Project Group "DynaSearch"
Final Presentation

Table of Contents

Introduction

Project Group

Our Work in the CRC 901

1. Big software from small pieces – Search for pieces that
   - Maximize some objective function or
   - Fulfill certain properties
2. Communicating entities with varying interests
   - Adapt network to these interests

Objective Function Search in P2P Networks

Introduction

Motivation

- Data items have several attributes
- Scenario: User specifies what is important to him
- Does not know which items exist
- Wants best possible result

Formal Definition

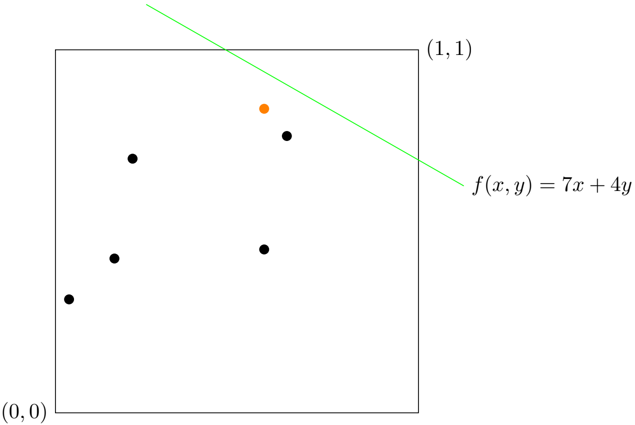
General:

- Data items in $[0,1)^d$
- Request is function $f : [0,1)^d \to \mathbb{R}$

For now:

- $d = 2$
- f is linear: $f(x,y) := a_1 x + a_2 y$

Example



First Idea

- Move sweep line through coordinate space
- Start at best corner
- Result is first item found

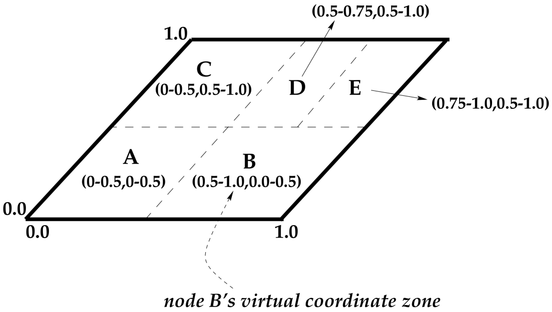
What do we need?

- Manage coordinate space in p2p system
- Efficient way of searching

Basic P2P System

- Use Content Addressable Network (CAN)
- Manages data items in $[0, 1)^d$ coordinate space
- Each node responsible for section of space

CAN Example



2-dimensional CAN with 5 nodes

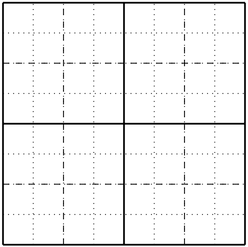
How to search?

- Many data items $\Rightarrow$ Result at corner
- Few data items $\Rightarrow$ Large empty sections
- Want to skip empty sections quickly

$\Rightarrow$ Meta structure with containment information

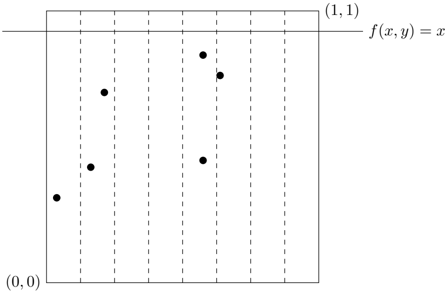
Hierarchy Meta Structure

- Create tree structure with containment information
- Root node responsible for whole coordinate space
- Partition recursively
- Node knows whether some child contains data item



Network Balance

Need to make some assumptions about network structure:



Network Balance

Introduce *c-balance*:

- $s :=$ shortest side length of any CAN-area
- $\ell :=$ longest side length of any CAN-area
- $c := \frac{\ell^2}{s^2}$

Objective Function Search in P2P Networks

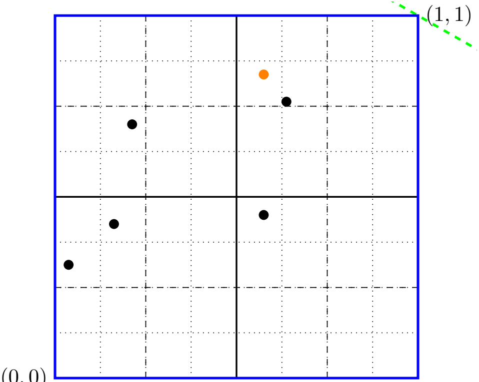
Algorithms

Algorithm FindMax

Basic Idea

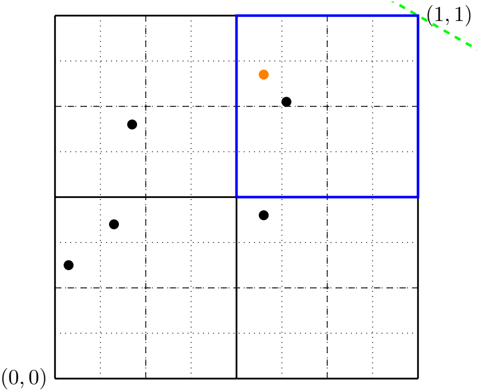
- Approach
  1. Start at root of hierarchy
  2. If area contains data item: search child areas; else: skip area
- Technique
  1. Sequentially process areas
  2. Best areas are processed first
  3. Areas with higher hierarchy level are preferred

