Efficient Demand Assignment in Multi-Connected Microgrids with a Shared Central Grid

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

Sergey Nikolenko
Steklov Mathematical Institute,
National Research University
Higher School of Economics
Email: sergey@logic.pdmi.ras.ru

Srinivasan Keshav
University of Waterloo
Email: keshav@uwaterloo.ca

Alejandro López-Ortiz
University of Waterloo
Email: 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 connected to the central grid, each such demand characterized by a power level and a duration. We study algorithms that allocate generation resources to demands by configuring switched paths from sources to loads. We consider the case when each demand can be met by two generators, one of them representing the central grid and thus shared among all demands.**

I. Introduction and Motivation

Rapid penetration of distributed generation resources, especially diesel gensets and solar generation, have made it possible for groups of homes or businesses to meet much or all of their elastic demand through local generation. This usage pattern is especially common in developing countries such as India, which suffers with electricity deficit and as a result, millions of consumers are affected by inadequate power supply [1], [2]. There, customers use local generation to complement the central grid which is unreliable and limited in capacity. We anticipate that in the future geographically close microgrids will opportunistically form connections with each other to increase reliability, 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 that defines a *load balancing vector* for each demand that determines which generators are connected/available to satisfy this demand. In the simplest case, the load balancing vector of each demand consists of a single generator, so each input generator can be scheduled independently. In this setting the problem reduces to the well-known two-dimensional strip packing problem that has been extensively studied in literature [3]. In our model, each demand may have a load balancing vector with several inputs, which makes the problem more complex. Demands can be *elastic* and *non-elastic* [4]; a demand is non-elastic if its scheduling should start immediately upon arrival. A demand is *non-preemptive* if it cannot be preempted until completion once it has begun servicing, and *preemptive* otherwise. When generation capacities are insufficient to satisfy all demands some non-preemptive demands should be delayed, which leads us to the need for switching of electricity. We concentrate on scheduling elastic non-preemptive demands since (1) they encompass an important class, (2) with higher resolution of scheduling non-elastic demands may become elastic, and (3) preemptive demands can be used to complement a schedule to reuse resources not utilized by non-preemptive demands.

The concept of "packetized" electricity is not new [5], [6]. A natural representation of a demand by power level and duration is isomorphic to a data packet represented by length (duration) and CPU processing requirement (power level). Under some simplifying assumptions, the abstract problem of meeting time-limited loads (that require a certain power for a certain time) from a set of generation resources with distributed switches is similar to assigning packets of a certain length arriving to a set of output ports through a rearrangeable optical switch [7], [8]. Each packet corresponds to a demand, each input port to a generation resource, and each output port to a load; in fact, the world's first electricity switch with packets of electricity has already been designed [7].

When a central grid has limited capacity, optimal reuse of the underlying infrastructure becomes more important than cost savings for individual consumers. Pricing methods that try to move generation peak are only a tool to achieve better utilization of the underlying central grid infrastructure [9], [10]. For scheduling with "cost constraints", input ports with higher costs can be excluded from the load balancing vector, and scheduling with cost constraints for this microgrid architecture can be reduced to optimizing the makespan or some other objective on a given set of demands with modified load balancing vectors. Moreover, cost minimization implies waiting for the cheapest source, which may be unacceptable even for non-preemptive elastic demands. Therefore, we are especially concerned with minimizing the delay in satisfying a set of resource demands. We assume that there is a non-negligible penalty, called *configuration overhead*, of V time slots to change the configuration in a switching system [11]; note that V may impact a scheduling decision. We also consider an additional objective: minimizing the total number of configurations. Among other things, this reduces the wear and tear of switching equipment that directly depends on the number of necessary switching operations. We extend past work in demand assignments in rearrangable optical switches [8] to compute lower and upper bounds on the minimum number of rearrangements needed to meet a set of demands and show a relation between objective functions.

978-3-901882-56-2 ©2013 IFIP

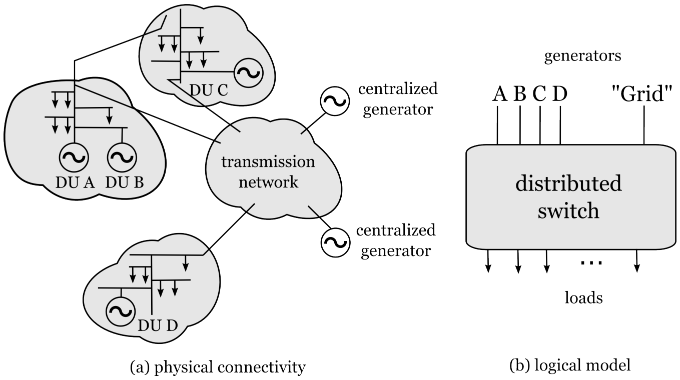

Fig. 1: The physical infrastructure in (a) is represented logically in (b). Note that "Grid" represents all centralized generators available through the transmission network. The "distributed switch" refers to the set of switches in microgrids that determines how loads are matched to generators.

