Design Patterns in Beeping Algorithms

A. Casteigts, Y. Métivier, J.M. Robson, A. Zemmari

LaBRI - Université de Bordeaux, CNRS, FRANCE

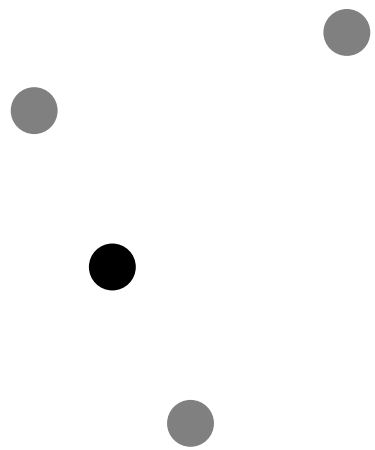
OPODIS 2016
Dec. 14, 2016

Outline

Outline

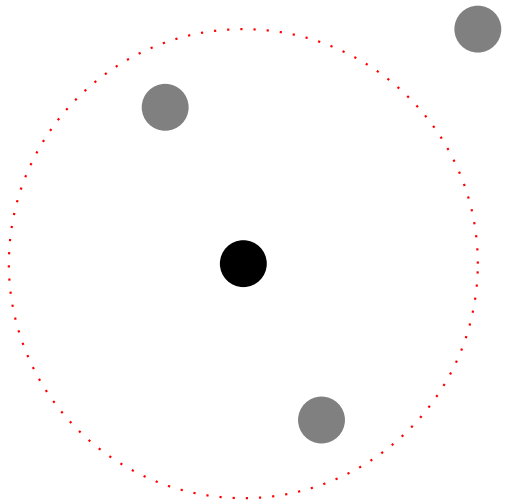
Beeping Models

Nodes interact with beeps.



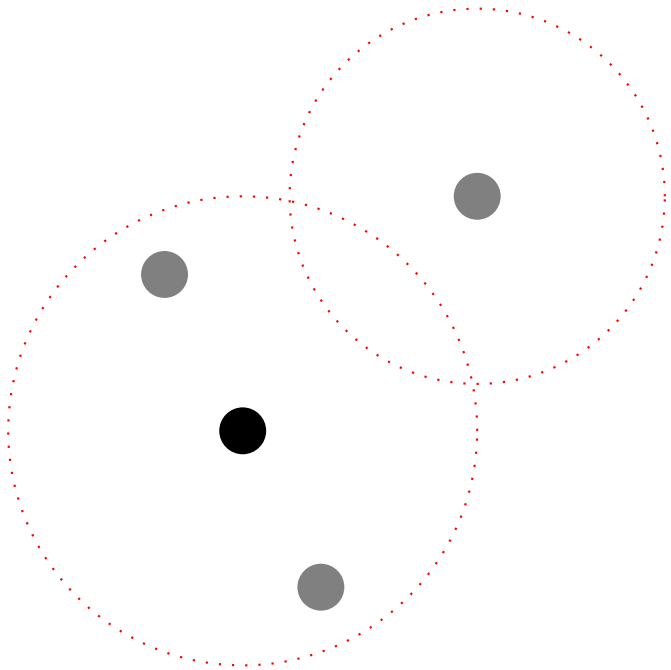
Beeping Models

Nodes interact with beeps.



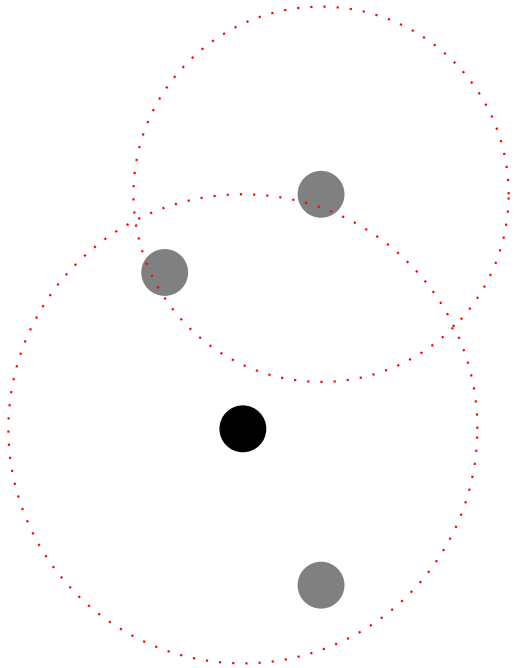
Beeping Models

Nodes interact with beeps.



