

Graph partitioning of power grids considering electricity sharing

Tomoya Tanjo, Kazuhiro Minami, Hiroshi Maruyama*

Institute of Statistical Mathematics, 10-3 Midori-cho, Tachikawa, Tokyo 190-8562, Japan

Abstract

After the 2011 Great East Japan Earthquake in Japan, which subsequently shut down the nuclear power plant in Fukushima, many researchers who foresee such future threats have been proposing decentralized electricity management systems based on renewable green energy. In this paper, we study a graph partitioning problem of power grids assuming that each microgrid can transfer its electricity surplus to other clusters in a peer-to-peer way. We extend an existing graph partitioning algorithm such that it can check whether a set of clusters are self-sufficient after electricity sharing. Our experiments show that our method can reduce the cost of constructing a microgrid system and that it can produce a resilient graph partition that accommodates time-varying electricity surplus over the year.

Keywords: Power grids, distributed electricity power management system, graph partitioning, maximum flow

1. Introduction

After the 2011 Great East Japan Earthquake in Japan, which subsequently shut down the nuclear power plant in Fukushima, we are aware of a significant risk of depending on highly centralized electricity power resources. Many researchers who foresee such future threats thus have been proposing decentralized electricity management systems based on renewable green energy.

We consider a distributed electricity system in which solar photovoltaic panels (PVs) are deployed for each microgrid. In such a system, electrical grids are partitioned into several microgrid that are self-sufficient in the sense that their electricity supply and demand are balanced. Our objective is to find a partition that minimizes the cost of constructing a set of microgrids while making that partition resilient in the sense that each microgrid is well-balanced under time-varying supply and demand over the year. However, it is not trivial to find a partition that minimizes the cost while each cluster is self-sufficient because there is a trade-off between the size of microgrids and the difficulty of balancing their surplus. That is, the larger the size of microgrids, the easier to balance their surplus though the construction cost becomes higher at the same time.

To overcome this issue, we propose a dynamic electricity management scheme in which microgrids exchange their electricity in a peer-to-peer way. We consider that a microgrid is self-sufficient if its surplus is under a given threshold after canceling out its surplus or shortage. Such a relaxed notion of electricity balance enables us to divide a grid into smaller ones further while satisfying the balancing requirement. We model an electrical power grid as a weighted graph in which every node and edge is associated with a weight value and study a graph partitioning problem in which each subgraph corresponds to a microgrid.

In this paper, we develop a new graph partition algorithm that guarantees that each cluster can balance its electricity surplus under a given threshold through electricity exchange with other peers. We combine the Recursive Coordinate Geometric Bisection (RCB) algorithm, which considers vertical or horizontal

* Manuscript received October 25, 2015; revised April 5, 2016.

Corresponding author. Tel.: +81-50-5533-8444; *E-mail address*: tanjo@ism.ac.jp.

doi: 10.12720/sgce.5.2.112-120

partitions [1], with a new verification algorithm for electricity balancing. The verification algorithm determines how clusters exchange electricity surplus with each other to distribute electricity surplus evenly.

However, the verification process involves high computational cost of solving the combinatorial optimization problem. We thus develop an approximate algorithm based on the maximum flow algorithm by Edmonds and Karp [2]. We apply the algorithm to an aggregated graph constructed from the partitioned subgraphs. The basic idea is to compute the maximum flow from clusters of electricity surplus to those of shortage. We also extend the original flow algorithm such that it computes a well-balanced augmented path among nodes by iteratively apply the algorithm while incrementing edge capacity.

We apply our graph partitioning algorithm to a synthesized surplus data for Yokohama city in Japan and evaluate how our proposed scheme reduce the construction cost of microgrids and how resilient computed partitions are in the presence of time-varying electricity demand and supply over the year.

The rest of this paper is organized as follows. In Section 2, we formulate our problem as a graph clustering problem. In Section 3, we describe the algorithm for solving the graph clustering problem. In Section 4, we show how to estimate the demand and supply in Yokohama city in Japan. In Section 5, we show experimental results with the synthesized data in Section 4. We discuss related works for the graph clustering problem in Section 6 and conclude in Section 7.

