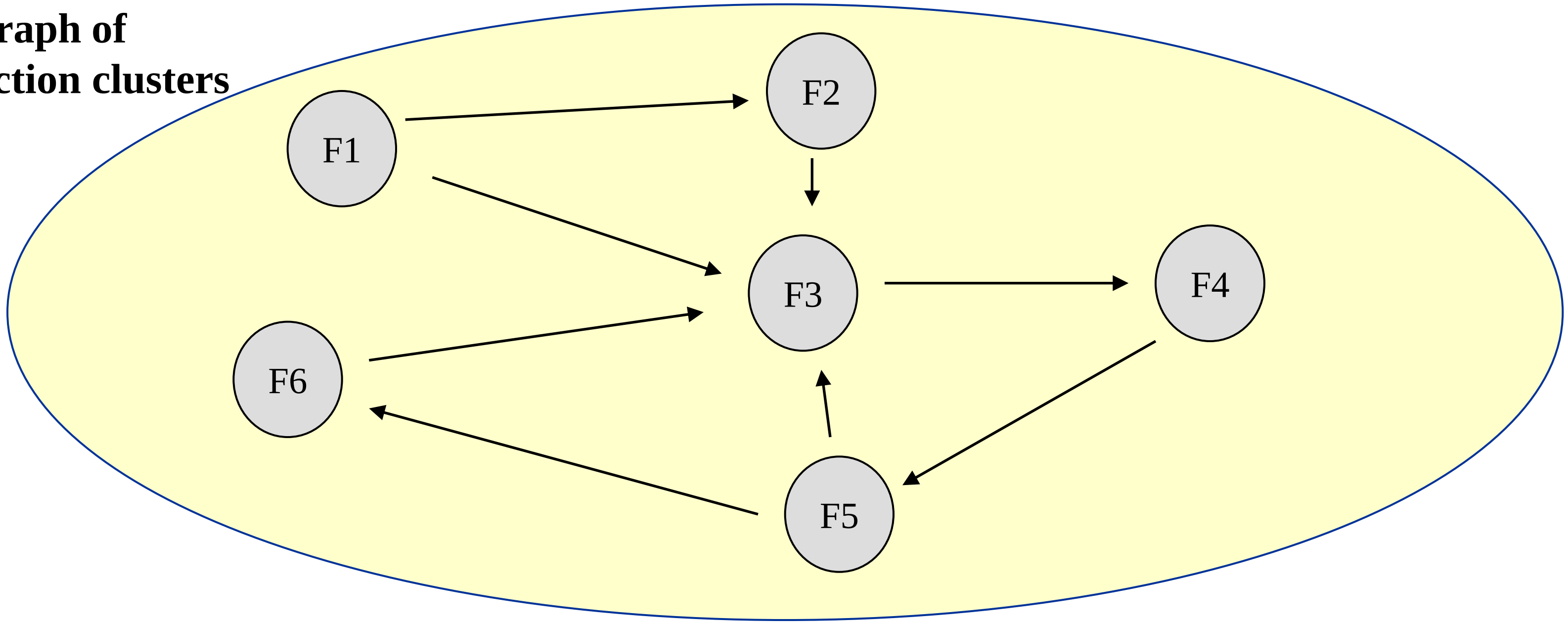
Test case for function cluster and a sequence of test cases for a chain of function clusters

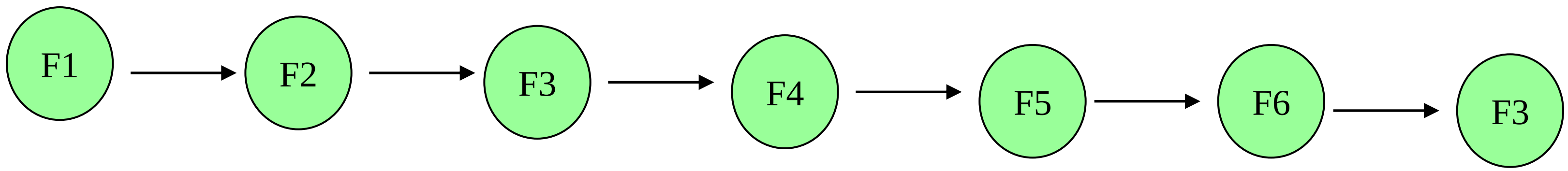


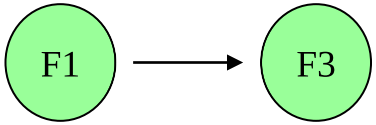


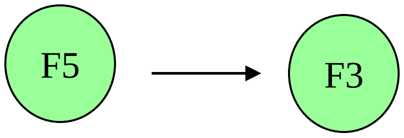
Chains of function clusters and covering

Digraph of function clusters

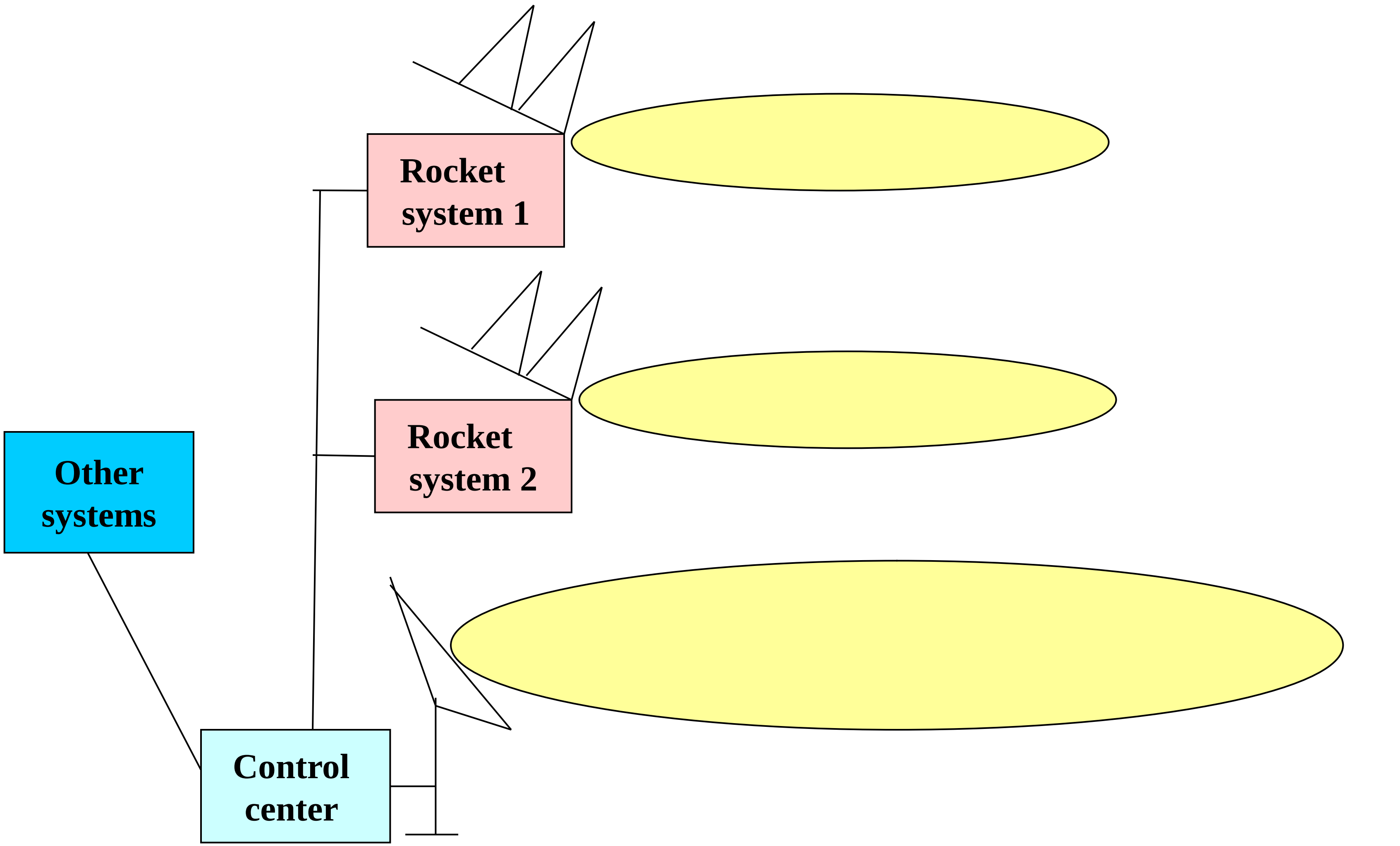








Applied missile defense / anti-aircraft system



Functions and function clusters

Functions:

1. **Scanning the examined area f1**
2. **Initialization of targets f2**
3. **Identification of targets f3**
4. **Tracking/maintenance of targets f4**
5. **Multi-target multi-track assignment f5**
6. **Fair control (assignment of rockets into targets) f6**
7. **Deletion of non-dangerous targets f7**
8. **Receiving date from other systems f8**
9. **Sending data to other systems f9**

Function clusters

F1: f1

F2: f1,f4

F3: f2,f3, f4

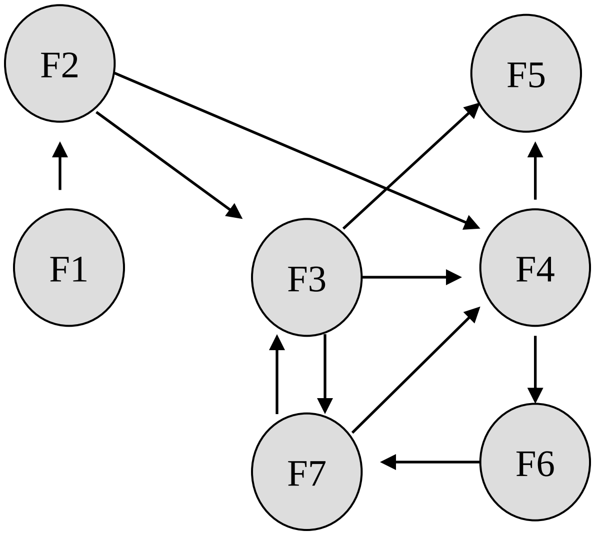
F4: f4,f5

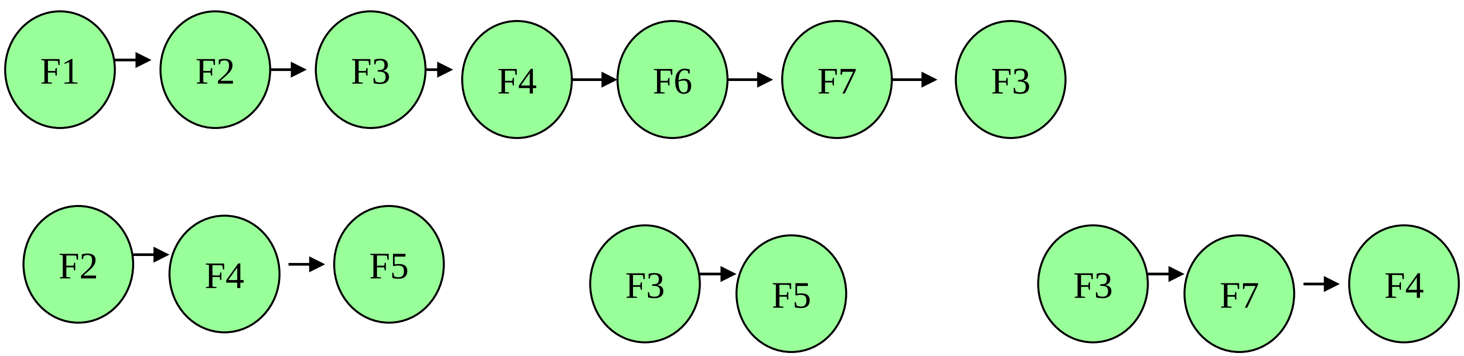
F5: f4,f7

F6: f5,f6

F7: f5,f8, f9

Functions cluster digraph





Recent papers on multi-function system testing

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.**

LECTURE 27. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Institute of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.27. System diagnosis, evaluation, improvement.

PLAN:

1.Hierarchical approach to diagnosis of complex systems

2.Hierarchical evaluation of composable system: example for building:

*models of building and corresponding evaluation scales for building parts

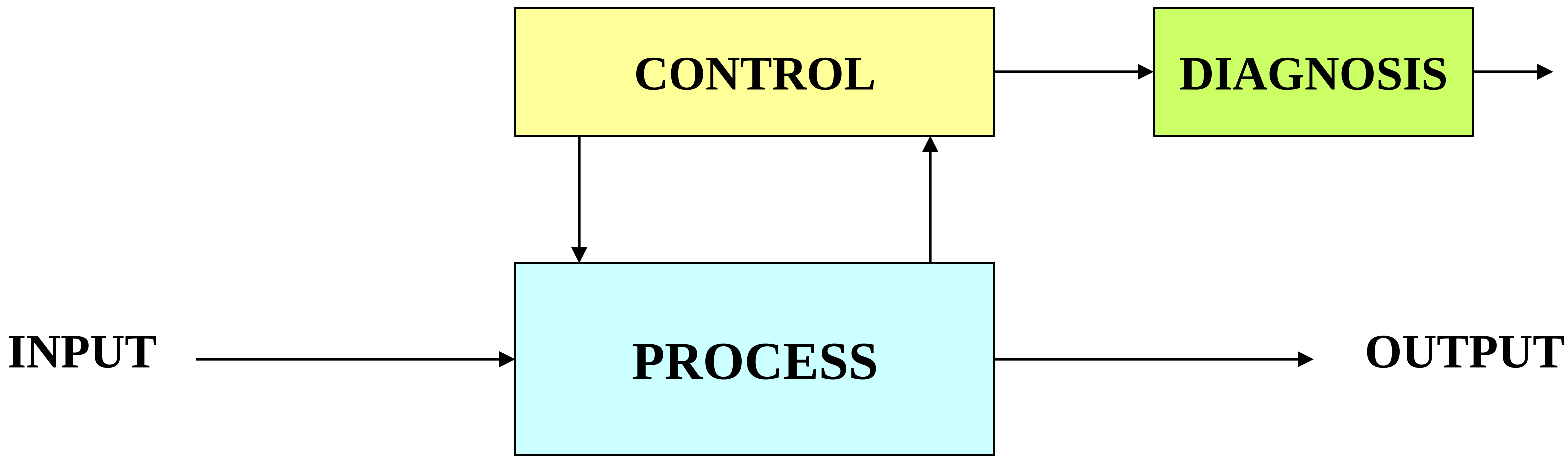
*method of integration tables

*usage of hierarchical combinatorial synthesis

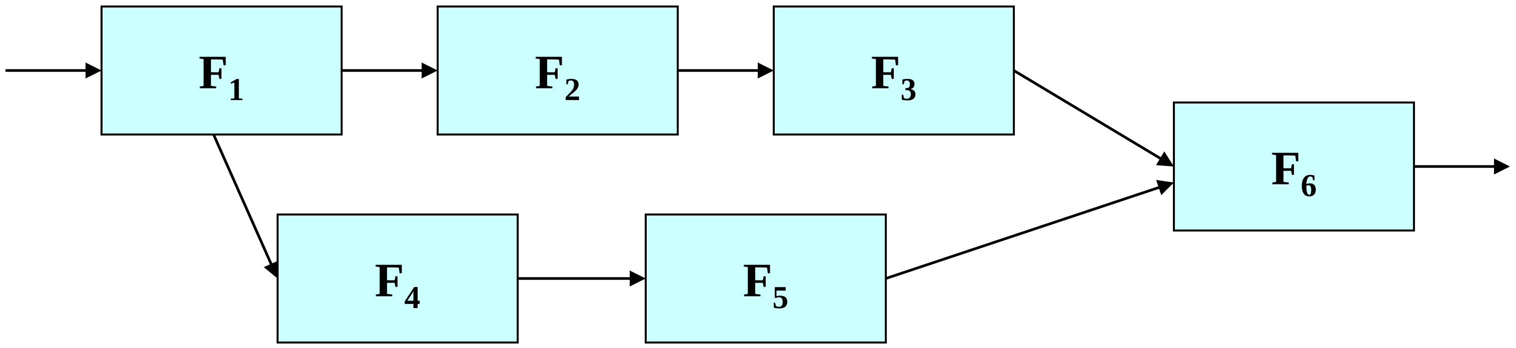
*change operations and planning an upgrade process

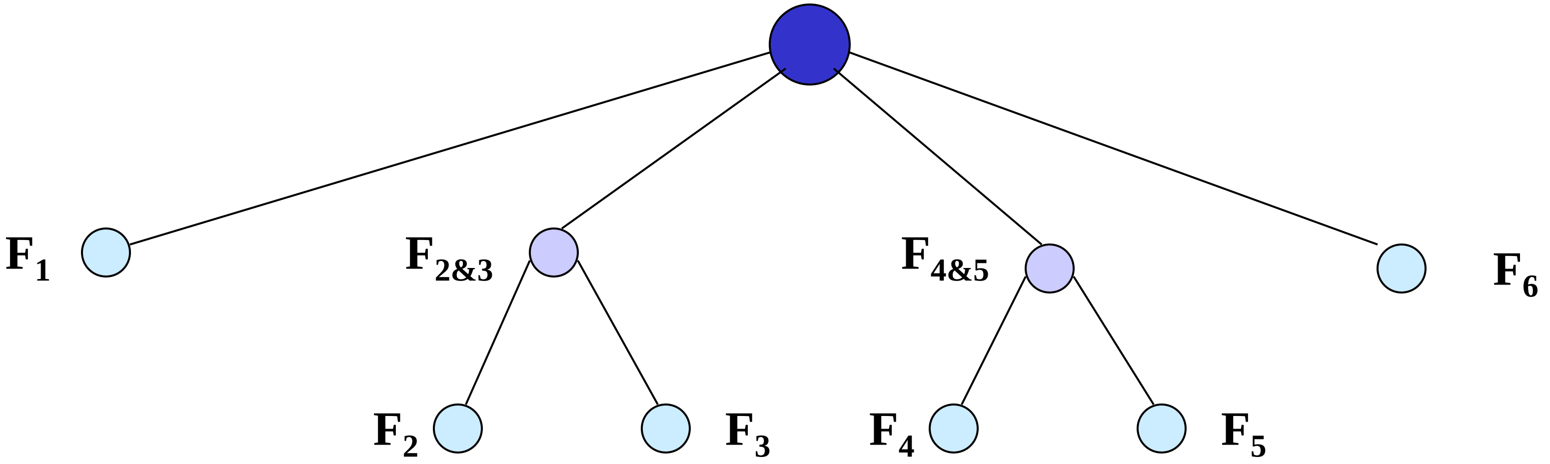
Nov. 12, 2004

Multi-level diagnosis of complex systems



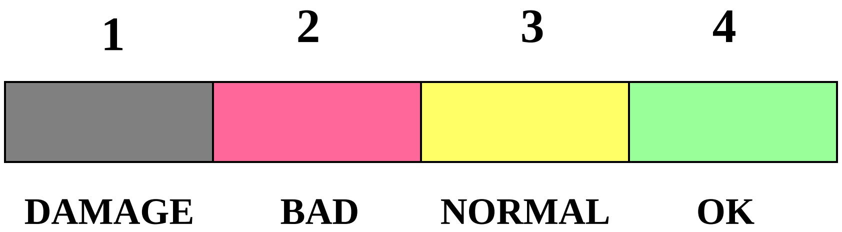
PROCESS

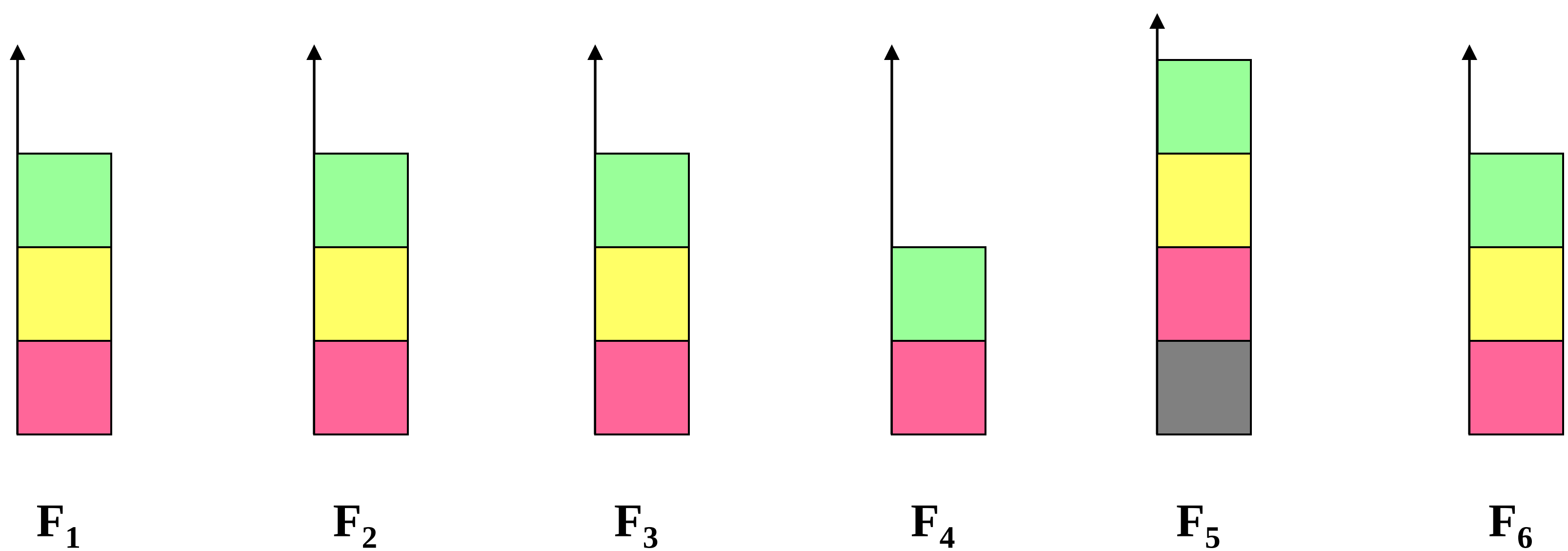




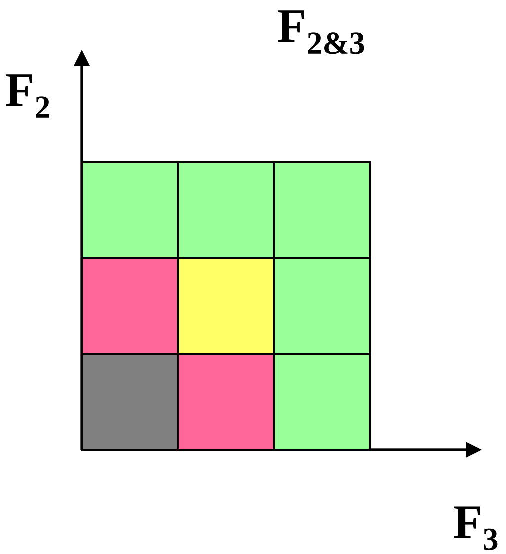
Multi-level diagnosis of complex systems

