A Novel Genetic Algorithm Based on Partitioning for Large-Scale Network Design Problems

Xiao-Ma Huang [a], Yue-Jiao Gong [a], Jing-Jing Li [b] (Corresponding Author) and Xiao-Min Hu [c]
[a] Department of Computer Science, Sun Yat-sen University
[a] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education
[a] Key Laboratory of Software Technology, Education Department of Guangdong Province
[b] School of Computer Science, South China Normal University
[c] School of Public Health, Sun Yat-sen University, P.R. China
jingjing.li1124@gmail.com

ABSTRACT
Network design is a broad class of essential engineering and science problems. The target of network design is to construct a graph that satisfies some restrictions. Many network design problems (NDPs) are known as NP-hard and become more challenging as networks grow fast in size. In this paper, we propose a novel genetic algorithm based on partitioning, termed PGA, which divides large-scale NDPs into low dimensional sub-problems and achieves global optimal solution by coordination of sub-problems. Experiments with PGA applied to the degree-constrained minimum spanning tree problem have shown the effectiveness of PGA for large-scale NDPs.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic methods*

Keywords
Network design; genetic algorithm; degree-constrained minimum spanning tree problem; problem decoupling; co-evolution

1. INTRODUCTION
With the rapid development of electronic and information technology, the size of networks and the complexity of Network Design Problems (NDPs) become enormous. The expansion of the search space of such large-scale NDPs makes it difficult for traditional Evolutionary Algorithms (EAs) to find satisfactory solutions in reasonable computational time.

To meet this challenge, this paper proposes a novel problem decoupling approach, termed Partitioning-based Genetic Algorithm (PGA) to tackle large-scale NDPs. By partitioning method, a given NDP is usually decomposed into several sub-problems by breaking the original network into small partitions. Then an optimization algorithm such as Genetic Algorithm (GA) is applied to each sub-problems independently. At the last stage, the solution of the original problem is obtained by merging the individual sub-solutions. In some cases, this approach show great advantages in solving high-dimensional problems. On one hand, the complexity of the problem is reduced dramatically. On the other, the total computational time can be decreased significantly if the optimization tasks of the sub-problems are dispatched to different processors. However, for most complex NDPs, after partitioning, one sub-network may have an implicit impact on another during the evolution. In this case, the accuracy loss of the result may occur if simply uniting the sub-networks which are optimized separately.

To overcome this shortcoming of partitioning method for large-scale NDPs, a scheme of co-evolution [1] may help. In the co-evolution scheme, a local evolutionary search in the subspace is performed on each processor and an infrequent inter-communication between subspaces is accessed. By the cooperative search of subspaces, the global optimal solution is found in the end.

2. ALGORITHM
The algorithm first runs a clustering algorithm to decompose the whole network into small partitions. Then a optimization algorithm is carried out to evolve these partitions of small sub-networks. The optimization of each sub-network uses the framework of a generic genetic algorithm employing the edge-set representation (GAES) [2]. During the optimization of each sub-network, the algorithm employs a co-evolution scheme to tune the evolution tendency of each sub-component by inter-communication. The global optimal solution is finally achieved by co-evolution of the sub-problems.

2.1 Network Partitioning
As for large-scale NDPs, it is promising to optimize the problem decoupled by using clustering-based partitioning methods. Clustering of the nodes in a network can discover structure and recover natural groups of nodes, dividing the network into several partitions. Nodes in the same partition are often closer than those in different partitions. That means nodes in different partitions may have weaker inherent interactions, which makes the algorithm effective to find good solutions of the entire problem by optimizing each sub-problem.

Given a network with weights on each edge between nodes, the algorithm uses k-medoid clustering [3] to accomplish the network partitioning process, generating k sub-networks with closest nodes in them.

2.2 Sub-network Optimization
After the network partitioning process, the entire network is divided into k disjoint sub-networks. Each of the sub-networks is then optimized using the framework of a generic genetic algorithm employing the edge-set representation (GAES) [2]. For each sub-problem, a GAES is carried out on a processor to evolve

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2881-4/14/07.
http://dx.doi.org/10.1145/2598394.2598431

the sub-network. During the optimization process, each GAES keeps a population of NP potential solutions of the sub-problem.

2.3 Co-evolution

As is mentioned before, the evolvement of one partition may have reliance on others. Even if each sub-network is perfectly optimized and reaches an optimal sub-solution, it may lead to bad network design (with a large total cost) as a whole by simply jointing these optimized sub-components.

In this case, the PGA introduces a co-evolution scheme to achieve the global optimal solution of the entire problem. As a sub-network evolves itself by iterative reproduction and selection process, it interacts with other sub-networks at intervals of p generations. By interaction with each other, the sub-networks can tune their evolution tendency to avoid local optima of the whole NDP. The interaction can be done by applying random spanning tree algorithm to the graph $G_{merge} = (V, \bigcup_{i=1}^{k} T_{subi} \cup T_{entire})$, where T_{entire} is the edge set of the spanning tree of the entire network and T_{subi} is the edge set of the spanning tree of the ith sub-network.

