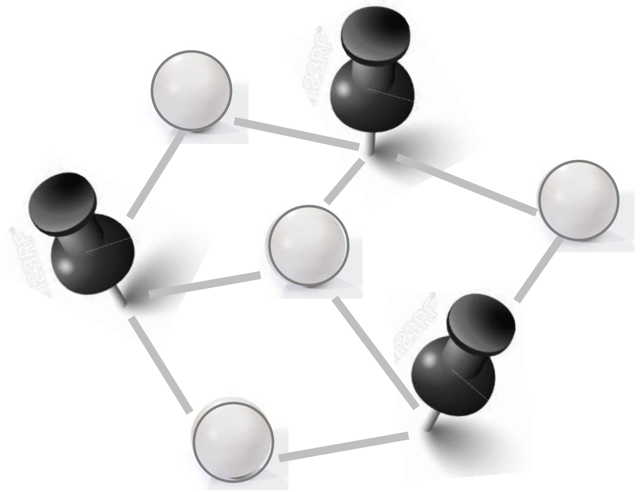
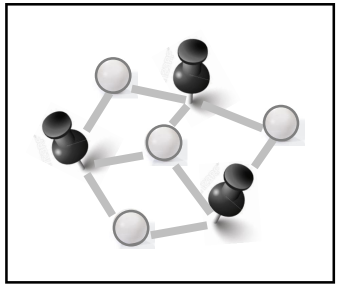
Planar Drawings of Fixed-Mobile Bigraphs



M. Bekos, F. De Luca, W. Didimo, T. Mchedlidze,
M. Nöllenburg, A. Symvonis, I.G. Tollis



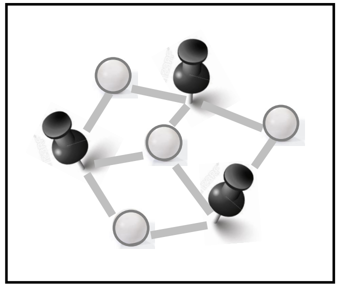
Problem

FM-bigraph: $G = (V_f, V_m, E)$ bipartite graph

- ● V_f fixed vertices (predefined locations)
- ○ V_m mobile vertices (can be freely placed)

K-bend FM-bigraph problem: Does G admit a *planar k-bend drawing*, i.e., a crossing-free drawing with at most k bends per edge ($k \geq 0$)?

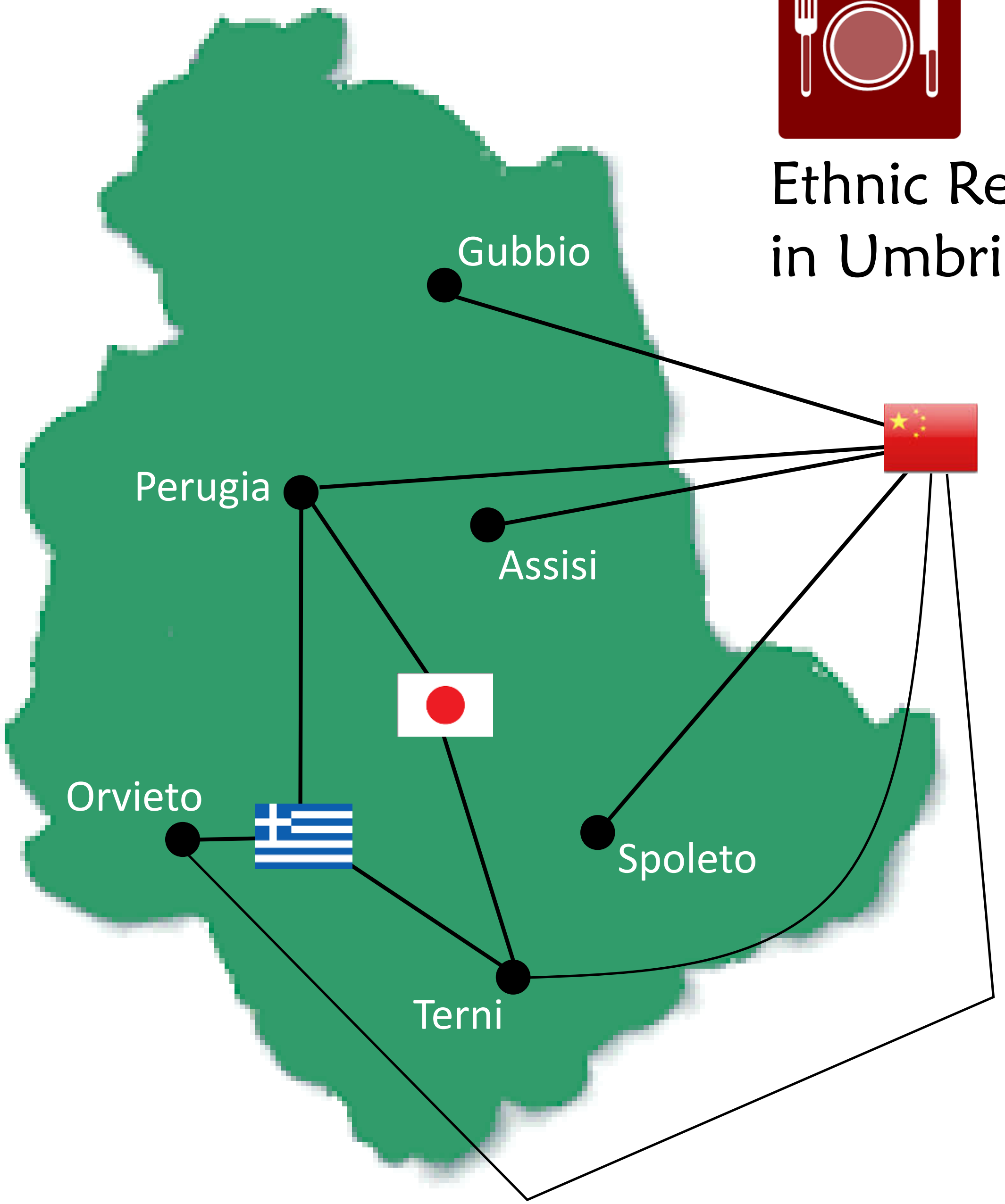
bend number of G: The minimum k for which G admits a planar k-bend drawing

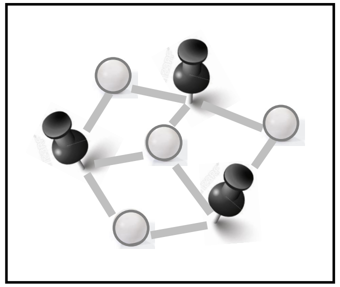


Motivation

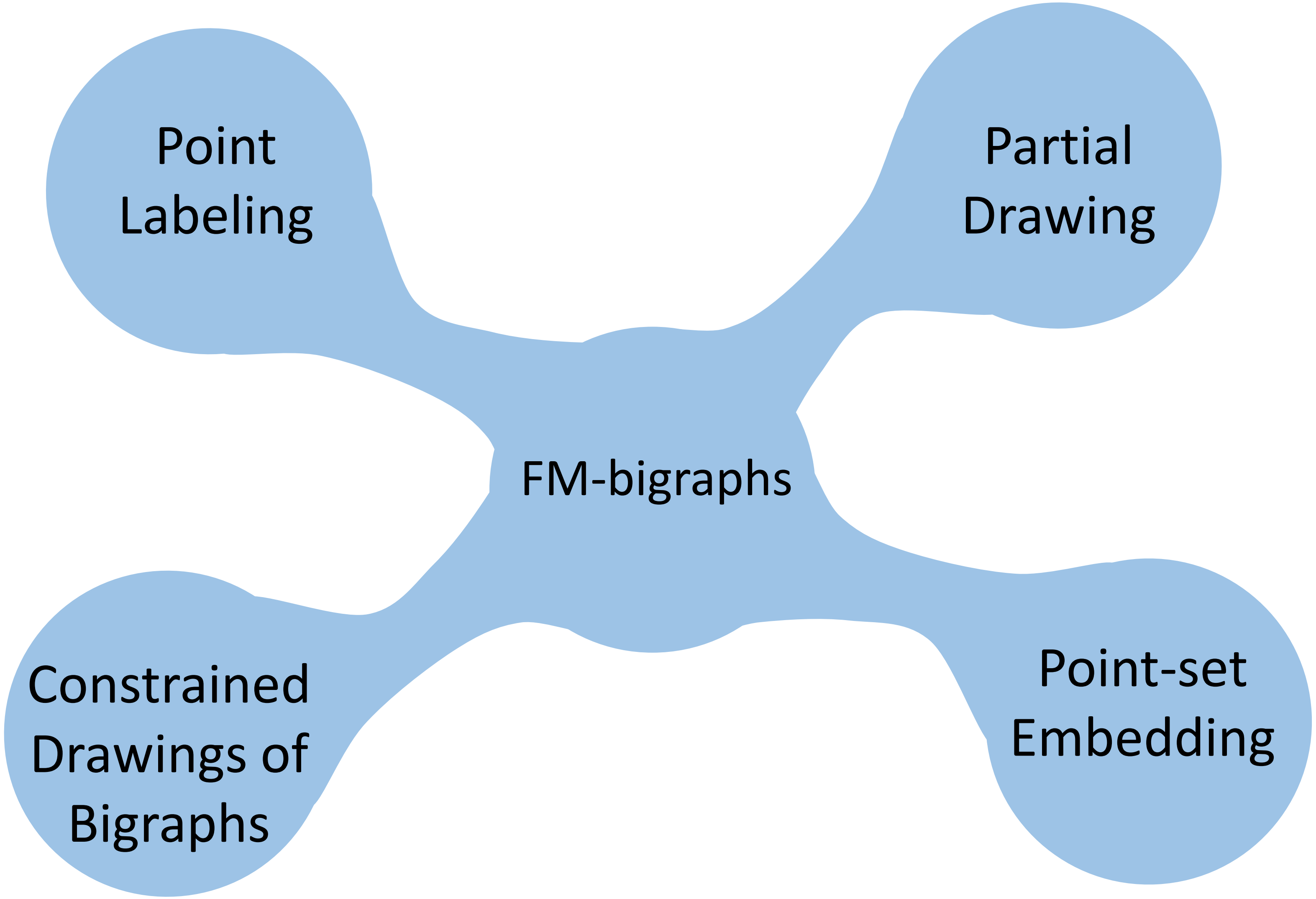
- Fixed vertices = geographic locations
- Mobile vertices = simple attributes
- Attributes are *connected* to their locations

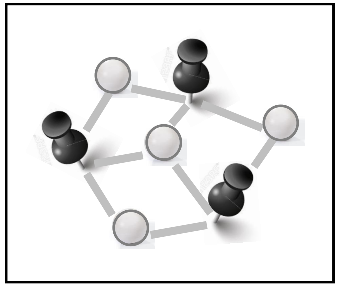
Ethnic Restaurants in Umbria



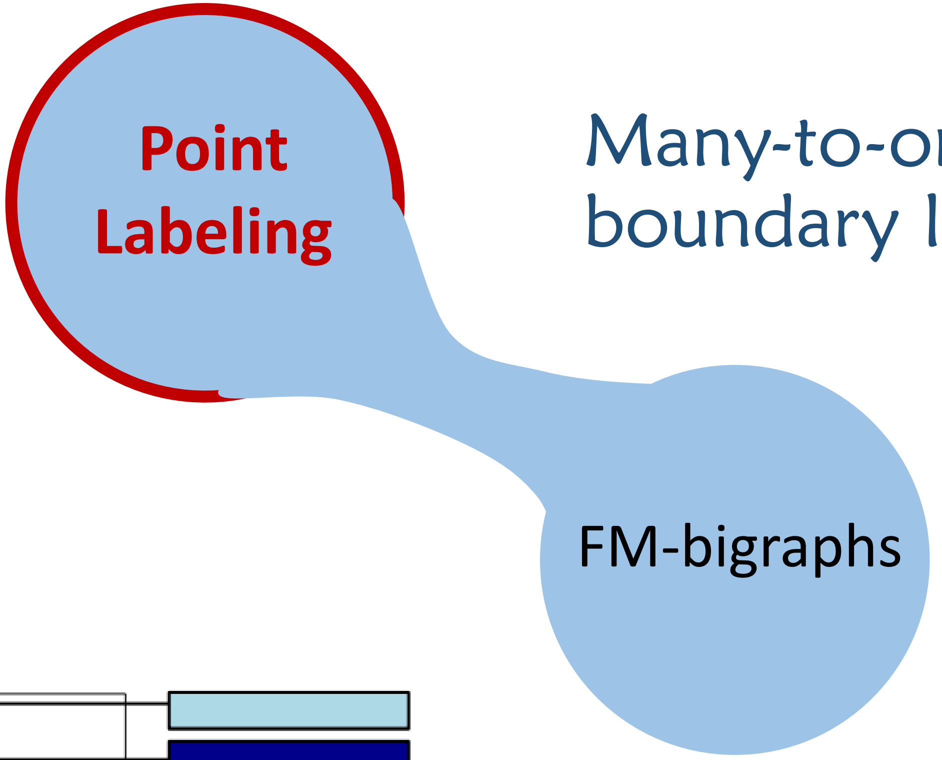


Related work



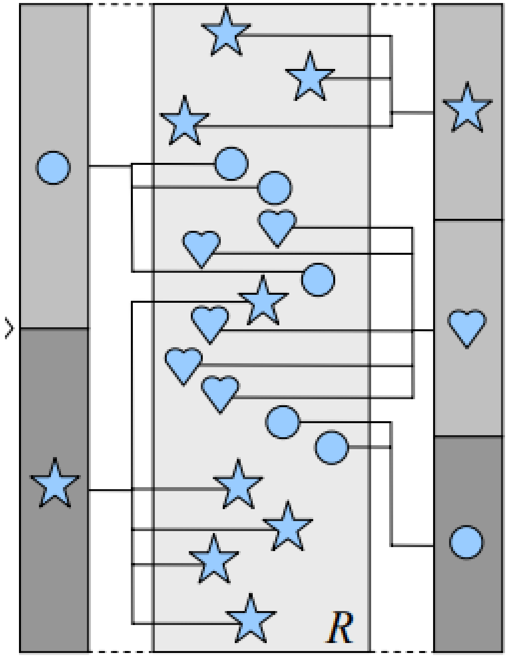


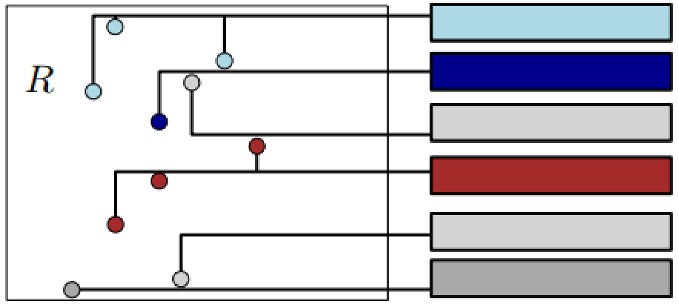
Related work: Point Labeling



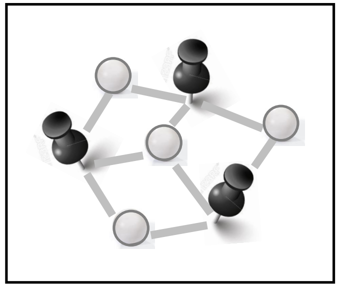
Many-to-one
boundary labeling

Lin, PacificVis 2010



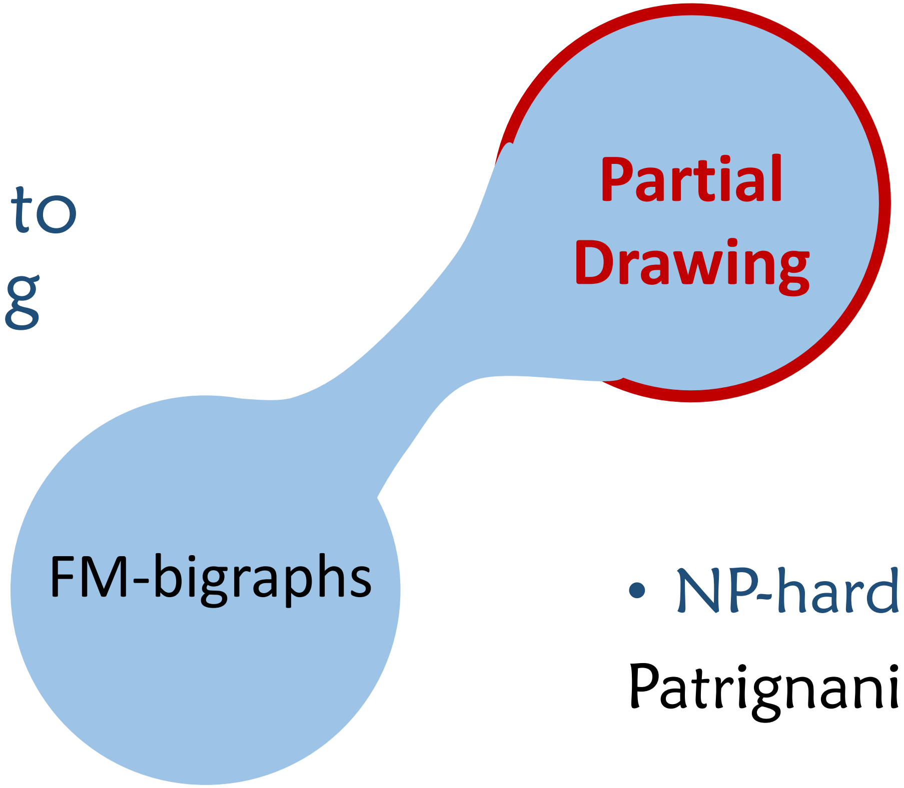


Bekos et al., JGAA 2015



Related work: Partial Drawing

Extending a partial drawing to a planar straight-line drawing

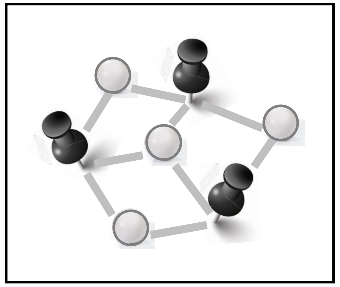


- NP-hard in the general case

Patrignani, IJFCS 2006

- Tractable for restricted cases

e.g., prescribed outer face, convex drawings

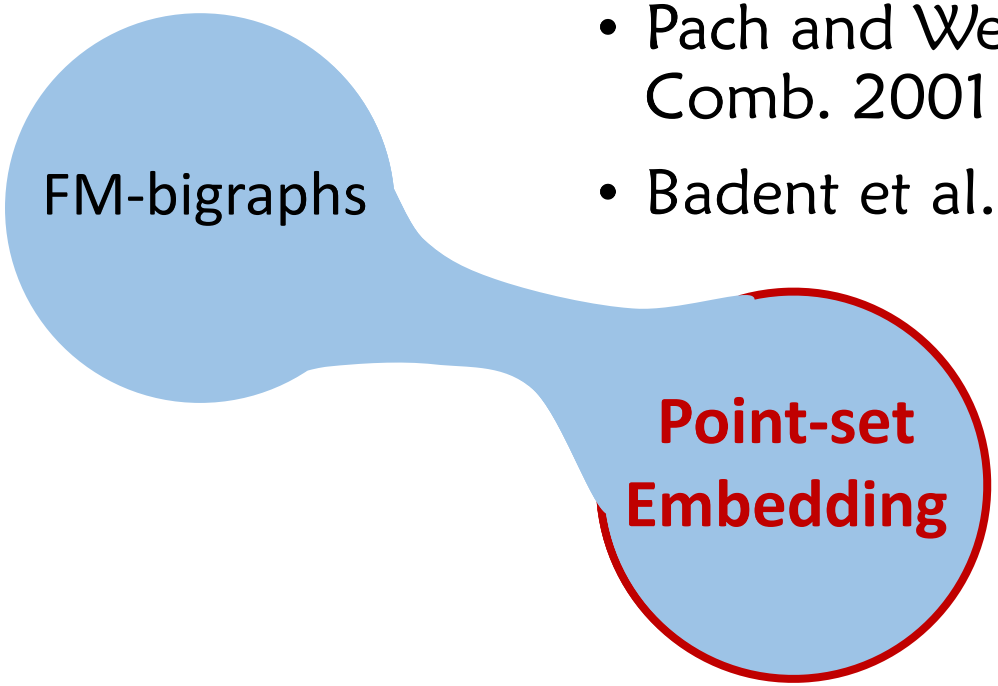


Related work: Point-set Embedding

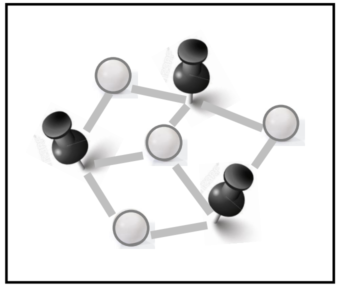
Each vertex is mapped to a specific point or to a finite set of points

Every planar graph has a planar embedding at fixed vertex locations (O(n) bends per edge)

- Pach and Wenger, Graphs and Comb. 2001
- Badent et al., TCS 2008



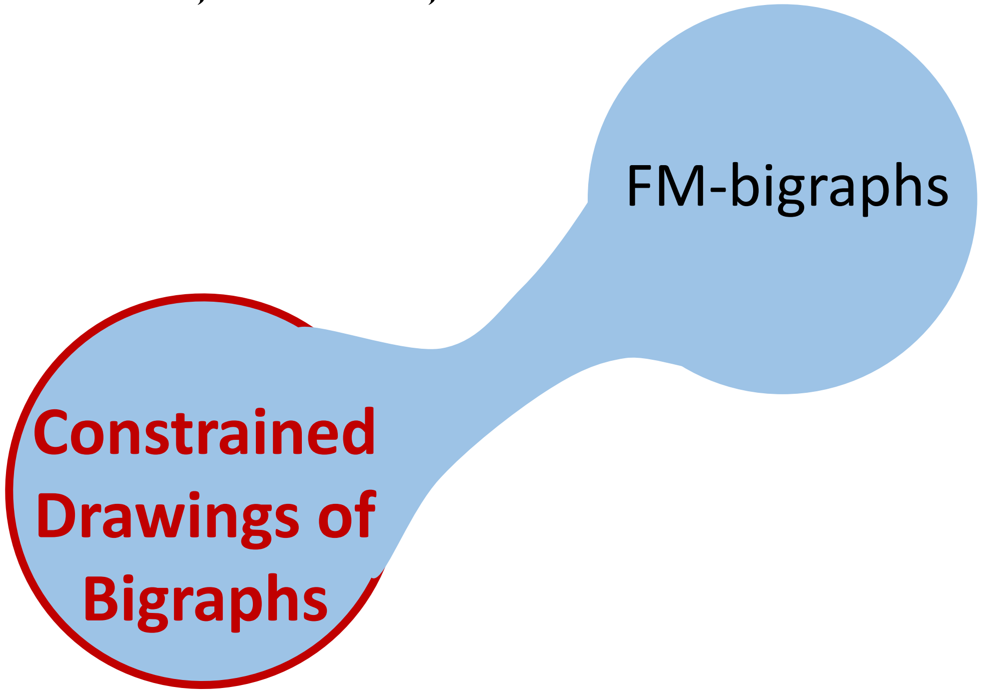
Consequence: Any planar FM-bigraph has a planar O(n)-bend drawing

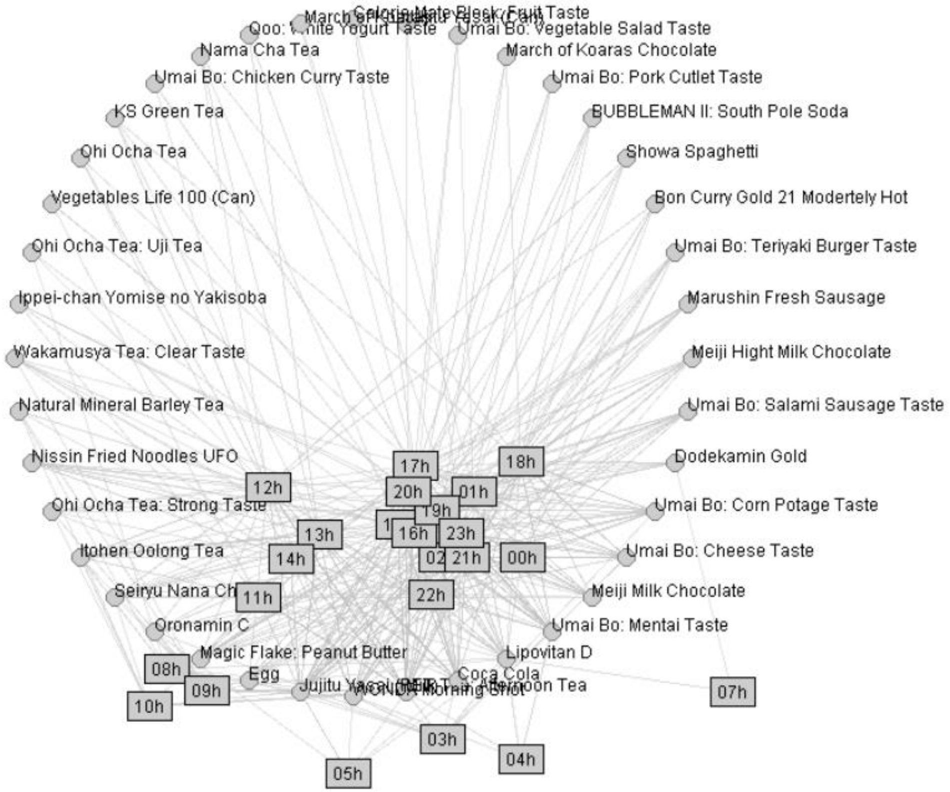


Related work: Constrained Drawings

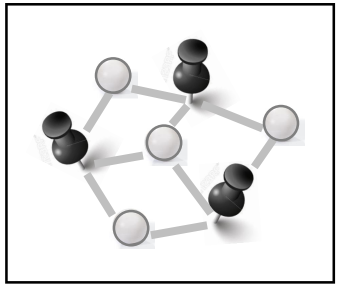
Drawing planar partitions
(vertices on predefined lines or plane regions)

- Biedl, SoCG 1998
- Biedl, Kaufmann, Mutzel, WG 1998



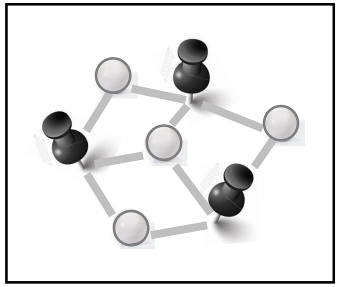


Anchored Maps
K. Misue, IEICE Trans. 2008



Contribution

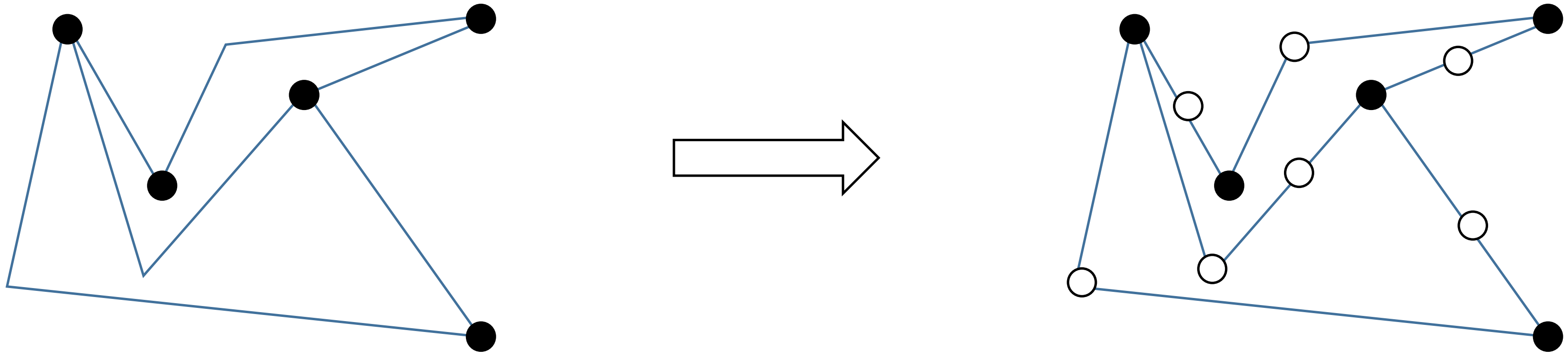
- Result 1. Computing the bend number of an FM-bigraph is NP-hard (connection with point-set embedding)
- Result 2. When mobile vertices lie in the convex hull (CH) of their neighbors, testing the existence of 0-bend drawings: (i) is in NP; (ii) is in P (tractable) if the intersection graph of the CHs is a cactus
- Result 3. A practical model for 1-bend drawings of FM-bigraphs, inspired by the boundary labeling approach, with polynomial-time algorithms

