Self-stabilizing Minimal Dominating Set Algorithms of Distributed Systems

& the Signed Star Domination Number of Cayley Graphs

Well Yu-Chieh Chiu (邱鈺傑)

Department of Applied Mathematics
National Chiao Tung University
Hsinchu, Taiwan

August 3, 2014

Outline

Basic graph definitions

- Distributed systems $\rightharpoonup$ graphs
  - $G = (V, E)$ is simple, undirected, finite, and connected.
  - processes $\rightharpoonup V$, $n = |V|$
  - interconnections $\rightharpoonup E$, $m = |E|$
- Notations:
  - $u \sim v$: u and v are adjacent.
  - $\deg(v)$: degree of v.
  - $N(v)$, $N[v]$: (closed) neighborhood of v.
  - $d(u, v)$: distance from u to v.
  - $G[S]$: subgraph of G induced by S.
  - u is a *k-neighbor* of v if $d(u, v) = k$.
  - $N_k(v)$: k-neighborhood of v.

Domination & Independence

- DS: dominating set.
- MDS: minimal dominating set.
- CDS: connected dominating set.
- IS: independent set.
- MIS: maximal independent set.
- (well known) An MIS of G is an MDS of G.

MDS prob

For a distributed system G, the *minimal dominating set problem* (MDS prob) is

- For each node $v \in V$, its neighbors are given as input.
- Each node $v \in V$ decides its decision d_v as output.
- The set $S = \{v \in V : d_v = 1\}$ must be an MDS.

Applications

Messages routing in wireless sensor networks (WSNs)

multihop delivering

$\uparrow$

virtual backbone in WSNs

$\uparrow$

CDS

$\uparrow$

MDS

Distributed algo

A distributed algo is:

- Executed independently on each node.
- Defined by a collection of rules.
  - Rule: **if** precondition **then** action;
  - A rule is enabled if precondition evaluates true.
  - A node is privileged if one of rules is enabled.
- Execution of a rule (taking action) is called a move.

Example

Two simple rules that maintain an MIS I:

R1: **if** $v \notin I$ having no neighbor in I **then** v joins I;

R2: **if** $v \in I$ having a neighbor in I **then** v leaves I;

Definition of Self-stabilization (SS)

Definition

An algo on a distributed system is called *self-stabilizing* (SS) if the following two properties hold:

- Starting from any illegitimate state, the distributed system reaches a legitimate state in a finite time without any external intervention.
- After convergence, the system remains in the set of legitimate state.

Example

The previous example (2 simple rules) is SS w.r.t. MIS.

SS models

Execution models (daemons, schedulers):

- *Central*: choose only one
- *Synchronous*: choose all
- *Distributed*: choose arbitrary subset (practical)

Deterministic uniform SS algo assuming synchronous or *distributed daemons* are impossible for an *anonymous system* because the difficulty encountered in **symmetry breaking**.

✓

Our algo. use an distributed daemon.
We assume each node v has $v.id$.

SS graph algo

For finding a vertex subset (such as MIS/MDS prob), we define:

- The states of nodes are categorized into two types:
  - `IN`: If the node is considered in the set S with the desired property
  - `OUT`: If the node is considered not in S with the desired property
- Membership move: A move s.t. the state of the node changes from `IN` to `OUT` or from `OUT` to `IN` after the execution.

✓

We use terms: `IN` nodes, `OUT` nodes, `IN` neighbors, etc.

Previous SS MDS algo

	stabilization time	daemon type
Hedetniemi et al. (2003)	$(2n+1)n$ moves	central
Xu et al. (2003)	$4n$ rounds	synchronous
Turau (2007)	$9n$ moves	distributed
Goddard et al. (2008)	$5n$ moves	distributed

Main result

We present `Well4n`, a $(4n-2)$-move SS MDS algo.

Well4n

In Well4n every node has state in 4 different values:

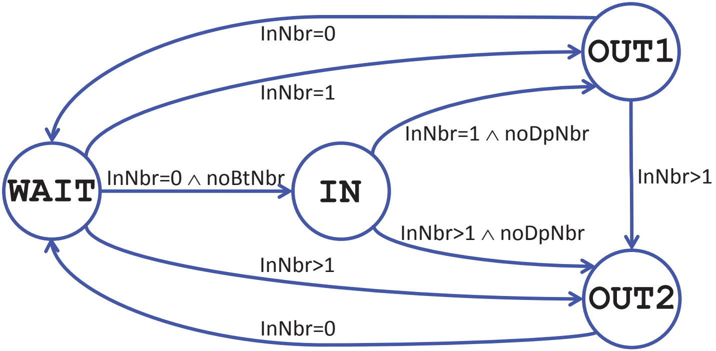
- IN (in the set),
- OUT1 (not in the MDS, having a unique IN neighbor),
- OUT2 (not in the MDS, having ≥ 1 IN neighbors),
- WAIT (not in the MDS, having no IN neighbor).