Our main contributions include: (1) a formal setting for the problem of meeting demands in multi-connected micro-grids; this problem turns out to be similar to scheduling in rearrangeable optical switches; (2) a comprehensive study of various properties that should be implemented by a "perfect" scheduling policy; (3) computing upper and lower bounds on worst-case performance of scheduling policies for various objective functions; (4) a simulation study that shows the behavior of scheduling algorithms for typical workloads. A preliminary version of this work has appeared as a poster at the ACM e-Energy 2013 conference.

II. Previous Work

Keshav and Rosenberg introduced the classification of demands [4]. Saitoh and Toyoda introduced the concept of packetized electric power tagged by power switches [5]; later this work was extended by [6]. The first prototype of a switch of electricity packets was announced by the Digital Grid Consortium [7]. Kesselman and Kogan introduced the Non-Preemptive Bipartite Scheduling (NPBS) problem in [8]. NPBS is directly relevant to scheduling in optical switches. In their model, packets are assigned to inputs *a priori* but several packets may compete for the same input and output; in [8], the results of [12], [13] were extended to show that the NPBS problem is NP-hard for any value of the configuration overhead, and approximating NPBS within a ratio smaller than $\frac{7}{6}$ is also NP-hard. Furthermore, they demonstrated that the greedy algorithm achieves an approximation factor of exactly 2 for the offline version of the NPBS problem. Time slot assignment (TSA) scheduling arises in the context of Satellite Switched Time Division Multiple Access systems. Here research efforts have concentrated on extreme values of the configuration delay, such as $V = 0$ or $V = \infty$. Gopal and Wong [12] proposed heuristic algorithms for decomposing a demand matrix with minimal number of configurations for $V = \infty$. Towles and Dally [14] studied the batch scheduling problem and presented algorithms for $V = 0$ and $V = \infty$. Li and Hamdi [15] proposed a self-adjusting algorithm with different configuration delay values. Unlike [15], the greedy algorithm of [8] computes a similar non-preemptive schedule for any reconfiguration delay.

III. Problem Statement and Notation

We make two simplifying assumptions in our work: that distribution losses are negligible (reasonable given that most distributed generators are geographically close to demands; otherwise, we can just forbid to use a generator for some demands) and that a demand can be satisfied by any one source available for it; this is generally the case for small loads and avoids issues with synchronization. The case when demands can use several generators simultaneously remains an open problem for future study.

We model a set of multi-connected microgrids with a switching system $(I, \mathcal{D})$ that consists of a set of inputs I (generators) with port capacities c_i (their nominal power) and a set of demands $\mathcal{D}$ that are to be scheduled; a demand d is characterized by its length $l(d)$ (how long it lasts), width $w(d)$ (power level), and a *load balancing vector* $v(d)$ that contains the set of input ports available to process d (set of generators that can meet this demand). We assume that $(I, \mathcal{D})$ is a directed acyclic bipartite graph. Scheduling proceeds in batches. While the previous batch is served, we complete a preparation phase for the next batch. We assume that demands can be delayed at least until the end of the preparation phase. A preparation phase is divided into demand prediction, when sub-demands are predicted for the next batch, auction, when each demand receives a load balancing vector, and scheduling, when the desired objective function is optimized. When a batch has finished scheduling, the resulting schedule is served. Time is discrete; L and l are respectively the longest and shortest length in time slots among all 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*, partial mappings of demands to inputs that have to satisfy constraints imposed by port capacities and load balancing vectors. The *length* of a configuration C is defined by the longest demand scheduled during C; a configuration can be represented by a vector where each element is a set of demands assigned to the corresponding input port. We use $\in$ liberally, e.g., a demand can be said to belong to a configuration (which is formally a set of demand-input pairs). We denote by $\mathcal{D}_i^t$ the set of demands that have the first (universally available) and the i^{th} ports enabled in the load balancing vector at time t; by n_i^t, the number of demands in $\mathcal{D}_i^t$ at time t. We also denote by $k(\mathcal{D}_i^t) = \left(\sum_{j=1}^{n_i^t} w_2(j)\right)/c_i$ the total "normalized load" for each port, that is, sum of all widths $w_i(j)$ of demands from $\mathcal{D}_i^t$ divided by the i^{th} port capacity c_i. We omit the superscript t when time is clear from context. W.l.o.g. we assume that in the beginning $k(\mathcal{D}_2) \geq k(\mathcal{D}_3) \geq \ldots \geq k(\mathcal{D}_I)$. To compare the performance of different policies, we use worst-case competitive analysis. An algorithm A has *approximation ratio* α (is α*-approximate*) with respect to some (minimized) objective function if for every input $(\mathcal{D}, I)$, A produces a schedule with objective function value at most α times greater than the optimal one.