Algorithm FINDMAX
Illustration



Initial scenario and Step 1

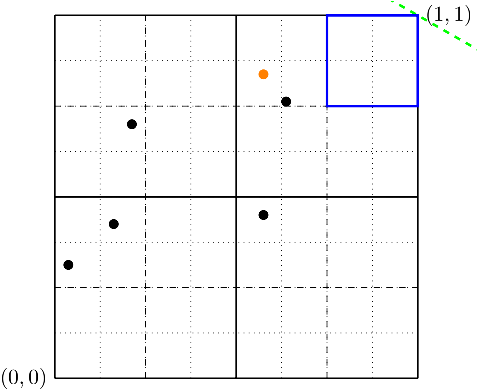
Algorithm FindMax
Illustration



Step 2

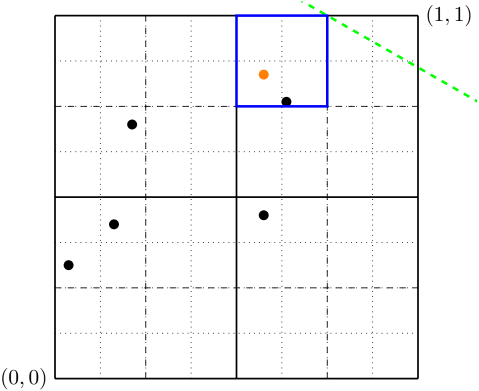
Algorithm FindMax

Illustration



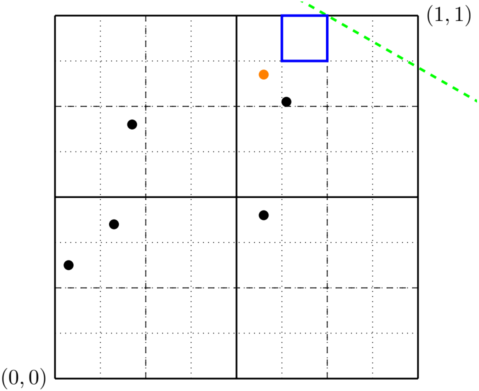
Step 3

Algorithm FindMax
Illustration



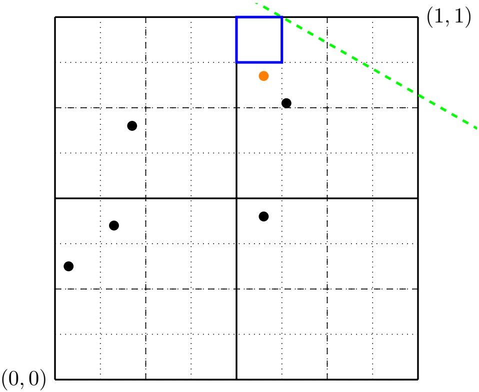
Step 4

Algorithm FindMax
Illustration



Step 5

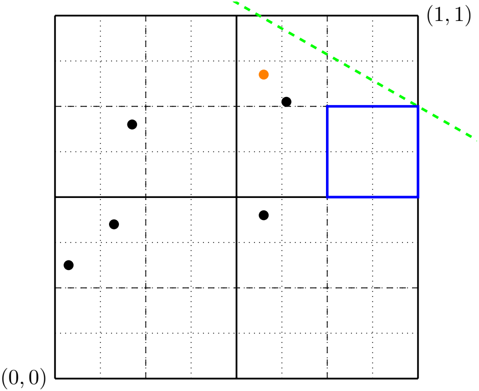
Algorithm FindMax
Illustration



Step 6

Algorithm FINDMAX

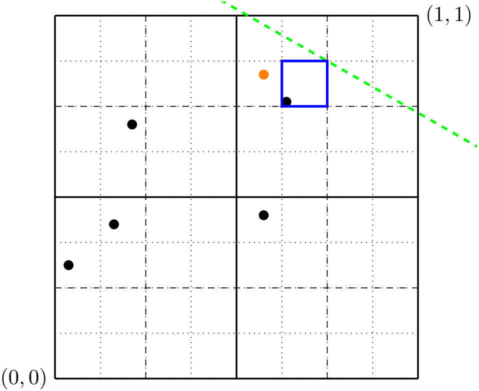
Illustration



Step 7

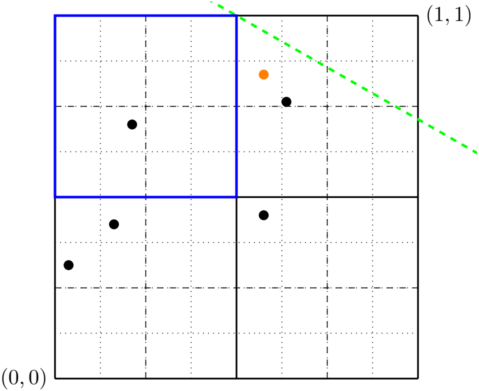
Algorithm FINDMAX

Illustration



Step 8

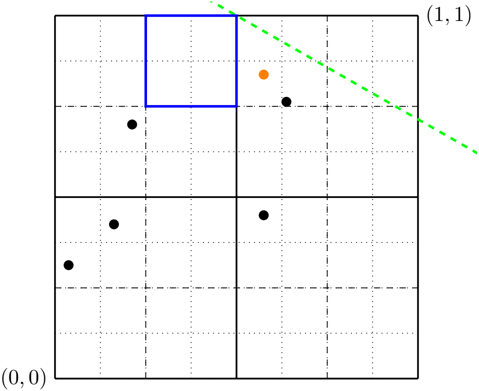
Algorithm FindMax
Illustration



Step 9

Algorithm FINDMAX

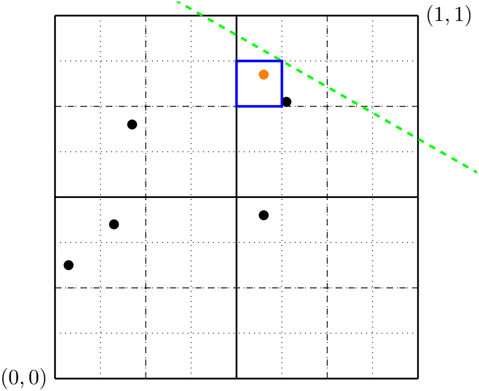
Illustration



Step 10

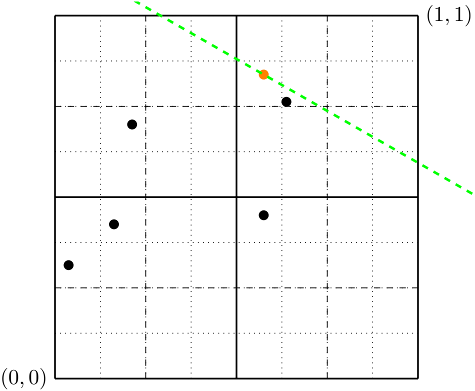
Algorithm FINDMAX

Illustration



Step 11