Definition

In Well4n a configuration is legitimate if the set of IN nodes form an MDS and for every OUT node, the state and the number of IN neighbors are consistent.

Notice that a legitimate configuration will not contain any WAIT node.

State diagram of `Well4n`



Correctness & convergence

Theorem

The proposed algorithm `Well4n` *is self-stabilizing under an unfair distributed daemon and it stabilizes after at most* $4n-2$ *moves with a minimal dominating set, where n is the number of nodes. Moreover, the bound* $4n-2$ *is tight.*

Main result

- MDS-silent: [Turau, 2007] initialized with an MDS does not make any moves.
- MDS-covered: [Hedetniemi et al., 2003] can potentially produce any MDS.
- MDS-stable: (this thesis) not make any membership move when the system is initialized with an MDS.

Level	Feature	SS Algo
S_1-stable	MDS	`Turau3n`
S_2-stable	MDS-covered	`Turau9n`, `Goddard5n`, `Well4n`
S_3-stable	MDS-stable	?
S_4-stable	MDS-silent	none

MDS-silent: non-existence

Theorem

There exists no SS MDS-silent algo in the normal model.

Corollary

A SS MDS-covered algo in the normal model cannot have only one Boolean variable as its state variable.

Conjecture

There exists no SS MDS-stable algo in the normal model.

Four levels of stableness

- MDS-covered $\Rightarrow$ MDS: by definition.
- MDS-stable $\Rightarrow$ MDS-covered: initialized with any MDS, then system stabilizes with the same MDS.
- MDS-silent $\Rightarrow$ MDS-stable: membership moves are moves. □

Theorem

S_{i+1}-stableness implies S_i-stableness, for $i \in \{1, 2, 3\}$.

Four levels of stableness

Stableness of SS DS algo

Reference	Output	Stableness	Required topology	Anonymous	Daemon	Model	Complexity
Hedetniemi et al. 2003-1	DS	S_1	Arbitrary	Yes	Central	Normal	$\leq n-1$ moves
Hedetniemi et al. 2003-2	MDS	S_2	Arbitrary	Yes	Central	Normal	$\leq (2n+1)n$ moves
Xu et al. 2003	MDS	S_2	Arbitrary	No	Synchronous	Normal	$\leq 4n$ rounds
Goddard et al. 2003,2005	MTDS	S_2	Arbitrary	No	Central	Normal	$\leq 2^n$ moves
Turau 2007	MDS	S_2	Arbitrary	No	Distributed	Normal	$\leq 9n$ moves
Goddard et al. 2008	MDS	S_2	Arbitrary	No	Distributed	Normal	$\leq 5n$ moves
Kamei et al. 2003-1	MkDS	S_1	Tree	Yes	Central	Normal	$\leq n^2$ moves
Kamei et al. 2003-2	MkDS	S_1	Tree	No	Distributed	Normal	$\leq 3(n-1)^2$ moves
Huang et al. 2008	M2DS	S_1	Arbitrary	Yes	Central	Normal	$\leq 10n+1$ moves
Kamei et al. 2005	MkDS	S_1	Arbitrary	No	Synchronous	Normal	$\leq n$ rounds
Huang et al. 2007	M2DS	S_1	Arbitrary	No	Distributed	Normal	
Gairing et al. 2004	MDS	S_4	Arbitrary	Yes	Central	Dist-2	$\leq 2n$ moves
Turau et al. 2012	MkDS	S_4	Arbitrary	Yes	Central	Dist-2	$\leq 2n$ moves
Belhoul et al. 2014	MTDS	S_4	Arbitrary	Yes	Central	Dist-2	$\leq 2n$ moves
Sec.3.2 in this thesis	MDS	S_2	Arbitrary	No	Distributed	Normal	$\leq 4n-2$ moves
Sec.5.3 in this thesis	MDS	S_4	Arbitrary	No	Distributed	Dist-2	$\leq 2n-1$ moves

Distance-2 model

Definition

A distributed system is in the *distance-2 model* if a node can instantaneously see the states of all nodes within distance two from it.

In the literatures, two *transformers* from distance-2 model to normal model are given.

Reference	Slowdown factor	Memory overhead
[Gairing et al. 2004]	$\times O(n^2 m)$	$\times \Omega(\Delta \log n)$
[Turau 2012]	$\times O(m)$	$+O(\log n)$

The transformed algorithms are not MDS-silent.

Implementing the distance-2 model

We implement the distance-2 model on WSNs by using the *beacon messages*.

Each sensor node:

1. Broadcast beacon message periodically.
2. Construct/update *neighbor list*.
3. Broadcast beacon message with neighbor list.
4. After receiving neighbor lists from all neighboring nodes.
   - Construct/update 2-*neighbor list* by received neighbor lists.
   - If one rule is enabled, take action.
   - Go back to step 1.