Beeping Models

Nodes interact with beeps.



Beeping Models

If a process beeps, there are two cases:

- B: it cannot know whether another process beeps simultaneously.
- B_{cd}: it can distinguish whether it beeped alone or if at least one neighbour beeped concurrently (internal collision).

If a process listens, there are two cases:

- L: it can distinguish between silence and the presence of at least one beep.
- L_{cd}: it can distinguish between silence or the presence of one beep or the presence of at least two beeps (peripheral collision).

Beeping Models

If a process beeps, there are two cases:

- B: it cannot know whether another process beeps simultaneously.
- B_{cd}: it can distinguish whether it beeped alone or if at least one neighbour beeped concurrently (internal collision).

If a process listens, there are two cases:

- L: it can distinguish between silence and the presence of at least one beep.
- L_{cd}: it can distinguish between silence or the presence of one beep or the presence of at least two beeps (peripheral collision).

Beeping Models

If a process beeps, there are two cases:

- B: it cannot know whether another process beeps simultaneously.
- B_{cd}: it can distinguish whether it beeped alone or if at least one neighbour beeped concurrently (internal collision).

If a process listens, there are two cases:

- L: it can distinguish between silence and the presence of at least one beep.
- L_{cd}: it can distinguish between silence or the presence of one beep or the presence of at least two beeps (peripheral collision).

Beeping Models

If a process beeps, there are two cases:

- B: it cannot know whether another process beeps simultaneously.
- B_{cd}: it can distinguish whether it beeped alone or if at least one neighbour beeped concurrently (internal collision).

If a process listens, there are two cases:

- L: it can distinguish between silence and the presence of at least one beep.
- L_{cd}: it can distinguish between silence or the presence of one beep or the presence of at least two beeps (peripheral collision).

Beeping Models

If a process beeps, there are two cases:

- B: it cannot know whether another process beeps simultaneously.
- B_{cd}: it can distinguish whether it beeped alone or if at least one neighbour beeped concurrently (internal collision).

If a process listens, there are two cases:

- L: it can distinguish between silence and the presence of at least one beep.
- L_{cd}: it can distinguish between silence or the presence of one beep or the presence of at least two beeps (peripheral collision).

Beeping Models

If a process beeps, there are two cases:

- B: it cannot know whether another process beeps simultaneously.
- B_{cd}: it can distinguish whether it beeped alone or if at least one neighbour beeped concurrently (internal collision).

If a process listens, there are two cases:

- L: it can distinguish between silence and the presence of at least one beep.
- L_{cd}: it can distinguish between silence or the presence of one beep or the presence of at least two beeps (peripheral collision).

Beeping Models

If a process beeps, there are two cases:

- B: it cannot know whether another process beeps simultaneously.
- B_{cd}: it can distinguish whether it beeped alone or if at least one neighbour beeped concurrently (internal collision).

If a process listens, there are two cases:

- L: it can distinguish between silence and the presence of at least one beep.
- L_{cd}: it can distinguish between silence or the presence of one beep or the presence of at least two beeps (peripheral collision).

Beeping Models

If a process beeps, there are two cases:

- B: it cannot know whether another process beeps simultaneously.
- B_{cd}: it can distinguish whether it beeped alone or if at least one neighbour beeped concurrently (internal collision).

If a process listens, there are two cases:

- L: it can distinguish between silence and the presence of at least one beep.
- L_{cd}: it can distinguish between silence or the presence of one beep or the presence of at least two beeps (peripheral collision).

Beeping Models

$\Rightarrow$ Different beeping models:

- BL: Cornejo and Kuhn, *Deploying wireless networks with beeps*. DISC'2010.
- $B_{cd}L$ and BL_{cd}: Afek et al. *Beeping a maximal independent set.* Distributed Computing 2013.
- $B_{cd}L$: Jeavons, Scott, and Xu, *Feedback from nature: an optimal distributed algorithm for maximal independent set selection.* PODC'2013.
- BL, $B_{cd}L$ and $B_{cd}L_{cd}$: in this talk.