Algorithm FindMax
Illustration



For comparison only: Optimality proven

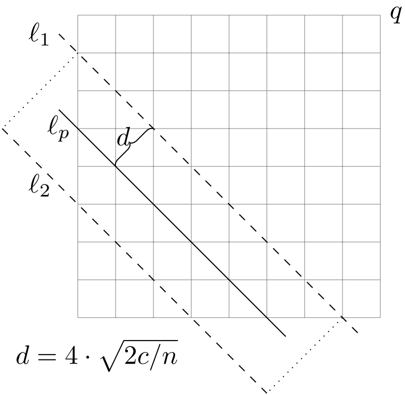
Algorithm FindMax

Analysis – Results & Techniques

- Scenario: n node c-balanced CAN, linear objective function
- Message count: $\mathcal{O}(c^{3/2} \cdot \sqrt{n})$
- Response time: $\mathcal{O}(c^{3/2} \cdot \sqrt{n})$
- Technique:
  - Line ℓ_p through optimal result p
  - Algorithm contacts non-empty areas intersecting ℓ_p
  - Upper bound number of intersecting areas using balance factor c

Algorithm FindMax

Analysis – Results & Techniques



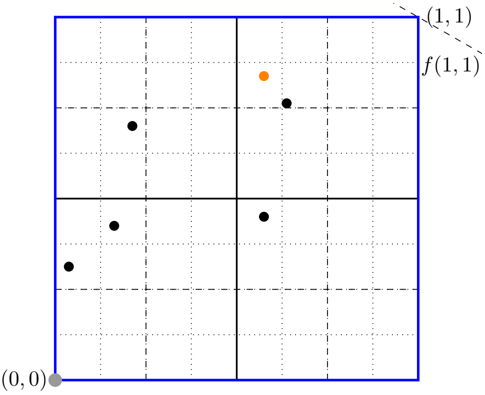
$$d = 4 \cdot \sqrt{2c/n}$$

Algorithm ParaMax

Basic Idea

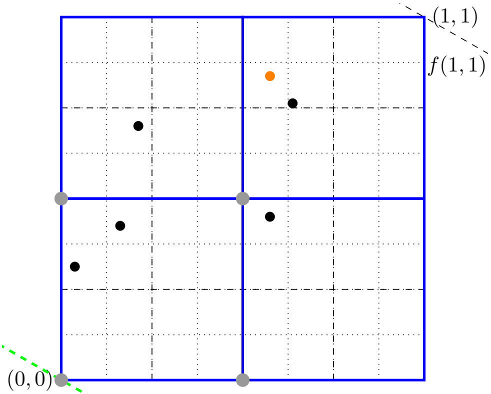
1. Lower bound function value of result
2. Multiple iterations; each time increase lower bound
3. Later iterations process areas deeper in the hierarchy
4. Process areas in parallel

