Movement problems on graphs

Stefano Leucci
stefano.leucci@graduate.univaq.it

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica, University of L'Aquila, Italy.

Scenario

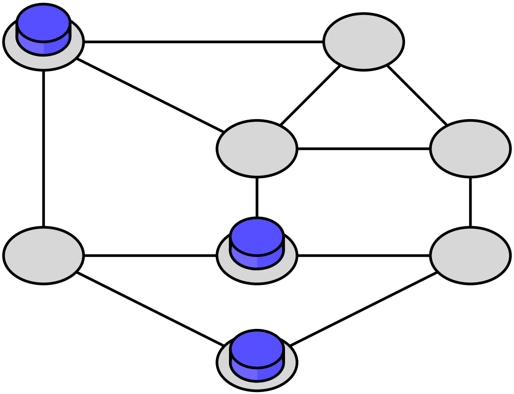
A central entity needs to plan the motion of a set P of agents (or *pebbles*) in a complex environment in order to reach a specific goal.

- The environment is modelled as an undirected graph G.
- Agents are placed on the vertices of G.
- We want to move the agents in order to reach a certain goal configuration (e.g. they must be on a clique of G).
- Moving an agent trough an edge costs 1 to the agent (e.g. one unit of energy, one unit of time, . . .).
- Amongst all feasible movements we want the one that minimizes a certain cost function, e.g. the sum of the agents' costs.

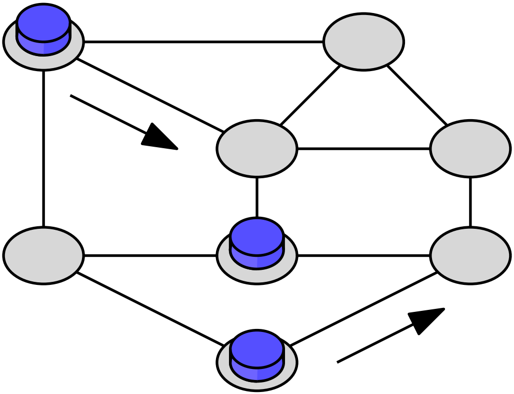
Assumptions

- Devices do not choose their trajectory autonomously: rather, their overall movement is planned by a central authority, and hence our focus is on the computational complexity of such a **centralized task**.
- Quite naturally, the pebbles should follow a *shortest path* in G.

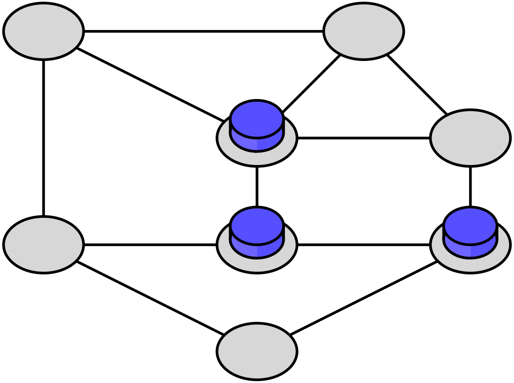
Example (Connectivity)



Example (Connectivity)



Example (Connectivity)



Motivation

- Robot motion planning:
  - Minimizing energy consumption.
  - Minimizing completion time.
- Radio-equipped agents: form a connected ad-hoc network (either single-hop or multi-hop).
- Moving antennas: build an interference-free networks.

Definition

An instance of the problem is defined as follows:

Input:

- An undirected, unweighed graph $G = (V, E)$ on n vertices.
- A set of k pebbles P.
- A function $\sigma : P \rightarrow V$ that assigns each pebble to its starting position.

Output:

- A function $\mu : P \rightarrow V$ that assigns each pebble to its final position, such that the set of final pebble positions achieves a certain goal.

Measure:

- A non-negative function that maps each feasible solution to its cost.

Goals

Let U be the set of the final position of the pebbles. We consider the following goals:

Connectivity(CON): the subgraph of G induced by the set U must be connected.

Independency (IND): U must be an independent set of size k ($|U| = k$) for G.
(Here we are not allowed to place more than one pebble on the same vertex).

Clique (CLIQUE): U must a clique of G.
(We are allowed to place more than on pebble on the same vertex).

Measures

Every pebble $p \in P$ is moved from its starting vertex $\sigma(p)$ to its end vertex $\mu(p)$ by using a shortest path on G.

Overall movement: sum of the distances travelled by pebbles.

$$\textsc{Sum}(\mu) = \sum_{p \in P} d_G(\sigma(p), \mu(p))$$

Maximum movement: maximum distance travelled by a pebble.

