A Warning Propagation-Based Linear-Time-and-Space Algorithm for the Minimum Vertex Cover Problem on Giant Graphs

Hong Xu Kexuan Sun Sven Koenig T. K. Satish Kumar
{hongx, kexuansu, skoenig}@usc.edu tkskwork@gmail.com

January 3, 2018

University of Southern California, Los Angeles, California 90089, the United States of America

The 15th International Symposium on Artificial Intelligence and Mathematics (ISAIM 2018)
Fort Lauderdale, Florida, the United States of America

Summary

- The minimum vertex cover (MVC) problem is a classical computer science problem.
- Local search algorithms often require a good starting state.
- For giant graphs, it is desirable to develop a linear-time-and-space algorithm to find a vertex cover as small as possible.
- We developed MVC-WP, a family of warning propagation-based linear-time-and-space algorithms.

Agenda

- Motivation
- MVC-WP: Warning Propagation-Based Linear-Time-and-Space Algorithms
- Experimental Evaluation
- Conclusion and Future Work

Agenda

- Motivation
- MVC-WP: Warning Propagation-Based Linear-Time-and-Space Algorithms
- Experimental Evaluation
- Conclusion and Future Work

Motivation: The Minimum Vertex Cover (MVC) Problem

- The minimum vertex cover (MVC) problem is a classical computer science problem.
  - On a graph $G = \langle V, E \rangle$, a vertex cover (VC) is a subset of vertices $S \subseteq V$ such that every edge in G has at least one endpoint vertex in S.
  - The MVC problem is to find a VC of minimum cardinality.
- Applications:
  - Computer network security (Filiol et al. 2007)
  - Crew scheduling (Sherali et al. 1984)
  - Construction of phylogenetic tree (Abu-Khzam et al. 2004)

Motivation: A Linear Algorithm for the MVC Problem

For giant graphs:

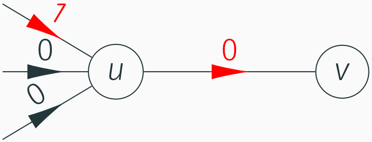
- Exact algorithms in general do not scale well.
- Local search algorithms require a small VC as a starting state (Andrade et al. 2012; Cai 2015; Cai et al. 2013; Pullan 2009).
- A linear-time-and-space algorithm to construct a small VC is desirable.

Agenda

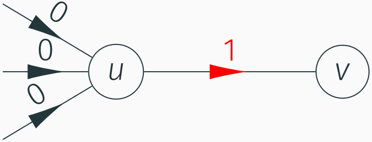
- Motivation
- MVC-WP: Warning Propagation-Based Linear-Time-and-Space Algorithms
- Experimental Evaluation
- Conclusion and Future Work

Warning Propagation (WP) for the MVC Problem

An iterative algorithm proposed and has only been used for theoretical analysis of properties of MVC on infinite graphs by (Weigt et al. 2006).



(a) u sends a message of 0 to v since one of its incoming messages from other vertices is 1.



(b) u sends a message of 1 to v since all other incoming messages are 0.

There is a message of either 0 or 1 along each direction of each edge. Assume that u and v are two adjacent vertices. u sends v a message of 1 to "warn" v to indicate that u will not be selected in the vertex cover. Otherwise, u sends v a message of 0.

Basic Idea

- Observation: Warning propagation (WP) does not work well in practice (slow to converge, output often not good). But, each iteration uses only linear amount of time and space.
- Basic idea of a linear-time-and-space algorithm: Run a few iterations of warning propagation and extract a vertex cover afterwards.
- However, can we improve the output by having some processing before and after warning propagation iterations? Yes.

