Politecnico di Milano
Dip. Elettronica e Informazione
Milan, Italy

Bottlenecks Identification in “Very Large” Multiclass Queueing Models

G.Casale – G.Serazzi

serazzi@elet.polimi.it

CS-UCDavis
Aug.11 2006

Outline

- ❑ motivations
- ❑ multiclass pitfalls
- ❑ networks with multiple bottlenecks
- ❑ “very large” models
- ❑ conclusions

Motivations

- complexity of modelling actual computer infrastructures
  - large installations comprising thousand of servers
  - strongly multiclass workload
  - ➢ “very large” models (VLM)

- emerging distributed technologies and applications
  - peer-to-peer technologies
    - collaboration, middlewares, file sharing, games, messaging, …
  - grid computing
  - web services
    - interoperability
  - wireless and ubiquitous computing
  - …

“Very large” models (VLM)

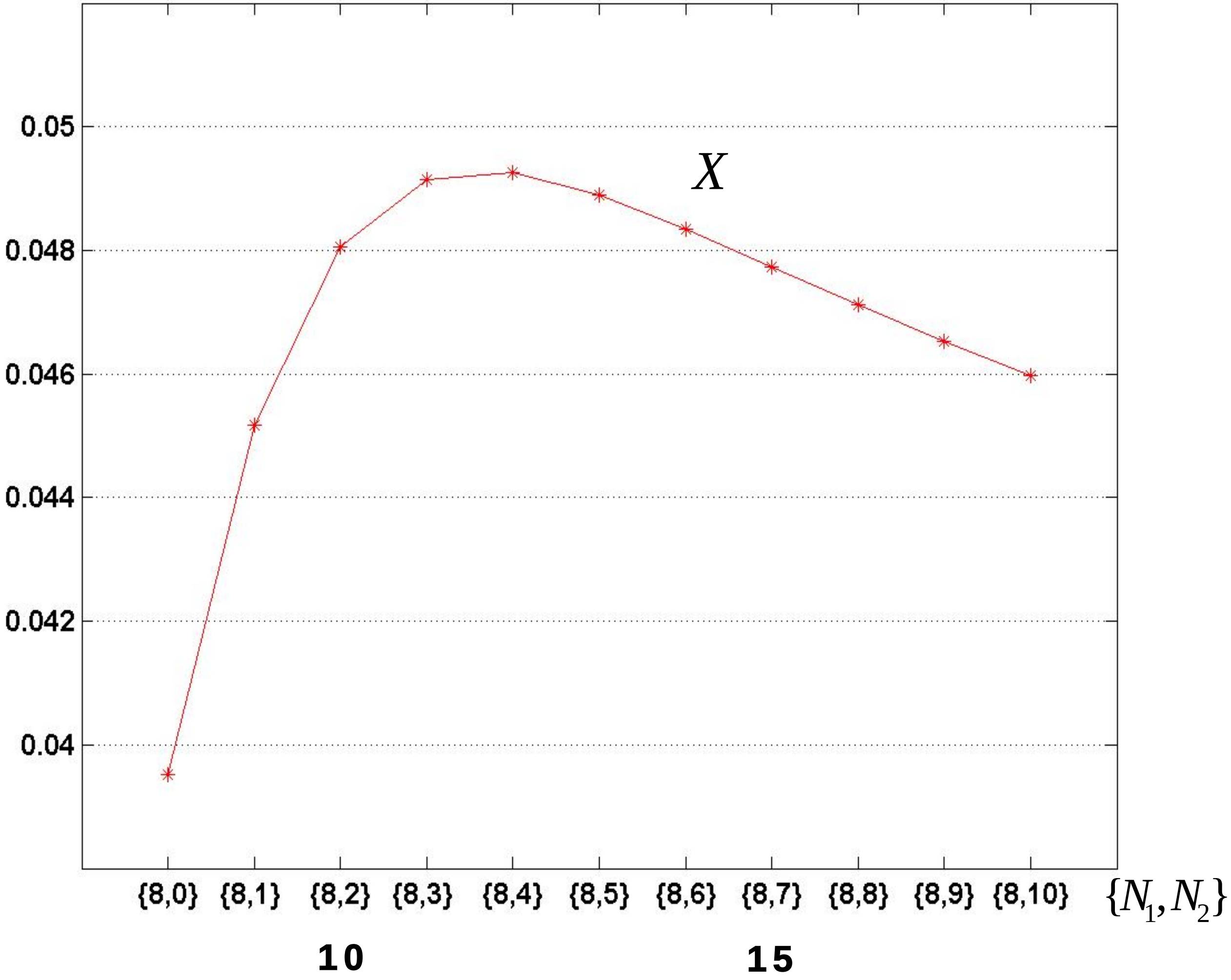
- ❑ Intel (2001)
  - ❑ 100000 clients
  - ❑ 3000 servers
- ❑ Vodaphone Italy (2004)
  - ❑ 500 server Sun, 400 server HP, 2000 server NT
  - ❑ 40 Millions/day of SMS, 20 Millions customers
  - ❑ 500 update/sec on the customer care DB
- ❑ Unicredit bank (2004)
  - ❑ 10 large mainframes
  - ❑ 1000-1500 servers
  - ❑ Transactions: 36 Millions/day

Multiclass pitfalls

- "the throughput X of the network increases as the total number of customers N increases"
- "the sum of the utilizations $U_{TOT} = \sum_{i=1}^{M} U_i$ of the stations increases with N"
- "for a given population, only one bottleneck station exists in the network"

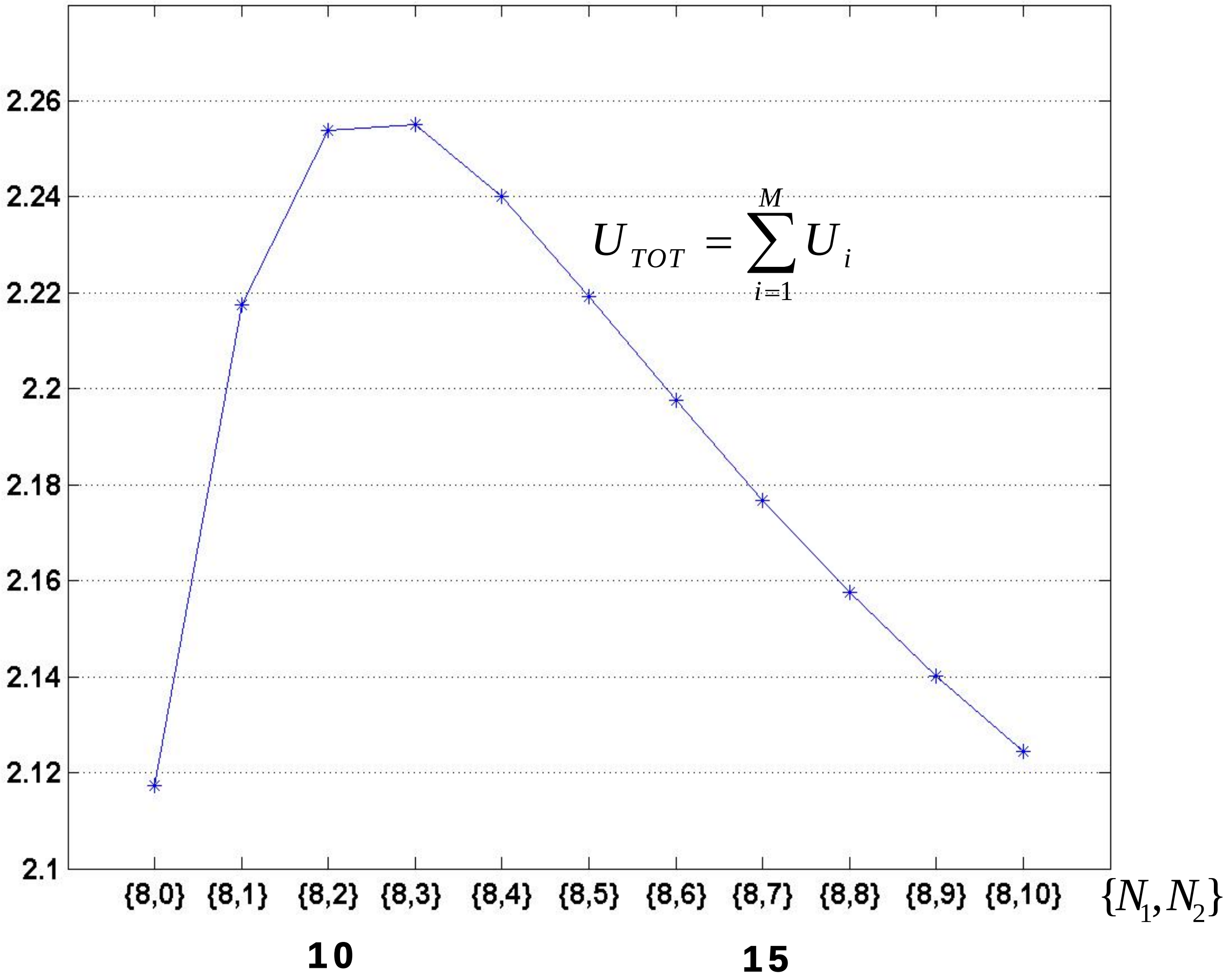
X increases as N

$$\begin{array}{cc} & \text{Classes} \\ \text{Stations} & \begin{bmatrix} 25 & 5 \\ 15 & 30 \\ 10 & 15 \end{bmatrix} \end{array}$$