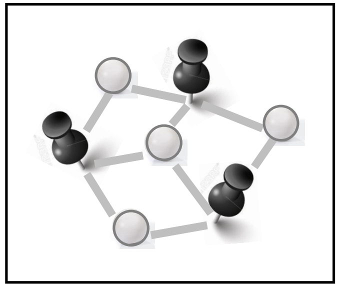


Result 1 – NP-hardness

Theorem. The 0-bend FM-bigraph problem is NP-hard, even if each vertex has degree at most two

Proof: Reduction from 1-bend point-set embedding with mapping (which is NP-hard – Goaoc et al., DCG 2009)

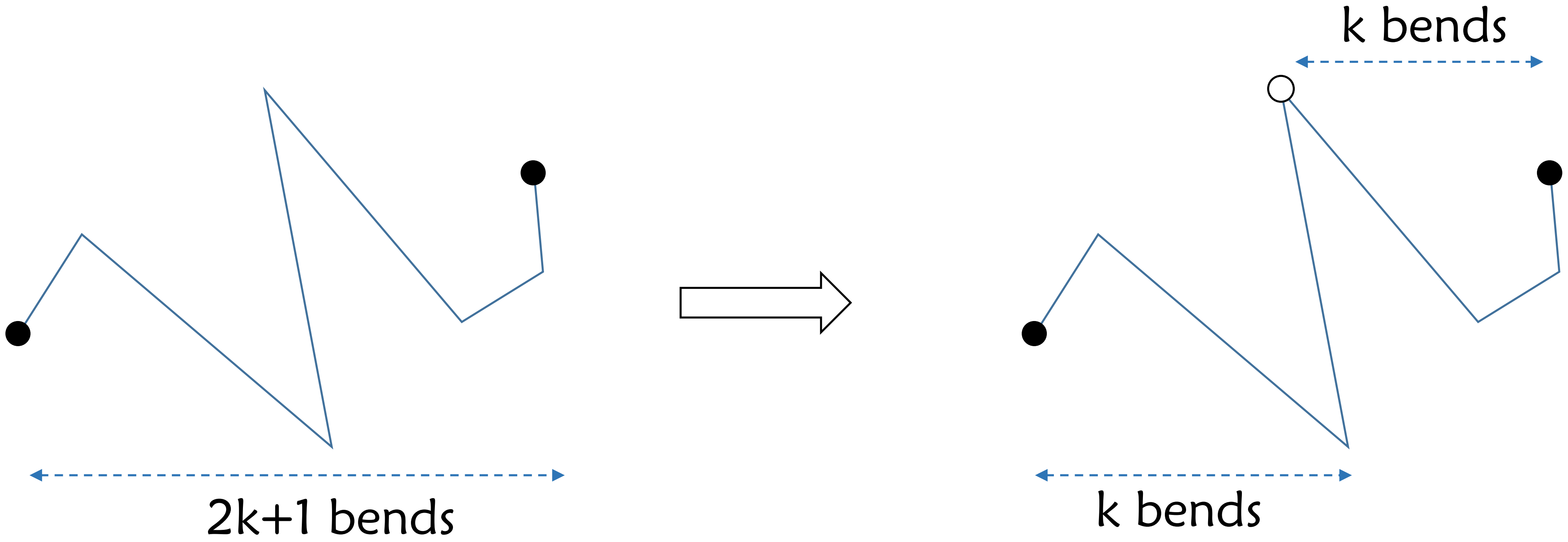


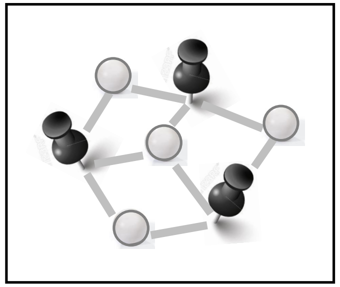


Result 1 – More in general

Theorem. The k-bend FM-bigraph problem is at least hard as the (2k+1)-bend point-set embedding with mapping

Proof: Same reduction

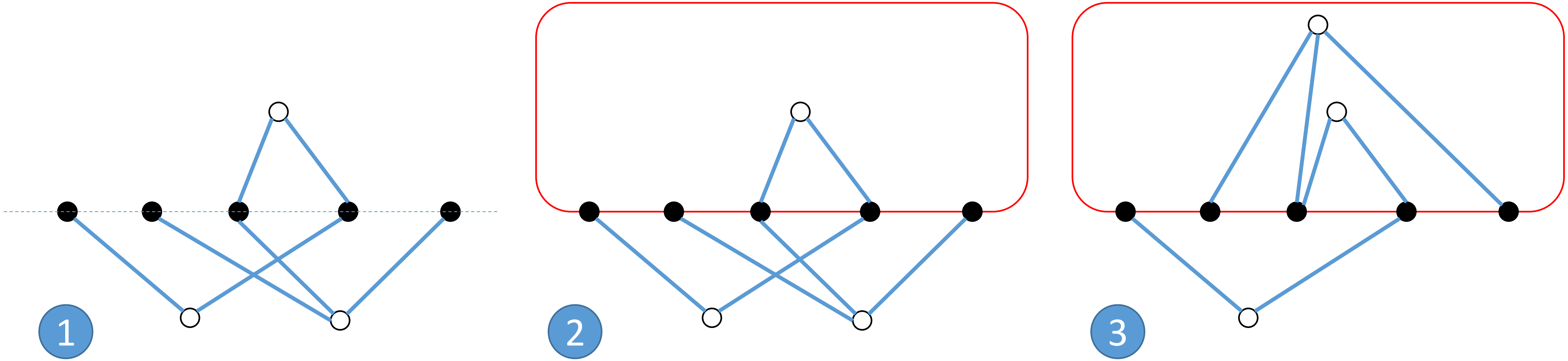


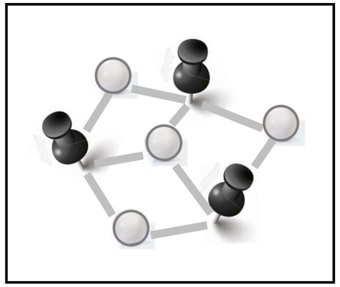


Result 1 – Special case

Theorem. If all fixed vertices are collinear, the 0-bend FM-bigraph problem is linear-time solvable

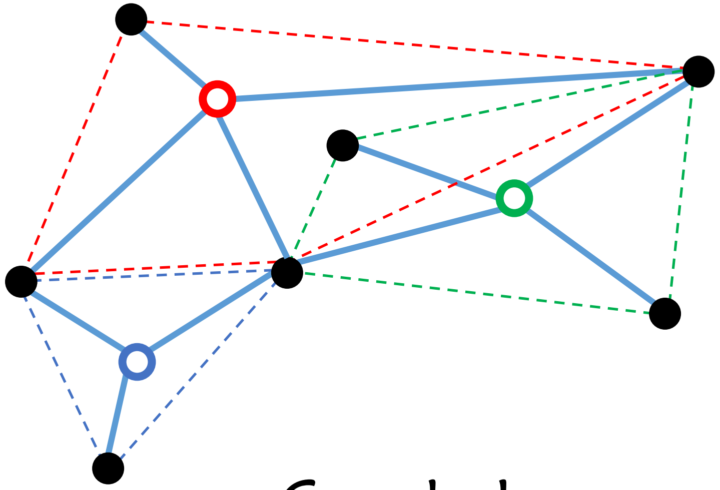
Proof: Reduce to planarity testing



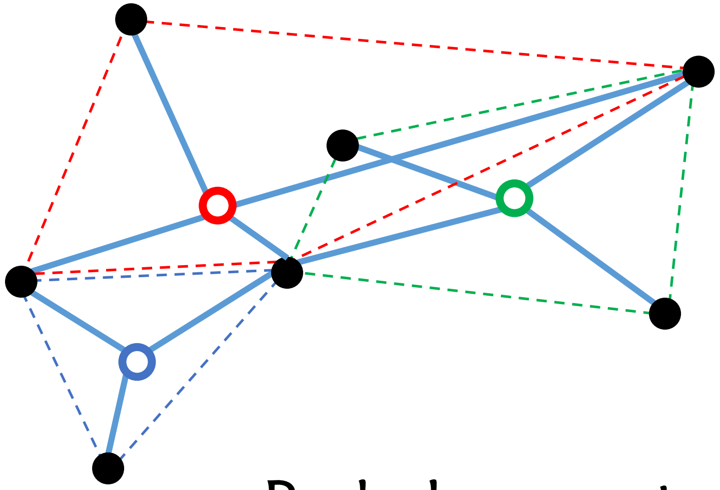


Result 2 – Convex-hull restriction

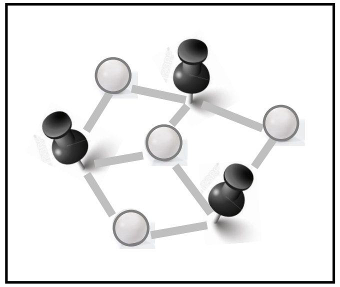
CH restriction for 0-bend drawings: fixed vertices in general position and every mobile vertex in the CH of its (fixed) neighbors



Good placement

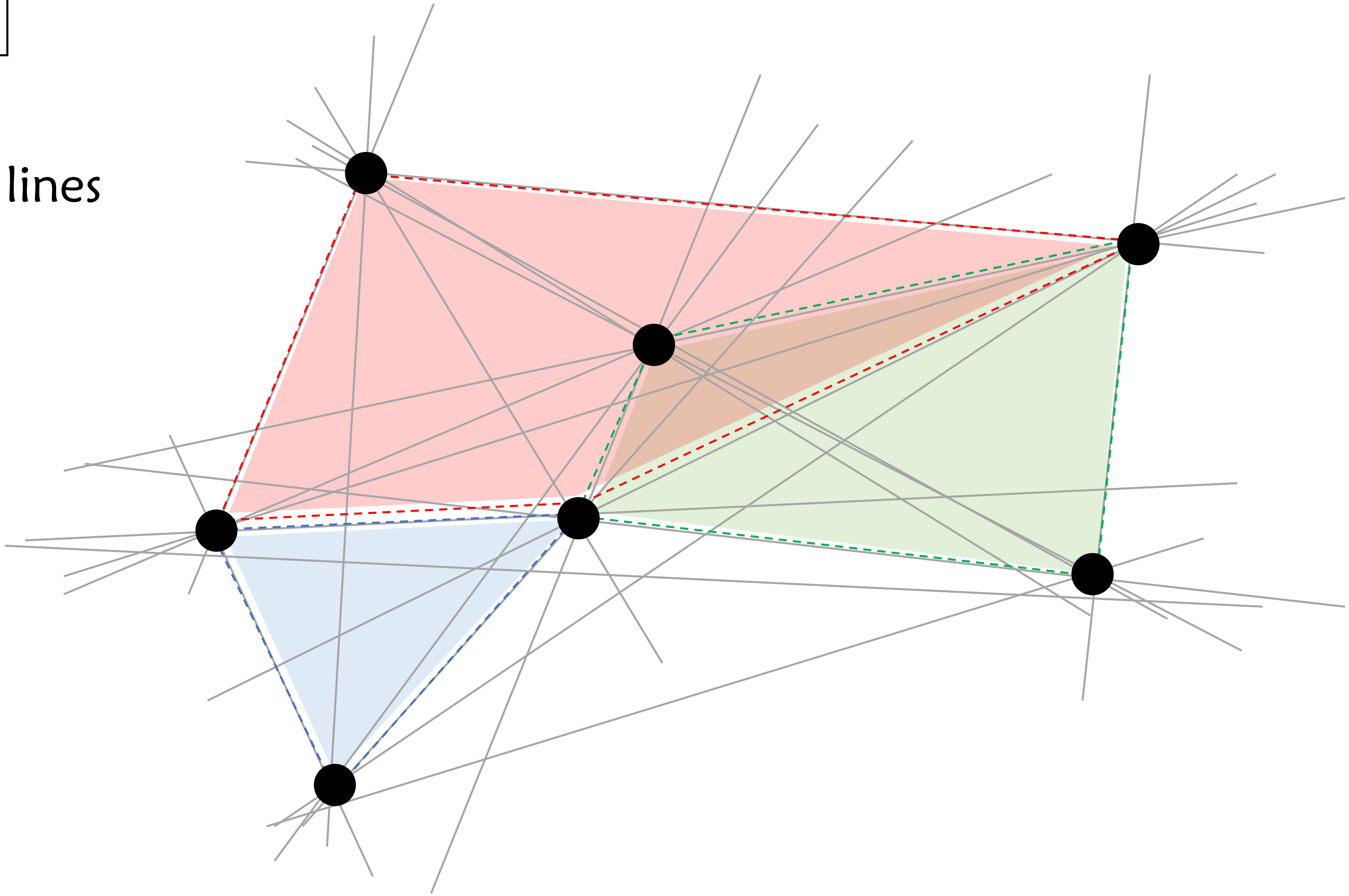


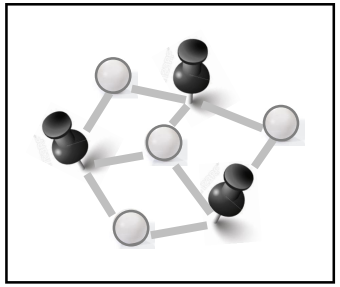
Bad placement



Result 2 – Line arrangement

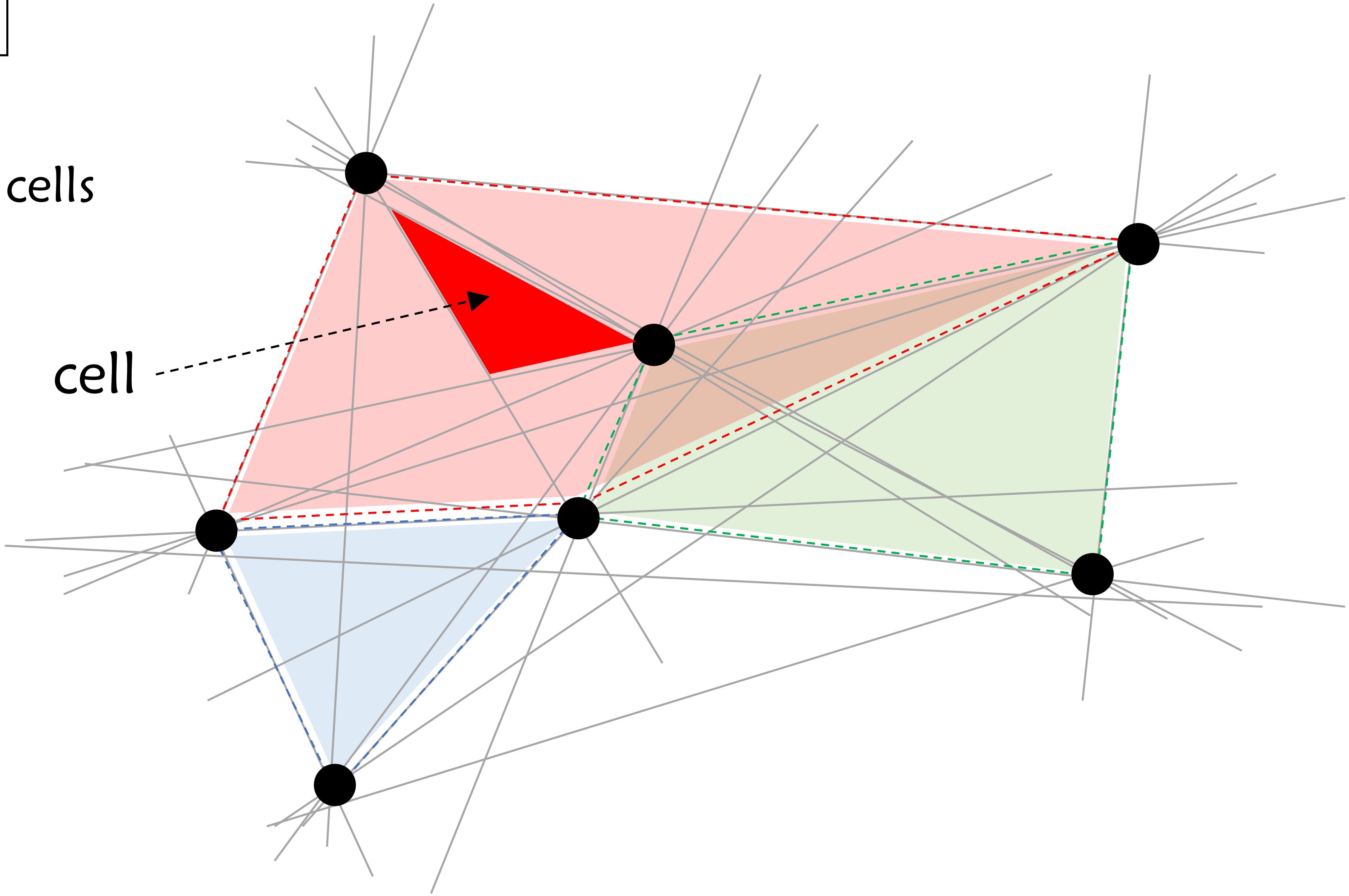
$O(|V_f|^2)$ lines

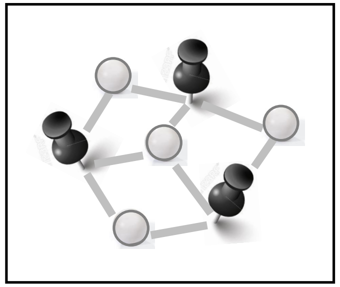




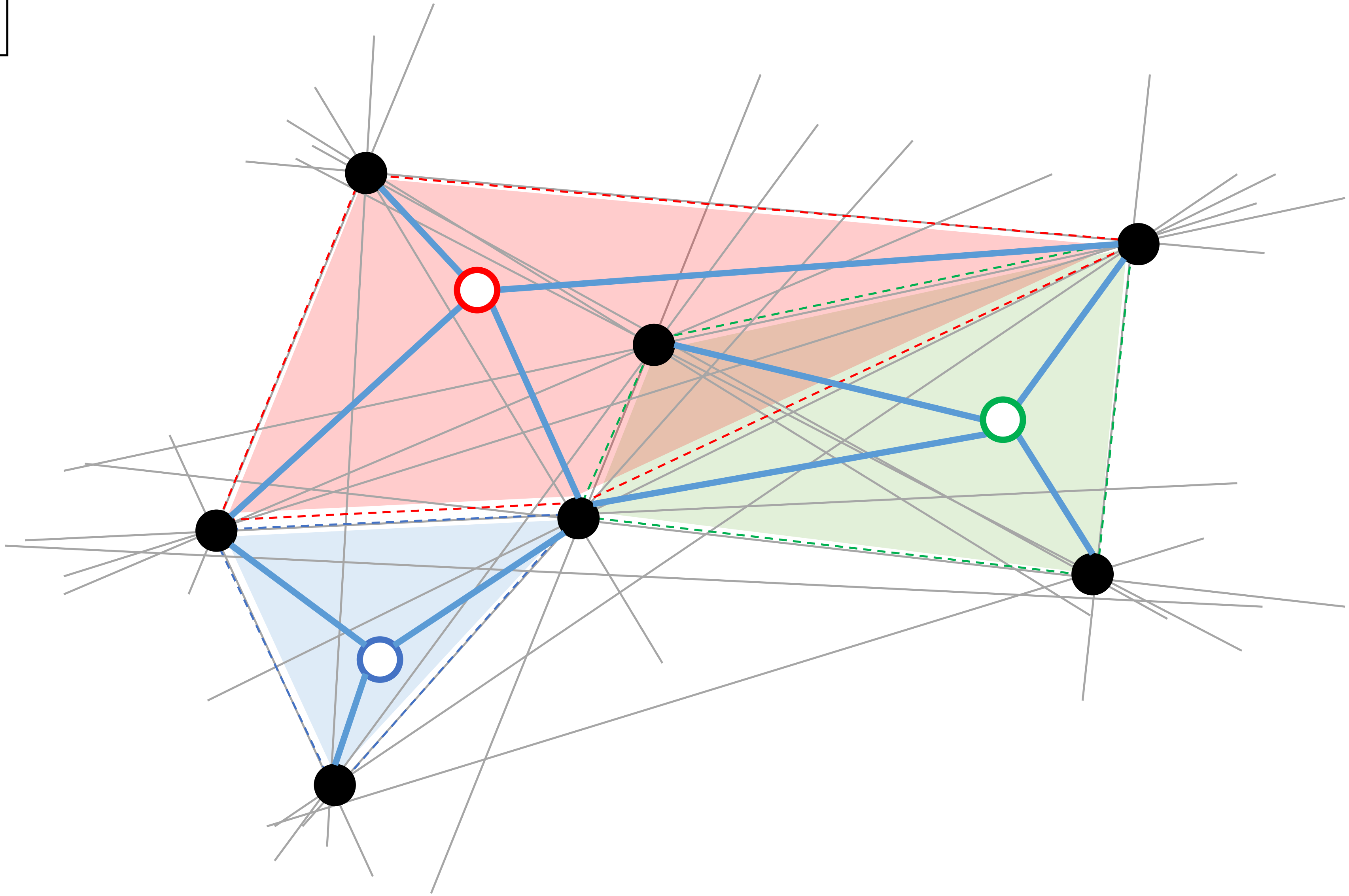
Result 2 – Line arrangement

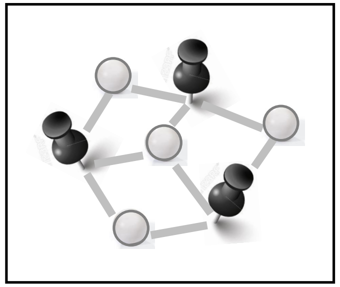
$O(|V_f|^4)$ cells



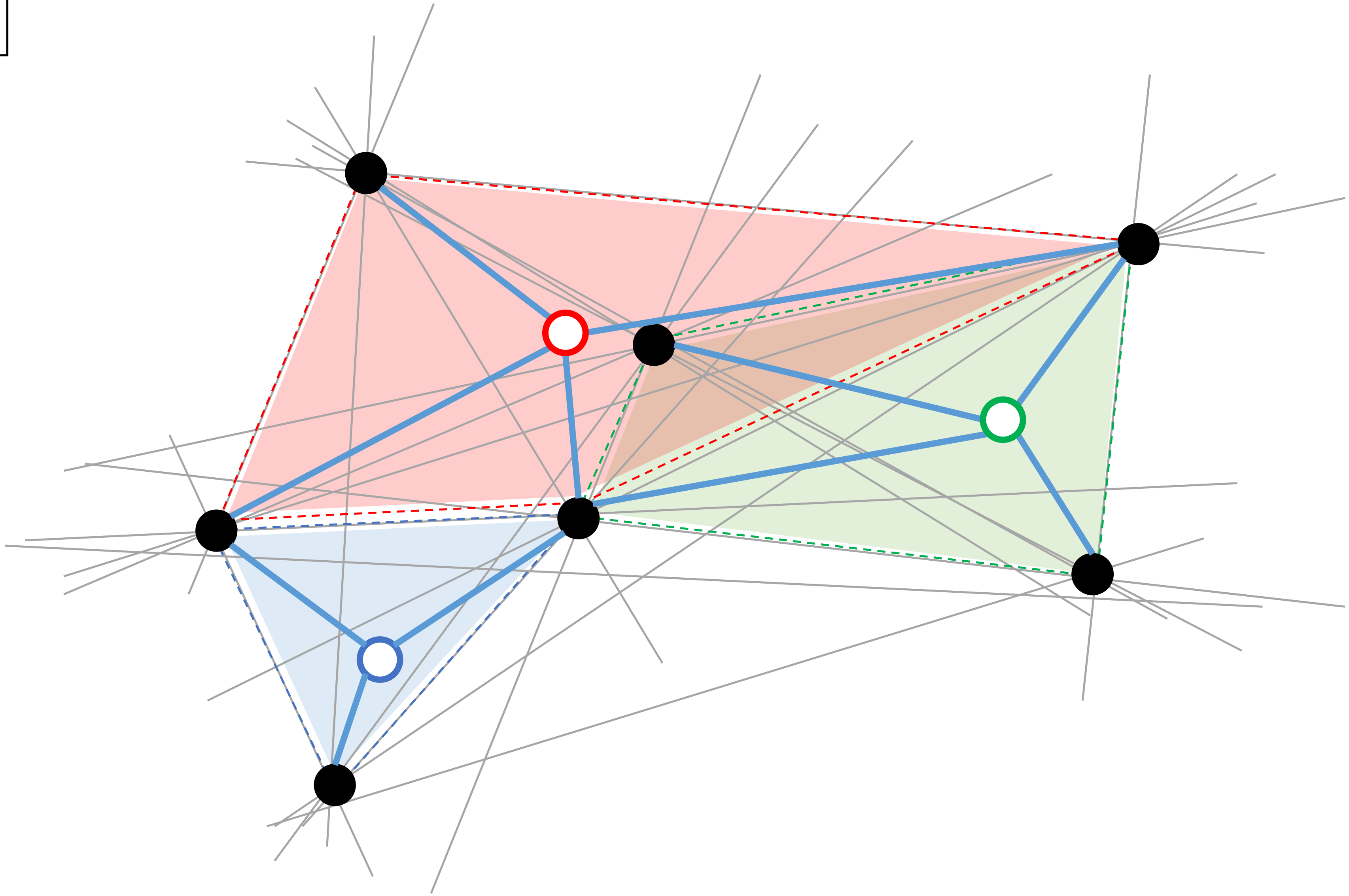


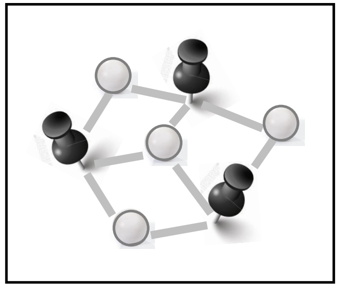
Result 2 – Line arrangement





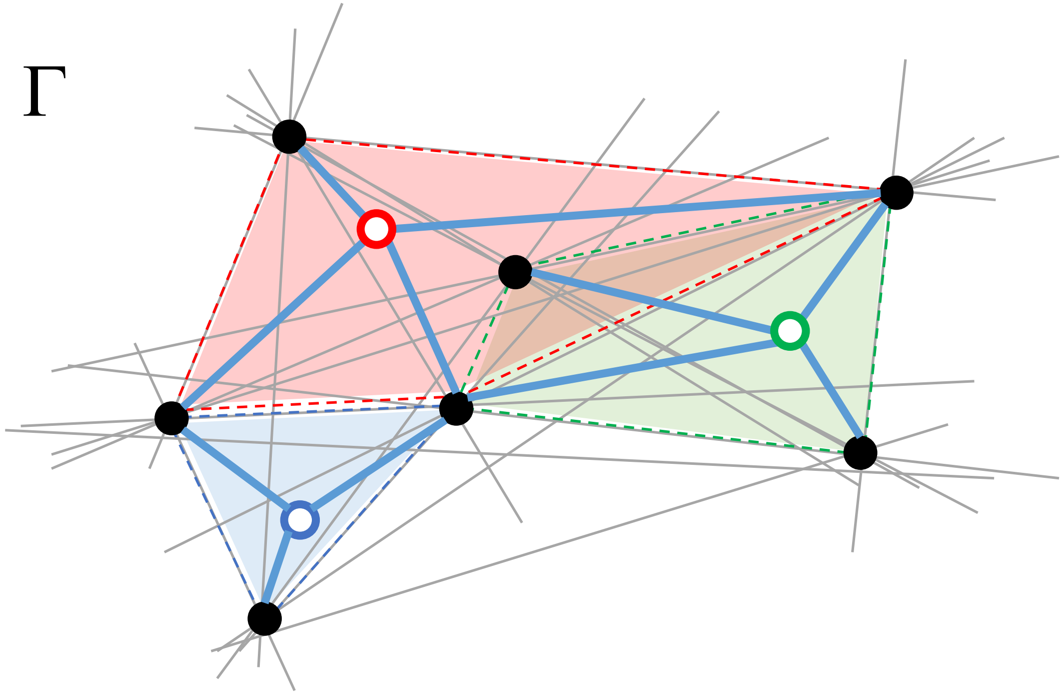
Result 2 – Line arrangement

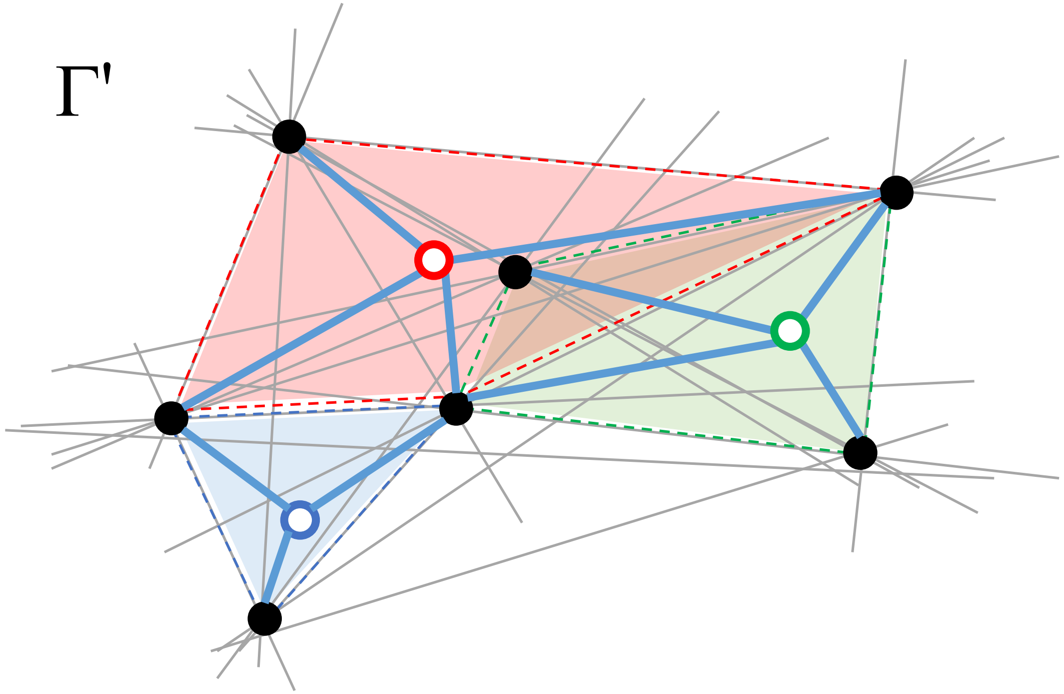


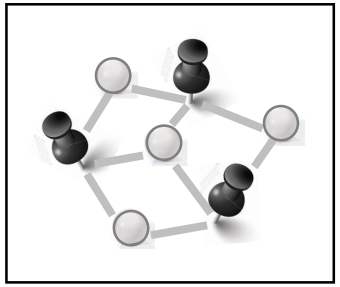


Result 2 – Discretization

Lemma. Let Γ and Γ' be two 0-bend drawings of G that differ only for the position of a (mobile) vertex. If this vertex is in the same cell in the two drawings, then Γ' is planar ⇔ Γ is planar





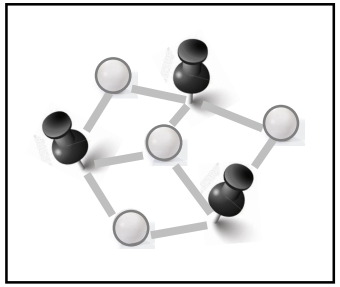


Result 2 – Membership in NP

Theorem. The 0-bend FM-bigraph problem belongs to NP if each mobile vertex is restricted to lie in the convex hull of its neighbors

Proof. A non-deterministic algorithm guesses an assignment of the $|V_m|$ mobile vertices to the $O(|V_f|^4)$ cells; for each assignment, the algorithm (deterministically) checks planarity in $O(|V_f|^2)$ time.

