Politecnico di Milano
Dip. Elettronica e Informazione

Bottlenecks Identification in “Very Large” Multiclass Queueing Models

Giuliano Casale
casale@elet.polimi.it
Giuseppe Serazzi
serazzi@elet.polimi.it

Roma, 9/06/2004

Outline

1. “Very large” models: motivations and examples

2. Geometric bottlenecks identification
   - multiclass modification analysis

3. Experimental results

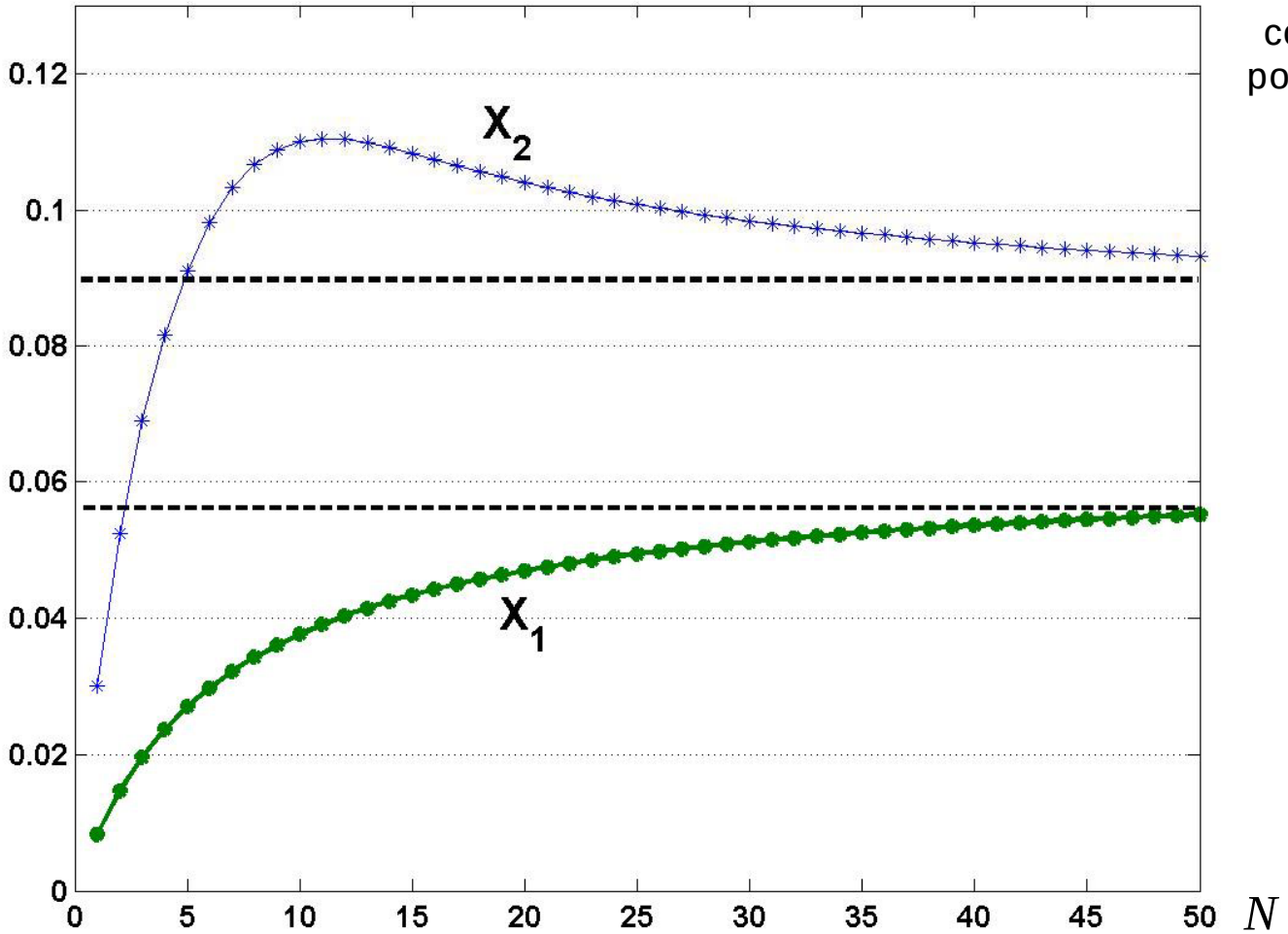
4. Conclusions and future work

Motivations

- Complexity of modelling actual computer infrastructures
  - large installations comprising thousand of servers
  - strongly multiclass workload
  - detailed informations can be collected using automated performance monitors (e.g. BMC Patrol, ...)