Major Steps of MVC-WP

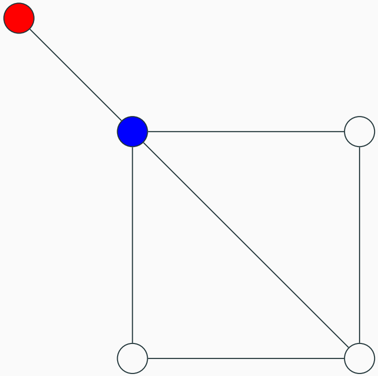
- MVC-WP: Warning Propagation-Based Linear-Time-and-Space Algorithms
    - Prune Leaves
    - Initialize Messages
    - Run Warning Propagation Iterations
    - Remove Redundant Vertices

Major Steps of MVC-WP

- MVC-WP: Warning Propagation-Based Linear-Time-and-Space Algorithms
  - Prune Leaves
  - Initialize Messages
  - Run Warning Propagation Iterations
  - Remove Redundant Vertices

Prune Leaves

- A leaf vertex is a vertex of degree 1.
- Idea: Adding a leaf vertex into the vertex cover is no better than adding its neighbor.



Adding the blue vertex to the vertex cover is at least as good as adding the red vertex. Therefore, we can remove the two colored vertices from the graph.

Major Steps of MVC-WP

- MVC-WP: Warning Propagation-Based Linear-Time-and-Space Algorithms
  - Prune Leaves
  - Initialize Messages
  - Run Warning Propagation Iterations
  - Remove Redundant Vertices

Initialize Messages: Assuming Random Graph Models

- Standard warning propagation initialize all messages to zero, but we should be able to do better.
- We treat the pruned graph as if it were generated by a random graph model (but the input graph itself is not required to be generated by a random graph model).
  - Erdős-Rényi: Vertex degrees follow a Poisson distribution ($P(d) \propto c^d/d!$).
  - Scale-free: Vertex degrees follow a power law ($P(d) \propto d^{-\lambda}$).
- Basic idea: Initialize messages based on the values of the messages when warning propagation converges on an infinitely large random graph from analytical results.

Warning Propagation (WP) for the MVC Problem

An iterative algorithm proposed and has only been used for theoretical analysis of properties of MVC on infinite graphs by (Weigt et al. 2006).

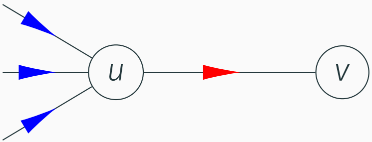
(a) *u* sends a message of 0 to *v* since one of its incoming messages from other vertices is 1.

(b) *u* sends a message of 1 to *v* since all other incoming messages are 0.

There is a message of either 0 or 1 along each direction of each edge. Assume that *u* and *v* are two adjacent vertices. *u* sends *v* a message of 1 to “warn” *v* to indicate that *u* will not be selected in the vertex cover. Otherwise, *u* sends *v* a message of 0.

Initialize Messages: Convergent Message Values on Infinite Graphs

On a given infinitely large random graph, let p_0 denote the probability that a message is zero and assume that p_0 is constant for all messages.



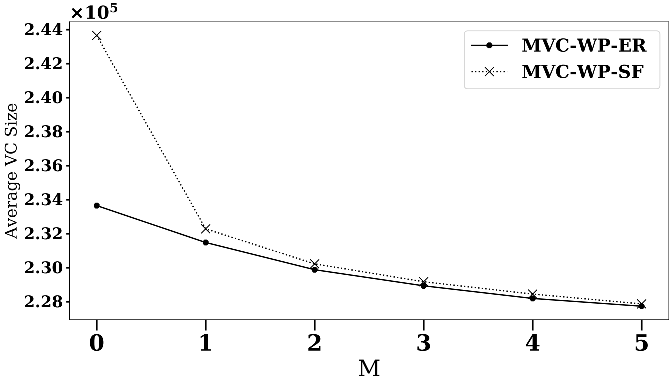
$$1 - p_0 = \sum_{d=1}^{\infty} p_0^{d-1} P(d)$$

- Erdős-Rényi: $p_0 = 1 - W(c)/c$ (Weigt et al. 2006)
- scale-free: $p_0 = (\zeta(\lambda) - 1)/(\zeta(\lambda) + \frac{1}{2^\lambda})$
- Randomly initialize messages according to the calculated p_0.