2. Graph Clustering Problem

In this section, we formulate a graph clustering problem in the context of power grids. We model an electrical grid as a graph $G = (V, E)$ where each node $v \in V$ represents geographical area and each edge $e_{ij} = (v_i, v_j) \in E$ represents a transmission line between two nodes v_i and v_j . We denote by $w(v)$ the weight of a node v corresponding to the electricity surplus in node v 's area. We denote by $c(e_{ij})$ the weight of an edge e_{ij} corresponding to the transmission capacity between v_i and v_j . We denote by $V(G)$ and $E(G)$ a finite set of nodes in graph G and a finite set of edges in graph G , respectively. A partition $P = \{G_1, G_2, \dots, G_n\}$ of graph G is a finite set of subgraphs where $\bigcup_{i \in \{1, \dots, n\}} V(G_i) = V$ and $V(G_i) \cap V(G_j) = \emptyset$ for all $i, j \in \{1, \dots, n\}, i \neq j$. We extend the notation of the ‘‘weight’’ to an aggregate graph consisting of subgraphs and edges between different subgraphs. That is, $w(G_i) = \sum_{v \in V(G_i)} w(v)$ is the total amount of surplus in a subgraph G_i and $c(G_i, G_j) = \sum_{v_i \in V(G_i), v_j \in V(G_j), e_{ij} \in E} c(e_{ij})$ is the total amount of transmission capacity between different subgraphs G_i and G_j .

2.1. Constraints for graph clustering problem

Briefly speaking, a graph clustering problem is a problem to find a partition $P = \{G_1, G_2, \dots, G_n\}$ of a given graph G that satisfies the following conditions:

- P minimizes the cost C for building clusters of microgrids by deploying transmission lines, and
- Each cluster $G_i \in P$ is self-sufficient.

In this section, we explain these conditions formulating our clustering problem.

2.1.1. Infrastructure cost of building clusters

In our scheme, we assume that every pair of two microgrids within a single cluster is connected and that two neighboring microgrids that belong to different clusters are also connected. To consider the cost for constructing such clusters, we define the cost C as follows:

$$C = \alpha \sum_{i=1}^n |V(G_i)|^2 + \beta |E(G) \setminus \bigcup_{i=1}^n E(G_i)|$$

where α is a coefficient of the cost for transmission lines within a single cluster and β is a coefficient of

the cost for transmission lines between different clusters. The first term represents an infrastructure cost for connecting each pair of microgrids within a single cluster and the second term represents that for connecting each pair of microgrids between different clusters.

2.1.2. Balancing constraint for the clusters

We say that a given partition P is self-sufficient if an absolute value of average surplus per unit area of G_i is less than a given threshold k for each subgraph $G_i \in P$ after electricity sharing. More precisely, we say that a partition P is self-sufficient if the following holds:

$$\max_{G_i \in P} |as'(G_i)| \leq k \quad (1)$$

such that

$$as'(G_i) = \frac{w(G_i) + \sum_{G_j \in P} f(G_j, G_i)}{|V(G_i)|}$$

where k is a given threshold, $as': P \rightarrow \mathbb{R}$ represents an average surplus of G_i after balancing, $f: P \times P \rightarrow \mathbb{R}$ is the flow function such that $f(G_j, G_i)$ outputs the flow from the cluster G_j to G_i . Here \mathbb{R} is a set of real numbers. Note that the flow $f(G_j, G_i)$ should not exceed the transmission capacity $c(G_j, G_i)$ and that $f(G_j, G_i) = -f(G_i, G_j)$ should hold for each pair of clusters G_i and G_j . We call $MUV(P) = \max_{G_i \in P} |as'(G_i)|$ the **Maximum Unbalanced Value (MUV)** of P and say that G_i is a **Maximum Unbalanced Subgraph (MUS)** of P if $MUV(P) = |as'(G_i)|$.