Algorithm PARAMAX
Illustration



Max depth: Level 0

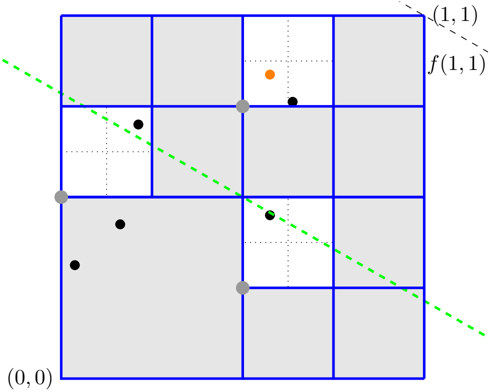
Algorithm ParaMax
Illustration



Max depth: Level 1

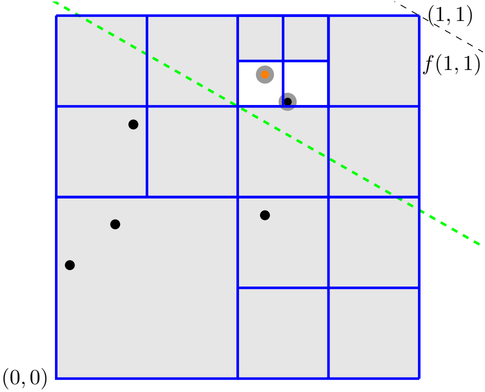
Algorithm ParaMax

Illustration



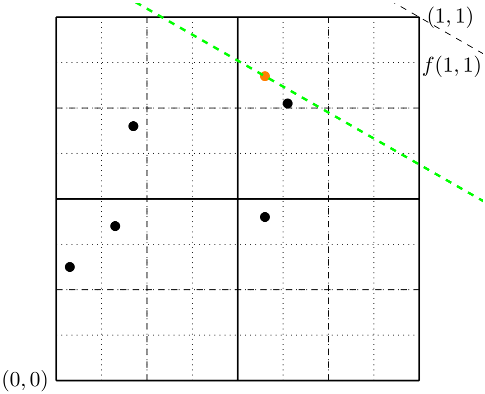
Max depth: Level 2

Algorithm ParaMax
Illustration



Max depth: Level 3

Algorithm ParaMax
Illustration



For comparison only: Optimality proven

Algorithm ParaMax

Analysis – Results & Techniques

- Scenario: n node c-balanced CAN, linear objective function
- Message count: $\mathcal{O}(\sqrt{c \cdot n})$
- Response time: $\mathcal{O}((\log c)^2 + (\log n)^2)$
- Techniques:
  - For each hierarchy level: stripes of contacted areas from that level
  - Upper bound number of areas in each stripe using balance factor c

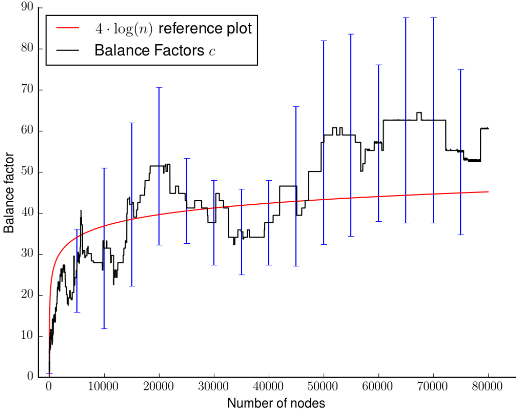
Objective Function Search in P2P Networks

Experimental Results

Experimental Results

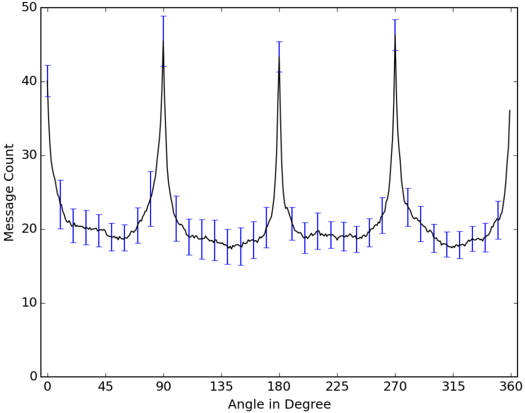
- Setting: 1000 Nodes, 100 Data Items, 1000 Requests, 25 runs
- Analyze the balance factor c
- Analyze the influence of different scaling factors:
  - Request angle
  - Number nodes

