DERIVING HIGH-LEVEL ABSTRACTIONS FROM LEGACY SOFTWARE USING EXAMPLE-DRIVEN CLUSTERING

MOSART, 11 Oct. 2011

Martin Faunes in collaboration with Marouane Kessentini and Houari Sahraoui
GEODES, Université de Montréal

High-Level Abstractions

- Reify actual concepts of the application domain
- Help in program understanding
- Could be extracted from concrete elements in the code (reverse engineering)
- Extraction
  - Software clustering

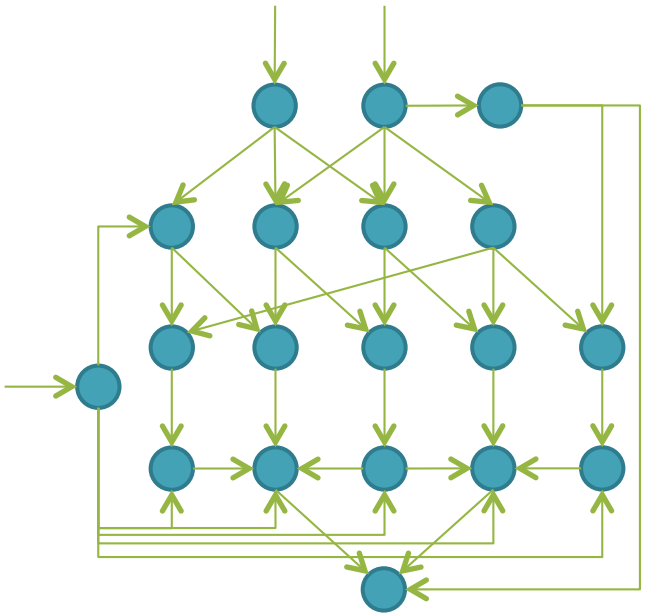
Software Clustering (SC)

- Definition
  - "decompose the structure of software systems into meaningful subsystem" [Mancoridis et al. 1999]
- Examples of problems
  - Grouping routines and variables into objects/classes (migration)
  - Grouping routines into modules (remodularization)
  - Grouping classes into modules/packages (remodularization)
  - Grouping classes into components (rearchitecture/migration)
  - Grouping elements into features (feature location)

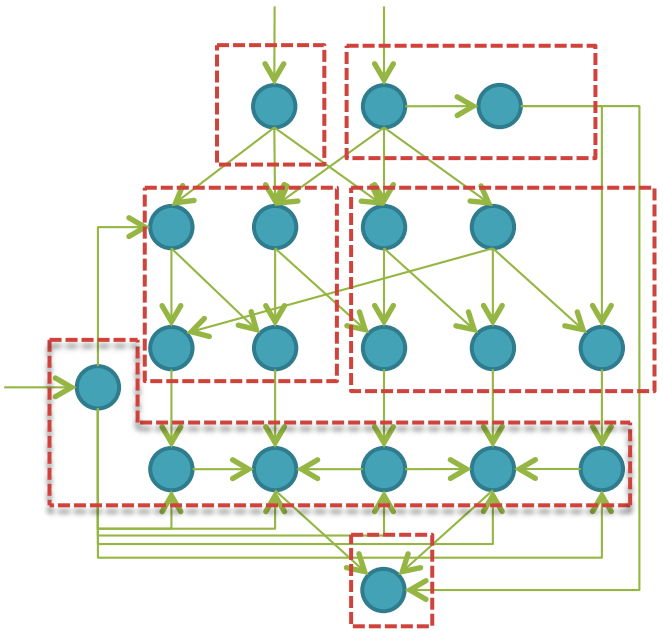
SC as a Graph Partitioning Problem

- Software generally defined as set of interconnected elements (graph)
- Clustering ≈ Graph partitioning problem
  - Graph $G(V, E)$ ➔ Software system
  - Vertices V ➔ Elements of interest in the software
  - Edges E ➔ Dependencies between these elements
- Objective: Find the best partition P_G of the graph G according to given criteria
- Large search space