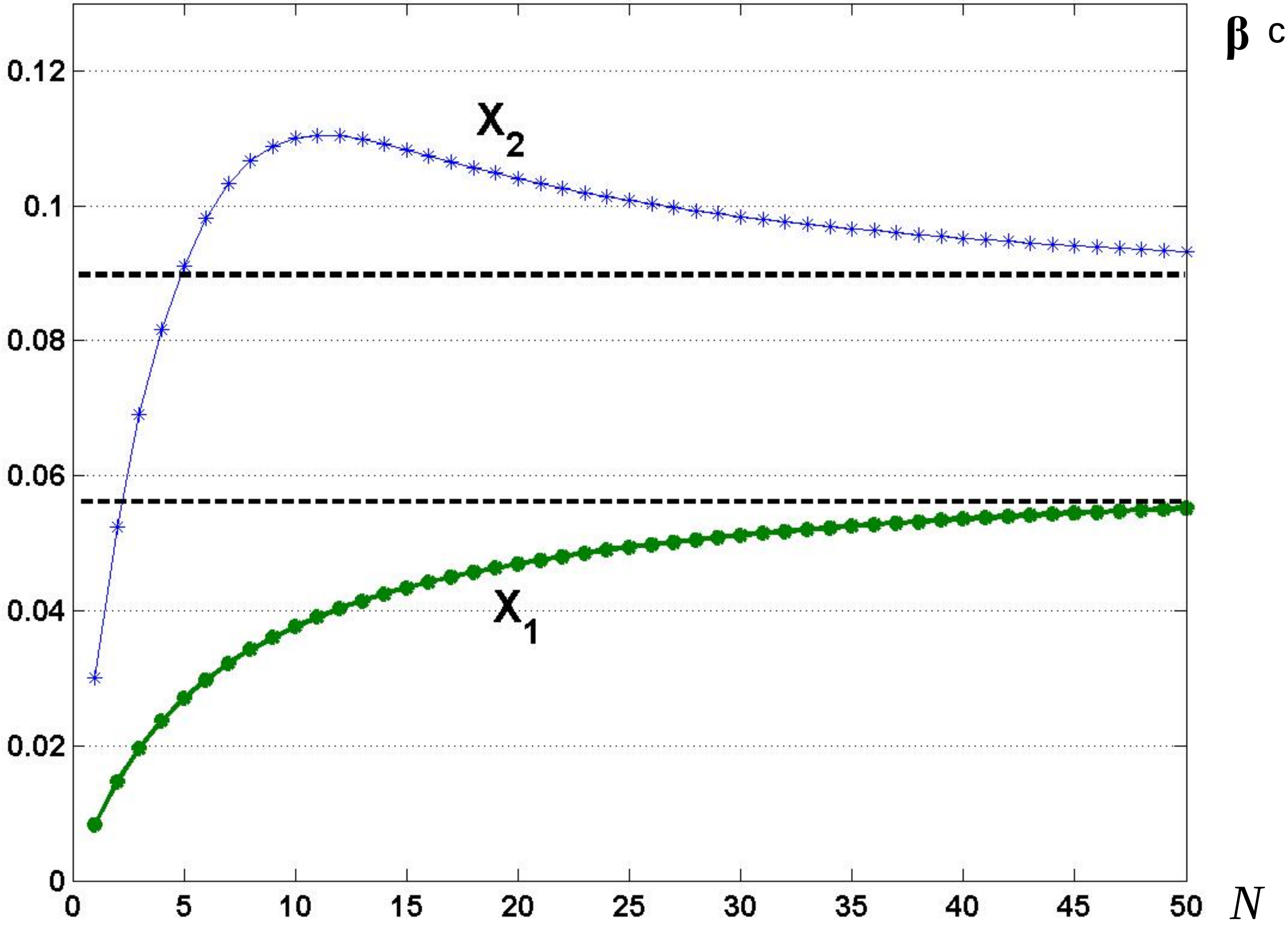
Global utilization increases as N

$$\begin{array}{cc} & \textit{Classes} \\ \textit{Stations} & \begin{bmatrix} 60 & 25 \\ 55 & 90 \\ 24 & 50 \end{bmatrix} \end{array}$$



$$U_{TOT} = \sum_{i=1}^{M} U_i$$

Asymptotic Performance Indices



β constant

Notation

- ❑ loading matrix

$$\mathbf{L} = \{L_{mr} = V_{mr} S_{mr}\}$$

$$\begin{array}{cc} & \textit{Customer Classes} \\ \textit{Queueing Stations} & \begin{bmatrix} L_{11} & L_{12} & \dots & L_{1R} \\ L_{21} & L_{22} & \dots & L_{2R} \\ \dots & \dots & \dots & \dots \\ L_{M1} & \dots & \dots & L_{MR} \end{bmatrix} \end{array}$$

- ❑ β (% of jobs per-class)

$$\beta_r = \frac{N_r}{\sum_{r=1}^{R} N_r} = \frac{N_r}{N}$$

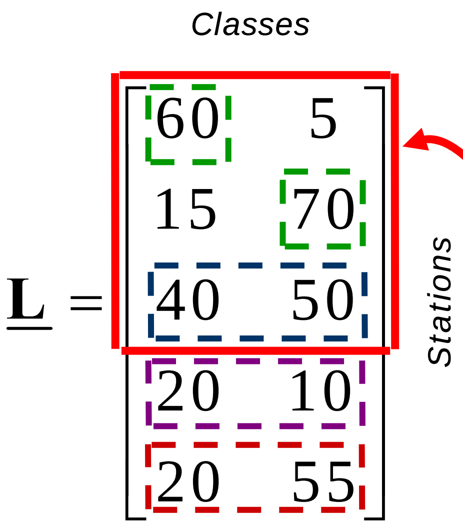
- ❑ population mix

$$\boldsymbol{\beta} = \{\beta_1, \beta_2, \dots, \beta_R\} \ , \quad \sum_{r=1}^{R} \beta_r = 1$$

- ❑ population vector

$$\mathbf{N} = \{N_1, N_2, \dots, N_R\} = N \cdot \boldsymbol{\beta}$$

Taxonomy of stations



$$\underline{\mathbf{L}} = \begin{bmatrix} 60 & 5 \\ 15 & 70 \\ 40 & 50 \\ 20 & 10 \\ 20 & 55 \end{bmatrix}$$

- ❑ Natural bottlenecks

  $\beta_1=100\% \;\; \beta_2 = 0\% \;\; \rightarrow U_{\max} = U_1$

  $\beta_1= 0\% \;\; \beta_2 = 100\% \rightarrow U_{\max} = U_2$

- ❑ Network bottlenecks

  $\beta_1= 50\%, \;\; \beta_2 = 50\% \rightarrow U_{\max} = U_3$

- ❑ Potential bottlenecks set

  (network + natural) bottlenecks

  Π = {1 , 2 , 3}

- ❑ Dominated stations

  4 has all components less than those of 3

- ❑ Masked-off stations

  5 not dominated, but never saturates

Types of population growth

- ❑ unbalanced population growth
  - ❑ only the customers of one class increases to infinity (degenerate to a single class case)

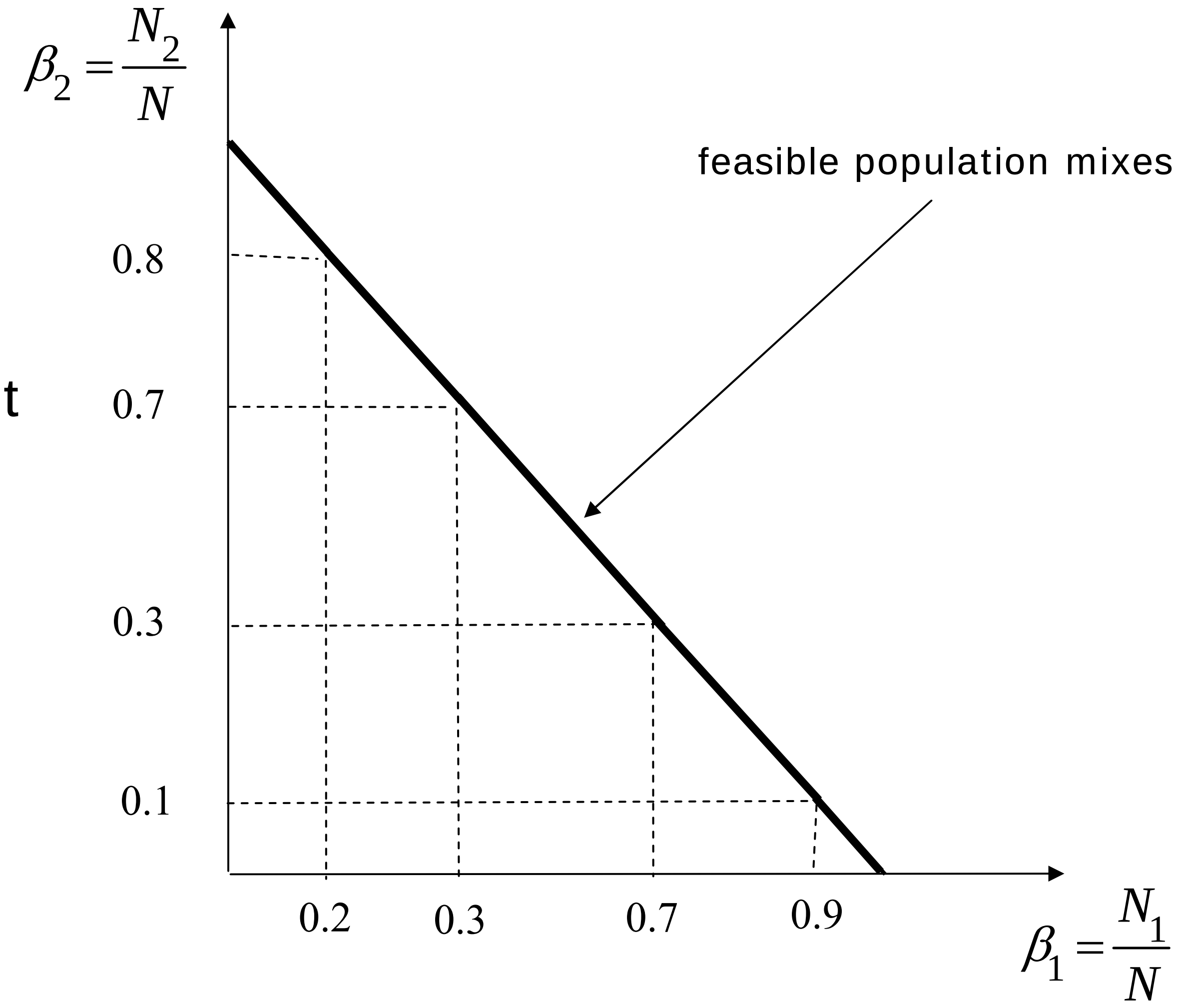
- ❑ proportional population growth
  - ❑ the customers grow to infinity keeping constant the population mix β (or the arrival rate mix λ)

e.g., $\beta_1 = 50\%,\ \ \beta_2 = 50\% \rightarrow (10,10)\ ,\ (30,30)\ ,\ (100,100)$

proportional pop. growth (closed model)

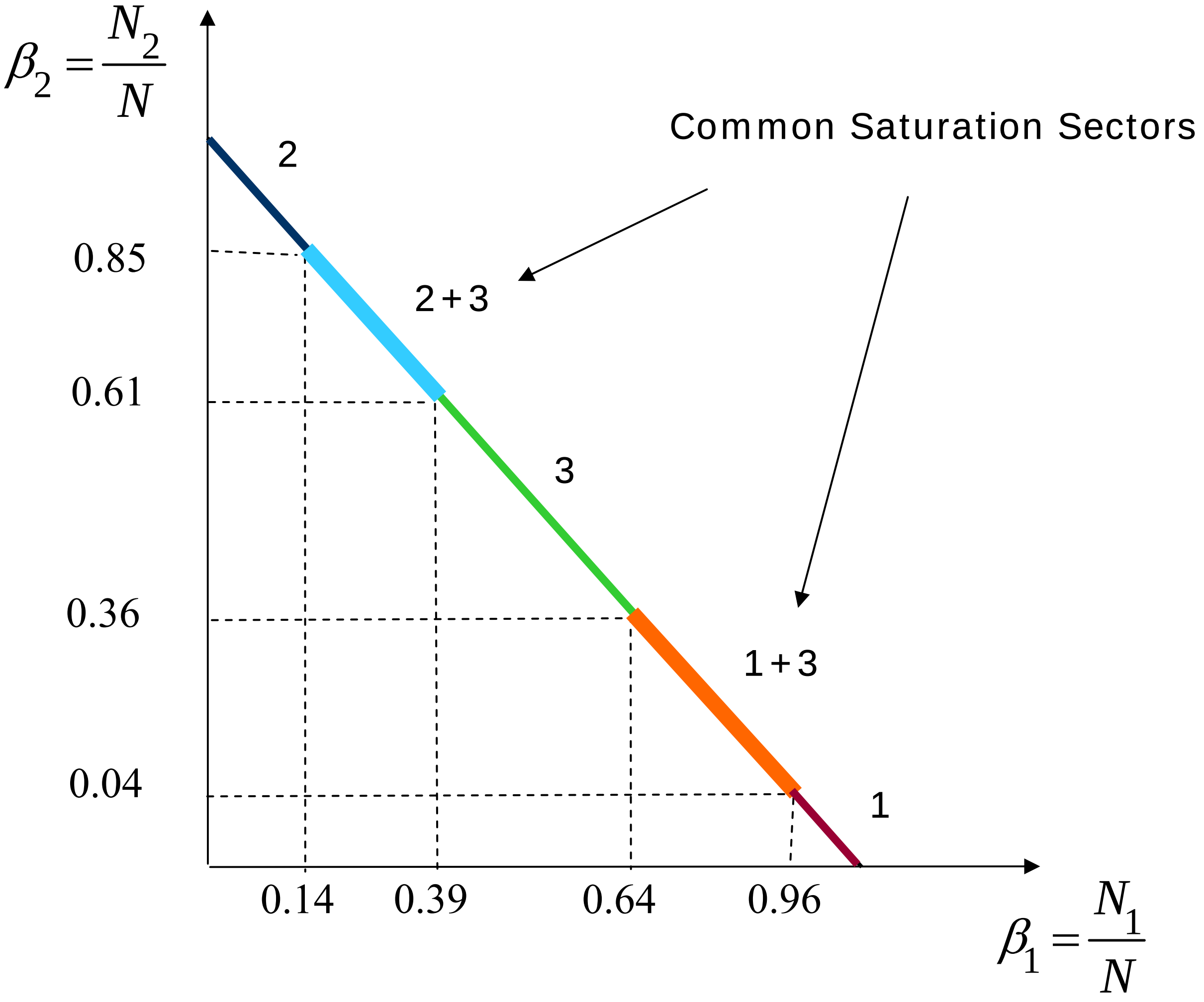
N constant

β variable



Bottlenecks in closed models (2 classes)

