Satisfaire un internaute impatient est difficile

Fedor V. Fomin\textsuperscript{1} Frédéric Giroire\textsuperscript{2} Alain Jean-Marie\textsuperscript{3} Dorian Mazauric\textsuperscript{2,4} Nicolas Nisse\textsuperscript{2}

\textsuperscript{1} University of Bergen, Norway
\textsuperscript{2} MASCOTTE, INRIA, I3S (CNRS, UNS) Sophia Antipolis, France
\textsuperscript{3} LIRMM, MAESTRO, INRIA, Montpellier, France
\textsuperscript{4} Columbia University, New York, USA

AlgoTel, la Grande Motte, 31\textsuperscript{st} May, 2012

\textsuperscript{1} to be presented next week at FUN’12 by Fred
Prefetching for faster data access

**Web**: pre-loading web pages before the web Surfer accesses it

**Goal**: Avoid the web Surfer to wait
Prefetching for faster data access

the web Surfer starts from a given web page in cache
try to load web pages in cache \(\rightarrow\) TAKES TIME
Prefetching for faster data access

at some point, web Surfer moves
if web page reached already in the cache

www.lequipe.fr/
http://www.lequipe.fr/Football/
www.lfp.fr/ligue1/
www.uefa.com/uefachampionsleague/
www.lephoceen.fr/ www.om.net/
www.qatarambassade.com/
http://www.psgfans.fr/
www.davidbeckham.com/
fr.wikipedia.org/wiki/Qatar/

www.uefa.com/uefachamps...
Prefetching for faster data access

web Surfer follows the hyperlinks in an unpredictable way. Web pages to be loaded may be “guessed”.

Fomin, Giroire, Jean-Marie, Mazauric and Nisse

Satisfaire un internaute impatient est difficile
Prefetching for faster data access

even if we guessed well
choices might be too numerous
Prefetching for faster data access

web Surfer may access a page not in cache and has to wait !!
Prefetching for faster data access

web Surfer may access a page not in cache

not good for research...

CACHE

http://www.superp...

Fomin, Giroire, Jean-Marie, Mazauric and Nisse
Satisfaire un internaute impatient est difficile
Prefetching for faster data access

Issue: download speed, NOT size of cache
Instance: network KNOWN ((di)graph)

Related work: Probabilistic algorithms (arcs + transition probabilities)
  - Markovian model [Vitter, Krishnan. JACM’96]
    [Morad, Jean-Marie. ROADEF’10]
  - Stochastic Dynamic Programming framework
    [Joseph, Grunwald. ISCA’97]
    [Grigoras, Charvillat, Douze. ACM Multimedia’02]

Our work: minimize download speed
to insure web Surfer never waits
(worst case, deterministic)
Web = connected (di)graph
Web = (di)graph with a specified starting point (beach)
then, we need a (web) Surfer
unloaded page = dangerous node
we need someone to help the Surfer let’s call it the Guard
download speed = amount of bullets
Guard uses one bullet to secure one node
clearly, $\text{degree}(\text{beach}) \leq \#$ of bullets required to save Surfer
then, Surfer may move
here one more bullet is needed
$\text{degree}(\text{beach}) \leq \text{amount of bullets} \leq \text{max degree}$
Surfer may move anywhere in its neighborhood
bullets may be used to prevent future moves
are 3 bullets sufficient to make sure Surfer never eaten?
Model: a Two players game

- A **Surfer** starts from safe homebase $v_0$ in $G$, a dangerous graph.
- A **Guard** with some amount $k$ of bullets.

Turn by turn:
1. The guard secures $\leq k$ nodes;
2. Then, the Surfer may move to an adjacent node.

**Defeat:** Surfer in unsafe node

**Victory:** $G$ safe

Minimize amount of bullets to win for any Surfer's trajectory.

Surveillance number of $G$ (connected) from $v_0$: $sn(G, v_0)$
Model: a Two players game

- a Surfer starts from safe homebase $v_0$
  in $G$, a dangerous graph
- a Guard with some amount $k$ of bullets

Turn by turn:
1. the guard secures $\leq k$ nodes;
2. then, the Surfer may move to an adjacent node.

Defeat: Surfer in unsafe node  Victory: $G$ safe

Minimize amount of bullets to win for any Surfer’s trajectory

Surveillance number of $G$ (connected) from $v_0$: $sn(G, v_0)$
Model: a Two players game

- a **Surfer** starts from safe homebase $v_0$
- in $G$, a dangerous graph
- a **Guard** with some amount $k$ of bullets

**Turn by turn:**

1. the guard secures $\leq k$ nodes;
2. then, the Surfer may move to an adjacent node.

