

Efficient Analysis to Protect Control into Critical Infrastructures

Shuo Zhang and Stephen D. Wolthusen

School of Mathematics and Information Security
Royal Holloway University of London
Egham UK TW20 0EX

MYVA375@live.rhul.ac.uk; stephen.wolthusen@rhul.ac.uk

Abstract. To protect control into critical infrastructures against single component-dependency attacks or failures, we analyse the importance of any given dependency in maintaining controllability with a minimum set of inputs. Since people use critical, redundant and ordinary categories to clarify how an edge maintains controllability of linear time-invariant (LTI) dynamical networks, according to graph-based models of infrastructures and the minimum input theorem, we firstly use a *Erdős-Rényi* random digraph with a precomputed maximum matching to model some LTI and controllable infrastructures by a minimum set of inputs. We then efficiently analyse any given arc's category before and during single-arc removals, as a way to further confirm how related dependency keeps control into infrastructures. After running our label operations with linear time and space complexity, any edge-category analysis can be thus executed in $O(1)$ time in both cases.

Keywords: Network Controllability · Modelling · Edge Classification

1 Introduction

Malicious attacks or random failures on pairwise dependencies among critical infrastructure components might make critical infrastructures out of control, and may lead unavailability of purposed products and services in a large region for a significant length of time, which would cause severe economic impacts or life loss and limb [7]. Thus, efficient analysis on any given pairwise dependency between components in terms of keeping control into infrastructures is forward-looking to protect critical infrastructures. Given a controllable and linear time-invariant (LTI) critical infrastructure by a minimum set of inputs, according to the graph-based models of critical infrastructures [8] and the minimum input theorem used to control networks with LTI dynamics, to protect its controllability, we model this given infrastructure by a large, sparse *Erdős-Rényi* random digraph, which also contains a precomputed maximum matching, as an input graph [1]. Then, we solve the problem of efficient edge-category analysis on the input graph to classify any given edge into critical, redundant or ordinary categories before and during single-edge removals[6], so that how related pairwise dependency keeps

control into critical infrastructure can be known. Specifically, removing a critical edge increases the minimum number of inputs; the absence of an ordinary edge only changes control structure rather than the minimum number of inputs; removing a redundant edge changes nothing [6]. Our edge-category analysis is executed through a bipartite graph mapped by the input graph. With this bipartite graph, we firstly introduce a generally static analysis, on which there is no edge removals. Based on our previous work [9], searching and labelling alternating cycles and paths related to a given maximum matching helps to confirm any edge's category. During the single-edge removal process, in addition of edge-category analysis, control into the residual network per edge removal should be recovered as well. With constraints on degree distribution of the input graph, label operations and control recovery are executed in $O(1)$ amortized time per edge removal. For our contribution, excluding the precomputed maximum matching of the input graph, by label operations executed in linear time and space complexity, category of any given edge of the input graph can be confirmed in $O(1)$ time. In the following paper: section 2 models critical infrastructures and formulates research question; section 3,4 executes static and dynamic edge-category analysis respectively; section 5 gives conclusion.

2 Modelling

By control theory [4,5], an linear time-invariant infrastructure with external inputs can be described by a differential equation:

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t) \quad (1)$$

where $x(t) \in \mathbb{R}^N$ is the system state vector holding the state of each infrastructure component at time t , and $x(t) = (x_1(t), x_2(t), \dots, x_N(t))^T$. $u(t) \in \mathbb{R}^M (M \leq N)$ is the control input vector, and $u(t) = (u_1(t), u_2(t), \dots, u_M(t))^T$. System matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ shows the interactions among N components, while input matrix $\mathbf{B} \in \mathbb{R}^{N \times M}$ shows interactions among M inputs and N components. A system described by equation 1 is fully controllable if and only if the rank of the matrix $\mathbf{C} \in \mathbb{R}^{N \times NM}$, where $\mathbf{C} = [\mathbf{B}, \mathbf{A}\mathbf{B}, \mathbf{A}^2\mathbf{B}, \dots, \mathbf{A}^{N-1}\mathbf{B}]$, has full rank, noted by $rank(\mathbf{C}) = N$. Concerning graph-based model of the critical infrastructures [8], we model the infrastructure by a digraph defined below:

Definition 1 (Modelling Digraph). *Given a \mathbf{A} of equation 1, let $G(\mathbf{A}) = (V_1, E_1)$ be a digraph, and $\alpha : \{\mathbf{A}\} \rightarrow G(\mathbf{A})$ be a bijection. For each non-zero entry $a_{ij} \in \mathbf{A}$, there are $\alpha : a_{ij} \rightarrow \langle v_j, v_i \rangle$, where $\langle v_j, v_i \rangle \in E_1$ and $\{v_i, v_j\} \subseteq V_1$.*

And controllability of $G(\mathbf{A})$ with a minimum set of inputs is confirmed by following theorem:

Theorem 1 (The Minimum Input Theorem [6]). *The minimum number of inputs to fully control a modelling digraph is one, if it has a perfect matching, where the input can directly drive any vertex. Otherwise, it equals to the number of unmatched nodes, which must be directly driven by the same number of inputs.*

In digraphs, a maximum matching is a set of arcs neither sharing common heads nor tails with highest cardinality [1]. When all nodes of a digraph are heads of arcs of a maximum matching, this digraph has a perfect matching. Also, there might be multiple maximum matchings in a same graph. By contrast, a maximum matching of a bipartite digraph is a set of vertex-disjoint edges with the highest cardinality. Above all, we define our modelling digraph, which is our input graph of following edge classification:

