An Exploration of the Network Installation and Recovery Problem with Blackstart Nodes

Kayla S. Cummings, Janie L. Neal, Andi Chen, and Tzu-Yi Chen(✉)

Computer Science Department, Pomona College, Claremont, CA 91711, USA
{ksc72014,jlnn2015,ac002014}@mymail.pomona.edu, tzuyi.chen@pomona.edu

Abstract. The Neighbor Aided Network Installation Problem asks how best to install the nodes in a network under the assumption that the cost of installing a node depends solely on the number of its neighbors which have been previously installed. We study a version which incorporates blackstart nodes into the model. Under the assumption of a decreasing convex cost function and a single blackstart node, we solve the problem for subclasses of almost-complete and augmented-tree networks. We also describe two heuristics and present experimental results on real world networks.

Keywords: Network models · Network recovery · Blackstart nodes

1 Introduction

The electric power transmission grid in the contiguous United States consists of approximately 120,000 miles of transmission lines operated by approximately 500 companies. This complex interconnected network, regulated by the North American Electric Reliability Corporation, delivers power from generators to individual consumers [10].

Ideally this network runs smoothly at all times. In practice the flow of electricity from generators to consumers can be interrupted by power outages, also known as blackouts. Because of the network structure, outages can cascade, leading to blackouts that affect large geographic areas. Once a blackout has occurred, the goal is to restore power as efficiently as possible. This can require not only repairing physical damage by, for example, clearing fallen tree branches, but also restarting power generators in an appropriate order. Approaches for determining how to restart the power grid after an outage include incorporating a subset of the variables into a complex mixed integer linear programming problem [9], applying heuristics motivated by complex network theory [7], agent-based approaches [6,8].

In contrast to this heuristic work, a more theoretical approach is taken in [4,5]. Here a graph problem which models a simplified version of the power recovery problem is described and studied.

L. M. Aiello et al. (Eds.): COMPLEX NETWORKS 2018, SCI 812, pp. 652–662, 2019.
https://doi.org/10.1007/978-3-030-05411-3_52

In this paper we extend the model proposed in [4] by incorporating blackstart nodes, which represent nodes in the network that must be started first because they contain the small diesel generators that some power stations have to facilitate a restart after a wide-area power outage. After such a power outage, these nodes are the only ones that can be restarted without any of their neighbors being restarted first. The question then becomes: given a network and a set of blackstart nodes, what is the most cost-efficient way to restart the entire network? We answer this question for a restricted set of networks. We also describe two heuristics and discuss preliminary experimental results on real world networks.

2 Background

An instance of the *Neighbor Aided Network Installation Problem* (NANIP) as defined in [4] consists of a network $G = (V, E)$ with an assigned cost function $f : \mathbb{N}_0 \rightarrow \mathbb{R}_+ \cup \{0\}$. If the highest-degree node in V has degree $\deg(V)$, then the domain of f is defined by $\mathbb{N}_0 = \{n \in \mathbb{Z} : 0 \leq n \leq \deg(V)\}$. The cost of installing a node after k of its neighbors have already been installed is $f(k)$. Given a permutation σ of the nodes in V, the cost of σ is denoted $C_\sigma(G)$ and is defined as the sum of the costs of installing each node in G in that order. The goal is to find a minimum-cost permutation of the nodes.

In [4] they show the general NANIP problem is NP-hard and then focus on the case where the cost function f is decreasing convex. A function $f : \mathbb{N}_0 \rightarrow \mathbb{R}_+ \cup \{0\}$ is *decreasing convex* if $f(i) - f(i+1) \geq f(j) - f(j+1)$ for all $i, j \in \mathbb{N}_0$ with $j \geq i$. This assumption models an essential aspect of network installation. As the number of previously installed nodes increases, the installation cost of one extra node decreases due to the extra neighbor aid. Convexity captures diminishing returns: the benefit of extra neighbor aid decreases as the number of neighbors increases. In other words, the benefit of adding a fifth neighbor that is already installed is more significant than the benefit of adding a hundredth neighbor.