SC as a Graph Partitioning Problem

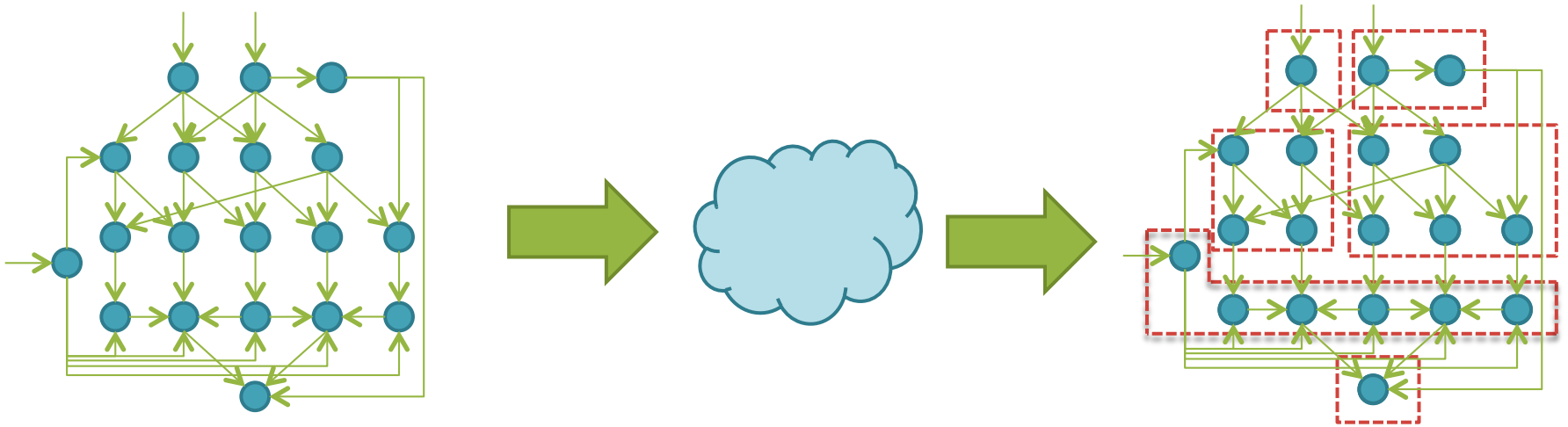


A graph representing
a software system



A graph partition

SC as a graph partitioning problem



Meaningfulness hypothesis

Meaningfulness Hypotheses

- Structural hypothesis: Entities that are structurally related are semantically related
  - Coupling and cohesion
    - Sahraoui et al. ASE, 1997
    - Marcondis et al. IWPC, 1998
    - Van Deursen et al. ICSE, 1999
    - Mitchell, Doctoral dissertation 2002
    - Harman et al. GECCO 2002
    - Sahraoui et al. COMPSAC 2002
    - Harman et al. GECCO 2007
    - Abdeen et al. WCRE 2009

Meaningfulness Hypotheses

- Structural hypothesis: Entities that are structurally related are semantically related
  - Cycling dependencies
    - Abdeen et al. WCRE 2009
  - Bottlenecks
    - Seng et al. GECCO 2005
  - Size and complexity
    - Seng et al. GECOO 2005

Meaningfulness Hypotheses

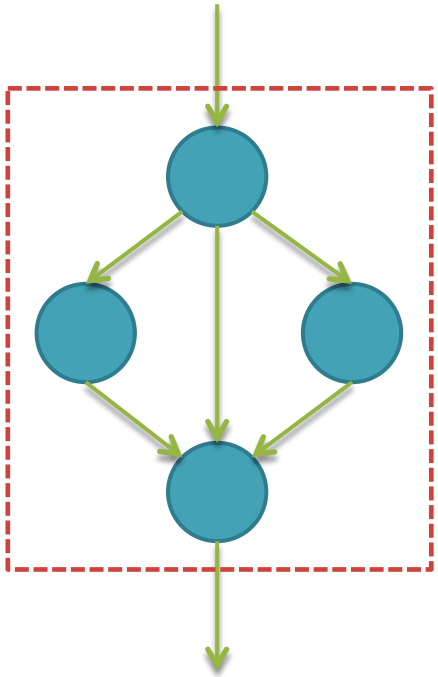
- Co-change hypothesis: Entities that change together are semantically related
  - Bayer et al. IWPC 2005

- Information-theory hypothesis: Entities that share the same information are semantically related (Groups should be formed trying to minimize information lost)
  - Andritsos et al. WCRE 2003.

Meaningfulness Hypotheses

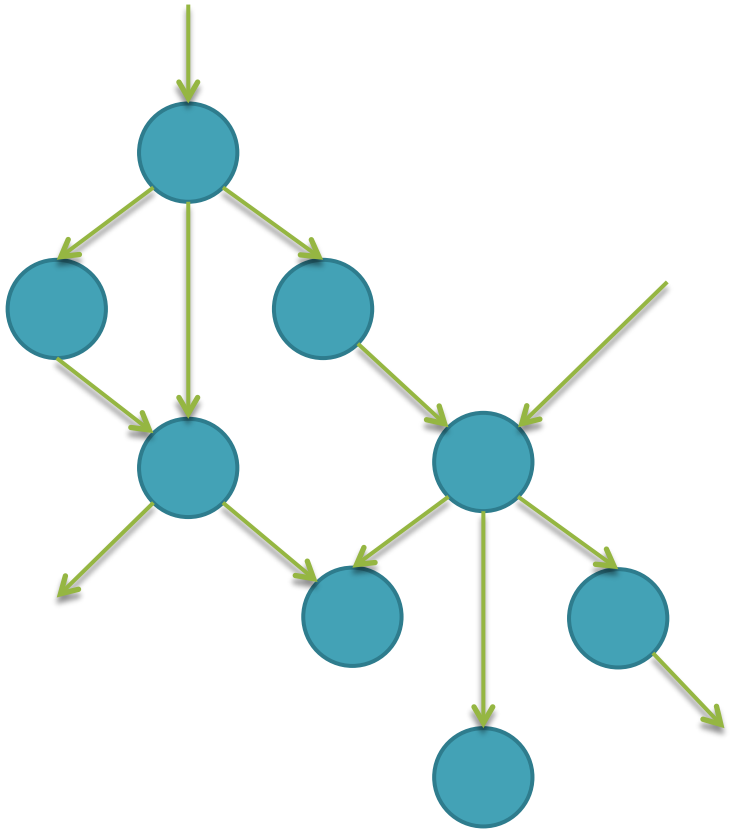
- Limited correctness of the previous hypotheses
- We explore a new hypothesis
  - If a set of elements was identified in the past as a meaningful cluster,
  - then elements, in a software to cluster, that are organized in a similar way could form a cluster as well
- Example-based software clustering

Our Approach: by Example



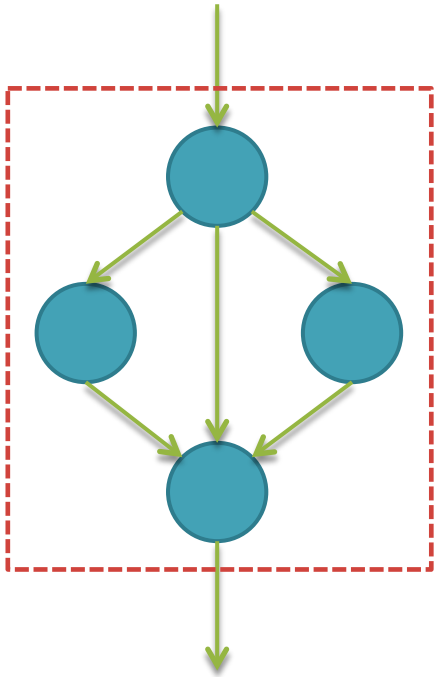
An example cluster

?