We also consider the case without electricity sharing. Then, as' in (1) is replaced with the following function as :

$$as(G_i) = \frac{w(G_i)}{|V(G_i)|}$$

2.2. Graph clustering problem

Finally we formulate the graph clustering problem and its subproblem as follows [3]:

Definition 1 (Partition balancing problem): Let P be a partition and $k \in \mathbb{R}$ ($k \geq 0$) be a threshold. A **Partition balancing problem** is a problem of verifying whether there exists a flow function f such that $MUV(P) \leq k$ holds under f .

Definition 3 (Graph clustering problem): Let G be a graph and $k \in \mathbb{R}$ ($k \geq 0$) be a threshold. A **graph clustering problem** is a problem of finding a partition P of graph G in which:

1. $C(P) \leq C(P')$ holds for any partition P' that satisfies the constraint (2), and
2. There is a flow function f , which is a solution for the partition balancing problem for threshold k .

3. Solving Graph Clustering Problem

In this section, we explain how to solve the graph clustering problem in Section 2. We first introduce the Recursive Coordinate Geometric Bisection (RCB) algorithm and then describe the weighted graph balancing algorithm for solving the partition balancing problem.

3.1. Recursive Coordinate Geometric Bisection (RCB) algorithm

The RCB algorithm is a simple graph partitioning algorithm that divides clusters either vertically or horizontally [1]. Here is an overview of the RCB algorithm:

1. Divide a graph into two subgraphs of almost equal weights with either a horizontal or a vertical partition.
2. For each subgraph, repeat step 1 to divide the subgraph into smaller ones in a recursive way.

The RCB algorithm is an approximate algorithm but it works efficiently in practice.

3.2. The RCB algorithm with electricity sharing

We combine a modified version of the RCB algorithm with the weighted balancing algorithm below. In the modified RCB algorithm, we consider multiple candidate divisions of a graph and choose the best one minimizing the infrastructure cost. Also, that division must have a solution flow of the partition balancing problem. The weighted graph balancing algorithm is used to reduce the number of candidates for divisions.

3.2.1. Weighted graph balancing algorithm

The weighted balancing algorithm consists of two parts: (i) improving MUV of a given partition P and (ii) comparing the improved MUV for a given threshold k . In this section, we explain the former part of the algorithm omitting the latter simple task.

The main idea to improve MUV is to make a flow from the MUS in P to other subgraphs repeatedly until $MUV(P) \leq k$ holds or there is no such flow that improves MUV. To find such flow, we use the maximum flow algorithm by Edmonds and Karp [2]. The maximum flow problem involves finding a flow from a source node to a sink node that is maximum.

To make a given partition applicable to the maximum flow algorithm, we introduce an **aggregated graph** $G^a = (V^a, E^a)$ in which each aggregated node $v_i^a \in V^a$ represents a cluster $G_i \in P$ and each aggregated edge $e_{ij}^a = (v_i^a, v_j^a) \in E^a$ represents a set of edges between clusters G_i and G_j . Each aggregated node $v_i^a \in V^a$ has a weight $w(v_i^a) = w(G_i)$ which represents the total amount of surplus in G_i and each aggregated edge $e_{ij}^a \in E^a$ has a weight $c(e_{ij}^a) = c(G_i, G_j)$ which represents the total amount of transmission capacity between G_i and G_j . We also introduce a source node s and a sink node t to V^a . We add edges $e_{si}^a = (s, v_i^a)$ for nodes with positive surplus and edges $e_{it}^a = (v_i^a, t)$ for nodes with negative surplus. Each edge e_{si}^a has a weight $c(e_{si}^a) = |w(v_i^a)|$ and each e_{it}^a has a capacity $c(e_{it}^a) = |w(v_i^a)|$.

To simplify the discussion, we extend the concept of the MUS for an aggregated graph as follows:

- We say that $v_i^a \in V^a$ is the **Most Unbalanced Node (MUN)** if the corresponding subgraph G_i is MUS.
- We say that $e_{sj}^a \in E^a$ is the **Most Unbalanced Edge (MUE)** if $v_j^a \in V^a$ is MUN and that $e_{it}^a \in E^a$ is also MUE if $v_i^a \in V^a$ is MUN.