In [4] they prove a lower bound on the cost of any permutation for the case where f is decreasing convex. As a corollary, if the graph G is a tree graph, they prove the optimal permutation σ attains a cost of:

$$C_\sigma(G) = f(0) + (n-1)f(1) \tag{1}$$

Then, in [5], it is shown that the NANIP problem remains NP-hard even for decreasing convex cost functions.

3 Blackstart Nodes

The NANIP-Blackstart problem extends the model in [4] by adding blackstart nodes. As previously described, blackstart nodes represent the only nodes that can be restarted before any of their neighbors. In this situation, all solutions σ

must be *valid* permutations, which means every node that is not a blackstart node must have at least one neighbor appear earlier in the permutation.

For now, assume there is a single blackstart node v_b and that the cost of installing v_b is $f(0)$.

Proposition 1. *NANIP-Blackstart with a single blackstart node is NP-hard*

Proof. Consider a decision version of NANIP-Blackstart in which a yes instance means there exists an ordering with cost less than c. An instance of NANIP $\{G = (V, E), f, c\}$ can be reduced to an instance of NANIP-Blackstart as follows. Let $G' = (V', E')$, where $V' = V \cup v_0$ and $E' = E \cup F$ where $F = \{(v, v_0) | v \in V\}$. Let v_0 be the single blackstart node in B. Define f' so $f'(k) = f(k-1) + 1$ for all $k \geq 1$ and $f'(0) = 0$. Let $c' = c$.

A valid ordering for the instance of NANIP-Blackstart is a minimum cost ordering for the instance of NANIP without the blackstart node. The blackstart version means the installed nodes must always form a connected component. However, since every node in V is connected to v_0 by an edge that costs nothing, this does not restrict the valid orderings in the instance of NANIP-Blackstart. □

We now turn to proving exact solutions for restricted subclasses of graphs before presenting the results of experiments using two heuristics.

3.1 Augmented Trees

Because many real world electrical networks have low treewidth [2], it is useful to study the NANIP-Blackstart problem on trees and on augmented tree graphs.

Proposition 2. *Let T be a tree with n nodes and let σ be a valid permutation of the nodes. Then*

$$C_\sigma(T) = f(0) + (n-1)f(1). \tag{2}$$

Proof. The constraint added by including a blackstart node means the installed portion of the tree must always be connected. This gives the minimum cost solution for a tree without a blackstart node [4]. □

Trees Augmented with Distinct Cycles

Definition 1. *A set of k cycles are distinct if no two cycles share an edge. In Fig. 1, the two cycles (A,B,C) and (B,D,E) are distinct. The two cycles (A,B,C) and (B,C,F,E) are not distinct.*

Proposition 3. *Let T be a tree graph with n vertices. Let T_m be a graph with m distinct cycles that is created by adding m edges to T. The cost of any valid permutation σ is given by the formula:*

$$C_\sigma(T_m) = f(0) + (n-m-1)f(1) + mf(2). \tag{3}$$

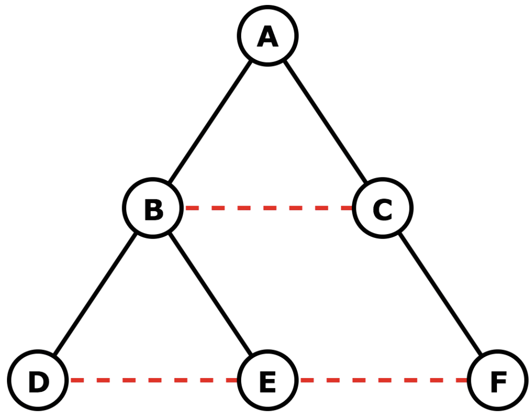

Fig. 1. A tree augmented with three edges (drawn as dashed lines).