3. EXPERIMENTS AND COMPARISONS

Experiments are carried out to test the proposed PGA on twelve hard dc-MSTP instances [4] with degree constraint d=5. The proposed PGA was compared with the genetic algorithm using the edge-set representation (GAES) proposed in [2], which does not apply network partitioning method. Both of PGA and GAES use pc=0.8 and pm=0.8. For GAES, the population size NP=100, whereas for PGA, the number of sub-populations k is set to be 4 empirically and the size of each sub-population NSP=20, the inter-communication interval p=20. Each case is executed independently 30 times with maximum number of function evaluations $NFEs$=10000.

We can see from the table clearly that PGA makes significant improvement in GA for large instances. It is because the network partitioning method divide the high-dimensional NDP into several low-dimensional problems, which decreases the complexity of the original problem and local optimization of sub-networks accelerates the convergence. In this way, the efficiency of the algorithm for complex NDP is improved.

Table 1. Best Gap and mean Gap of solutions of GAES and PGA

		GAES		PGA	
instance name	**C_{opt}**	**Gap_{best} (%)**	**Gap_{avg} (%)**	**Gap_{best} (%)**	**Gap_{avg} (%)**
m050n1	6.60	2.46	7.40	**1.98**	**6.43**
m050n2	5.78	2.67	9.19	**2.64**	**7.53**
m050n3	5.50	**0.20**	**2.91**	0.70	4.62
m100n1	11.08	7.46	13.23	**1.46**	**6.06**
m100n2	11.33	9.56	15.21	**4.00**	**7.81**
m100n3	10.19	13.37	17.76	**3.54**	**8.40**
m200n1	18.33	27.17	30.65	**5.50**	**8.89**
m200n2	19.16	27.98	35.69	**8.29**	**12.21**
m200n3	16.13	28.55	31.52	**4.72**	**7.59**
m300n1	40.69*	17.92	19.38	**8.90**	**10.47**
m400n1	54.69*	27.47	29.28	**16.65**	**18.32**
m500n1	79.34*	24.85	25.66	**17.00**	**18.48**

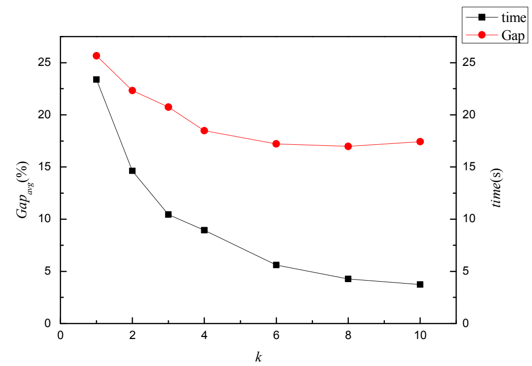

Figure 1. Gap_{avg} and average time of PGA on m500n1 with different number of sub-populations

To do a more extensive analysis, an extensive experiment on the M-graph instance m500n1 is carried out. From Figure 1, we can see that with k increasing, the problem is divided into sub-problems with lower dimension and needs less consuming time. However, shortening total executive time is at the cost of more processors. Moreover, a bigger k doesn't mean better solutions. Although partitioning method can decrease the complexity of the problem and result in faster convergence, yet too many partitions will weaken the global exploration of the algorithm.

4. CONCLUSION

This paper proposes a novel genetic algorithm based on partitioning for large-scale network design problems. The algorithm uses a clustering algorithm to decompose the total NDP into several small cooperative sub-problems. It then introduces a co-evolution scheme to tune the evolution tendency of each sub-component by inter-communication during the sub-network optimization process. By coordination of each sub-component, fairly good global solutions of large-scale NDPs can be achieved. Experiments are carried out to compare the proposed PGA with the original GA. The experiment results show that the proposed network partitioning method improves the efficiency and makes an improvement in the performance of GA for large-scale complex NDPs.

5. ACKNOWLEDGMENTS

This work was supported in part by the NSFC No.61379061, No.61309003, No.61202130, and No.61379060, in part by NSFC Joint Fund with Guangdong under Key Projects U1201258 and U1135005.

6. REFERENCES

[1] R.Subbu and A. C. Sanderson, "Modeling and convergence analysis of distributedcoevolutionaryalgorithms,"*IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 34, no. 2, pp. 806-822, Apr. 2004.

[2] G. R. Raidl, and B. A. Julstrom, "Edge sets: an effective evolutionary encoding of spanning trees," IEEE Trans. Evol. Comput., vol. 7, no. 3, pp. 225-239, June 2003.

[3] C.B. Lucasius, A.D. Dane and G. Kateman, "On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasibility and comparison," Analytical Chimica Acta, vol. 282, no.3, pp. 647-669, 1993.

[4] J. Knowles, and D. Corne, "A new evolutionary approach to the degree-constrained minimum spanning tree problem," IEEE Trans. Evol. Comput., vol. 4, no. 2, pp. 125-134, July 2000.