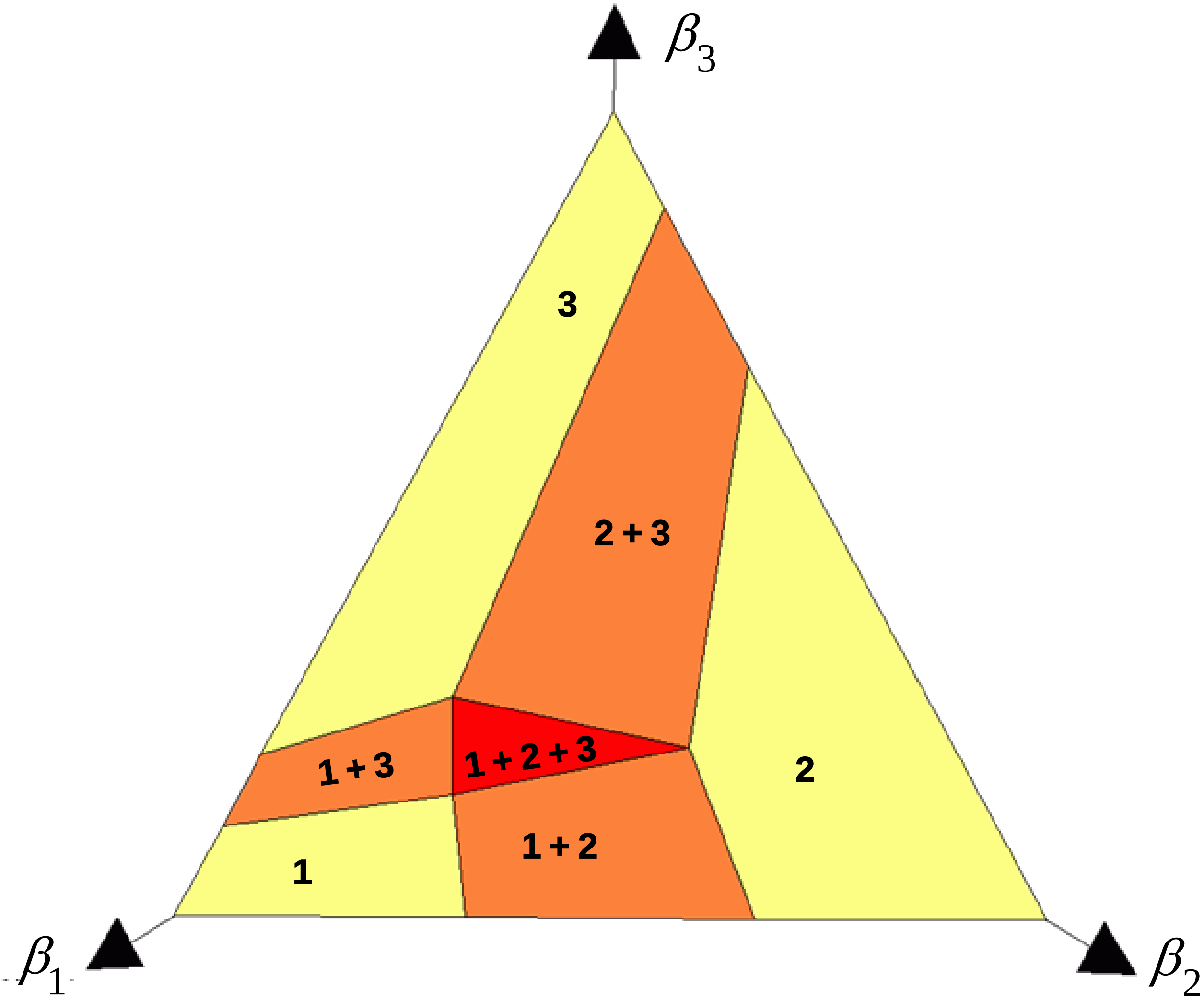
$$\begin{array}{cc} & \textit{Classes} \\ \textit{Stations} & \begin{bmatrix} 60 & 5 \\ 15 & 70 \\ 40 & 50 \end{bmatrix} \end{array}$$



Common Saturation Sectors

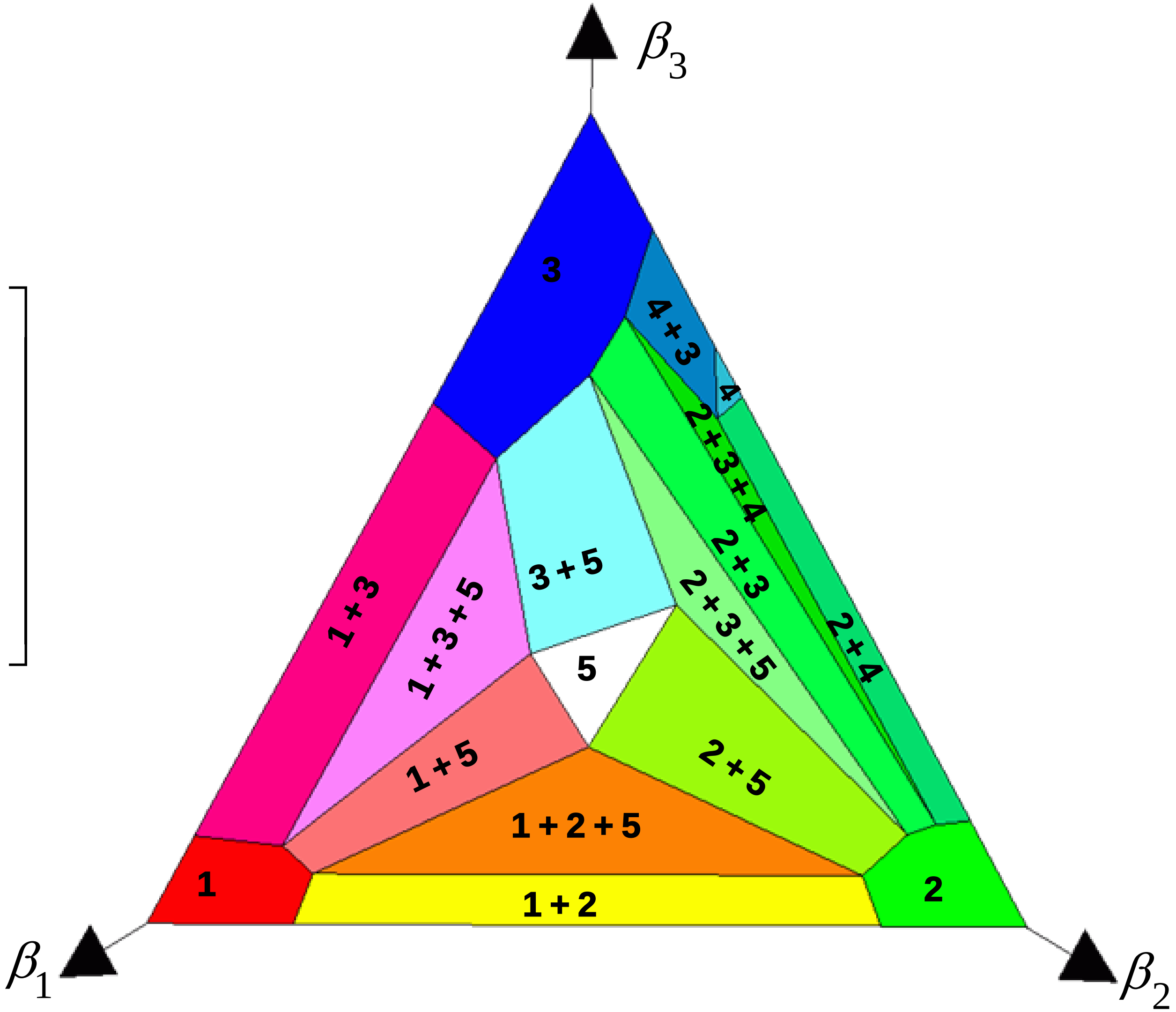
Bottlenecks in closed models (3 classes)

$$\begin{array}{cc} & \textit{Classes} \\ \textit{Stations} & \begin{bmatrix} \boxed{100} & 40 & 50 \\ 50 & \boxed{80} & 70 \\ 90 & 30 & \boxed{90} \end{bmatrix} \end{array}$$



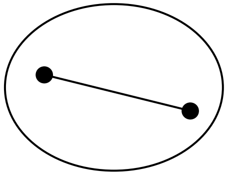
Complex saturation behavior

$$\begin{array}{c|ccc} & & \textit{Classes} & \\ \hline \textit{Stations} & \begin{bmatrix} 50 & 10 & 10 \\ 10 & 50 & 10 \\ 20 & 10 & 60 \\ 10 & 20 & 50 \\ 25 & 25 & 35 \end{bmatrix} \end{array}$$

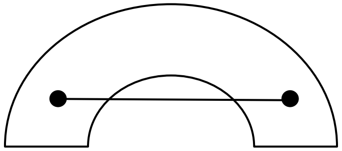


The convex hull problem

- **Convex set**
  - Every line segment joining any pair of points lies entirely in the set