Proof. Partition V into 3 sets: V_B, V_C, and V_A where nodes in V_C are nodes that are the last node installed in a cycle, V_B contains only the blackstart node, and $V_A = V \backslash \{V_B \cup V_C\}$. The cost of installing the node in V_B is $f(0)$ by definition. The cost of installing each node in V_C is $f(2)$, while the cost of each node in V_A is $f(1)$. □

Wheel Graphs

Definition 2. *Let $C = (V, E)$ be a cycle graph and let $|V| = n$. Let $V' = V \cup v'$ and let $E' = E \cup \{(v', v)|v \in V\}$. We define $W_m = (V', E')$ as a wheel graph on $m = n + 1$ vertices (Fig. 2).*

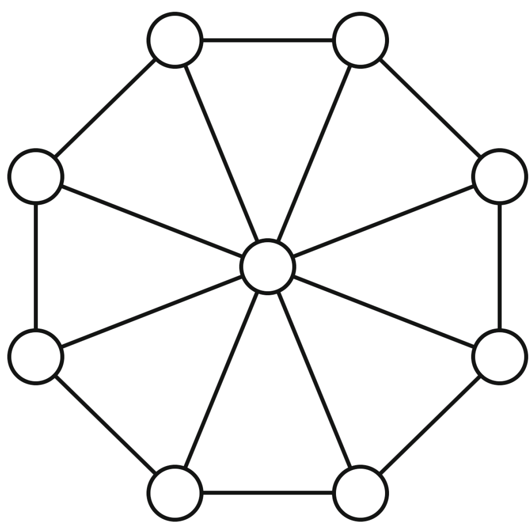

Fig. 2. The wheel graph W_9.

Proposition 4. *Let W_n be a wheel graph and let the central vertex be the blackstart node. The optimal permutation σ has cost*

$$C_\sigma(W_n) = f(0) + f(1) + (n-3)f(2) + f(3). \tag{4}$$

Proof. The second node in σ incurs a cost of $f(1)$. If σ goes around the cycle taking the vertices in order, Eq. 4 gives the overall cost. If we change this ordering so that a non-contiguous vertex is chosen from the cycle at any point, this would convert two of the $f(2)$ costs to an $f(1) + f(3)$. The latter is more expensive since f is a decreasing convex function. □

Proposition 5. *Let W_n be a wheel graph such that $n \geq 3$. Let v_c be the central node and let one of the nodes in the cycle be the blackstart node. Depending on the function f, the optimal permutation σ has cost either*

$$C_\sigma(W_n) = f(0) + f(1) + (n-3)f(2) + f(3) \tag{5}$$

or

$$C_\sigma(W_n) = f(0) + (n-3)f(1) + f(3) + f(n-2). \tag{6}$$

Proof. Let ρ be any valid permutation. Partition the vertices of W_n into 3 sets $V_1, \{v_c\}$, and V_2 where V_1 is the set of nodes installed before v_c and V_2 is the set of nodes installed after v_c. Let $y = |V_1|$.

There are three cases to analyze. Case 1 is when $1 \leq y < n-2$, case 2 is when $y = n-2$, and case 3 is when $y = n-1$.

Case 1: If $1 \leq y < n-2$, then the cost of installing the blackstart node is $f(0)$ and the cost to install all the other nodes in V_1 is $(y-1)f(1)$. The cost of installing v_c is $f(y)$. Following the same reasoning as in the previous proof, installing the nodes in V_2 is best done by taking them in order for a cost of $(n-y-2)f(2) + f(3)$.

Summing these costs gives:

$$f(0) + (y-1)f(1) + (n-y-2)f(2) + f(3) + f(y) \tag{7}$$

Substituting $y = k+1$ into Eq. 7 gives:

$$f(0) + (k)f(1) + (n-k-3)f(2) + f(3) + f(k+1) \tag{8}$$

Subtracting Eq. 7 with $y = k$ from Eq. 8 gives $f(1) - f(2) + f(k+1) - f(k)$, which is positive since f is a decreasing convex function. Hence Eq. 7 is minimized when $y = 1$ and Eq. 5 gives the minimum cost.