Balance Factor



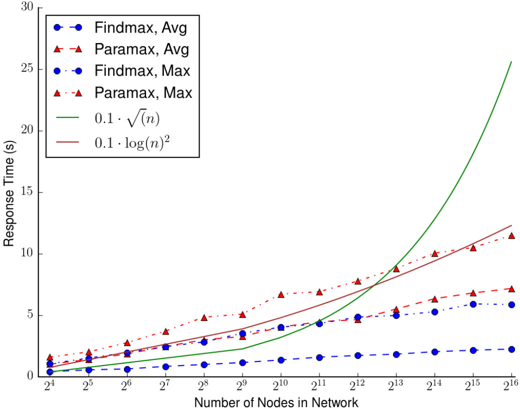
Balance factor by number nodes

Influence of Request Angle



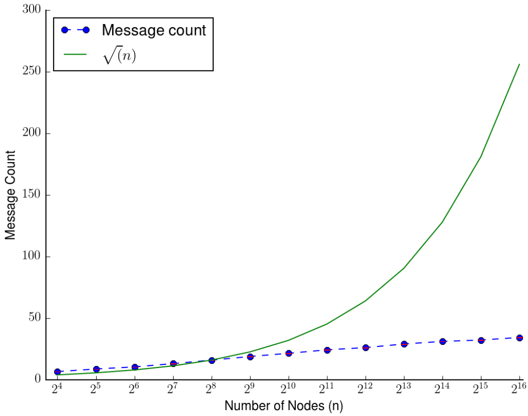
Performance of FINDMAX by request angle

Scaling of Response Time



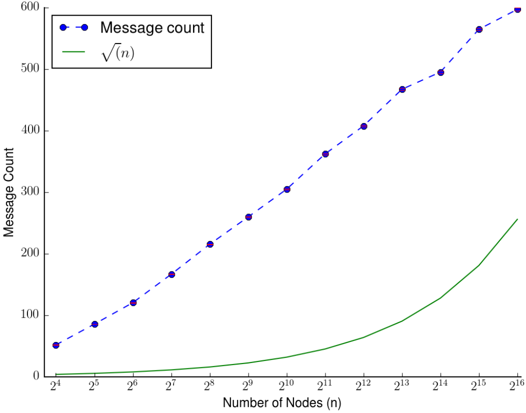
Response time by number nodes

Scaling of Message Count / Number Nodes



Message count of FINDMAX by number nodes

Scaling of Message Count / Number Nodes



Message count of PARAMAX by number nodes

Objective Function Search in P2P Networks

Further Results & Conclusion

Further Results

- Algorithms can be applied to higher dimensions
  - Bad scaling of worst cases
- Work for convex functions with minor modifications
  - Similar performance in experiments
  - No theoretical results

Conclusion

- Good first approach to function search
- Tree structure leads to balance problems
- Balance factor (of network) does not matter
- Theoretical worst cases on algorithm behavior happen in practice

Outlook

- Observe: Possible results are from convex hull of data items
- Approach: Construct meta structure managing convex hull

Network Creation Processes

Network Creation under Dynamic Communication Interests

Network Creation Games

Classical notion:

- n agents
- A strategy for every agent
- Costs for every agent depending on strategy
- Nash Equilibria

Network Creation Games

Classical notion:

- n agents
- A strategy for every agent
- Costs for every agent depending on strategy
- Nash Equilibria

Recent developement:

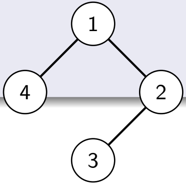
1. One-shot games with direct equilibria
2. Investigate convergence of processes
3. Our contribution: investigate sequence of processes

Network Creation Processes

Definition

A *network creation process* on a node set V consists of:

1. Initial undirected graph G_0



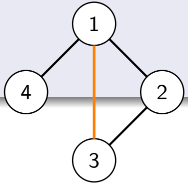
Network Creation Processes

Definition

A *network creation process* on a node set V consists of:

1. Initial undirected graph G_0
2. Set of undirected friendships F

   $F(v)$ denotes the friends of $v \in V$

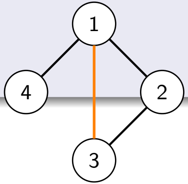


Network Creation Processes

Definition

A *network creation process* on a node set V consists of:

1. Initial undirected graph G_0
2. Set of undirected friendships F
   $F(v)$ denotes the friends of $v \in V$
3. Costs of $v \in V$ in G: $c_G(v) = \sum_{u \in F(v)} d_G(u, v)$ or $c_G(v) = \max_{u \in F(v)} d_G(u, v)$

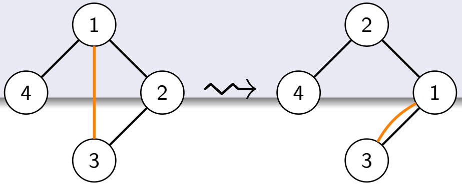


Network Creation Processes

Definition

A *network creation process* on a node set V consists of:

1. Initial undirected graph G_0
2. Set of undirected friendships F
   $F(v)$ denotes the friends of $v \in V$
3. Costs of $v \in V$ in G: $c_G(v) = \sum_{u \in F(v)} d_G(u,v)$ or $c_G(v) = \max_{u \in F(v)} d_G(u,v)$
4. Game operation: How nodes can transform the current graph, e.g., a node can swap position with one of its neighbors

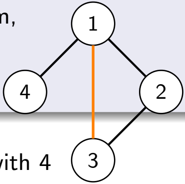


Network Creation Processes

Definition

A *network creation process* on a node set V consists of:

1. Initial undirected graph G_0
2. Set of undirected friendships F
   $F(v)$ denotes the friends of $v \in V$
3. Costs of $v \in V$ in G: $c_G(v) = \sum_{u \in F(v)} d_G(u, v)$ or $c_G(v) = \max_{u \in F(v)} d_G(u, v)$
4. Game operation: How nodes can transform the current graph, e.g., a node can swap position with one of its neighbors
5. Strategy: Which operations a node is allowed to perform, e.g., a node can perform a swap iff its costs decrease



1 can swap with 2, but not with 4

Network Creation Processes

Definition

A *network creation process* on a node set V consists of:

1. Initial undirected graph G_0
2. Set of undirected friendships F
   $F(v)$ denotes the friends of $v \in V$
3. Costs of $v \in V$ in G: $c_G(v) = \sum_{u \in F(v)} d_G(u, v)$ or $c_G(v) = \max_{u \in F(v)} d_G(u, v)$
4. Game operation: How nodes can transform the current graph, e.g., a node can swap position with one of its neighbors
5. Strategy: Which operations a node is allowed to perform, e.g., a node can perform a swap iff its costs decrease
6. Move policy: Node order to perform operations

Reachable Network Creation Processes

Idea: Communication interests can vary
$\Rightarrow$ Observe influence of simple dynamics

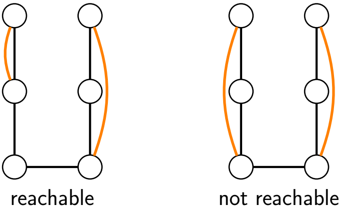
Reachable Network Creation Processes

Idea: Communication interests can vary
$\Rightarrow$ Observe influence of simple dynamics

Definition

A network creation process is *reachable* if it can be built up by

- starting with the empty friendship set, and
- adding exactly one new friendship every time a Nash equilibrium is reached.



