A Self Fixing Intelligent Ant-Based Algorithm for Graph Clustering

Ying Ying Liu, Parimala Thulasiraman

InterDisplinary Evolving Algorithmic Science Lab
Department of Computer Science, University of Manitoba, Canada
umliu369@myumanitoba.ca, thulasir@cs.umanitoba.ca

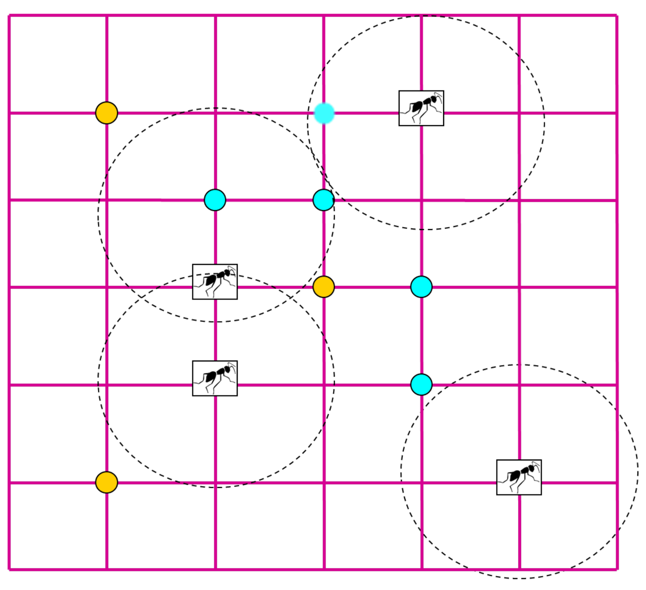
The Graph Clustering Problem

Given an undirected graph $G = (V, E)$, where V denotes the set of vertices and E is the set of edges, the graph clustering problem can be defined as dividing the vertices into k disjoint sets, $V_1, V_2, \ldots, V_k$, such that the following criteria are met:

- $\bigcup_{i=1}^{k} V_i = V \forall 1 \leq i \leq k$
- $V_i \cap V_j = \emptyset, \forall 1 \leq i, j \leq k$
- $\forall u_i \in V_i, u_j \in V_i, u_l \notin V_i, Similarity(u_i, u_j) \geq Similarity(u_i, u_l)$

Ant-based Clustering

Deneubourg et al.[1] develop a distributed sorting algorithm for robot applications based on clustering of corpses in a colony of *Pheidole Pallidula*. Lumer and Faieta [2] extend the ant sorting algorithm to cluster data objects on a grid. Kuntz et al. [3] propose ant-based clustering - KLS algorithm to solve the graph partitioning problem. In KLS, vertices and ants are randomly distributed on a 2D grid, and each ant has three states: random walk, pick up a vertex, and drop a vertex.



Educated parameters: k_p, k_d, α, l

$$P_{pickup}(i) = \left(\frac{k_p}{k_p + f(i)}\right)^2$$

$$P_{drop}(i) = \left(\frac{f(i)}{k_d + f(i)}\right)^2$$

$$f(i) = \begin{cases} \frac{1}{l}\sum_j \left(1 - \frac{d(i,j)}{\alpha}\right) & \text{if } f > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$d(i,j) = \frac{|N_i \triangle N_j|}{|N_i| + |N_j|}$$