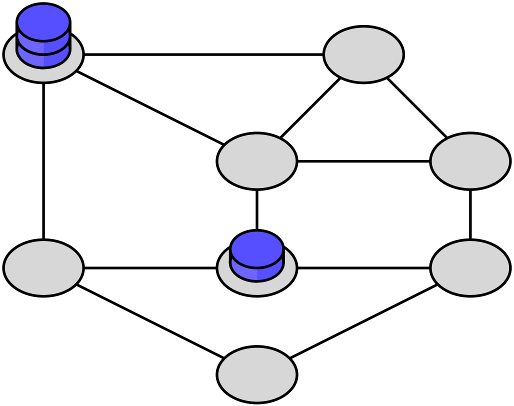
$$\textsc{Max}(\mu) = \max_{p \in P} d_G(\sigma(p), \mu(p))$$

Number of moved pebbles: number of pebbles that moved from their starting positions.

$$\textsc{Num}(\mu) = |\{p \in P : \sigma(p) \neq \mu(p)\}|$$

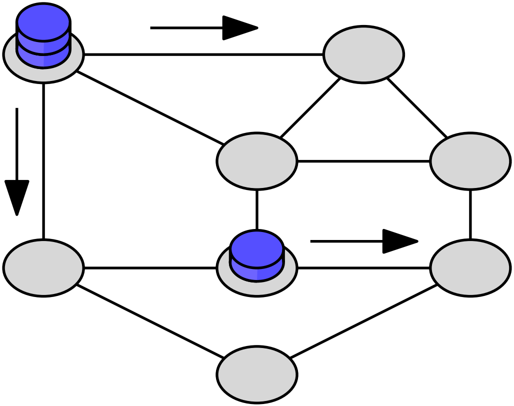
Example

Ind-Max.



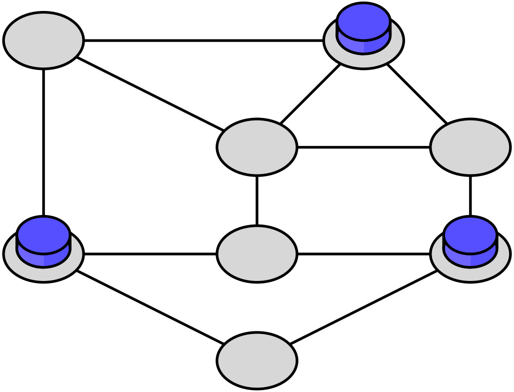
Example

Ind-Max.



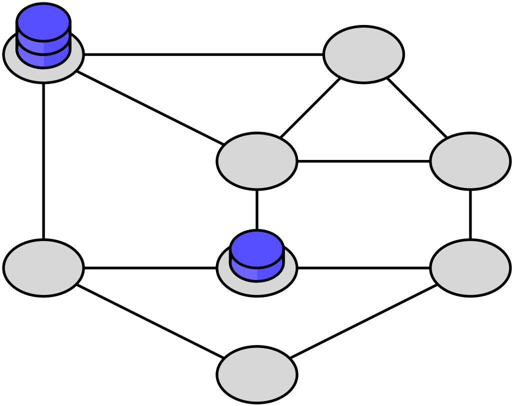
Example

Ind-Max. Cost=1



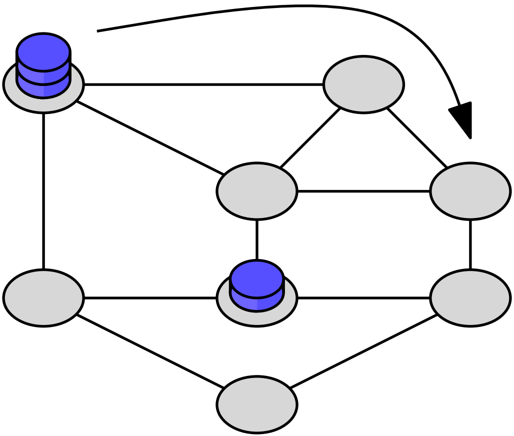
Example

Ind-Sum.



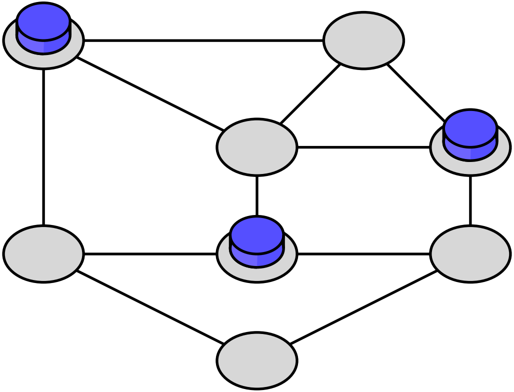
Example

Ind-Sum.