Convex set

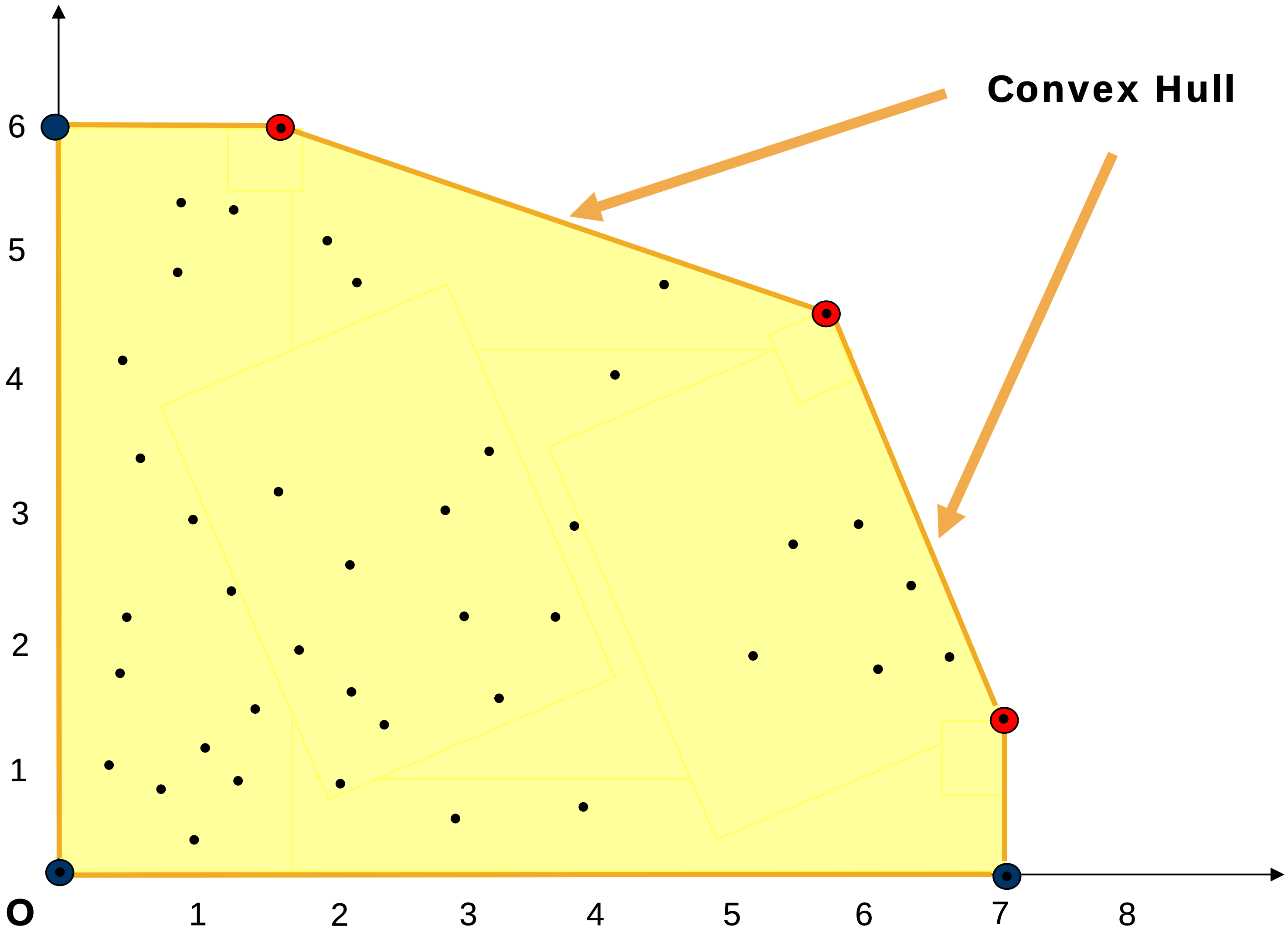


Non-convex set

- **Convex hull problem**: find the smallest convex set containing a given set of points

- Fast algorithms in 2D [O(M logM)] and in 3D [O(M^2)] exist (M number of points)

Convex hull in 2 dimensions

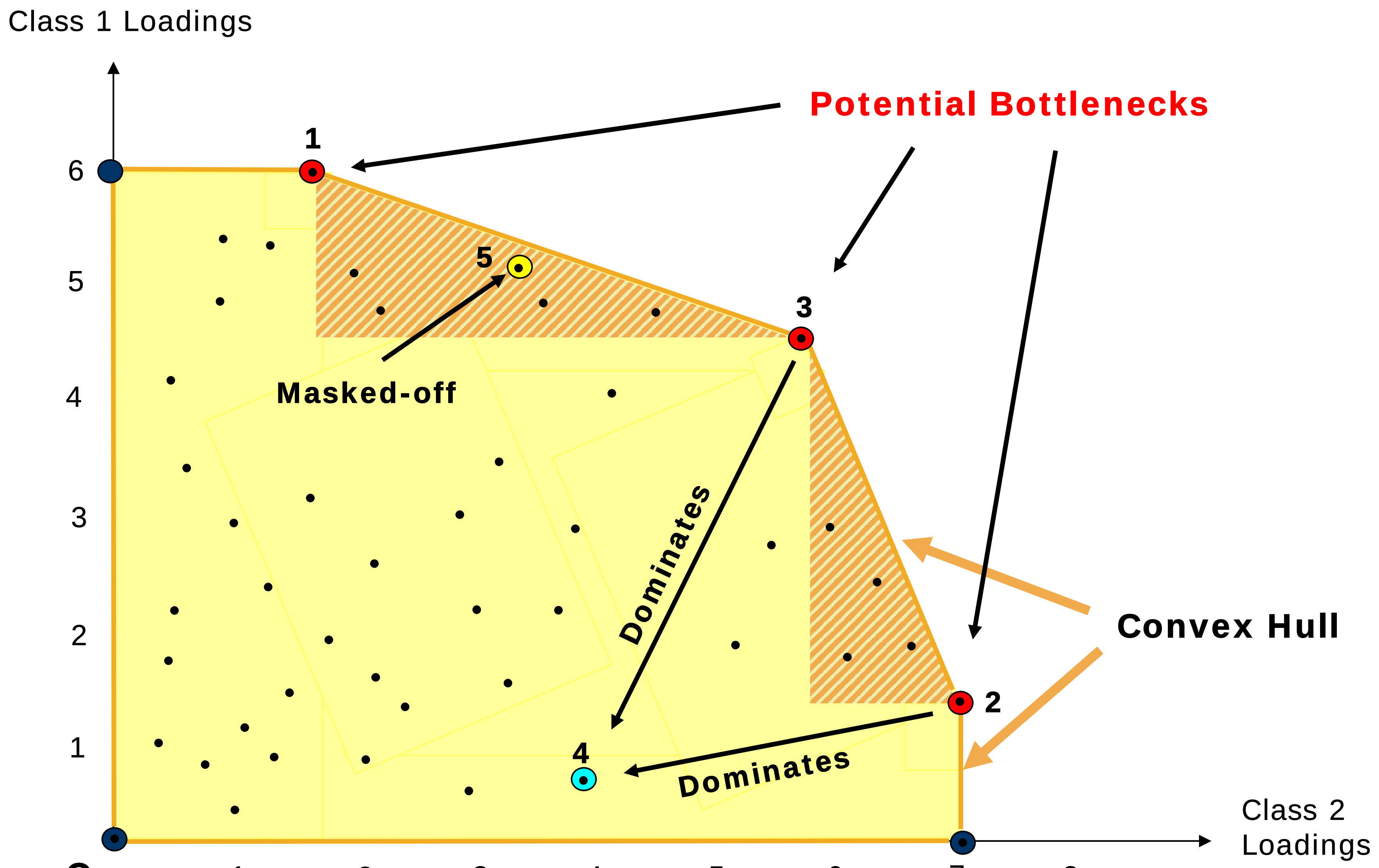


Potential Bottleneck Identification

- Use of computational geometry techniques for the queueing network analysis
- Define the loading matrix L
- Solve the **convex hull problem**
  - Gives knowledge on the sets of stations that can saturate together (common saturation sectors)
  - Allows fast computation of the bottlenecks set as a function of the population mix