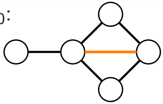
Equilibria

Definition

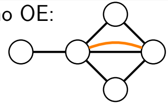
Consider a network creation process.

- A graph is a *Nash equilibrium (NE)* if no node can perform a game operation according to the strategy.
- A graph is an *operation equilibrium (OE)* if
  - it is a Nash equilibrium, and
  - it can be reached from the initial graph according to the game operation.
- A graph is a *process equilibrium (PE)* if
  - it is an operation equilibrium, and
  - it can be reached from the initial graph according to the strategy and the move policy.

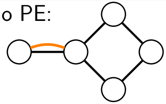
G_0:



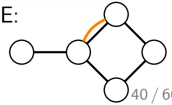
NE, no OE:



OE, no PE:



PE:



Network Creation Processes

Node Swap Processes

Convergence: SNSP

Definition

A network creation process is a *Selfish Node Swap Process (SNSP)* if

- game operation: a node can swap with one of its neighbors
- strategy: a node can perform exactly those swaps that decrease its costs

Convergence: SNSP

Definition

A network creation process is a *Selfish Node Swap Process (SNSP)* if

- game operation: a node can swap with one of its neighbors
- strategy: a node can perform exactly those swaps that decrease its costs

Theorem

For any connected graph $G = (V, E)$ with diameter ≥ 2, there is a reachable SNSP with G as initial graph and the maximum cost function for which no OE exists. This also holds for the average cost function.

Convergence: WPNSP

Definition

A network creation process is a *Weak Pairwise Node Swap Process (WPNSP)* if

- game operation: a node can swap with one of its neighbors
- strategy: a node can perform exactly those swaps that decrease its own costs and do not increase the costs of the swap partner

Convergence: WPNSP

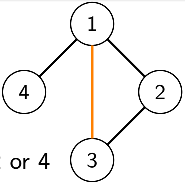
Definition

A network creation process is a *Weak Pairwise Node Swap Process (WPNSP)* if

- game operation: a node can swap with one of its neighbors
- strategy: a node can perform exactly those swaps that decrease its own costs and do not increase the costs of the swap partner

Definition

A move policy is *improving* if it always chooses one of the nodes that can perform a game operation according to the strategy.



1 or 3 be can chosen, not 2 or 4

Convergence: WPNSP – AVE

Theorem

Consider a WPNSP with initial graph G, set of friendships F, the average cost function and an improving move policy. Then it reaches a PE after at most $|F|(\mathrm{diam}(G) - 1)$ steps.

Convergence: WPNSP – AVE

Theorem

Consider a WPNSP with initial graph G, set of friendships F, the average cost function and an improving move policy. Then it reaches a PE after at most $|F|(\text{diam}(G) - 1)$ steps.

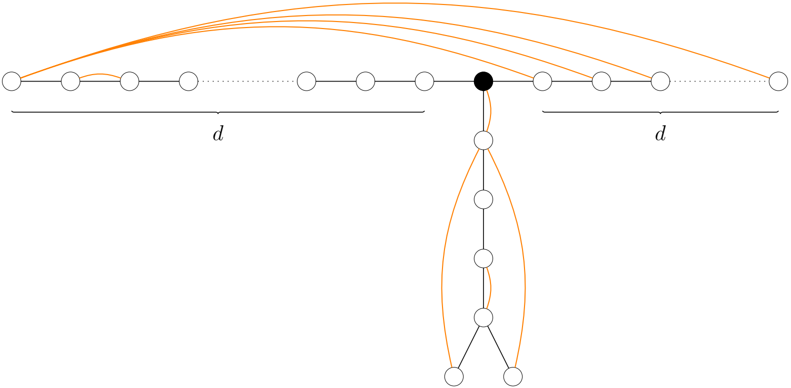
Lemma

For every $d \in \mathbb{N}, d \geq 3$, there is a reachable WPNSP with initial graph G with diameter $\Theta(d)$, $\Theta(d)$ friendships F, the average cost function and an improving move policy that reaches a PE in $\Theta(|F|(\text{diam}(G) - 1)$ steps.