- Performance evaluation using queueing networks models requires to deal with **"Very Large" Models (VLM)**
  - handling the curse of dimensionality
  - complex behavior of multiclass models

Complex behavior of multiclass models

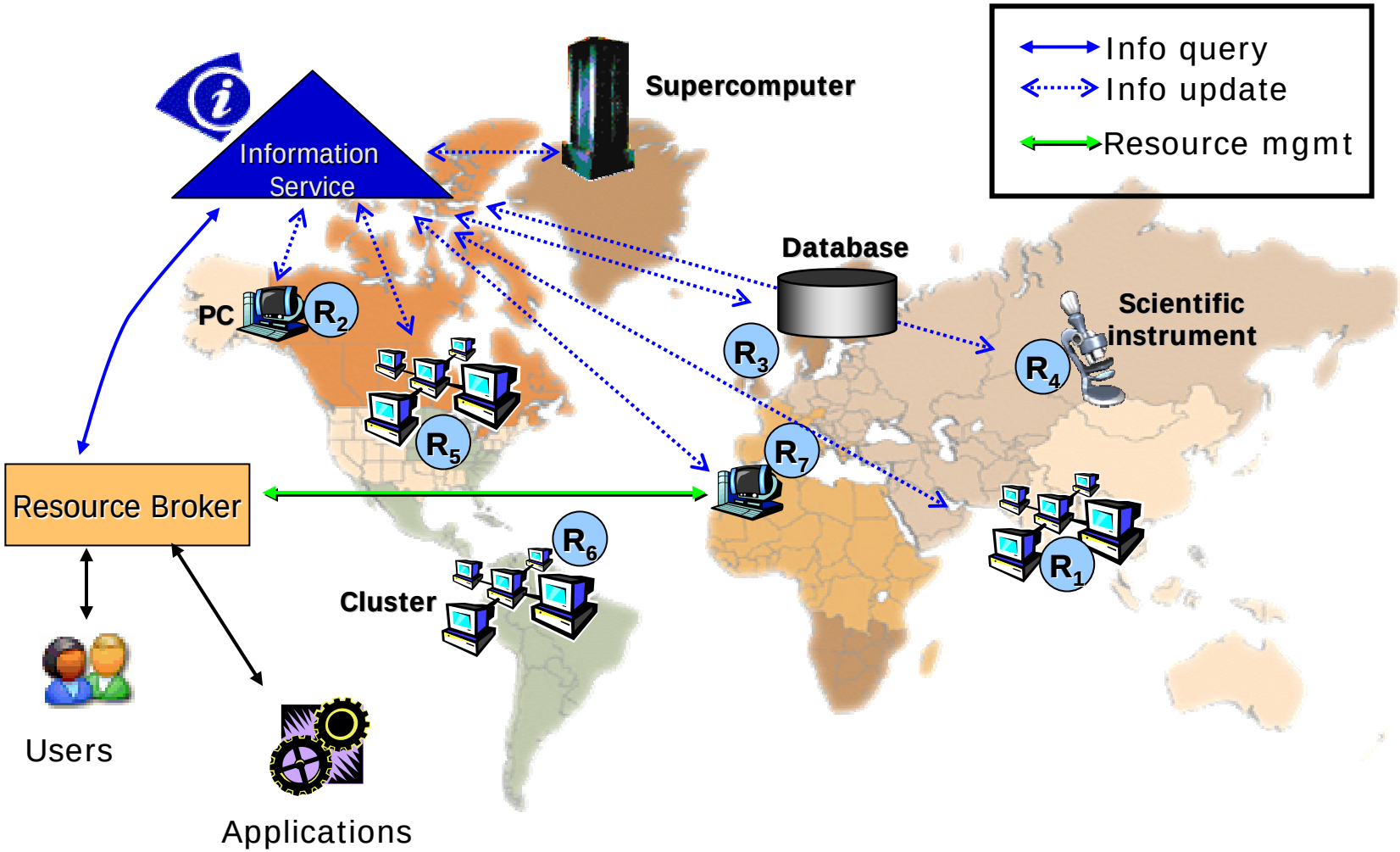


constant population mix

VLM Examples
Network infrastructures

- 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

VLM Examples
Emerging Distributed Technologies: Grid computing



- Several issues: optimal scheduling, load balancing, ...