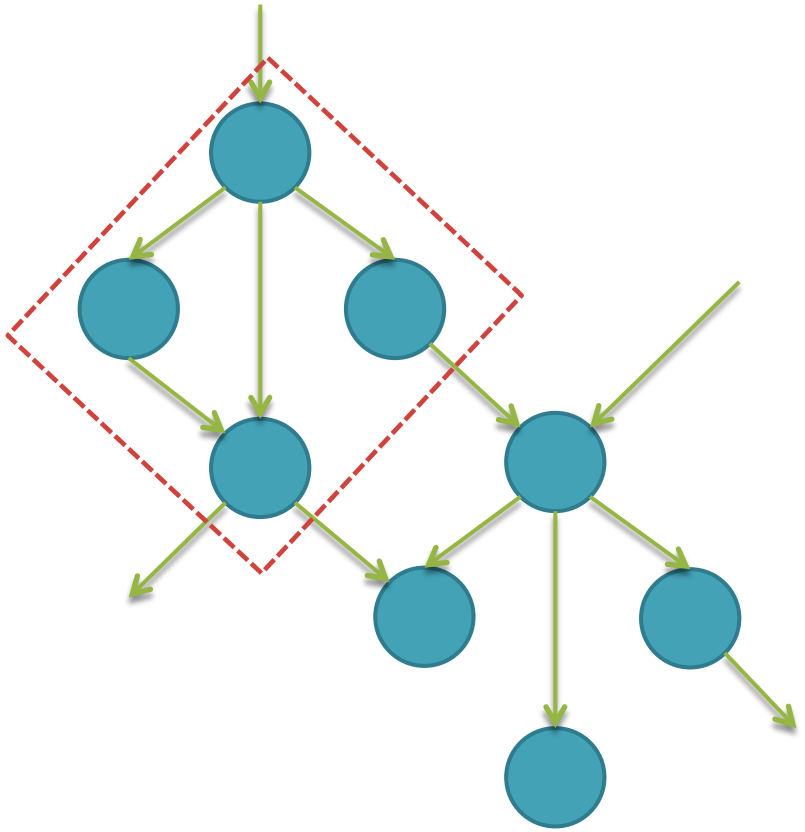
A graph of software system to be partitioned