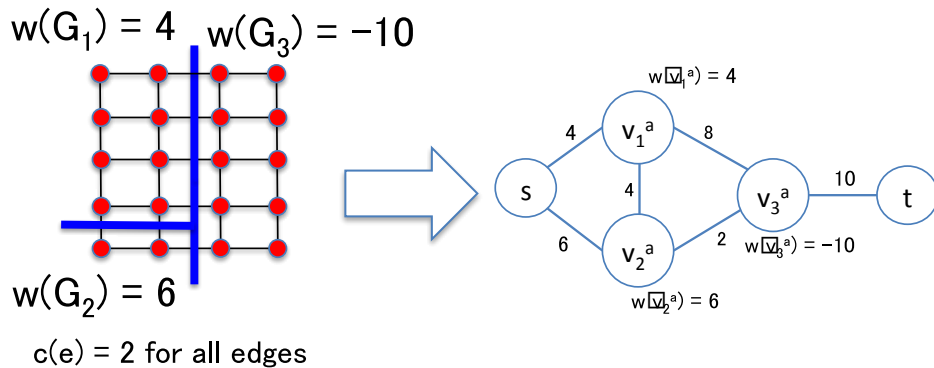


Fig. 1. Reduction from a partition to the aggregated graph. Each cluster G_i is aggregated into the node v_i^a and edges between different clusters G_i and G_j is aggregated into the edge $e_{ij}^a = (v_i^a, v_j^a)$. The nodes s and t are virtual nodes that represent source node and sink node respectively.

Fig. 1 shows an example reduction from a partition into the corresponding aggregated graph. The node v_2^a is the MUN because corresponding subgraph G_2 has the largest average surplus and e_{s2} is the MUE.

By considering an aggregated graph, we can apply the max flow algorithm to find a path that may improve the MUV. However, just applying the max flow algorithm is not enough. First, it is not guaranteed that the MUV is improved for each application of the algorithm because the max flow algorithm may find a flow that does not pass through the MUN. To prevent such a situation, we modify the capacities of non MUEs that are connected with either the source or sink node. Second, making a flow with too much capacity may block other flows that can improve the MUV in the further iteration. We gradually increase edge capacities before each application of the max flow algorithm.

Here is an overview of the weighted balanced algorithm.

1. Construct an aggregated graph from a partition P .
2. Increase capacities for edges e_{ij}^a ($i \neq s \wedge j \neq t$) to make a flow.
3. Modify the capacities of non MUEs that are connected with either the sink or source node as follows:
 - Let $v_{mun} \in V^a$ be the MUN. If $w(v_{mun}) > 0$, set the capacities of all non MUEs e_{sj}^a to 0. Otherwise set the capacities of all non MUEs e_{it}^a to 0.
4. Compute the max flow f from the source node to the sink node.
5. If there is no such flow f , this algorithm returns *false* since P cannot be balanced.
6. Pick up a path p from the source node to the sink node from the flow f to limit the total amount of flow, and update the surplus for all nodes according to p .
7. This algorithm returns *true* (i.e. P can be balanced) if the aggregated graph G is self-sufficient under the flow function.
8. Otherwise, go to step 2 and continue to make a further flow to improve the MUV.

4. Estimation of Electricity Affordability for Yokohama City

In this section, we will explain how we estimate the electricity supply and demand for every 250m region in Yokohama city. We assume that the electricity is supplied from PVs which is installed on the roof for each house, and that it is stored in electric vehicles (EVs) not in use (See [4] for more details).

First, we estimate the number of cars not in use to estimate the storage capacity of EVs. It is estimated by the following steps:

- Determine the number of cars not in use in each unit area called cho-cho-moku by simulating the daily movements of peoples in Yokohama city using the agent-based transport simulator MATSim (<http://matsim.org/>). We use the Origin-Destination (OD) trip data (source: the Fourth Person Trip Survey in Tokyo Metropolitan Area) and the road-network data (source: the National Digital Road Map Database) whose attributes include road capacity, road width classification, link length, number of lanes, and travel speed. Note that the OD trip is available only at the distinct-wise level.
- Convert the distinct-wise estimation into those at the 250m grid level. We apply the standard geostatistical method of Kyriakidis [5] for converting the distinct-wise estimation.