Convergence: WPNSP – AVE

Lemma

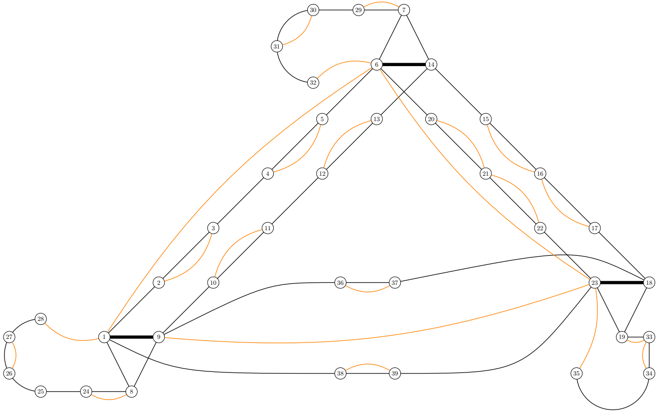
For every $d \in \mathbb{N}, d \geq 3$, there is a reachable WPNSP with initial graph G with diameter $\Theta(d)$, $\Theta(d)$ friendships F, the average cost function and an improving move policy that reaches a PE in $\Theta(|F|(\mathrm{diam}(G) - 1))$ steps.



Convergence: WPNSP – MAX

Lemma

There is a reachable WPNSP with the maximum cost function that has no PE even if its move policy is arbitrarily changed.



Convergence: SPNSP

Definition

A network creation process is a *Strong Pairwise Node Swap Process (SPNSP)* if

- game operation: a node can swap with one of its neighbors
- strategy: a node can perform exactly those swaps that decrease its own costs as well as the costs of the swap partner

Convergence: SPNSP

Definition

A network creation process is a *Strong Pairwise Node Swap Process (SPNSP)* if

- game operation: a node can swap with one of its neighbors
- strategy: a node can perform exactly those swaps that decrease its own costs as well as the costs of the swap partner

Theorem

Every SPNSP with the maximum cost function and an improving move policy reaches always a PE.

Overview

	SNSP		WPNSP		SPNSP	
	AVE	MAX	AVE	MAX	AVE	MAX
Existence of OE	no		yes	open	yes	
Existence of PE	no		yes	no	yes	
Always convergence	no		yes	no	yes	
Convergence speed	∞		$\Theta(\lvert F\rvert \operatorname{diam}(G))$	∞	$\Theta(\lvert F\rvert \operatorname{diam}(G))$	$\Omega(\lvert V_F\rvert^2 \operatorname{diam}(G))$ $O\left(\binom{\lvert V_F\rvert + \operatorname{diam}(G) - 1}{\lvert V_F\rvert}\right)$

$$V_F := \{v \in V \mid F(v) \neq \emptyset\}$$

Quality of OE

Definition

Consider a network creation process on a node set V.

Quality of OE

Definition

Consider a network creation process on a node set V.

- The *social costs* of a graph G are $\mathrm{sc}(G) := \sum_{v \in V} c_G(v)$.
- A graph G that can be reached from the initial graph according to the game operation is a *social optimum* if it has lowest social costs among all those graphs.

Quality of OE

Definition

Consider a network creation process on a node set V.

- The *social costs* of a graph G are $\mathrm{sc}(G) := \sum_{v \in V} c_G(v)$.
- A graph G that can be reached from the initial graph according to the game operation is a *social optimum* if it has lowest social costs among all those graphs.
- The *operational Price of Anarchy (oPoA)* is

$$\frac{\max\{\mathrm{sc}(G) \mid G \text{ OE }\}}{\mathrm{sc}(H)},$$

and the *operational Price of Stability (oPoS)* is

$$\frac{\min\{\mathrm{sc}(G) \mid G \text{ OE }\}}{\mathrm{sc}(H)},$$

where H is a social optimum.

Overview

	SNSP		WPNSP		SPNSP	
	AVE	**MAX**	**AVE**	**MAX**	**AVE**	**MAX**
Existence of OE	no		yes	open	yes	
Existence of PE	no		yes	no	yes	
Always convergence	no		yes	no	yes	
Convergence speed	∞		$\Theta(\lvert F\rvert \operatorname{diam}(G))$	∞	$\Theta(\lvert F\rvert \operatorname{diam}(G))$	$\Omega(\lvert V_F\rvert^2 \operatorname{diam}(G))$ $O\left(\left(\frac{\lvert V_F\rvert + \operatorname{diam}(G) - 1}{\lvert V_F\rvert}\right)\right)$
oPoA	$\Theta(\operatorname{diam}(G))$		$\Theta(\operatorname{diam}(G))$		$\Theta(\operatorname{diam}(G))$	
oPoS	$\Theta(\operatorname{diam}(G))$		1	$\Theta(\operatorname{diam}(G))$	1	$O(\operatorname{diam}(G))$

Quality of OE: oPoA for WPNSPs

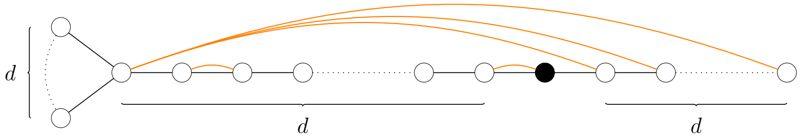
Theorem

1. *Consider a WPNSP with initial graph G, a non-empty friendship set and the maximum or average cost function that has some OE. Then, oPoA* $\leq \text{diam}(G)$.