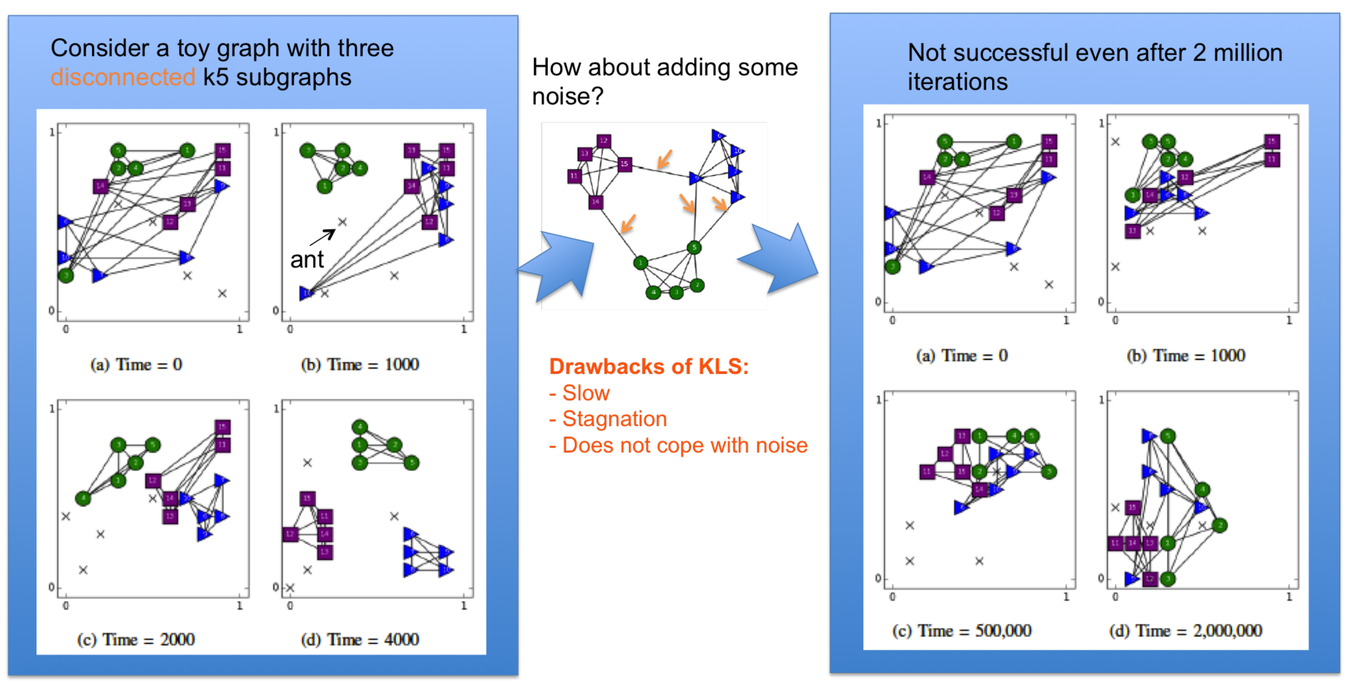


Figure 1: Ant-based Clustering - KLS

Intelligent Ant-based Clustering

Improvements made to KLS algorithms include:

- The random walk of an ant is now a combination of hopping[4] and walking. The hopping is for efficiency, and the walking is for exploration.
- Relaxed drop function[2] to prevent stagnation.

$$P_{drop}(i) = \begin{cases} 2f(i) & \text{if } f(i) < k_d \\ 1 & \text{otherwise} \end{cases}$$

- Each ant is assigned a small amount of memory[2] to remember the past few positions it resided.
- An ant is forced to drop the object if it passes a threshold of failed attempts.

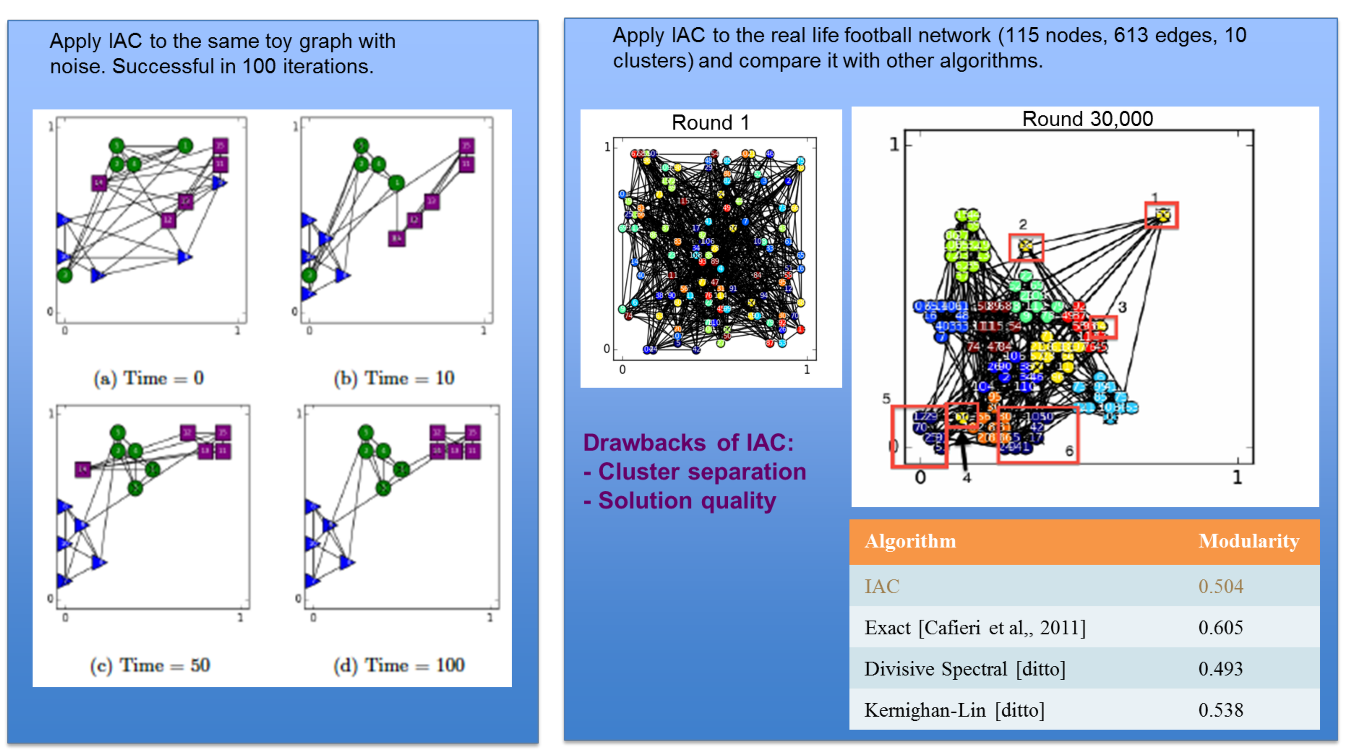


Figure 2: Intelligent Ant Clustering (IAC)