Outline

Our Contribution

- Generic building blocks (design patterns) which seem to occur often in the design of beeping algorithms.
- Algorithms for various graph problems which improve upon previous solutions (or their analysis does).
- Emulation techniques for using collision detection if it is not available.

Outline

Design patterns for beeping algorithms

Exclusive beeps (using B_{cd}).

repeat
    beep with some probability;
    if *I beeped alone* **then**
        `do something exclusive;`
    ...
until *finished*;

Remark. Nearly all algorithms rely on this pattern.

2-hop exclusive beeps (using $B_{cd}L_{cd}$).

repeat
  switch *slot* **do**
    slot 1 // contending
      beep with some probability;
    slot 2 // detection of peripheral collision
      if *several neighbours beeped in slot 1* **then**
        beep;
    after slot 2
      if *I beeped alone in slot 1 and no neighbour beeped in slot 2* **then**
        do something 2-hop exclusive
  ...
until *finished*;

Multi-slot phases

See e.g. above.

Adaptive beeping probability (using $B_{cd}L_{cd}$).

Float $p \leftarrow 1/2$ // say
repeat
  beep with probability p;
  if *I beeped alone* **then**
    do something exclusive;
  else
    if *no one beeped* **then**
      increase p;
    else
      decrease p;
until *finished*;

Emulation of Collision Detection

Detailed later.

Outline

Colouring

A Las Vegas colouring algorithm in $B_{cd}L$ (without knowledge).

Float $p \leftarrow 1/2$
Integer colour $\leftarrow$ *nil*;
Integer counter $\leftarrow 0$;
repeat
  beep with probability p;
  if *I beeped alone* **then**
    colour $\leftarrow$ *counter* ;
  else
    if *no one beeped* **then**
      increase p;
    else
      decrease p;
    counter $\leftarrow$ *counter* $+ 1$;
until *finished*;

Colouring

A Las Vegas colouring algorithm in $B_{cd}L$ (without knowledge).

Float $p \leftarrow 1/2$
Integer colour $\leftarrow$ *nil*;
Integer counter $\leftarrow 0$;
repeat
  beep with probability p;
  if *I beeped alone* **then**
    colour $\leftarrow$ *counter* ;
  else
    if *no one beeped* **then**
      increase p;
    else
      decrease p;
    counter $\leftarrow$ *counter* $+ 1$;
until *finished*;

Colouring

Theorem

There are constants α, β and γ such that for any graph $G = (V, E)$ of n vertices and maximum degree Δ, the number of phases to colour all the nodes in G is:

1. *less than $\alpha(\Delta + \log n)$ with probability $1 - o\left(n^{-1}\right)$,*
2. *less than $\beta(\Delta + \log n)$ on average,*
3. *less than $\gamma(\Delta + \log n)$ with probability $1 - o\left(n^{-c}\right)$, for any $c > 1$.*

Computing in the Beeping Models

Problem	Design Pattern	Time Complexity (w.h.p)
$(K+1)$-colouring	Adaptive Probability	$O\left(K(\log n + \log^2 K)\right)$
2-hop colouring (without knowledge)	2-hop exclusive beeps	$O\left(\log n + \Delta^2\right)$
2-hop colouring (with a bound K on Δ)	2-hop exclusive beeps	$O\left(K^2(\log n + \log^2 K)\right)$
Degree Computation	2-hop exclusive beeps	$O\left(\log n + \Delta^2\right)$
MIS	Adaptive Probability	$O\left(\log n\right)$
2-hop MIS	2-hop exclusive beeps	$O\left(\log n\right)$

Computing in the Beeping Models

Problem	Design Pattern	Time Complexity (w.h.p)
$(K+1)$-colouring	Adaptive Probability	$O\left(K(\log n + \log^2 K)\right)$
2-hop colouring (without knowledge)	2-hop exclusive beeps	$O\left(\log n + \Delta^2\right)$
2-hop colouring (with a bound K on Δ)	2-hop exclusive beeps	$O\left(K^2(\log n + \log^2 K)\right)$
Degree Computation	2-hop exclusive beeps	$O\left(\log n + \Delta^2\right)$
MIS	Adaptive Probability	$O\left(\log n\right)$[1]
2-hop MIS	2-hop exclusive beeps	$O\left(\log n\right)$

[1] the time complexity is $76 \log n$ which compares well with the $e^{25} \log n$ proved by Jeavons, Scott, and Xu

Collision Detection

Lemma.