Our Approach: by Example



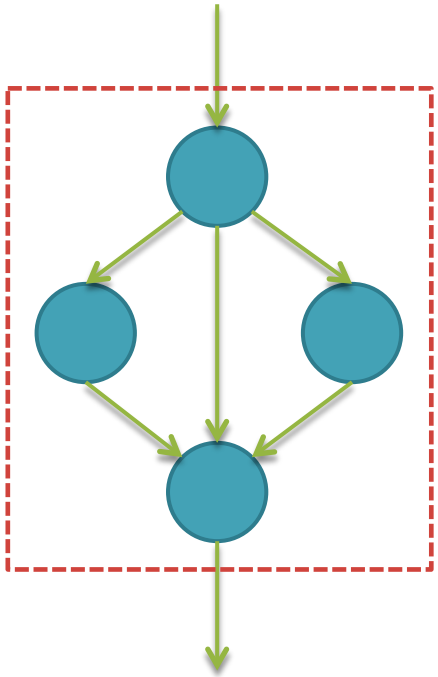
An example cluster

similar



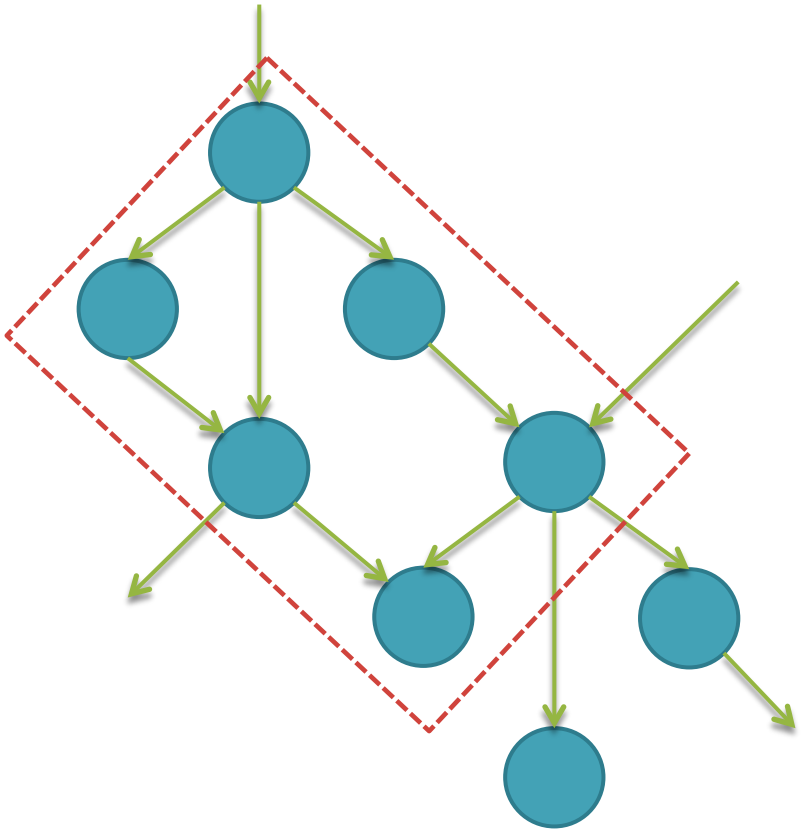
A possible cluster

Our Approach: by Example



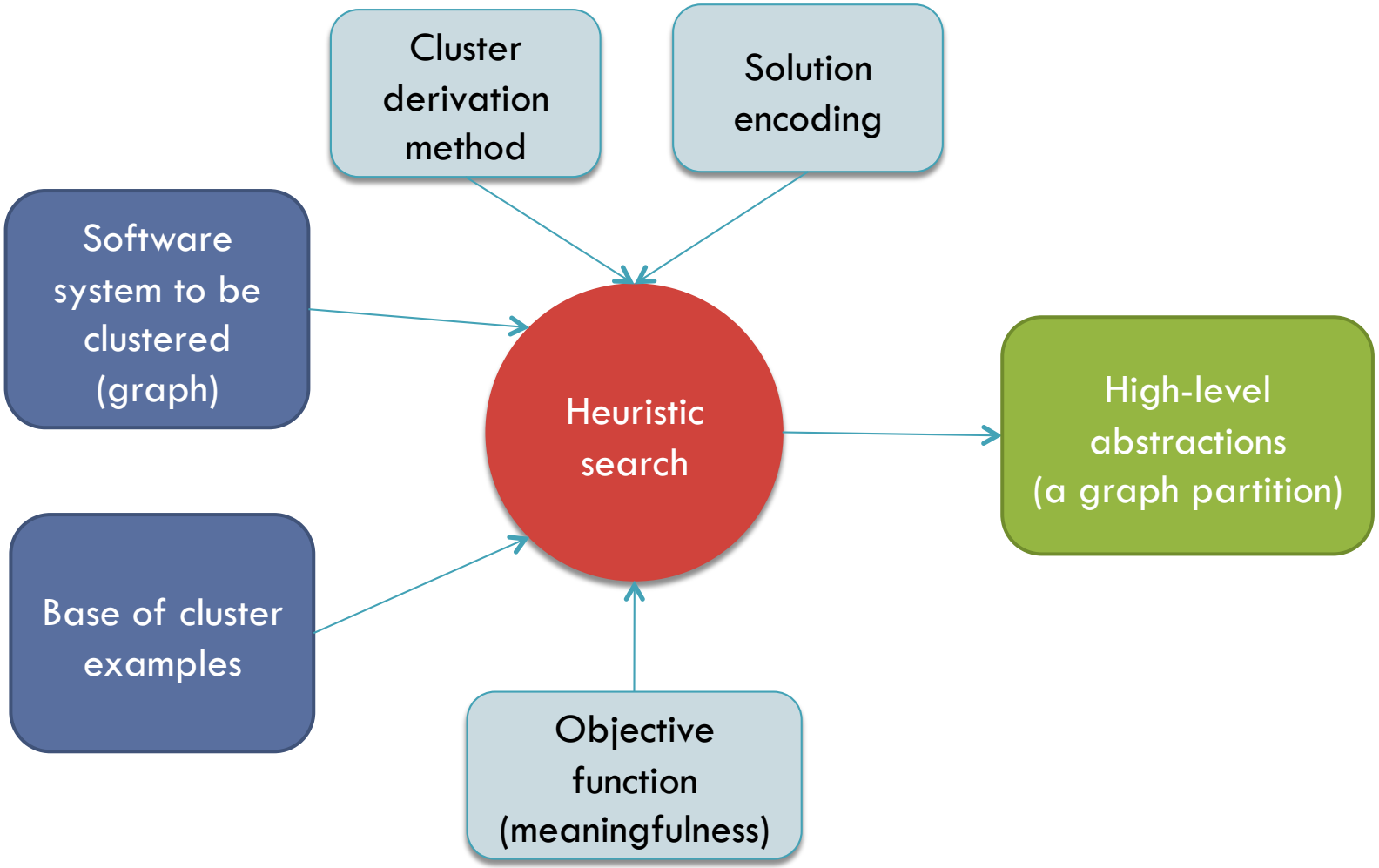
An example cluster

less similar

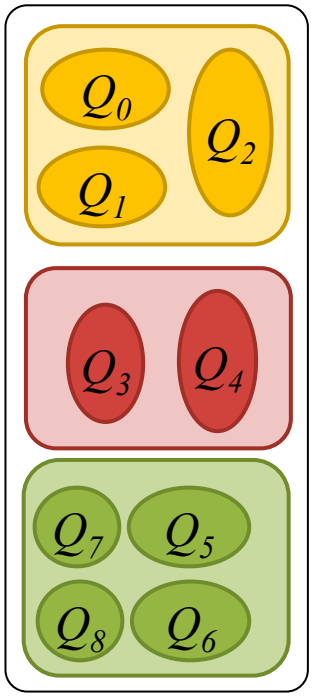


Another possible cluster