Modelling Techniques for VLMs

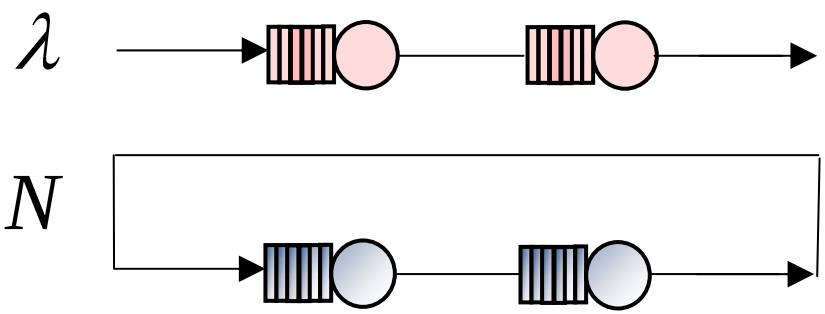
- How to analyze a VLM with product-form queueing networks?
  - excessive computational requirements for an exact solution
  - approximate techniques?
    - suffer decrease in accuracy as the number of classes grows

      [Zahorjan, Eager, Sweillam. *Accuracy, Speed, and Convergence of Approximate Mean Value Analysis.* Perf. Eval. 8(4), 255–270 (1988)]
    - little is known for a large number of classes (say >> 4)
      - empirically Linerizer converges slowly
    - execution times: only B-S looks suitable for an online solution

- However, product-form requirements may not be satisfied

Notation and Assumption

- Both open and closed general multiclass queue nets



- M stations
- R customer classes
- Loading matrix

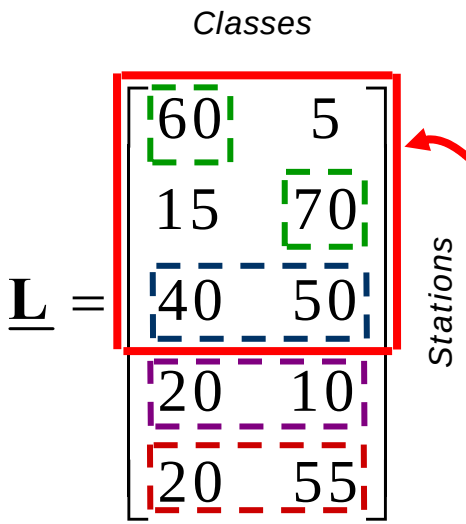
$$\mathbf{L} = \{L_{ir} = V_{ir} S_{ir}\}$$

Customer Classes

Queueing Stations

$$\begin{bmatrix} L_{11} & L_{12} & \dots & L_{1R} \\ L_{21} & L_{22} & \dots & L_{2R} \\ \dots & \dots & \dots & \dots \\ L_{M1} & \dots & \dots & L_{MR} \end{bmatrix}$$

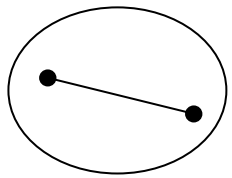
Taxonomy of stations



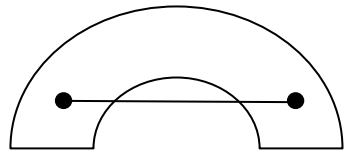
- **Natural bottlenecks**
  bottlenecks when a single class is present in the network
- **Network bottlenecks**
  can saturate only under a multiclass population mix
- **Potential bottlenecks set**
  (network + natural) bottlenecks
  $\Pi = \{1, 2, 3\}$
- **Dominated stations**
  4 has all components less than those of 3
- **Masked-off stations**
  5 not dominated, but never saturates