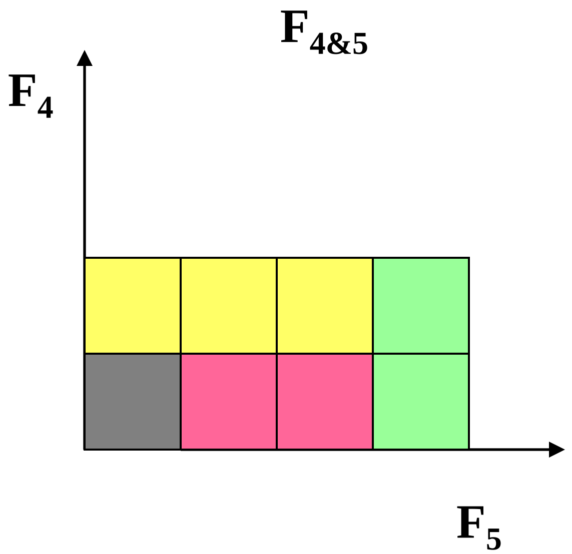
SCALE





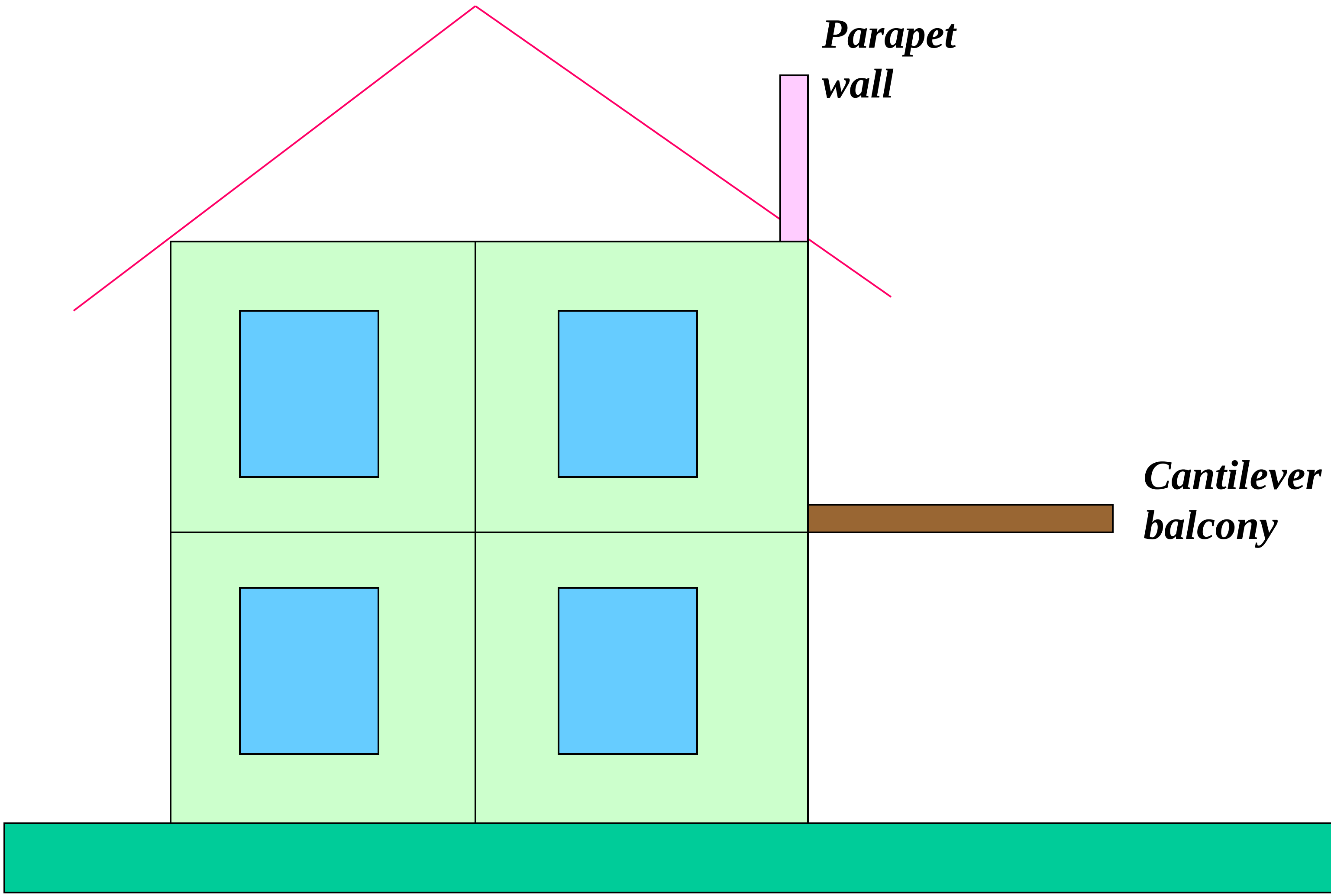
Multi-level diagnosis of complex systems





F_1 and $F_{2\&3}$ and $F_{4\&5}$ and F_6 ⟶ TOTAL ESTIMATE

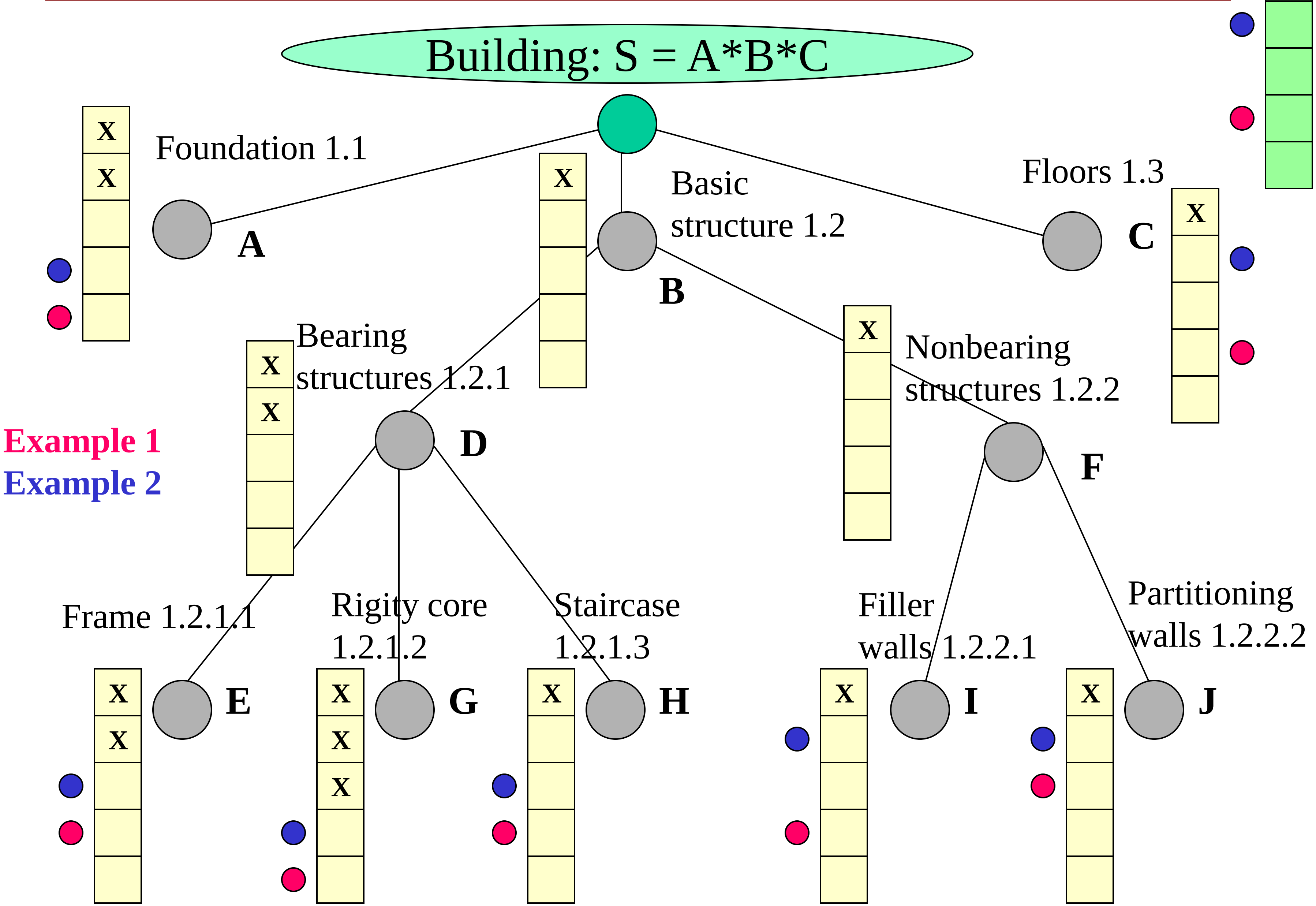
Example of building (evaluation from the viewpoint of earthquake engineering)



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



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

Method 1: integration tables

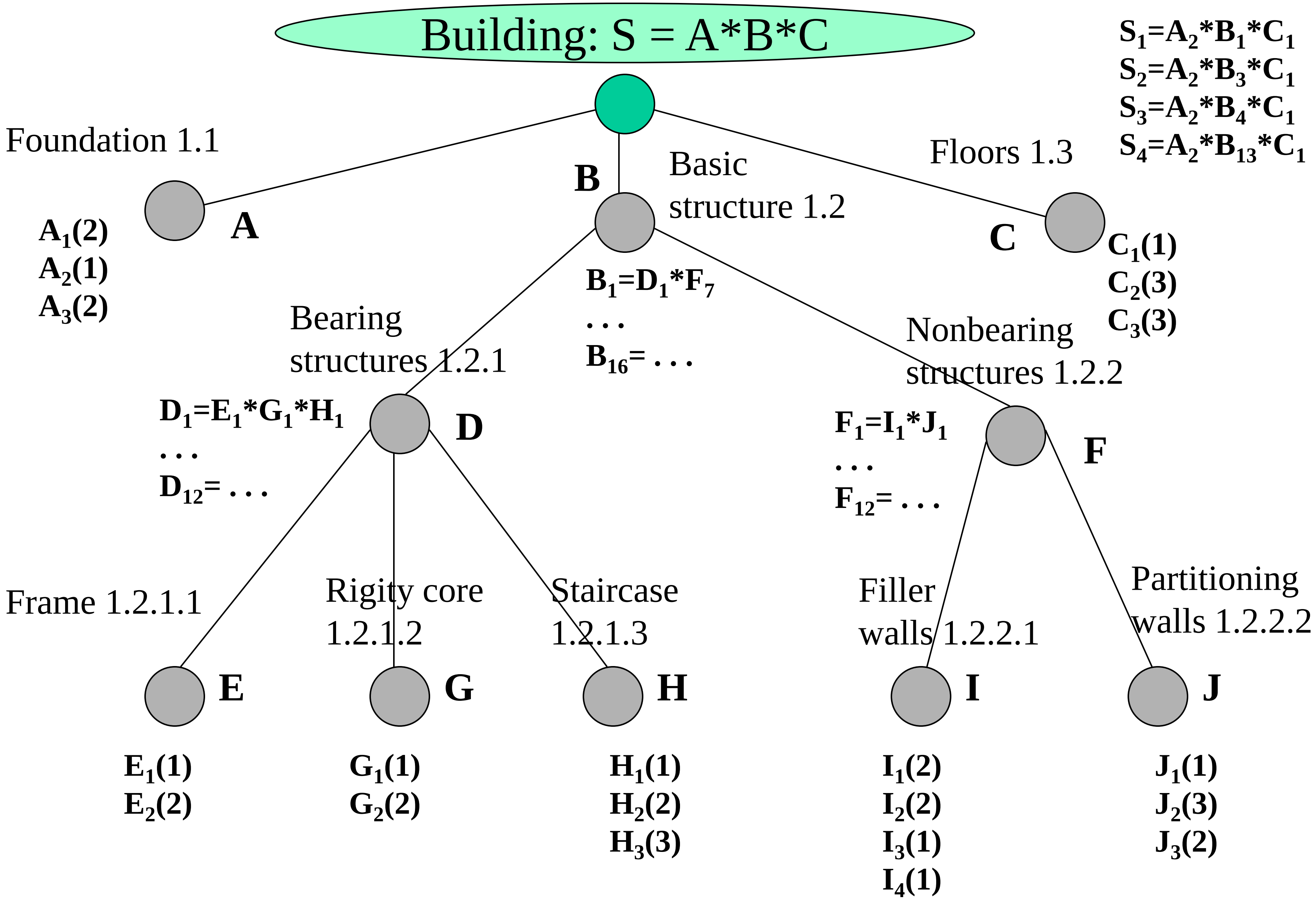
Building S, scale [2,3,4,5]

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

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

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

Method 2: Hierarchical morphological design (combinatorial synthesis)



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)

Compatibility

	G_1	G_2	H_1	H_2	H_3
E_1	3	2	3	1	2
E_2	2	1	2	1	2
G_1			3	2	1
G_2			2	1	1

	I_1	I_2	I_3	I_4
J_1	1	2	3	3
J_2	1	1	1	1
J_3	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

Compatibility

	C_1	C_2	C_3	B_1	B_3	B_4	B_{13}
A_1	2	2	2	2	2	2	1
A_2	3	2	2	3	3	3	2
A_3	2	2	2	2	2	2	1
C_1				3	3	3	2
C_2				3	3	3	2
C_3				2	2	2	3

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

Examples for building :

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

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

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

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

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

Improvement (upgrade) of building

Operation group I (frames):

O_1 increasing a geometrical dimension and active reinforcement

O_2 increasing of active reinforcement

Operation group II (joints):

O_3 increasing a level for fixing a longitudinal active reinforcement in zone of joints

O_4 decreasing the step of reinforced cross rods in zone of joint

Operation group III (cantilever and cantilever balcony):

O_5 decreasing the projection cantilever

O_6 supplementary supporting the cantilever

Operation group IV (fronton and parapet wall):

O_7 fixing a bottom part

O_8 designing a 3D structure (special)

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

O_9 design of shear keys

O_{10} design of mesh reinforcement

O_{11} 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

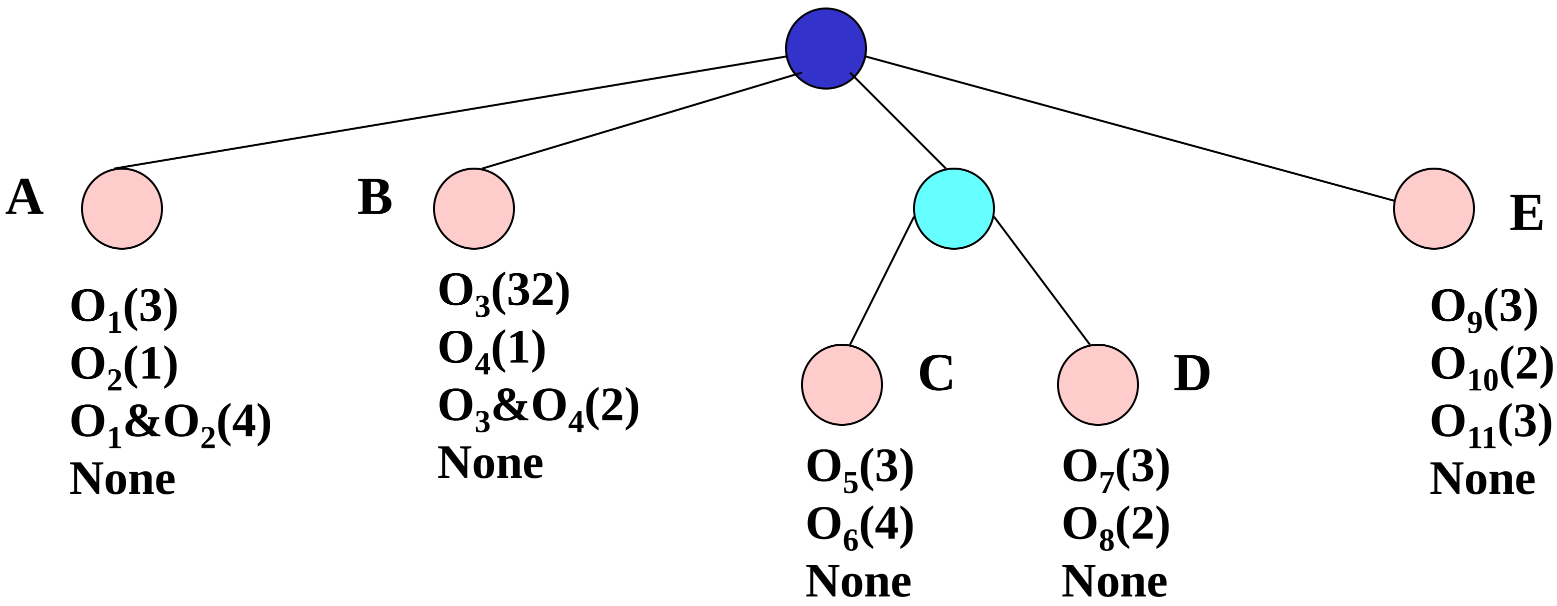
Model 4: Morphological clique problem

Model 5: Scheduling

ETC.

Combinatorial synthesis for planning of redesign (improvement, upgrade)

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



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

LECTURE 28. Course: “Design of Systems: Structural Approach”

Dept. “Communication Networks &Systems”, Faculty of Radioengineering & Cybernetics

Moscow Institute of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.28. System maintenance

PLAN:

1.Preliminaries: life cycle, systems, utilization, personnel, maintenance, roles

2.Framework of maintenance process:

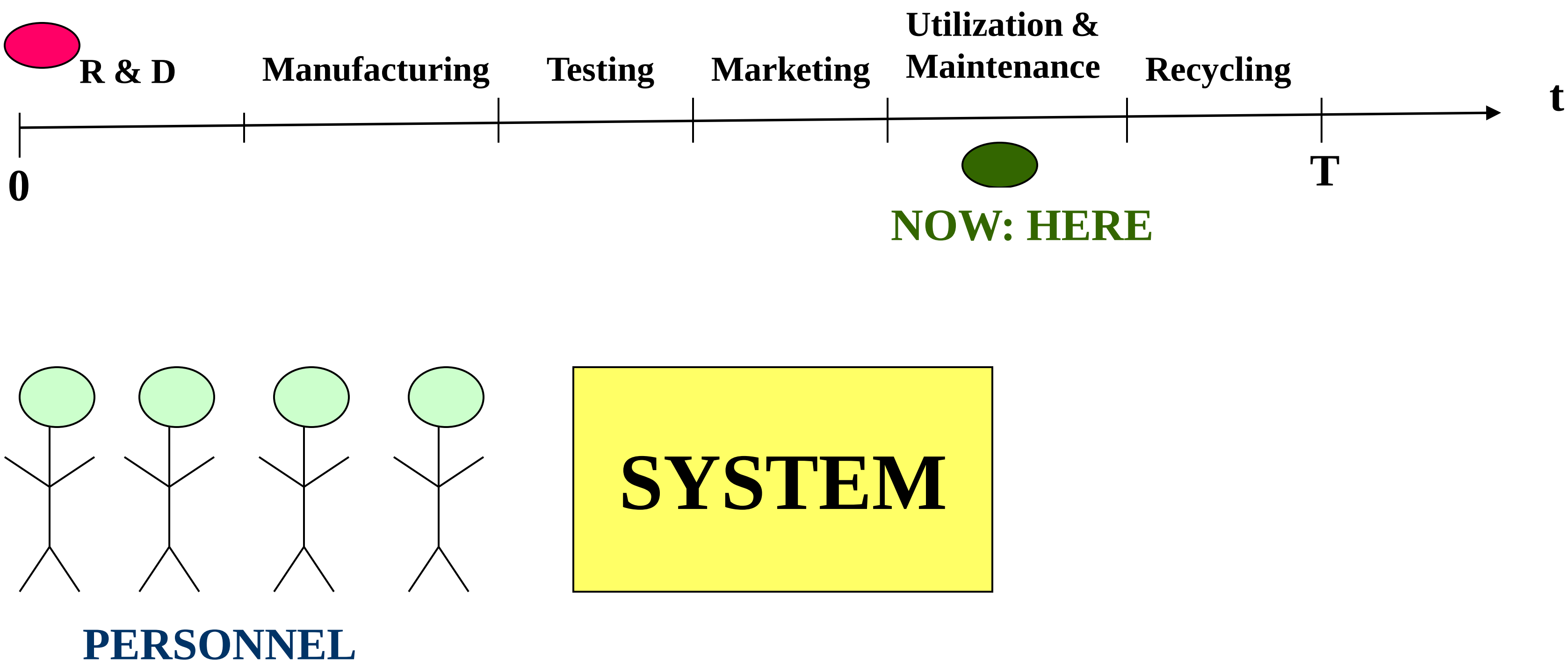
*basic framework, *systems under maintenance

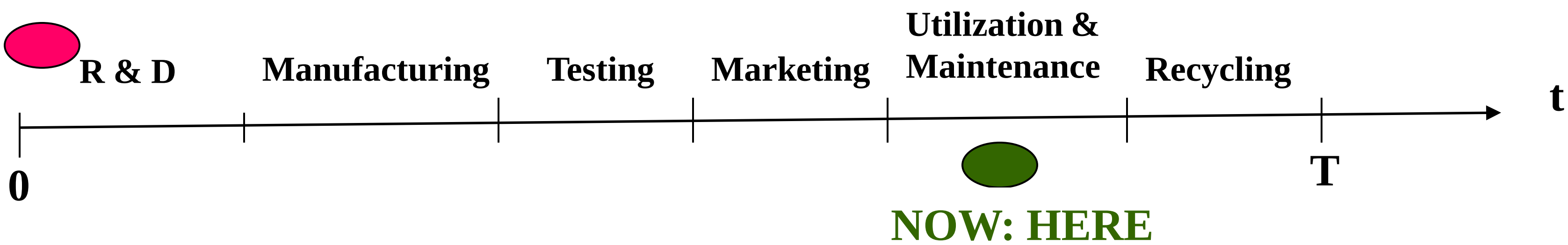
*maintenance operations (inspection as testing/analysis/diagnosis, repair, *replacement) *etc.

3.Illustrations: basic analogue (monitoring system), trajectory of fault.