We concentrate on the practically interesting case when each demand can be met by two generators, and one of the generators, corresponding to the central grid, 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 reduces to reusing the central grid (of limited capacity) in the most efficient manner.

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');
        $C := C \cup \{(i, d)\}$, $c'_i := c'_i - w(d)$, $D := D \setminus \{d\}$;
    $\mathcal{C} := \mathcal{C} \cup \{C\}$, $D := D \setminus \{d \mid d \in C\}$.
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$(\mathcal{D}_i, c_i))$;
    Return $(1,$ CHOOSEFIRST$(\{\mathcal{D}_i\}_i, I))$.

Algorithm 3 SLD

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

Algorithm 4 SLP

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

IV. ELECTRICITY SWITCHING ALGORITHMS

A. Proposed Algorithms

We consider simple policies amenable to efficient implementation; such policies can also scale well. A general scheme for such policies 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 populates each configuration by greedily choosing the next demand to process. Once there are no more ports (generators) available for existing demands, the current configuration is finalized and a new one begins.

The CHOOSEPORTDEMAND procedure is the heart of Algorithm 1; it takes the 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 in their CHOOSEPORTDEMAND procedures. The obvious general algorithm for this case is SG, which stands for "Shared Greedy" (Algorithm 2): fill the capacities of every port except the first, then choose 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). 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; it does not matter which one for the CHOOSEDEMAND procedure, and 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. Note that in all cases, maximization and minimization is done over demands that can fit into available inputs, i.e., the constraint $w(d) < c_i$ is always present but omitted for clarity.

Four important parameters define the behaviour of a scheduling policy: (i) input port capacities, (ii) demand lengths, (iii) demand widths, and (iv) "normalized load"; all of them also affect the schedule length objective. Demand length has no impact on the number of configurations but can significantly influence the total length as L/l grows. The impact of input capacity is interesting even for unit-sized demand widths; the number of configurations is related to utilization of input capacities. The underlying problem is NP-hard: even optimal scheduling with a single port encompasses the knapsack problem. In our study, we explore the impact of each parameter on the performance of scheduling policies. We begin with a trivial relation between the two objective functions.

Theorem 1: For an input with maximal demand length L and minimal demand length l: (1) an SG algorithm with approximation ratio $\leq \alpha$ w.r.t. configurations has approximation ratio $\leq \frac{\alpha L h}{L+(l-1)h}$ w.r.t. length, where h is the optimal number of configurations; (2) an SG algorithm with approximation ratio $\leq \alpha$ w.r.t. length has approximation ratio $\leq \frac{L}{t}\left(1 + \frac{\alpha t - L}{l}\right)$ w.r.t. configurations, where t is the optimal length.

We begin with minimizing the number of configurations and then proceed to extend these results to minimize the total length of the resulting schedule. Thus, our approximation factors remain valid for *any* configuration overhead.

B. Unit Generation and Demand

We begin with the case of unit capacities: all input ports have $c_i = 1$ and all demands have $w(d) = 1$; in this case $k(\mathcal{D}_i) = |\mathcal{D}_i| = n_i$. Now the i^{th} input port is fully utilized if $\mathcal{D}_i$ is not empty at the end of a configuration, and algorithms differ in how they reuse the shared port. Intuitively, assigning demands from $\mathcal{D}_i$ with currently longest n_i should optimize the number of configurations, and indeed, we will show that SLP has the optimal number of configurations even in a more general setting. Furthermore, it turns out that we can show an upper bound on the approximation factor of *any* algorithm following the SG general scheme.

Theorem 2: For a single shared port system with unit capacities, any scheduling policy ALG from the SG family has approximation ratio at most $\frac{3}{2}$ w.r.t. configurations.