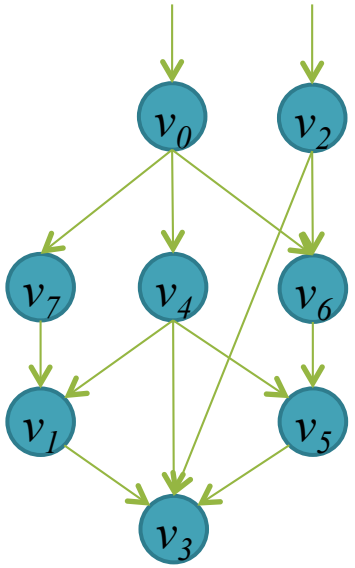
Approach Overview



Deriving a Partition

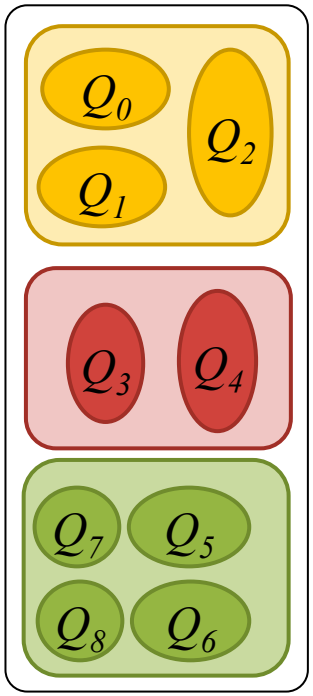


Base of examples

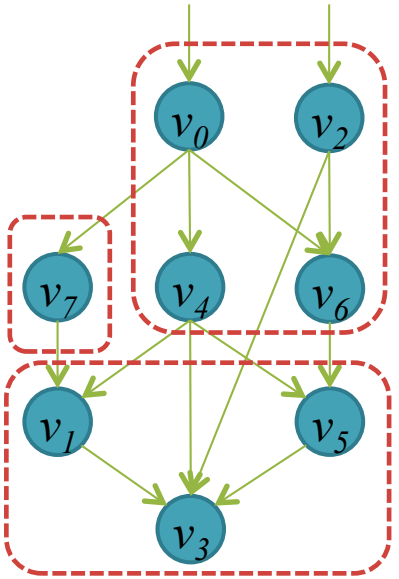


Graph representing the software system to be clustered

Deriving a Partition



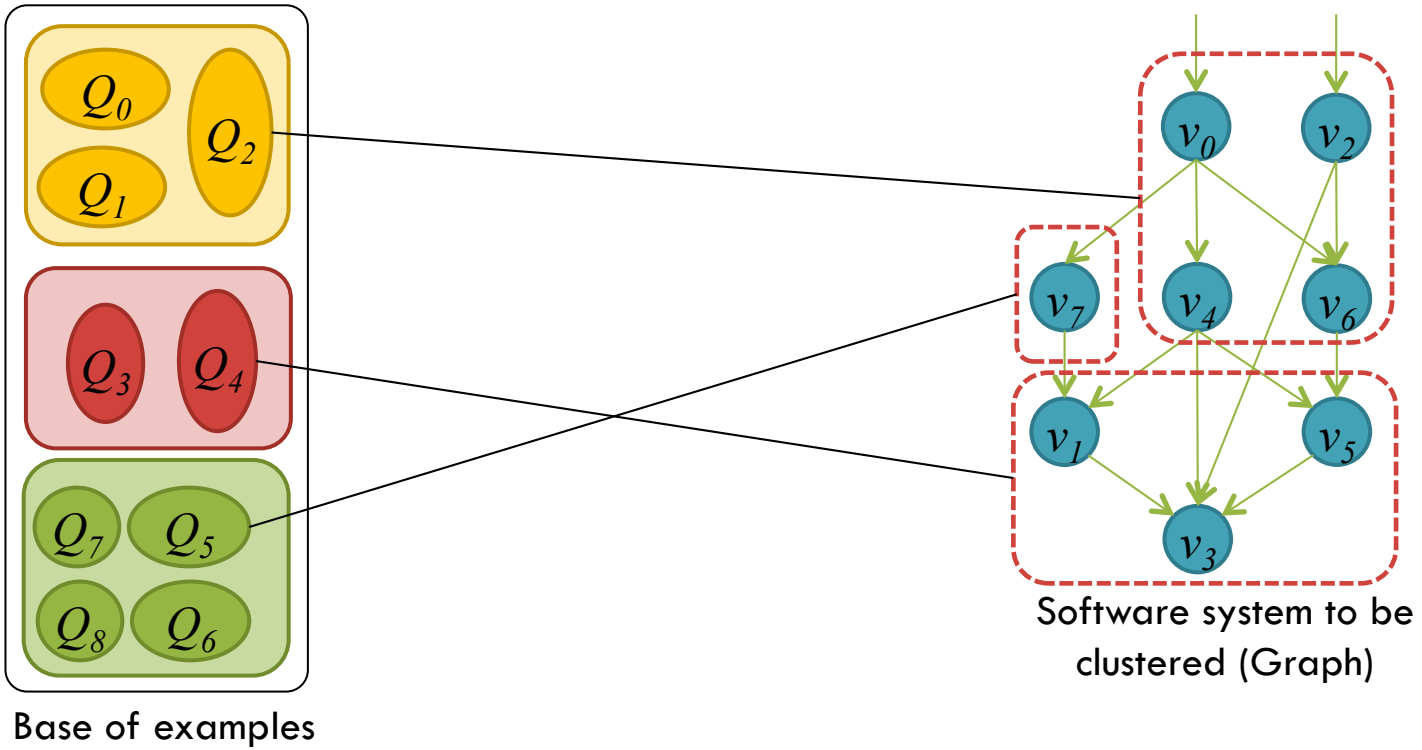
Base of examples



Software system to be clustered (Graph)