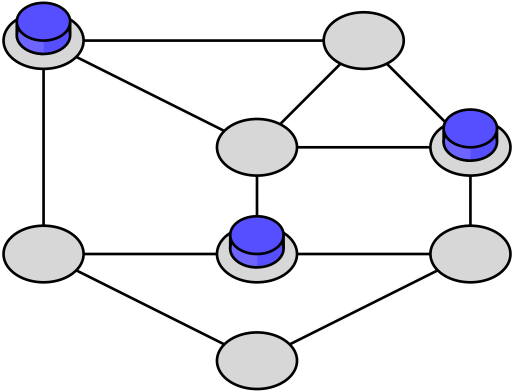
Example

Ind-Sum. Cost=2



Example

IND-NUM. Cost=1



Complexity results

All the movement problems defined here are NP-hard.

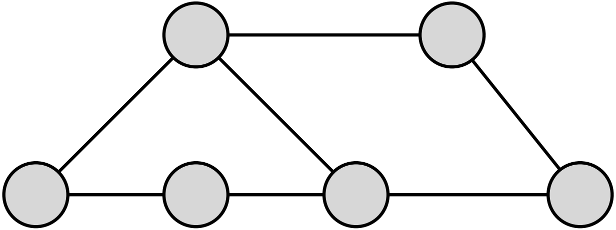
Some are known to admit a polynomial-time algorithms for special classes of graphs:

- All connectivity problems (Sum, Max, Num) on trees.
- Ind-Sum and Ind-Num on trees.
- Ind-Max on paths.
- Clique-Num on graphs where a maximum weight clique can be computed in polynomial time.

Independent set

Definition (Independent set)

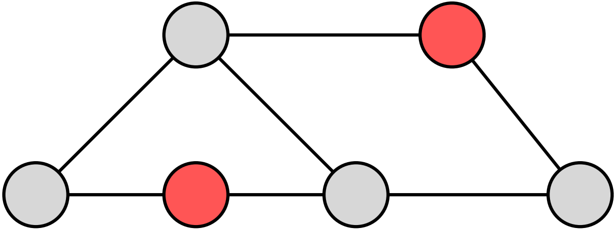
An *independent set* of a graph $G = (V, E)$ is a set of vertices $U \subseteq V$ that are pairwise non-adjacent, i.e. such that $\forall u, v \in U, (u, v) \notin E$.



Independent set

Definition (Independent set)

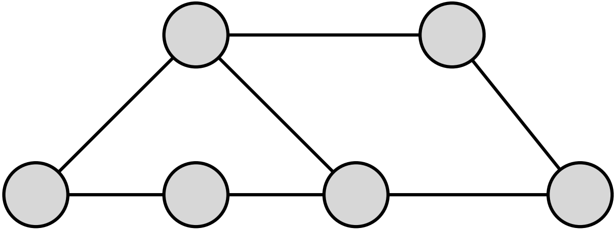
An *independent set* of a graph $G = (V, E)$ is a set of vertices $U \subseteq V$ that are pairwise non-adjacent, i.e. such that $\forall u, v \in U, (u, v) \notin E$.



Maximum independent set

Definition (Maximum independent set)

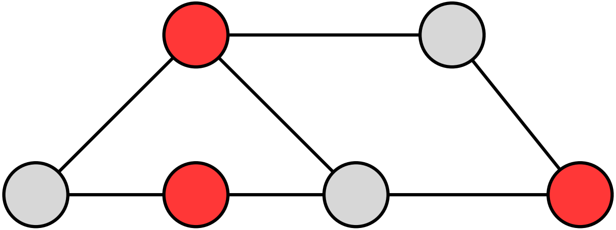
A *maximum independent set* of a graph $G = (V, E)$ is an independent set U^* of maximum cardinality, i.e. such that for every other independent set U we have $|U^*| \geq |U|$.



Maximum independent set

Definition (Maximum independent set)

A *maximum independent set* of a graph $G = (V, E)$ is an independent set U^* of maximum cardinality, i.e. such that for every other independent set U we have $|U^*| \geq |U|$.



Maximum independent set

- On general graphs the problem of finding a maximum independent set is NP-hard.
- The *decision version* of this problem requires determining if there exists an independent set of at least a certain size.
- In independency motion problems we need to find an independent set of size at least $|P|$.
- This means that it is NP-hard *even to find a feasible solution*.
- **Idea:** We restrict to classes of graphs where a maximum independent set can be computed in polynomial time.

Maximum independent set

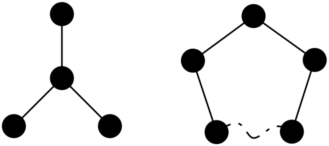
- On general graphs the problem of finding a maximum independent set is NP-hard.
- The *decision version* of this problem requires determining if there exists an independent set of at least a certain size.
- In independency motion problems we need to find an independent set of size at least $|P|$.
- This means that it is NP-hard *even to find a feasible solution*.
- **Idea:** We restrict to classes of graphs where a maximum independent set can be computed in polynomial time.