Self-fixing Intelligent Ant-based Clustering

Inspiration: The major and minor workers in *Pheidole* genus.

Pheidole genus	Major workers	Minor workers
Picture of *P. purpurea* [Wikipedia]		
Size	Larger	Smaller
Task	Foraging for food or defending the nest	Housekeeping tasks such as feeding the brood or cleaning the nest

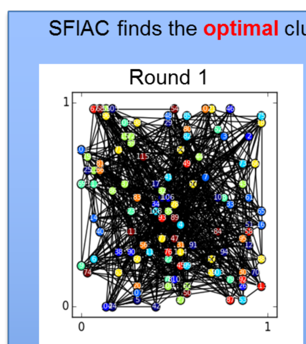
SFIAC	Major workers	Minor workers
Workload	Every iteration	Every N (e.g. 1000) iterations
Task	Run IAC	Improve cluster separation and exploit global optimality: move objects based on modularity gain

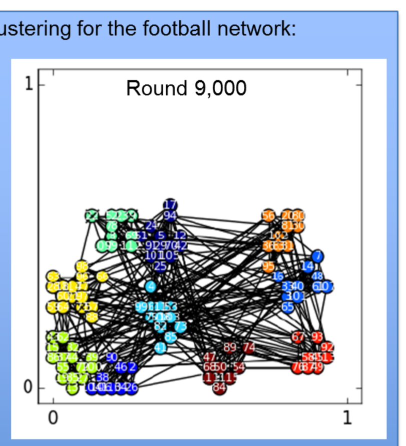
Figure 3: The Major and Minor Ants in SFIAC

Comparison with other algorithms on benchmark graphs

	Louvain	ACO-MMAS	IAC	SFIAC
karate k=2			✓	✓
karate k=4		✓		
dolphins	✓			
lesmis	✓			
polbooks				✓
adjnoun		✓		
football		✓		✓
jazz				✓
celegansneural	✓			
celegans metabolic	✓			
email	✓			

SFIAC finds the **optimal** clustering for the football network:





Drawbacks of SFIAC:
- Computation cost for minor ants

Algorithm	Modularity
IAC	0.504
SFIAC	0.605
Exact [Cafieri et al,, 2011]	0.605
Divisive Spectral [ditto]	0.493
Kernighan-Lin [ditto]	0.538

Figure 4: Evaluations

Conclusion

To improve the original KLS algorithm, we introduce *Intelligent Ant-based Clustering (IAC)* using techniques such as hopping ants, relaxed drop function, ants with memories, and stagnation control. Based on IAC, we introduce *Self Fixing Intelligent Ant-based Clustering (SFIAC)* that adds housekeeping minor ants to improve the solution quality. When tested on the benchmark networks, SFIAC outperforms or achieves the same solution quality as both ACO-MMAS and IAC on 7 out of 10 networks and is robust against different graphs. In practice, the speed of SFIAC is at least 10 times faster than ACO-MMAS, making it a comparatively scalable algorithm.

Forthcoming Research

We are working on an interdisciplinary collaboration project with Department of Biological Sciences of University of Manitoba to observe the clustering behaviors of real ants.

References

[1] Jean-Louis Deneubourg, Simon Goss, Nigel Franks, Ana Sendova-Franks, Claire Detrain, and Laeticia Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pages 356–363, 1991.

[2] Erik D Lumer and Baldo Faieta. Diversity and adaptation in populations of clustering ants. In *Proceedings of the third international conference on Simulation of adaptive behavior: from animals to animats 3*, pages 501–508. MIT Press, 1994.

[3] Pascale Kuntz, Dominique Snyers, and Paul Layzell. A stochastic heuristic for visualising graph clusters in a bi-dimensional space prior to partitioning. *Journal of Heuristics*, 5(3):327–351, 1999.

[4] Jian Ping Wang, Eseosa Osagie, Parimala Thulasiramn, and Ruppa K. Thulasiram. Hopnet: A hybrid ant colony optimization routing algorithm for mobile ad hoc network. *Ad hoc Networks*, 7(4):690–705, June 2009.

Acknowledgements

University of Manitoba, Natural Sciences and Engineering Research Council (NSERC) Canada.