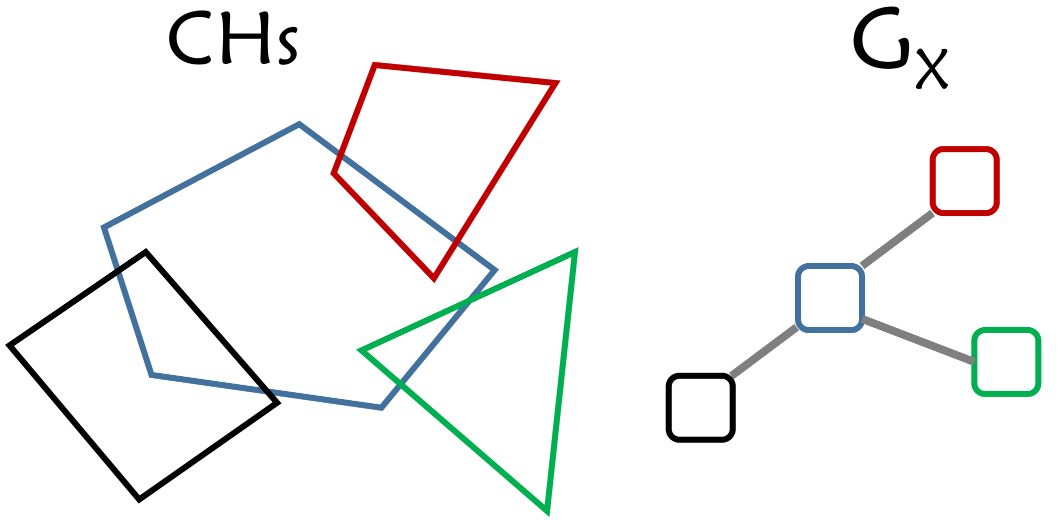
Result 2 – From NP to P

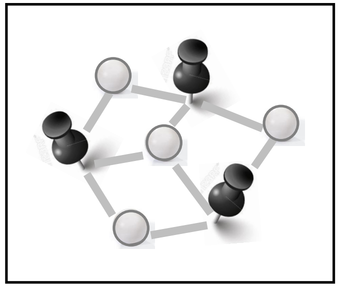


Result 2 – Support graphs – G_X

G_X = intersection graph of all CHs

- CH(u) ↔ the CH of the neighbors of u

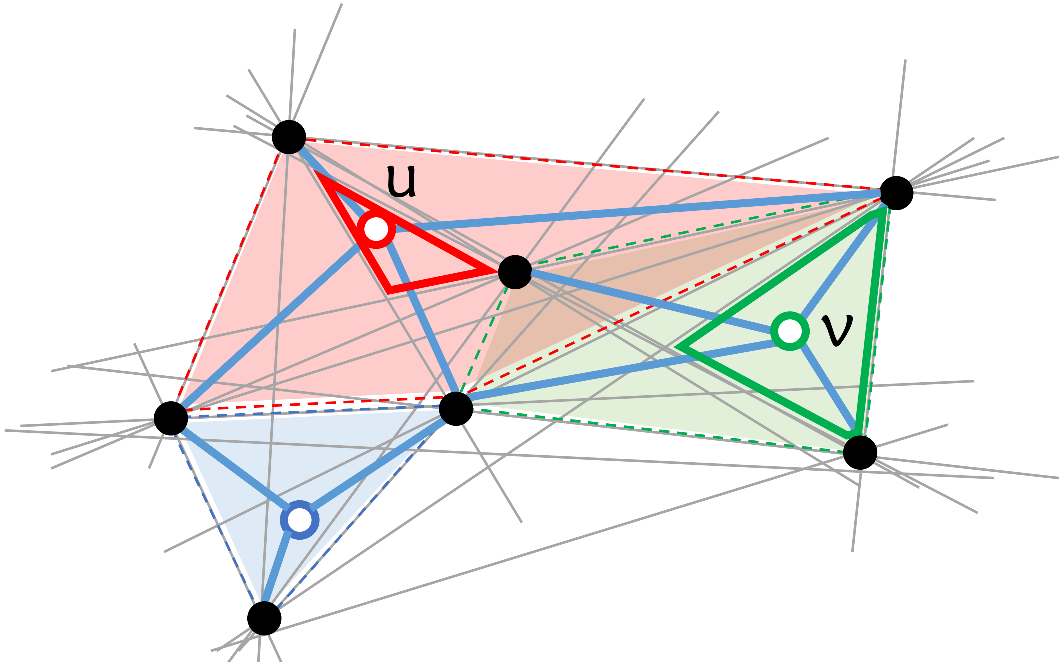


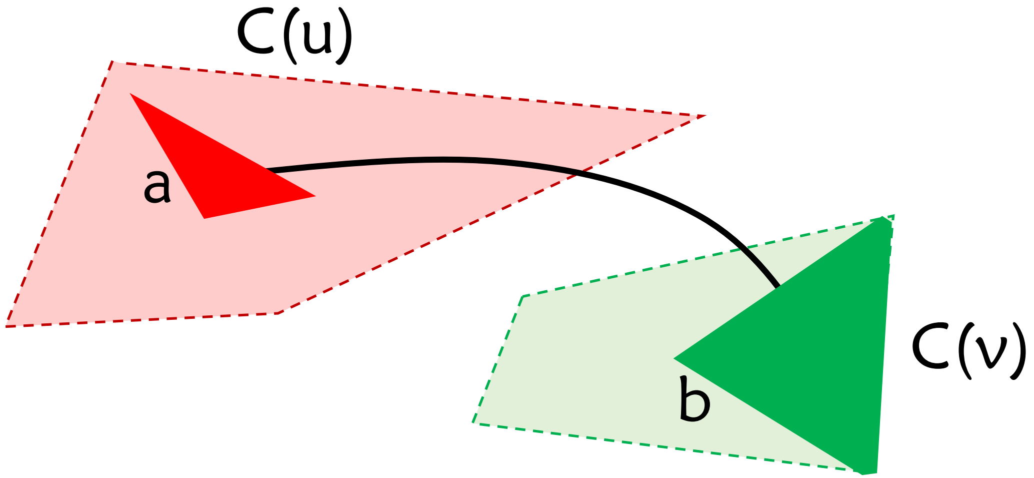


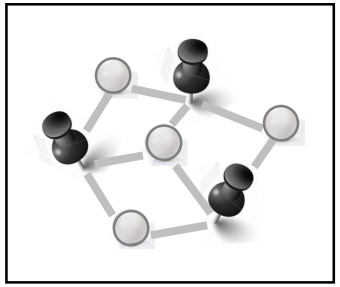
Result 2 – Support graphs – G_C

G_C = clustered graph

- cluster C(u) ↔ CH(u) (for each mobile vertex u)
- nodes of C(u) ↔ cells in CH(u)
- edge (a,b) ↔ a∈C(u), b∈C(v), CH(u)∩ CH(v) ≠ ∅, and placing u in a and v in b does not cause crossings



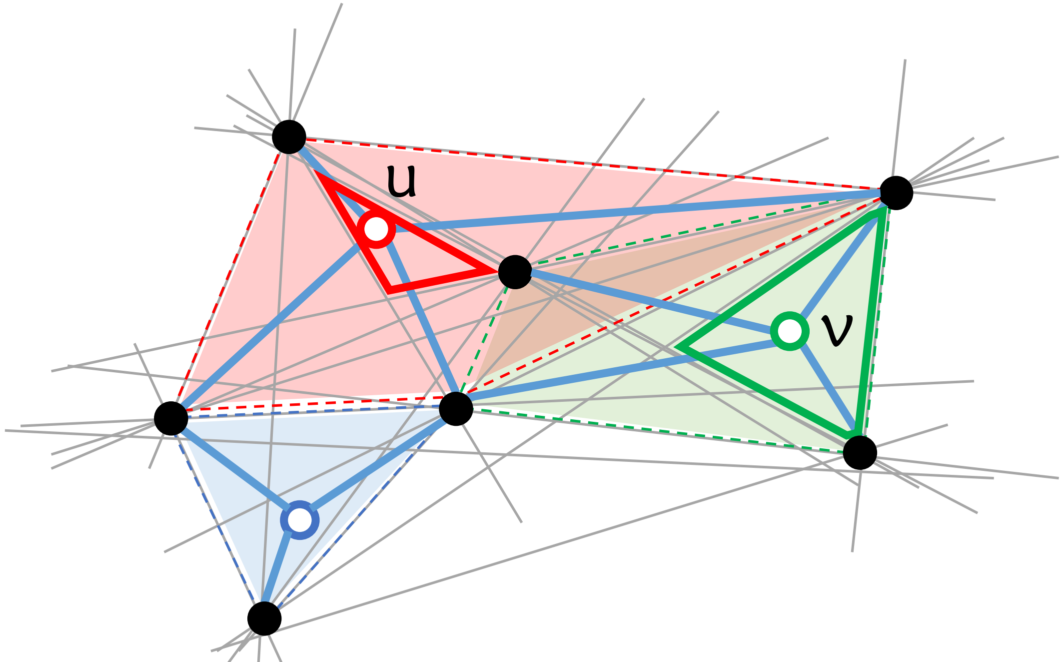


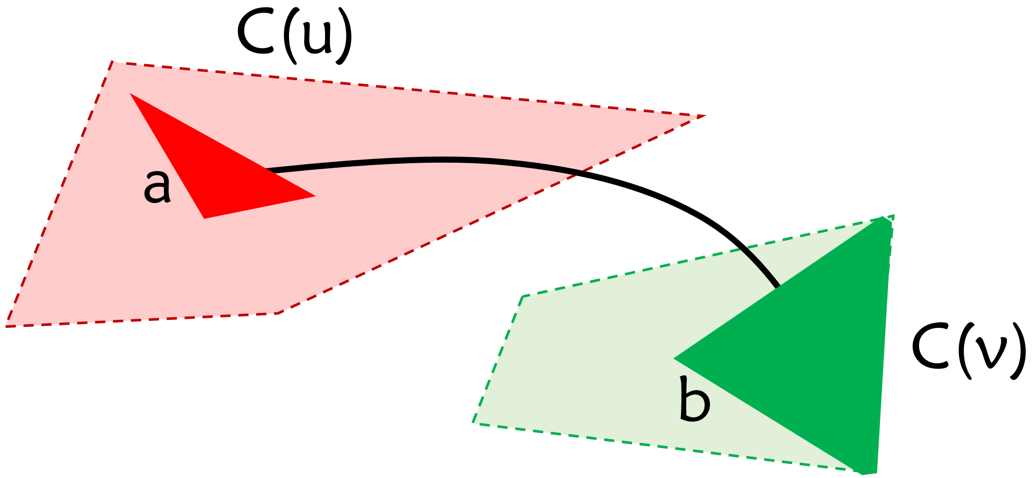


Result 2 – Support graphs – G_C

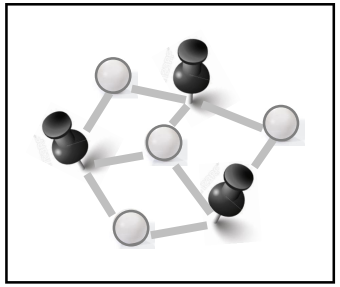
G_C = clustered graph

- cluster C(u) ↔ CH(u) (for each mobile vertex u)
- nodes of C(u) ↔ cells in CH(u)
- edge (a,b) ↔ a∈C(u), b∈C(v), CH(u)∩ CH(v) ≠ ∅, and placing u in a and v in b does not cause crossings



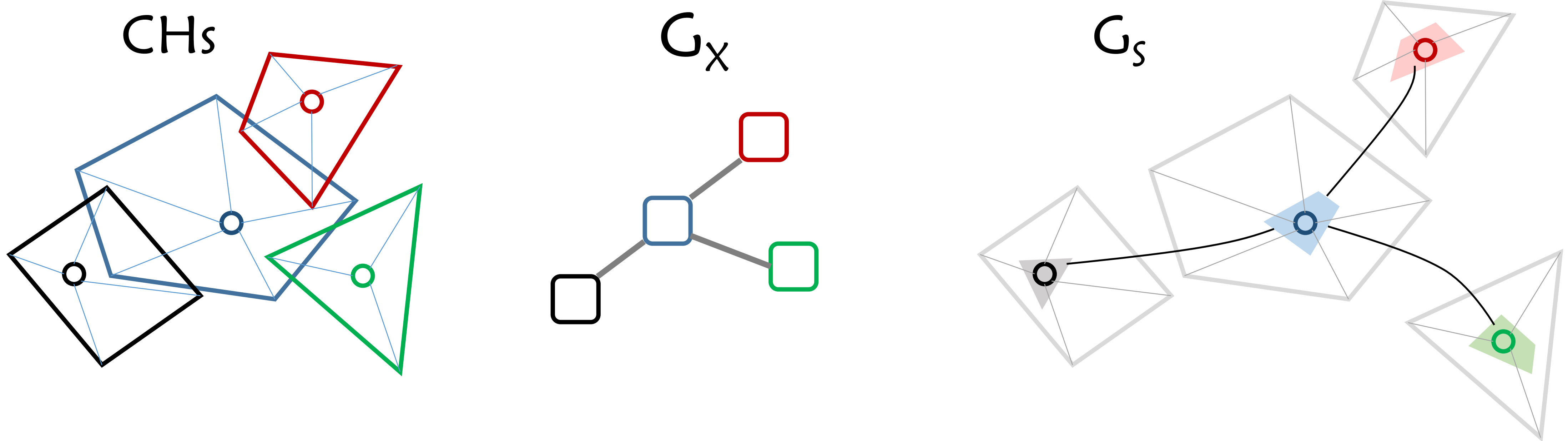


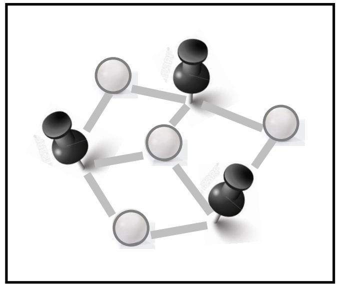
nodes = $O(|V_f|^4 \, |V_m|)$ edges = $O(|V_f|^8 \, |V_m|^2)$



Result 2 – Support graphs – G_S

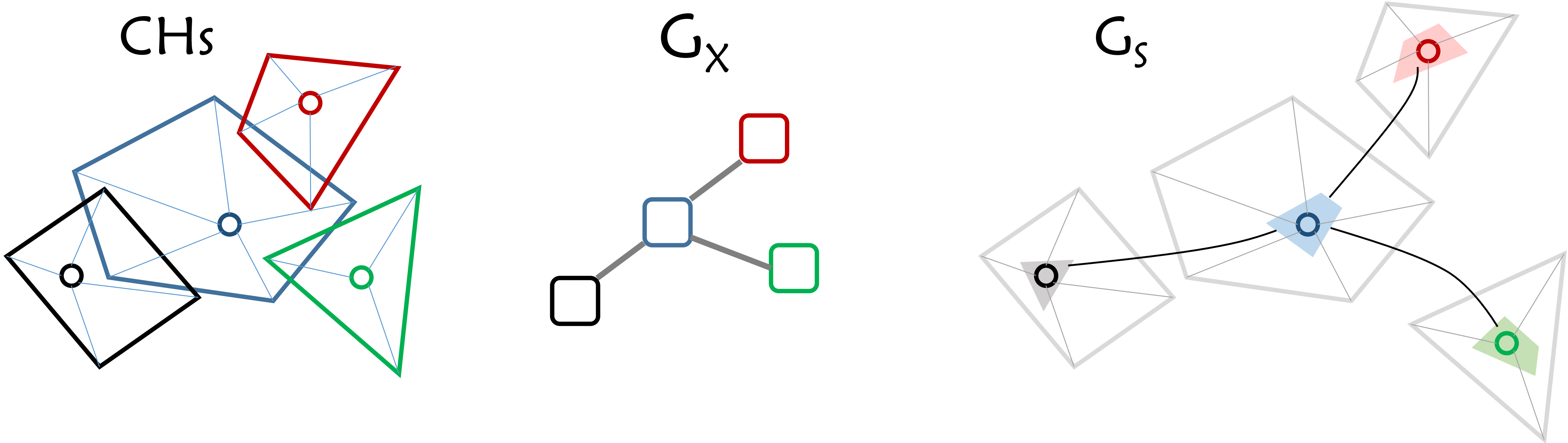
G_S = skeleton of G_C – subgraph of G_C induced by exactly one cell-node per cluster and isomorphic to G_X

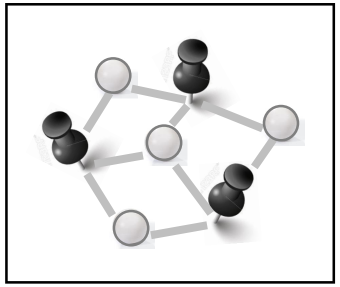




Result 2 – Characterization

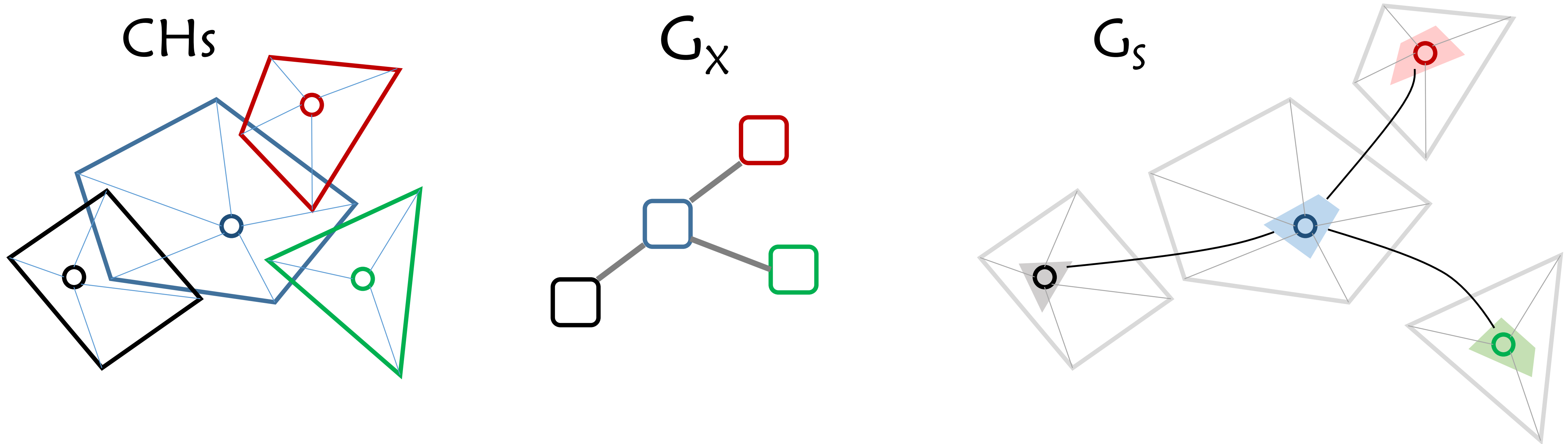
Theorem. An FM-bigraph G admits a planar 0-bend drawing in the CH restriction setting ⇔ there exists a skeleton G_S

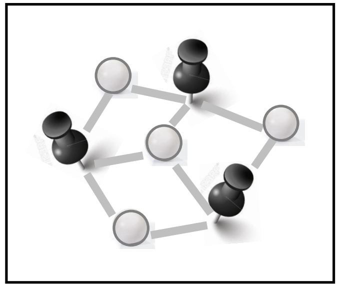




Result 2 – Hardness

It is in general NP-hard to decide whether a certain skeleton exists in a clustered graph defined as in our problem …. But the problem is tractable for specific types of G_X

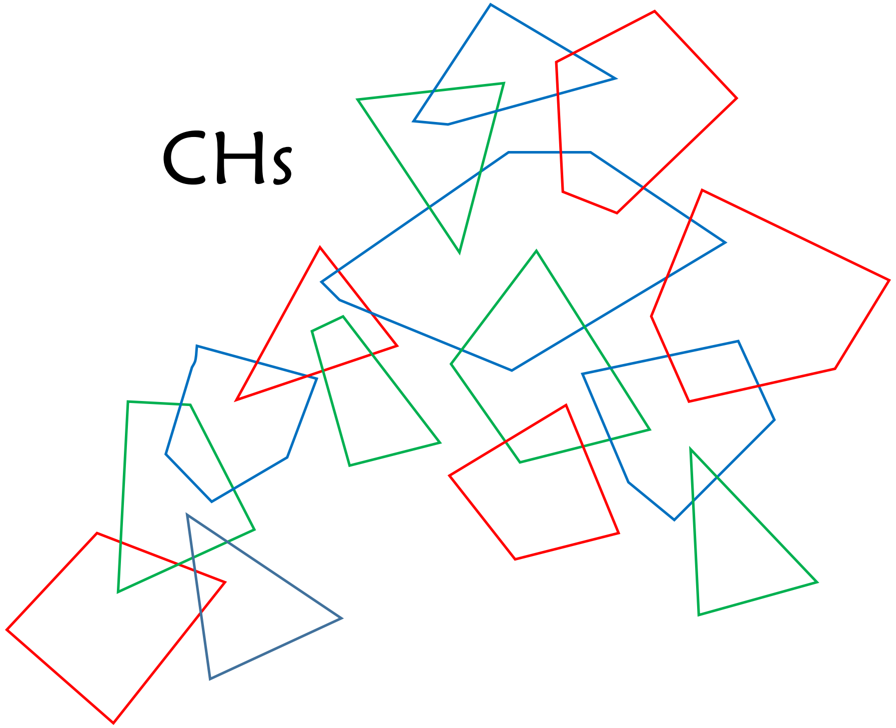




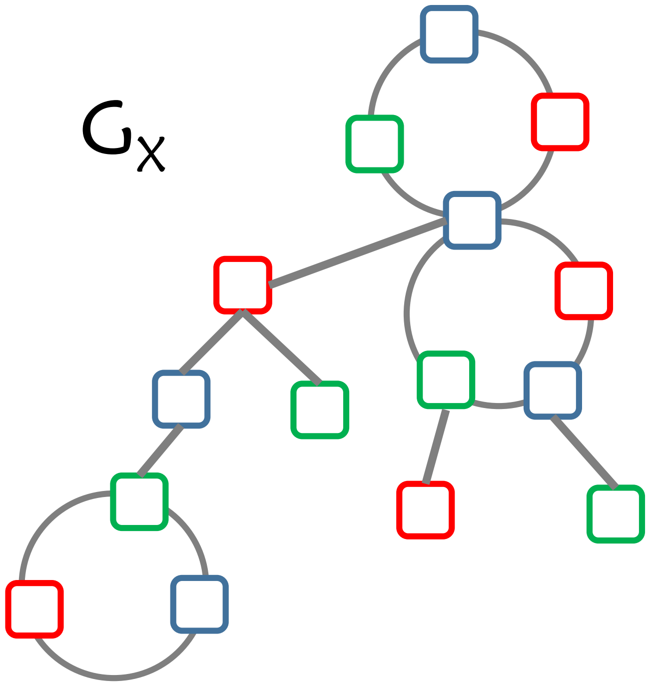
Result 2 – Tractability

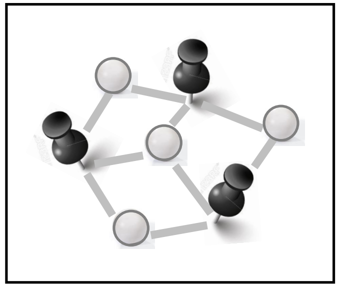
Theorem. If G_X is a cactus (or a forest of cacti), one can test in polynomial time whether G admits a planar 0-bend drawing

