Efficient Demand Assignment in Multi-Connected Microgrids

Kirill Kogan
University of Waterloo
kirill.kogan@gmail.com

Sergey Nikolenko
Steklov Mathematical Institute
sergey@logic.pdmi.ras.ru

Srinivasan Keshav
University of Waterloo
keshav@uwaterloo.ca

Alejandro Lopez-Ortiz
University of Waterloo
alopez-o@uwaterloo.ca

ABSTRACT

With the proliferation of distributed generation, an electrical load can be satisfied either by a centralized generator or by local/nearby distributed generators. Given a set of resource demands in a collection of geographically co-located microgrids that are connected to the central grid and also potentially to each other, each such demand characterized by a power level and a duration, we study algorithms that allocate generation resources to the set of demands by configuring switched paths from sources to loads.

1. INTRODUCTION AND MOTIVATION

In recent years, electricity generation has been rapidly becoming more diverse: power is generated today not only from large, capital-intensive plants but also from numerous smaller-capacity resources including solar panels, wind turbines, and diesel gensets. This proliferation has made it possible for electric demands to be met with local generation, reducing distribution losses and simultaneously increasing energy security. Increasingly, sets of loads can rely nearly entirely on local generation resources, forming a *microgrid*, with access to the central grid used only as a backup.

We anticipate that in the future geographically-close microgrids will opportunistically form connections with each other to increase reliability. This would allow, for instance, a set of apartment complexes to augment their own diesel gensets with shared solar generation from a nearby office complex on weekends. This is a natural recapitulation of the self-organizing process by which electricity grids were formed in the first place, before centralized generation essentially eliminated micro-generation a century ago.

The focus of our work is on efficient demand satisfaction in the context of multi-connected microgrids, where a demand can be met by different generation resources: local, nearby, or on a regional grid. Specifically, we are concerned with minimizing the delay in satisfying a set of resource demands (assuming that these demands can be temporarily delayed but non-preemptive (*elastic* [1]). An additional concern is to minimize the number of switching operations needed to meet a particular set of demands because the wear and tear induced by each re-organization of switches eventually leads to equipment failure.

With some simplifying assumptions, we find that the abstract problem of meeting time-limited loads (i.e., each load requires a certain power for a certain time) from a set of generation resources using a set of distributed switches is similar to the problem of assigning packets of a certain length arriving at the input ports of a rearrangable optical switch to a set of output ports. Each packet corresponds to a demand, each input port to a generation resource, and each output port to a load Given a set of demands, the minimum make-span assignment of these demands to loads is also the assignment that minimizes total delay, while the minimum set of configurations also minimizes switch wear and tear. We therefore extend past work in demand assignments in rearrangable optical switches [2] to compute lower and upper bounds on the minimum number of rearrangements needed to meet a set of demands.

e-Energy'13, May 21–24, 2013, Berkeley, California, USA.
ACM 978-1-4503-2052-8/13/05.

2. PROBLEM STATEMENT AND NOTATION

We make two simplifying assumptions in our work. First, we assume that all generators have the same cost of power production. Second, we assume that distribution losses are negligible.

We model a set of multi-connected microgrids with a switching system $(I, \mathcal{D})$ that consists of a set of inputs I (the generators) with port capacities c_i (the nominal power that they currently generate) and a set of demands $\mathcal{D}$ (elastic electrical loads) that are to be scheduled; a demand d is characterized by its length $l(d)$ (how long the demand lasts), width $w(d)$ (the power level of the demand), and a *load balancing vector* $v(d)$ that contains the set of input ports available to process d (i.e., the set of generators that can feasibly meet this demand).

Time is discrete; we denote by L and l respectively the longest and shortest length in time slots among all given demands. If a demand d is assigned to input i at time t, d uses $w(d)$ bandwidth of port i during the time interval $[t, t + l(d) - 1]$. A schedule P is a sequence of configurations, where each configuration is a partial mapping of the demands to the inputs that has to satisfy constraints imposed by port capacities and load balancing vectors. The length of a configuration C is defined by the longest demand that is scheduled during C. There is a non-negligible penalty, called *configuration overhead*, of V time slots between two consecutive configurations. Our goal is to satisfy loads in $\mathcal{D}$ as fast as possible. Note that the value of V can impact a scheduling decision. Therefore, we consider an additional objective: to minimize the total number of configurations.

The practically interesting case is one where each demand can be met from exactly two input ports, and one of them is shared among all demands. This situation arises naturally if local distribution networks, each covering its own region, are supplemented by a central grid. In this case, the problem is to reuse the central grid input in the most efficient manner in order to optimize either makespan or the number of configurations.

Algorithm 1 GREEDYSCHEDULINGPOLICY($\mathcal{D}$, I)

$D := \mathcal{D}, \mathcal{C} := \emptyset$.
while $D \neq \emptyset$ **do**
  start new configuration $C := \emptyset, I' := I$;
  while there are available ports and demands **do**
    $(i, d) :=$ CHOOSEPORTDEMAND(D, I');
    $C := C \cup \{(i, d)\}, c_i' := c_i' - w(d), D := D \setminus \{d\}$;
  end while
  $\mathcal{C} := \mathcal{C} \cup \{C\}, D := D \setminus \{d \mid d \in C\}$.
end while
Return $\mathcal{C}$.