Major Steps of MVC-WP

- MVC-WP: Warning Propagation-Based Linear-Time-and-Space Algorithms
  - Prune Leaves
  - Initialize Messages
  - Run Warning Propagation Iterations
  - Remove Redundant Vertices

Run Warning Propagation Iterations



M: Number of iterations

- How many iterations?
- We choose $M = 3$: A small number but has fairly good effectiveness.

Major Steps of MVC-WP

- MVC-WP: Warning Propagation-Based Linear-Time-and-Space Algorithms
  - Prune Leaves
  - Initialize Messages
  - Run Warning Propagation Iterations
  - Remove Redundant Vertices

Remove Redundant Vertices

- Proposed by (Cai 2015).
- Starting from a vertex cover, for each edge, if it has both endpoint vertices in the vertex cover, mark them as removable.
- Recursively remove removable vertices and update marks of other vertices accordingly.
- Guaranteed to output a minimal vertex cover.

Agenda

- Motivation
- MVC-WP: Warning Propagation-Based Linear-Time-and-Space Algorithms
- Experimental Evaluation
- Conclusion and Future Work

Benchmark Instances

- Network repository (`http://networkrepository.com/`)
  - misc networks (397)
  - web networks (18)
  - brain networks (26)
- Street networks (8) (`http://www.cc.gatech.edu/dimacs10/archive/streets.shtml`)
- All graphs have more than 100,000 vertices.
- We have also converted these graphs in various formats to the DIMACS format: `http://files.hong.me/papers/xu2018b-data`

Algorithms

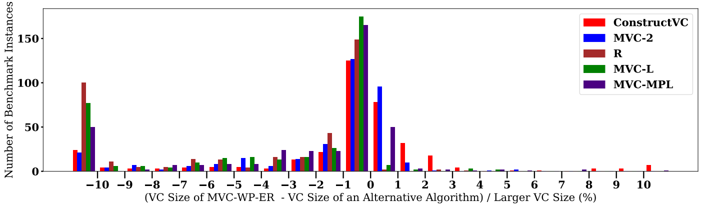
- Our Algorithms:
  - MVC-WP-ER: Assuming Erdős-Rényi random graphs.
  - MVC-WP-SF: Assuming scale-free random graphs.
- Competitors:
  - MVC-2 (Vazirani 2003)
  - ConstructVC (Cai 2015)
  - R (Andrade et al. 2012)
  - MVC-MPL (Xu et al. 2017)
  - MVC-L (Xu et al. 2017)
  - To ensure fair comparison, leaf pruning and redundant vertex removal are also applied to each algorithm.

Results

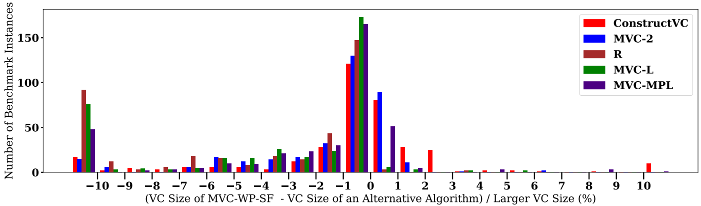
Our Algorithm	Alternative Algorithm	Misc (397)	Web (18)	Street (8)	Brain (26)
MVC-WP-ER	ConstructVC	211/39/147	12/1/5	8/0/0	0/0/26
	MVC-2	241/46/110	16/1/1	8/0/0	26/0/0
	R	376/16/5	17/1/0	8/0/0	26/0/0
	MVC-MPL	317/18/62	17/1/0	1/0/7	26/0/0
	MVC-L	364/19/14	17/1/0	8/0/0	26/0/0
MVC-WP-SF	ConstructVC	209/38/150	11/1/6	8/0/0	0/0/26
	MVC-2	249/45/103	15/1/2	8/0/0	26/0/0
	R	377/15/5	17/1/0	8/0/0	26/0/0
	MVC-MPL	316/18/63	17/1/0	1/0/7	26/0/0
	MVC-L	363/21/13	17/1/0	8/0/0	26/0/0

Comparison of sizes of vertex covers produced by MVC-WP-ER and MVC-WP-SF, respectively, with those of alternative algorithms.