We create random clusters

Deriving a Partition



We assign the created clusters to example clusters and assess their similarity

Objective Function f

- f : measures the structural similarity between
  - The clusters in the partition P_G
  - Their corresponding clusters in the base of examples

$$f(P_G, BE) = \frac{\sum_{i=1}^{k} |K_i| Sim(K_i, Q_j)}{|V|} \in [0,1]$$

Objective Function f

- Sim compares the structure of
  - a potential cluster K
  - with its corresponding cluster Q in the base of examples

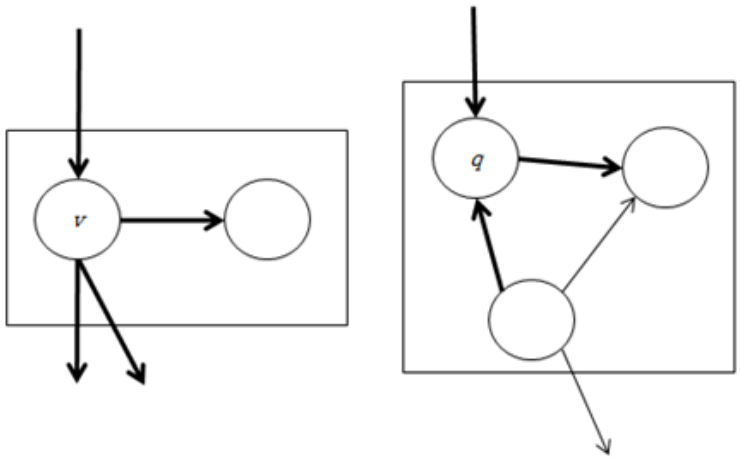
$$Sim(K, Q) = \frac{1}{|K|} \sum_{v \in K} \max_{q \in Q} vSim(v, q) \in [0,1]$$

Objective Function f

- $vSim$ compares two vertexes v and q
- $vSim(v, q) =$
  - If v and q are of different types
  - $vSim(v, q) = 0$
  - Otherwise
  - $vSim(v, q)$ = ratio of matching edges
    - Two edges of v and q match if
      - both are of the same type
      - both are internal (respectively external) edges
      - both are incoming (respectively outgoing) edges
      - Neither v nor q have been matched yet

Objective Function f

- $vSim(v, q)$



$$vSim(v,q) = \frac{2+2}{4+3} = \frac{4}{7}$$

Objective Function f

- The $max\ vsim(v,q)$ is calculated using a similarity-matrix

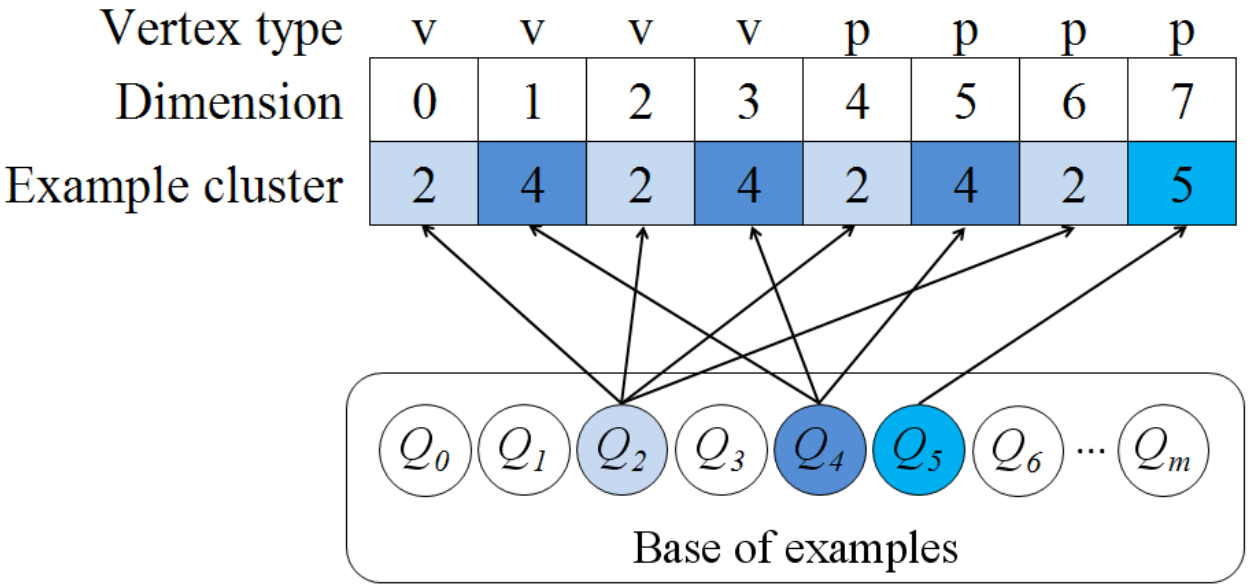
		Q_2						
		q3	***q4***	***q5***	***q11***	***q13***	***q16***	**Max**
K_1	***v1***	0.83	0.40	0.90	0.00	0.00	0.00	0.90
	v3	0.50	0.58	0.25	0.00	0.00	0.00	0.50
	v4	0.25	0.83	0.50	0.00	0.00	0.00	0.83
	v6	0.00	0.00	0.00	0.67	0.50	0.33	0.67
	v9	0.00	0.00	0.00	0.67	0.50	0.33	0.50
	v11	0.00	0.00	0.00	0.67	0.50	0.33	0.33
							Sum	3.73