CHs

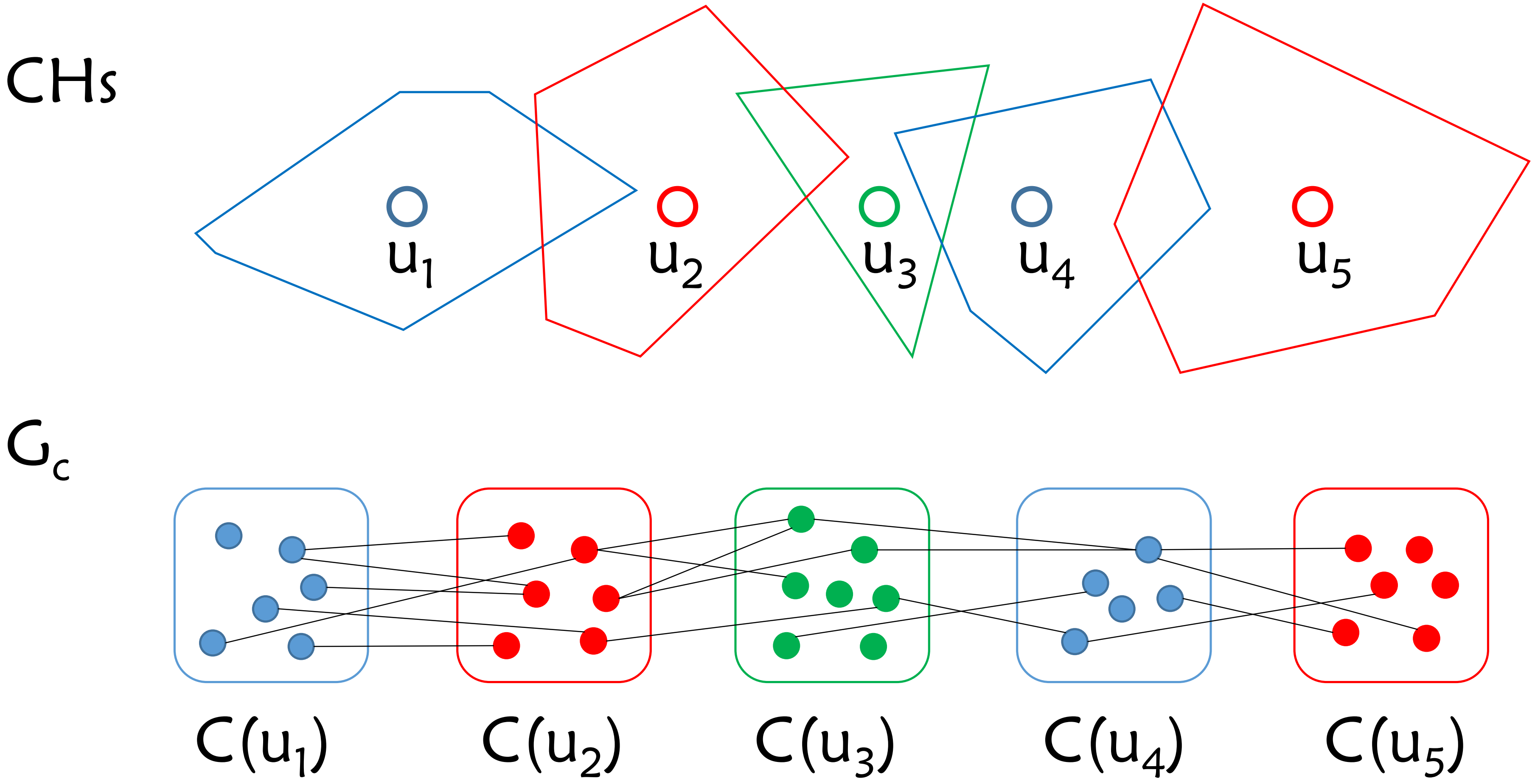


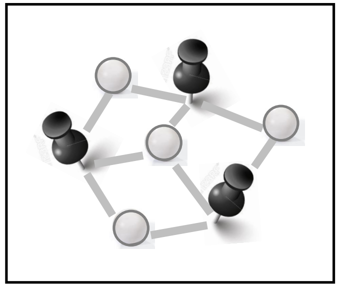
G_X



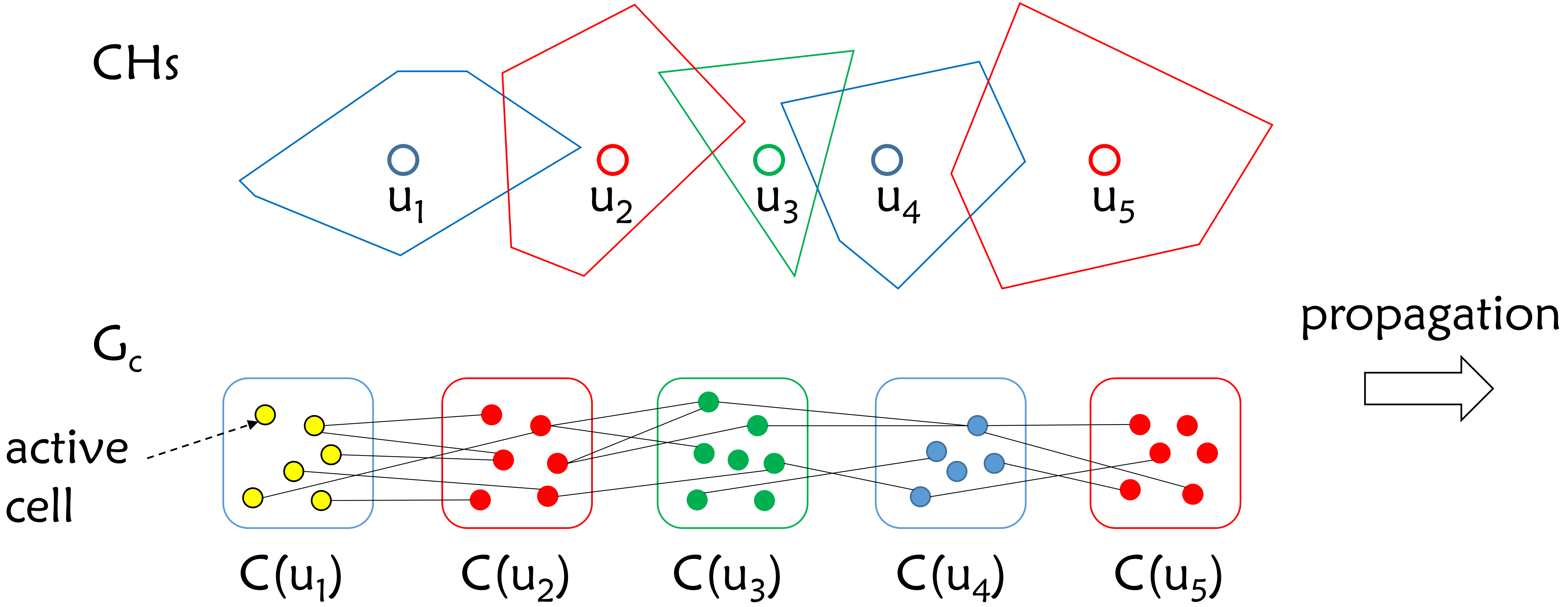


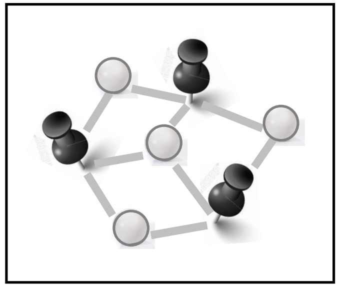
Result 2 – When G_X is a path



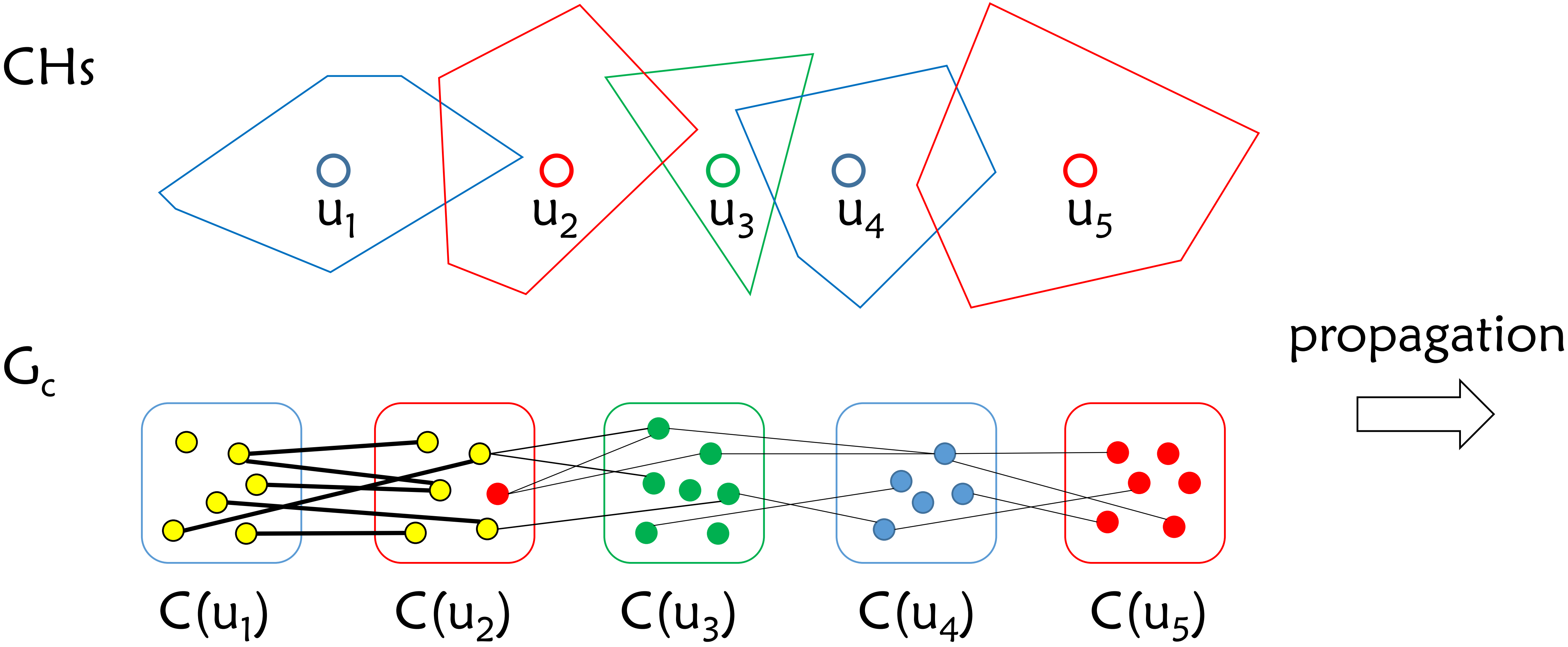


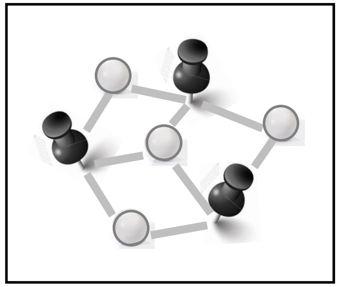
Result 2 – When G_X is a path



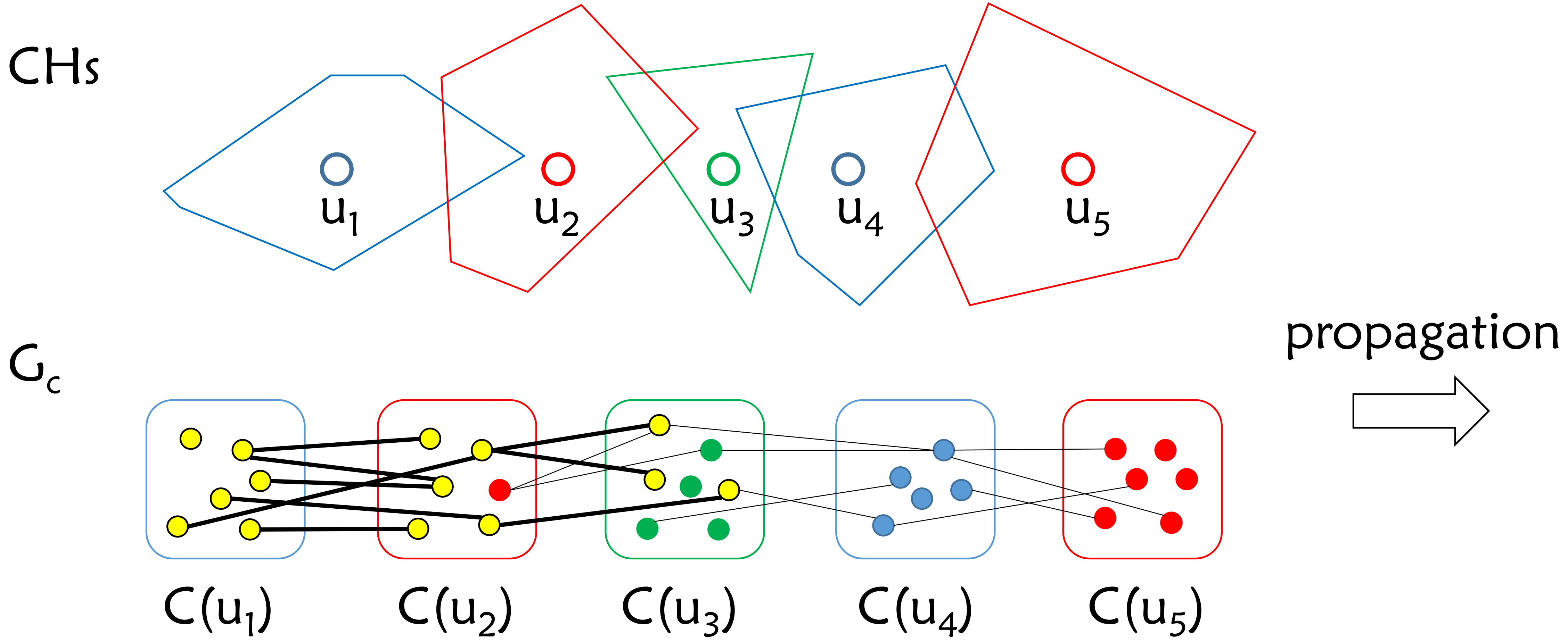


Result 2 – When G_X is a path

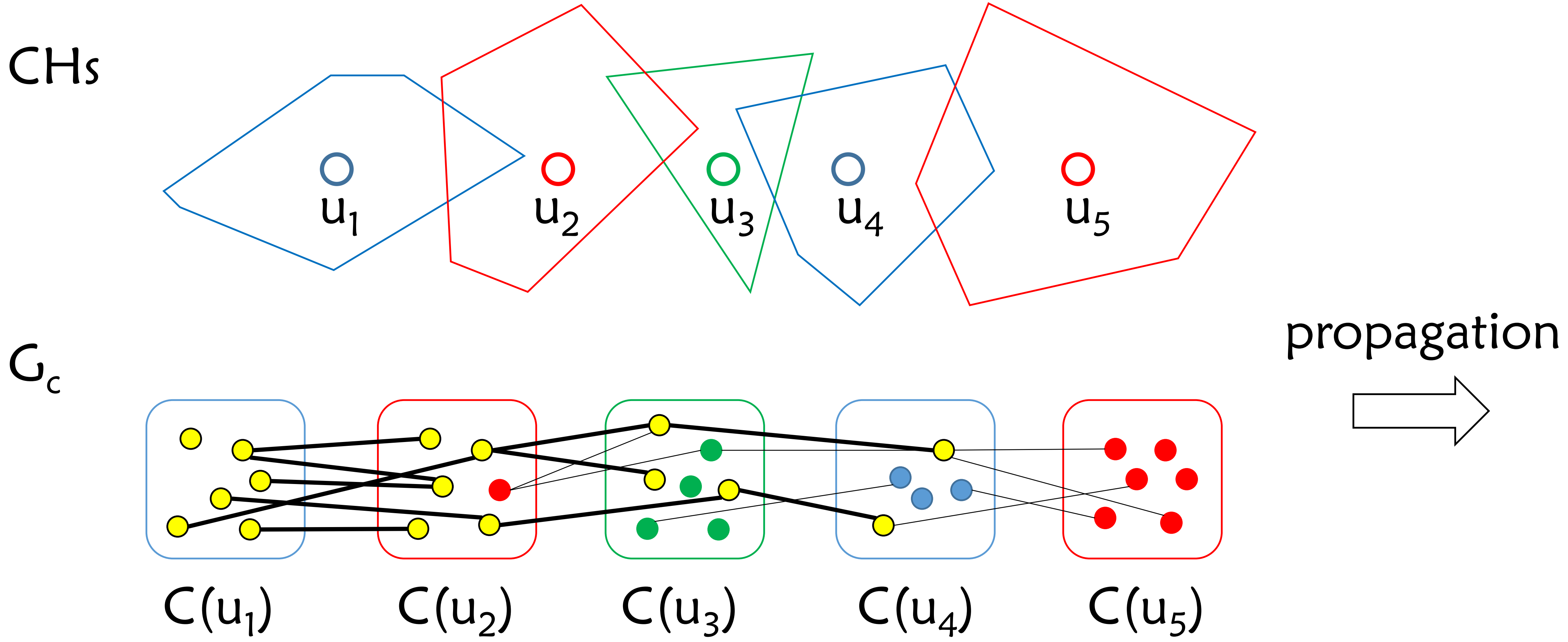


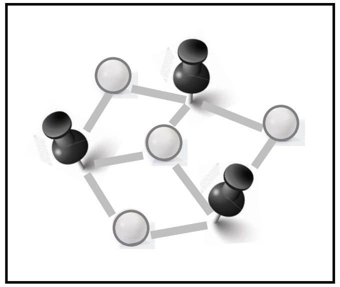


Result 2 – When G_X is a path

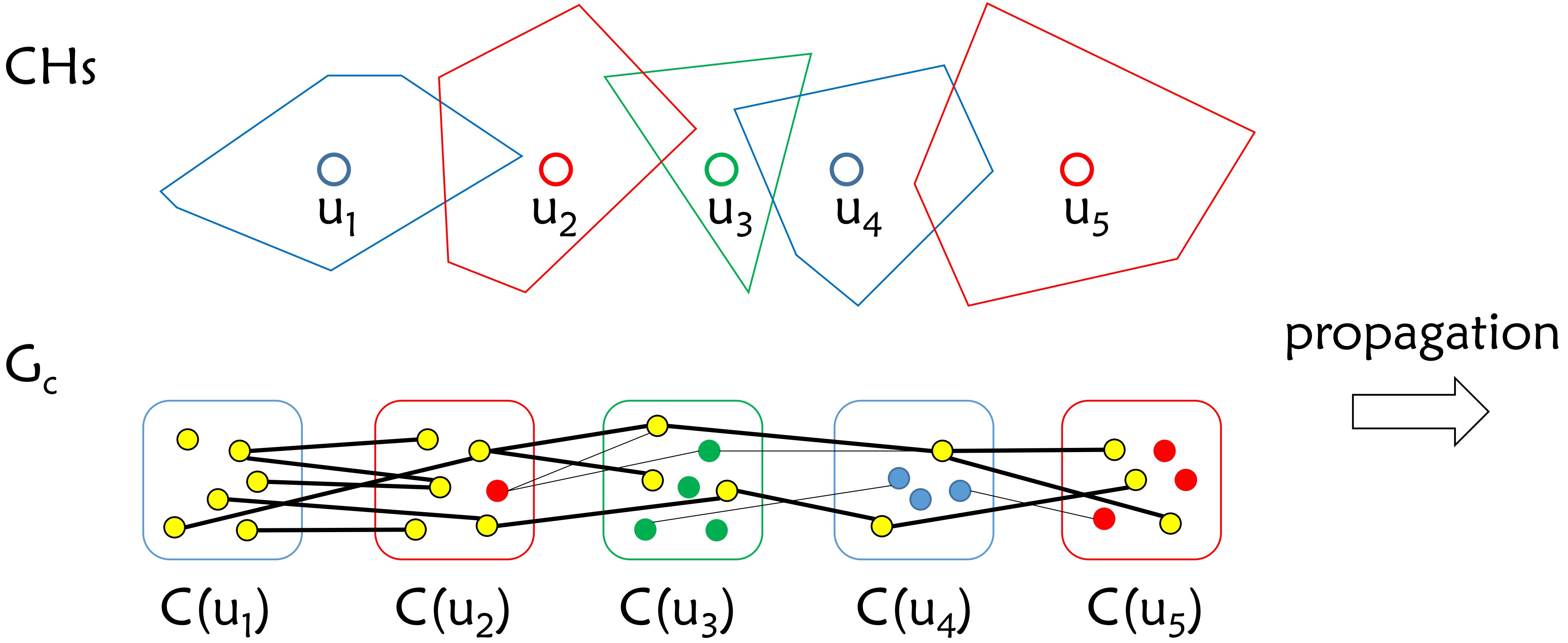


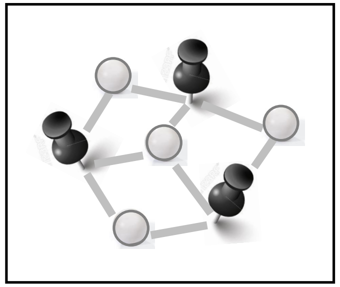
Result 2 – When G_X is a path



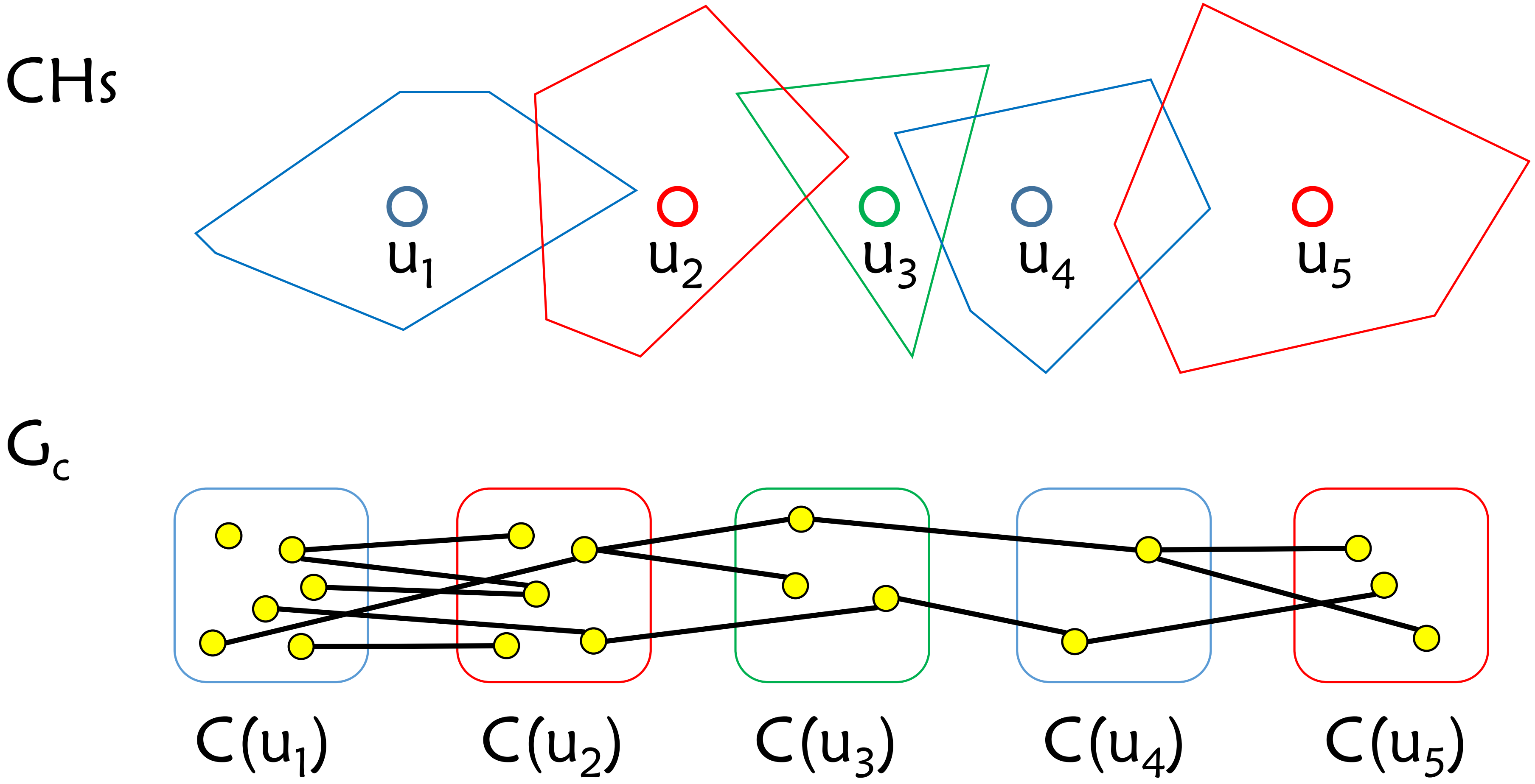


Result 2 – When G_X is a path

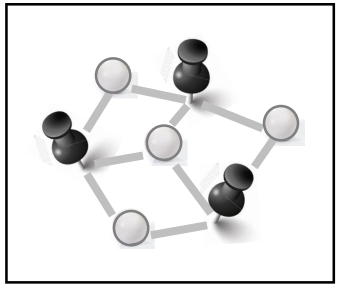




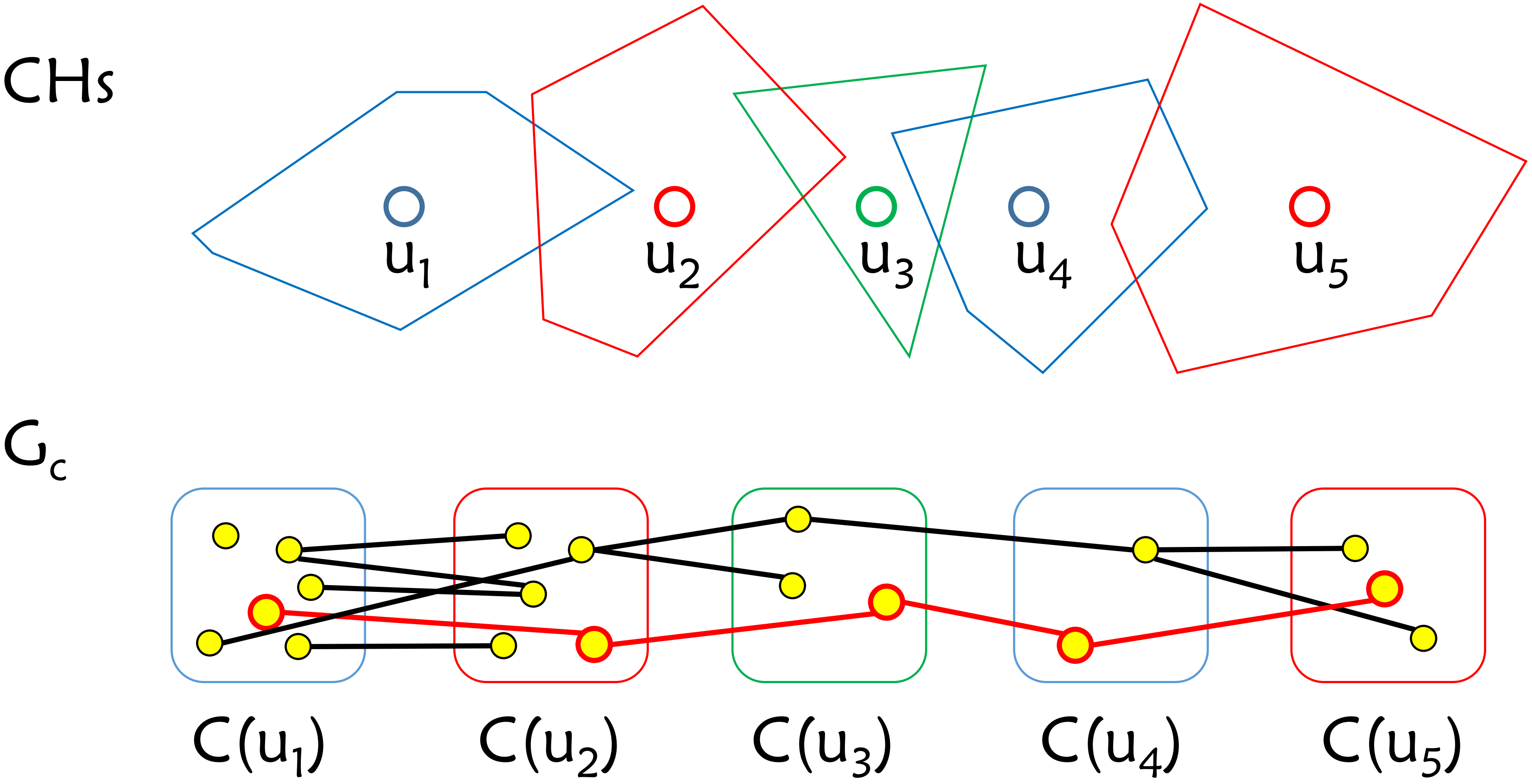
Result 2 – When G_X is a path



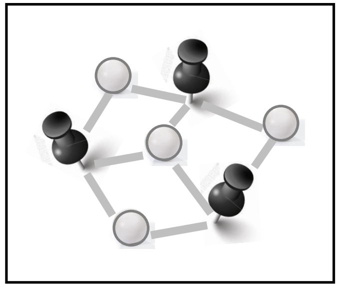
remove not visited nodes



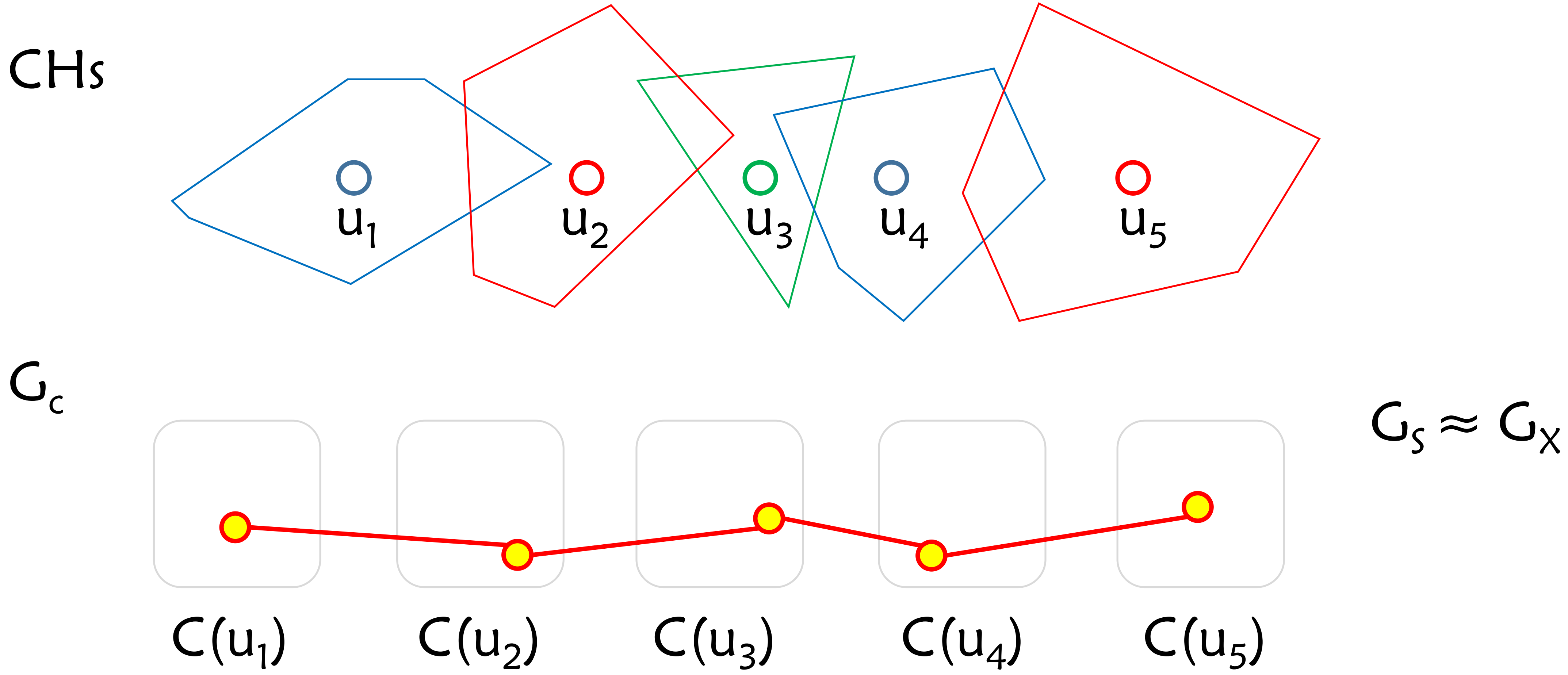
Result 2 – When G_X is a path

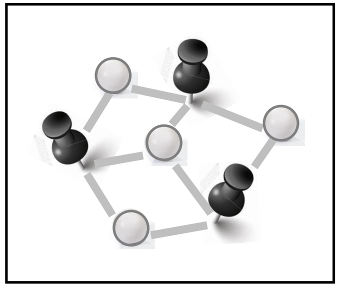