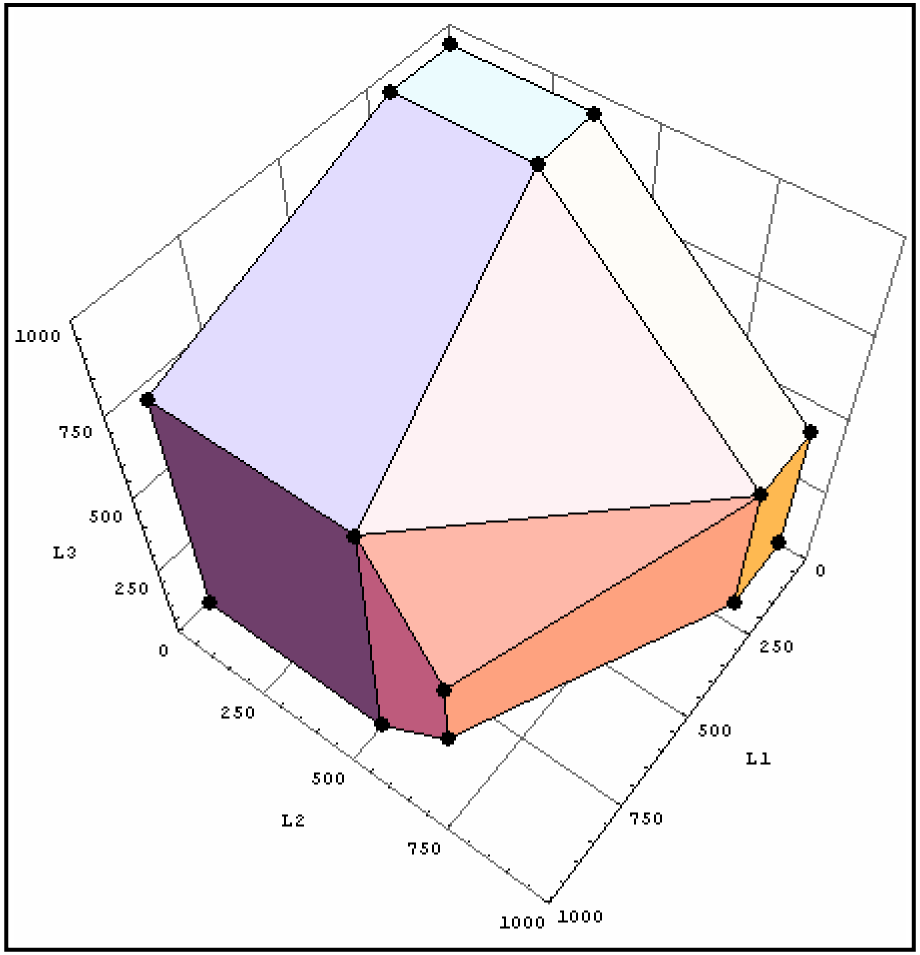
Potential Bottleneck Identification

Convex hull of the loading matrix



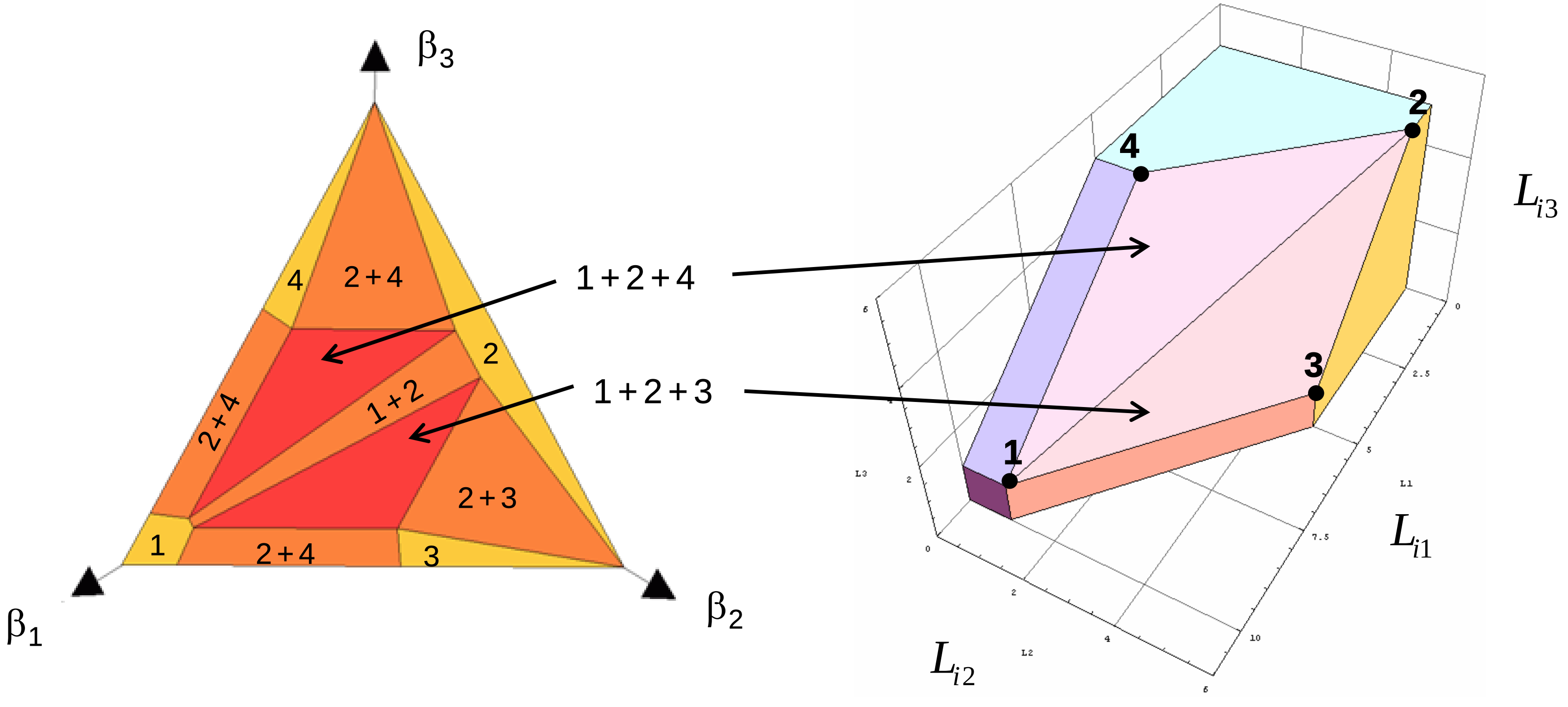
Potential Bottleneck Identification

Convex hull of a 3-class model



Saturation sectors and convex hulls

- The convex hull of the loading matrix and the beta space are closely related



- **3+4** and **1+3+4** are not saturation sectors because there is no edge between **3** and **4** on the convex hull
  - allows fast computation of the saturation sectors

Redundancy elimination

- The time complexity of the convex hull is exponential in the number of classes

CONVEX HULL CPU TIME	R=3	R=6	R=7	R=8	R=9
M=1000	<0.1 s	1 s	32 s	161 s	>1day
M=10000	<0.1 s	21 s	200 s	>1day	>1day
M=100000	0.12 s	100 s	>1day	>1day	>1day
M=1000000	72 s	463 s	>1day	>1day	>1day

Tested on a AMD Athlon 2800XP+ - 256KB CACHE – 768Mb RAM

- Redundancy elimination techniques instead of convex hulls
  - Polynomial time complexity in the number of stations M and in the number of classes R
  - Loose information on the set of stations saturating together (indetifies only the potential bottleneck set)

Potential bottenecks Identification

Experimental results

Redundancy Elimination CPU TIME	R=5	R=10	R=25	R=50	R=100
M=1000	4 secs	6 secs	15 secs	48 secs	80 secs
M=10000	2 minutes	4 minutes	10 minutes	31 minutes	66 minutes
M=100000	5 hours	7 hours	9 hours	16 hours	34 hours

Tested on a Intel Xeon Dual Processor 2.80 Ghz – 512KB CACHE – 1Gb RAM

- Redundancy Elimination is formulated as a set of independent problems
  - parallelization: grid computing, clusters, …

- Heuristic strategies for quick identification of dominated and masked-off stations are available

Fast computation of the saturation sectors

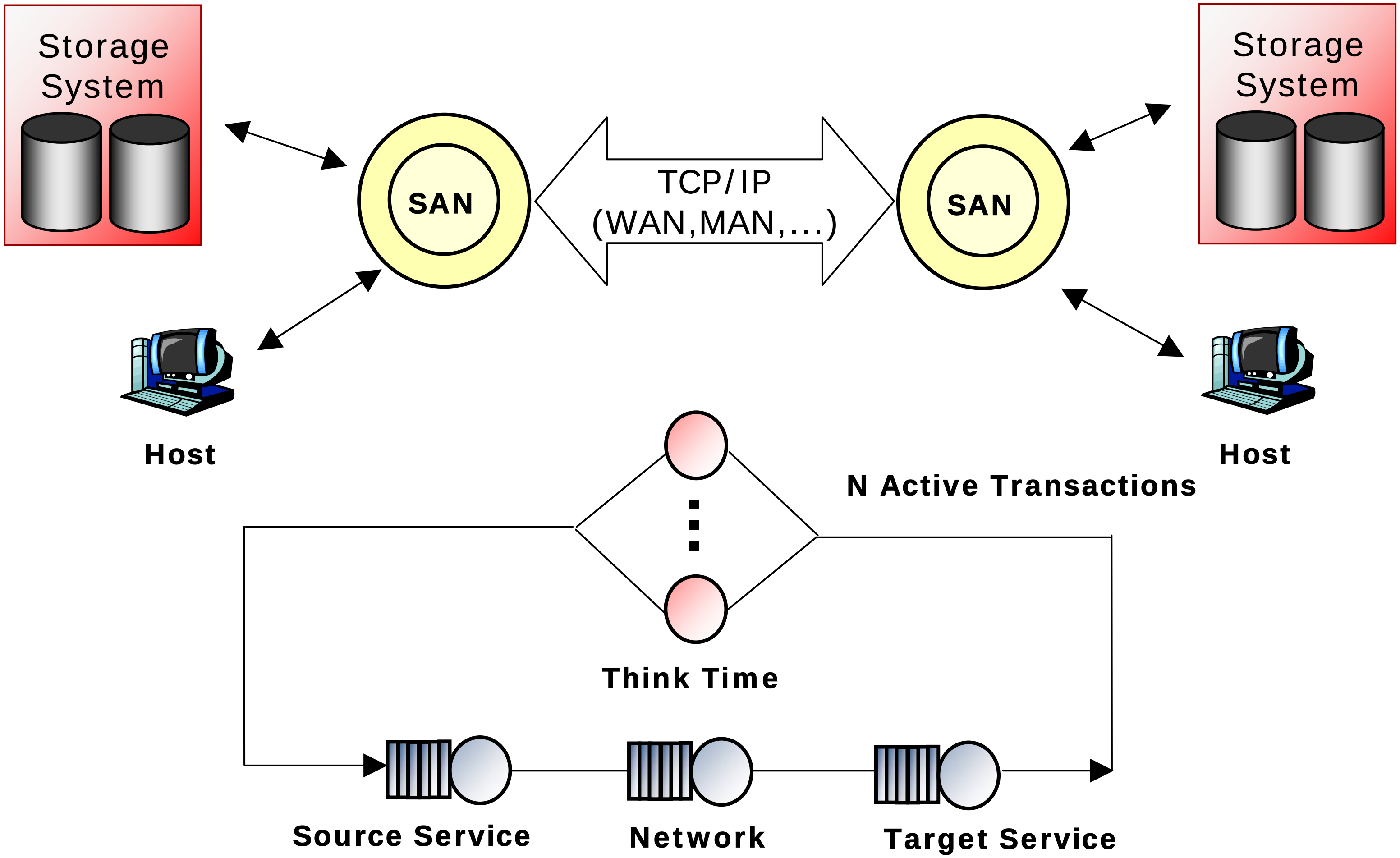
- “Old” computation scheme: brute force approach, explore all possible sets of saturating stations. Requires the solution of a large number of linear systems to identify the ♌s of the saturation sectors edges
- “New” computation scheme: use convex hull informations to solve only a minimal set of linear systems

	3 customer classes			4 customer classes		
	Old	**New**	👎	**Old**	**New**	👎
M=10 stations	108.72	42.50	**-60.56%**	297.78	114.82	**-61.44%**
M=100	1914.88	173.06	**-90.96%**	33299.83	1123.02	**-99.66%**
M=1000	9712.82	390.56	**-95.98%**	384068.5	3954.01	**-98.97%**

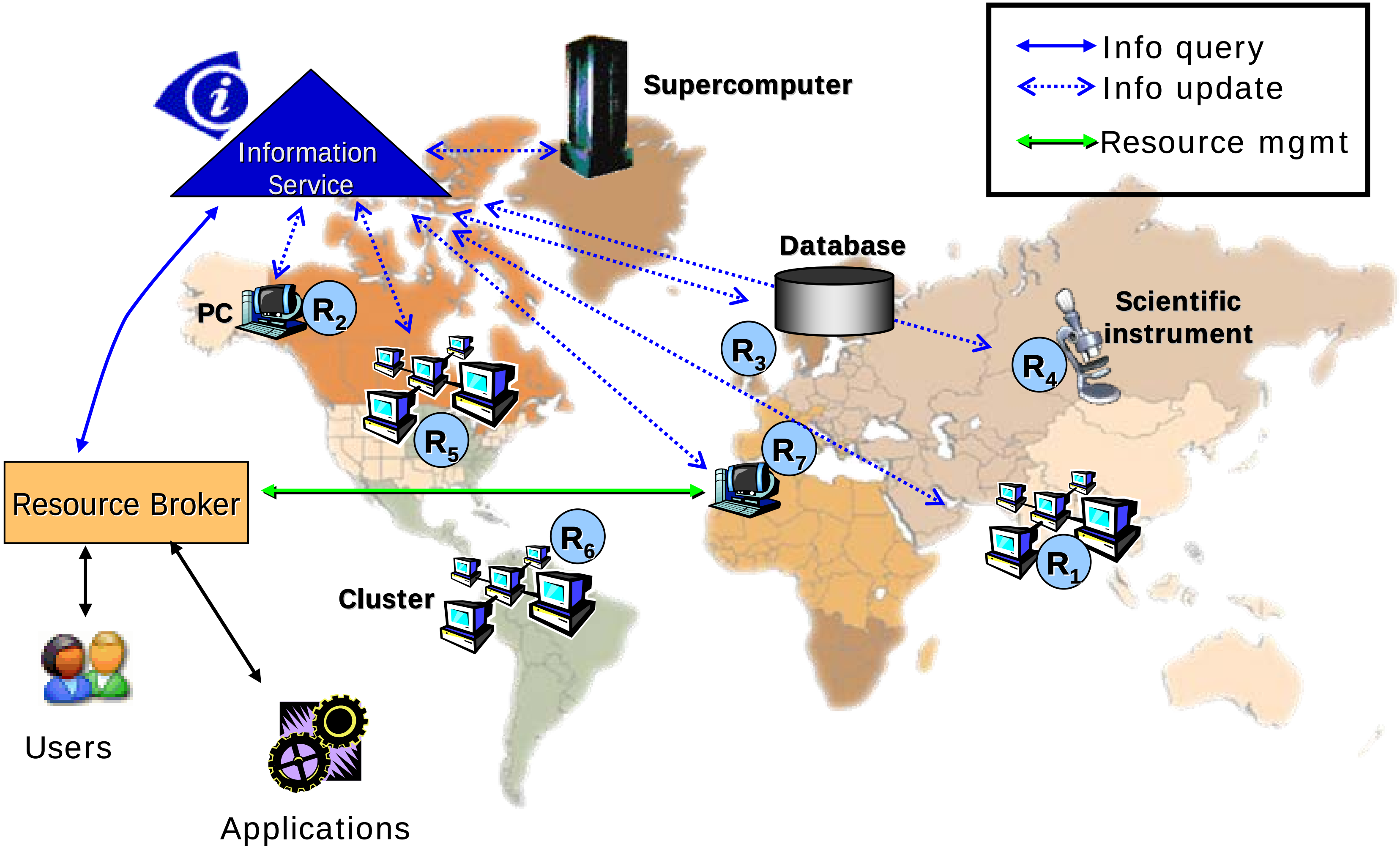
Mean number of linear systems to be solved to
identify the saturation sectors (tested on 100 random models)