Application to Object Identification

- Input
  - $G(V, E)$: A procedural software system to be clustered
    - V : Procedures and variables
    - E : Procedure calls and variable accesses.
  - The example base BE
    - Contains a set of procedural programs already partitioned
- Output
  - A partition of $G(V, E)$
    - Cluster represents potential objects

Solution Coding

- Search space: n-dimensional
- Each dimension: a vertex of $G(V, E)$

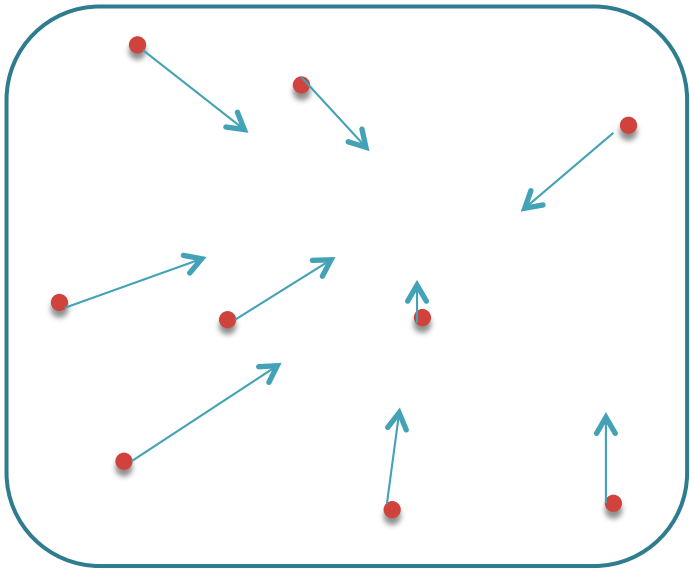


Heuristic Search

- Hybrid heuristic search method that combines:
  - global search
    - Particle Swarm Optimization (PSO) to find an initial solution
  - local search
    - Simulated Annealing (SA) to refine this solution

Heuristic Search - PSO

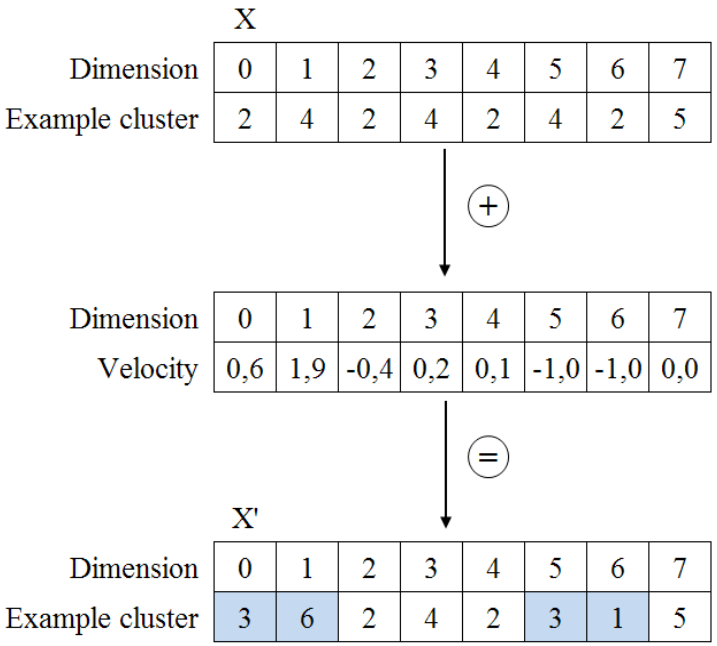
- Starts from a swarm of particles (solutions) scattered randomly in the solution pace
- Each particle knows
  - Particle best solution visited (*pbest*)
  - Swarm best solution visited (*gbest*)
- Each particle has
  - Weight and velocity
- At each iteration
  - Calculate particles objective function
  - Update *pbest*, *gbest*
  - Move the particles
- A change operator
  - $X_i' = X_i + V_i'$
  - $V_i' = W \times V_i + C_1 \times rand_1() \times (pbest_i - X_i) + C_2 \times rand_2() \times (gbest - X_i)$



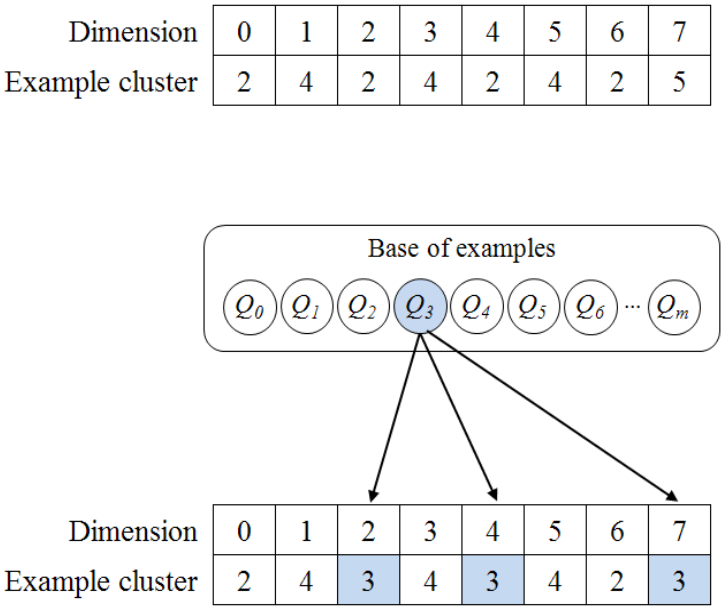
Heuristic Search - SA

- Based on the annealing principle used in metallurgy
- Starts from
  - a random solution s
  - a high temperature t
- In each iteration
  - s' ← a random solution in the neighborhood of s
  - $d \leftarrow f(s') - f(s)$
  - if ($d > 0$)
    - $s \leftarrow s'$
  - else if Prob(d, t)>rand(0,1)
    - $s \leftarrow s'$
  - decrease t