choose a node in the last cluster and reconstruct a path backward



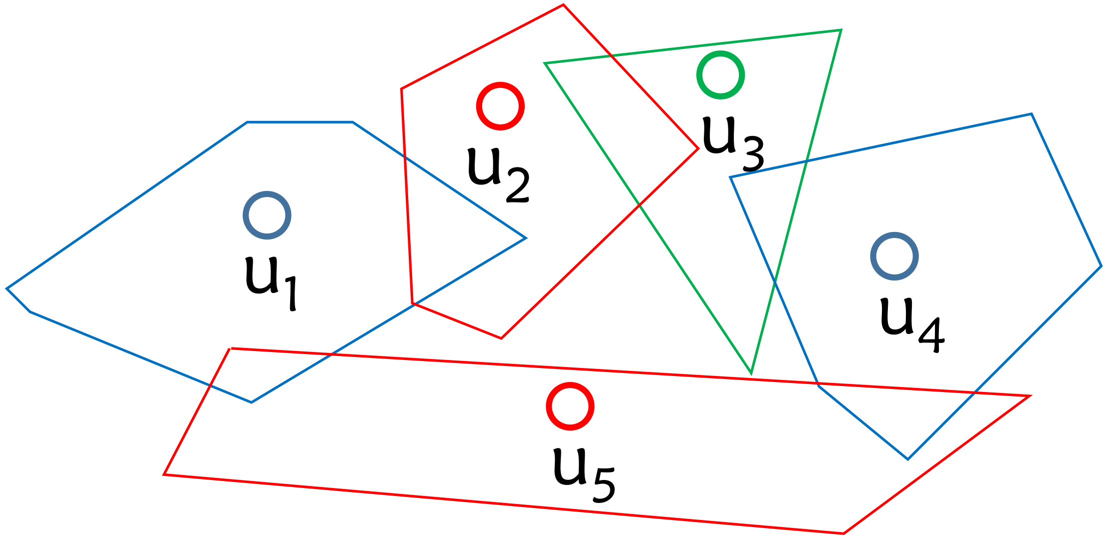
Result 2 – When G_X is a path



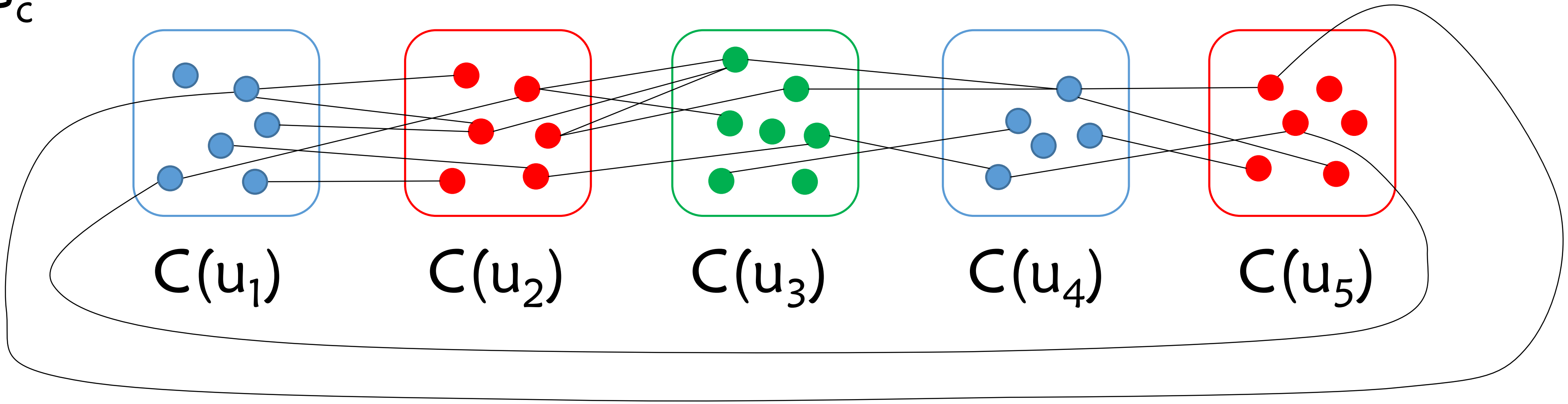


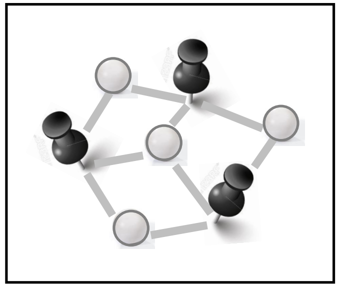
Result 2 – When G_X is a cycle

CHs

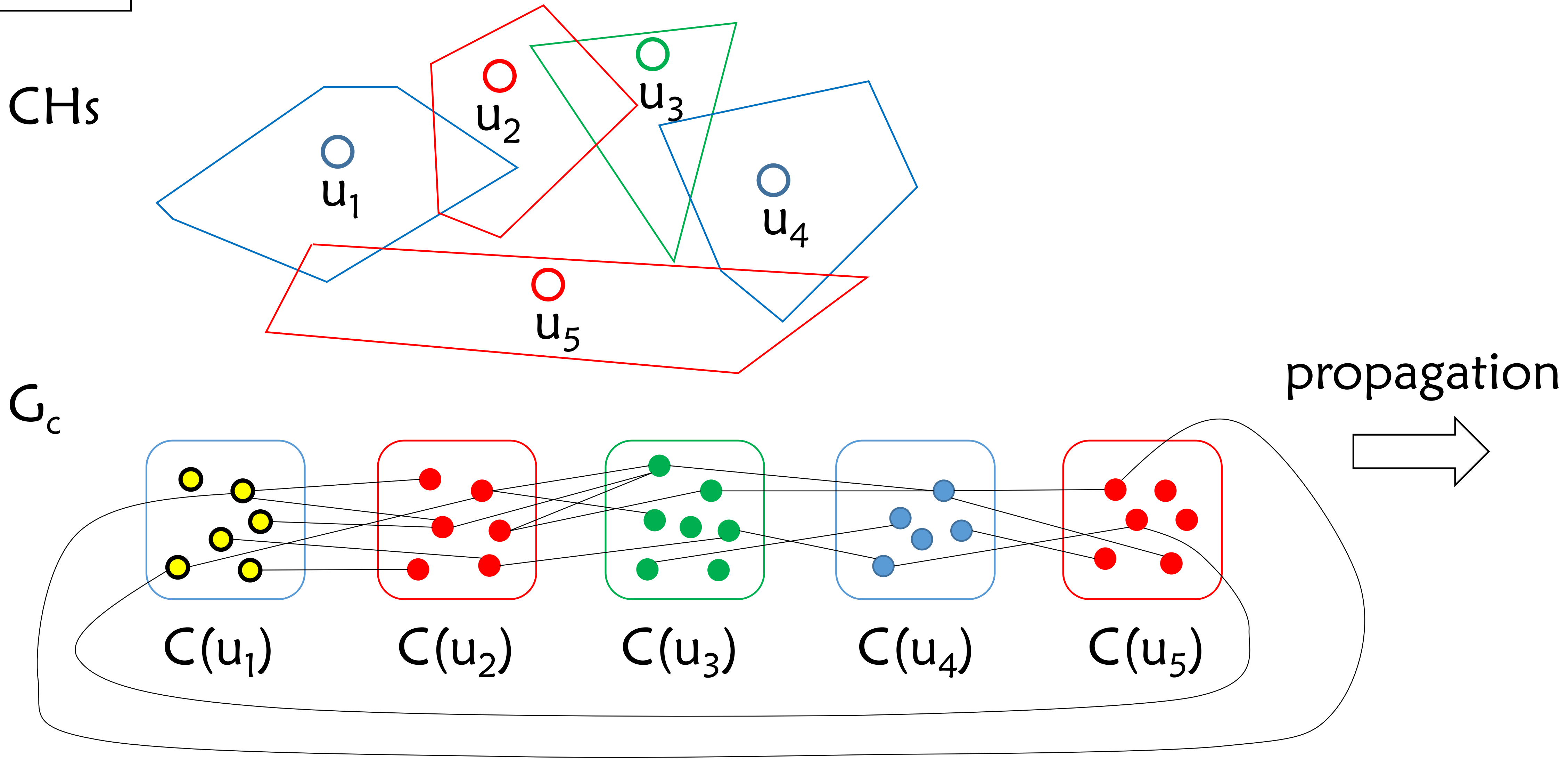


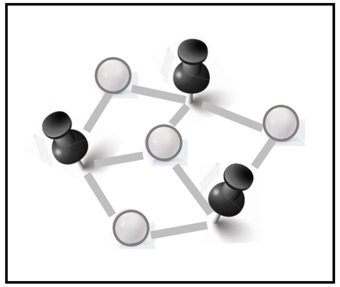
G_c



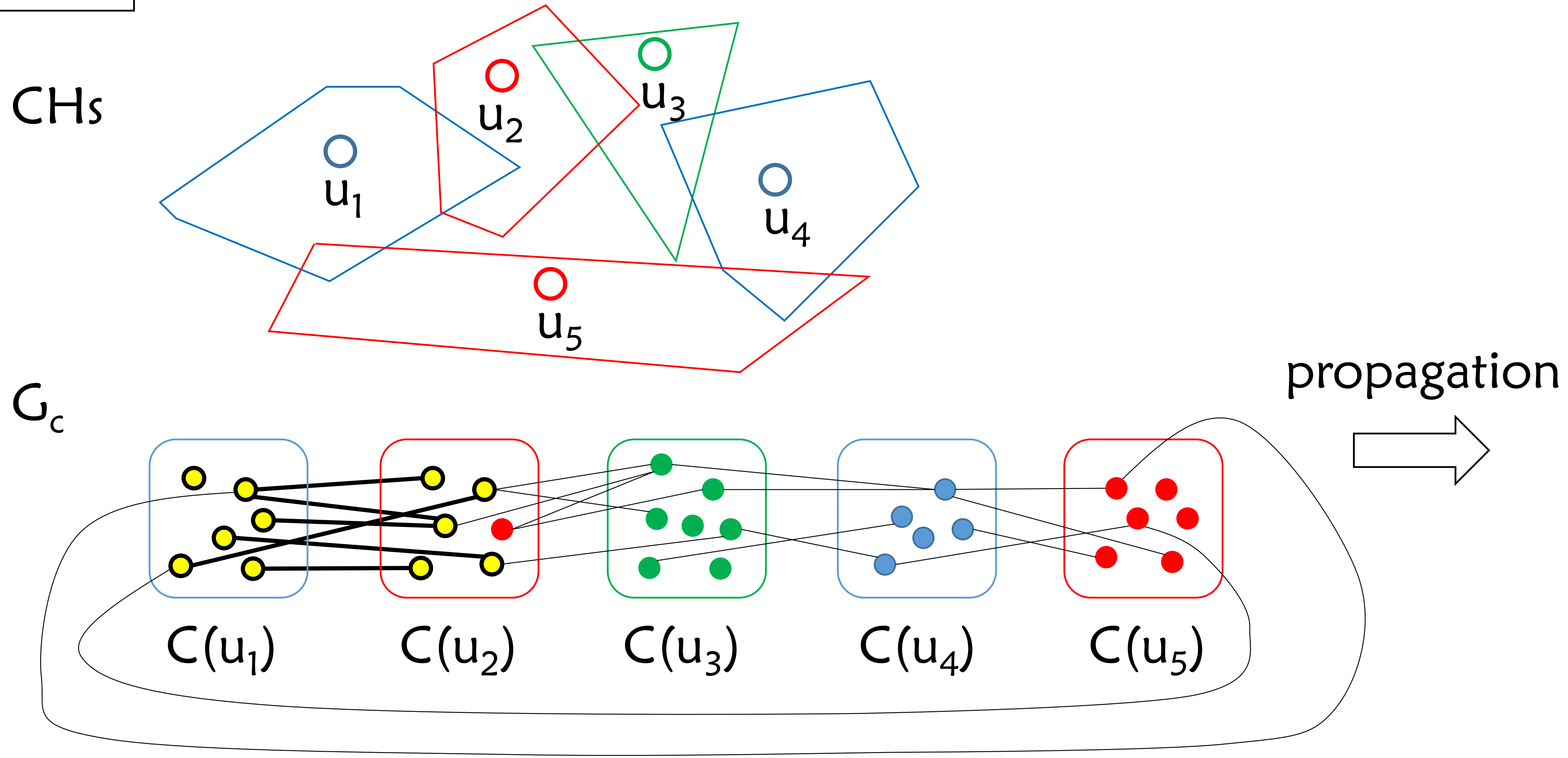


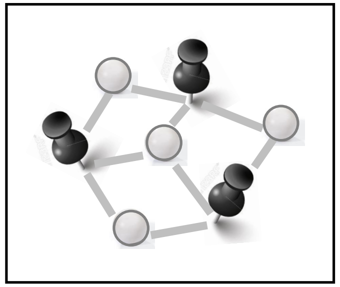
Result 2 – When G_X is a cycle



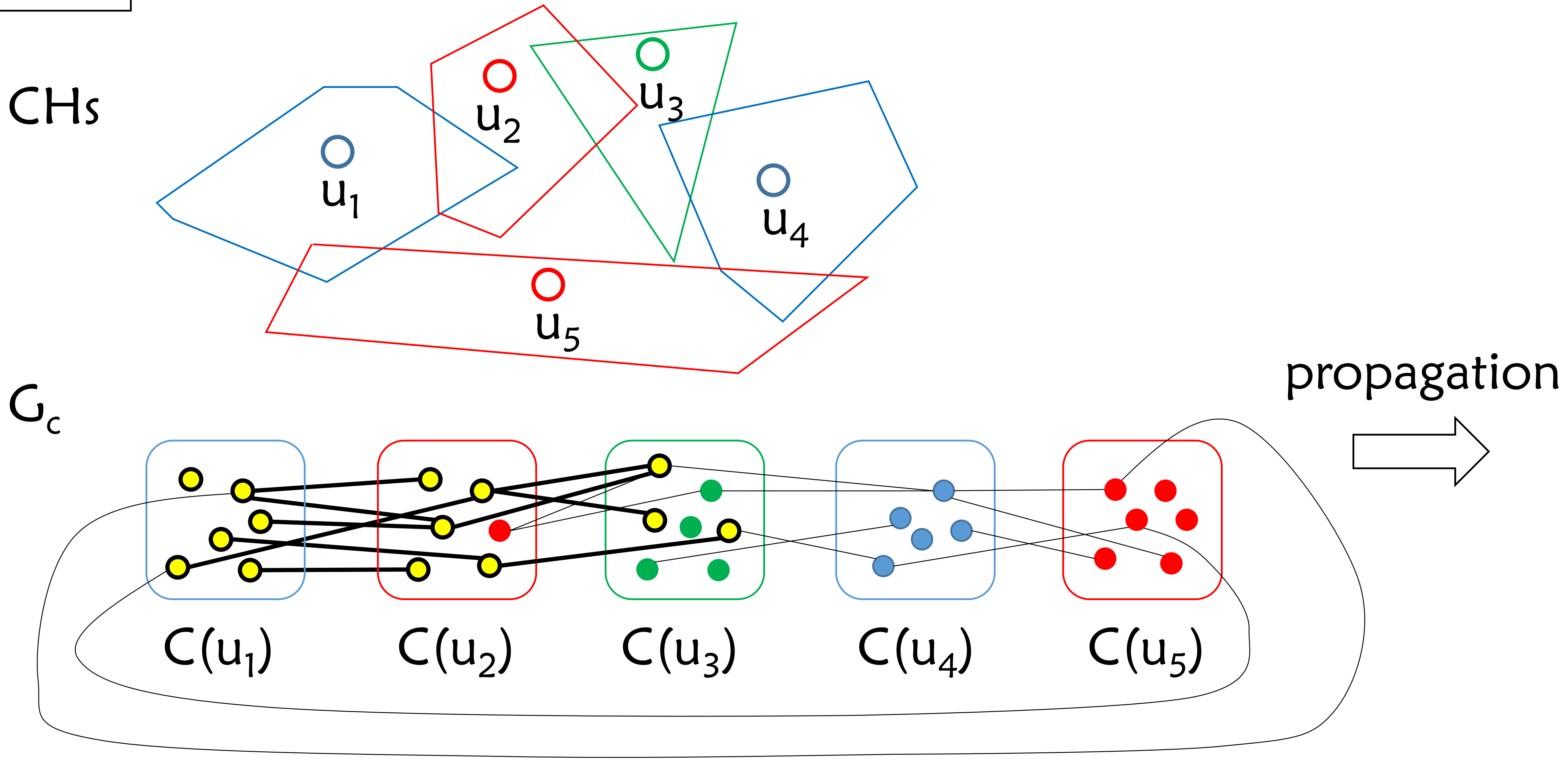


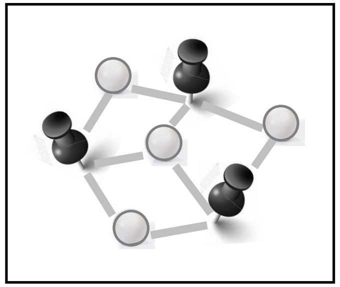
Result 2 – When G_X is a cycle



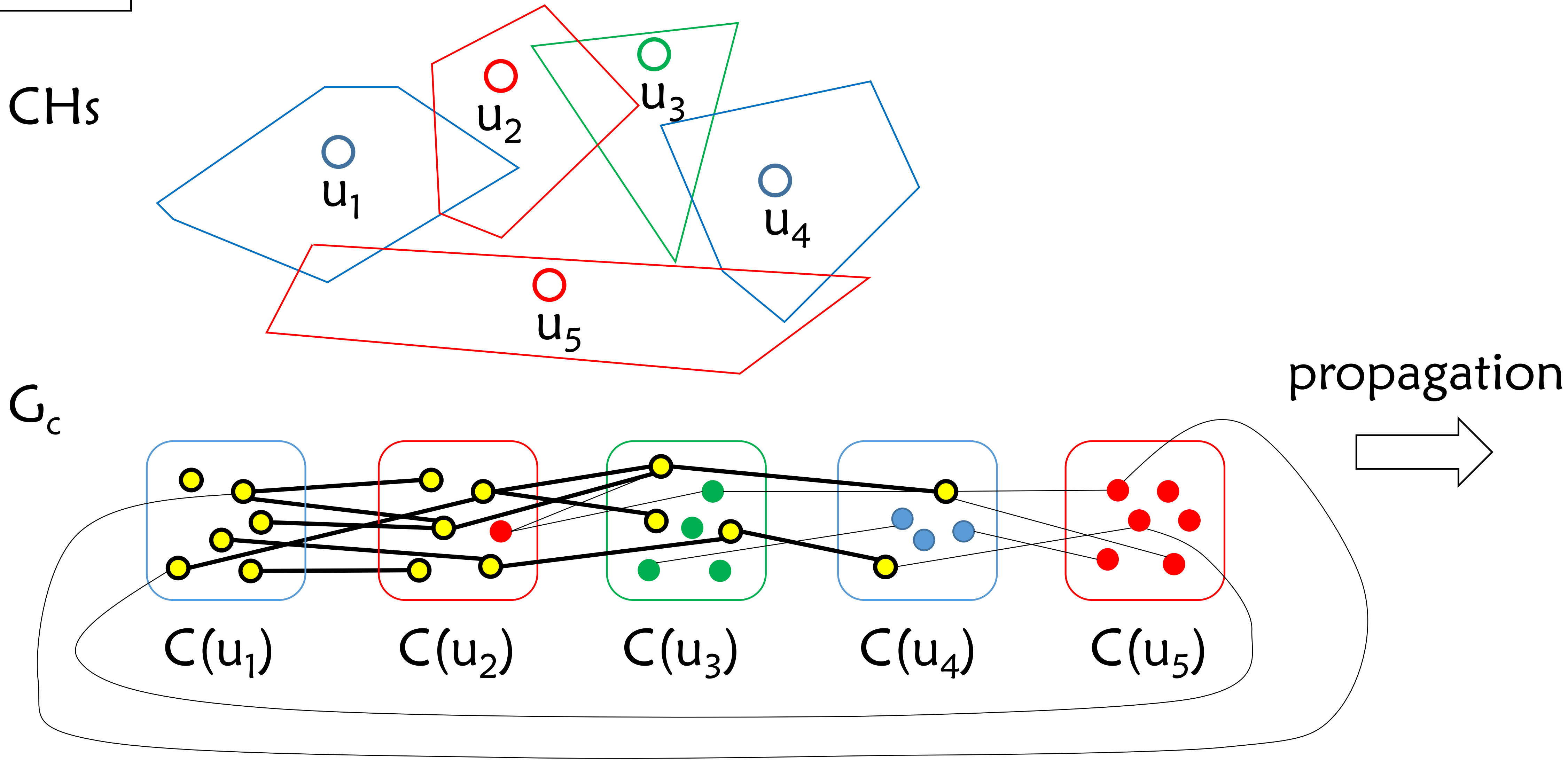


Result 2 – When G_X is a cycle

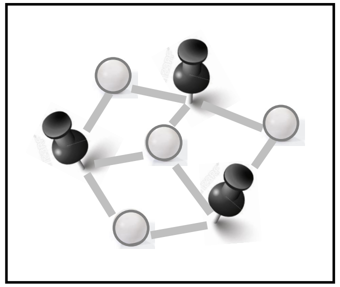


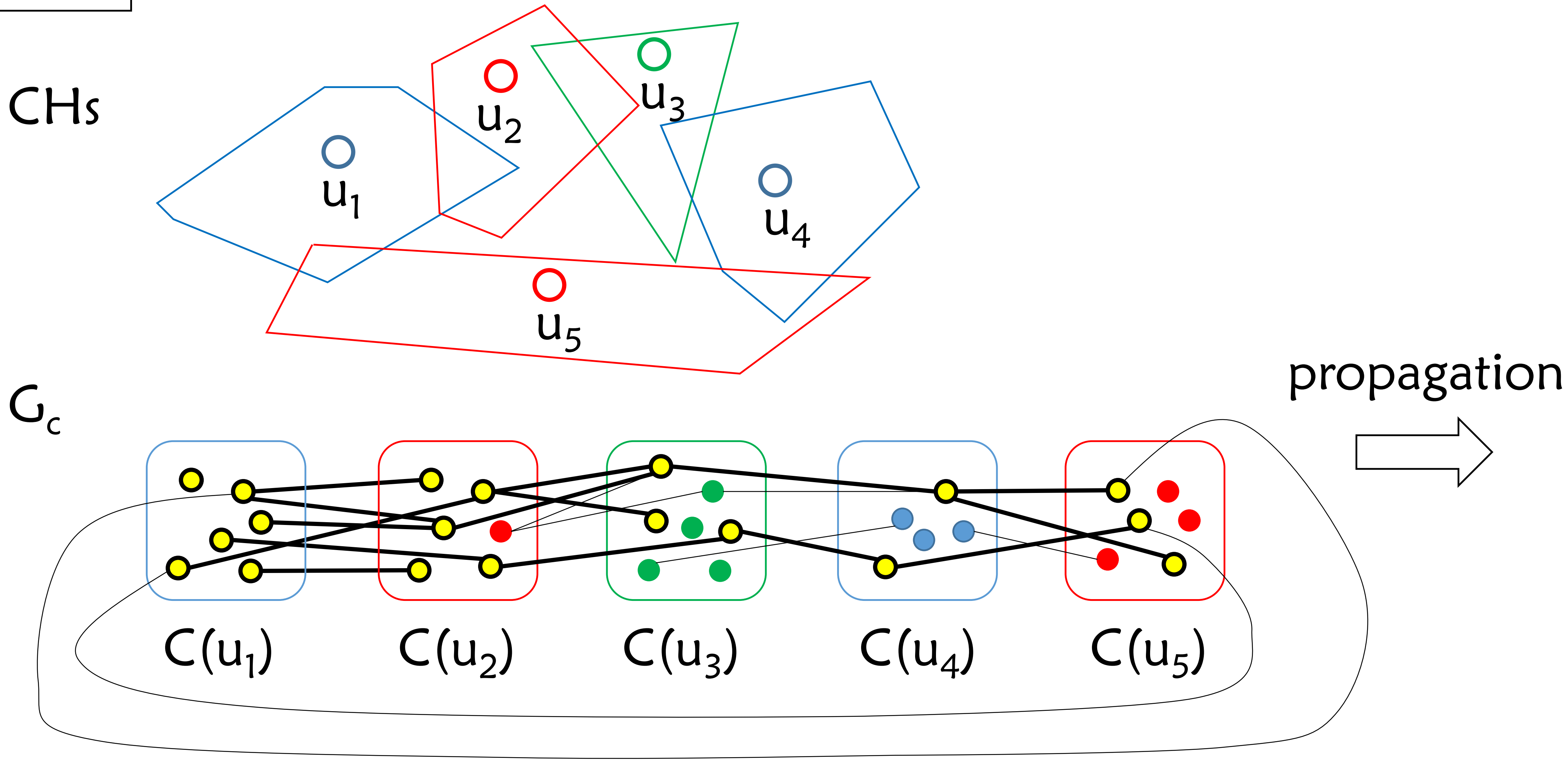


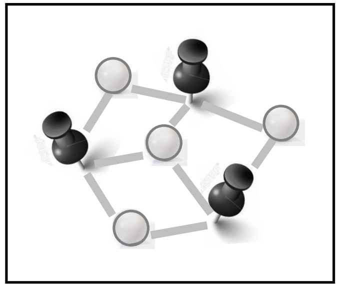
Result 2 – When G_X is a cycle



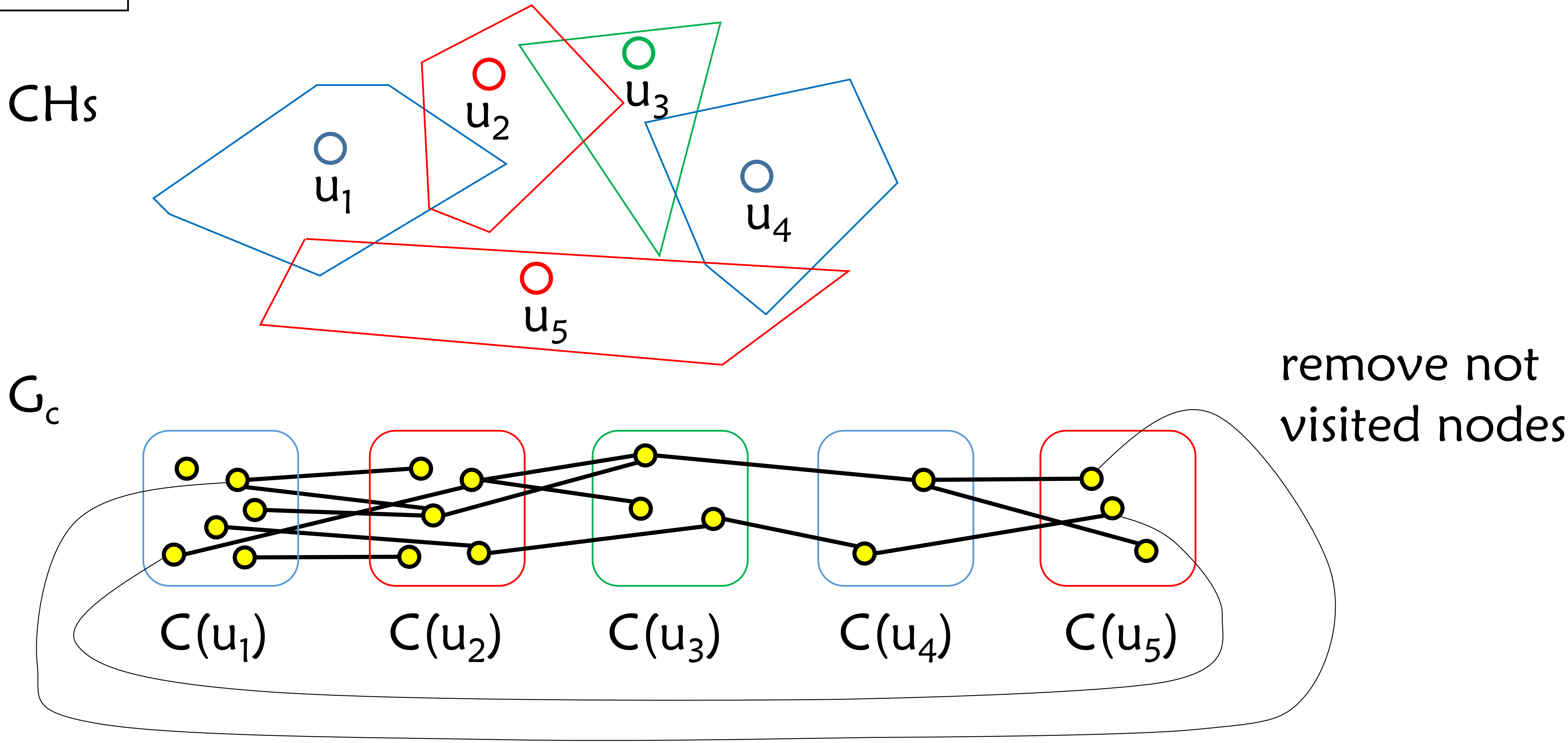
Result 2 – When G_X is a cycle



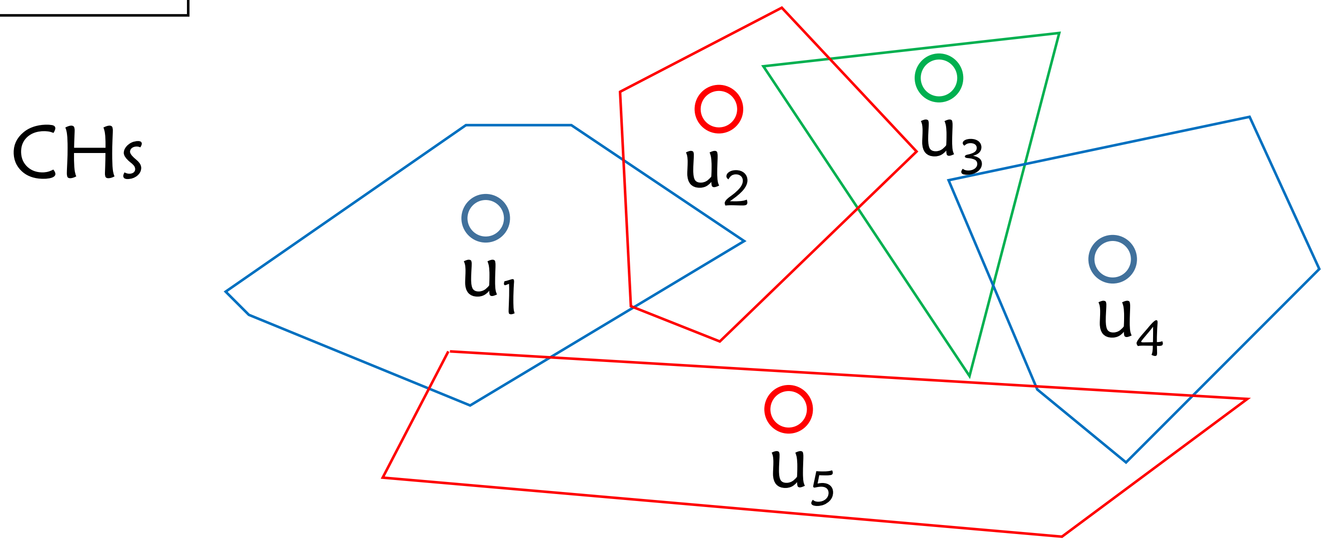


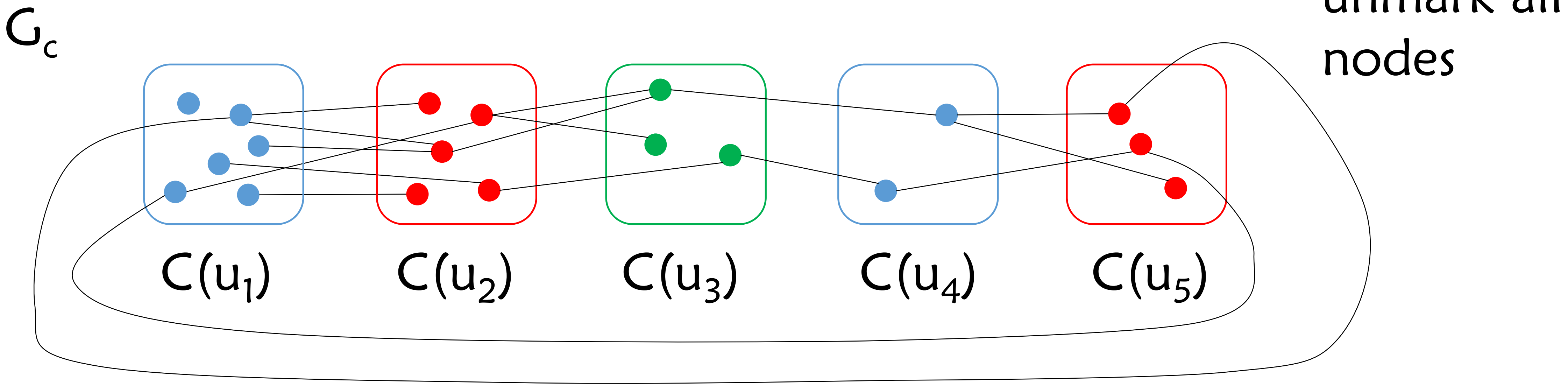


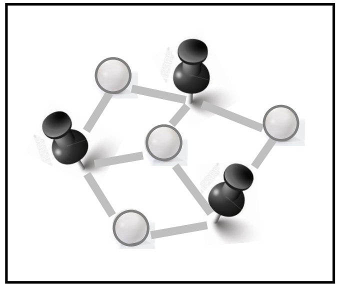
Result 2 – When G_X is a cycle