Proof: Recall that n_i are sorted, $n_2 \geq n_3 \geq \ldots$ In n_3 configurations, even under the worst possible choices for the shared port the algorithm will meet all demands except $n_2 - n_3$ demands in $\mathcal{D}_2$ and will have no choice but concentrate on port 2. Hence, the worst possible number of configurations for any SG strategy is $n_3 + \frac{n_2 - n_3}{2} = \frac{n_2 + n_3}{2}$. On the other hand, it is impossible for OPT to process $n_2 + n_3$ demands in $\mathcal{D}_2$ and $\mathcal{D}_3$ faster than in $\frac{n_2+n_3}{3}$ steps because it can process at most 3 demands per configuration with ports 1, 2, and 3. ■

We will see that for some algorithms, this bound is tight. Theorems 2 and 1 also imply a general upper bound of $\frac{3Lh}{2(L+(l-1)h)} \leq \frac{3L}{2l}$ on the approximation ratio w.r.t. length. Since OPT also falls into the SG family, we cannot show nontrivial general lower bounds; thus, we turn to lower bounds for specific algorithms. The tradeoff is between the number of configurations (optimal for SLP, worse for SLD) and their total length (good for SLD, worse for SLP).

Theorem 3: For a system with unit capacities and I ports, SLD has approximation ratio at least $\left(\frac{3}{2} - 2^{-(I-1)}\right)$ w.r.t. configurations and length.

Proof: Let $n_2 = 2^k$ for some $k \geq \lceil \log_2 I \rceil$, and let $n_i = \frac{n_2}{2}$ for $3 \leq i \leq I$. Suppose all demands in $\mathcal{D}_2$ have minimal length l while but all other demands have maximal length L. OPT can schedule the demands of $\mathcal{D}_2$ on the first two inputs over $\frac{n_2}{2}$ configurations and process all demands in every other $\mathcal{D}_i$ over the same time. On the other hand, SLD always chooses one of the $\mathcal{D}_i$ ports for $i \geq 3$ for port 1 (it does not matter which one). Thus, SLD processes all $\mathcal{D}_i$ for $i \geq 3$ over $m = \frac{n_2}{2}\left(\frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{2^{I-2}}\right) = \frac{n_2}{2}\left(1 - 2^{-(I-2)}\right)$ configurations, together with m demands from $\mathcal{D}_2$ on the second input. To schedule the rest of $\mathcal{D}_2$, SLD needs some $\frac{(n_2-m)}{2}$ configurations more, for the total of $m + \frac{n_2-m}{2} = \frac{3-2^{-(I-2)}}{4}n_2$. Since the algorithm with minimal total length has at least as many configurations as one with minimal number of configurations, SLD has approximation ratio $\left(\frac{3}{2} - 2^{-(I-1)}\right)$ w.r.t. length too. ■

Note that this approximation ratio tends to $\frac{3}{2}$ as $I \to \infty$, so the bound from Theorem 2 is tight for SLD.

C. Unit Widths, Heterogeneous Capacities

Next we turn to the case of heterogeneous input port capacities (but still unit demand widths). The first interesting question is whether it is possible to break the upper bound of SG stated in Theorem 2 in case of heterogeneous input capacities. For this purpose, we introduce additional notation. Suppose that OPT schedules all demands over h configurations. We classify all input ports except the shared one into two groups: $\mathcal{I}_1$ and $\mathcal{I}_2$. $\mathcal{I}_1$ contains all input ports such that $k(\mathcal{D}_i) > h$, $2 \leq i \leq I$; $\mathcal{I}_2$, all other ports (excluding the first). We prove that SG performance depends on the relationship between capacity of the shared port and the capacities of $\mathcal{I}_1$ and $\mathcal{I}_2$. As a special case of SG we consider SLD. Moreover, at least for the case when $c_1 > c(\mathcal{I}_1)$ and $c_1 \leq c(\mathcal{I}_2)$ and non-unit-sized capacity constraints, the upper bound of $\frac{3}{2}$ for unit-capacity constraints can be broken. Intuitively, SG can mistakenly reuse the first port for the demands of $\mathcal{D}(\mathcal{I}_2)$, so that it cannot reuse the first port for the demands of $\mathcal{D}(\mathcal{I}_1)$ enough times.

Theorem 4: For a system with capacities $c_1 = 2c(\mathcal{I}_1)$ and $c(\mathcal{I}_2) = 2c_1$, the number of configurations in SLD's schedule is at least $\frac{5}{3}$ times more than optimal.