Bad news: the problem is still hard!

Special classes of graphs

A maximum independent set can be found in polynomial time on:

- Paths
- Trees
- **Bipartite graphs**
- Claw-free graphs (no induced claws)
- Perfect graphs
- ...



A claw and an hole.

Definition (Perfect graph)

A graph G is perfect if neither G nor it's complement have odd holes.

Hardness of IND-MAX

Polynomial reduction from the 3-SAT problem to IND-MAX.

Ingredients of 3-SAT:

- A set $X = \{x_1, x_2, \dots\}$ of boolean variables.
- A literal is either an asserted or a negated variable.
- A clause is a disjunction of three literals.
- A formula f is a conjunction of clauses.

The 3-SAT problem: **There exists a truth assignment to the variables so that f is true?**

Hardness of Ind-Max

Polynomial reduction from the 3-SAT problem to Ind-Max.

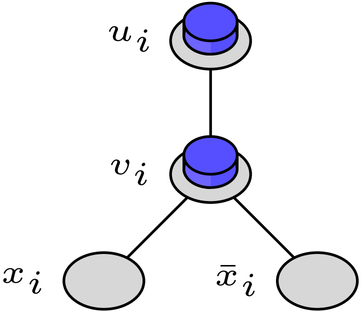
Ingredients of 3-SAT:

- A set $X = \{x_1, x_2, \dots\}$ of boolean variables. E.g. $X = \{x_1, x_2, x_3, x_4\}$.
- A literal is either an asserted or a negated variable. E.g. x_1, $\bar{x}_3$, $\bar{x}_1$, x_2, $\dots$.
- A clause is a disjunction of three literals. E.g. $(x_1 \vee \bar{x}_2 \vee x_4)$, $(\bar{x}_1 \vee \bar{x}_2 \vee x_3)$.
- A formula f is a conjunction of clauses. E.g. $(x_1 \vee \bar{x}_2 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$.

The 3-SAT problem: **There exists a truth assignment to the variables so that f is true?**

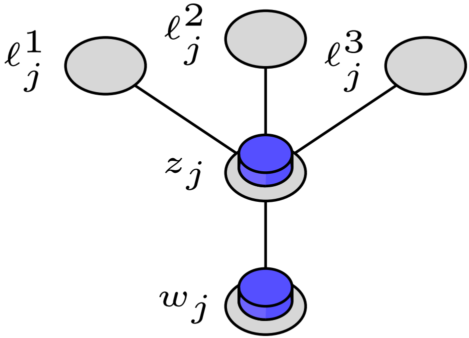
The variable gadget

For each variable x_i of f we build the following "variable" gadget:



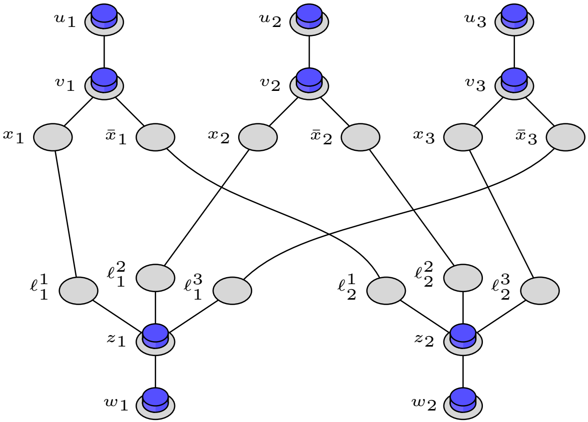
The clause gadget

For each clause $c_j = (\ell_j^1, \ell_j^2, \ell_j^3)$ of f we build the following "clause" gadget:



Putting all together

For each clause $c_j = (\ell_j^1, \ell_j^2, \ell_j^3)$ of f we connect each literal to the **opposite** node of the corresponding variable gadget.



$$(\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$$

Completing the proof (forward)

Claim

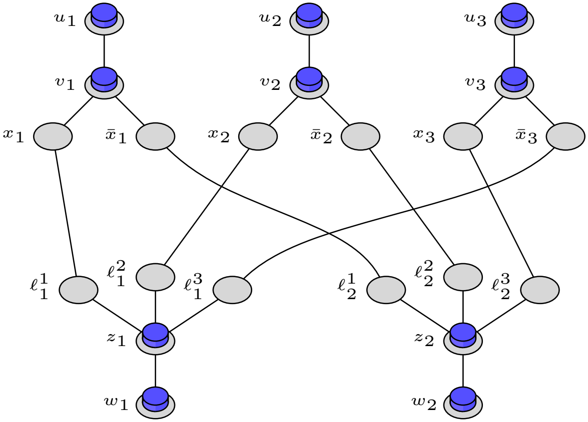
The formula f can be satisfied $\iff$ there exists a solution for the IND-MAX instance of cost 1.