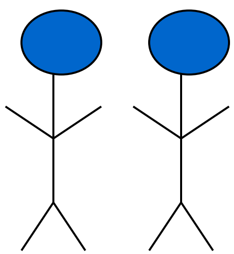
Nov. 19, 2004

1.Preliminaries

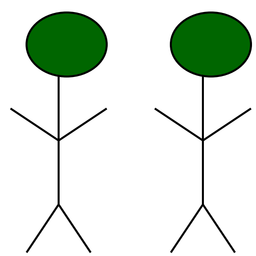




UTILIZATION
(e.g., driver, pilot, operator)



SYSTEM



MAINTENANCE
(support of system & personnel)

MAINTENANCE ACTIONS:

***system installation**

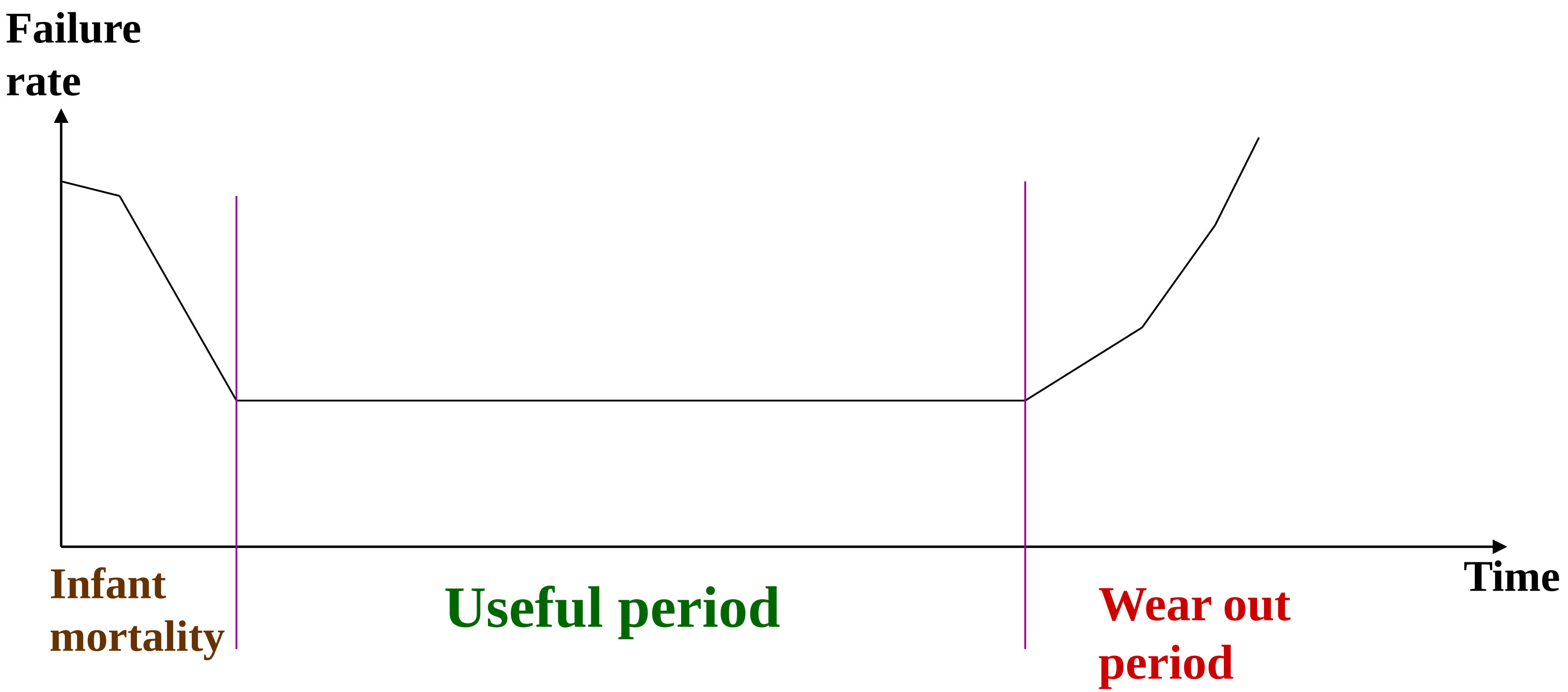
***training of personnel**
***regular testing of personnel**

***planning / scheduling of maintenance operations**
***execution of operations (inspection, repair, replacement)**
***analysis of the system, accumulation of data, prediction**

EXAMPLES:

SYSTEM	UTILIZATION	MAINTENANCE
1.Car	Driver, passengers	Maintenance personnel
2.Airlane	Pilot, passengers	Maintenance personnel
3.Human	Human	Doctor, human
4.House	Owner	Maintenance personnel, owner
5.Computer	User(s)	Maintenance personnel, special software, user(s)

TYPICAL FAILURE RATE CURVE:



AN ORDINAL SCALE FOR FAULTS:

1.OK

2.Small fault

3.Significant fault

4.Damage (destriction)

OBJECT UNDER MAINTENANCE:

1.System and / or system part (component, unit)

2.System state

3.System function or function cluster
(as a group of interrelated functions)

MAINTENANCE FRAMEWORK (problems):

1.System analysis / evaluation (systems, its part, etc.)
2.System prediction
3.Operational management / preventive maintenance:
  ***testing**
  ***evaluation**
  ***additional information**
  ***repair / replacement**
4.Design of information model for system and each part (object)
5.System strategy:
  ***selection of object under maintenance**
  ***selection of operation (i.e., inspection, repair, replacement)**
  ***assignment of time for operation(s)**
  ***execution of the operation(s)**

2.Framework

Planning

SYSTEM

Models for system, faults, prediction

Maintenance operations (scanning, repair, replacement)

Analysis, accumulation, processing, integration (fusion), distribution of information

MODELS:
***selection, *knapsack, *routing, *assignment / allocation, etc.**
***probabilistic models, Markov processes**
***reliability evaluation / analysis**
***safety analysis**
***simulation**
***etc.**

SYSTEMS UNDER MAINTENANCE:

1.Whole system
2.Multi-component (modular) system:
***one-layer modular system**
***hierarchical (multi-layer) system**
***multi-layer modular system with complex module interrelation (including interrelation between different layers and branches)**
3.Developping systems (e.g., upgrade of components, structure, interconnection)
4.Change of external environment

KINDS OF MAINTENANCE:

1.BASIC MAINTENANCE:
faults => operation

2.PREVENTIVE MAINTENANCE:
prediction of faults => preliminary maintenance operation

3.SELF-MAINTENANCE

SYSTEM CHARACTERISTICS:

1.Reliability (stability, etc.)

2.Safety

3.Viability, survivability

4.Robustness

4.Performance

PROBLEMS & MODELS
OF PREVENTIVE MAINTENANCE:

1.REVELATION (ALLOCATION) OF TEST POINTS
Models: multicriteria selection, knapsack-like problems, allocation problems, etc.

2.PLANNING TEST OPERATIONS
Models: multicriteria selection, knapsack-like problems, scheduling, etc.

MODELING & INFORMATION PROCESSING:

1.Modeling of faults

2.Diagnosis of faults

3.Monitoring of faults

4.Tracking of faults

5.Integration (fusion) of information on local faults

6.Distribution of information on faults for various information systems and specialists

SYSTEM

BASIC ACTIONS

1.Scanning

2.Small repair / replacement

3.Essencial repair / replacement

SYSTEM LAYERS

1.System
2.System parts
(group of states,
group of functions)
3.Components
(state, function)

3.Basic analogue: Monitoring systems

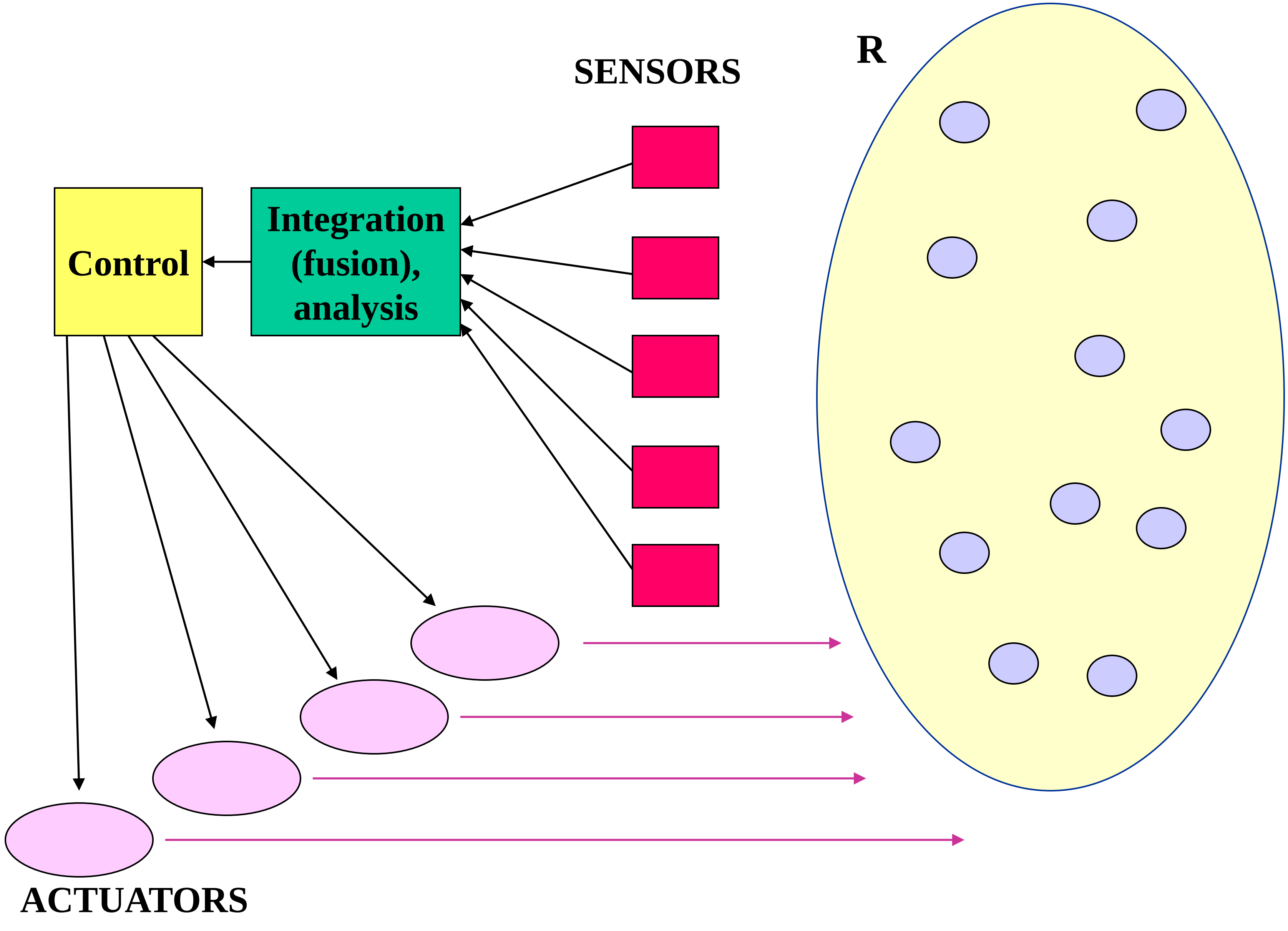
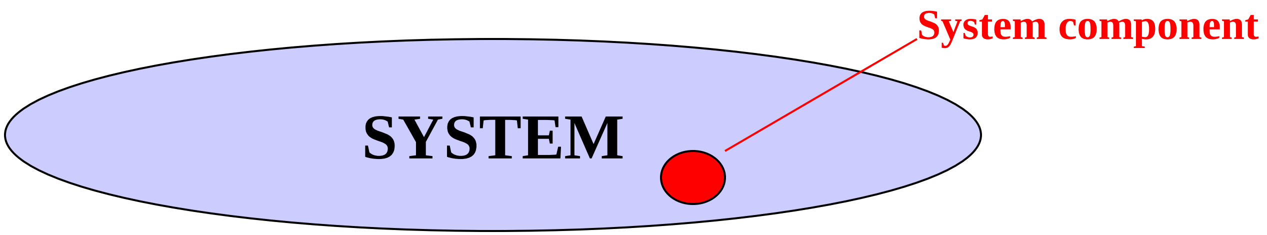
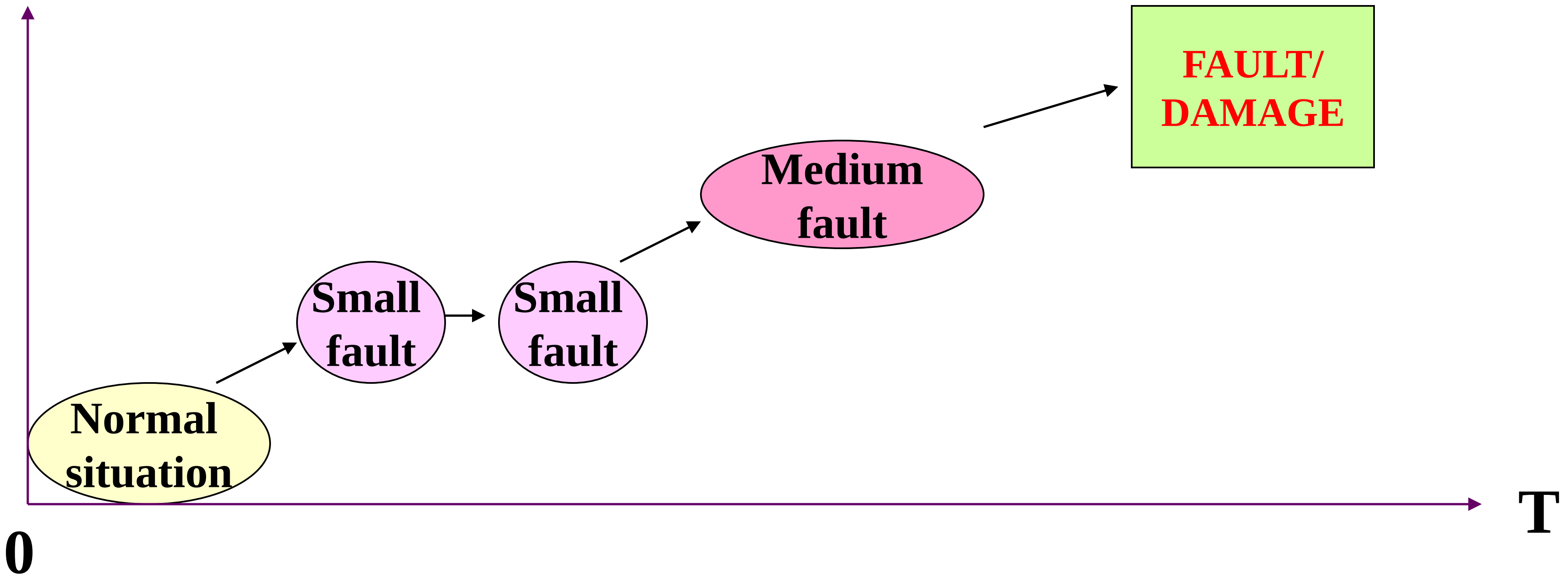


Illustration: Accumulation of information



Trajectory of faults (for components)



LECTURE 29. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.29. Requirements engineering

PLAN:

1.Requirements engineering: preliminaries

2.Types of requirements

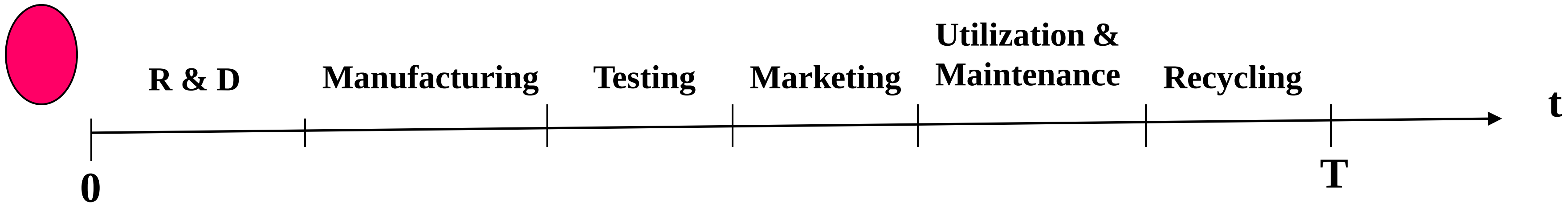
2.Additions

3.Systems under analysis

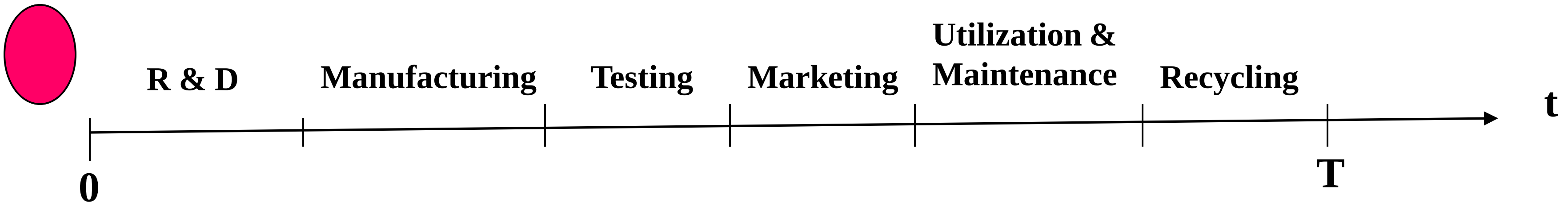
4.Models

Nov. 26, 2004

1.Preliminaries

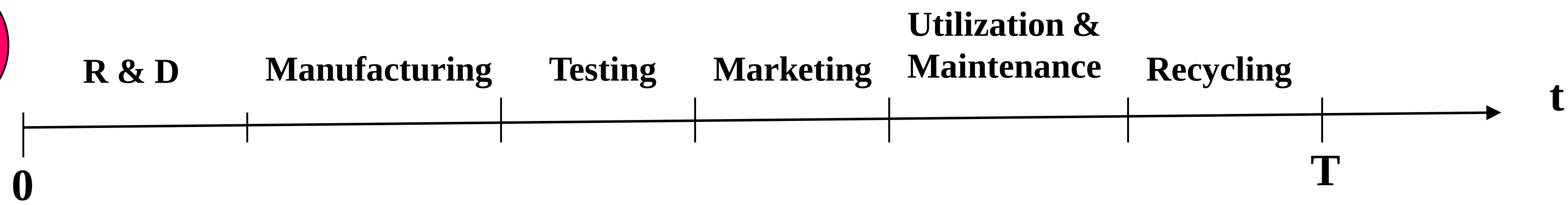


1.Preliminaries



OLD RUSSIAN ENGINEEIRNG EXPERIENCE: NB!!!

1.Preliminaries

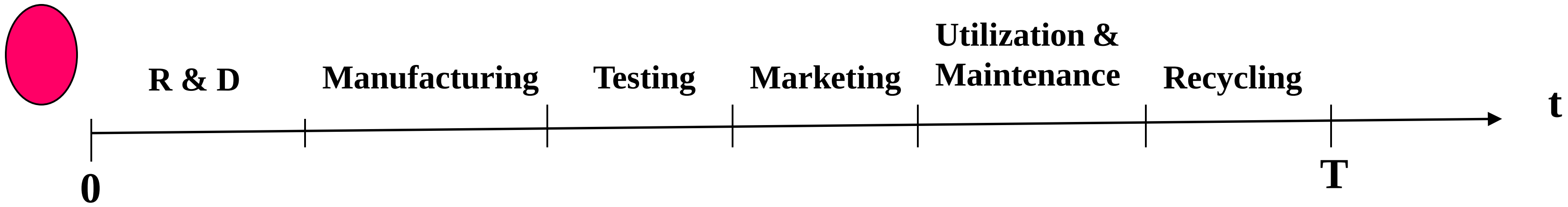


OLD RUSSIAN ENGINEEIRNG EXPERIENCE: NB!!!

WEST EXPERIENCE:

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. Wiegers, Software Requirements. 2nd ed., Microsoft Press 2003.**

1.Preliminaries



OLD RUSSIAN ENGINEEIRNG EXPERIENCE: NB!!!

WEST EXPERIENCE:

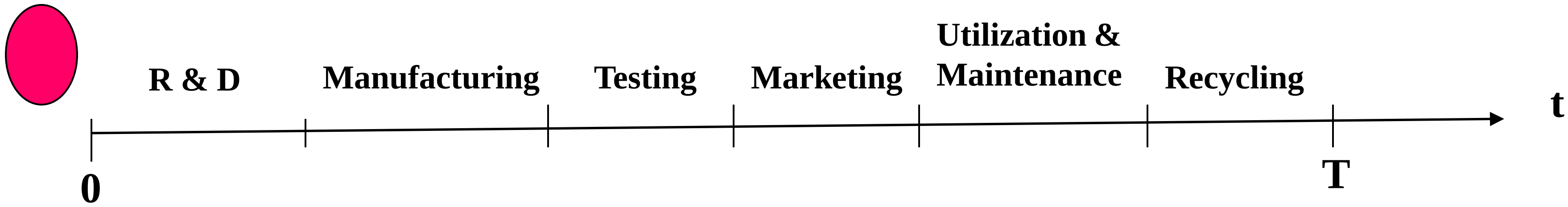
1.Ralph R. Young, The Requirements Engineering Handbook, Artech House, 2004 (Carnegie Mellon Univ.)

2.S. Robertson, J. Robertson, Mastering the Requirements Process. Addison-Wesley, 1999.

3.K.E. Wiegers, Software Requirements. 2nd ed., Microsoft Press 2003.

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

1.Preliminaries



OLD RUSSIAN ENGINEEIRNG EXPERIENCE: NB!!!

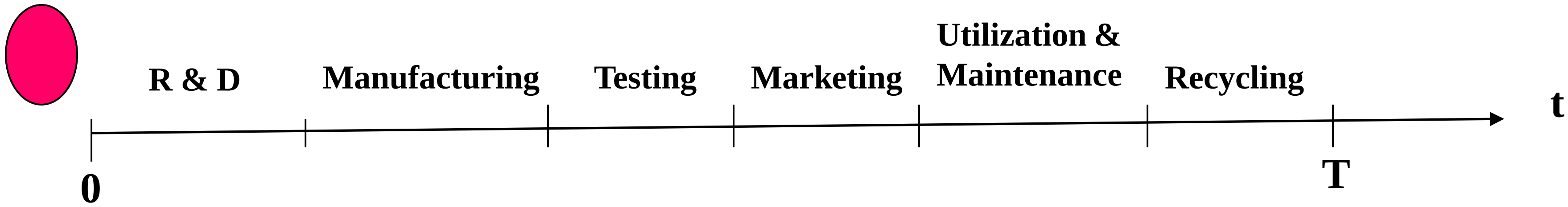
WEST EXPERIENCE:

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. Wiegers, Software Requirements. 2nd ed., Microsoft Press 2003.**

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

CONFERENCES:
IEEE Requirement Engineering Conference, etc.

1.Preliminaries: Layers (product, requirements, standards)

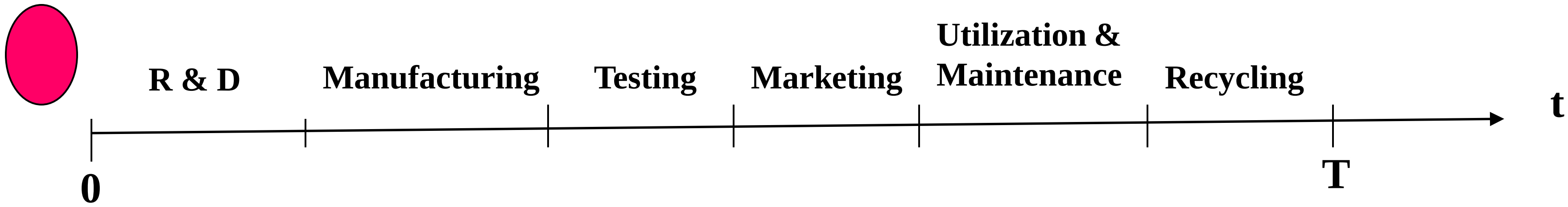


1.Standards

2.Requirements

3.System (product, product family, platform)

1.Preliminaries



Neighbor disciplines:

1.Systems engineering (& life cycle engineering)

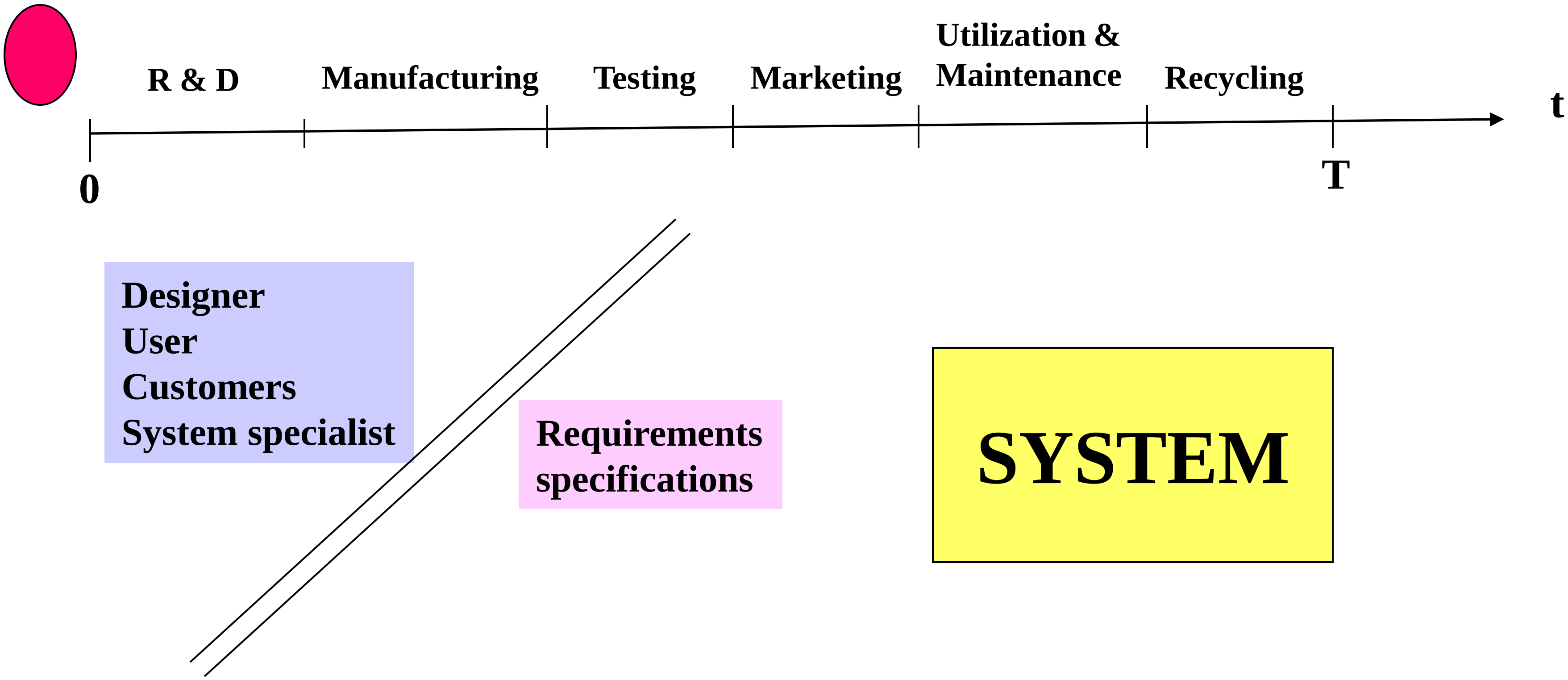
2.Strategic management

3.Marketing

4.Forecasting

5.Knowledge engineering (acquisition of experience)