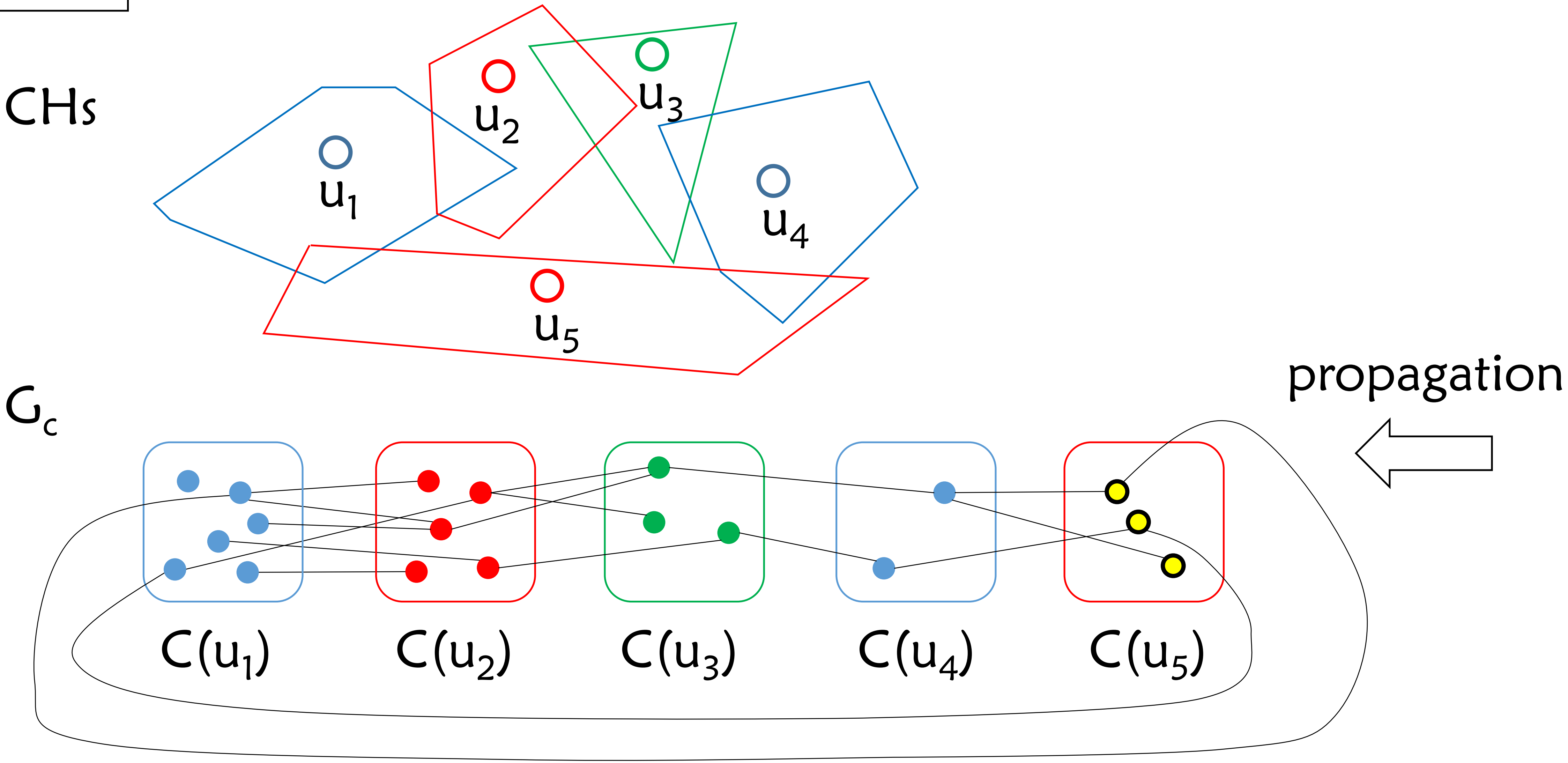
Result 2 – When G_X is a cycle

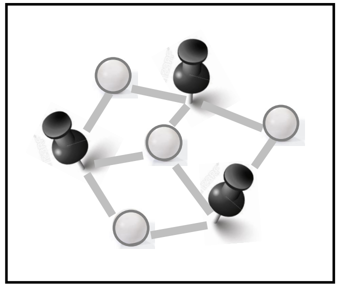




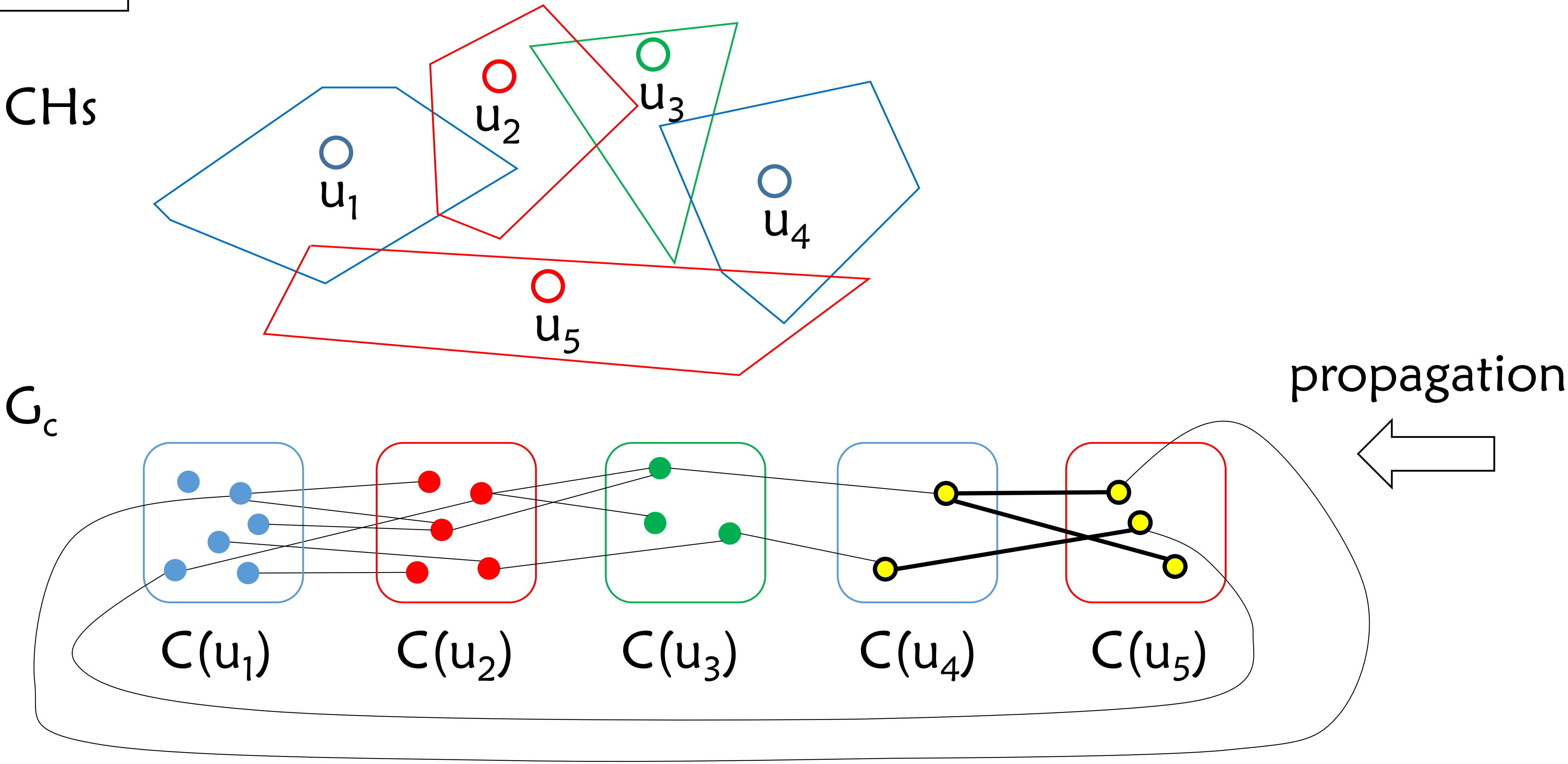


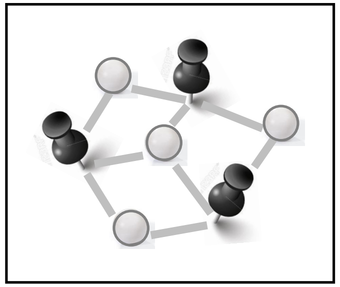
Result 2 – When G_X is a cycle



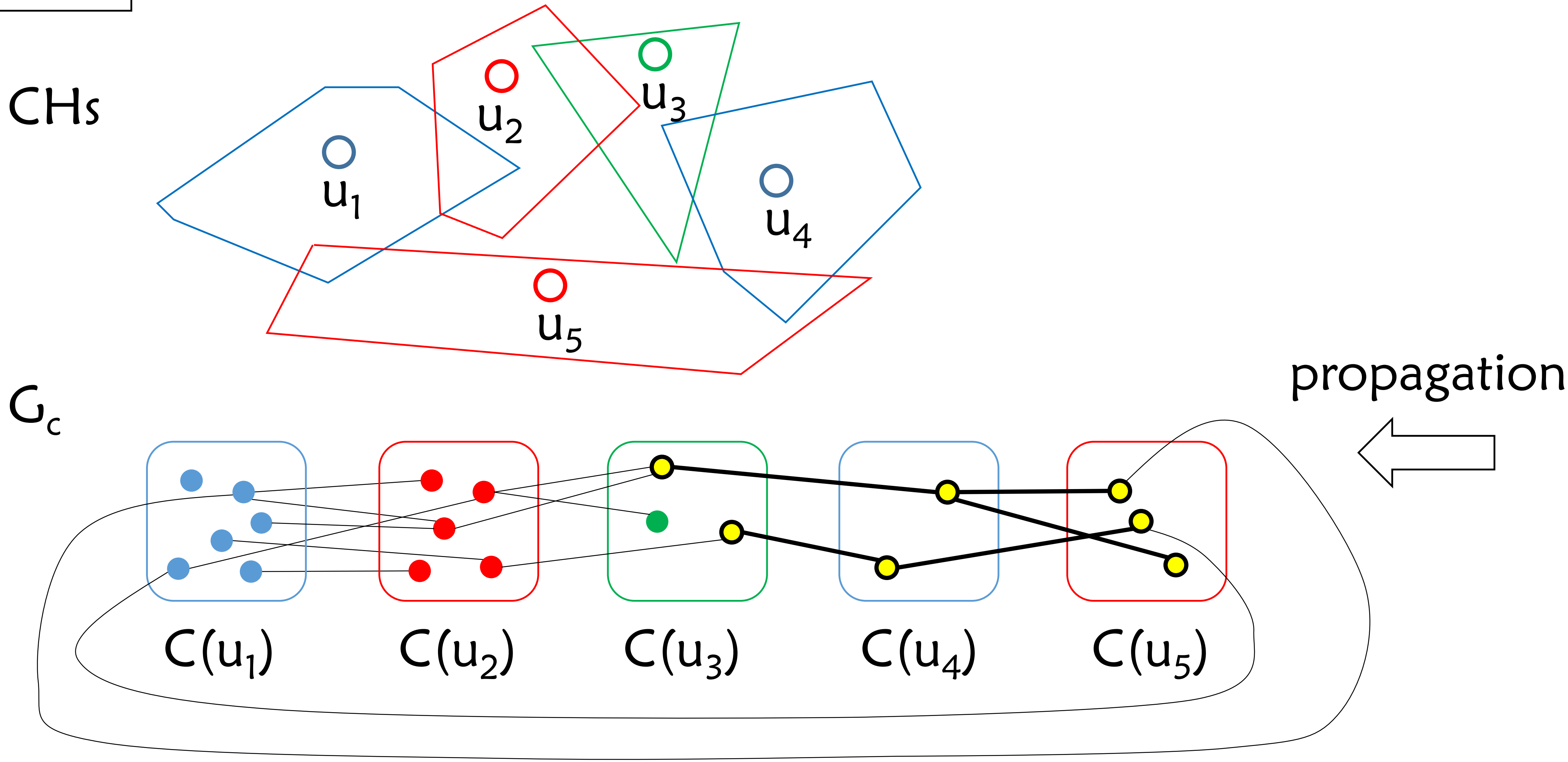


Result 2 – When G_X is a cycle

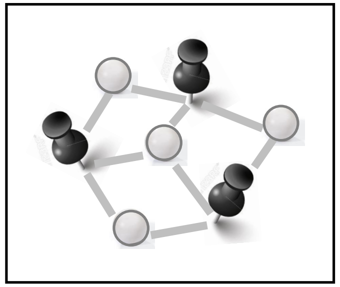


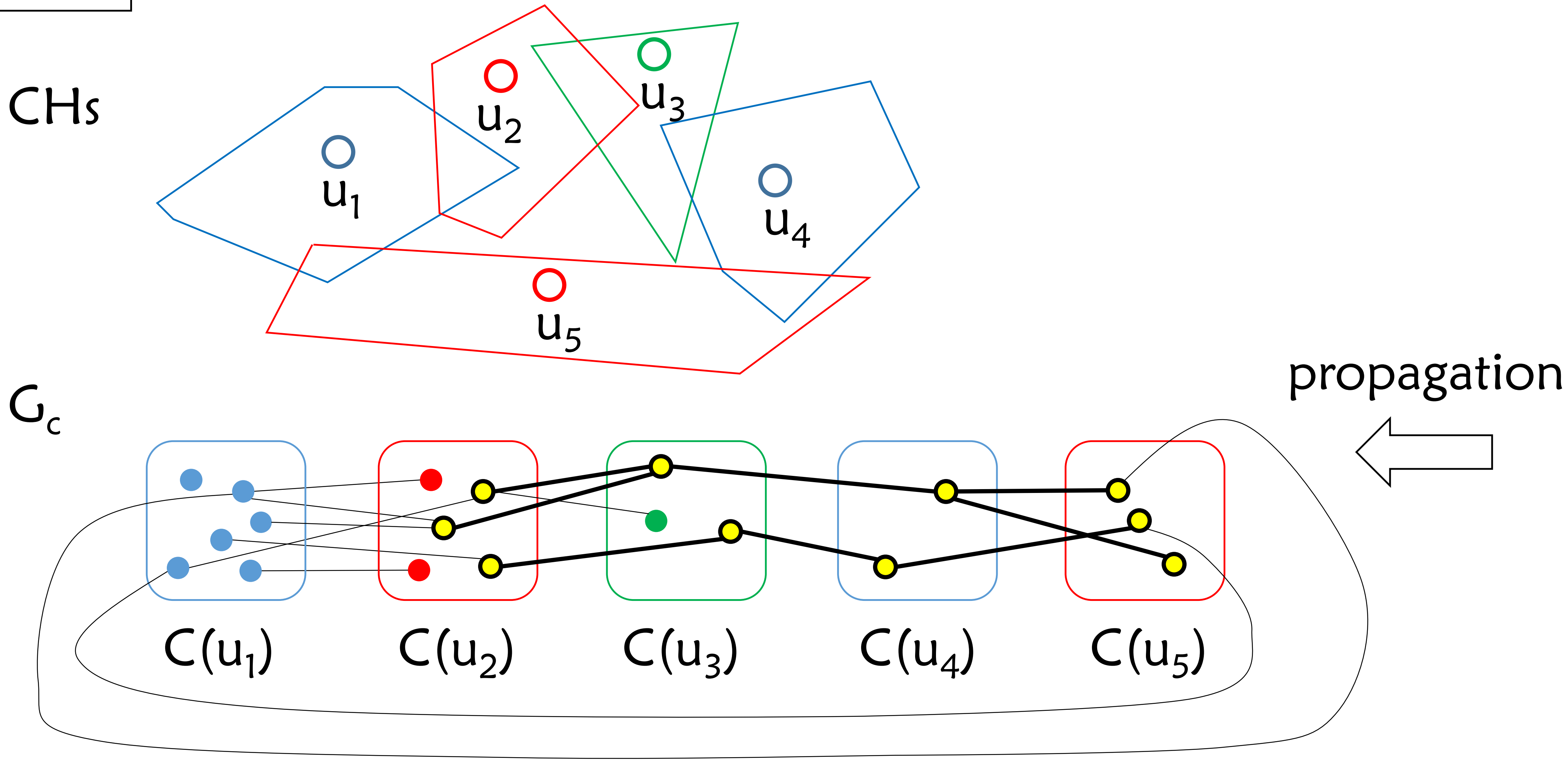


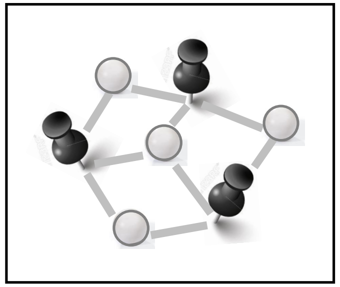
Result 2 – When G_X is a cycle



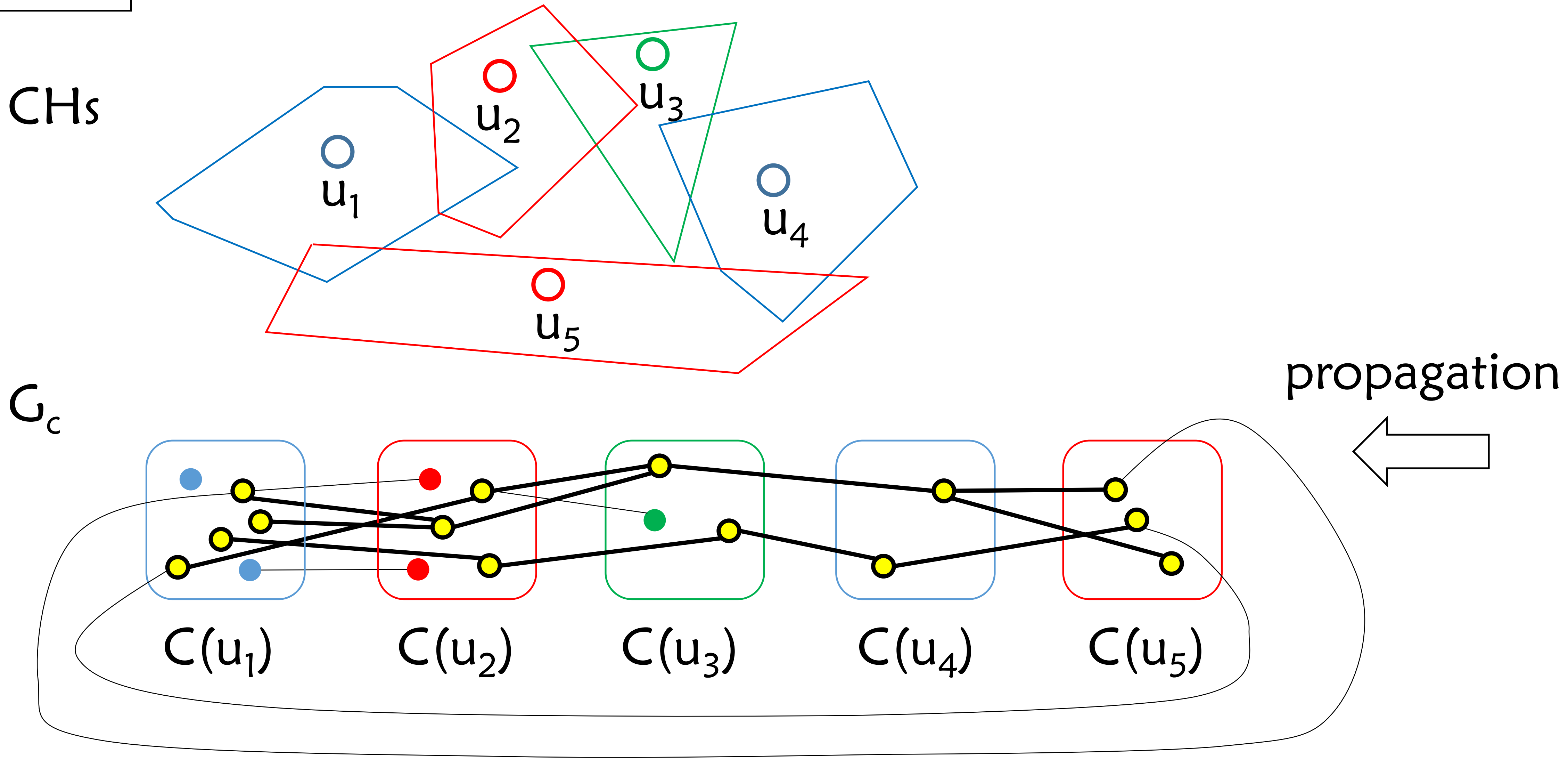
Result 2 – When G_X is a cycle





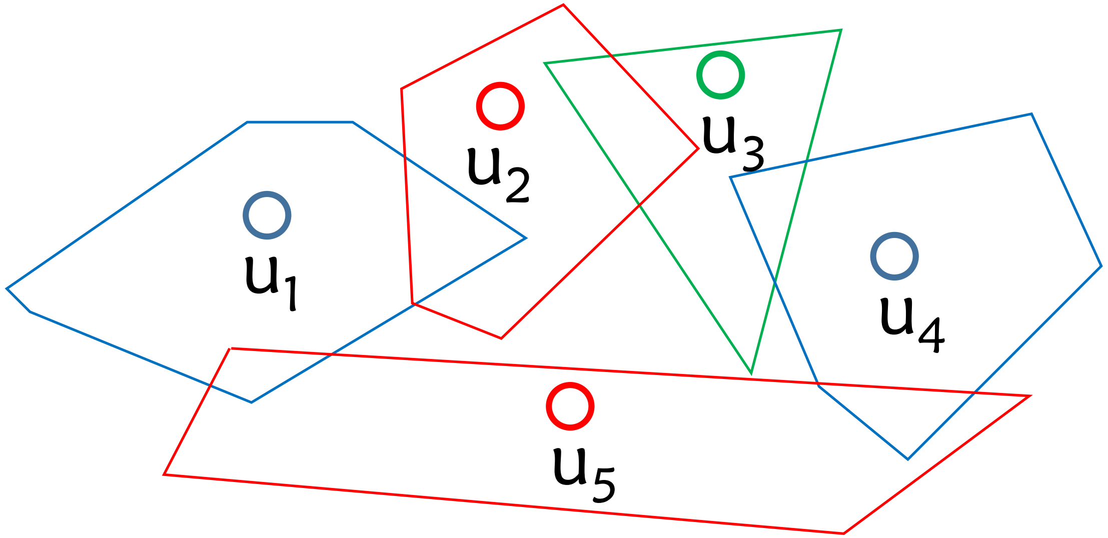


Result 2 – When G_X is a cycle

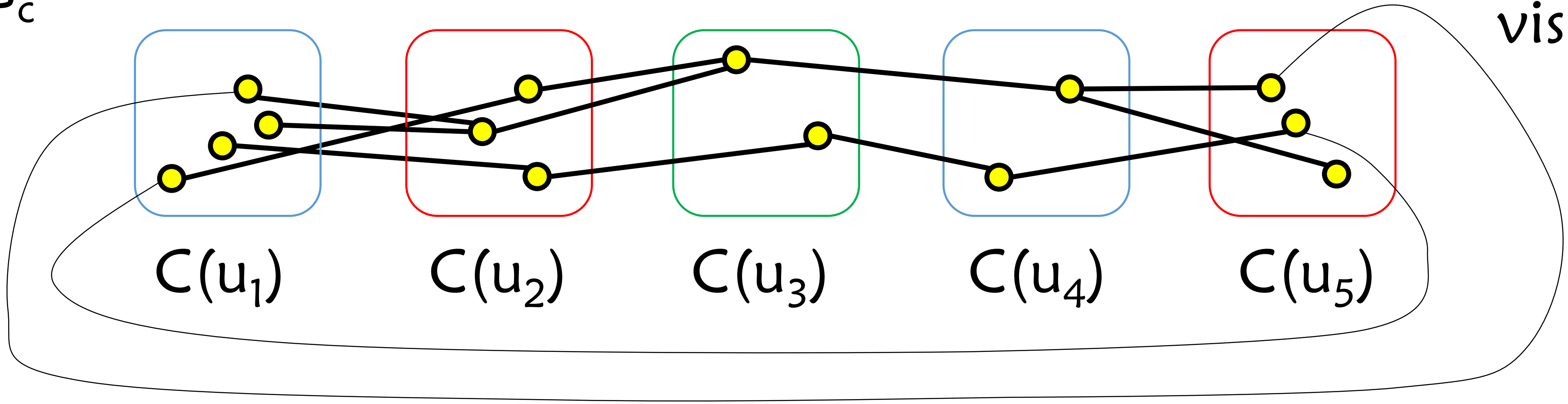


Result 2 – When G_X is a cycle

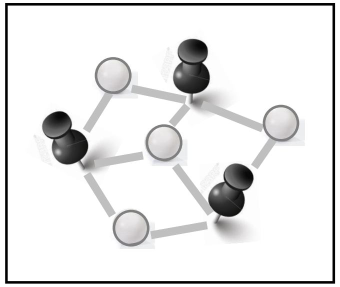
CHs



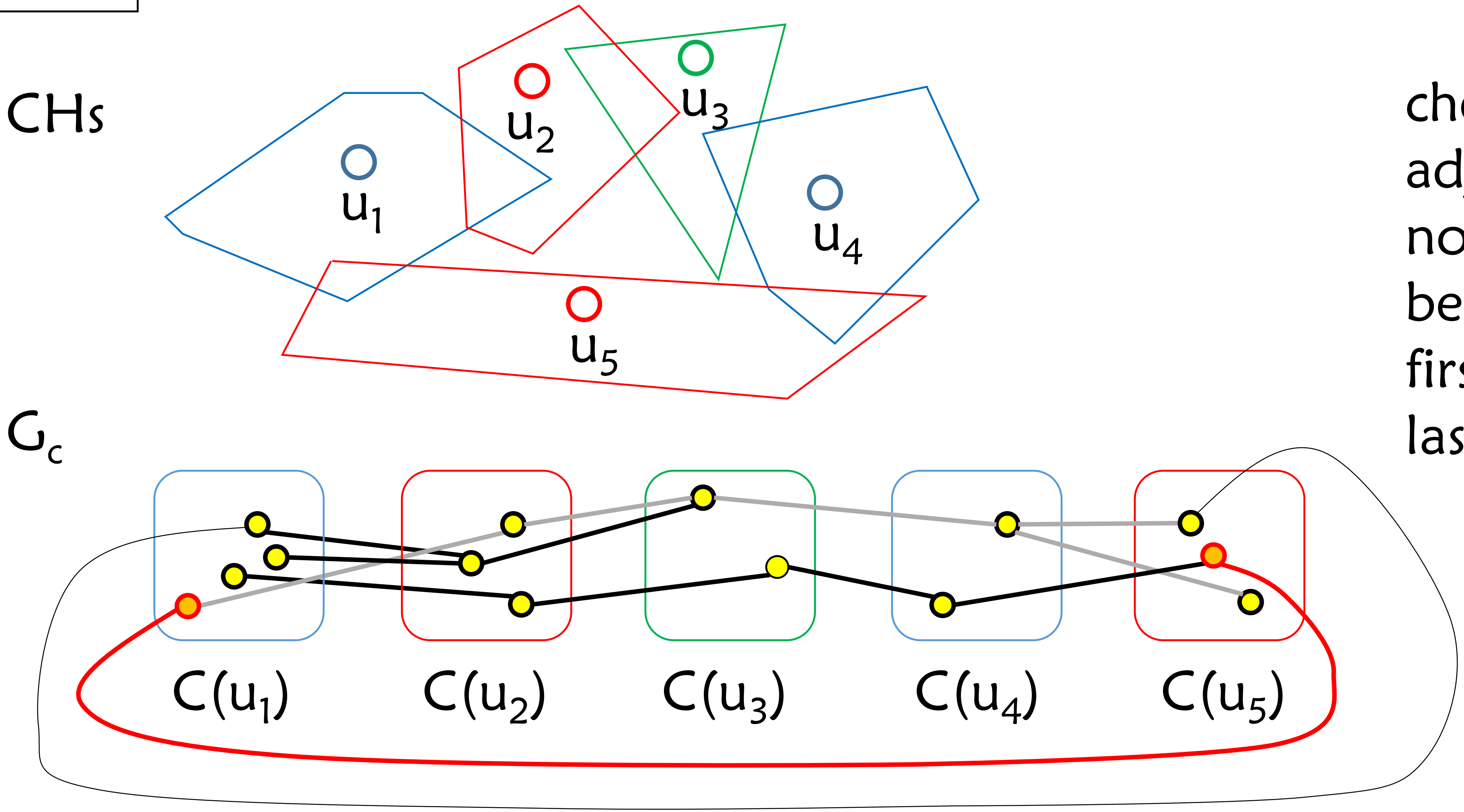
G_c



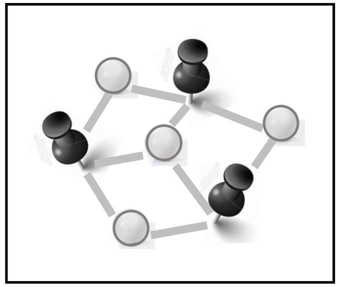
remove not visited nodes



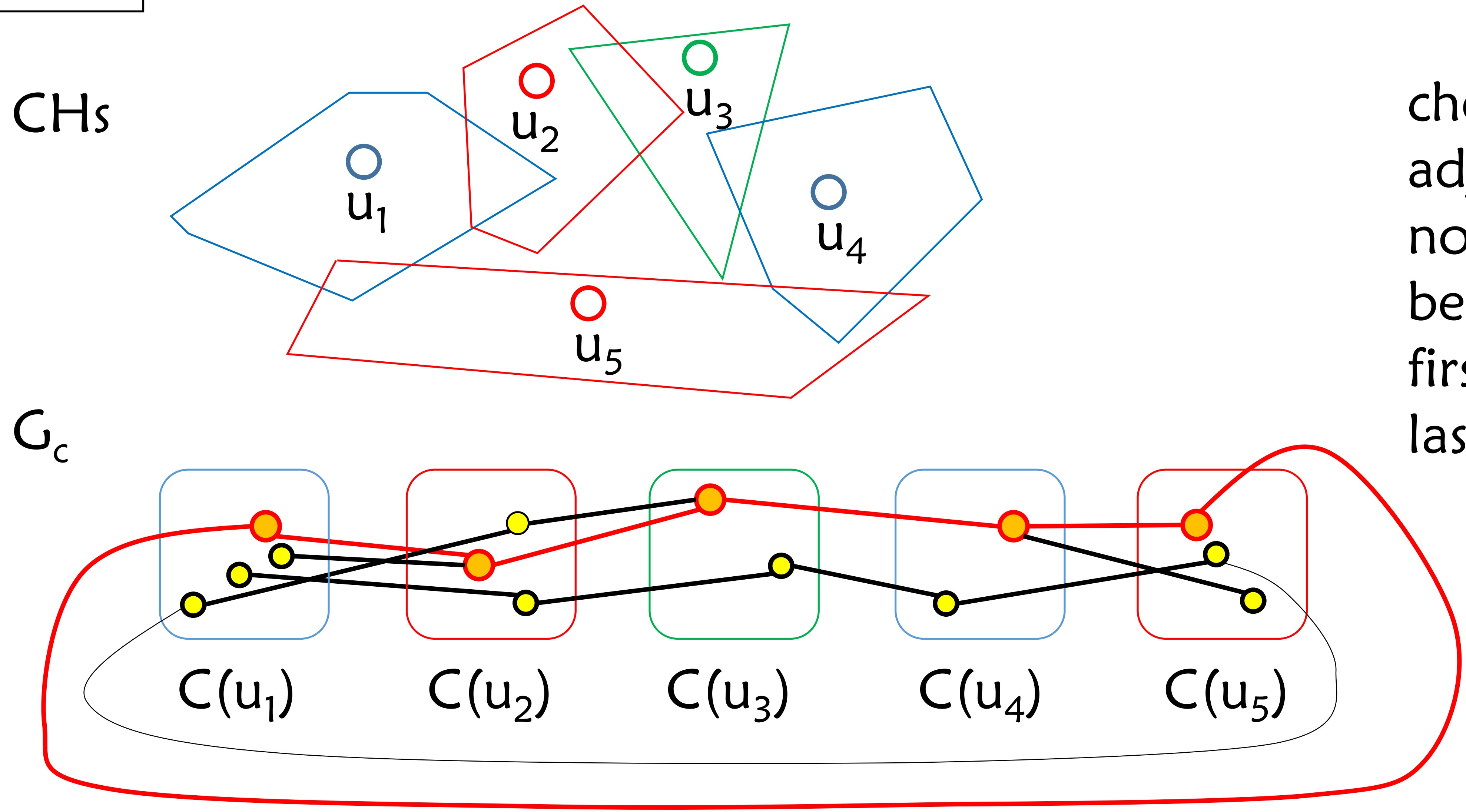
Result 2 – When G_X is a cycle



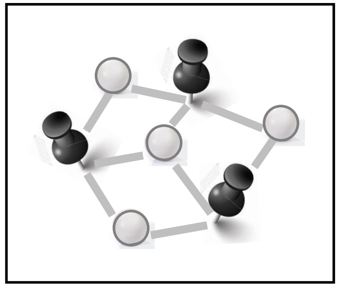
check pairs of adjacent nodes between the first and the last cluster