Change Operators



Particle Swarm Optimization



Simulated Annealing

Case Study

- Setting
  - 10 program written in C[1]
  - Manually identify candidates objects
  - 10-fold cross-validation procedure
  - We compare the partition obtained by our clustering method to the partition generated manually
  - We also compare our results with the ones based on the structural hypothesis (cohesion and coupling)

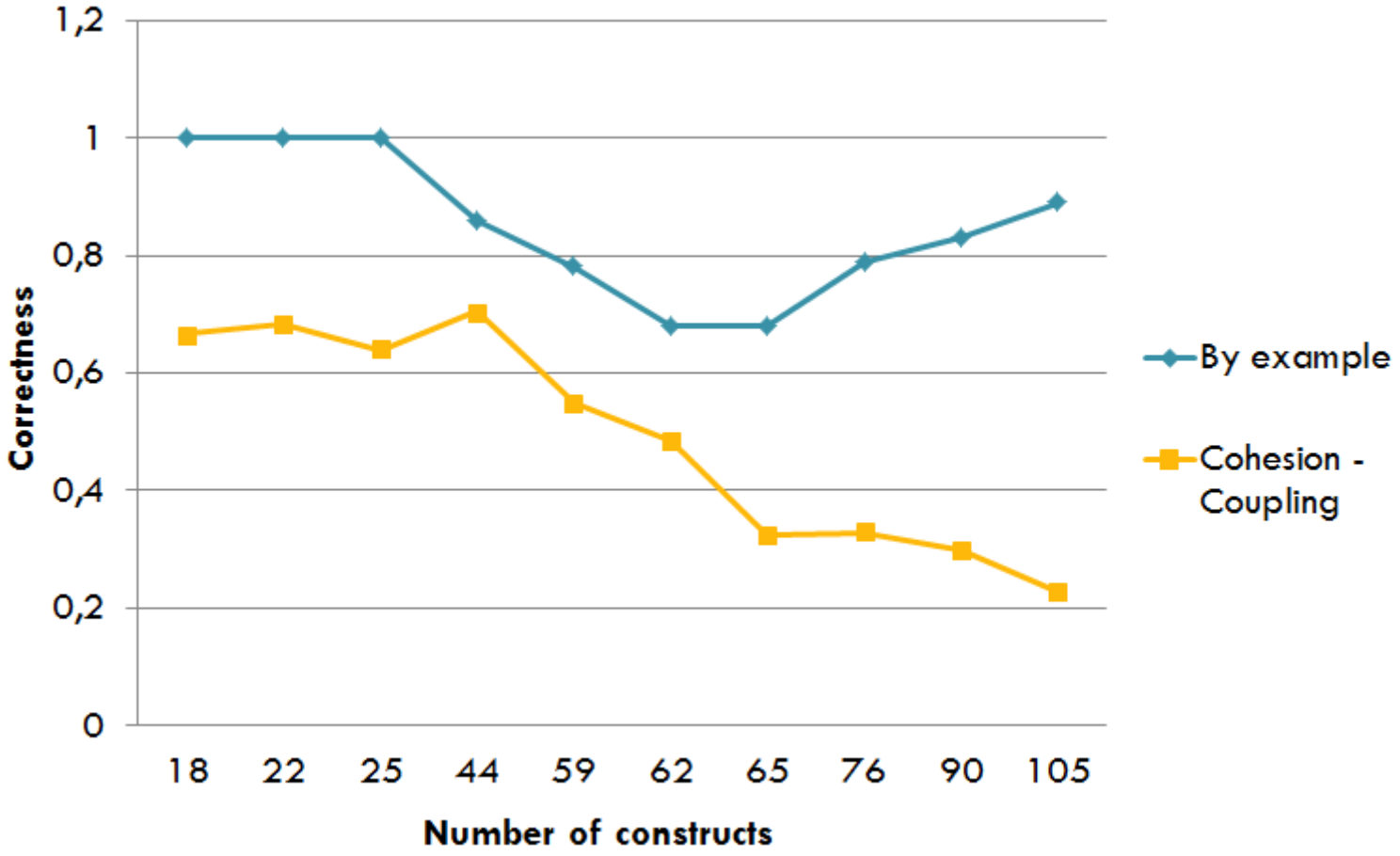
[1] source : Planet Source Code website

10-Fold Cross Validation Results

C Program	Num. of elements	Fitness	Correctness
C Prg. 1	18	0,92	100%
C Prg. 2	22	0,89	100%
C Prg. 3	25	0,91	100%
C Prg. 4	44	0,86	86%
C Prg. 5	59	0,89	78%
C Prg. 6	62	0,74	68%
C Prg. 7	65	0,78	66%
C Prg. 8	76	0,89	79%
C Prg. 9	90	0,85	83%
C Prg. 10	105	0,88	89%
Avg.	57	0.82	85%

Correlation of 0.82 between the our objective function and the correctness of a solution

Comparison with the Structural Hypothesis



Conclusion

- Novel approach for the software clustering problem
  - Clustering by example
- Could be used for various software clustering problems
- Case study: object identification
  - Encouraging results
  - Limitation: need for examples
- Validation with industrial systems
- More details in

M. Faunes, M. Kessentini and H. Sahraoui, Deriving High-Level Constructs in Legacy Software using Example-Driven Clustering, CASCON 2011