Correctness and convergence

Rules of `Well2n`

S1: if $v.state = \texttt{OUT} \wedge noInNbr(v) \wedge noBtNbr_2(v)$
then $v.state := \texttt{IN}$; // enter S

S2: if $v.state = \texttt{IN} \wedge \neg noInNbr(v) \wedge noDpNbr_2(v)$
then $v.state := \texttt{OUT}$; // leave S

Theorem

Algorithm `Well2n` is self-stabilizing and MDS-silent under an unfair distributed daemon in the distance-2 model. It stabilizes after at most $2n-1$ moves with a minimal dominating set, where n is the number of nodes.

The signed star domination: definitions

- A function $f : E(G) \to \{-1, 1\}$ is called a *signed star dominating function* (SSDF) on G if $\sum_{e \in E_G(v)} f(e) \geq 1$ for every $v \in V(G)$, where $E_G(v)$ is the set of all edges incident to v.
- The *signed star domination number* of G is defined as $\gamma_{SS}(G) = \min\{\sum_{e \in E(G)} f(e) | \ f$ is an SSDF on $G\}$.
- [Xu 2005]
  - Initiated the study of γ_{SS} of graphs
  - $\lceil \frac{n}{2} \rceil \leq \gamma_{SS}(G) \leq 2n - 4$
- [Atapour et al. 2010]
  - $\gamma_{SS}(K_n), \forall n$
    - 1-factorable
    - H-factorable

Cayley graphs

- $Cay(\Gamma, \Omega)$: Cayley graph
  - Γ: a finite nontrivial group
  - Ω: a symmetric generating subset of nonidentity elements of Γ
- arrange the elements of Ω in the form: $\{\sigma_1, \sigma_2, \ldots, \sigma_{2s}, \tau_1, \tau_2, \ldots, \tau_t\}$ satisfying $\sigma_i^{-1} = \sigma_{i+s},\ 1 \leq i \leq s$ and $\tau_j^{-1} = \tau_j,\ 1 \leq j \leq t$.
  - s 2-factors
  - t 1-factors

$$E(G) = \left(\bigcup_{i=1}^{s} \mathcal{C}_i\right) \cup \left(\bigcup_{j=1}^{t} \mathcal{M}_j\right).$$

Signed star domination of the Cayley graphs

Lemma

Let G be an r-regular graph.
Then we have

$$\gamma_{ss}(G) \geq \begin{cases} \frac{n}{2} & \text{if } r \text{ is odd,} \\ n & \text{if } r \text{ is even,} \\ n+1 & \text{if } r \text{ is a multiple of } 4 \text{ and } n \text{ is odd.} \end{cases}$$

Theorem

Let Γ be a finite group of even order n, and let Ω be a symmetric generating set of Γ. If $|\Omega|$ is odd, then

$$\gamma_{ss}(Cay(\Gamma, \Omega)) = \frac{n}{2}.$$

Signed star domination of the Cayley graphs

Theorem

Let Γ be a finite group of order n, and let Ω be a symmetric generating set of Γ with even $|\Omega|$. If Ω has an odd number of distinct $\{\sigma, \sigma^{-1}\}$ pairs, or contains an element of even order, then

$$\gamma_{SS}(Cay(\Gamma, \Omega)) = n.$$

Theorem

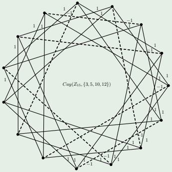
Let Γ be a finite cyclic group of odd order n. Assume that Ω is a symmetric subset containing a generator of Γ. If $|\Omega|$ is a multiple of 4, then

$$\gamma_{SS}(Cay(\Gamma, \Omega)) = n + 1.$$

Signed star domination of the Cayley graphs

Example

$\mathbb{Z}_{15} = \{0, 1, \ldots, 14\}$,
$\Omega = \{3, 5, 10, 12\}$,
$\gamma_{SS}(Cay(\mathbb{Z}_{15}, \Omega)) = 16$



Conjecture

Let Γ be a finite group of order n, and let $\Omega = \{\sigma_1, \sigma_2, \ldots, \sigma_{2s}\}$ be a symmetric generating set of Γ. If s is even and no element of Ω is of even order, then $\gamma_{SS}(Cay(\Gamma, \Omega)) = \begin{cases} n & \text{if } n \text{ is even,} \\ n+1 & \text{if } n \text{ is odd.} \end{cases}$

Summary

- Main results
  - Provide a $(4n-2)$-move SS MDS-covered algo in normal model; bound is tight
  - Prove an MDS-silent algo in normal model is impossible
  - Develop an SS MDS-silent algo in dist-2 model
  - Give partial results on γ_{SS} of Cayley graphs
- Future works
  - Design an SS MDS-covered algo in normal model using fewer than $4n-2$ moves.
  - Prove an MDS-stable algo in normal model is impossible
  - Multi-self-stabilization
  - Complete the results on γ_{SS} of Cayley graphs

謝謝！

Thank you for your attention!