Proof (forward).

- Consider a truth assignment for f.
- For each variable x_i, if x_i is asserted move the pebble starting on v_i to the vertex x_i, otherwise move it to $\bar{x}_i$.
- For each clause $(\ell_j^1, \ell_j^2, \ell_j^3)$ there must at least literal ℓ_j^k that is true.
- This means that the vertex of the variable gadget that is adjacent to ℓ_j^k does not contain pebble.
- Move the pebble starting on z_j to ℓ_j^k.

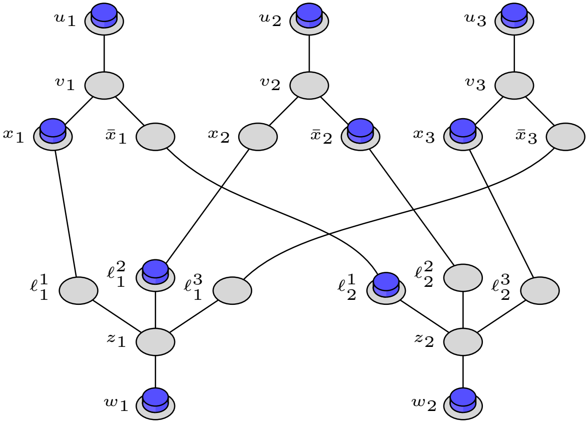
□

Completing the proof



$$(\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$$

Completing the proof



$$(\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$$

$$x_1 = \top,\ x_2 = \bot,\ x_3 = \top$$

Completing the proof (backward)

Claim

The formula f can be satisfied $\iff$ there exists a solution for the IND-MAX instance of cost 1.

Proof (backward).

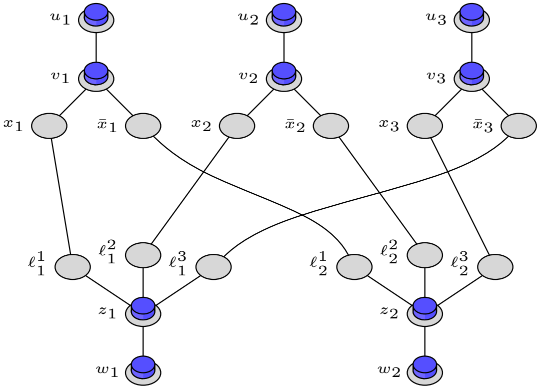
- Consider a solution or the IND-MAX instance of cost 1.
- Each pebble starting on v_i must have been moved to either x_i or $\bar{x}_i$. Set the truth value of the variable x_i accordingly.
- For each clause, the pebble starting on z_j must have been moved to a vertex $\ell_j^k \in \{\ell_j^1, \ell_j^2, \ell_j^3\}$.
- This means that the vertex of the variable gadget that is adjacent to ℓ_j^k does not contain a pebble.
- Therefore ℓ_j^k, and the whole clause are satisfied.

□

Completing the proof

Theorem

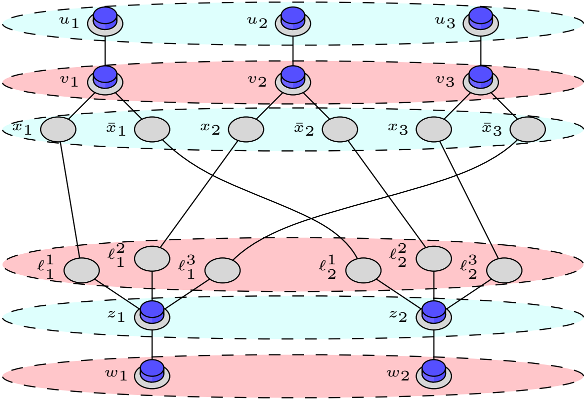
The problem Ind-Max *is* NP*-hard.*



Completing the proof

Theorem

The problem IND-MAX *is* NP*-hard. This holds even when G is a bipartite graph.*



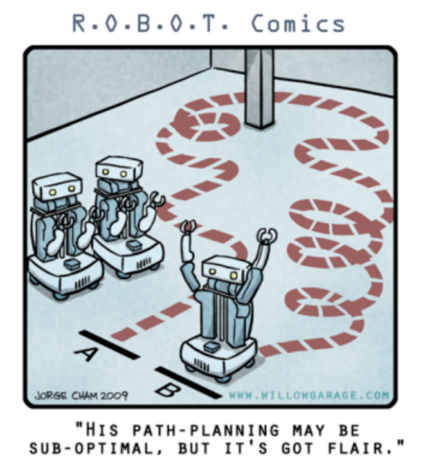
Approximability of IND-MAX

Theorem

If a maximum independent set of G can be found in polynomial time (e.g. on perfect graphs), IND-MAX *can be approximated within an additive error of* 1*.*

That's the best we could possibly do in polynomial time! (unless $\mathsf{P} = \mathsf{NP}$).