Definition 2 (Input Graph). *Let $D = (V, E)$ be a large, sparse and finite Erdős-Rényi digraph, where $V = \{v_i | 1 \leq i \leq N\} (N > 3)$, $E = \{\langle v_i, v_j \rangle | i \neq j, v_i, v_j \in V\} (|E| > 3)$. D excludes selfloops, parallel arcs and isolated nodes. Also, let M_D be a precomputed maximum matching of D by the algorithm of [3].*

With $D = (V, E)$ of definition 2, efficiently analysing how any given dependency among infrastructure components maintains the control into the infrastructure with a minimum set of inputs, can be modelled into efficient edge-category analysis on D to confirm its any edge's category. Further more, we solve such analysis before and during the process of single-edge removals.

3 Generally Static Edge Analysis

This section confirms the category of any given arc of $D = (V, E)$ of definition 2, when no arc is removed from D . A bipartite graph defined below to execute following edge-category analysis with items of definition 4 and corollary 1, 2:

Definition 3 ($B = (V_B, E_B)$). *Given $D = (V, E)$ with M_D of definition 2, let $B = (V_B, E_B)$ be an undirected bipartite graph, β be a bijection. Also, let V_B^-, V_B^+ be two independent sets, where $V_B = V_B^+ \cup V_B^-$ and M_B be a maximum matching of B . For any $\langle v_i, v_j \rangle \in E$, $\beta : \langle v_i, v_j \rangle \rightarrow (v_i^+, v_j^-)$, where $v_i^+ \in V_B^+, v_j^- \in V_B^-$ and $(v_i^+, v_j^-) \in E_B$. Besides, let M_B mapped by M_D .*

Definition 4 (Alternating Cycle & Alternating Path [9]). *Given $B = (V_B, E_B)$ with M_B of definition 3, a set of edges is either an alternating path or cycle, iff it alternatively involves the same number of edges of M_B and $E_B \setminus M_B$.*

Corollary 1. *In $D = (V, E)$, let e be an edge of E , given $B = (V_B, E_B)$ and M_B , let e' be an edge of E_B and mapped by e . Then, e is a critical edge iff $e' \in M_B$ and out of any alternating cycles or paths related to M_B [9].*

Corollary 2. *In $D = (V, E)$, let e be an edge of E , given $B = (V_B, E_B)$ and M_B , let e' be an edge of E_B and mapped by e . Then, e is an ordinary edge iff $e' \in M_B$ and involved into an alternating cycle or path related to M_B [9].*

Thus, static analysis depends on finding all alternating paths and cycles related to M_B . Specifically, each identified edge is a mapped by an ordinary edge related to M_D of D , while each edge of M_B not identified corresponds to a critical edge of D . And any edge out of M_B and not identified is related to a redundant edge of D . To view related algorithms, please ask for the first author.

4 Conditionally Dynamic Edge Analysis

This section confirms the category of any given arc of residual network after removing single edges from $D = (V, E)$ of definition 2, where control recovery per single-edge removal is also implemented. Solving this problem also depends on $B = (V_B, E_B)$ of definition 3 during removing $p(1 \leq p < |E_B|)$ single edges. To increase efficiency, we assume that: (i) the number of in-degree and out-degree of any node of D should be less than or equal to 2; (ii) the average degree of D is bigger than one. By these assumptions, all paths and cycles can not share common vertex incident to M_B . Even though, several cases still should be concerned and related algorithms can be asked for the first author.

5 Conclusion

We use a digraph with a precomputed maximum matching to model a LTI controllable critical infrastructure with a minimum set of inputs, and execute efficient edge-category analysis, to confirm the importance of any given pairwise dependency in keeping controllability against removing single infrastructure dependency. We find and label alternating paths and cycles of a bipartite graph mapped by the graph model during continuous single-edge removals to support edge-category analysis. As a result, our entire operations cost linear time and space in the worst case for both static and conditionally dynamic edge analysis. Also, based on the aggregate analysis [2], a label operation per single-edge removal for conditionally dynamic edge analysis costs $O(1)$ amortized time. With those operations, category of any given arc can be confirmed in $O(1)$ time in both cases, which excludes label operations.

References

1. Chartrand, G., Lesniak, L., Zhang, P.: Graphs & digraphs. CRC Press (2010)
2. Cormen, T.H.: Introduction to algorithms. MIT press (2009)
3. Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing* **2**(4), 225–231 (1973)
4. Kalman, R.: On the general theory of control systems. *Automatic Control, IRE Transactions on* **4**(3), 110–110 (1959)
5. Kalman, R.E.: Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control* **1**(2), 152–192 (1963)
6. Liu, Y.Y., Slotine, J.J., Barabási, A.L.: Controllability of complex networks. *Nature* **473**(7346), 167–173 (2011)
7. Lopez, J., Setola, R., Wolthusen, S.: Critical Infrastructure Protection: Advances in Critical Infrastructure Protection: Information Infrastructure Models, Analysis, and Defense, vol. 7130. Springer Science & Business Media (2012)
8. Svendsen, N.K., Wolthusen, S.D.: Modelling approaches. In: *Critical Infrastructure Protection*, pp. 68–97. Springer (2012)
9. Zhang, S., Wolthusen, S.D.: Security-aware network analysis for network controllability. In: *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE (2018)