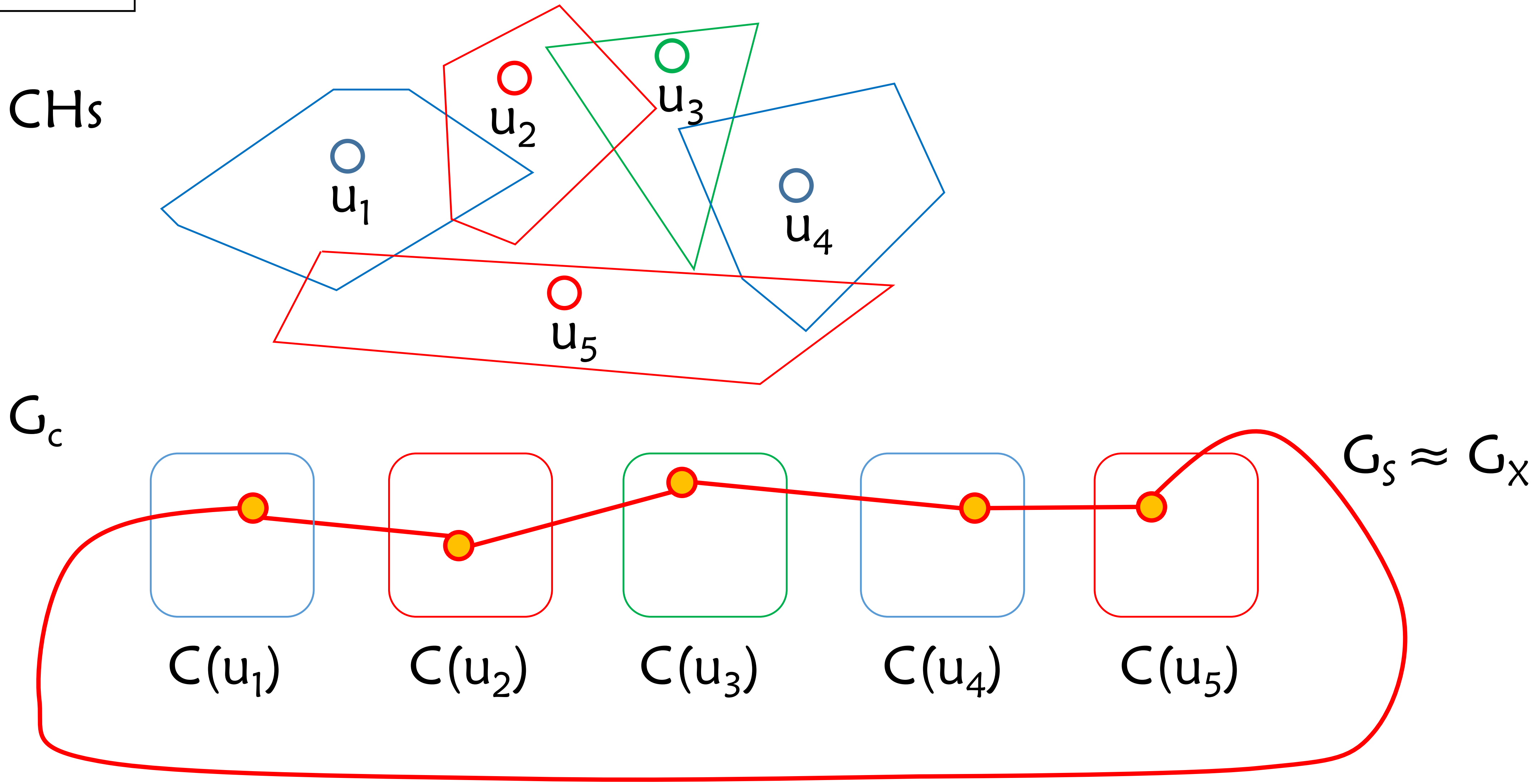
Result 2 – When G_X is a cycle

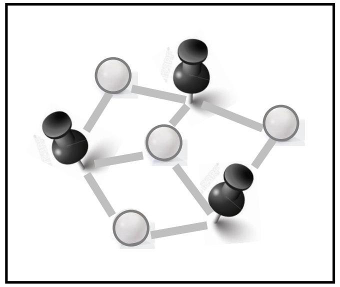


check pairs of adjacent nodes between the first and the last cluster

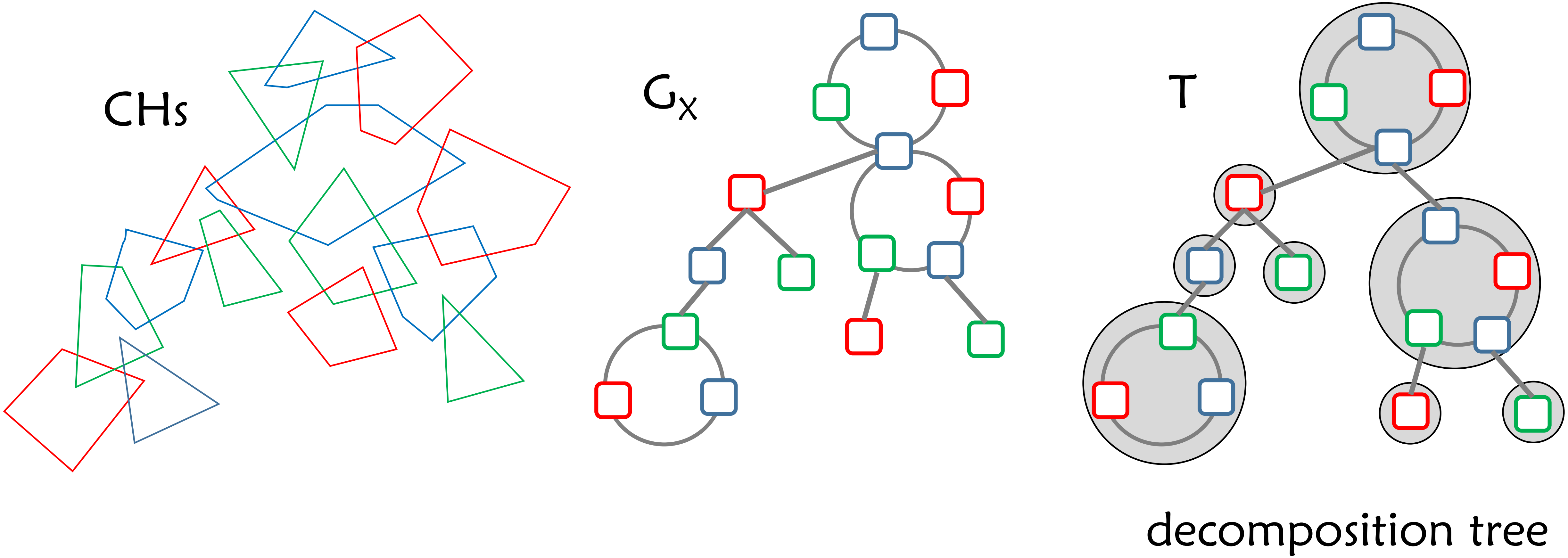


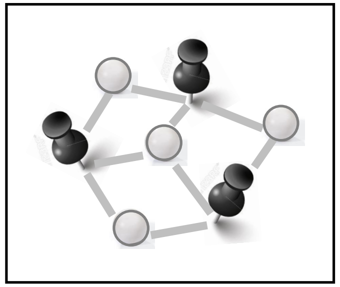
Result 2 – When G_X is a cycle



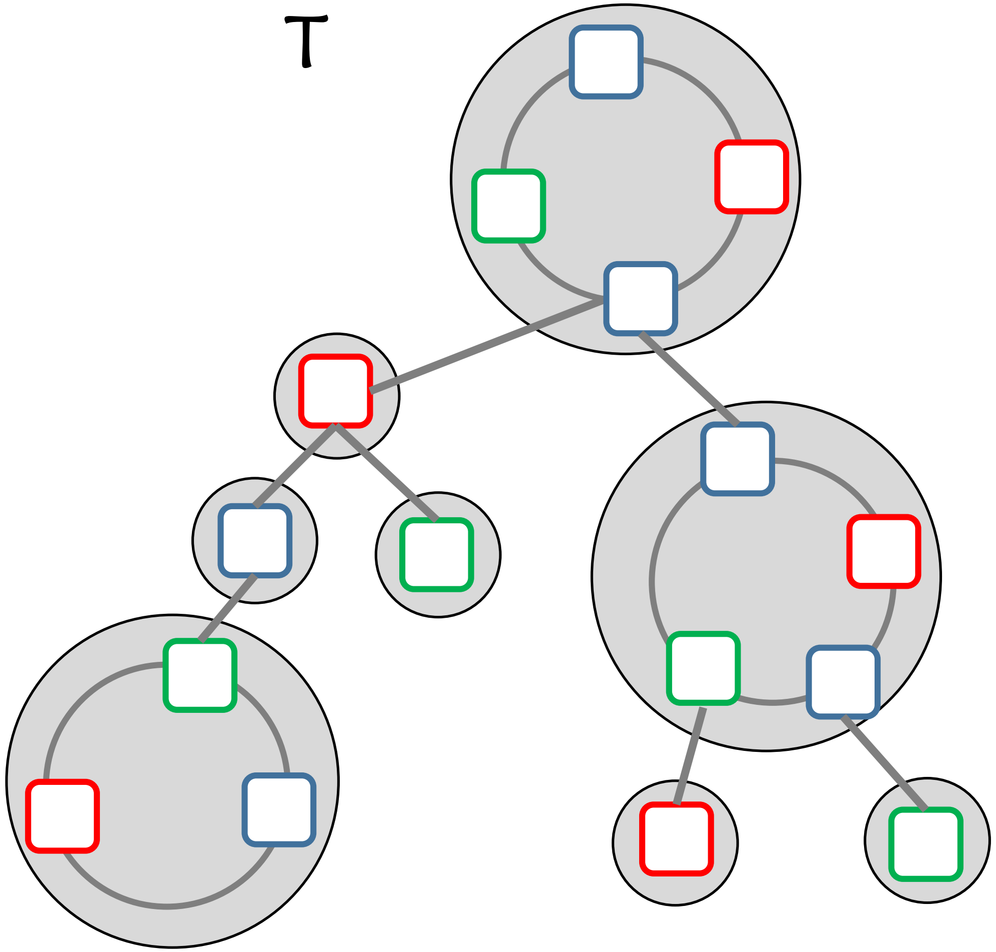


Result 2 – When G_X is a cactus

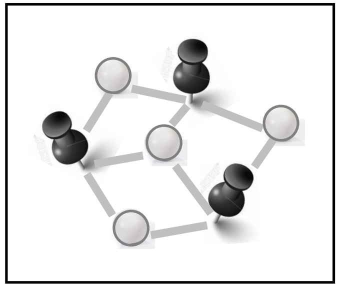




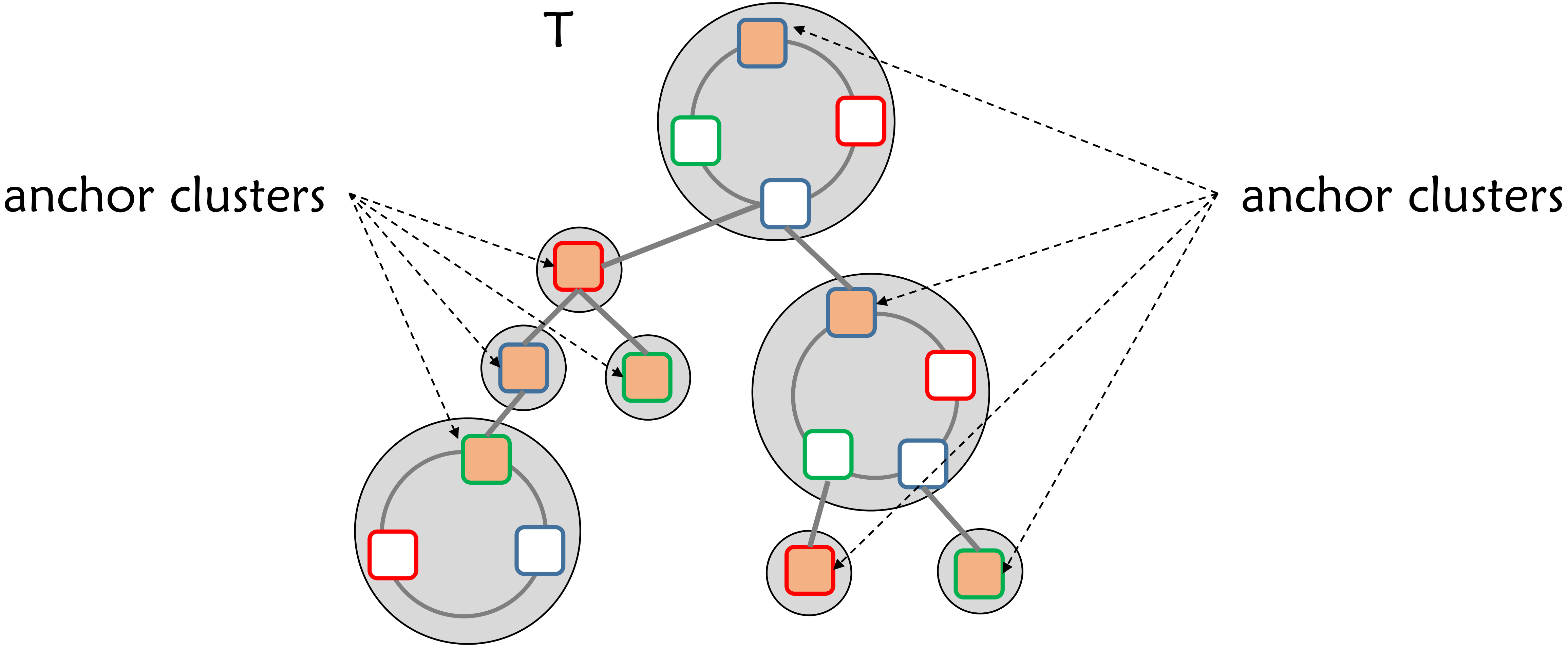
Result 2 – When G_X is a cactus

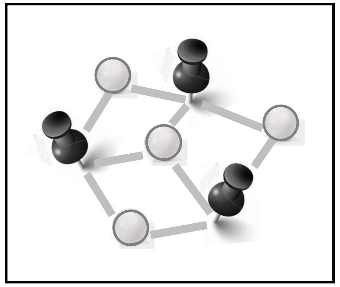


each node of T is either a cluster of G_c or a cycle of clusters of G_C

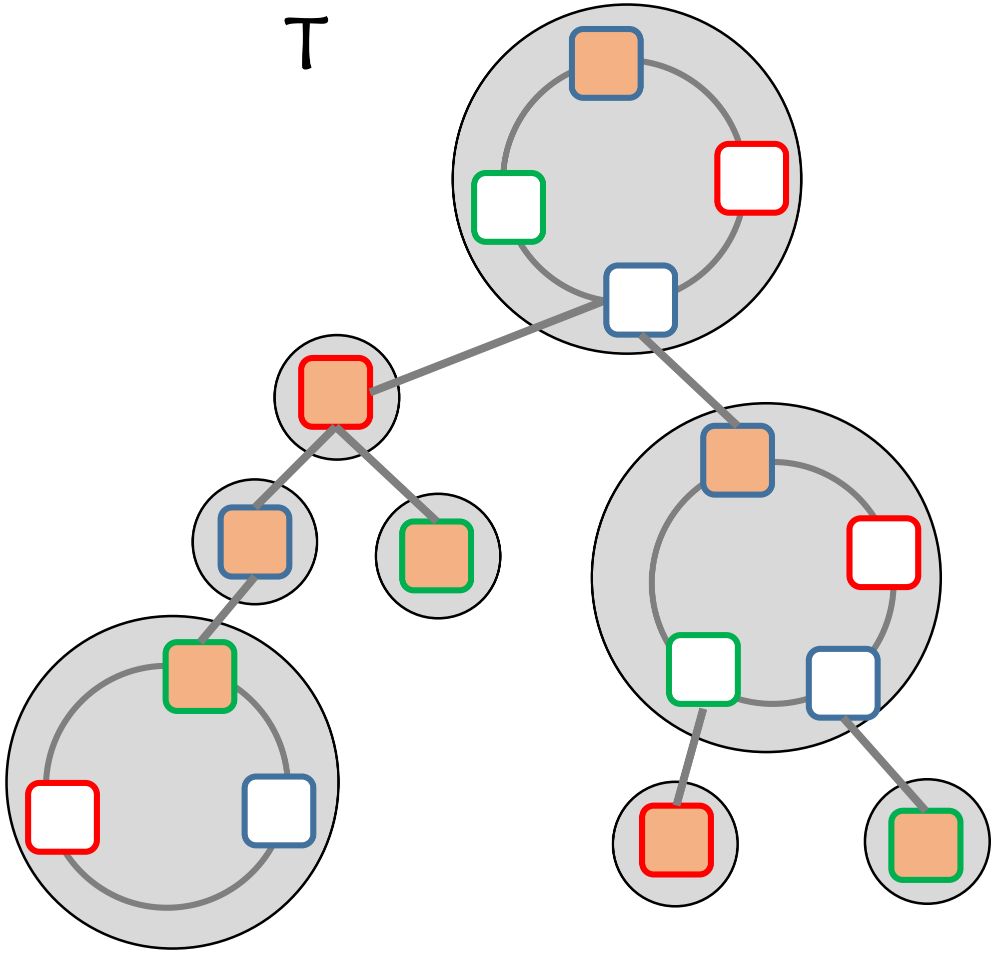


Result 2 – When G_X is a cactus

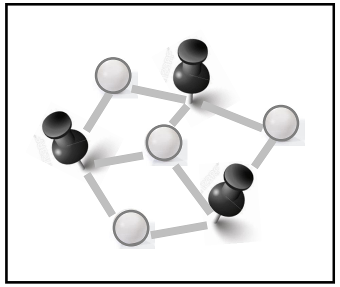




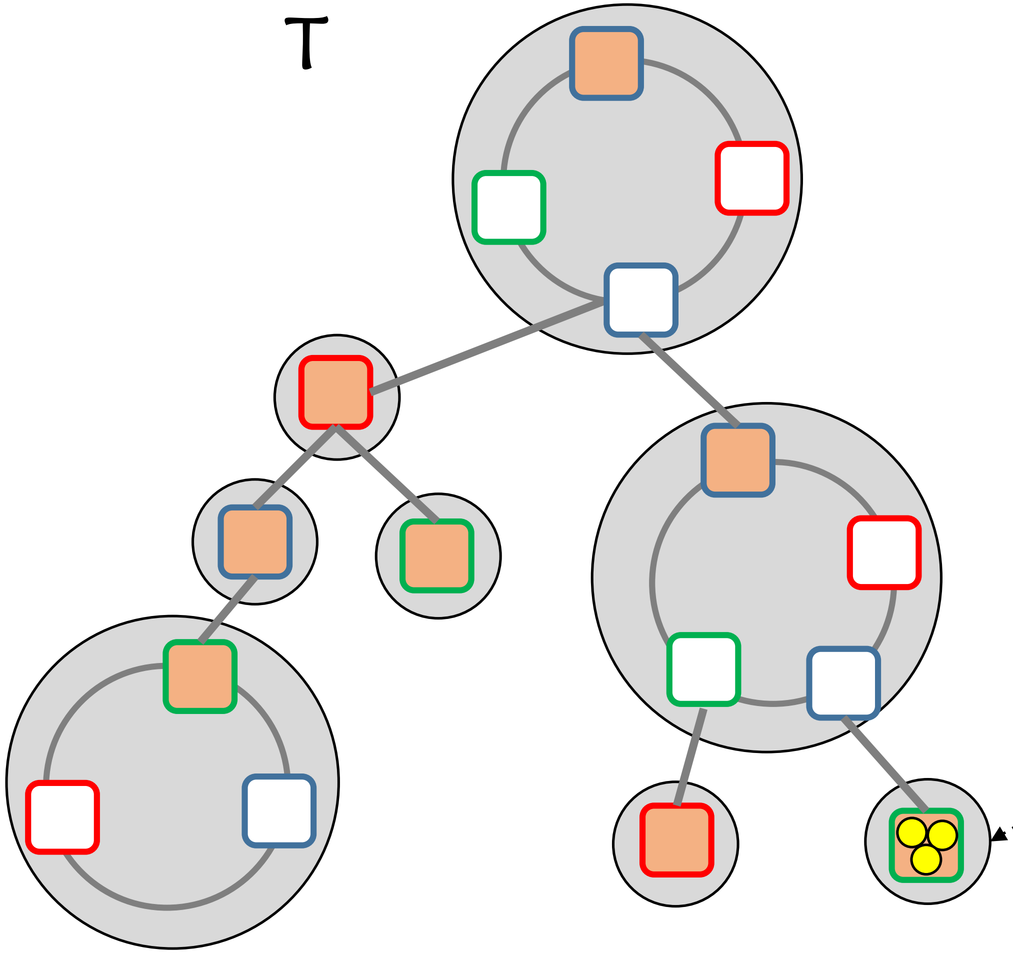
Result 2 – When G_X is a cactus



visit T <u>bottom-up</u> to test whether a skeleton exists

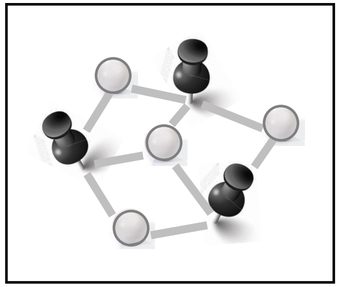


Result 2 – When G_X is a cactus



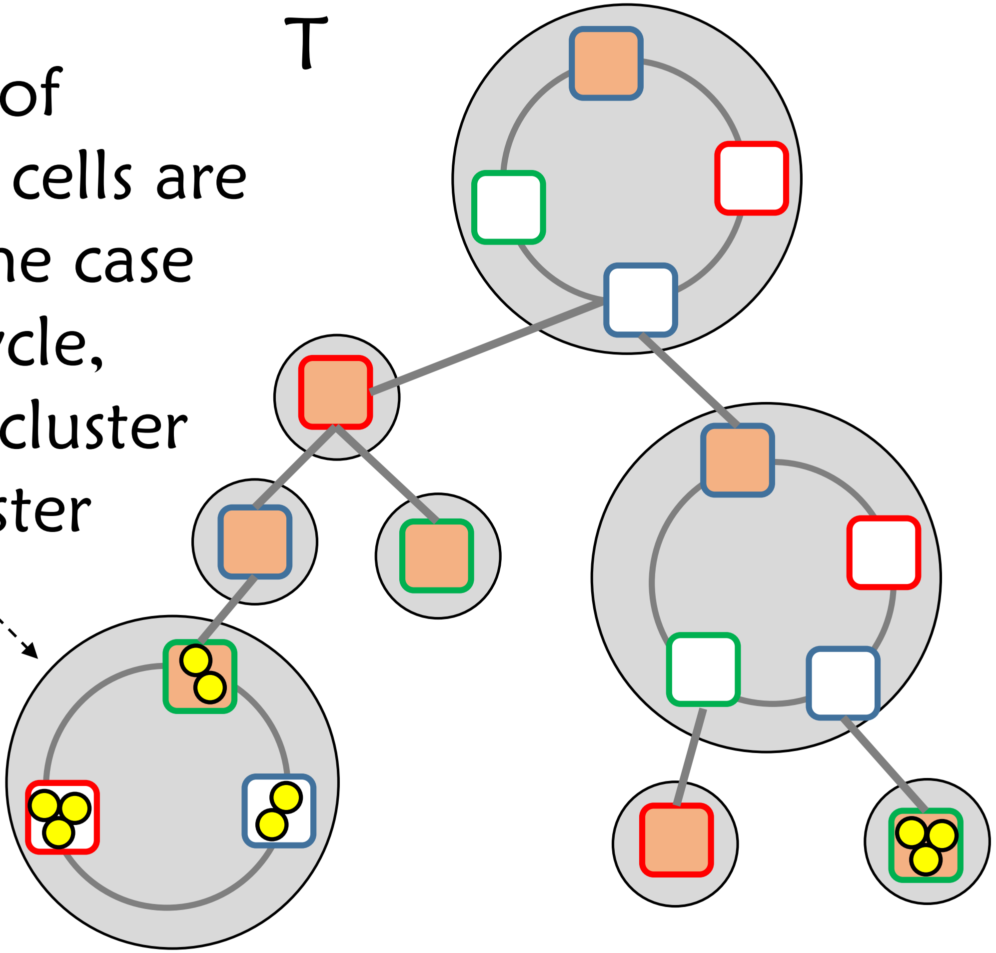
visit T bottom-up to test whether a skeleton exists

If a leaf is a single cluster, all its cells are made active

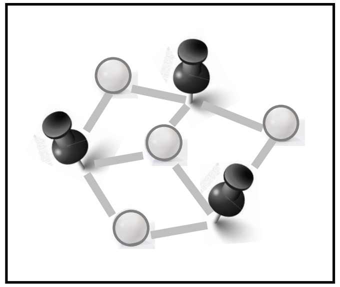


Result 2 – When G_X is a cactus

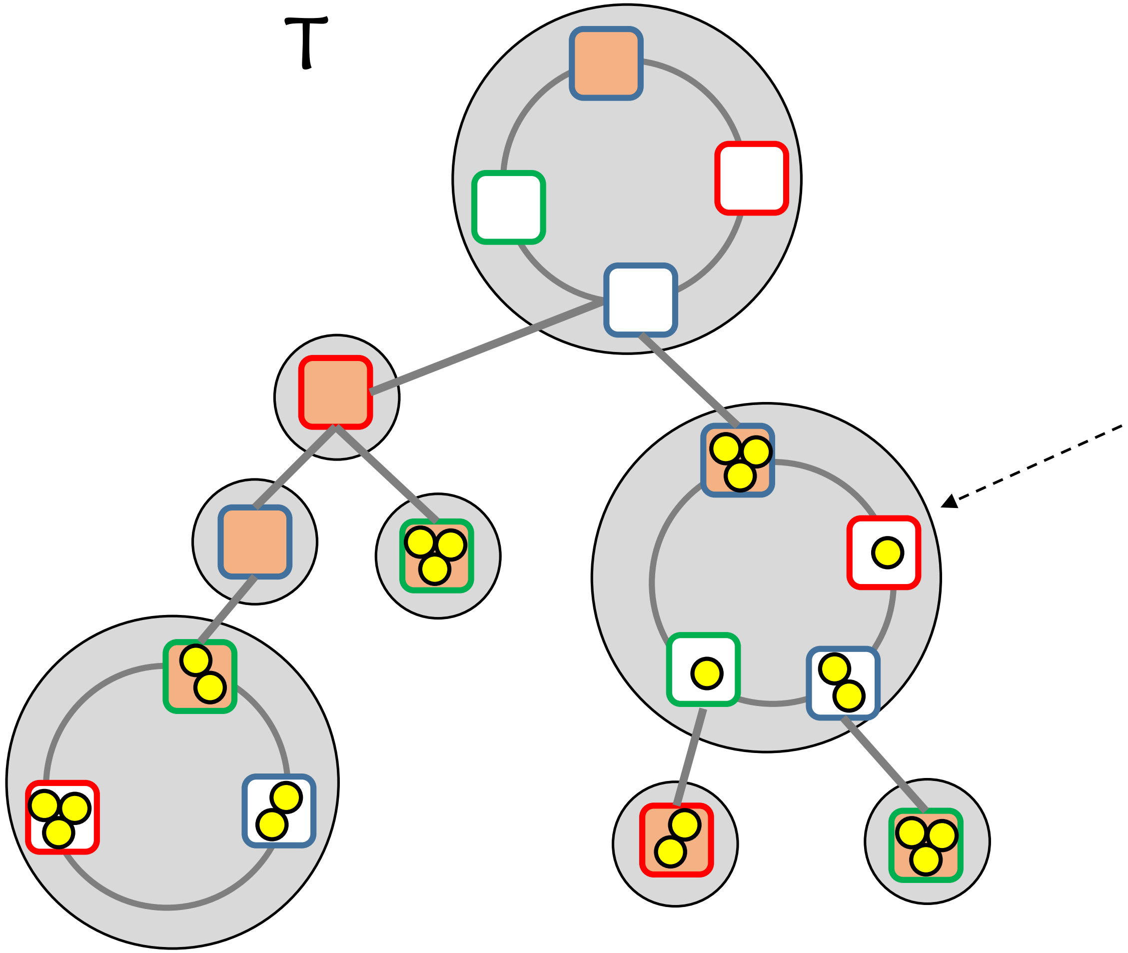
If a <u>leaf</u> is a cycles of clusters, the active cells are computed as for the case in which G_X is a cycle, where the anchor cluster acts as the first cluster



visit T <u>bottom-up</u> to test whether a skeleton exists



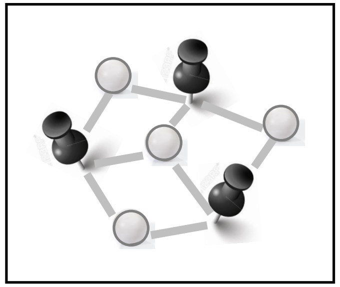
Result 2 – When G_X is a cactus



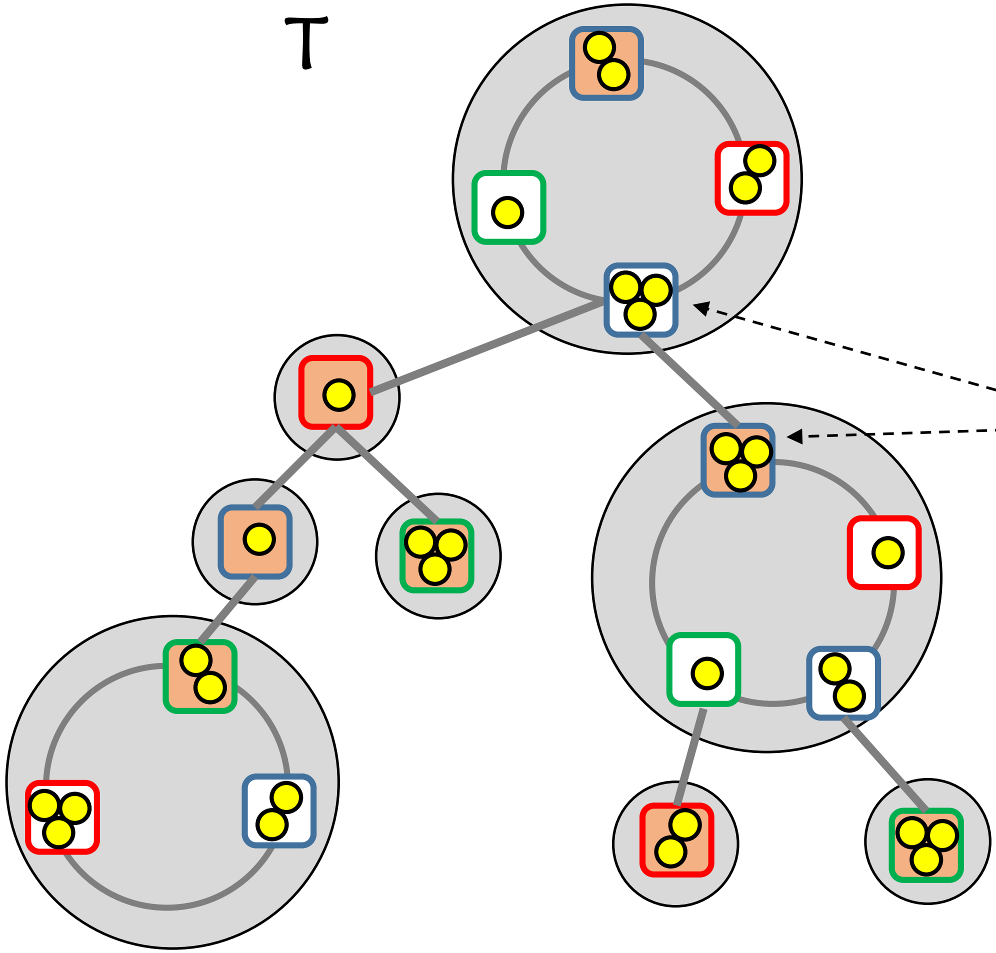
visit T bottom-up to test whether a skeleton exists

for an internal node:

1) first remove the cells not adjacent to active cells in the clusters connected to the anchor clusters of their children