Case 2: If $y = n-2$, then the cost of installing the blackstart node is $f(0)$, the cost to install all the other nodes in V_1 is $(n-3)f(1)$, the cost of installing v_c is $f(n-2)$, and the cost of installing the single node in V_2 is $f(3)$. The total cost is given in Eq. 6.

Case 3: If $y = n-1$, then v_c is the last node installed. The cost of installing the blackstart node is $f(0)$ and the cost to install all the other nodes in V_1 is $(n-3)f(1) + f(2)$. The cost of installing v_c is $f(n-1)$.

Summing these costs gives:

$$f(0) + (n-3)f(1) + f(2) + f(n-1) \quad (9)$$

Subtracting Eq. 6 from Eq. 9 gives $f(2) - f(3) + f(n-1) - f(n-2)$, which is non-negative for any $n \geq 3$ since f is a decreasing convex cost function. As a result, Case 2 is always less expensive than Case 3 if $n \geq 3$. □

3.2 Almost-Complete Networks

The low treewidth of many real-world networks leads to certain structural vulnerabilities [2]. Because it can be less expensive to add edges that are near each other geographically, power networks can evolve to have dense local substructures. As a result, we also considered the NANIP-Blackstart problem on complete and almost-complete networks.

Proposition 6. *Let $K_n = (V, E)$ be a complete graph with $|V| = n$. The cost of installing K_n for any σ is*

$$C_\sigma(K_n) = \sum_{i=0}^{n-1} f(i). \quad (10)$$

Proof. By definition, v_1 is the blackstart node and has cost $f(0)$. Consider node v_{i+1} in the permutation σ. Because K_n is complete, v_{i+1} will be adjacent to each of the i vertices before it in the permutation. Therefore, the cost of installing v_{i+1} is $f(i)$. By induction on i, the cost of any σ is Eq. 10 (Fig. 3). □

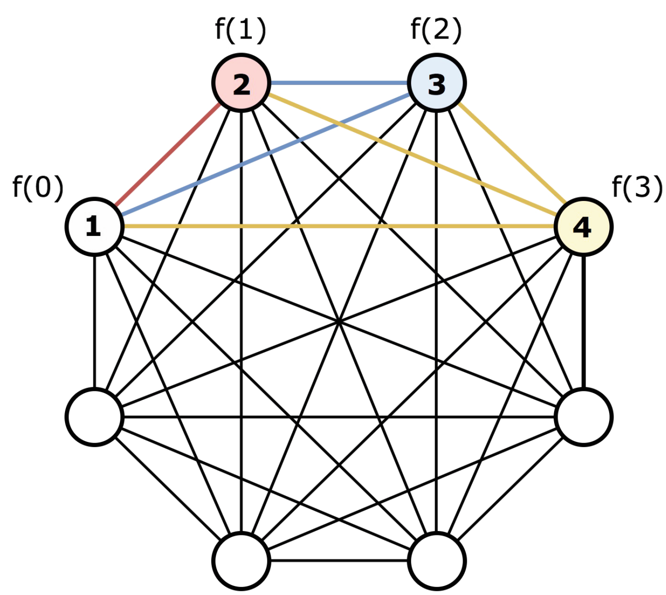

Fig. 3. A K_8 mid-installation to illustrate Proposition 6. Nodes are enumerated according to installation order and labeled by installation cost.

Proposition 7. *Let $S_n = (V, E)$ be a graph with $|V| = n$ and $|E| = \frac{n(n-1)}{2} - 1$. In other words, S_n is a complete graph with one missing edge. Let σ be the minimum cost permutation for installing S_n. The cost of σ is given by:*

$$C_\sigma(S_n) = \Big(\sum_{i=0}^{n-2} f(i)\Big) + f(n-2). \tag{11}$$

In addition, one of the two nodes with degree $n-2$ must be installed last in σ.