Quality of OE: oPoA for WPNSPs

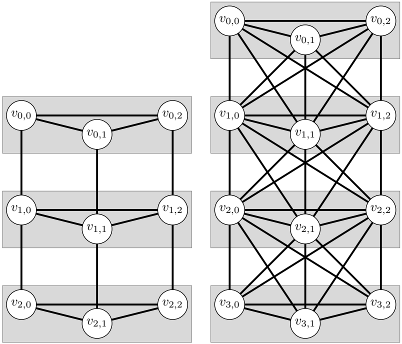
Theorem

1. *Consider a WPNSP with initial graph G, a non-empty friendship set and the maximum or average cost function that has some OE. Then, oPoA* $\leq \text{diam}(G)$.
2. *For every $d \in \mathbb{N}, d \geq 4$, there is a reachable WPNSP with the average cost function and an initial graph with diameter $\Theta(d)$ such that oPoA* $= \Theta(\text{diam}(G))$. *This also holds for the maximum cost function.*



Layered Graphs

Layers are cliques and the edges between neighboring layers build a perfect matching or a complete bipartite graph.



Overview

	SNSP		WPNSP		SPNSP	
	AVE	**MAX**	**AVE**	**MAX**	**AVE**	**MAX**
Existence of OE	no		yes	open	yes	
Existence of PE	no		yes	no	yes	
Always convergence	no		yes	no	yes	
Convergence speed	∞		$\Theta(\lvert F\rvert \operatorname{diam}(G))$	∞	$\Theta(\lvert F\rvert \operatorname{diam}(G))$	$\Omega(\lvert V_F\rvert^2 \operatorname{diam}(G))$ $O\left(\binom{\lvert V_F\rvert + \operatorname{diam}(G) - 1}{\lvert V_F\rvert}\right)$
oPoA						
General graphs	$\Theta(\operatorname{diam}(G))$		$\Theta(\operatorname{diam}(G))$		$\Theta(\operatorname{diam}(G))$	
Layered graphs	$\Theta(\operatorname{diam}(G))$	$\Theta(\sqrt{\operatorname{diam}(G)})$	———		———	
oPoS	$\Theta(\operatorname{diam}(G))$		1	$\Theta(\operatorname{diam}(G))$	1	$O(\operatorname{diam}(G))$

NP-completeness

Definition

An OE G is *optimal* if $\mathrm{sc}(G) = \min\{\mathrm{sc}(H) \mid H \text{ OE}\}$.

Theorem

The problem of finding an optimal OE for a SNSP, WPNSP or SPNSP with the maximum or the average cost function is NP-complete. This also holds for the problem of finding a social optimum.

NP-completeness

Definition

An OE G is *optimal* if $\mathrm{sc}(G) = \min\{\mathrm{sc}(H) \mid H \text{ OE}\}$.

Theorem

The problem of finding an optimal OE for a SNSP, WPNSP or SPNSP with the maximum or the average cost function is NP-complete. This also holds for the problem of finding a social optimum.

- Reduction uses clique problem.
- There is a solutions where all friendships have distance 1 iff friendship graph is isomorphic to a subgraph of the initial graph.
  $\Rightarrow$ Node swap processes are game theoretical version of subgraph problems.

Network Creation Processes

Shortcut Process

Definition

Definition

Consider a network creation process. A *shortcut* of node v is an undirected edge containing v that is not contained in the initial graph and that is owned by v.

Definition

Definition

Consider a network creation process. A *shortcut* of node v is an undirected edge containing v that is not contained in the initial graph and that is owned by v.

Definition

A network creation process is a *Shortcut Process (SCP)* if

- game operation: a node can choose a (new) shortcut
- strategy: a node can choose exactly those shortcuts that decrease its costs

Theoretical Results

Lemma

Consider an SCP, an initial graph with diameter 2 which is an OE and the maximum or average cost function. After starting a new process with an improving move policy by adding a new friendship, this process reaches a PE after at most 1 step.

Theoretical Results

Lemma

Consider an SCP, an initial graph with diameter 2 which is an OE and the maximum or average cost function. After starting a new process with an improving move policy by adding a new friendship, this process reaches a PE after at most 1 step.

Theorem

The problem of finding an optimal OE for an SCP is NP-complete for both the average and the maximum cost function.

Theoretical Results

Lemma

Consider an SCP, an initial graph with diameter 2 which is an OE and the maximum or average cost function. After starting a new process with an improving move policy by adding a new friendship, this process reaches a PE after at most 1 step.

Theorem

The problem of finding an optimal OE for an SCP is NP-complete for both the average and the maximum cost function.

Lemma

There is an algorithm that approximates social optima with factor < 2 for SCPs with the average cost function. This also holds for the maximum cost function.

Simulations

Instance: Sequence of SCPs with

- circle as initial graph,
- a new friendship every time a PE is reached,
- the empty friendship graph in the first SCP until the complete friendship graph in the last SCP,
- the strategy restricted such that every node has to perform a best move,
- cyclic move policy

Simulations

Instance: Sequence of SCPs with

- circle as initial graph,
- a new friendship every time a PE is reached,
- the empty friendship graph in the first SCP until the complete friendship graph in the last SCP,
- the strategy restricted such that every node has to perform a best move,
- cyclic move policy

Outcome: In every simulated sequence

- all processes reached a PE,
- a huge star was created,
- the social costs of every PE was at most 4 times the social costs of a social optimum.

Network Creation Processes

Open Problems

Open Problems

- Fill out Node Swap Table using only reachable examples.
- Proof analog results for SCPs.
- Find characterizations of initial graph or friendship set that imply better convergence behavior or better quality of equilibria (cf., layered graphs).
- Consider more complex dynamics of friendships (e.g., deletion).
- Examine dynamic friendships in other network creation games.