1.Preliminaries



1.Preliminaries

R & D | Manufacturing | Testing | Marketing | Utilization & Maintenance | Recycling

0 ... T → t

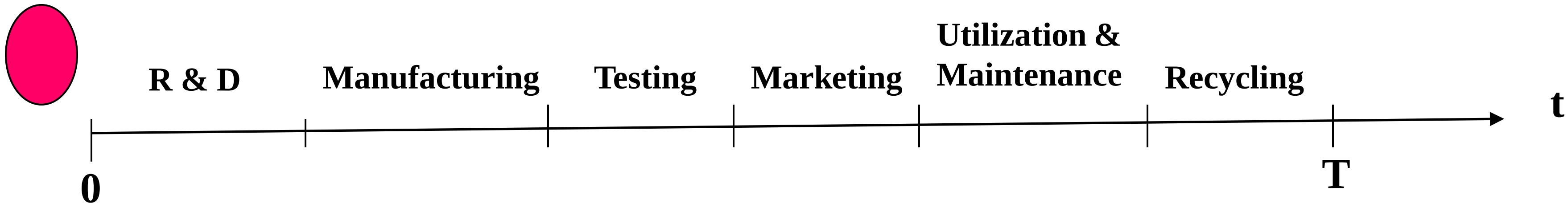
Designer
User
Customers
System specialist

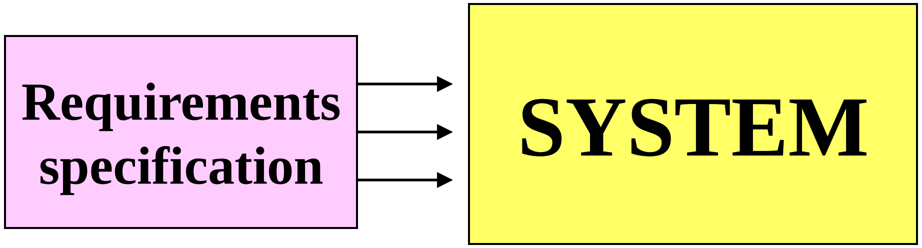
Requirements specifications

Personnel with skills (LITERACY)

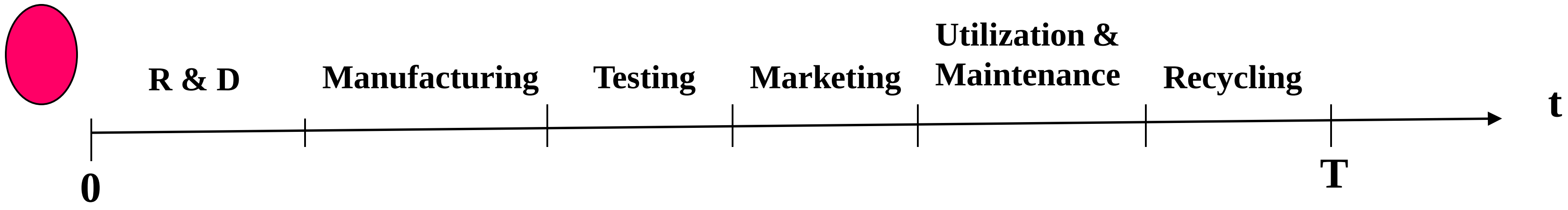
SYSTEM

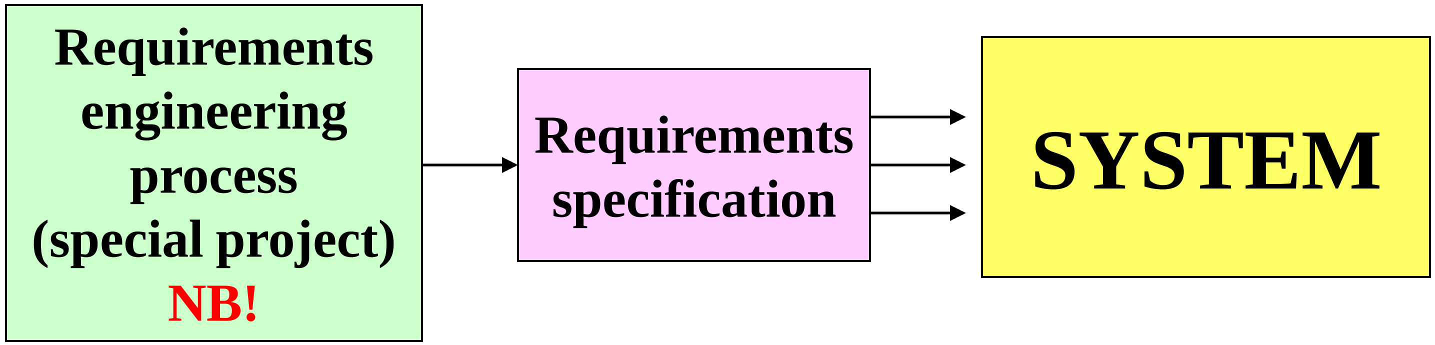
1.Preliminaries



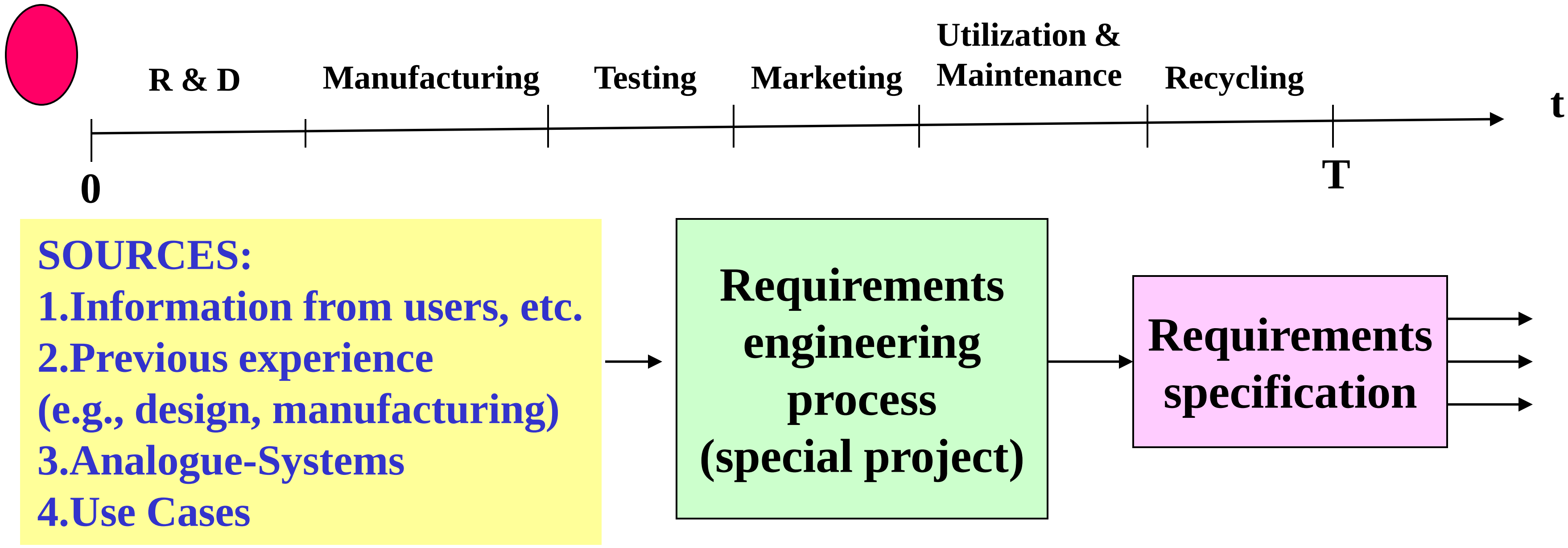


1.Preliminaries





1.Preliminaries



1.Preliminaries

R & D | **Manufacturing** | **Testing** | **Marketing** | **Utilization & Maintenance** | **Recycling**

0 — **T** — **t**

SOURCES:
1.Information from users
2.Previous experience
(e.g., design, manufacturing)
3.Analogue-Systems
4.Use cases

Requirements engineering process (special project)

Requirements specification

PROBLEMS & TOOLS (TECHNIQUES):
1.System analysis
2.Discovering
3.Acquisition of knowledge, skills, experience
4.Structuring & Integration
5.Modeling & Representation
6.Analysis of dynamics
6.Testing
7.Forecasting

2.Types of requirements (a hierarchical system of requirements)

TYPES:

1.Business requirements
2.User's requirements
3.High-level or system requirements
4.Functional requirements (things the system must do)
5.Non-functional requirements (properties the system must have)
6.Design requirements / design constraints
7.Manufacturing constraints
8.Performance requirements
9.Interface requirements (with other systems)
10.Qualification requirements
11.Logistics requirements
12.Environmental requirements
13.System, subsystem and component requirements
14.Reusing requirements
ETC.

ADDITIONS:

1.Criteria for evaluation of requirements

2.Prototyping

3.Scenarios

4.Reusing requirements

OBJECT & HIERARCHY

1.System and / or system part (component, unit)

2.System state, group of states, state chart

3.System function, function cluster, digraph of function clusters

OBJECT & HIERARCHY

1.System and / or system part (component, unit)

2.System state, group of states, state chart

3.System function, function cluster, digraph of function clusters

1.SYSTEM / PRODUCT

2.PRODUCT FAMILY

3.PLATFORM

SCENARIOS:

1.STRUCTURE (e.g., chain, tree) of system states, functions

2.Qualitative scenarios

3.Integration of use cases & forecasting

SCENARIOS:

1.STRUCTURE (e.g., chain, tree) of system states, functions

2.Qualitative scenarios

3.Integration of use cases & forecasting

MODELS:
1.Entity relationship
2.State transition model
3.Entity relationship & state transition diagrams

MODELS:

I.HIERARCHY OF REQUIREMENTS
1.Hierarchy of information
2.Integration of information (fusion), etc.

II.SCENARIOS
1.Coneptual maps
2.Graph models, etc.

III.DYNAMICAL MODELING
1.Simulation
2.Testing, etc.

LECTURE 30 (compressed version). Course: “Design of Systems: Structural Approach”

Dept. “Communication Networks &Systems”, Faculty of Radioengineering & Cybernetics

Moscow Inst. of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.30. Allocation problems.

PLAN:

1.Allocation problem (problem formulations as assignment, matching, location):

*assignment problem (marriage problem), *quadratic assignment problem (QAP),

*generalized assignment problem (GAP) , *string matching (illustration), *multiple matching (illustration)

2.List of basic algorithm approaches

3.Evolution chart of allocation-like problems

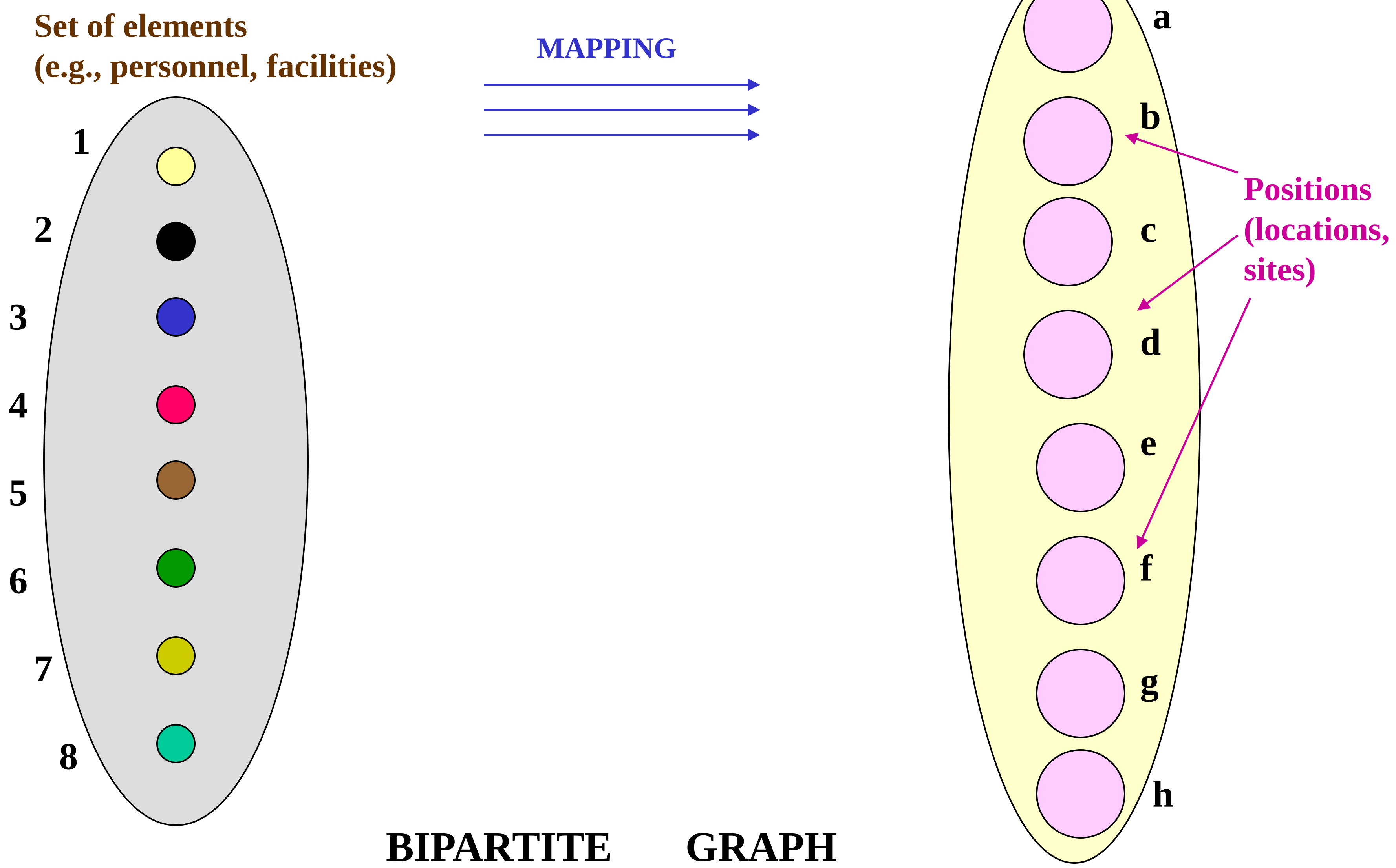
4.List of application domains

5.Basic references (books, sites)

Dec. 3, 2004

Allocation problem

Allocation (assignment, matching, location):

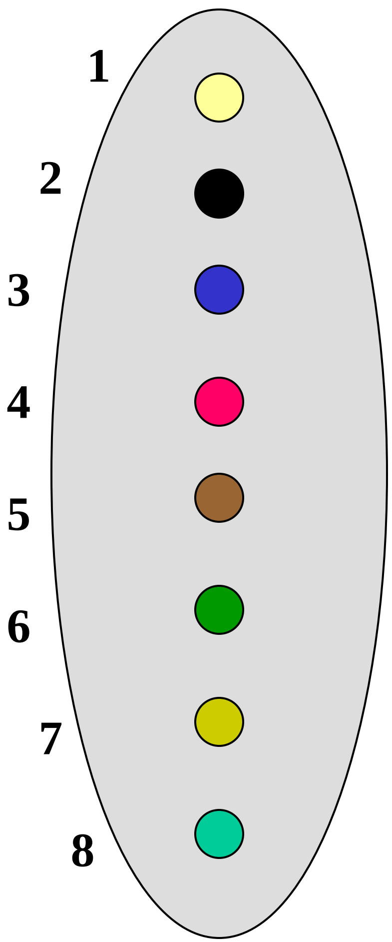


BIPARTITE GRAPH

Allocation problem

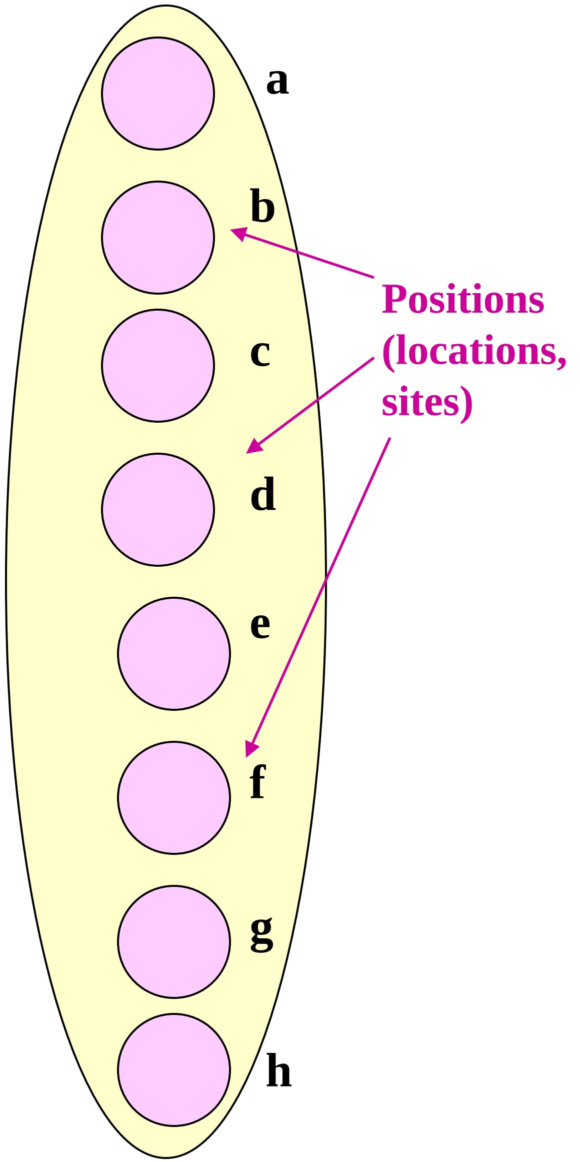
Allocation (assignment, matching, location):

Set of elements
(e.g., personnel, facilities)



matrix of weights c_{ij}

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



BIPARTITE GRAPH

Allocation problem: applied examples for elements & positions

1.Boys -- *Girls* *(marriage problem)*

2.Workers -- *Work positions*

3.Facilities --*Positions in manufacturing system* *(facility layout)*

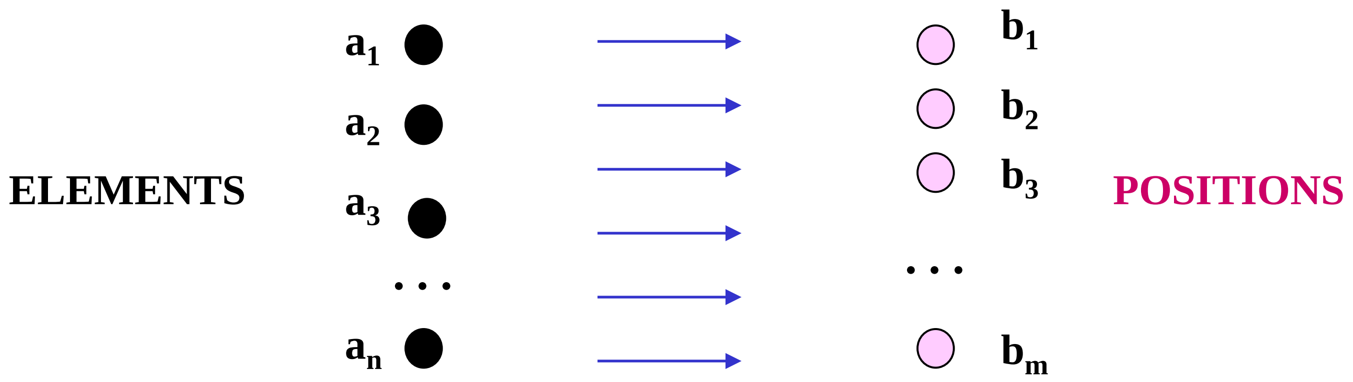
4.Tasks -- *Processors in multiprocessor system*

5.Anti-rockets --*Target in defense systems*

6.Files -- *Databases in distributed information systems*

Etc.

Assignment problem AP



FORMULATION (combinatorial):

Set of elements (e.g., personnel, facilities, tasks): $A = \{ a_1 , \ldots , a_i , \ldots , a_n \}$

Set of positions (e.g., locations, processors) $B = \{ b_1 , \ldots , b_j , \ldots . b_m \}$

(now let $n = m$)

Effectiveness of pair a_i and b_j is: $c (a_i , b_j)$

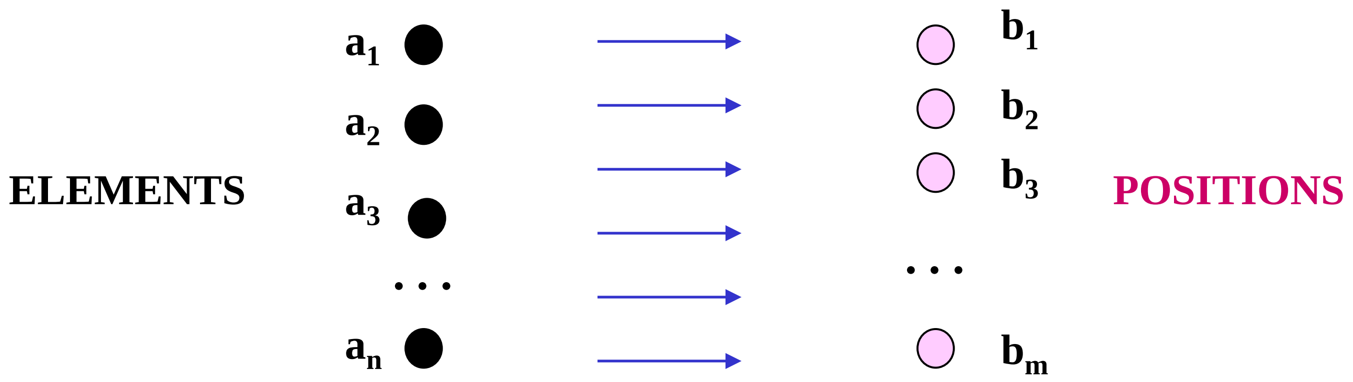
$\prod = \{s\}$ is set of permutations (assignment) of elements of A into position set B:

$s^* = < (s^*[1]), \ldots ,(s^*[i]), \ldots ,(s^*[n]) >$, i.e., element a_i into position s[i]

The goal is: $\max \sum^{n}_{i=1} c (i, s[i])$

Maximum (minimum) weight matching in a weighted bipartite graph

Assignment problem AP



ANOTHER FORMULATION (algebraic):

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

Set of positions (e.g., locations, processors) $B = \{ b_1 , \dots , b_j , \dots . b_m \}$

(now let $n = m$)

Effectiveness of pair a_i and b_j is: $c (a_i , b_j)$

x_{ij} = 1 if a_i is located into position b_j and 0 otherwise ($x_{ij} \in \{ 0,1 \}$)

The problem is: $\max \sum^{n}_{i=1} \sum^{n}_{j=1} c_{ij} \; x_{ij}$

s.t.

$$\sum^{n}_{i=1} x_{ij} = 1 \qquad \forall \; j$$

$$\sum^{n}_{j=1} x_{ij} = 1 \qquad \forall \; i$$

ALGORITMS:

1.Polynomial exact algorithm ($O(n^3)$)

2.Hungarian method

Etc.