Approximability of IND-MAX

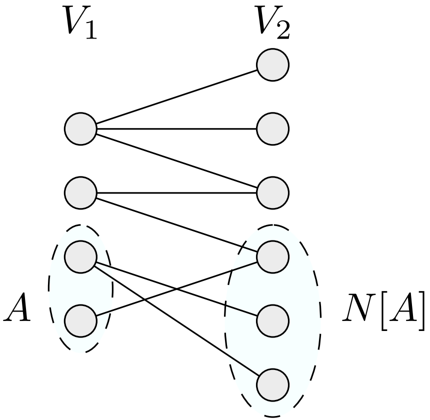


We will need...

Theorem (Hall's Matching Theorem)

Let $H = (V_1 + V_2, E)$ be a bipartite graph. There exists a matching of size $|V_1|$ on H iff $|A| \leq |N_H(A)|$, $\forall A \subseteq V_1$.

$N_H(A)$ and $N_H[A]$ are the open and the closed neighborhood of A, respectively.

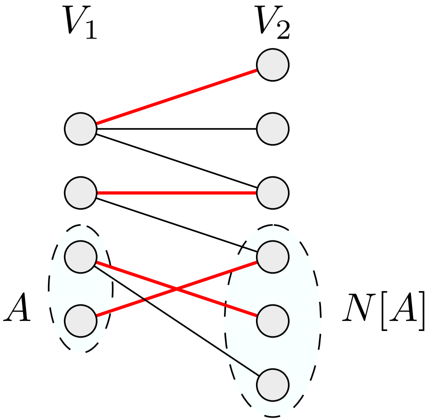


We will need...

Theorem (Hall's Matching Theorem)

Let $H = (V_1 + V_2, E)$ be a bipartite graph. There exists a matching of size $|V_1|$ on H iff $|A| \leq |N_H(A)|$, $\forall A \subseteq V_1$.

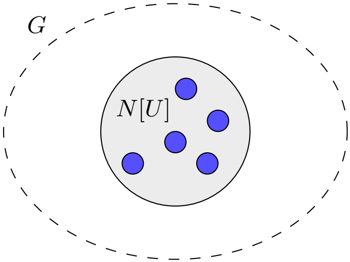
$N_H(A)$ and $N_H[A]$ are the open and the closed neighborhood of A, respectively.



Approximability of IND-MAX

Lemma

Let U^* be a maximum independent set of G. For each independent set U of G: $|U^* \cap N_G[U]| \geq |U|$.



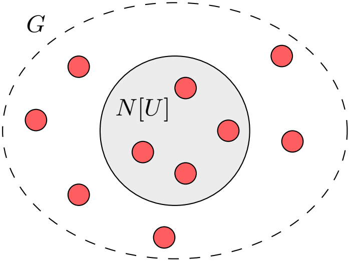
Approximability of IND-MAX

Lemma

Let U^* be a maximum independent set of G. For each independent set U of G: $|U^* \cap N_G[U]| \geq |U|$.

Proof.

Suppose $|U^* \cap N_G[U]| < |U|$.



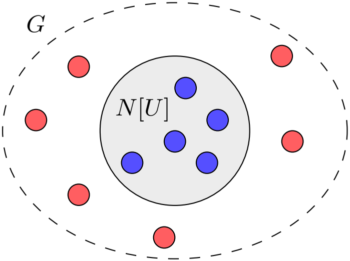
Approximability of IND-MAX

Lemma

Let U^* be a maximum independent set of G. For each independent set U of G: $|U^* \cap N_G[U]| \geq |U|$.

Proof.

Suppose $|U^* \cap N_G[U]| < |U|$.
$U' = (U^* \setminus N_G[U]) \cup U$ is an independent set.



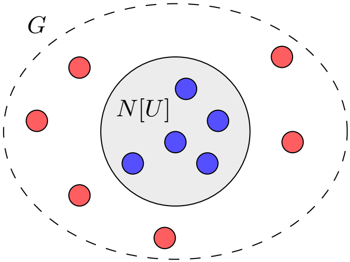
Approximability of Ind-Max

Lemma

Let U^* be a maximum independent set of G. For each independent set U of G: $|U^* \cap N_G[U]| \geq |U|$.

Proof.