Computational Geometry

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



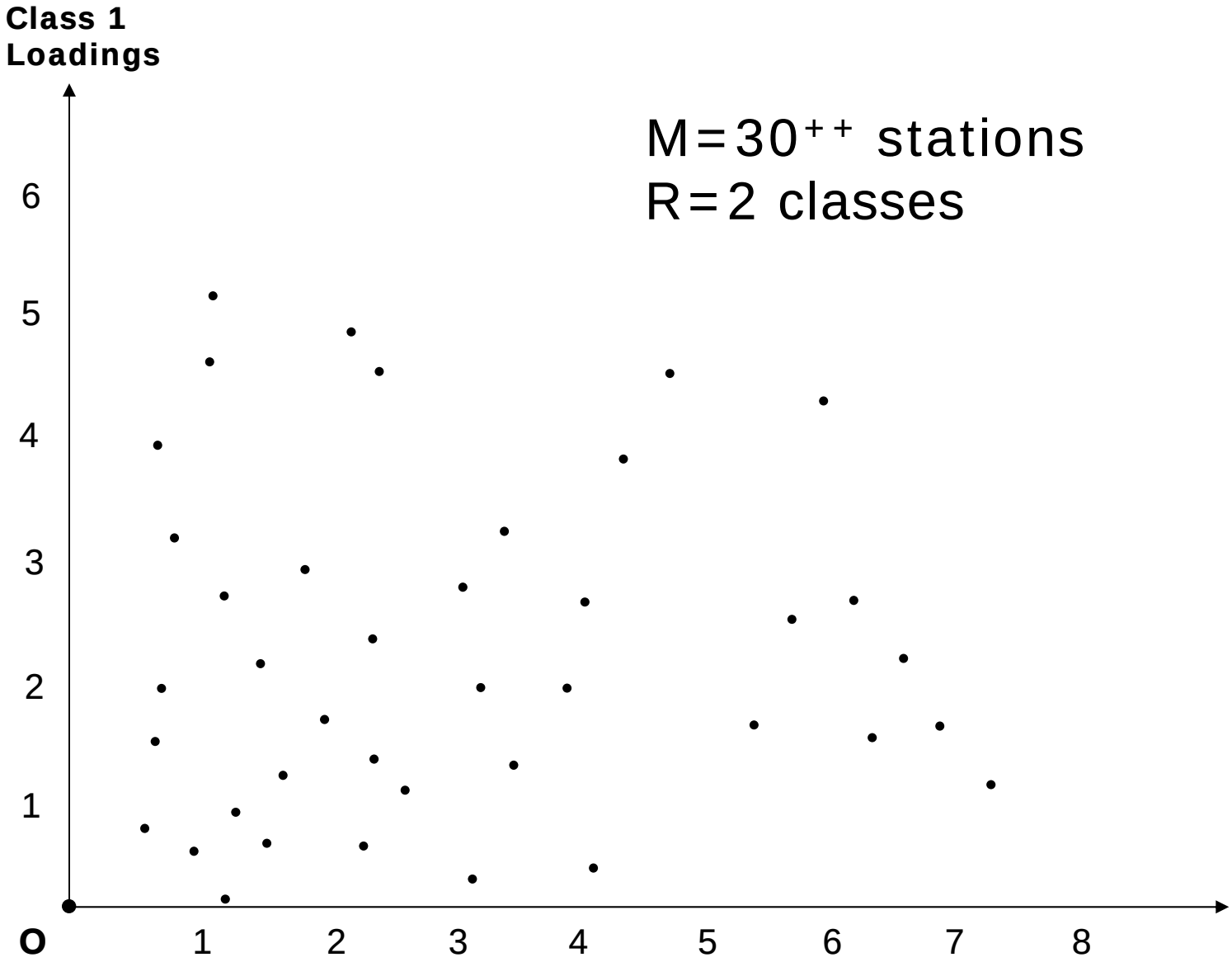
Convex set



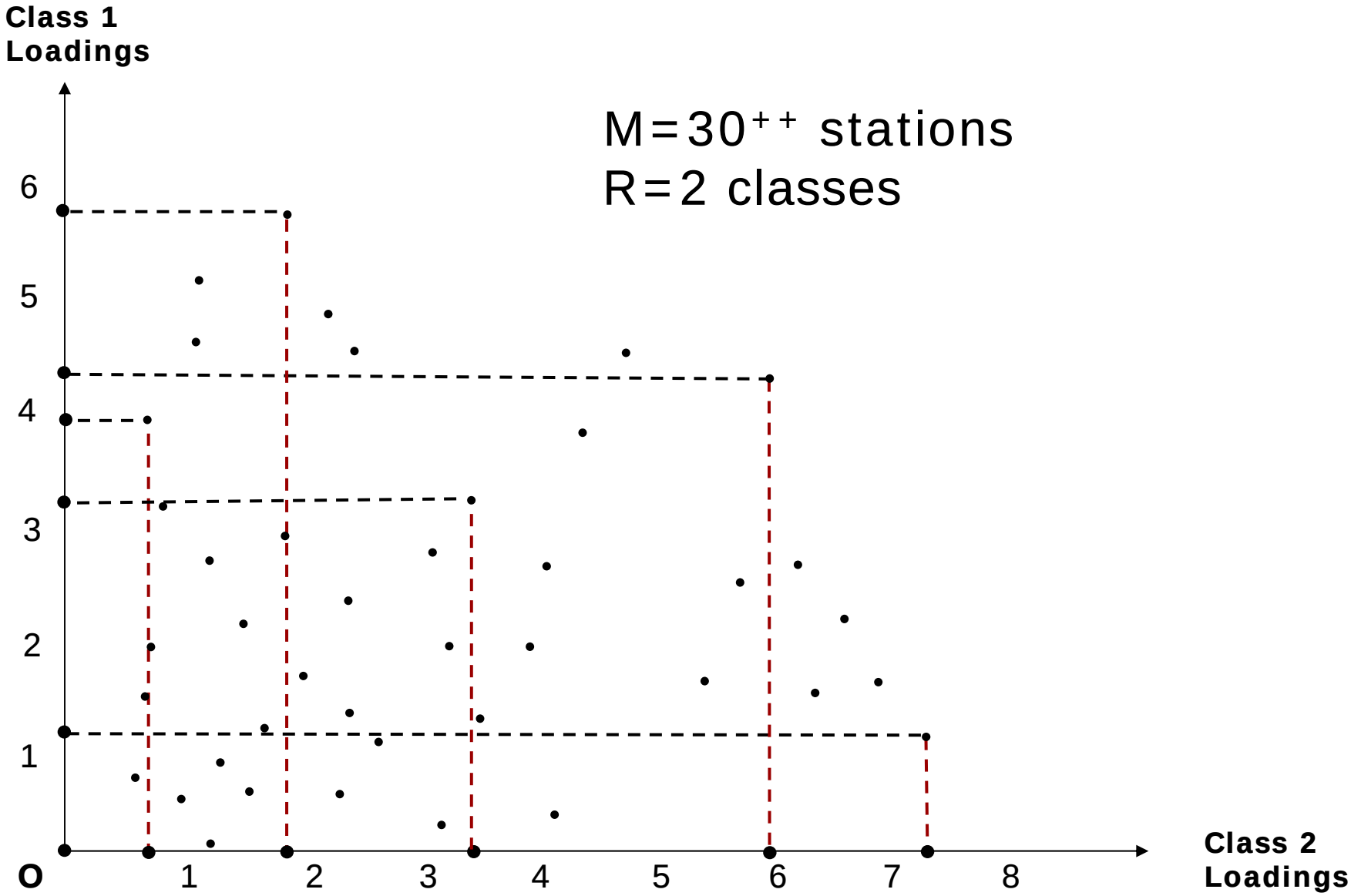
Non-convex set

- **Convex hull problem**: find the smallest convex set containing a given set of M points
  - several applications: computer vision, information theory, ...
  - fast algorithms in 2D [O(M logM)] and in 3D [$O(M^2)$] exist
  - efficient algorithms up to 7-8 dimensions (QHULL, CDD)
  - both offline and online algorithms are available