No Las Vegas collision detection algorithm in BL, $B_{cd}L$ and BL_{cd} models.

Collision Detection

A Monte Carlo Collision Detection Algorithm in BL

We define a phase P by the sequence of the 3 following actions:

- vertices wishing to beep, randomly and uniformly select 0 or 1;
- slot 1: vertices that have drawn 0 beep, the others listen;
- slot 2: vertices that have drawn 1 beep, the others listen.

Each vertex executes k phases.
$\Rightarrow$ A vertex detects a collision if:

- it does not beep and it hears beeps at two slots in a phase,
- or if it beeps itself at a slot of a phase and hears a beep at the other slot of the same phase.

Collision Detection

A Monte Carlo Collision Detection Algorithm in BL

We define a phase P by the sequence of the 3 following actions:

- vertices wishing to beep, randomly and uniformly select 0 or 1;
- slot 1: vertices that have drawn 0 beep, the others listen;
- slot 2: vertices that have drawn 1 beep, the others listen.

Each vertex executes k phases.
$\Rightarrow$ A vertex detects a collision if:

- it does not beep and it hears beeps at two slots in a phase,
- or if it beeps itself at a slot of a phase and hears a beep at the other slot of the same phase.

Collision Detection

The Algorithm

Var:

k : **Global integer constant;**

...

collision := *false*;

for $i := 1, k$

if v wishes to beep **then**

Choose b uniformly at random from $\{0, 1\}$;

if $b = 0$ **then** beep; listen **else** listen; beep;

if a beep was heard **then** *collision* := *true*

Else

listen; listen;

if two beeps were heard **then** *collision* := *true*;

Collision Detection

Lemma.

Let G be a graph having n vertices. Any collision in G is detected after at most $O\left(\ln(\frac{n}{\epsilon})\right)$ phases (slots) with probability at least $1-\epsilon$, and after at most $O(\ln n)$ phases (slots) with probability $1-o\left(\frac{1}{n}\right)$.

Theorem.

The complexity of a Monte Carlo algorithm which detects collision with high probability in the BL model is $\Omega(\log n)$.

Collision Detection

Lemma.

Let G be a graph having n vertices. Any collision in G is detected after at most $O\left(\ln(\frac{n}{\epsilon})\right)$ phases (slots) with probability at least $1-\epsilon$, and after at most $O(\ln n)$ phases (slots) with probability $1-o\left(\frac{1}{n}\right)$.

Theorem.

The complexity of a Monte Carlo algorithm which detects collision with high probability in the BL model is $\Omega(\log n)$.

Outline

Emulating Strong Models

The idea: devise Monte Carlo procedures to emulate B_{cd} and L_{cd} in the BL model.
$\Rightarrow$ This allows us to design Monte Carlo algorithms (for MIS and Colouring for example) in the BL model.

Colouring in *BL*

Combining the CD procedure with the colouring algorithm in $B_{cd}L$ yields:

Corollary

Assuming the *BL* model, there exists a colouring algorithm $\mathcal{C}''$ s.t. for any graph G of size n and any $0 < \varepsilon < 1$:

- $\mathcal{C}''$ computes a colouring for G in $O\left(K\left(\log n + \log^2 K\right)\log\left(\frac{n}{\varepsilon}\right)\right)$, and the result is correct with probability at least $1-\varepsilon$.
- It computes a colouring for any vertex v in $O\left(K\left(\log n + \log^2 K\right)\log\left(\frac{1}{\varepsilon}\right)\right)$, and the result is correct with probability at least $1-\varepsilon$.
- It computes a colouring of G in $O\left(K\left(\log n + \log^2 K\right)\log n\right)$, and the result is correct with probability at least $1 - o\left(\frac{1}{n}\right)$.

Outline

Conclusion and Future Works

- We considered classical problems in the beeping model: CD, Degrees, MIS, Colouring, ...
- Can we do the same for other problems (e.g. Election) ?
- What about lower bounds ?

Thank you !