Results: Misc Networks

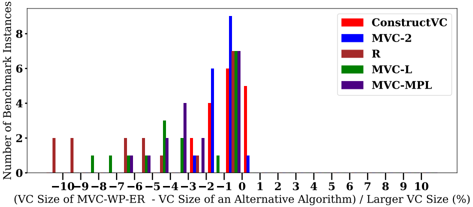


(a) MVC-WP-ER versus ConstructVC/MVC-2/R/MVC-L/MVC-MPL

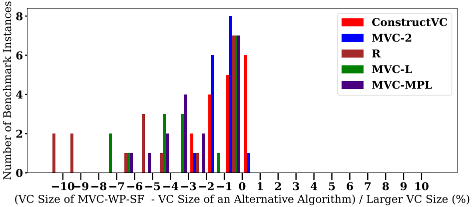


(b) MVC-WP-SF versus ConstructVC/MVC-2/R/MVC-L/MVC-MPL

Results: Web Networks

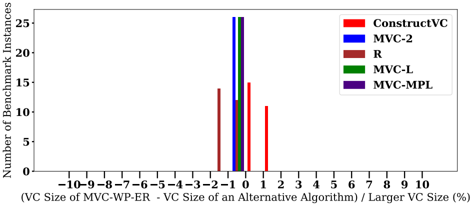


(a) MVC-WP-ER versus ConstructVC/MVC-2/R/MVC-L/MVC-MPL

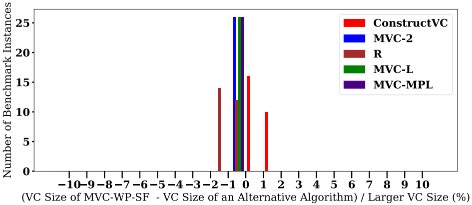


(b) MVC-WP-SF versus ConstructVC/MVC-2/R/MVC-L/MVC-MPL

Results: Brain Networks

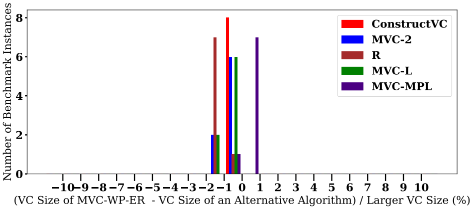


(a) MVC-WP-ER versus ConstructVC/MVC-2/R/MVC-L/MVC-MPL

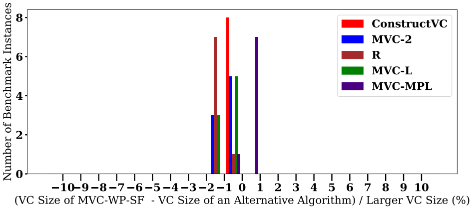


(b) MVC-WP-SF versus ConstructVC/MVC-2/R/MVC-L/MVC-MPL

Results: Street Networks



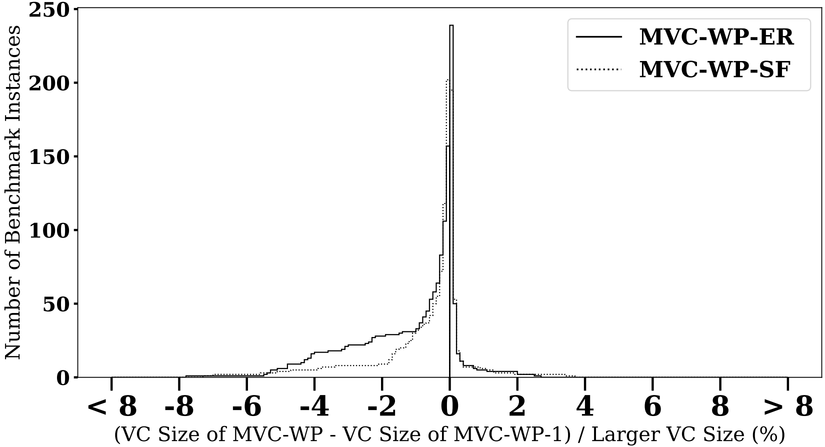
(a) MVC-WP-ER versus ConstructVC/MVC-2/R/MVC-L/MVC-MPL



(b) MVC-WP-SF versus ConstructVC/MVC-2/R/MVC-L/MVC-MPL

Is the Message Initialization Useful?

MVC-WP-1: The standard message initialization, i.e., all messages are 0.



Agenda

- Motivation
- MVC-WP: Warning Propagation-Based Linear-Time-and-Space Algorithms
- Experimental Evaluation
- Conclusion and Future Work

Conclusion and Future Work

- We need a linear-time-and-space algorithm for the MVC problem, because local search algorithms require a good starting vertex cover.
- We developed MVC-WP, warning propagation-based linear-time-and-space algorithms that find small vertex covers for giant graphs.
- Our experimental results showed that MVC-WP outperformed other competitors and each step of MVC-WP is important.
- We have compiled various sets of graphs in many different formats into a set of giant graphs in the DIMACS format: `http://files.hong.me/papers/xu2018b-data`
- (Future work) Combine MVC-WP with local search algorithms. Apply similar techniques on other combinatorial problems.

References I