Second, we estimate the electricity surplus by subtracting electricity household demand from electricity PV supply. The electricity supply $PV_{i,m}$ (kWh/h) in each grid in each month is estimated by using the following formula which is used in [4] and [6]:

$$PV_{i,m} = I \times \tau \times roof_i^{PV} \times \eta_{pc} \times K_{m,pt} \times T$$

where i is an index of 250m grids, m is an index of months, I is the total solar irradiance ($kWh/m^2/h$) calculated by MTPV-2 database [7], τ is the array conversion efficiency ($= 0.1$), $roof_i^{PV}$ is the installation area in i -th grid (m^2), η_{pc} is the efficiency of power conditioner ($= 0.95$), $K_{m,pt}$ is the temperature correction coefficient set for each month m (e.g., May: 0.92; August: 1.00), and T is the performance ratio ($= 0.89$). $roof_i^{PV}$ is calculated, following [8], as

$$roof_i^{PV} = \frac{B_i \times l \times 1}{\cos(\psi)}$$

where B_i is the building area in i -th grid, l is the ratio of possible installation area on a roof ($= 0.3$), and ψ is the optimal angle of inclination ($= 30^\circ$). Electricity demand in each month $D_{i,m}$ is estimated as

$$D_{i,m} = F_i \times w_m$$

where F_i is the total floor area in i -th grid, and w_m is the unit electricity demand in each month.

Finally, we can calculate the electricity affordability for each grid for each month by subtracting the electricity surplus from the storage capacity. Fig. 2 shows estimated storage affordability in February and July. As shown in Fig. 2, the affordability can vary from month to month.

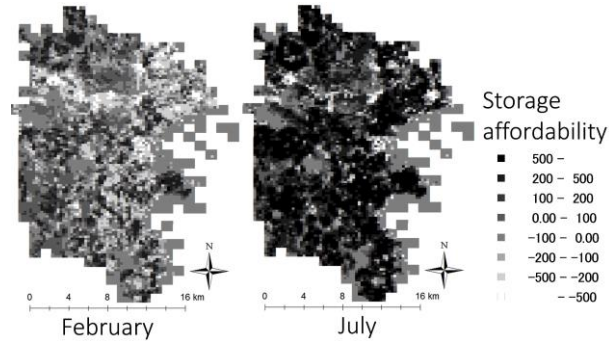


Fig. 2. Electricity affordability in February and July in Yokohama, Japan.

5. Evaluation

To evaluate the effectiveness of electricity sharing, we apply our graph partitioning algorithm to the estimated surplus data in Yokohama city in Japan that is described in the previous section. In this section, we evaluate our proposal with respect to:

- The number of clusters and the construction cost, and
- The metrics of resilience as described below.

In general, the partition obtained from the surplus data in a certain month may not be feasible for the surplus in other months. If it is feasible for other months by electricity sharing, we say that that partition is resilient even in the situation when the electricity surplus varies over the year. Therefore we use the minimized MUV as a metric of the resilience for partitions because it represents how balanced the whole clusters are after electricity sharing.

5.1. Comparison of the number of divisions and the cost ratio

First, we compare the number of clusters and the cost ratios after applying our method and the original RCB method to the average estimated surplus data over the year. We use $(\alpha, \beta) = (1, 1)$ and $(1, 10)$ for the different combination of coefficients and varying the threshold k within 100, 150, 200, 250, 300, 350 and 400.

Fig. 3 shows the number of divisions resulted by RCB with and without electricity sharing. The x -axis represents thresholds and y -axis represents the number of divisions. The solid line represents the number of divisions for our method and the dotted line represents the number of divisions for the RCB algorithm. Fig. 4 shows the cost ratio of our method compared with the RCB algorithm for each threshold. The x -axis represents thresholds and y -axis represents the cost ratios. For all thresholds, our method can achieve greater number of divisions with less cost ratio than the original RCB algorithm. In Fig. 3, our method obtains the cluster whose size is four times larger than that with the original RCB algorithm when threshold is 400. In Fig. 4, our method reduces 70% of the construction cost compared with the RCB algorithm when thresholds are 200 and 250. It is because the construction cost is highly dominated by the number of divisions especially in the case the number of divisions is not large enough.