Proof. Let u_1 and u_2 be the two vertices of degree $n-2$ in S_n. Without loss of generality assume u_1 appears before u_2 in the minimum cost ordering σ and let u_2 be the k^{th} node in σ.

We know $k \geq 2$ because u_2 must be preceded in σ by u_1. In addition, $k \geq 3$ because even if $v_b = u_1$, u_2 cannot be installed immediately after u_1 since such an ordering would not be valid. Thus, $3 \leq k \leq n$.

The cost of installation from the first node through node $k-1$ is $\sum_{i=0}^{k-2} f(i)$ for the reasons given in Eq. 10. This is because every node including u_1 will be adjacent to every node that has been previously installed.

However, u_2 is not adjacent to u_1, which has already been installed, so u_2 has cost $f(k-2)$. Thus,

$$C_\sigma(S_n) = \Big(\sum_{i=0}^{n-1} f(i)\Big) + f(k-2) - f(k-1). \tag{12}$$

Because f is a decreasing convex cost function, $f(j-2)-f(j-1) \geq f(j-1)-f(j)$ for all j. Repeated substitution shows that the above equation is minimized when $k = n$, giving us Eq. 11. □

Proposition 8. *Let S_{nk} be a complete graph with k removed edges. Furthermore, assume the k removed edges are vertex-disjoint so that the minimum degree of a node in S_{nk} is $n-2$. Let σ be a minimum cost valid permutation for installing S_{nk}. The cost of σ is given by:*

$$C_\sigma(S_{nk}) = \sum_{i=0}^{(n-1)-k} f(i) + \sum_{j=1}^{k} f(n-j). \tag{13}$$

In addition, one vertex from each of the k removed edges must be among the last k nodes in σ (Fig. 4).

Proof. Let (u_i, v_i) be the pair of vertices that define the i^{th} removed edge in S_{nk}. Without loss of generality, always install u_i before v_i, and always install v_i before v_{i+1}. Note that u_i is adjacent to all nodes in $V\backslash\{v_i\}$, and vice versa.

Let $v_1, \ldots, v_k$ be respectively installed as the $m_1^{th}, \ldots, m_k^{th}$ nodes σ, with $m_1 < \cdots < m_k$. If u_i is the p^{th} installed node, then the cost of installing u_i is $f(p-1)$ because u_i is adjacent to all $p-1$ previously installed nodes. However, if

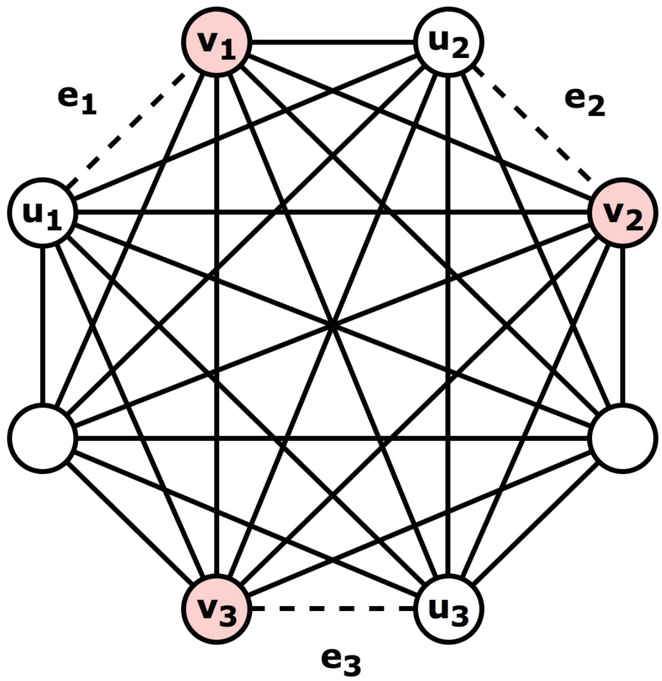

Fig. 4. A K_8 with three vertex-distinct removed edges e_1, e_2, e_3 to illustrate Proposition 8. Without loss of generality, the permutation (v_1, v_2, v_3) defines the last three nodes of all valid, minimum cost permutations for this K_8 network.