OTHER VERSIONS:

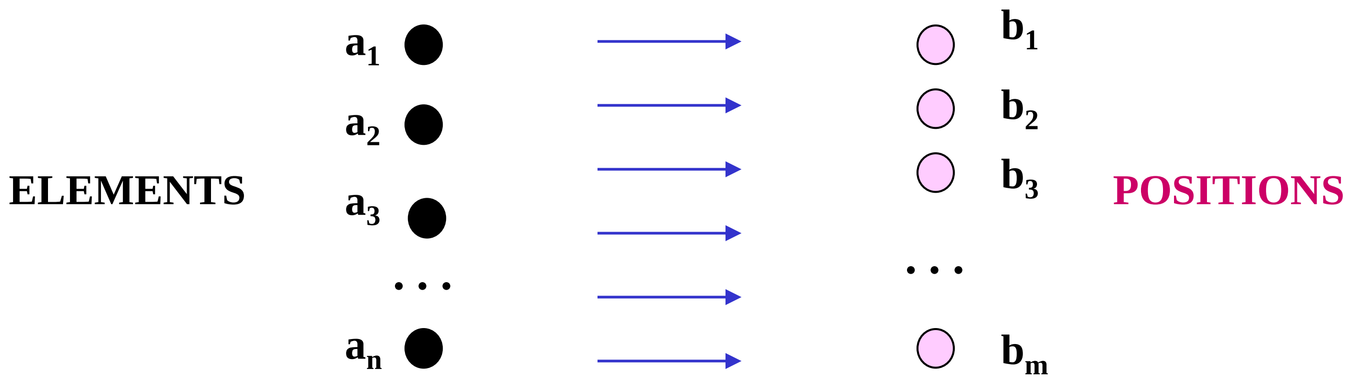
1.”*Minimum*” problem

2.”*Min max”* problem

3.Multicriteria problem

Etc.

Quadratic Assignment Problem QAP



matrix of weights ("flow") c_{ij}

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

matrix of disctance d_{ij}

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

A Basic ("flow") Mathematical Formulation of the Quadratic Assignment Problem

An assignment of elements $\{ i \mid i \in \{1,\ldots,n\} \}$ to positions (locations) $\{ p(i) \}$, where p is permutation of numbers $\{1,\ldots,n\}$, a set of all possible permutations is $\prod = \{ p \}$.

Let us consider two n by n matrices:
(i)a "flow" (or "utility")matrix C whose (i,j) element represents the flow between elements (e.g., facilities) i and j, and
(ii)a distance matrix D whose (i,j) (p(i), p(j)) element represents the distance between locations p(i) and p(j).

With these definitions, the QAP can be written as

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

ALGORITHMS:

1.Branch & Bound method

2.Relaxation approach

3.Greedy algorithms

4.Genetic algorithms (evolutionary computing)

5.Metaheurstics (i.e., local optimization, hybrid schemes)

BASIC BOOKS:

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.

SITES:

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

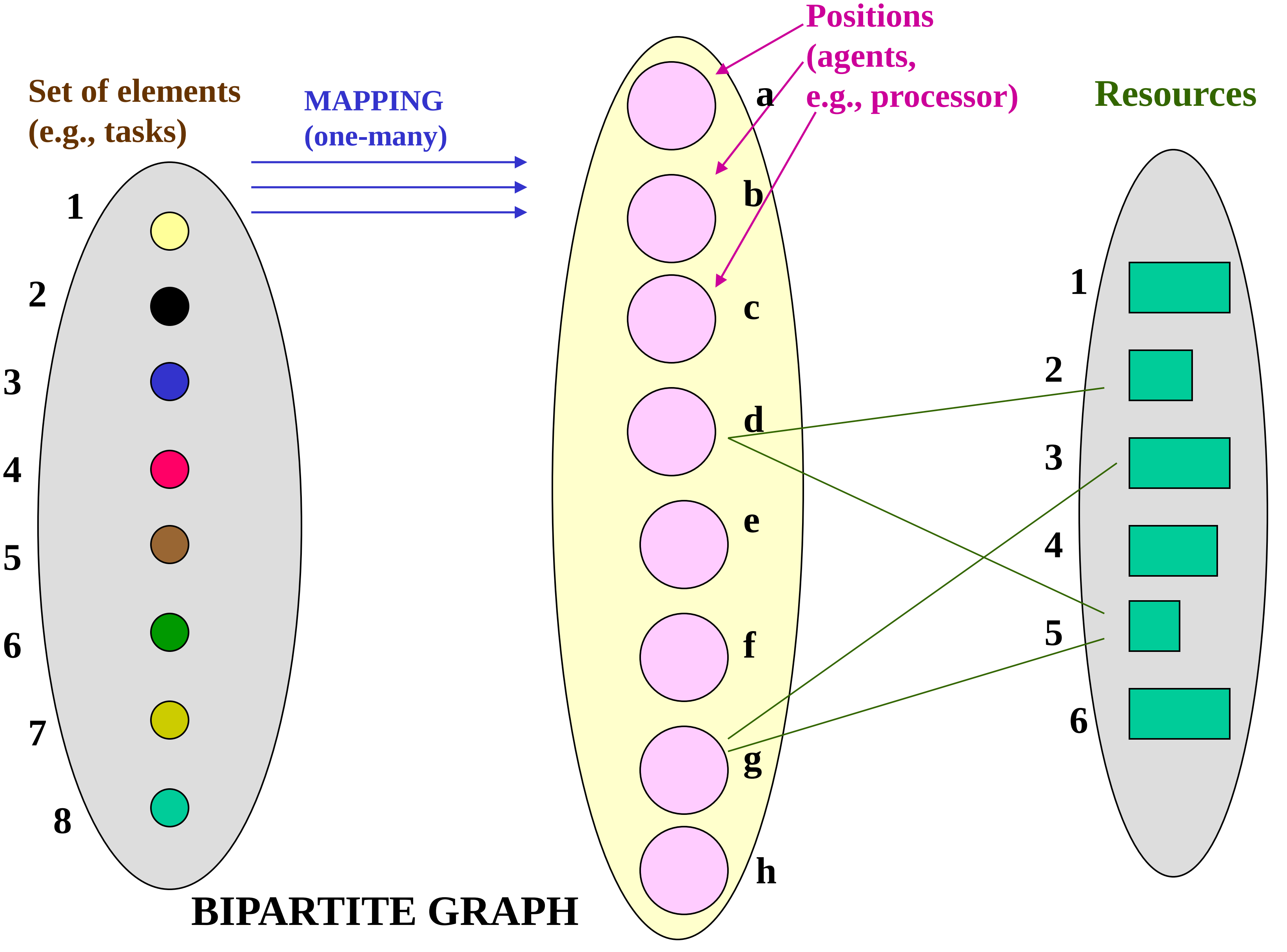
Multi-objective (multicriteria) Assignment Problem (Quadratic Assignment Problem):

Elements in matrix of weights (“flows”) are vectors.

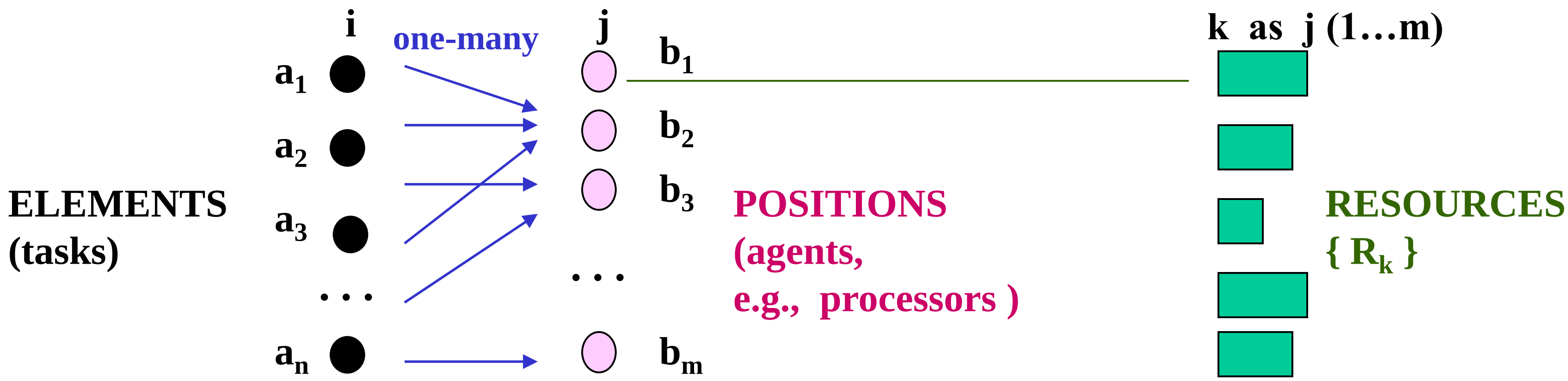
ALGORITHMS

1. **Branch & Bound method**
2. **Relaxation approach**
3. **Greedy algorithms**
4. **Metaheurstics (i.e., local optimization, hybrid schemes)**
5. **Multi-objective evolutionary optimization**

Generalized Assignment Problem GAP



Generalized Assignment Problem GAP



FORMULATION (algebraic):

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

Set of positions (e.g., locations, processors) $B = \{ b_1 , \dots , b_j , \dots . b_m \}$

(now let n = m)

Effectiveness of pair a_i **and** b_j **is:** $c (a_i , b_j)$

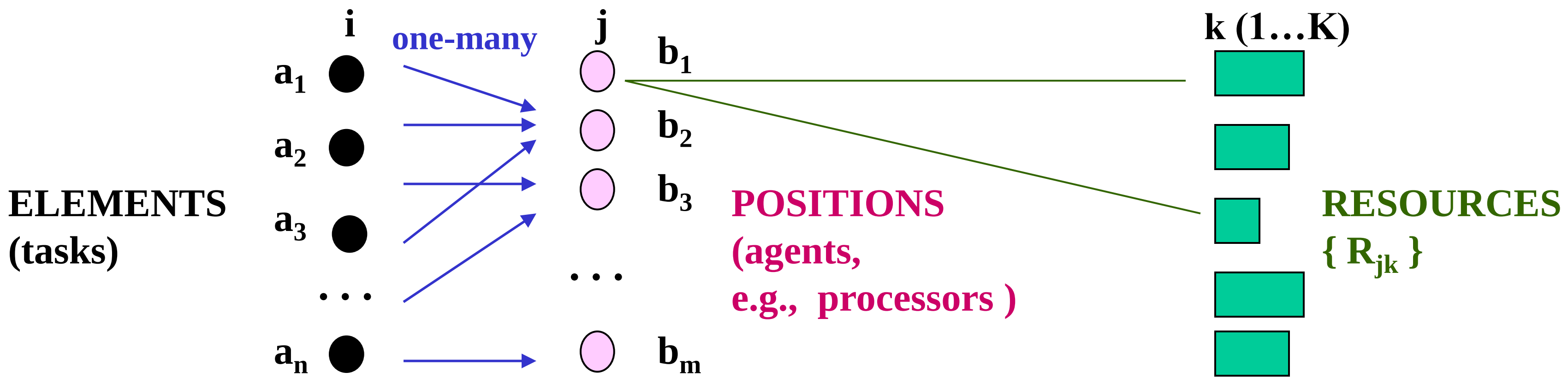
x_{ij} = 1 if a_i is located into position b_j and 0 otherwise ($x_{ij} \in \{ 0,1 \}$)

The problem 1 is: $\max \sum^n_{i=1} \sum^n_{j=1} c_{ij} \; x_{ij}$

s.t. $\sum^n_{i=1} r_{ik} \; x_{ij} \le R_j \quad \forall \; j$ (R_k is resource of agent k)

$\sum^n_{j=1} x_{ij} = 1 \quad \forall \; i$

Generalized Assignment Problem GAP (multiple common resources)



FORMULATION (algebraic):

Set of elements (e.g., personnel, facilities, tasks): $A = \{ a_1 , \ldots , a_i , \ldots , a_n \}$

Set of positions (e.g., locations, processors) $B = \{ b_1 , \ldots , b_j , \ldots . b_m \}$

(now let n = m)

Effectiveness of pair a_i and b_j is: $c (a_i , b_j)$

x_{ij} = 1 if a_i is located into position b_j and 0 otherwise ($x_{ij} \in \{ 0,1 \}$)

The problem 2 is: $\max \sum^n_{i=1} \sum^n_{j=1} c_{ij} \; x_{ij}$

s.t. $\sum^m_{j=1} \sum^n_{i=1} r_{ik} \; x_{ij} \leq R_k \quad \forall \; k$ **(common K resources)**

$\sum^n_{j=1} x_{ij} = 1 \quad \forall \; i$

ALGORITHMS:

1.Branch & Bound method

2.Relaxation approach

3.Heuristics (greedy algorithms, etc.)

4.Genetic algorithms (evolutionary computing)

5.Metaheurstics (i.e., local optimization, hybrid schemes)

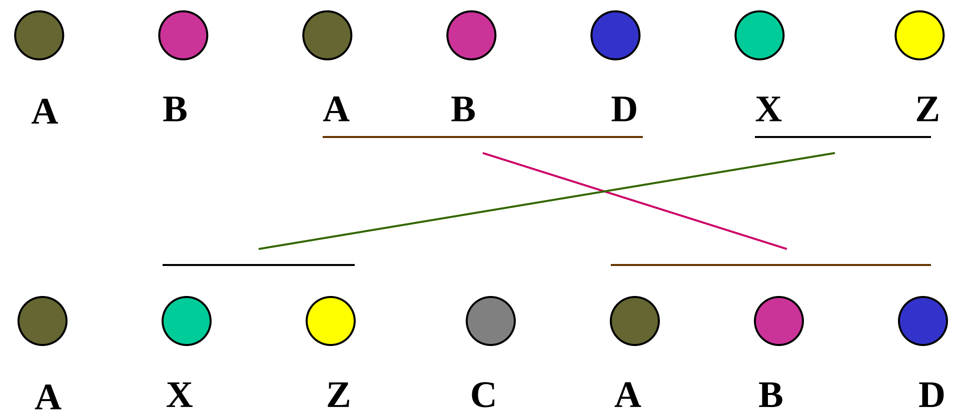
EXTENSION:
1.Multicriteria cases
2.Uncertainty
3.Etc. (e.g., dependence of problem parts, dependence on time)

String matching

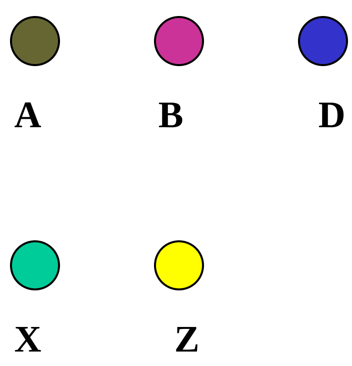
CASE OF 2 WORDS:

String (word) 1

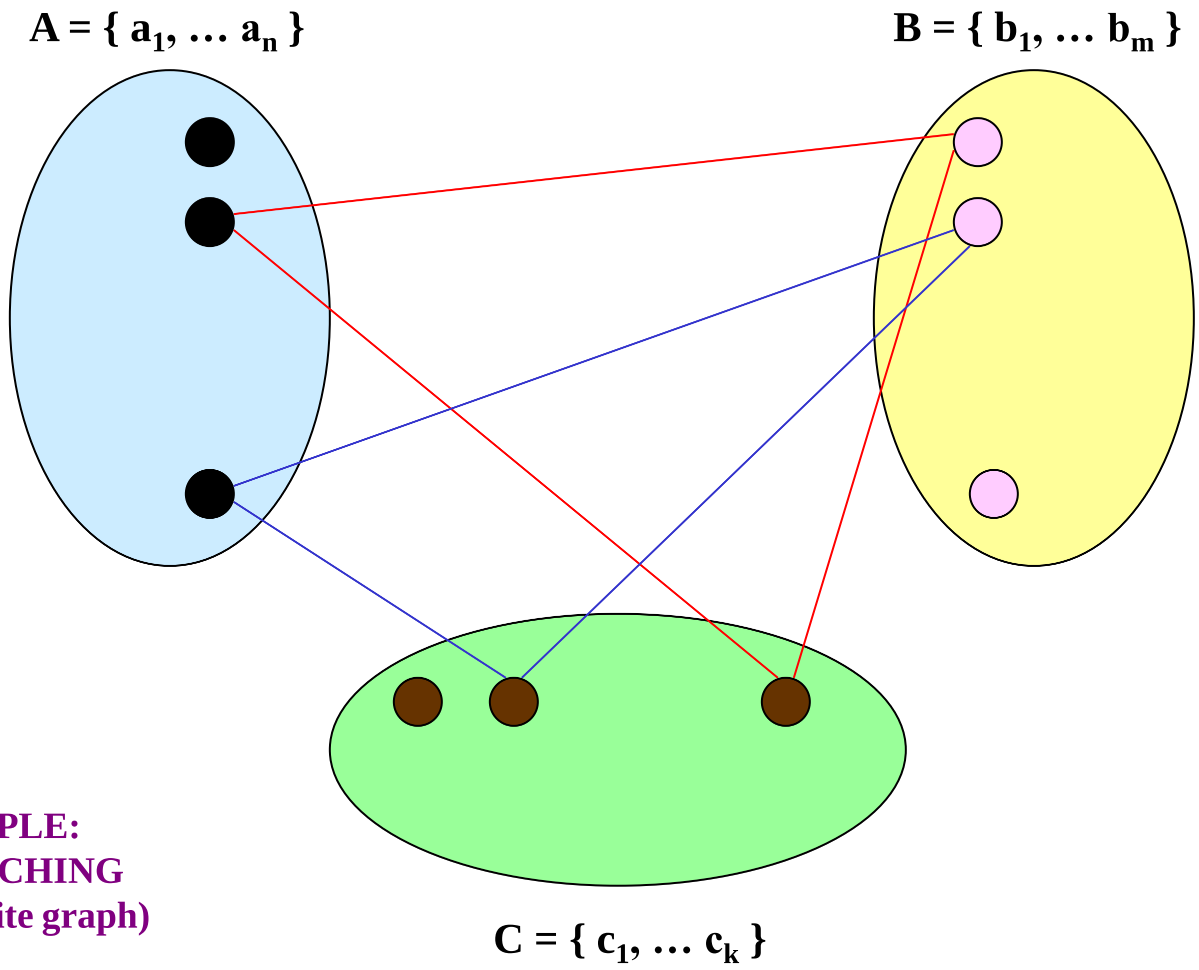
String (word) 2



PATTERNS:



Multiple matching problem (Lecture 17-18)



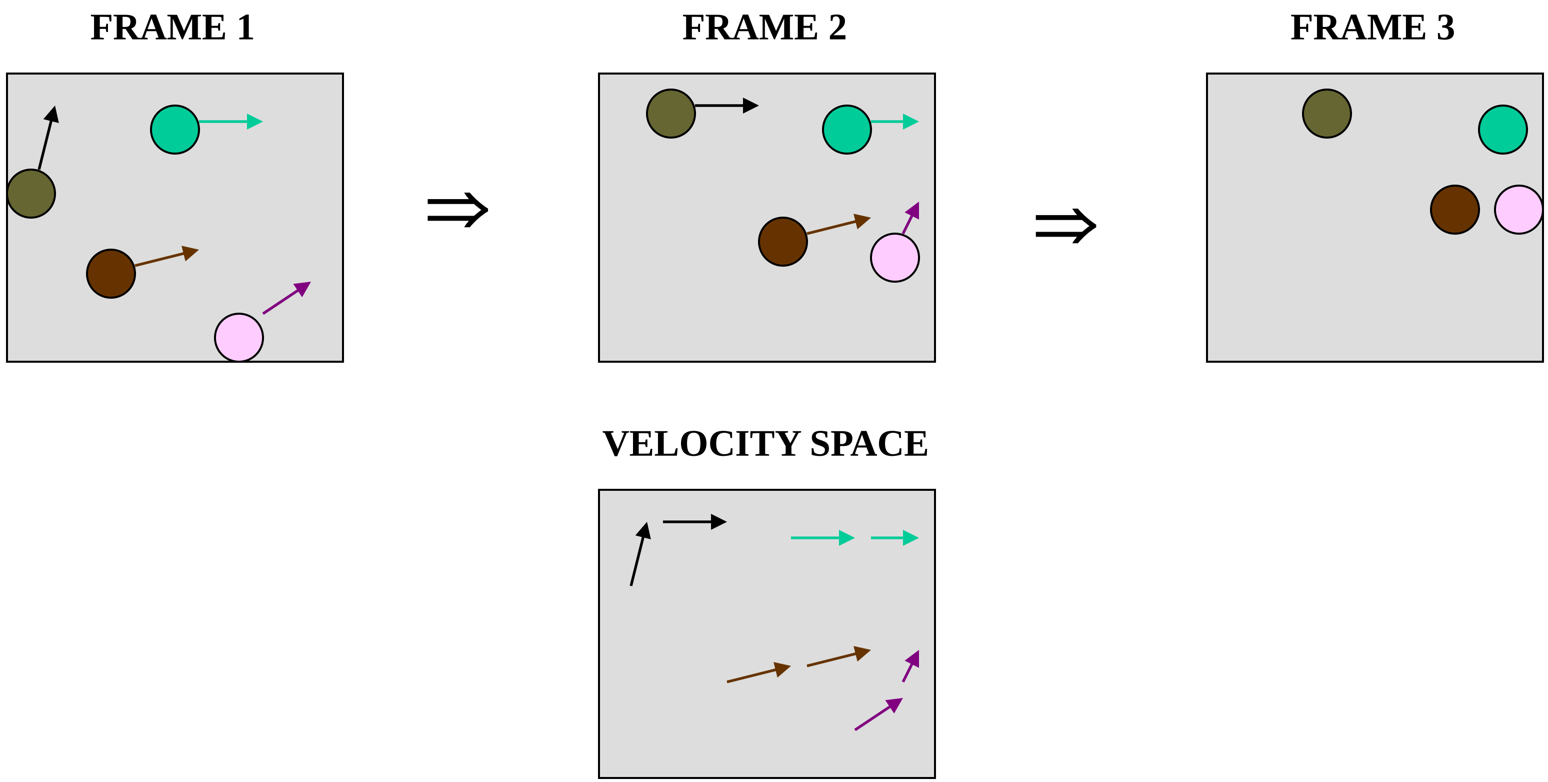
EXAMPLE:
3-MATCHING
(3-partite graph)

Multiple matching problem (Lecture 17-18)

ALGORITMS:
1.Enumerative algorithms (e.g., Branch-And-Bound)
2.Heursitcs (e.g., greedy algorithms,
various local optimization techniques)
3.Metaheuristics including hybrid methods
4.Morphological approach

VERSIONS:
1.Dynamical (multi-stage) problem (multiple track assignment)
2.Problem with errors
3.Problem with uncertainty (probabilistic estimates, fuzzy sets)
Etc.