5.2. Comparison of the minimized MUV for each month

We next compare the minimized MUV for each month by using the estimated data during the year. We calculated the minimized MUV by using:

- The partition obtained from the average surplus during the year by our partitioning algorithm, and
- The surplus data in Yokohama city for each month.

Fig. 5 shows the minimized MUV by using RCB with and without our scheme. The x -axis represents months and y -axis represents the minimized MUV for each month. The solid line represents the result for our method and the dotted line represents the result for the RCB algorithm. Our method achieves less MUV than the original RCB for all months. This result shows that our scheme achieves a resilient partition under the presence of time-varying electricity surplus over the year.

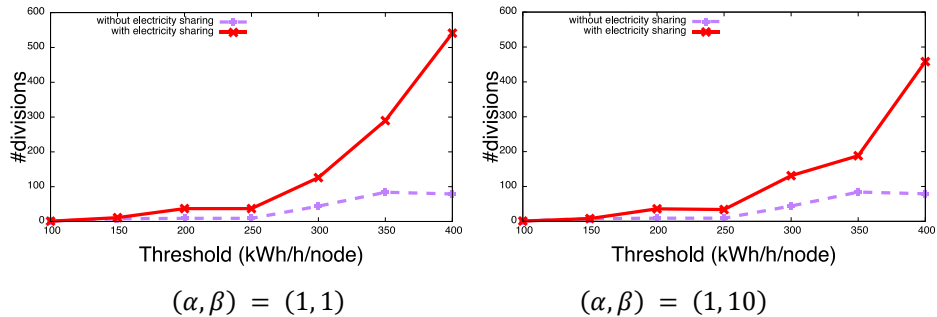


Fig. 3. Comparison of partition results with and without electricity sharing. We use the different combination of coefficients α and β varying the threshold from 100 to 400. The solid line represents the result for our method and the dotted line represents the result for the RCB algorithm.

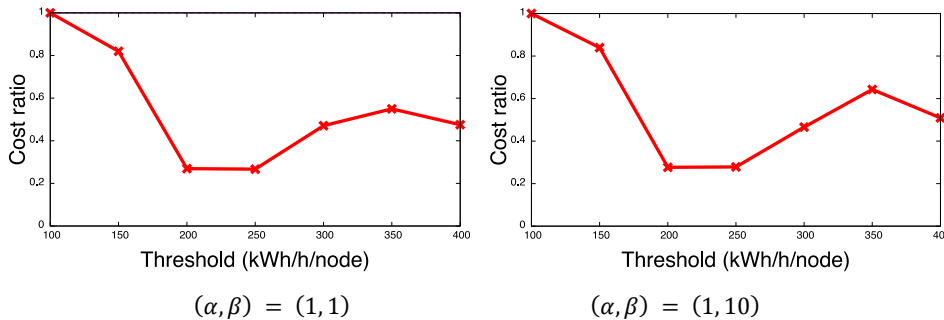


Fig. 4. Cost ratios of RCB with electricity sharing compared with the original RCB. We use the different combination of coefficients α and β varying the threshold from 100 to 400.

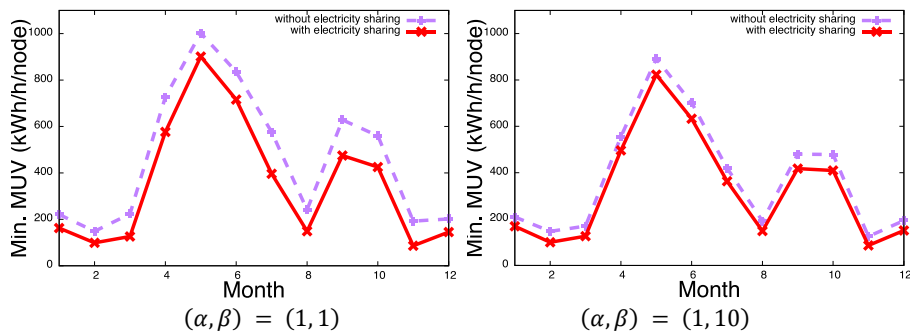


Fig. 5. Comparison of the minimized MUV with and without electricity sharing for each month. The solid line represents the result for our method and the dotted line represents the result for the RCB algorithm.