**Defeat:** Surfer in unsafe node  
**Victory:** $G$ safe

Minimize amount of bullets to win for any Surfer’s trajectory

Surveillance number of $G$ (**connected**) from $v_0$: $sn(G, v_0)$
with 1 bullet: after 2 steps, Surfer faces 2 dangerous nodes!!

\[ sn(G, v_0) > 1 \]
Guard uses his bullets
Guard uses (all) his bullets
Guard uses (all) his bullets, **then** Surfer may move.

Clearly: worst case if Surfer always move.
Guard uses (all) his bullets, then Surfer may move
First Example

Guard uses (all) his bullets, **then** Surfer moves
Guard uses (all) his bullets, then Surfer moves

Guard may secure any node in the graph
Guard uses (all) his bullets, **then** Surfer moves
Guard uses (all) his bullets, **then** Surfer moves

**strategy:** safe nodes + Surfer’s node $\Rightarrow \leq k$ nodes to secure
Guard uses (all) his bullets, then Surfer moves
Guard uses (all) his bullets, then Surfer moves
Guard uses (all) his bullets, then Surfer moves
Guard uses (all) his bullets, **then** Surfer moves

All nodes safe: Victory **against this trajectory of the Surfer**
First Example

In this example, all Surfer’s trajectory similar (by symmetry)
Victory whatever Surfer’s trajectory $\Rightarrow sn(G, v_0) = 2$
Results: Complexity, Algorithms and Combinatorics

Undirected

- Chordal graphs: $s_n \leq 2$ ? NP-Hard
- Split Graphs: $s_n \leq k$ ? NP-Hard
- Interval Graphs: Polynomial

Directed

- DAGs: $s_n \leq 4$ ? P-SPACE-Complete
- DAGs: $s_n \leq 2$ ? NP-Hard

General

$\text{(out)-degree (v0)} \leq s_n \leq \max\left\{ \text{degree (v0)} \right\}$

$\max\left( \text{degree (v0)} \right) \leq s_n \leq \max\left( \text{degree (v0)} \right) - 1$

$O(2^n)$ exact algorithm

Fomin, Giroire, Jean-Marie, Mazauric and Nisse

Satisfaire un internaute impatient est difficile
Positive Results

Combinatorial characterization in Trees
For any tree $T$, $v_0 \in V(T)$, $sn(T, v_0) = \max \left\lceil \frac{|N[S]| - 1}{|S|} \right\rceil$, taken for any subtree $S$ containing $v_0$.

Exact Algorithms

- $O(2^n)$ algorithm in $n$-node graphs;
- $sn(T, v_0)$ can be computed in time $O(n \log n)$ in any $n$-node tree $T$ and for any $v_0 \in V(T)$;
- $sn(G, v_0)$ can be computed in time $O(n \cdot \Delta^3)$ in any $n$-node interval graph $G$ with maximum degree $\Delta$ and for any $v_0 \in V(T)$. 
Further Work: Connected Variant

**Constraint:** safe vertices must induce a connected subgraph
Further Work: Connected Variant

**Constraint:** safe vertices must induce a connected subgraph
Further Work: Connected Variant

**Constraint:** safe vertices must induce a connected subgraph
Further Work: Connected Variant

**Constraint:** safe vertices must induce a connected subgraph

**Connectivity costs:**

\[
\text{connected-} sn(G, v_0) = 3 > sn(G, v_0) = 2
\]

All previous results hold for the connected variant

Fomin, Giroire, Jean-Marie, Mazauric and Nisse

Satisfaire un internaute impatient est difficile
Further Work: Connected Variant

**Constraint:** safe vertices must induce a connected subgraph

**Connectivity costs:**
\[ \text{connected-} sn(G, v_0) = 3 = sn(G, v_0) + 1 \]

\[ \exists G \text{ and } v_0 \text{ such that } c-sn(G, v_0) \geq sn(G, v_0) + 2 \]
Open Questions

- complexity in bounded degree graphs? (polynomial if $\Delta \leq 3$)
- complexity in bounded treewidth graphs?
- $\exists c < 2$ and $O(c^n)$ algorithm in $n$-node graphs?
- cost of connectivity? $\frac{\text{connected}}{\text{sn}} \leq \text{cte}$?
- $\exists G$ and $v_0$ such that $c \cdot \text{sn}(G, v_0) \geq \text{sn}(G, v_0) + 2$ ????
- online variant?
- ...
Thank you for your attention\textsuperscript{2}

\textsuperscript{2}No seafood... no animal has been hurt when preparing this talk.