F. N. Abu-Khzam, R. L. Collins, M. R. Fellows, M. A. Langston, W. H. Suters, and C. T. Symons. “Kernelization Algorithms for the Vertex Cover Problem: Theory and Experiments”. In: *the Workshop on Algorithm Engineering and Experiments*. 2004.

D. V. Andrade, M. G. C. Resende, and R. F. Werneck. “Fast local search for the maximum independent set problem”. In: *Journal of Heuristics* 18.4 (2012), pp. 525–547. DOI: `10.1007/s10732-012-9196-4`.

S. Cai. “Balance between Complexity and Quality: Local Search for Minimum Vertex Cover in Massive Graphs”. In: *the International Joint Conference on Artificial Intelligence*. 2015, pp. 747–753.

S. Cai, K. Su, C. Luo, and A. Sattar. “NuMVC: An Efficient Local Search Algorithm for Minimum Vertex Cover”. In: *Journal of Artificial Intelligence Research* 46.1 (2013), pp. 687–716.

É. Filiol, E. Franc, A. Gubbioli, B. Moquet, and G. Roblot. “Combinatorial Optimisation of Worm Propagation on an Unknown Network”. In: *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 1.10 (2007), pp. 2931–2937.

References II

W. Pullan. “Optimisation of unweighted/weighted maximum independent sets and minimum vertex covers”. In: *Discrete Optimization* 6.2 (2009), pp. 214–219. DOI: 10.1016/j.disopt.2008.12.001.

H. D. Sherali and M. Rios. “An Air Force Crew Allocation and Scheduling Problem”. In: *The Journal of the Operational Research Society* 35.2 (1984), pp. 91–103.

V. V. Vazirani. *Approximation Algorithms*. Springer, 2003.

M. Weigt and H. Zhou. “Message passing for vertex covers”. In: *Physical Review E* 74.4 (2006), p. 046110. DOI: 10.1103/PhysRevE.74.046110.

H. Xu, T. K. S. Kumar, and S. Koenig. “A Linear-Time and Linear-Space Algorithm for the Minimum Vertex Cover Problem on Giant Graphs”. In: *the International Symposium on Combinatorial Search*. 2017, pp. 173–174.