v_i is the q^{th} installed node, then the cost of v_i is $f(q-2)$ because v_i is adjacent to all previously installed nodes except for u_i. Then the cost of installing S_n under σ is

$$C_\sigma(S_{nk}) = \sum_{i=0}^{(n-1)} f(i) + \sum_{j=0}^{k} \Big(f(m_j - 2) - f(m_j - 1) \Big).$$

The idea is to begin with the cost of installing a complete graph on n nodes, then adjust the costs of each v_j. Now it remains to determine the values of m_j that minimize $C_\sigma(S_{nk})$.

Because f is a decreasing convex cost function, $f(n-2) - f(n-1) \leq f(m_k - 2) - f(m_k - 1)$ for all possible values of m_k. Thus, $m_k = n$, which means that v_k should be installed last.

As the cost of each $v_{k-(i-1)}$ is fixed, consider the cost of v_{k-i} for $i \in \{1, \ldots, k-1\}$. Given that $m_{k-j} = n - j$ for $0 \leq j \leq i - 1$, observe that

$$f((n-i)-2) - f((n-i)-1) \leq f(m_{k-i} - 2) - f(m_{k-i} - 1),$$

which supports the conclusion that $m_{k-i} = n - i$. Using strong induction on i, we show that $C_\sigma(S_{nk})$ is minimized when the last k nodes in σ are $v_1, \ldots, v_k$. Equation 13 follows. $\square$

4 Experimental Heuristics

We implemented two greedy heuristics to estimate solutions to the NANIP-Blackstart problem.

- The `MaxNeighbors` heuristic repeatedly chooses the uninstalled node that has the largest number of installed neighbors. Since we are using decreasing convex cost functions, this heuristic repeatedly installs the cheapest node next.
- The `MaxPercent` heuristic repeatedly chooses the uninstalled node that has the highest percentage of installed neighbors. This heuristic is motivated by the observation that there may be local subnetworks that would benefit from being installed as a coherent unit.

The codes take a network and a blackstart node as input; the output is an ordering on the vertices and a list of coefficients $a_0, a_i, \ldots a_n$ representing the cost $\sum_{i=0}^{n-1} a_i f(i)$.

The implementation uses a max-priority queue to store the nodes that can be installed; each node in the queue has a priority that is the number of neighboring nodes installed for `MaxNeighbors` or the percentage of neighboring nodes installed for `MaxPercent`. Initially only the blackstart node is in the queue. In each step the code installs the maximum priority node, updates the priorities of its neighbors that are already in the queue, and adds to the queue any neighbors that have not yet been enqueued.

4.1 Experimental Framework

We implemented these two heuristics in Python and tested them on a set of networks from the SuiteSparse Matrix Collection [3]. By way of contrast, previous heuristics described in [5] for the original NANIP problem are tested only on much smaller, randomly generated graphs.

Table 1 details a subset of our test suite.

Table 1. Description of 6 test networks

Name	n	Application area
LeGresley_2508	2508	Power flow analysis of an electrical grid
LeGresley_4908	4908	Power flow analysis of an electrical grid
Powersim	15838	Power simulation matrix
S20PI_n	1182	Power system model
S40PI_n	2182	Power system model
S80PI_n	4182	Power system model

For each network we tested three different blackstart nodes, chosen at random. We then ran the two heuristics on each network and blackstart node pair.

The results were written as a sum of installation costs. For example, in one run on the LeGresley_4908 network, `MaxNeighbors` had a cost of:

$$f(0) + 21f(1) + 2236f(2) + 2227f(3) + 212f(4) + 211f(5) \tag{14}$$

whereas `MaxPercent` had a cost of:

$$f(0) + 18f(1) + 725f(2) + 696f(3) + 637f(4) + 411f(5) + 14f(6) + 3f(7). \tag{15}$$

4.2 Results

Looking just at the equations returned by our heuristics suggests a trend where `MaxPercent` has a longer tail than `MaxNeighbors` in the sense that the equations include $f(k)$ terms for larger k. Because the cost function is a decreasing function, this might seem to suggest that `MaxPercent` should typically outperform `MaxNeighbors`.