Applications
Remote Storage Area Networks (SAN) Replication

- Experimental studies report accurate performance modelling of remote SAN environments

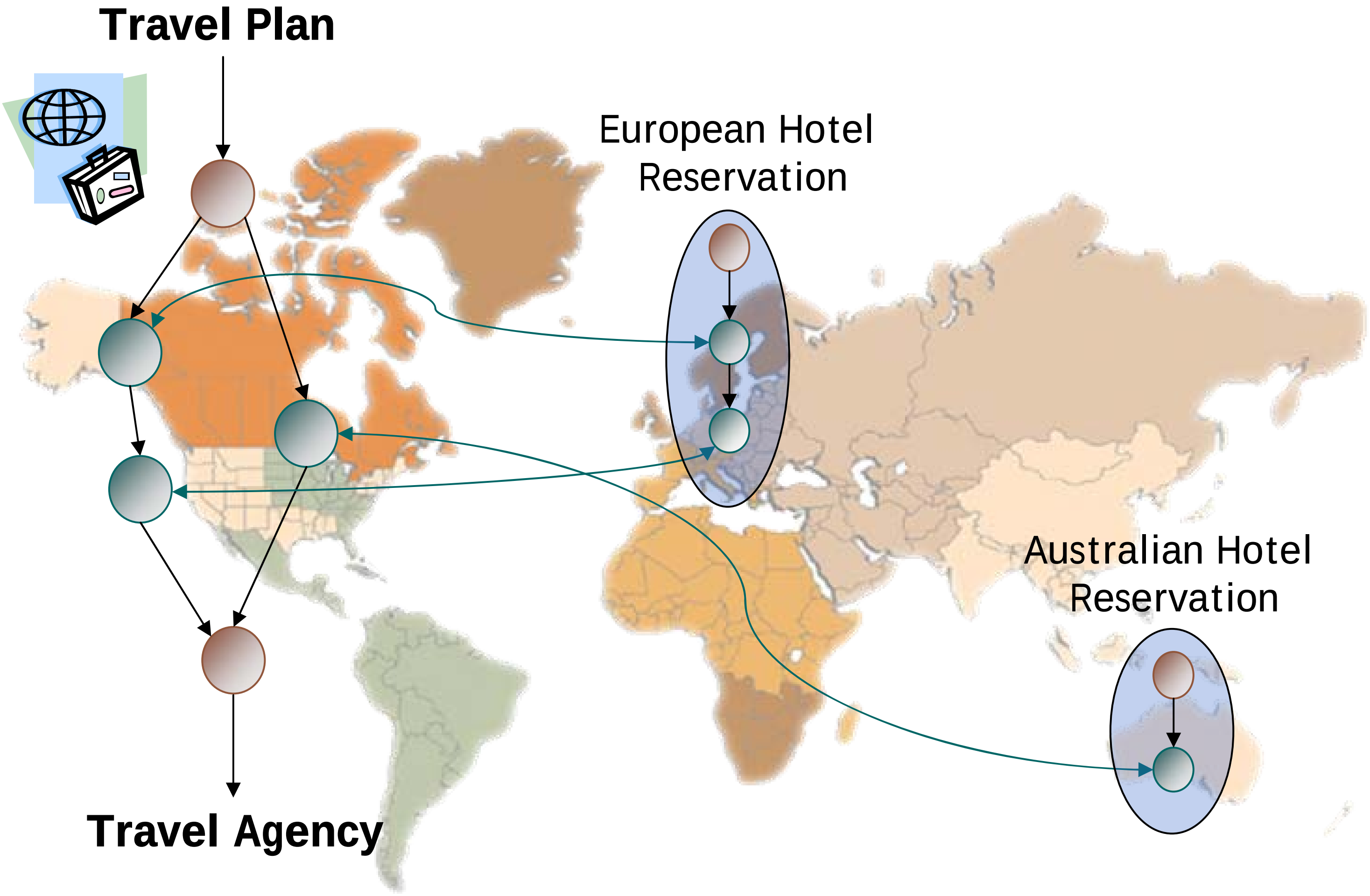


Applications
Grid computing



- Several issues: optimal scheduling, performance forecasting, …

Applications
Web services



- ❑ Each workflow requires cooperation with several different services
- ❑ Several queues can be experienced
- ❑ Strongly multiclass workload (different types of services)

Applications
Miscellanea

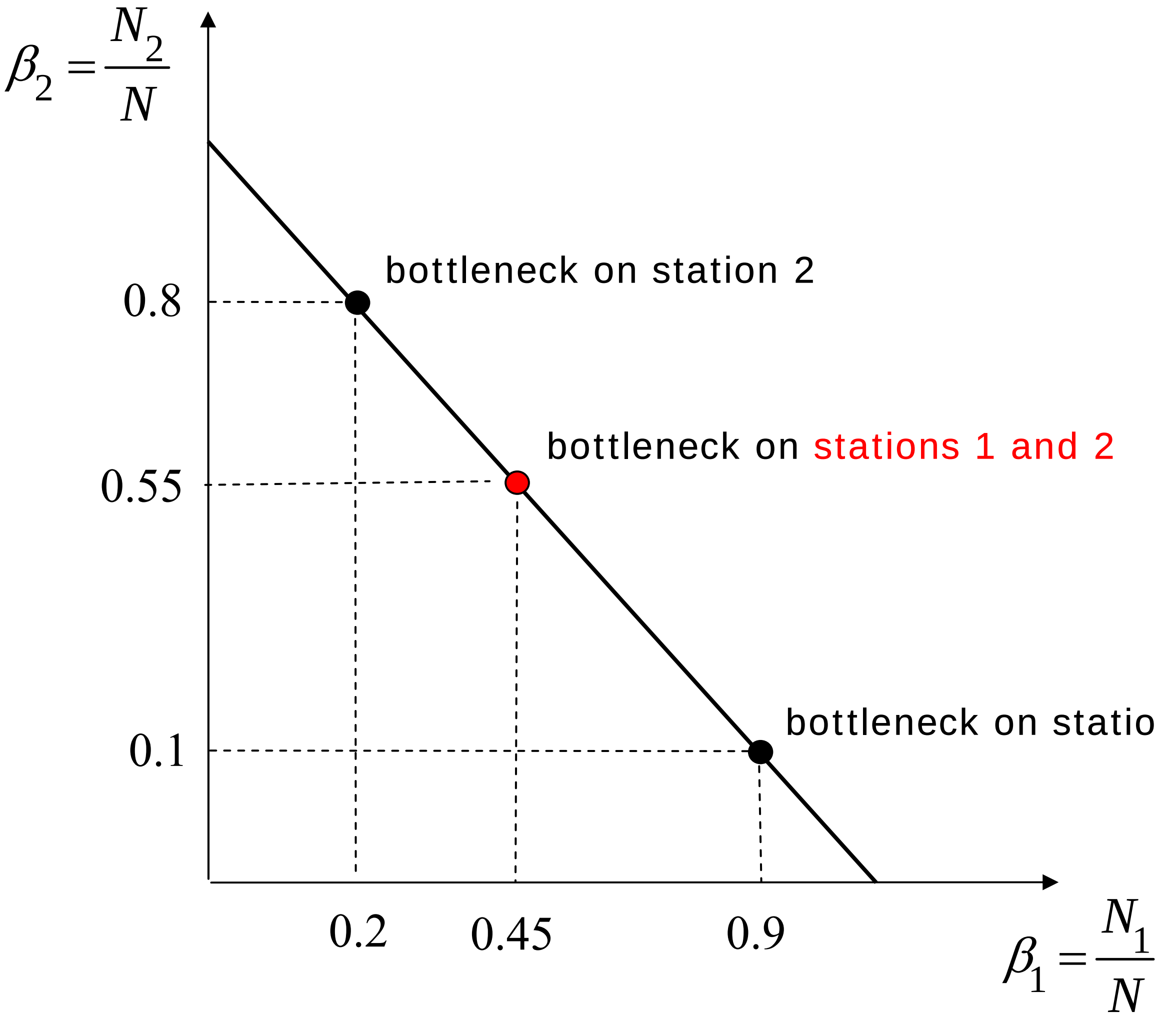
- Web caching and Web replication
  - proxy TCP connection caching models
  - content-aware load balancing
  - content delivery networks (CDN)
- Massively Multiplayer Games
  - scalability issues and load balacing
- P2P File sharing
- …

Bibliography

- Balbo G., Serazzi G., *Asymptotic analysis of multiclass queueing networks: common bottlenecks*. Performance Evaluation, 26: pp. 51-72 (1996)
- Balbo G., Serazzi G., *Asymptotic analysis of multiclass queueing networks: multiple bottlenecks*. Performance Evaluation, 30: pp. 115-152 (1997)
- Litoiu M., Rolia J. A., Serazzi G.: *Designing Process Replication and Activation: A Quantitative Approach*. IEEE Trans. Software Eng. 26(12): 1168-1178 (2000)
- Rosti E., Schiavoni F., Serazzi G., *Queueing network models with two classes of customers*, Proc. of Mascots 97, pp 229-234, Israel (1997)
- Casale G., Serazzi G., *Estimating Bottlenecks of Very Large Models*, In Performance Evaluation: Stories and Perspectives, pp. 89-104. Austrian Computing Group (OGC) Press (2003)

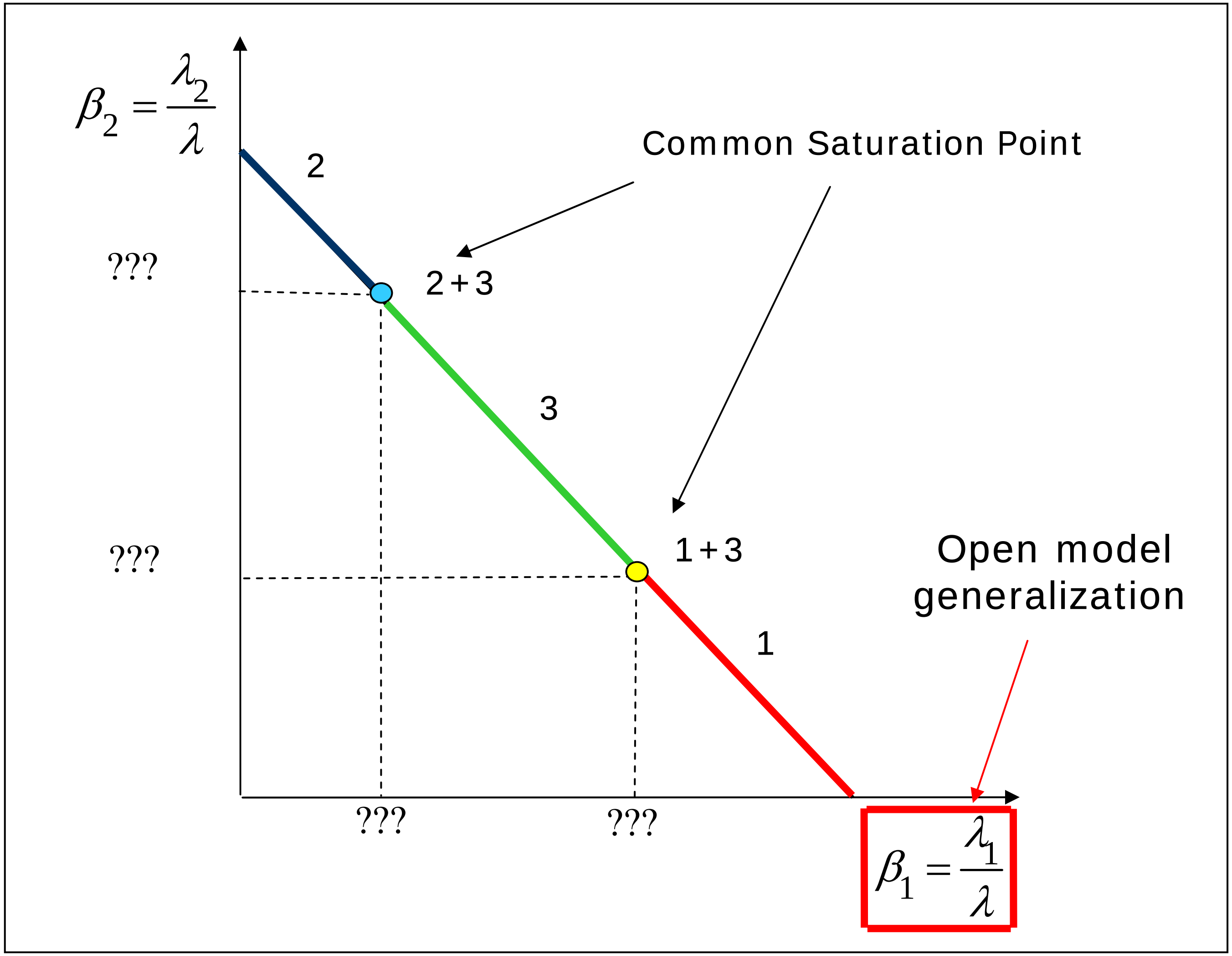
Only one bottleneck in the network

$$\begin{array}{cc} & \textit{Classes} \\ \textit{Stations} & \begin{bmatrix} 100 & 50 \\ 50 & 90 \\ 40 & 40 \end{bmatrix} \end{array}$$