Suppose $|U^* \cap N_G[U]| < |U|$.
$U' = (U^* \setminus N_G[U]) \cup U$ is an independent set.
We have $|U'| > |U^*| \Rightarrow\Leftarrow$ □



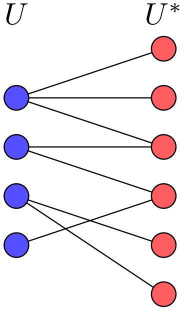
Approximability of Ind-Max

Lemma

For each independent set U of G, there exists an injective function $f : U \to U^*$ such that $d_G(u, f(u)) \leq 1$.

Proof.

Construct the bipartite graph $H = (U + U^*, E)$ and connect each vertex $u \in U$ to $U^* \cap N[\{u\}]$.



Approximability of IND-MAX

Lemma

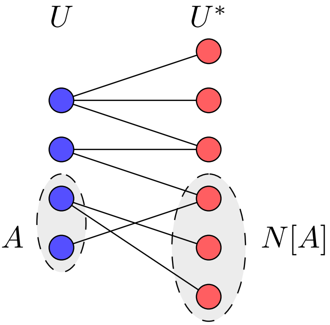
For each independent set U of G, there exists an injective function $f : U \to U^*$ such that $d_G(u, f(u)) \leq 1$.

Proof.

Construct the bipartite graph $H = (U + U^*, E)$ and connect each vertex $u \in U$ to $U^* \cap N[\{u\}]$.

$\forall A \subseteq U$, $N(A) = |U^* \cap N_G[A]| \geq |A|$.

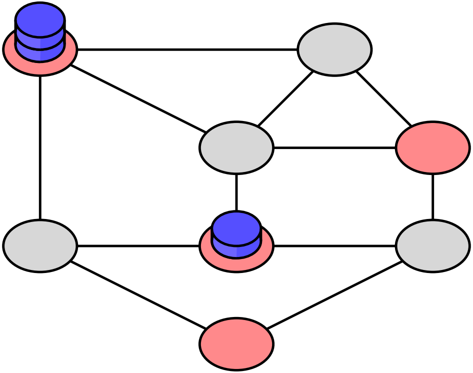
Claim follows using Hall's Matching Theorem. □



Approximability of Ind-Max

There exists a solution of cost OPT+1 that places all the pebbles on U^*.

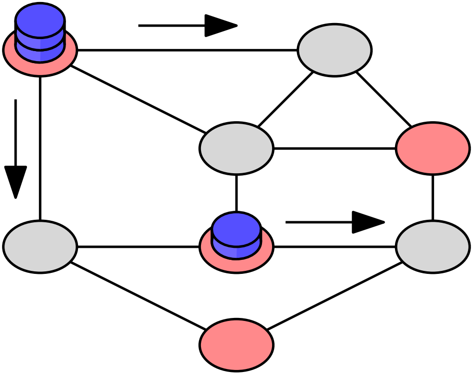
Cost = 0



Approximability of IND-MAX

There exists a solution of cost OPT+1 that places all the pebbles on U^*.

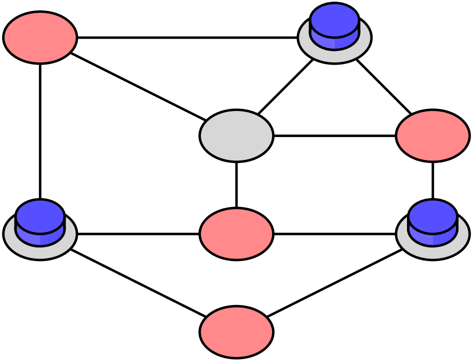
Cost = 0



Approximability of Ind-Max

There exists a solution of cost OPT+1 that places all the pebbles on U^*.

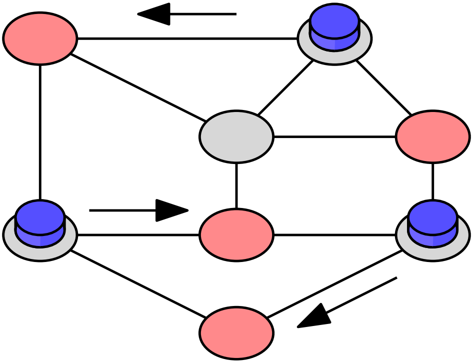
Cost = OPT



Approximability of Ind-Max

There exists a solution of cost OPT+1 that places all the pebbles on U^*.