Usage of assignment problem to define velocity of particles, etc. (Lecture 17-18)



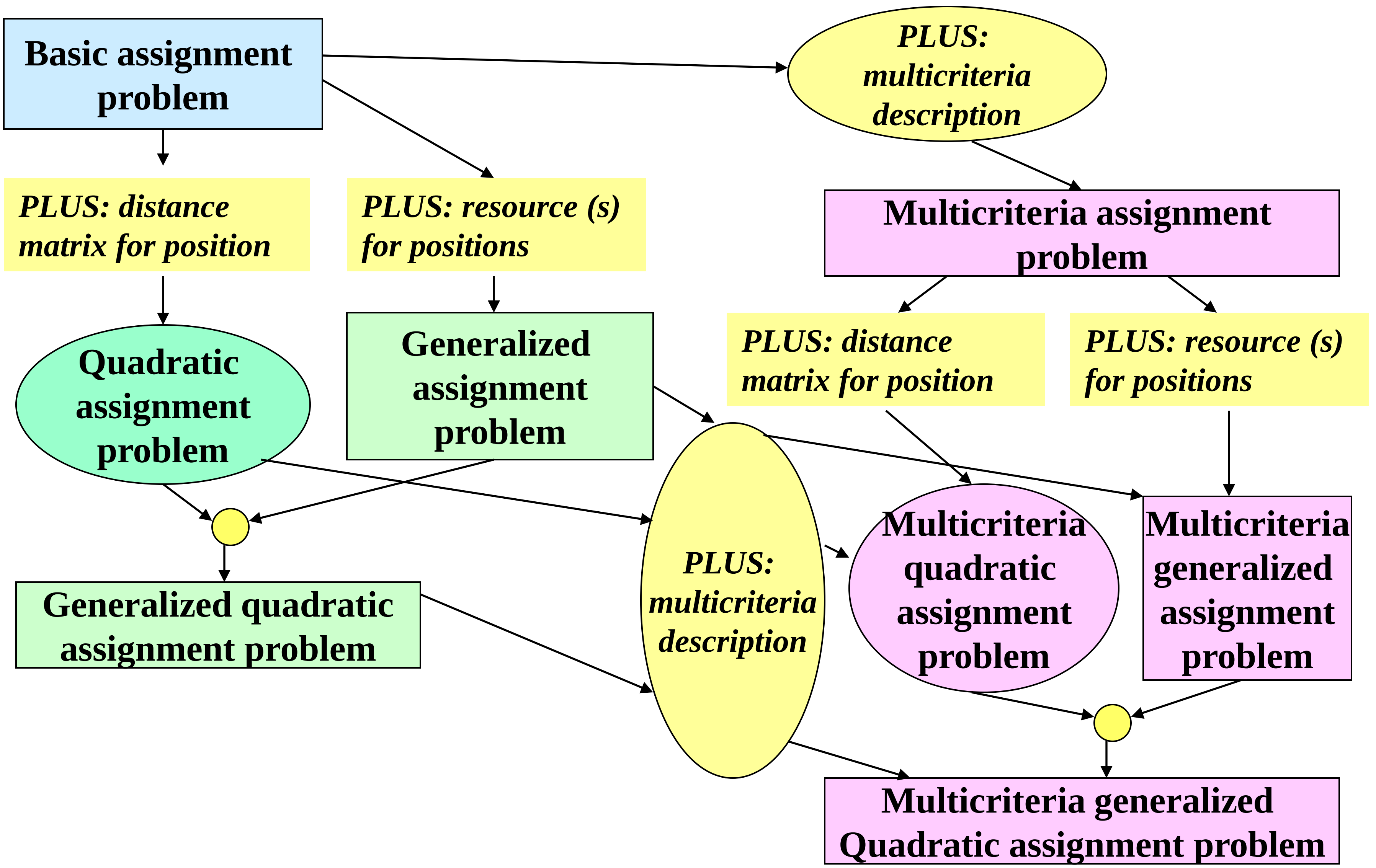
APPLICATIONS (air/ water environments):
1. Physical experiments
2. Climat science
3. Chemical processes
4. Biotechnological processes

Contemporary sources:
1. PIV systems (laser/optical systems)
2. Sattelite photos
3. Electronic microscope, etc.

List of algorithms for allocation-like problems

1.polynomial algorithms (e.g., Hungarian method, techniques on the basis of flow-like ideas, etc.)
2.LP-based algorithms
3.epsilon-approximation polynomial algorithms
4.data-correction algorithms
5.Branch-And-Bound method
6.dynamic programming approach
7.evolutionary and genetic algorithms
8.Bender's decomposition scheme
9.Local optimization techniques as various heuristics (Tabu-search algorithms, simulated annealing, hybrid schemes etc.)
10.techniques of multiple criteria analysis (e.g., multi-attribute utility analysis, Analytic Hierarchy Process, ELECTRE outranking method)
11.simulation based approaches
12.polyhedral methods
13.hierarchical approaches including morphological (decomposition) approach
14."constraint satisfaction problem" approach
15.fuzzy methods
16.knowledge-based algorithmic rules
20.neural networks

Evolution chart of allocation-like problems



Application domains of allocation-like problems

1.facility allocation (layout) in manufacturing systems
2.allocation of resources in investment actions
3.allocation of facilities in supply chain management
4.assignment routing problems
5.allocation / layout in VLSI design
6.allocation / layout in the space design of buildings including topological / layout in multi-floor building design
7.allocation in architectural planning, e.g., in urban systems (allocation of buildings and / or components of infrastructure as railway stations, hospitals, shop-centers, schools, parks, etc.) including site search (selection) problem, i.e., allocation of sites on a map
8.allocation of emergency service facilities
9.allocation of buffer capacities in production lines (including distribution of maintenance operations)
10.allocation of total maintenance investment among products
11.allocation of inspection efforts in various systems (e.g., production systems), allocation of test positions and test time and optimal allocation of test resources
12.allocation of operations for balance control in assembly lines
13.allocation of human-machine functions
14.portfolio selection (optimization) on the basis of assignment-like models (e.g., quadratic assignment problem)
15.project tasks assignment

Application domains of allocation-like problems

16.allocation of resources in large team
17.allocation of traffic police resources
18.blood assignment in a donation-transfusion system
19.assignment of papers to reviewers
20.assignment in sport
21.dynamic storage allocation , note this problem is a complex one and often corresponds to a set of combinatorial problems including bin-packing, segmentation / partitioning, scheduling, etc.
22.task allocation / assignment in distributed (e.g., computer and / or information) systems
23.allocation of production tasks in a virtual organization
24.information (e.g., databases files, documents) allocation in distributed computer / information environments and networks, for example, allocation / reallocation of arrived data items in a distributed information system (a set of servers), placement and replacement for large-scale distributed cashes in digital libraries or in Internet
25.capacity assignment / allocation in communication networks
26.allocation of Internet resources 27.allocation of resources in e-business infrastructure
28.design of standards
29.multisensor multitarget tracking on the basis of non-linear and multidimensional assignment problems (e.g., in radar systems)
30.allocation of reliability among components that are to be assembled into a system or component redundancy allocation
31.assignment problem in experimental high energy physics
32.locomotive assignment to train-segments

Application domains of allocation-like problems

33. assignment / allocation problems in universities (e.g., assignment of students to exams, assignment of faculty members to courses, assignment of auditoriums for lectures, allocation of the most desirous projects to the most qualified students)

34. assignment / allocation of particles / points (particle matching) in particle image velocimetry (PIV) and particle tracking velocimetry (PTV) for measurement in fluid mechanics

35. personnel management systems (e.g., allocation of personnel, allocation of tasks, role assignment) including assignment of technicians to handle computer system faults (e.g., hardware, software, communication)

36. allocation of rooms among people

37. allocation of rights (e.g., in social networks for social choice and welfare, for participants of electronic financial markets

38. allocation of discrete resources

39. channel and frequency assignment in mobile radio systems (including dynamical assignment)

40. assigning cells to switches in cellular mobile networks

41. allocation of tolerances in manufacturing systems

42. wavelength allocation on trees of rings for optical communication networks

Application domains of allocation-like problems

43.allocation in medical organizations, for example:
(a) allocation of surgeries to operating rooms, (b) allocation of inpatient resources,
(c) staff assignment problem, (d) the workshift and rest assignment of nursing personnel,
(e) hospital facility layout, (f) bed allocation, (g) organ allocation,
(h) patient assignment, and (i) medical service allocation and reallocation

44.bottleneck allocation methodology for scheduling in manufacturing systems

45.traffic assignment in transportation networks and in communication / computer networks

46.hierarchical location-allocation problems on networks

47.dynamic allocation of resources in multi-project research and development systems

48.allocation of ecological facilities

49.hierarchical location-allocation of banking facilities

Basic Books on Allocation/ Location Problems

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/

LECTURE 31. Course: "Design of Systems: Structural Approach"

Dept. "Communication Networks &Systems", Faculty of Radioengineering & Cybernetics

Moscow Institute of Physics and Technology (University)

Mark Sh. Levin
Inst. for Information Transmission Problems, RAS

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

L.31. Statisfiability problem. Six basic combinatorial problems. Timetabling.

PLAN:

1.Satisfiablity problem:

*formulation and illustration , *modification, *applications

2.Basic combinatorial problems

(satisfiability, 3-satisfiability, vertex covering, 3-matching, clique, Hamiltonian cycle,

partitioning,)

3.Timetabling problems

*formulation and illustration, *applications, *algorithms and solving schemes, *basic references (papers, sites)

Dec. 10, 2004

Satisfiability problem

BOOLEAN VARIABLES:

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

LOGICAL OPERATIONS (AND, OR, NOT):

*logical negation NOT *conjunction AND,
*parentheses for grouping

SET OF BOOLEAN EXPRESSIONS C:

Examples:

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

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

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

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

ALL TRUE (1)

Satisfiability is *the* original NP-complete problem.

Despite its applications to
***constraint satisfaction,**
***logic, and**
***automatic theorem proving,**

it is perhaps most important theoretically as
the root problem
from which all other NP-completeness proofs originate.

Satisfiability

Satisfiability, or SAT for short, is the following problem:

Given an expression in propositional logic is there a satisfying assignment?
For example, can we assign a value of true or false to each of the variables x, y, z such that the following expression is true?
(x or y) & (x or not y) &(not x or z) & (not z or not x or not y)
In this case the answer is yes: the satisfying assignment is x=true, y=false, z=true
There is no algorithm known that is guaranteed to solve any problem of this kind in a time polynomial in the number of variables.
The amazing discovery of Cook and others in the early 70s was that many natural problems such as scheduling, and many questions in networks, graphs, etc. can be transformed into the above form (see Garey & Johnson, 1979).
Recently a popular area of interest is SAT-based planning (Kautz and Selman, 92).
In this approach the task of finding a plan in a given domain is converted to the task of finding a model for a certain satisfiability problem.

MAXIMUM SATISFIABILITY

INSTANCE: Set *U* of variables,
collection *C* of disjunctive clauses of literals,
where a literal is a variable or a negated variable in *U*.

SOLUTION: A truth assignment for *U*.

MEASURE:
Number of clauses satisfied by the truth assignment.

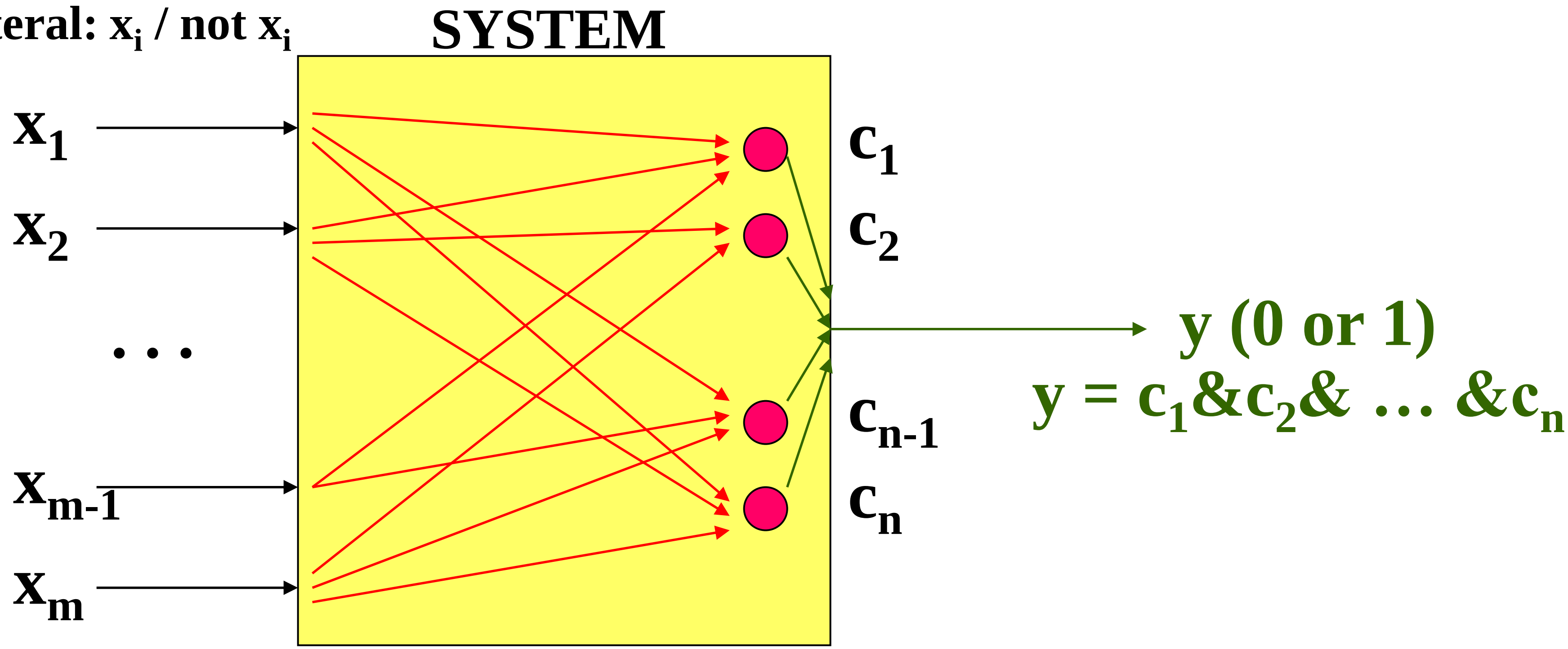
APPLICATIONS:

1. **Software Verification**
2. **Electronic Design Automation and Verification**
3. **Model Analysis**
4. **Model Checking**
5. **Theorem Prover**
6. **AI Planning**

Satisfiability problem: illustration for application in software / electronic systems

PROBLEM: Exist $x_o=(x_1,\ldots,x_n)$ that $y(x_o)=1$ OR not

Literal: x_i / not x_i



Example:

c_1 = not x_1 OR x_2 OR x_4 OR not x_5 OR x_7

c_2 = x_1 OR not x_2 OR not x_3 OR x_5 OR x_7

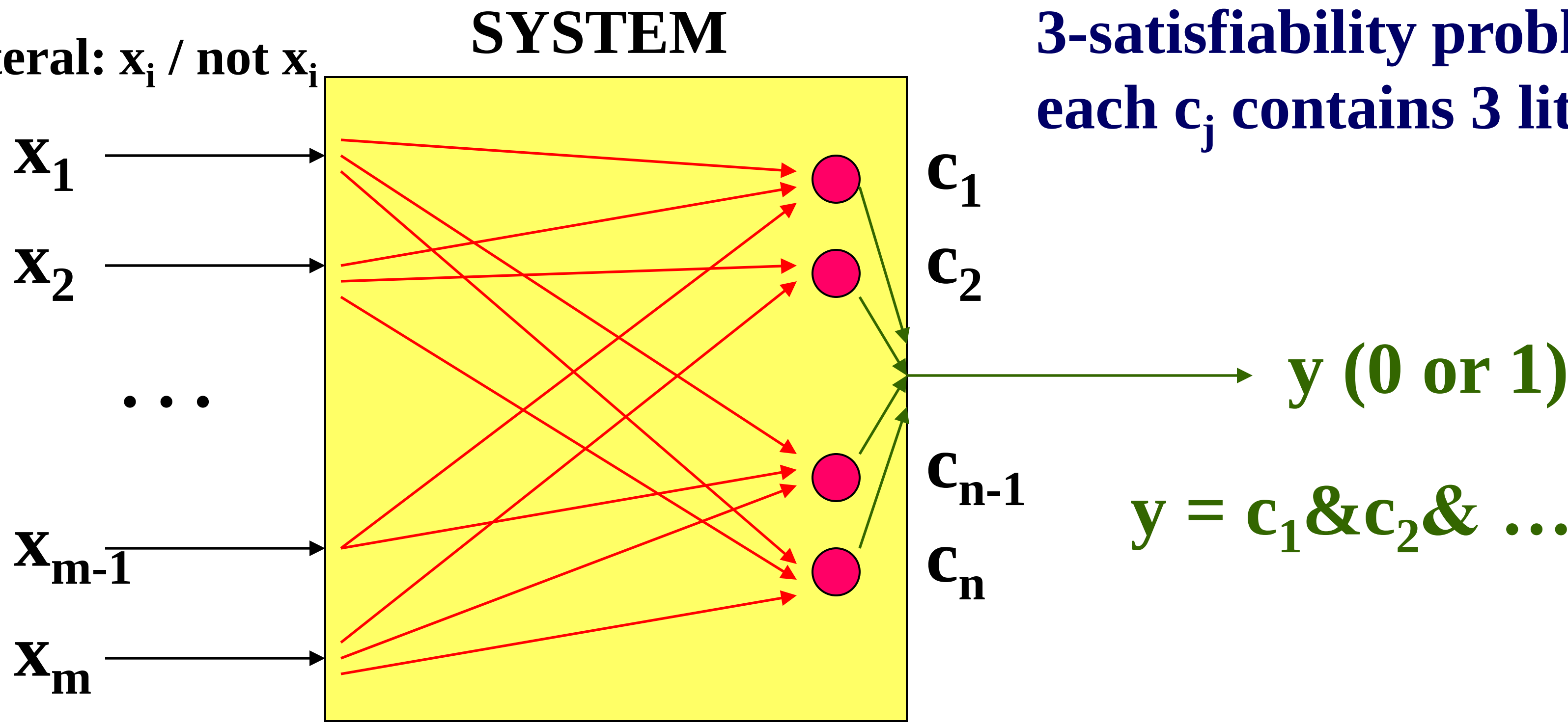
c_3 = not x_1 OR not x_2 OR x_3 OR not x_5 OR not x_n

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

. . .

3-satisfiability problem: each c_j contains 3 literals

Literal: x_i / not x_i



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

Example:

c_1 = not x_1 OR x_2 OR x_4

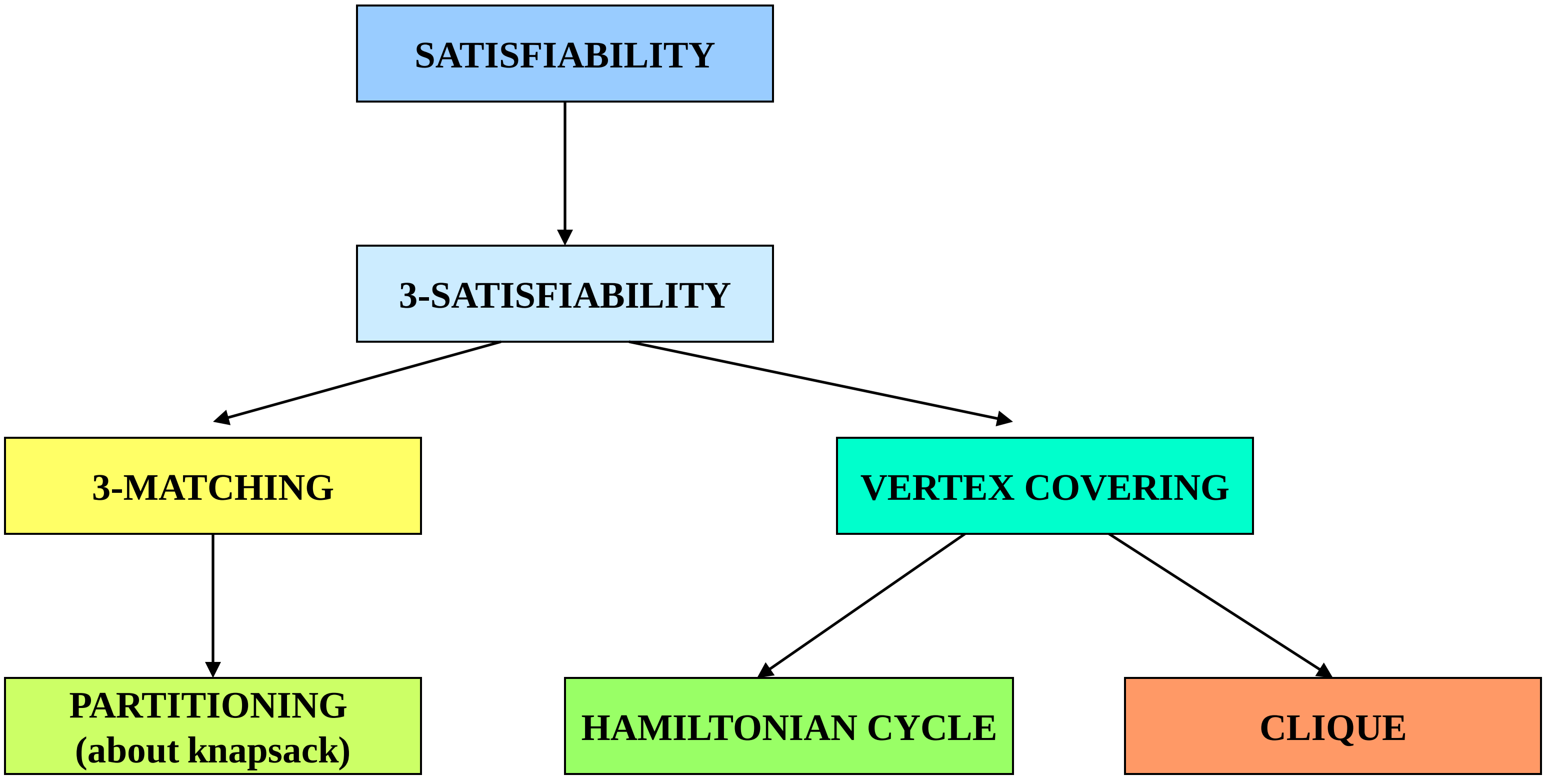
c_2 = x_2 OR not x_3 OR x_7

c_3 = not x_1 OR not x_5 OR not x_n

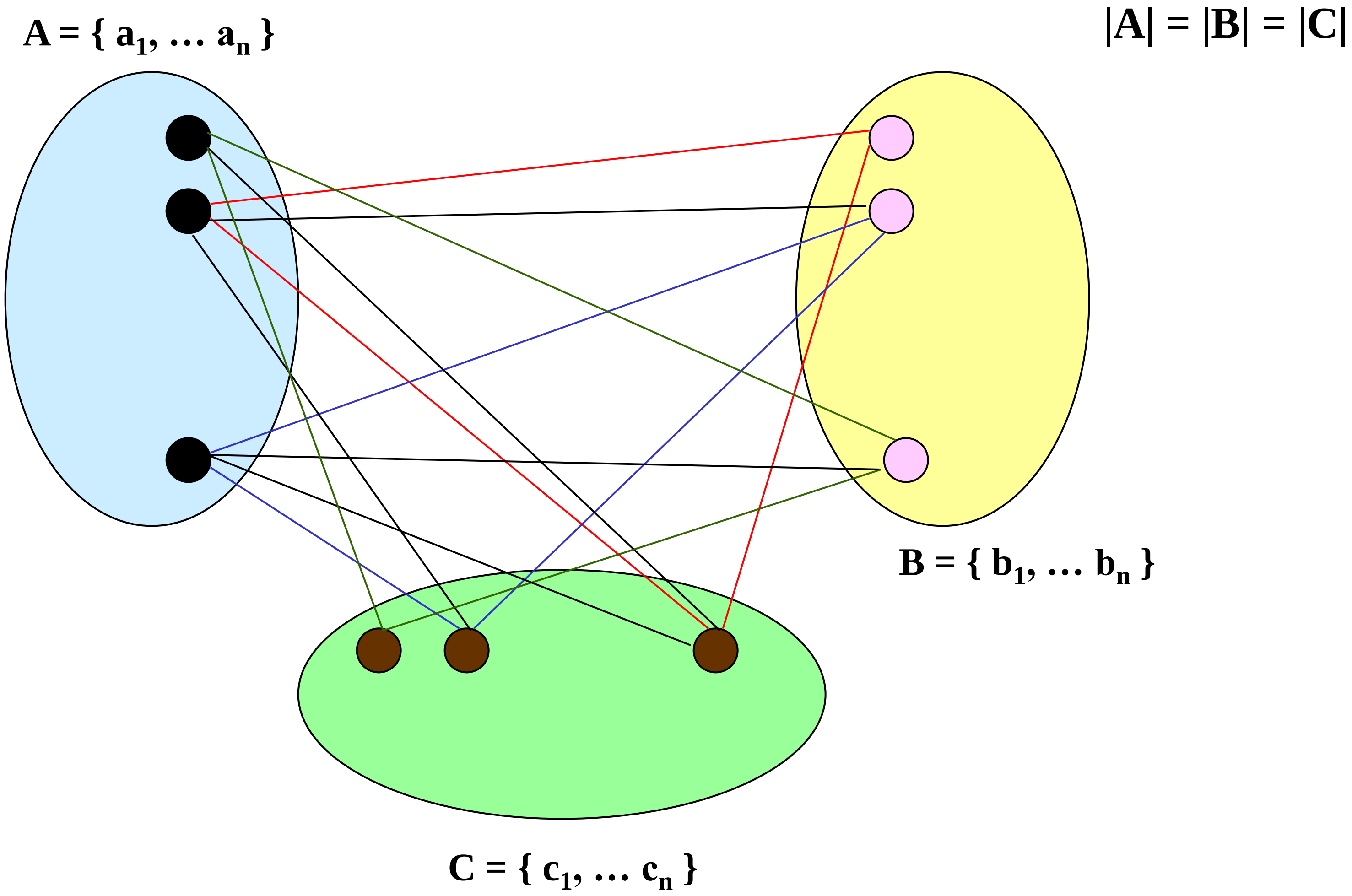
c_4 = not x_2 OR x_{n-2} OR x_{n-1}

. . .