Algorithm 2 SG

function CHOOSEPORTDEMAND$(\{\mathcal{D}_i\}_i, I)$
  for $i := 2$ **to** I **do**
    if $c_i > w(d)$ for some $d \in \mathcal{D}_i$ **then**
      return $(i,$ CHOOSEDEMAND$(\mathcal{D}_i, c_i))$;
    end if
  end for
  Return $(1,$ CHOOSEFIRST$(\{\mathcal{D}_i\}_i, I))$.
end function

Algorithm 3 SLD

function CHOOSEDEMAND$(\mathcal{D}_i, c_i)$
  Return $\arg\max \{l(d) \mid d \in \mathcal{D}_i\}$.
end function
function CHOOSEFIRST$(\mathcal{D} = \{\mathcal{D}_i\}_i, I)$
  $D := \{d \in \mathcal{D} \mid l(d) = \max_{d'} l(d')\}$.
  Return $\arg\max_{d \in D} \{k(\mathcal{D}_i) \mid d \in \mathcal{D}_i\}$.
end function

Algorithm 4 SLP

function CHOOSEDEMAND$(\mathcal{D}_i, c_i)$
  Return $\arg\max \{l(d) \mid d \in \mathcal{D}_i\}$.
end function
function CHOOSEFIRST$(\mathcal{D} = \{\mathcal{D}_i\}_i, I)$
  $I' := \{i \mid k(\mathcal{D}_i) = \max_j k(\mathcal{D}_j)\}$.
  Return $\arg\max \{l(d) \mid d \in \mathcal{D}_i, i \in I'\}$.
end function

3. ANALYTICAL STUDY

In this work we compare the performance of a class of greedy scheduling policies with those of the optimal policy in the worst case. Formally speaking, we say that an algorithm A has approximation ratio α (is α-approximate) with respect to some objective function (we assume that the objective is to be minimized) if for every input $(\mathcal{D}, I)$, A produces a schedule with objective function value at most α times greater than the optimal objective function value.

We concentrate our efforts on simple policies, amenable to efficient implementation since such policies can also scale well. The general algorithm describing such a policy is presented in Algorithm 1. Given a set of demands $\mathcal{D}$ and a set of input ports I with capacities c_i, $i \in I$, a greedy scheduling policy creates each consecutive configuration by greedily choosing the next demand to process. Once there are no more ports (i.e., generators) available to meet the existing demands, so that all relevant capacities have been exhausted, the current configuration is finalized and a new configuration begins.

The heart of Algorithm 1 is the CHOOSEPORTDEMAND procedure that takes current state (remaining demands and leftover capacities) as input and outputs the input-demand pair (i, d) for the next assignment. Various algorithms considered in this work differ from each other precisely in their CHOOSEPORTDEMAND procedures.

The obvious general algorithm for this case is SG, which stands for "Shared Greedy" (Algorithm 2): that first fills the capacities of every port except the first, then chooses demands for the first port.

Different algorithms may differ in choosing a demand for a single port (CHOOSEDEMAND procedure) and in choosing which demand to send to the first port for extra processing (CHOOSEFIRST procedure).

ALG	Unit capacities		Unit widths		General case	
	Lower	Upper	Lower	Upper	Lower	Upper
SG	1	$3/2$	$5/3$	2	-	4
SLD	$3/2 - 2^{-(I-1)}$	$3/2$	$5/3$	2	-	4
SLP	1	1	1	1	-	2

Table 1: Results summary for minimizing the number of configurations.

The basic tradeoff here is the balance between minimizing the number of configurations and minimizing their total length (duration). In this regard, we define two algorithms from the SG family: SLD ("Shared Longest Demand", Algorithm 3) and SLP ("Shared Longest Port", Algorithm 4). SLD chooses the longest available demand for the current configuration; for the CHOOSEDEMAND procedure it does not matter which one, for the CHOOSEFIRST procedure SLD splits ties with the largest port heuristic (maximal $k(\mathcal{D}_i)$). SLP, on the other hand, chooses for the CHOOSEFIRST procedure a demand from the port with maximal normalized load $k(\mathcal{D}_i)$; for splitting ties and CHOOSEDEMAND, it uses the longest demand heuristic.

The interplay of the following four parameters define the behaviour of a scheduling policy: (i) input port capacities, (ii) demand lengths, (iii) demand widths, and (iv) "normalized load". Clearly, all parameters that have an impact on the number of configurations also affect the schedule length objective. The notion of "normalized load" has significant impact on the both objectives. Observe that demand length has no impact on the number of configurations but can have significant influence on the total length of schedule as $\frac{L}{l}$ grows. The impact of input capacity constraint is interesting even for unit-sized demand widths. There is a correlation between the number of configurations and utilization of input capacities. The underlying problem is obviously NP-hard: even optimal scheduling with a single port encompasses the knapsack problem. During our study we carefully explore the impact of each one of the considered parameters on the performance of scheduling policies. The main contribution of this paper is an analysis of characteristics that should be implemented by an "ideal" policy. A short summary of our theoretical results is shown in Table 1 (lower bounds obviously propagate from more special cases to more general, but we only list a single bound once). Our work provides the first steps towards establishing a strong theoretical foundation for the scheduling of demands in multi-connected microgrids.

4. REFERENCES

[1] S. Keshav and C. Rosenberg. On load elasticity. In *Proc. IEEE COMSOC MMTC E-Letter*, To Appear.

[2] A. Kesselman and K. Kogan. Nonpreemptive scheduling of optical switches. *IEEE Transactions on Communications*, 55(6):1212–1219, 2007.