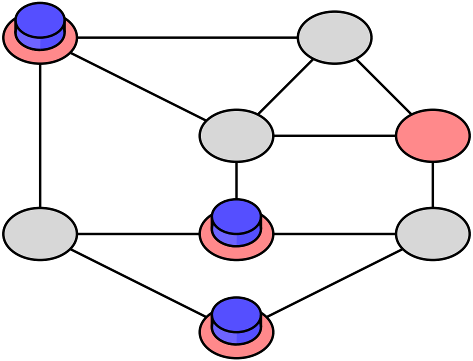
Cost = OPT



Approximability of IND-MAX

There exists a solution of cost OPT+1 that places all the pebbles on U^*.

Cost = OPT+1



Approximability of IND-MAX

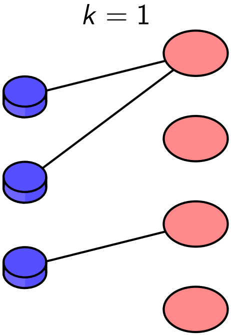
Theorem

If a maximum independent set of G can be found in polynomial time, there exists a polynomial-time algorithm that approximates IND-MAX *within an additive error of* 1.

$U^* \leftarrow$ `MaximumIndependentSet(`G`)`

if $|U^*| < |P|$ **then**
  return No solution

for $k \leftarrow 0$ **to** $|V| - 1$ **do**
  $F \leftarrow \{(p, u) \in P \times U^* | d(\sigma(p), u) \leq k\}$
  $H \leftarrow (P + U^*, F)$



Approximability of IND-MAX

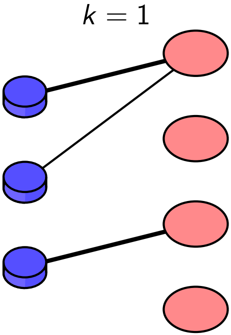
Theorem

If a maximum independent set of G can be found in polynomial time, there exists a polynomial-time algorithm that approximates IND-MAX *within an additive error of* 1.

$U^* \leftarrow$ `MaximumIndependentSet(`G`)`

if $|U^*| < |P|$ **then**
  return No solution

for $k \leftarrow 0$ **to** $|V| - 1$ **do**
  $F \leftarrow \{(p, u) \in P \times U^* | d(\sigma(p), u) \leq k\}$
  $H \leftarrow (P + U^*, F)$
  $\mathcal{M} \leftarrow$ `MaximumBipartiteMatching(`H`)`



Approximability of IND-MAX

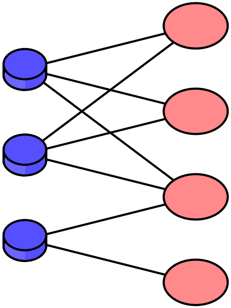
Theorem

If a maximum independent set of G can be found in polynomial time, there exists a polynomial-time algorithm that approximates IND-MAX *within an additive error of* 1.

$U^* \leftarrow$ `MaximumIndependentSet(`G`)`

if $|U^*| < |P|$ **then**
  return No solution

for $k \leftarrow 0$ **to** $|V| - 1$ **do**
  $F \leftarrow \{(p, u) \in P \times U^* | d(\sigma(p), u) \leq k\}$
  $H \leftarrow (P + U^*, F)$
  $\mathcal{M} \leftarrow$ `MaximumBipartiteMatching(`H`)`
  if $|\mathcal{M}| = |P|$ **then**
    return $\mathcal{M}$



Approximability of IND-MAX

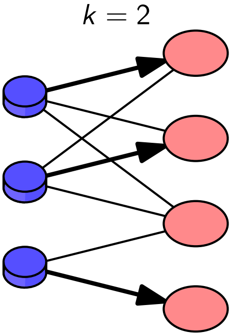
Theorem

If a maximum independent set of G can be found in polynomial time, there exists a polynomial-time algorithm that approximates IND-MAX *within an additive error of* 1.

$U^* \leftarrow$ `MaximumIndependentSet(`G`)`

if $|U^*| < |P|$ **then**
  return No solution

for $k \leftarrow 0$ **to** $|V| - 1$ **do**
  $F \leftarrow \{(p, u) \in P \times U^* | d(\sigma(p), u) \leq k\}$
  $H \leftarrow (P + U^*, F)$
  $\mathcal{M} \leftarrow$ `MaximumBipartiteMatching(`H`)`
  if $|\mathcal{M}| = |P|$ **then**
    return $\mathcal{M}$



References

- D. Bilò, L. Gualà, S. Leucci, and G. Proietti, Exact and approximate algorithms for movement problems on (special classes of) graphs, *SIROCCO'13*.

Further readings:

- E.D. Demaine, M. Hajiaghayi, H. Mahini, A.S. Sayedi-Roshkhar, S. Oveisgharan, and M. Zadimoghaddam, Minimizing movement, *SODA'07*.
- P. Berman, E.D. Demaine, and M. Zadimoghaddam, O(1)-approximations for maximum movement problems, *APPROX-RANDOM'11*.