BASIC 6 NP-COMPLETE PROBLEMS AND DIAGRAM



3-matching problem



PROBLEM: Exist covering by vertex triples (without intersection) OR not

Set of elements A = { 1, … , i , … , n }
"weights" of elements { b_1, …, b_i , … , b_n }

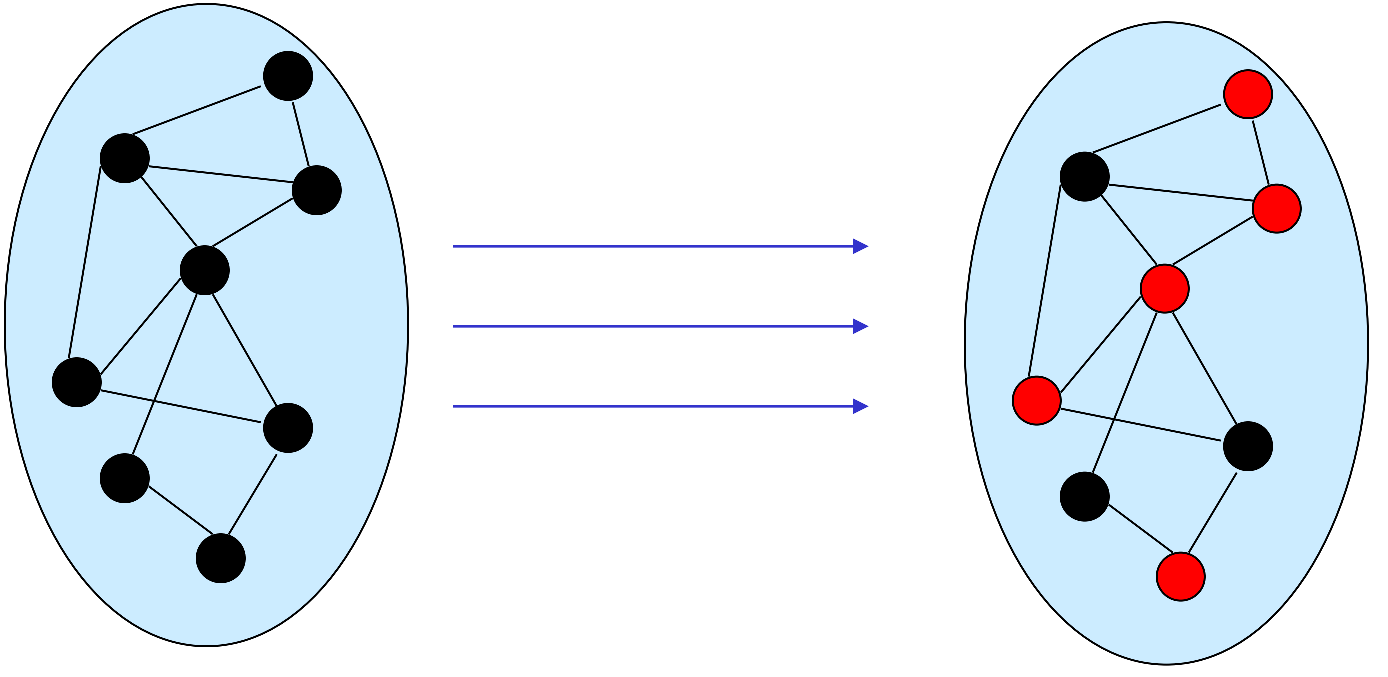
PROBLEM: Exist A' ⊂ A that

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

OR not

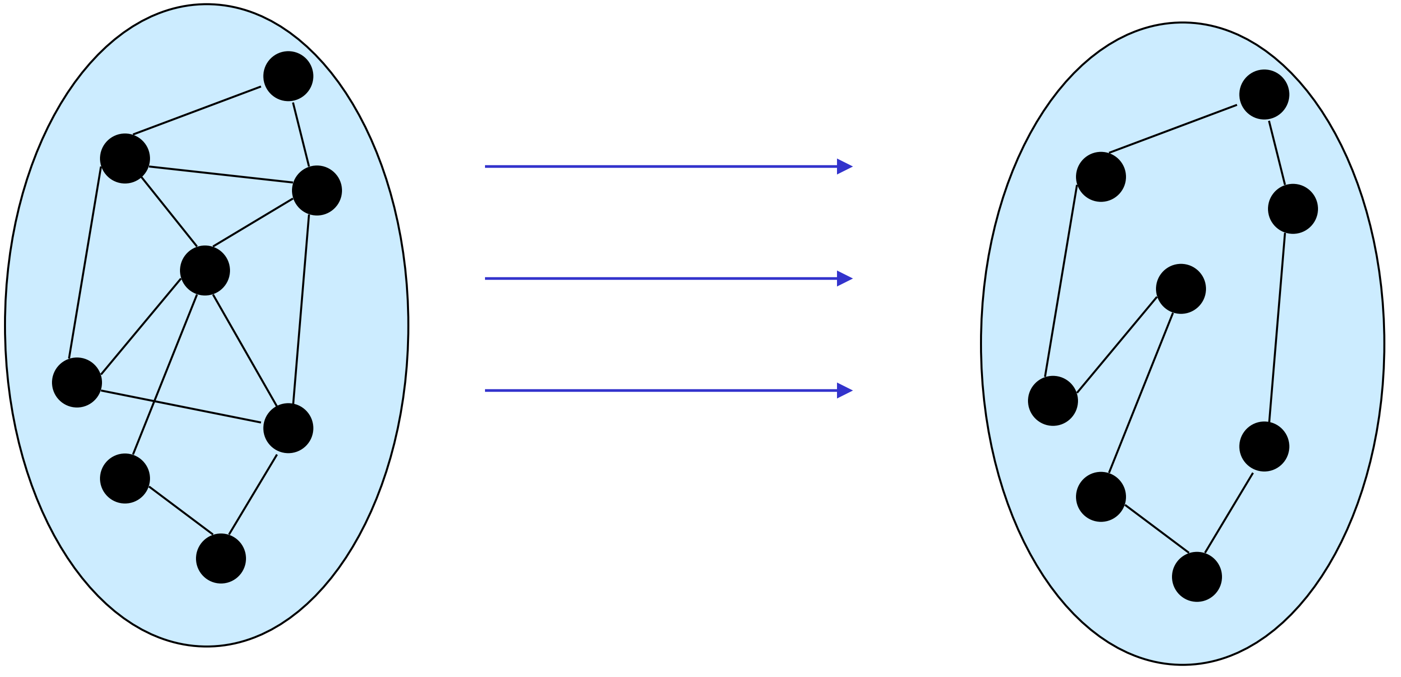
Vertex covering problem

Vertex set A = { a_1, … a_n }, edge set E={e_1, …,e_k}, graph G = (A, E)



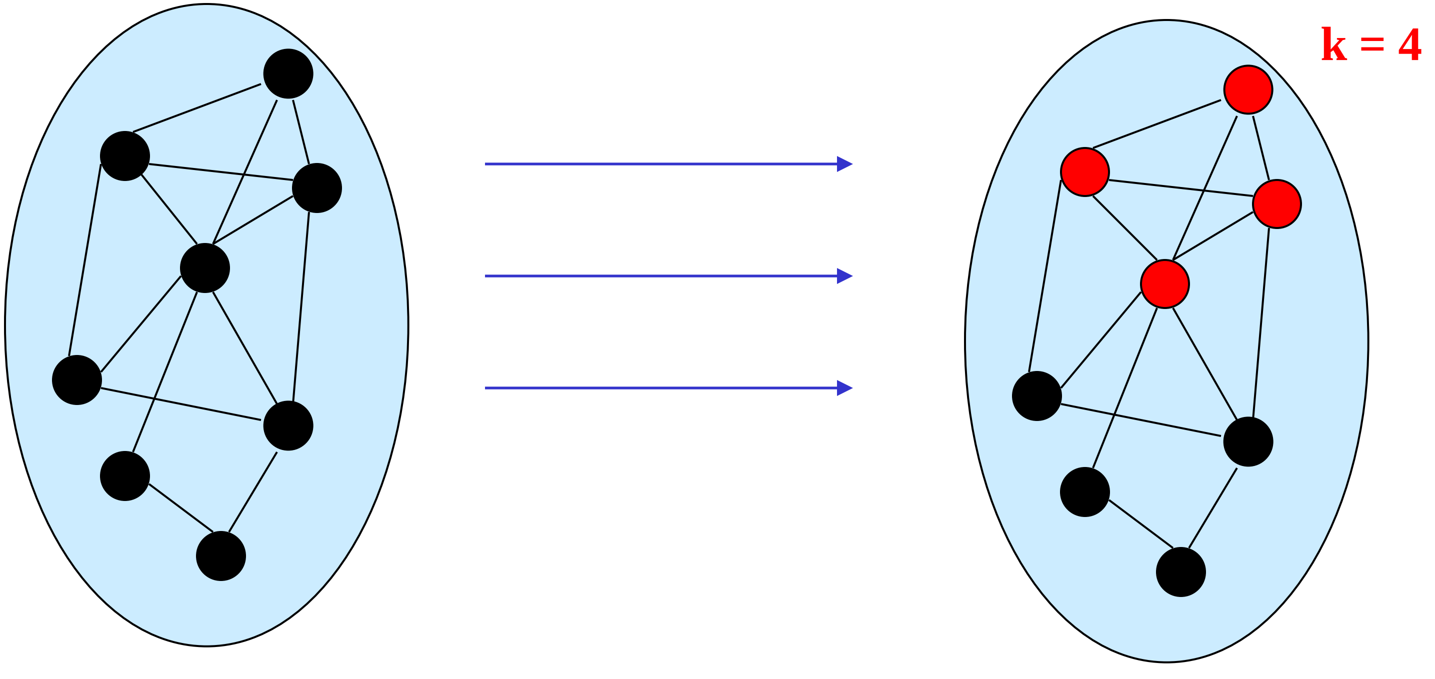
PROBLEM: find vertex covering (A' ⊆ A) that covers ∀e ∈ E

Hamiltonian cycle problem



PROBLEM: Exist Hamiltonian cycle OR not

Clique problem



PROBLEM: Exist subgraph as clique with vertex number k OR not

BASIC (& CLOSE) COMBINATORIAL MODELS:

1. **Graph coloring**
2. **Assignment / Allocation / Matching problems**
3. **Marriage problems (e.g., stable marriage problem)**
4. **Scheduling problems**
5. **Cyclic scheduling problems**
6. **Maximum clique problem**
7. **Combinatorial design (e.g., Latin square)**

Etc.

ALGORITHMS AND ALGORITHM SCHEMES:

1.ENUMERATIVE METHODS

2.AI METHODS:

expert systems (e.g., production-based systems)

knowledge-based systems

neural networks

3.CONSTRAINED PROGRAMMING

4.METAHEURISTICS (various local optimization techniques):

ant colony optimization,

iterated local search

simulated annealing

Tabu search

genetic algorithms

5.COMBINATORIAL DESIGN

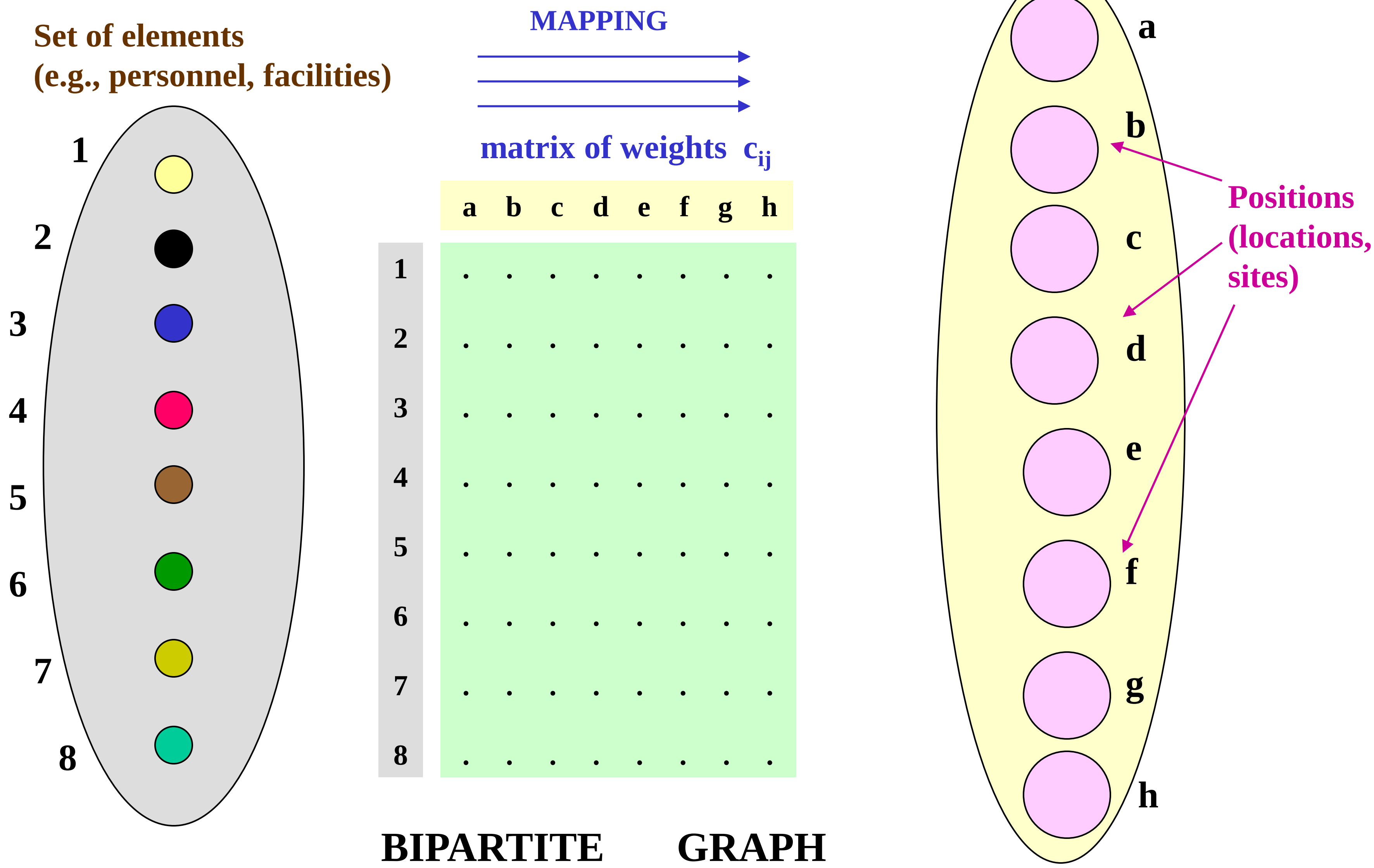
6.EVOLUTIONARY COMPUTATION

(e.g., multi-objective evolutionary optimization)

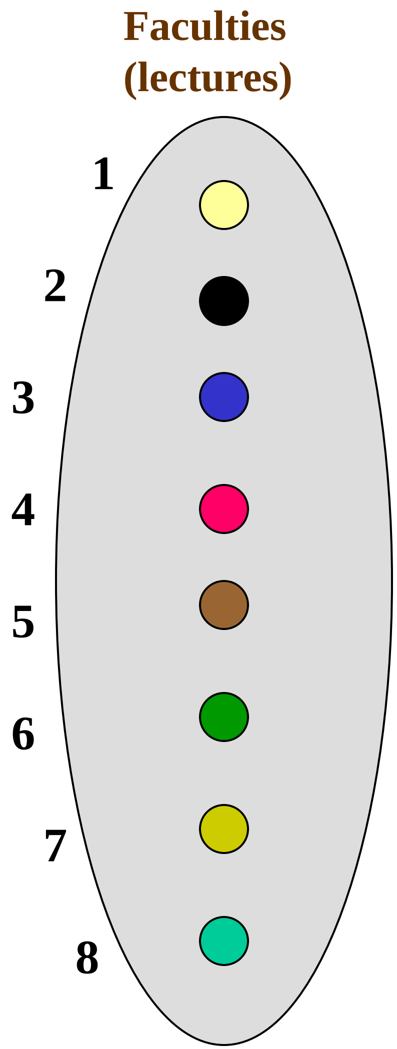
7.HYBRID SOLVING SCHEMES

Allocation problem (from lecture 30)

Allocation (assignment, matching, location):



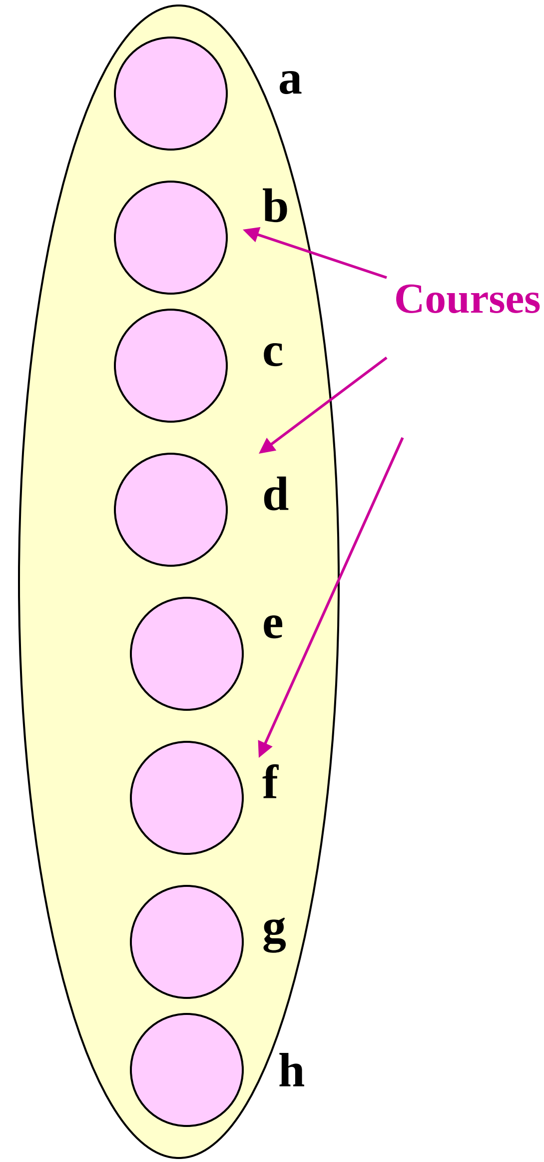
Allocation (assignment, matching, location):



MAPPING

matrix of weights c_{ij}

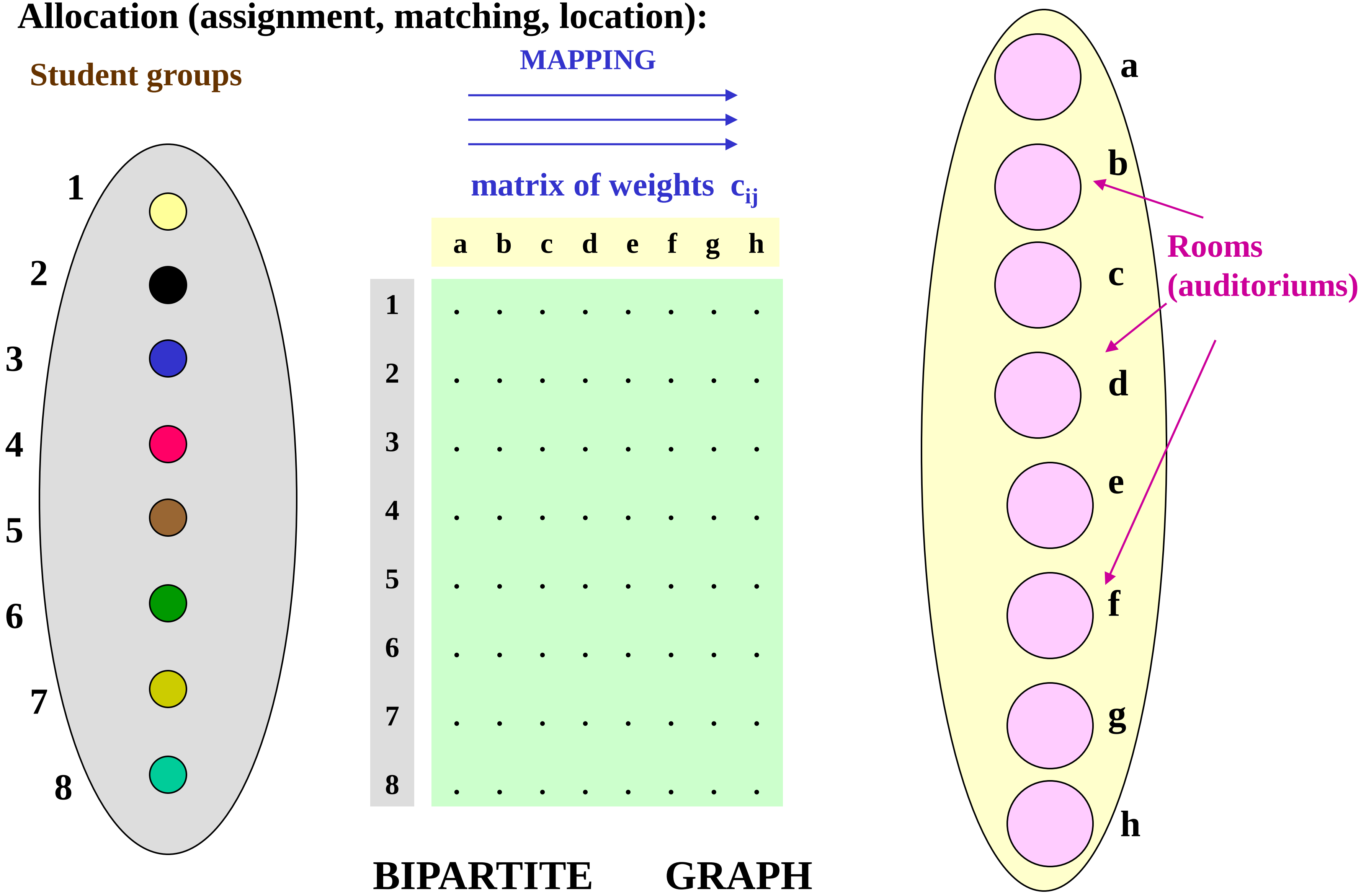
	a	b	c	d	e	f	g	h
1	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.
4	.	.	.	.	.	.	.	.
5	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	.	.
8	.	.	.	.	.	.	.	.