Proof: Consider the following inputs: $\mathcal{D}(\mathcal{I}_1)$ consists of $9c(\mathcal{I}_1)$ demands of minimal length l and $\mathcal{D}(\mathcal{I}_2)$ consists of $12c(\mathcal{I}_1)$ demands of maximal length L, $h > 0$. OPT schedules all demands of $\mathcal{D}(\mathcal{I}_1)$ on the first and $\mathcal{I}_1$ input ports and in parallel all demands of $\mathcal{D}(\mathcal{I}_2)$ on $\mathcal{I}_2$ during 3 configurations. On the other hand, SLD schedules $\mathcal{D}(\mathcal{I}_2)$ first on the first and $\mathcal{I}_2$ input ports during $\frac{12}{2+4} = 2$ configurations; and later during $\lceil \frac{9-2}{2+1} \rceil = 3$ configurations the remaining demands of $\mathcal{D}(\mathcal{I}_1)$. So the number of configurations in SLD's schedule is at least 5. Thus, SLD's schedule has at least $\frac{5}{3}$ times more configurations than the optimal schedule. ■

The result stated in Theorem 4 demonstrates that generalizing the results for unit capacity constraints from Section IV-B is non-trivial even for unit-sized demand width. The following theorem specifies an upper bound on a number of configurations of SG in the case of non-unit-capacity constraints. Recall that we still consider unit-sized width of demands.

Theorem 5: The number of configurations in the schedule of SG is at most twice more than in the optimal schedule if one of the following conditions holds: (1) $c_1 \leq c(\mathcal{I}_1)$; (2) $c_1 > c(\mathcal{I}_1)$ and $c_1 > c(\mathcal{I}_2)$.

Proof: Suppose that OPT schedules all demands over h configurations. Consider the following two cases.

Case 1: $c_1 \leq c(\mathcal{I}_1)$. Since $c_1 \leq c(\mathcal{I}_1)$, OPT requires at least $h/2$ configurations to schedule all demands of $\mathcal{D}(\mathcal{I}_1)$. Since $\mathcal{I}_1$ is utilized over h configurations, $c_1 \leq c(\mathcal{I}_1)$, and the first port is used only after all other ports are utilized, SG schedules at least $k(\mathcal{D}_i)/2$. Thus, at most h additional configurations are required for SG to process the rest.

Case 2: $c_1 > c(\mathcal{I}_1)$ *and* $c_1 > c(\mathcal{I}_2)$. W.l.o.g. SG schedules all demands of $\mathcal{D}(\mathcal{I}_2)$ during the first $x \leq h$ configurations. By definition of SG, the first port can be reused only during the same $x \leq h$ configurations for scheduling of $\mathcal{D}(\mathcal{I}_2)$. If during the first h configurations SG reuses at most $\frac{c_1 h}{2}$ total capacity of the first port to schedule the demands of $\mathcal{D}(\mathcal{I}_2)$, SG reuses at least $\frac{c_1 h}{2}$ capacity of the first port. Since $c_1 > c(\mathcal{I}_1)$, and each assigned port of $\mathcal{I}_1$ with unassigned demands is fully utilized, over the next $h/2 \geq x/2$ configurations (after $\mathcal{D}(\mathcal{I}_2)$ is fully scheduled) SG schedules at least the $\frac{1}{2}$ of total width of $\mathcal{D}(\mathcal{I}_1)$. So over at most h additional configurations SG schedules all remaining demands of $\mathcal{D}(\mathcal{I}_1)$. If during the first h configurations SG reuses more than $\frac{c_1 h}{2}$ overall capacity of the first port to schedule the demands of $\mathcal{D}(\mathcal{I}_2)$, then, since $c_1 > c(\mathcal{I}_2)$, SG schedules $\mathcal{D}(\mathcal{I}_2)$ over less than $h/2$ configurations. Similar to the previous case, over less than h additional configurations SG schedules all remaining demands of $\mathcal{D}(\mathcal{I}_1)$. Thus, the number of configurations in SG's schedule is at most twice more than in the optimal schedule. ■

Clearly, for the case of heterogeneous input port capacities and unit-sized width, the optimal algorithm OPT w.r.t. number of configurations belongs to the family of policies represented by SG. Moreover, in this case it is possible to identify OPT explicitly. We show that Algorithm 4 (SLP) that prioritizes the $k(\mathcal{D}_i)$ characteristic is optimal.