Bottleneck migration in open models

$$\begin{array}{c|cc} & \multicolumn{2}{c}{\textit{Classes}} \\ & 60 & 5 \\ \textit{Stations} & 15 & 70 \\ & 40 & 50 \end{array}$$

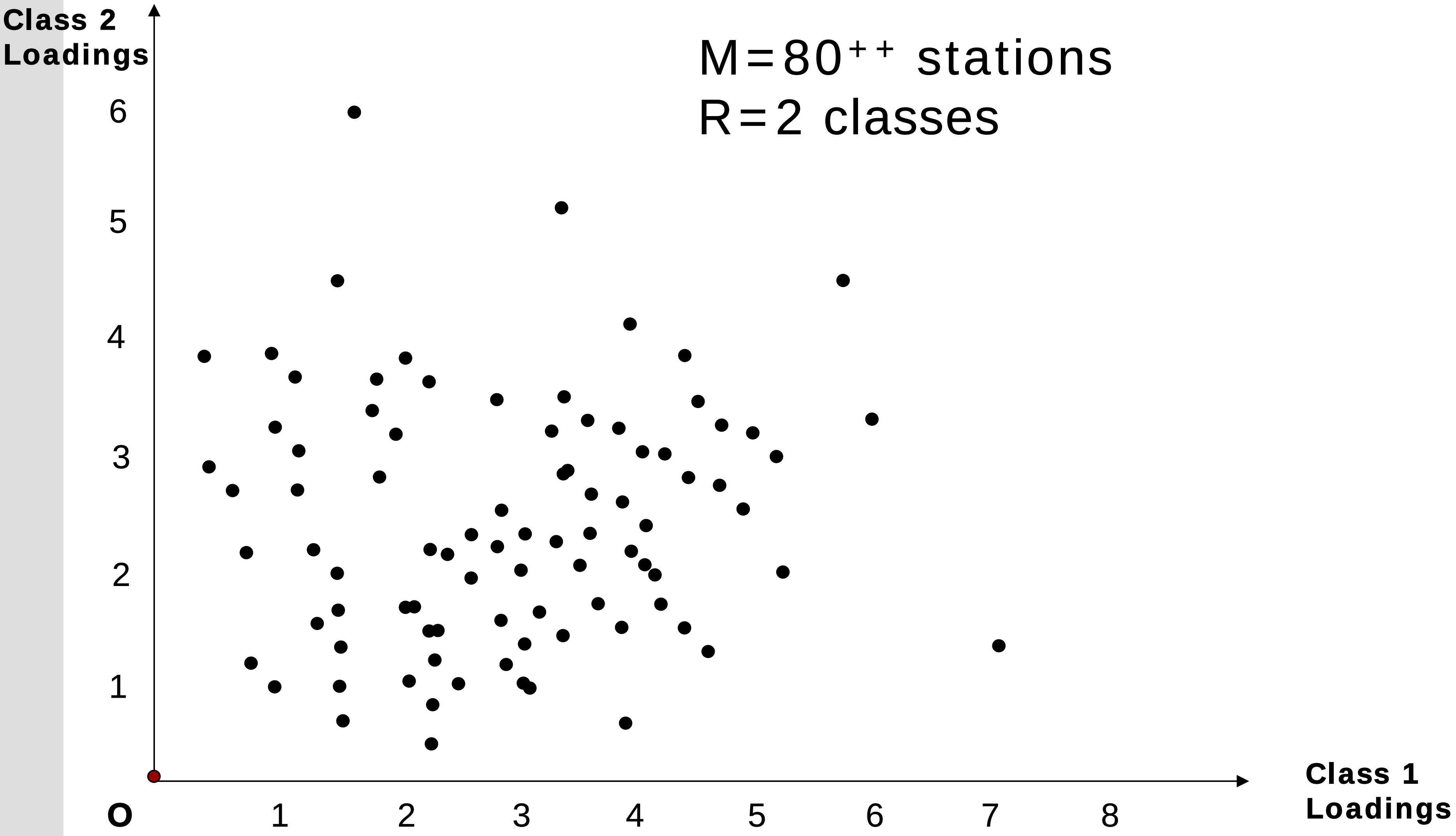


Common Saturation Point

Open model generalization

$\beta_2 = \frac{\lambda_2}{\lambda}$
2
???
2+3
3
???
1+3
1
???
???
$\boxed{\beta_1 = \frac{\lambda_1}{\lambda}}$