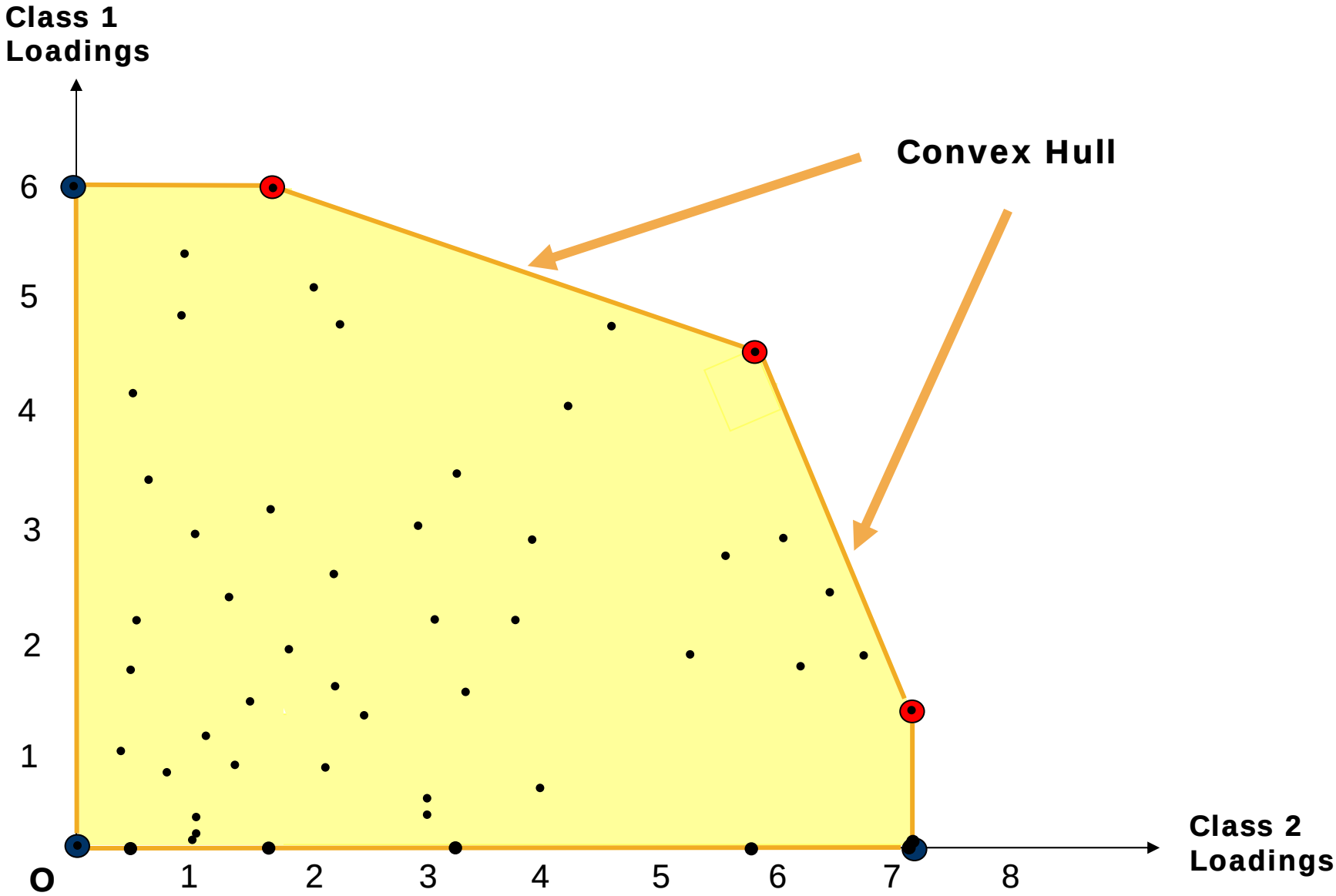
Loadings space



Apply projection to all points

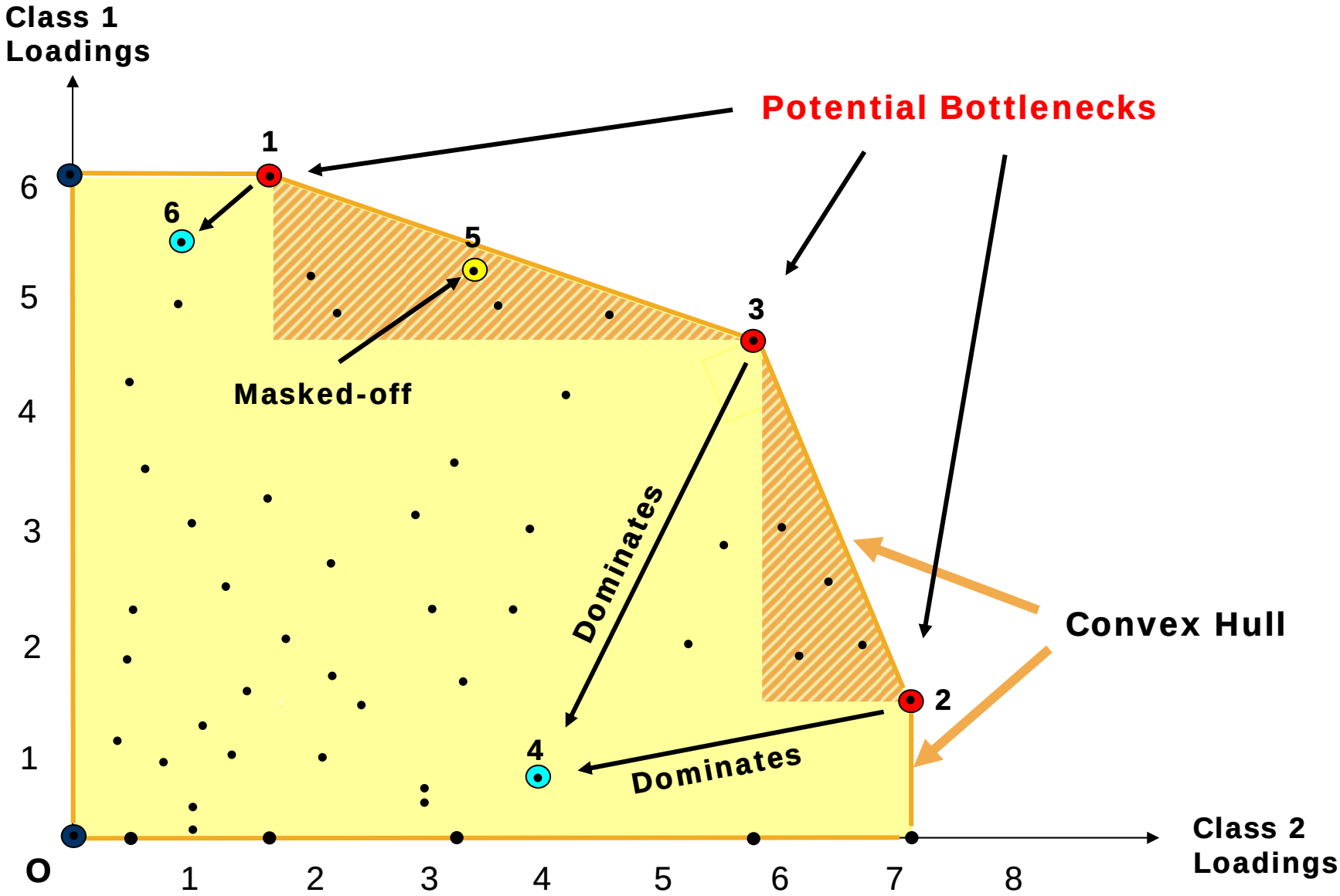


Convex hull in 2 dimensions



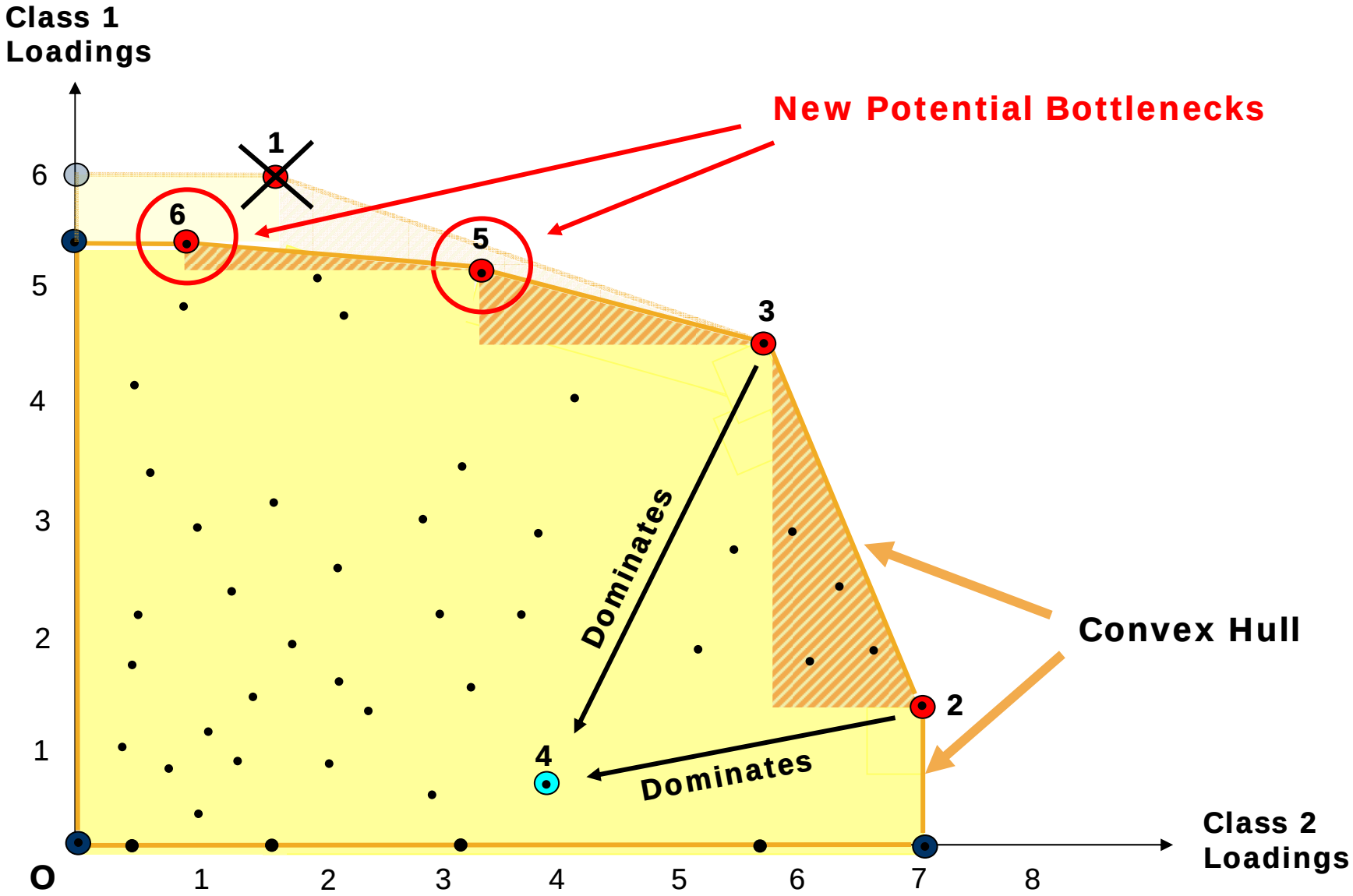
Potential Bottlenecks Identification

Convex hull of the loading matrix



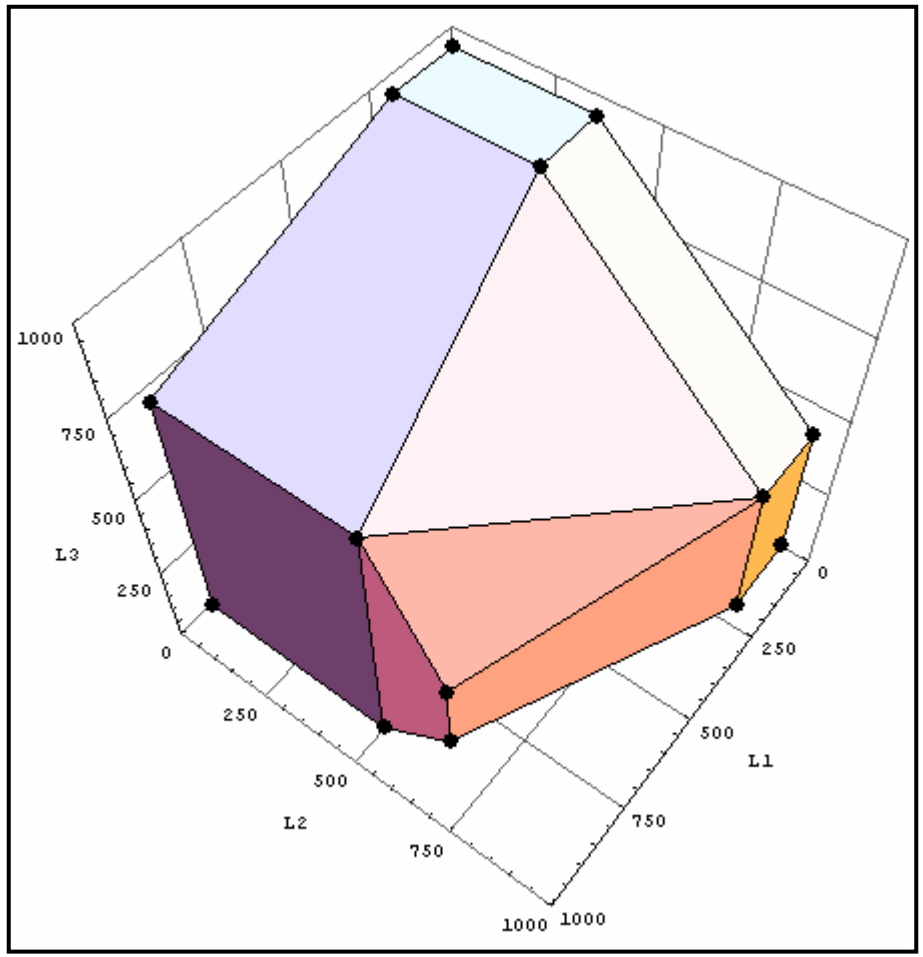
Potential Bottlenecks Identification

Modification Analysis



Potential Bottleneck Identification

Convex hull of a 3-class model



Redundancy elimination

- The time complexity of the convex hull of M points in higher dimensions is $O(M^{R/2}) \rightarrow$ exponential in the num of classes R

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

Excessive Requirements!

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

- LP techniques instead of convex hulls
  - Polynomial time complexities in the number of classes R (and in the number of stations M)

Potential bottenecks Identification
Experimental results

LP Techniques CPU TIME WORST CASE	R=5 classes	R=10	R=25	R=50
M=1000 stations	4 secs	6 secs	15 secs	48 secs
M=10000	2 minutes	4 minutes	10 minutes	31 minutes
M=100000	5 hours	7 hours	9 hours	16 hours

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

- LP techniques are formulated as a set of independent problems
  - easy to parallelize
- Heuristic strategies for quick identification of dominated and masked-off stations are available

Conclusions and Future work

- Multiclass generalization of single class modification analysis
- Further studies required to relate convex hulls with asymptotic performance indices
- Time requirements comparison with approximate techniques
- Applications to real-time performance management