However, when we substituted in decreasing convex functions such as $f(k) = 1/k$ or $f(k) = 1/\sqrt{k}$, we found that in all cases `MaxNeighbors` did at least as well as `MaxPercent`. One possibility is that the convexity of the cost function means the advantage of having $f(k)$ terms with larger k decreases as k increases. As a result, it is the terms with smaller k that have more of an effect.

That said, the differences seen were minimal as there were no instances where `MaxNeighbors` outperformed `MaxPercent` by more than 9%. The differences might be greater using other cost functions or on other networks with different characteristics.

5 Conclusion

In this paper we describe an extension to the model for the network installation problem described in [4]. We study a variant with blackstart nodes, which represent locations which can be self-started after a widespread power outage. We show the overall problem remains NP-hard, then prove exact results for certain subclasses of graphs. We also describe two heuristics and analyze their behaviors on a set of real-world networks.

Looking ahead, we are interested in studying what happens as the model better approximates real world power networks. For example, dividing a network into subsystems is often advantageous for restoration since the subsystems can be restored in parallel [1]. Introducing multiple blackstart nodes models exactly this situation. In addition, we are interested in analyzing the inverse question where the network is given and the goal is to identify the nodes at which k blackstart nodes should be located in order to minimize the restart cost.

Acknowledgements. We are grateful to Pomona College for helping fund this work.

References

1. Adibi, M.M., Fink, L.H.: Power system restoration planning. IEEE Trans. Power Syst. **9**(1), 22–28 (1994)
2. Atkins, K., Chen, J., Kumar, V.A., Marathe, A.: The structure of electrical networks: a graph theory-based analysis. Int. J. Crit. Infrastruct. **5**(3), 265–284 (2009)
3. Davis, T.A., Hu, Y.: The University of Florida Sparse Matrix Collection. ACM Trans. Math. Softw. **38**(1), 1:1–1:25 (2011). https://doi.org/10.1145/2049662.2049663, http://doi.acm.org/10.1145/2049662.2049663
4. Gutfraind, A., Bradonjić, M., Novikoff, T.: Modelling the neighbour aid phenomenon for installing costly complex networks. J. Complex Netw. (2014). https://doi.org/10.1093/comnet/cnu033
5. Gutfraind, A., Kun, J., Lelkes, Á.D., Reyzin, L.: Network installation and recovery: approximation lower bounds and faster exact formulations. CoRR abs/1411.3640 (2014). http://arxiv.org/abs/1411.3640
6. Ketabi, A., Feuillet, R.: Ant colony search algorithm for optimal generators startup during power system restoration. Math. Probl. Eng. **2010**, 1–11 (2010). https://doi.org/10.1155/2010/906935
7. Lin, Z.Z., Wen, F.S., Chung, C.Y., Wong, K.P., Zhou, H.: Division algorithm and interconnection strategy of restoration subsystems based on complex network theory. IET Gener. Trans. Distrib. **5**(6), 674–683 (2011). https://doi.org/10.1049/iet-gtd.2010.0586
8. Rokrok, E., Shafie-khah, M., Siano, P., Catalão, J.P.S.: A decentralized multi-agent-based approach for low voltage microgrid restoration. Energies **10**(10) (2017). https://doi.org/10.3390/en10101491, http://www.mdpi.com/1996-1073/10/10/1491
9. Van Hentenryck, P., Coffrin, C.: Transmission system repair and restoration. Math. Programm. **151**(1), 347–373 (2015). https://doi.org/10.1007/s10107-015-0887-0
10. Wikipedia: North American Electric Reliability Corporation. https://en.wikipedia.org/wiki/North_American_Electric_Reliability_Corporation. Accessed 27 July 2018