2) then compute the active cells as for the leaves

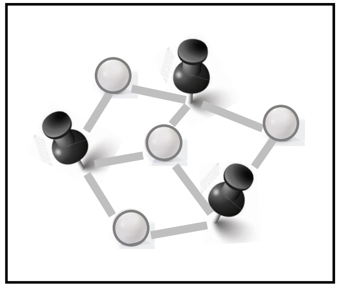


Result 2 – When G_X is a cactus

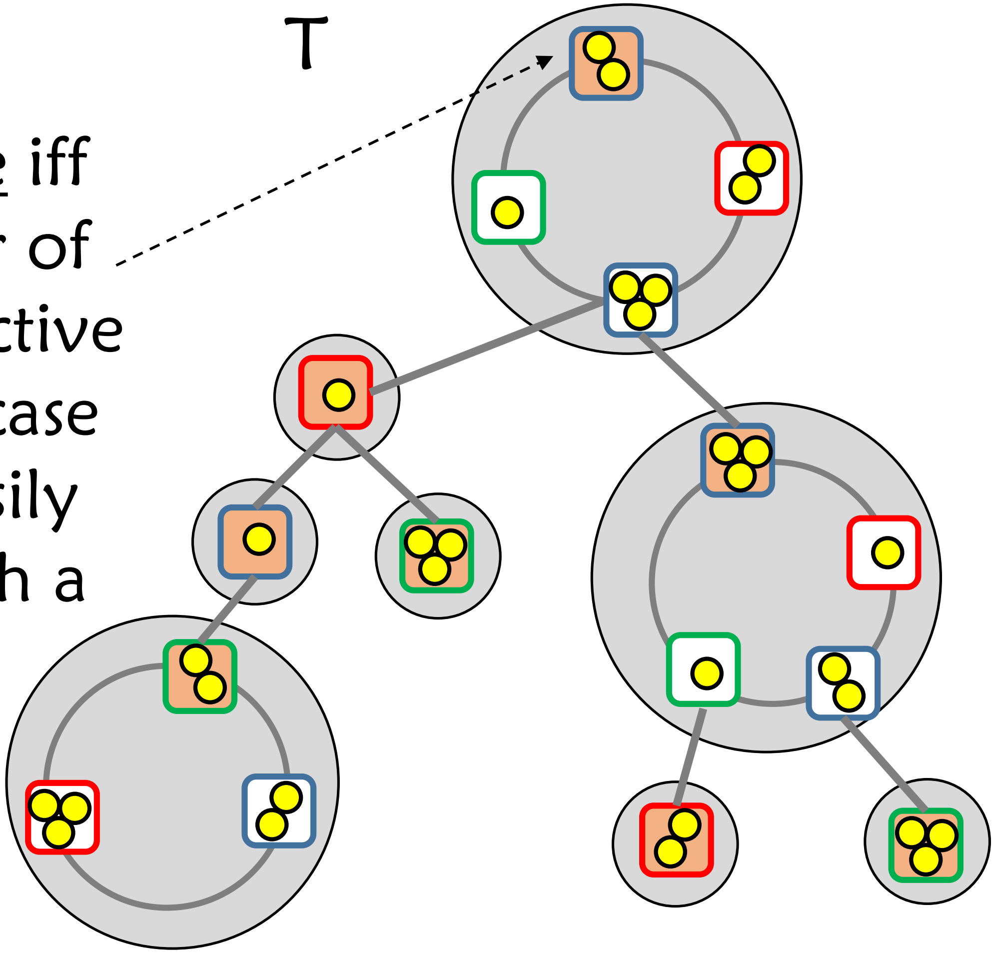


visit T <u>bottom-up</u> to test whether a skeleton exists

If two clusters coincide, remove from the father the non-active cells of its child

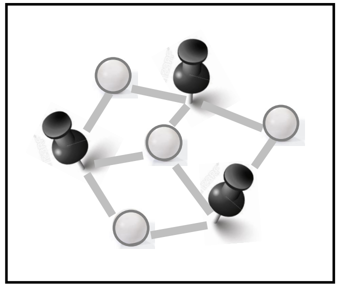


Result 2 – When G_X is a cactus

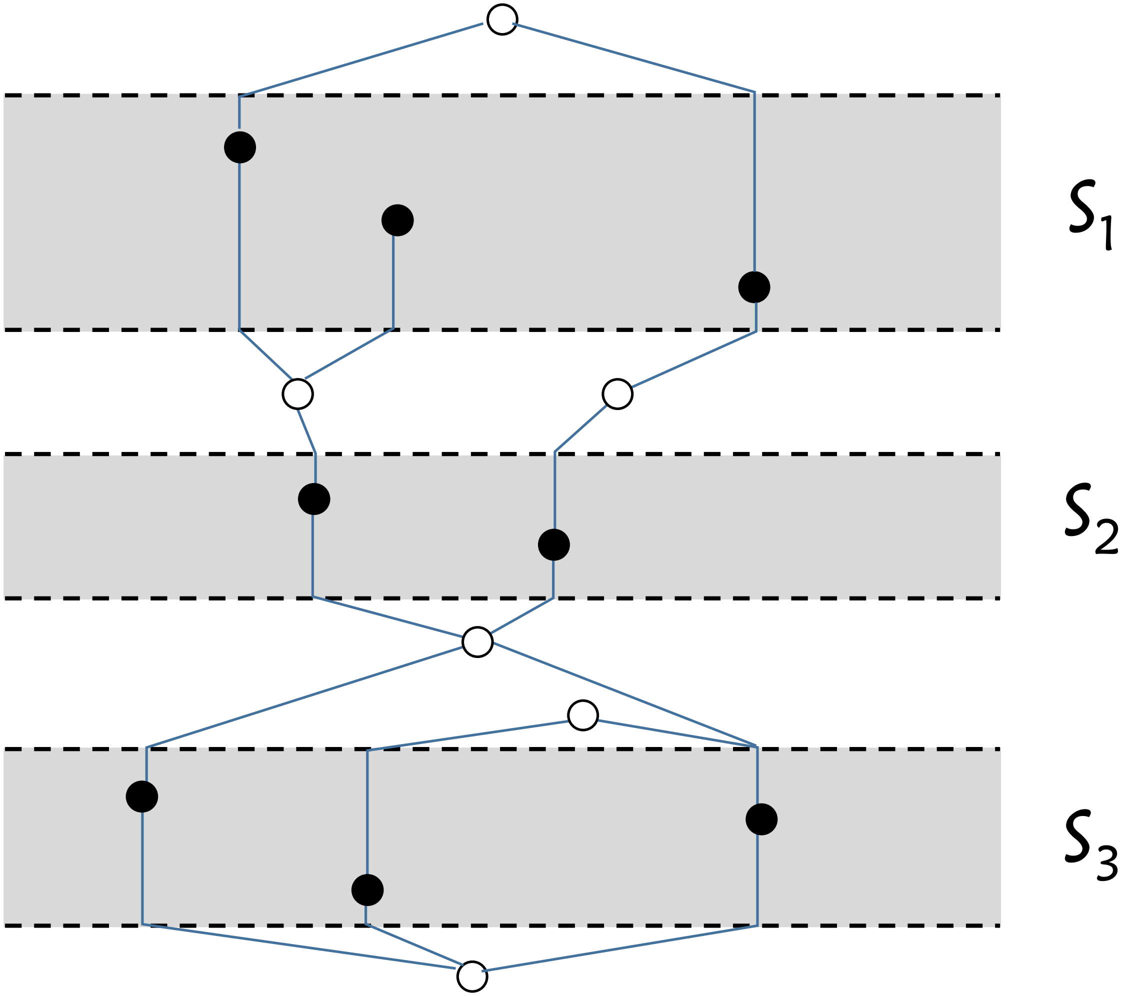


the test is positive iff the anchor cluster of the root has an active vertex; in which case the skeleton is easily reconstructed with a top-down visit

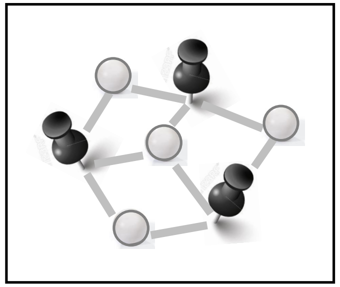
visit T bottom-up to test whether a skeleton exists



Result 3 – Model for 1-bend drawings



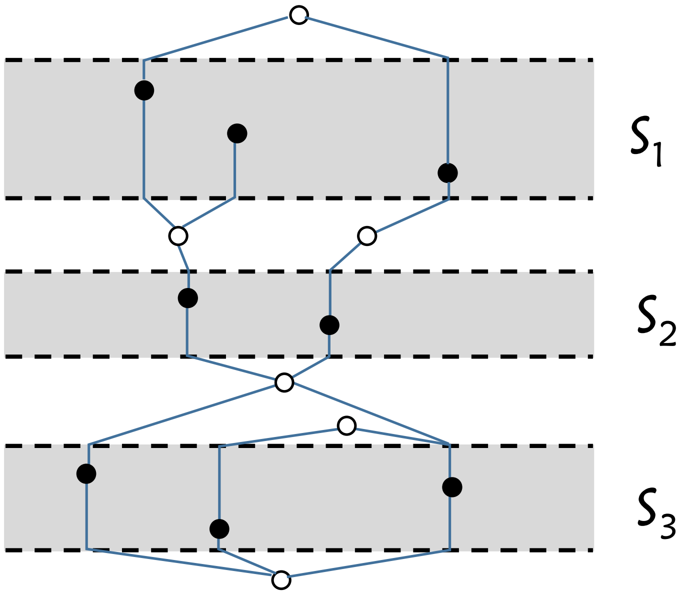
- fixed vertices are partitioned into a sequence of plane strips
- mobile vertices are placed outside the strips
- an edge cannot traverse a strip and must reach the fixed vertex with a vertical segment

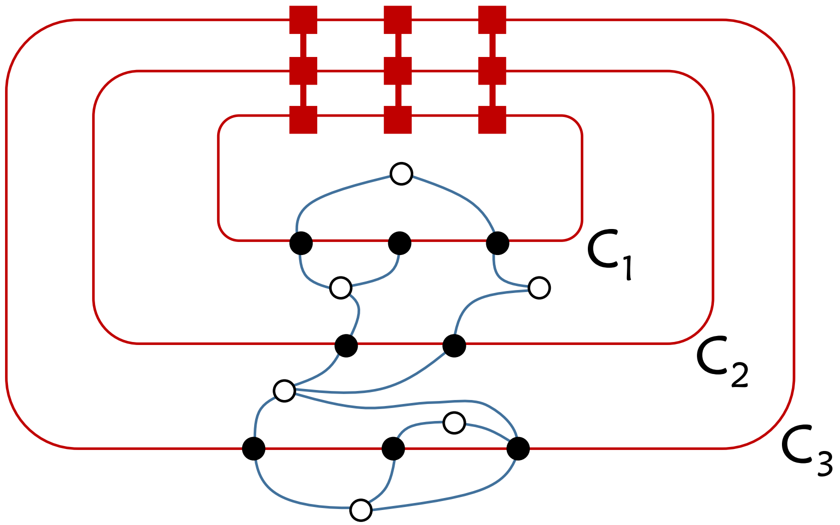


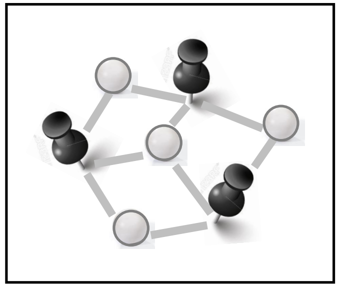
Result 3 – Model for 1-bend drawings

Theorem. For a given set of strips that partitions the fixed vertices, one can test in linear time whether a 1-bend drawing exists

Proof: It is equivalent to assume that the fixed vertices in each strip lie on a single horizontal line; then, reduce to planarity testing

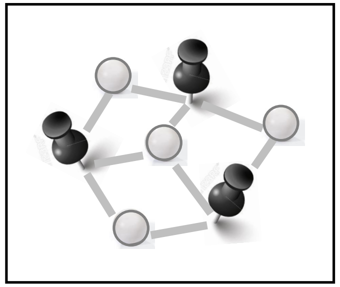






Open questions

- Question 1. Are there polynomial-time testing algorithms for 0-bend drawings in the CH restriction setting for families of G_X other than cacti?
- Question 2. Is it possible to find more efficient algorithms for 0-bend drawings?
- Question 3. What about relaxing the planarity requirement? e.g., by considering heuristics or exact algorithms for crossing/bend minimization?



Thanks