6. Related Works

The weighted balancing algorithm is to improve the MUV by determine the flow from clusters with surplus to those with shortage. This problem can be considered as a dynamic resource allocation problem for evenly distributing resources to each user.

Geng *et al.* [9] propose a “smart parking” system that suggests a parking spot for each driver to make an allocation for as many users as possible while minimizing user cost that reflects the distance from the parking spot to the user’s location for each time step. They formulate the requirements for each decision point as a Mixed-Integer Linear Programming (MILP) problem that is a problem of finding a solution of the equations in which all the variables are restricted in real numbers or integers. This allocation problem is a kind of matching problem between users and parking spots while our problem needs to allocate the surplus, which is numerical data, for all clusters.

The Maximum Concurrent Flow Problem (MCFP) [10] is a similar mathematical problem to our partition balancing problem. In MCFP, an undirected graph, a set of pairs of nodes to make a flow, and the demands for each pair of nodes to be transferred are given assuming that the all edges have the same capacity. The objective of MCFP is to maximize the ratio of the flow relative to the demand. In [10], Shahrokhi and Matula formulate MCFP as a linear programming problem and show that it can be approximately solvable in polynomial time. Although MCFP is similar to our problem concerning a balanced flow in a given graph, our problem needs to support edges with different capacities in general. Also, we are interested in the weights of nodes rather than edges after electricity sharing.

7. Conclusion

We study a graph partitioning problem of a power grid to find a optimal partition that minimizes the cost for constructing the transmission lines. In this paper, we consider a graph partition in which clusters can exchange electricity in a peer-to-peer way to reduce the construction cost. We develop a graph partition algorithm that verifies whether a resulting partition can be balanced through electricity exchanging between clusters.

Through the experiments with the synthesized surplus data in Yokohama city in Japan, our method outperformed the RCB algorithm that does not consider the electricity sharing in terms of the number of divisions and the cost ratios. Furthermore, our method produce a resilient partition under the presence of varying surpluses from month to month. As future work, we plan to evaluate our method with more comprehensive datasets.

References

- [1] Berger MJ, Bokhari SH. A partitioning strategy for nonuniform problems on multiprocessors. *IEEE Transactions on Computers*, 1987; 100(5):570-580.
- [2] Edmonds J, Karp RM. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 1972; 19(2):248-264.
- [3] Minami K, Tanjo T, Arizumi N, Maruyama H. flexible graph partitioning of power grids with peer-to-peer electricity exchange. In: *Proc. of the 7th International Conference on Information and Automation for Sustainability*, 2014:1-6.
- [4] Yamagata Y, Seya H. Community-based resilient electricity sharing: Optimal spatial clustering. In: *Proc. of the 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, 2013:1-8.
- [5] Kyriakidis PC. A geostatistical framework for area-to-point spatial interpolation. *Geographical Analysis*, 2004; 36(3):259-289.
- [6] Yokoi T, Yamamoto Y, Tokai A, Morioka T. Development of decision support system to integrate block renewal and energy planning towards low-carbon city (in Japanese). *Journal of Japan Society of Civil Engineers*, 2010; 66(1):17-34.
- [7] Itagaki A, Okamura H, Yamada M. Preparation of meteorological data set throughout Japan for suitable design of PV systems. In: *Proc. of the 3rd World Conference on Photovoltaic Energy Conversion*, 2003:2074-2077.
- [8] Yamagata Y, Seya H. Integrated modelling for a future smart city: toward efficient CO₂ management of EV transport using PV systems. In: *Proc. of the 19th ITS World Conference*, 2012:2-26.

- [9] Geng Y, Cassandras CG. Dynamic resource allocation in urban settings: A “smart parking” approach. In: *Proc. of the IEEE International Symposium on Computer-Aided Control System Design*, 2011:1-6.
- [10] Shahrokhi F, Matula DW. The maximum concurrent flow problem. *Journal of the ACM*, 1990; 37(2):318-334.