BIPARTITE GRAPH

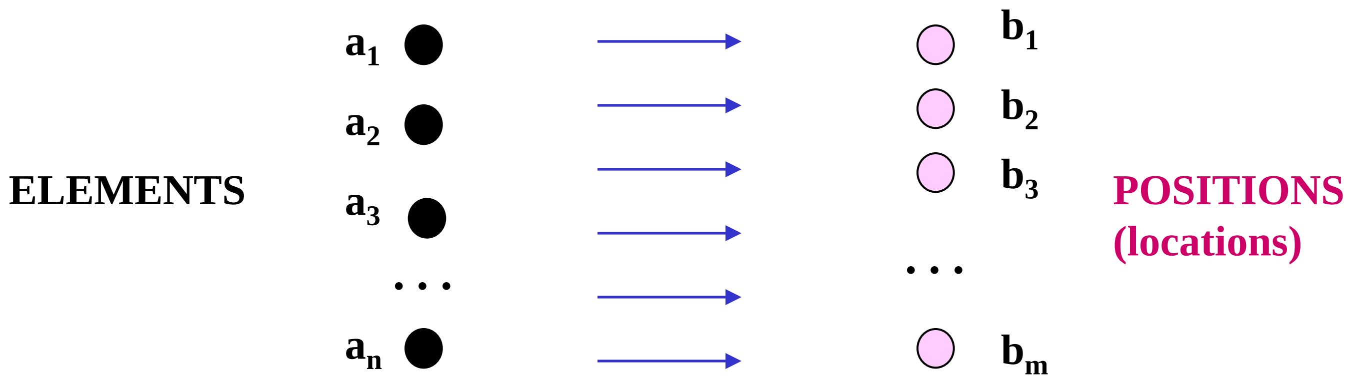
Allocation problem (from lecture 30)

Allocation (assignment, matching, location):



BIPARTITE GRAPH

Assignment problem AP (from lecture 30)



ANOTHER FORMULATION (algebraic):

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

Set of positions (e.g., locations, processors) $B = \{ b_1 , \dots , b_j , \dots . b_m \}$

(now let $n = m$)

Effectiveness of pair a_i and b_j is: $c (a_i , b_j)$

x_{ij} = 1 if a_i is located into position b_j and 0 otherwise ($x_{ij} \in \{ 0,1 \}$)

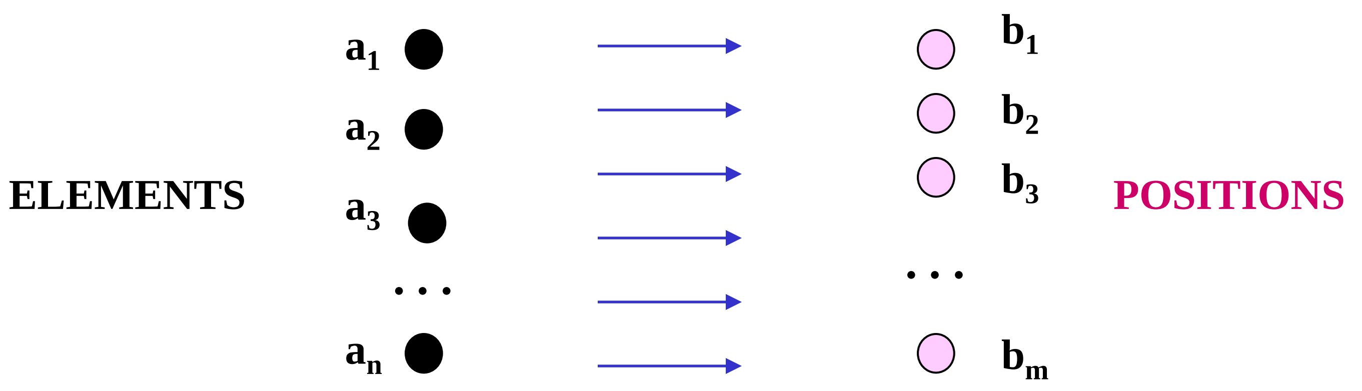
The problem is: $\max \sum^{n}_{i=1} \sum^{n}_{j=1} c_{ij} \; x_{ij}$

s.t.

$$\sum^{n}_{i=1} x_{ij} = 1 \qquad \forall \; j$$

$$\sum^{n}_{j=1} x_{ij} = 1 \qquad \forall \; i$$

Quadratic Assignment Problem QAP (from lecture 30)



matrix of weights ("flow") c_{ij}

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

matrix of disctance d_{ij}

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

A Basic ("flow") Mathematical Formulation of the Quadratic Assignment Problem

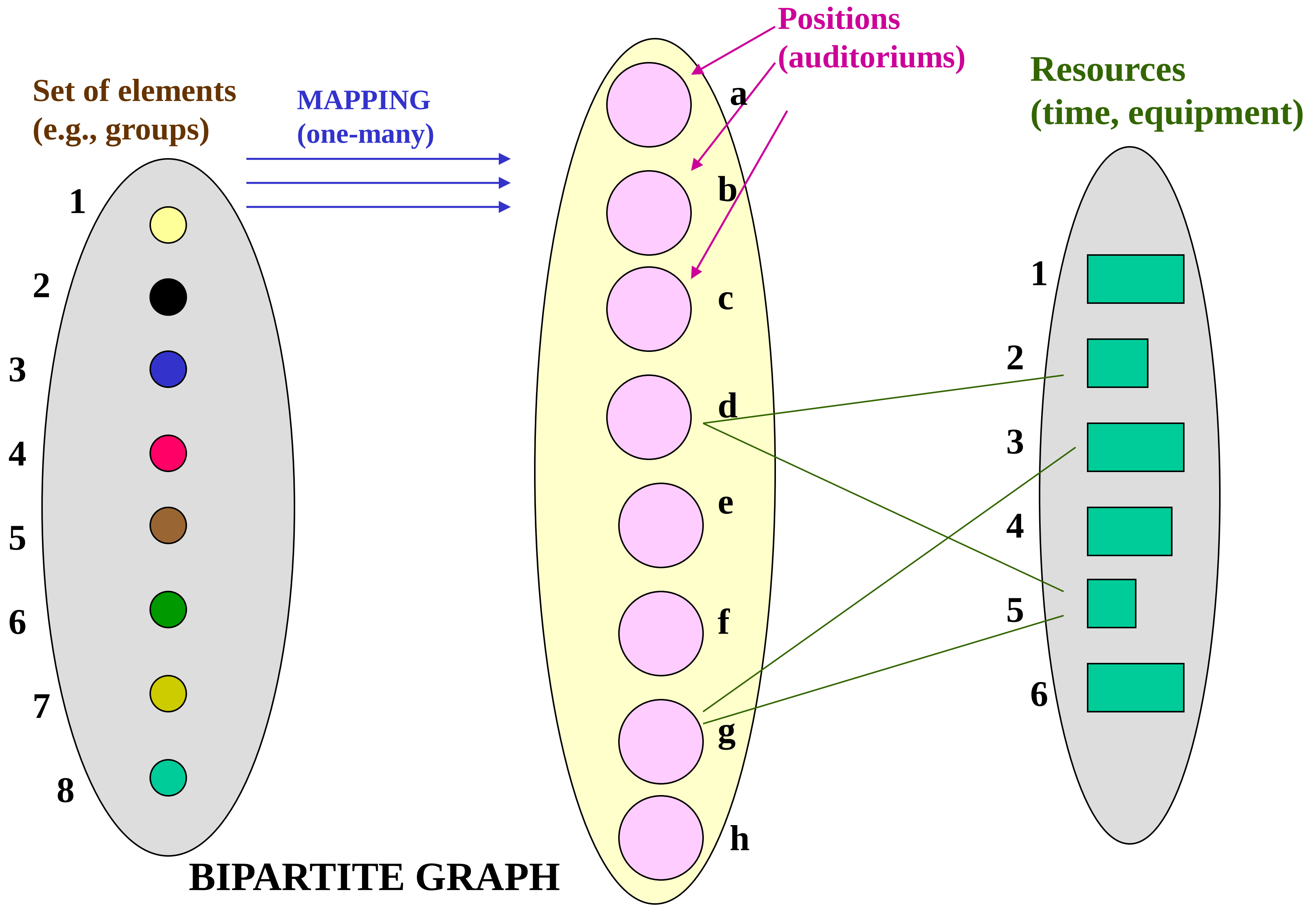
An assignment of elements { i | i∈{1,…,n} } to positions (locations) { p(i) }, where p is permutation of numbers {1,…,n }, a set of all possible permutations is ∏ = { p }.

Let us consider two n by n matrices:
(i)a "flow" (or "utility")matrix C whose (i,j) element represents the flow between elements (e.g., facilities) i and j, and
(ii)a distance matrix D whose (i,j) (p(i), p(j)) element represents the distance between locations p(i) and p(j).

With these definitions, the QAP can be written as

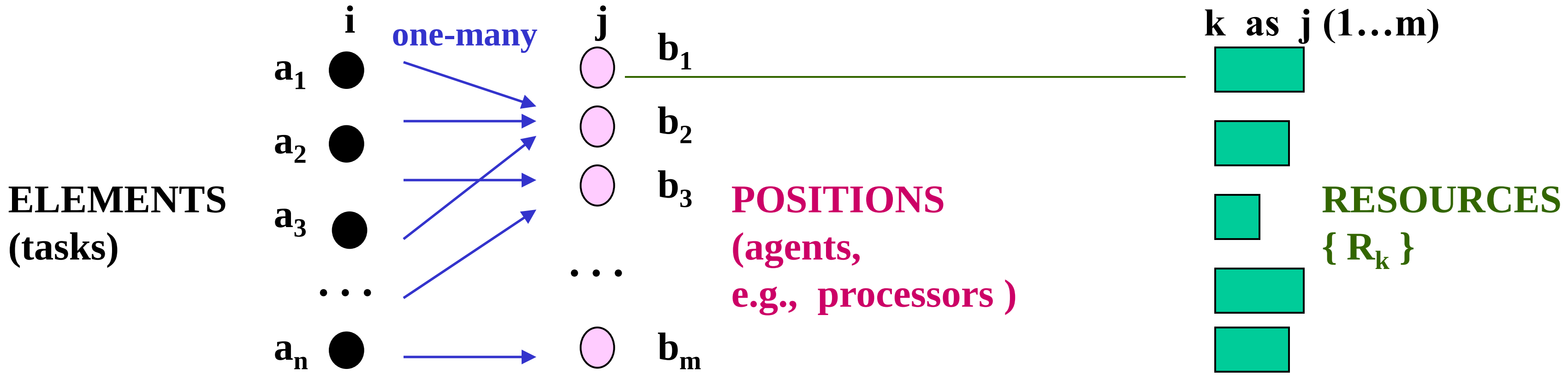
$$\max_{p \in \prod} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \, d_{p(i)p(j)}$$

Generalized Assignment Problem GAP (from lecture 30)



BIPARTITE GRAPH

Generalized Assignment Problem GAP (from lecture 30)



FORMULATION (algebraic):

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

Set of positions (e.g., locations, processors) $B = \{ b_1 , \dots , b_j , \dots . b_m \}$

(now let n = m)

Effectiveness of pair a_i **and** b_j **is:** $c (a_i , b_j)$

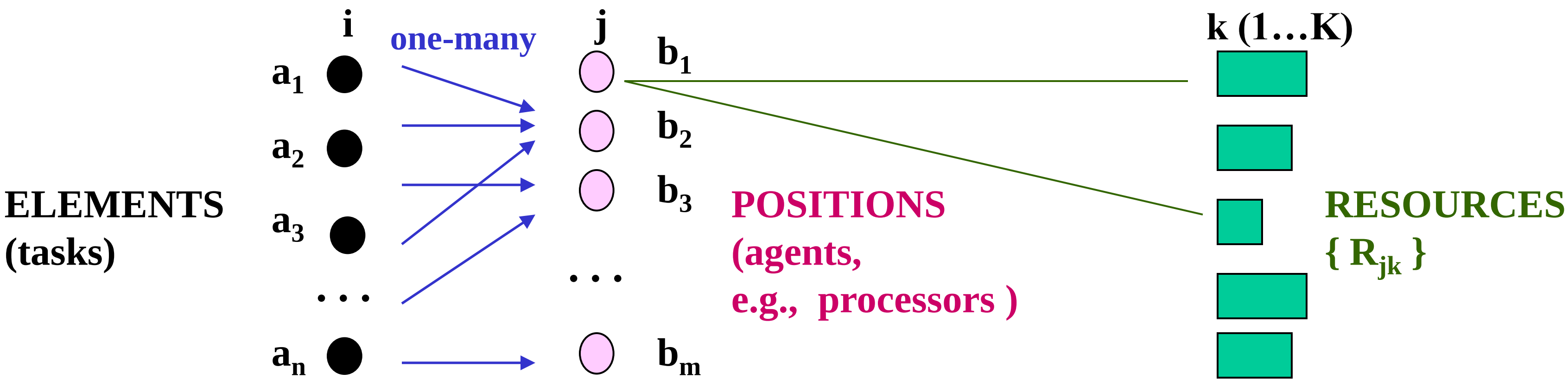
$x_{ij} =$ **1 if** a_i **is located into position** b_j **and 0 otherwise** $(x_{ij} \in \{ 0,1 \})$

The problem 1 is: $\max \sum^n_{i=1} \sum^n_{j=1} c_{ij} \; x_{ij}$

s.t. $\sum^n_{i=1} r_{ik} \; x_{ij} \leq R_j \quad \forall \; j$ **(** R_k **is resource of agent k)**

$\sum^n_{j=1} x_{ij} = 1 \quad \forall \; i$

Generalized Assignment Problem GAP (multiple common resources) (from lecture 30)



FORMULATION (algebraic):

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

Set of positions (e.g., locations, processors) $B = \{ b_1 , \dots , b_j , \dots . b_m \}$

(now let n = m)

Effectiveness of pair a_i **and** b_j **is:** $c (a_i , b_j)$

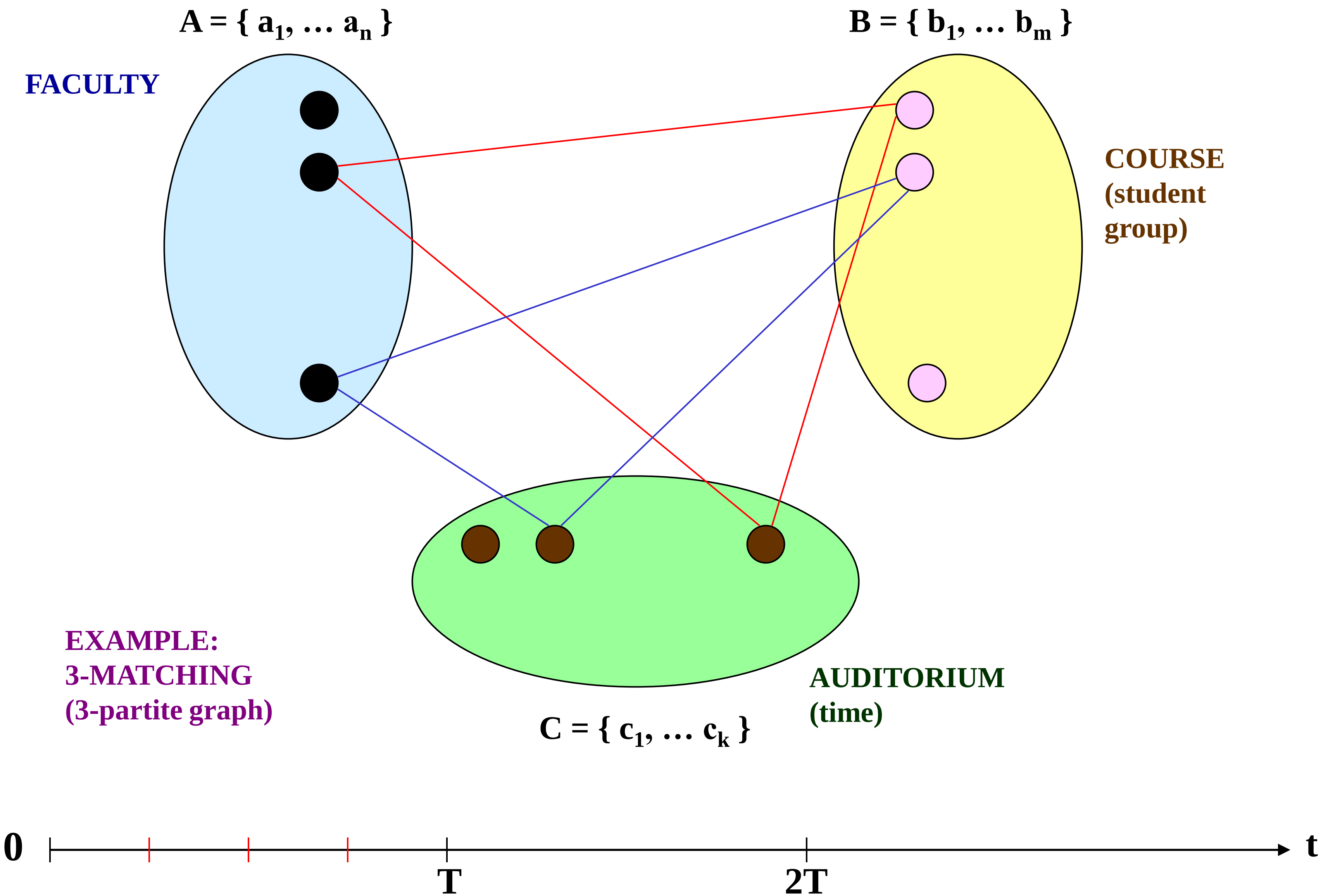
$x_{ij} =$ **1 if** a_i **is located into position** b_j **and 0 otherwise** $(x_{ij} \in \{ 0,1 \})$

The problem 2 is: $\max \sum^n_{i=1} \sum^n_{j=1} c_{ij} \; x_{ij}$

s.t. $\sum^m_{j=1} \sum^n_{i=1} r_{ik} \; x_{ij} \le R_k \quad \forall \; k$ **(common K resources)**

$\sum^n_{j=1} x_{ij} = 1 \quad \forall \; i$

Timetabling problem: PLUS time (cycles)



CONSTRAINTS (examples):

FOR LECTURERS:

Lecturer 2 can teach only on Monday and Friday (lecturers - time)

Lecture 11 must be after Lecturer 12 (over lecturers)

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

FOR STUDENT GROUPS:

Group 1 needs auditorium 5 on Monday morning (groups - time)

Groups 7, 8, and 9 must have the same Lecturer 1 (the same course)
(groups - lecturers) * (groups - groups)

Group 4 prefers Lecturer 10 (groups - lecturers)

Group 5 needs Lecturer 5 & Lecturer 8 as time neighbors (groups - lecturers)

FOR AUDITORIUMS:

Auditorium 1 is maintained on Wednesday (closed) (auditoriums - time)

Auditorium 4 corresponds only for groups 5, 7, 8, and 9 (auditoriums - groups)

TYPES OF CONSTRAINTS (by elements):

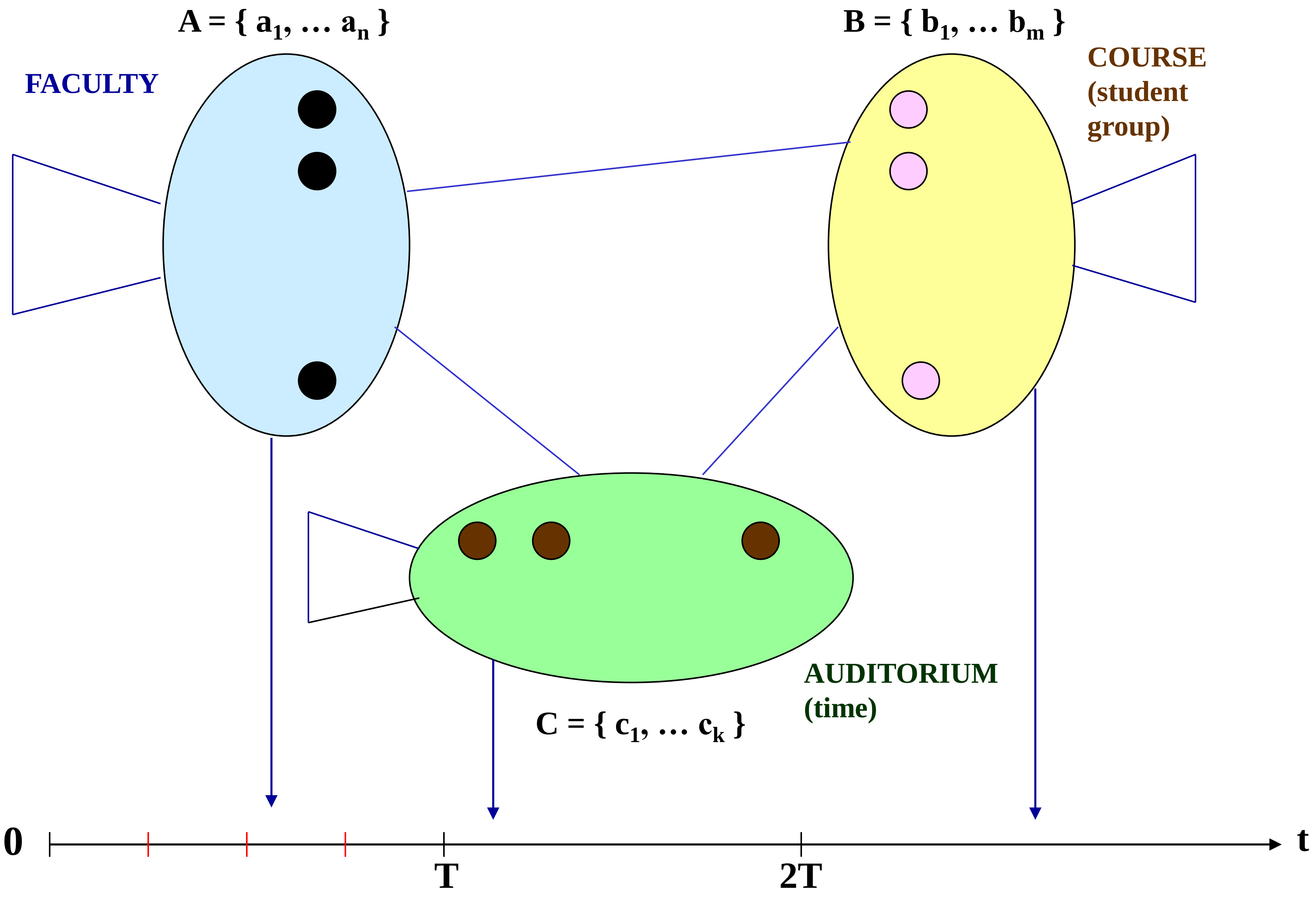
Lecturers – lecturers
Lecturers - groups
Lecturers-auditoriums
Lecturers - time

Groups - groups
Groups – auditoriums
Groups - time

Auditoriums – auditoriums
Auditoriums - time

TYPES OF CONSTRAINTS (by kind):
Logical (e.g., binary relations)
Quantitative (e.g., resource constraints)

Timetabling problem: illustration for constraints



Prospective application domains for timetabling problems

1.SCHEDULING IN COMMUNICATION SYSTEMS
AS COMMUNICATION TIMETABLING

2.SCHEDULING IN MONITORIG SYSTEMS
AS MONITORING TIMETBLING

Timetabling: sport example

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

FORMULATION, DATA & CONSTRAINS:

1.Basic objects: teams, slots (weekday slots, weekend slots),
each team plays twice in a week,8 home slots, 8 away slots
2. Constraints for teams and slots: home slots, away slots, chain: home-away-etc.
3. Patterns and constraints: <= 2 away games, <= 2 home games
4.Team paring constraints: candidates of team pair (from previous schedule)

SOLVING SCHEME:

STEP 1. A pattern is a string of H (home), A (away), and B (bye).
Examples: HAA, HBW
To find a set of pattern, example: HHA, AHA, HAH, AAH (teams a, b, c, d)
(enumeration & integer programming)
STEP 2. To assign games to the patterns. This is timetabling
(integer programming)
STEP 3. To assign teams to the patterns. This with patterns gives schedule
(quadratic assignment problem)

COMPUTING: 24 hours