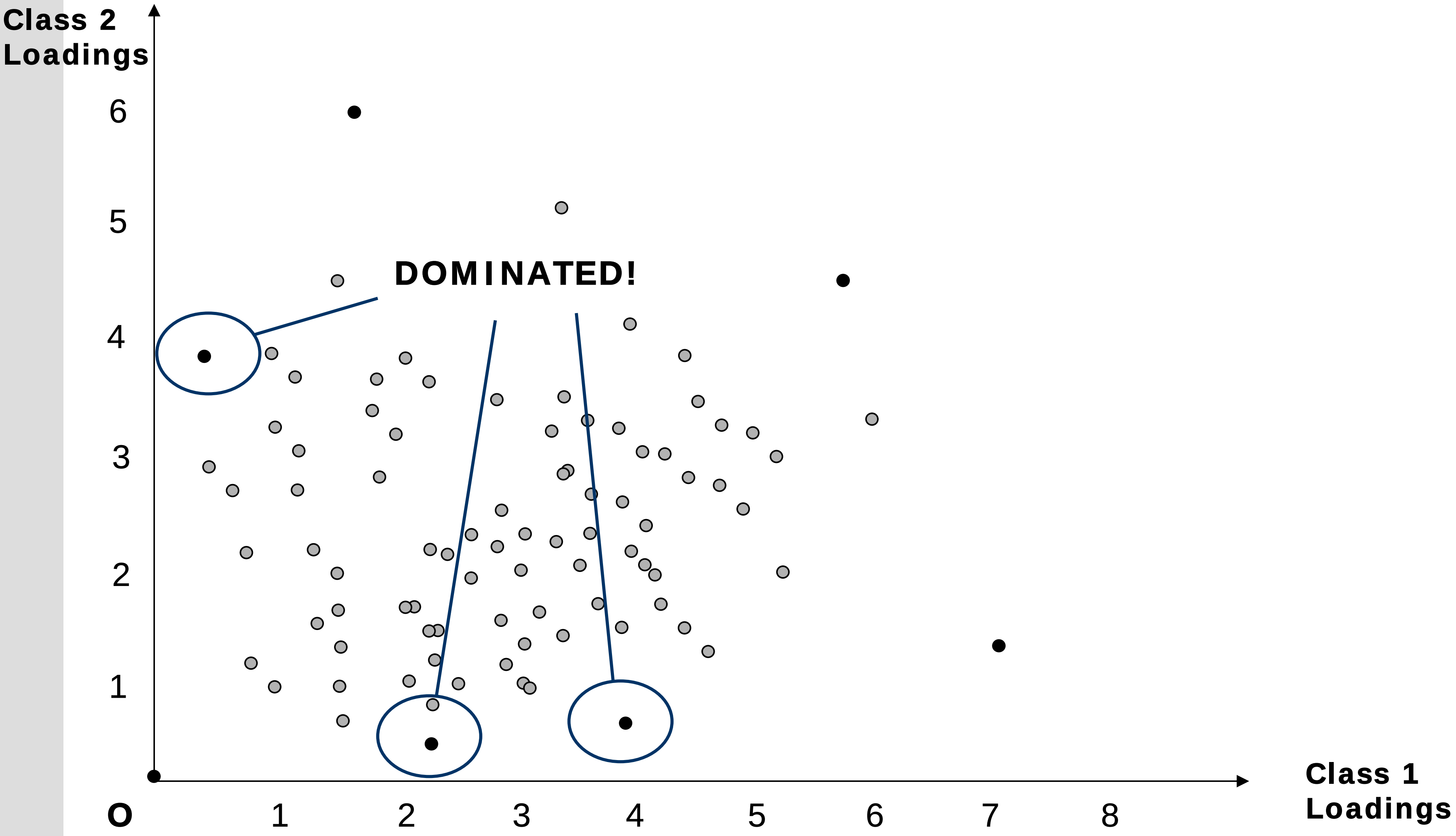
A simple case study



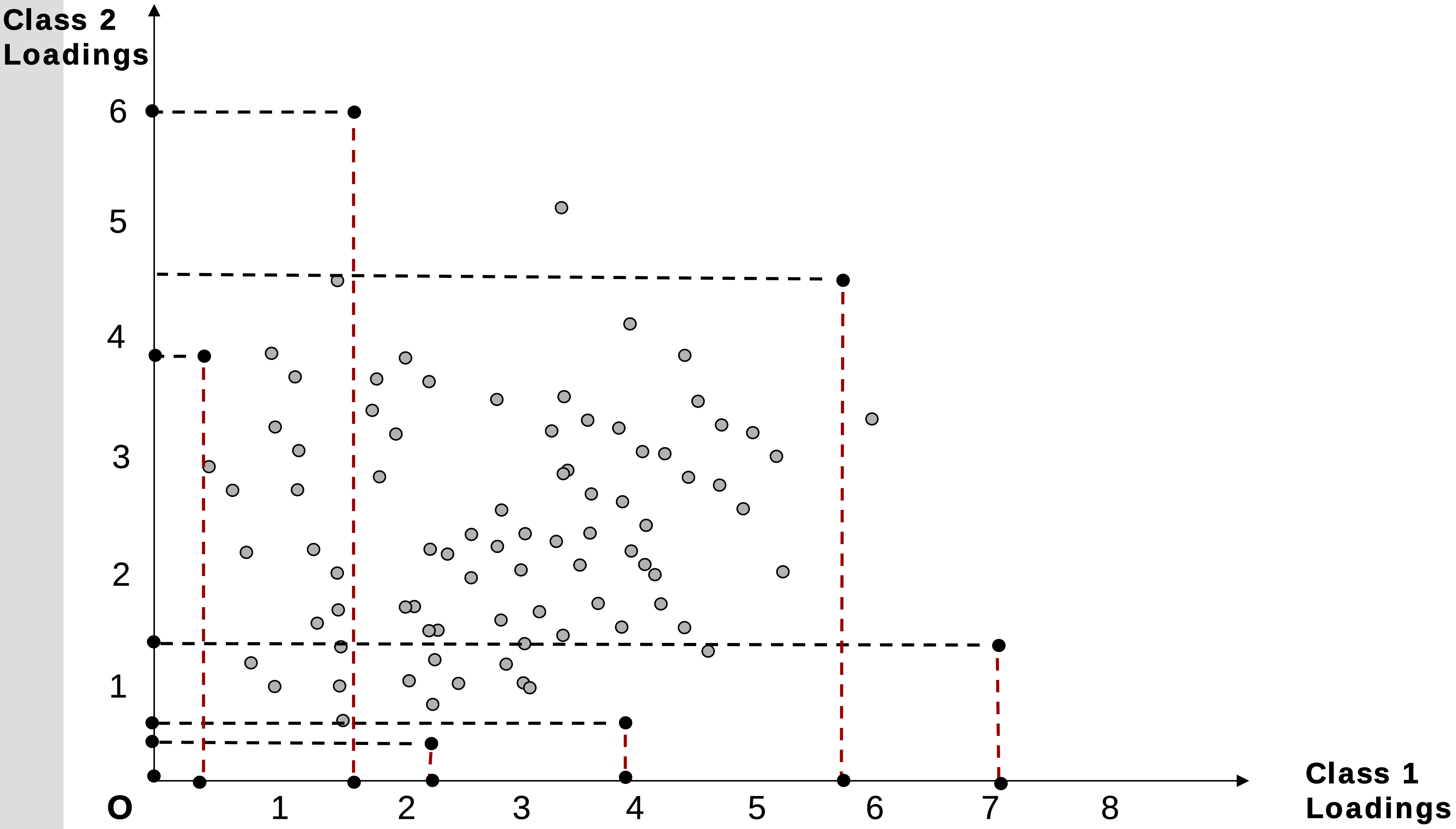
M=80^{++} stations

R=2 classes

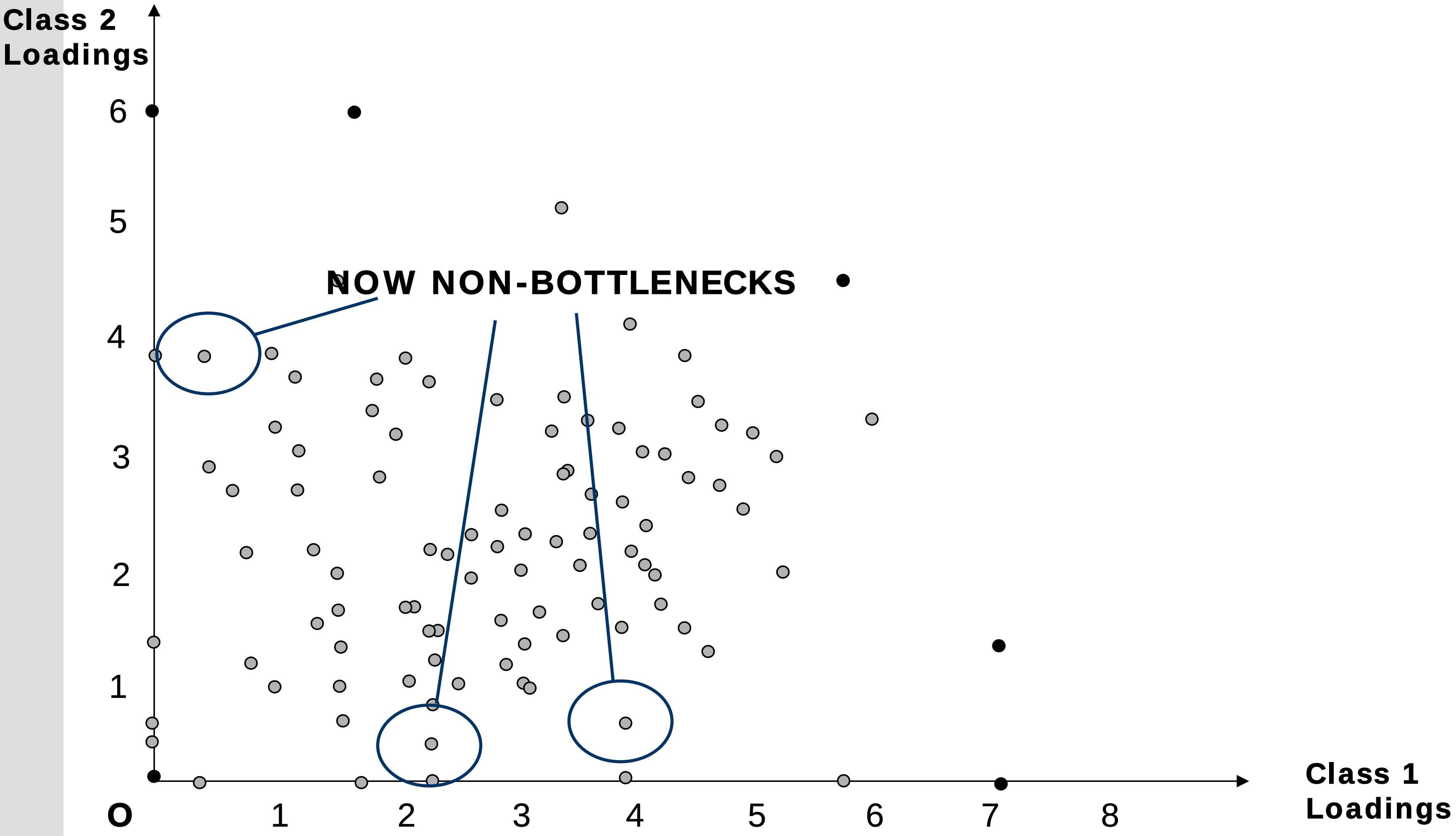
A superset of the potential bottlenecks set



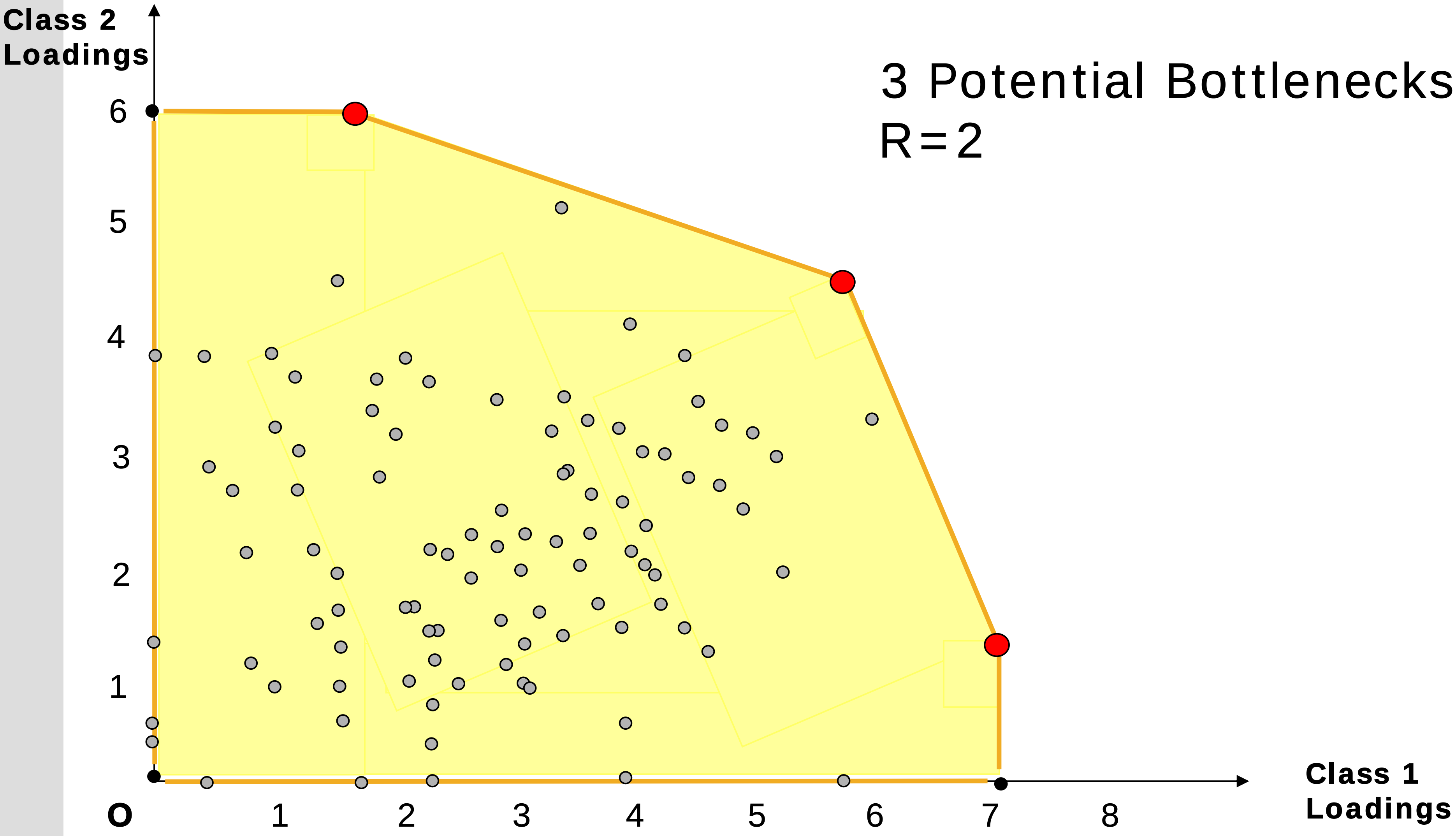
Projection of the remaining stations



A new REP gives the pot. bottleneck set



The convex hull of the model

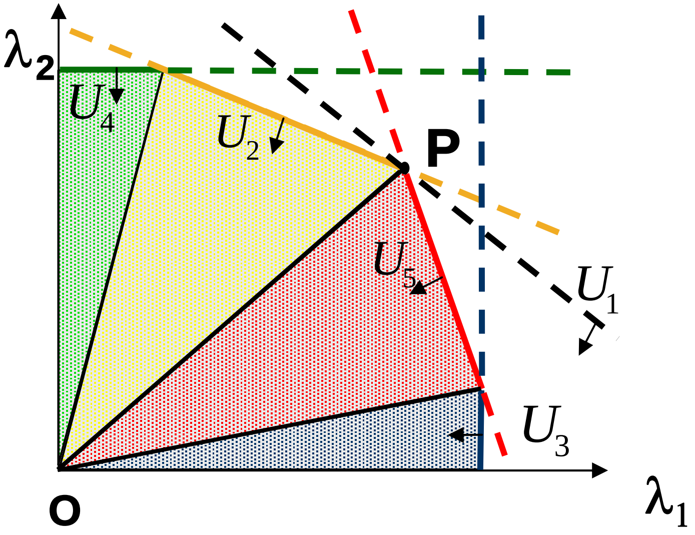


3 Potential Bottlenecks
R=2

Optimal operational points

Multiclass open networks

- Saturation occurs when one or more contraints $U_k \leq 1$ are active $(U_k = 1)$
- Any set of active contraints defines a face of the convex hull of the feasible region



1. The convex hull identifies the actual bottlenecks
2. Bottleneck migration occurs at the interface of two or more faces
3. The common saturation sector in the arrival rate space is at most (R-1)-dimensional