Theorem 6: SLP has optimal number of configurations for the case of heterogeneous input capacities and unit-sized demands width.

Proof: Suppose that after t configurations $\mathcal{D}_i$ has a maximal value of $k(\mathcal{D}_i^t)$ in SLP's schedule for the first time. By definition of SLP and since $\mathcal{D}_i$s compete only for the first port, the following claim holds: for any $t' \geq t$, $s \leq i$, $k(\mathcal{D}_i^{t'}) - k(\mathcal{D}_s^{t'}) \leq 1$ and $i \leq b \leq I$, $k(\mathcal{D}_b^{t'}) \leq k(\mathcal{D}_i^{t'}) \leq 1$.

So there is a configuration $t'' \geq t$ after which for any two non-fully-scheduled $\mathcal{D}_i$ and $\mathcal{D}_j$, $\left|k(\mathcal{D}_i^{t''}) - k(\mathcal{D}_j^{t''})\right| \leq 1$.

Case 1. Configuration t'' is last in SLP*'s schedule.* Since SLP reuses the first port only for $\mathcal{D}_i$ with currently largest $k(\mathcal{D}_i^{t''})$, the first port is never used for the demands of $\mathcal{D}_j$ with the total original width of $(t''-1)k(\mathcal{D}_j)$, and during the first $(t''-1)$ configurations in SLP's schedule the first port is fully utilized. Thus, OPT cannot schedule all input demands in less than t'' configurations.

Case 2. Configuration t'' is not last in SLP*'s schedule.* Since SLP reuses the first port only for $\mathcal{D}_i$ with currently largest $k(\mathcal{D}_j)$, the first port is never used for the demands of $\mathcal{D}_j$ with total original width $t''k(\mathcal{D}_j)$. Also, during the first t'' configurations in SLP's schedule the first port is fully utilized and OPT cannot schedule more demands with total width $t''\sum_{m=2}^{I} c_m$ during that time. Let SLP and OPT schedules contain respectively h_0 and h_1 configurations, $h_0 \geq h_1 > t''$. Since during (t'', h_0) configurations for any two $\mathcal{D}_i$ and $\mathcal{D}_j$ in SLP's schedule, $|k(\mathcal{D}_i^{t''}) - k(\mathcal{D}_j^{t''})| \leq 1$, all ports are fully utilized, and OPT cannot schedule demands with total width of $(h_0-1)\sum_{m=2}^{I} c_m$ in less than h_0-1 configurations. Since there are additional demands that SLP schedules in one additional configuration its schedule has the optimal number of configurations. ■

SLD is a special case of SG, so Theorems 5 and 4 imply the following corollaries.

Corollary 7: For a system with heterogeneous input capacities, SLD's schedule is at most twice longer than the optimal schedule if one of the following conditions holds: (1) $c_1 \leq c(\mathcal{I}_1)$; (2) $c_1 > c(\mathcal{I}_1)$ and $c_1 > c(\mathcal{I}_2)$.

The following corollary obviously follows since SLP is optimal; a better bound remains an interesting open problem.

Corollary 8: Under heterogeneous input capacities, SLP schedule is at most $\frac{L}{l}$ longer than OPT schedule.

D. Heterogeneous Demand Widths

Since all considered algorithms utilize the i^{th} port if some unassigned demands have port i in their load balancing vectors, the i^{th} port's capacity is at least half utilized at the end of a configuration. Let w_{max}^i be the maximal demand width that has the i^{th} port available in the load balancing vector.

Claim 9: For a system with non-unit-sized width of demands: (1) SG has an approximation ratio at most 4 w.r.t. the number of configurations; (2) SG with capacity of i^{th} port extended by w_{max}^i has an approximation ratio at most 2 w.r.t. configurations as compared to the optimal algorithm with original capacity constraints.

Once we begin to consider heterogeneous demand width, by Claim 9 we simply introduce an additional factor of two in all upper bounds stated above. Moreover, if we allow for an upgraded system, as in item (2) of Claim 9, all upper bounds remain the same.

V. Conclusion

In this work, we have considered the scheduling of demands in a geographically close set of microgrids. A short summary of our theoretical results is shown in Table I (lower bounds obviously propagate from more special cases to more general, but we only list a single bound once). Our work contains the first steps towards establishing a theoretical foundation for demand scheduling in multi-connected microgrids.

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

TABLE I: Theoretical results summary.

In this work we have only considered the case when a demand is allowed to have only two available ports, with one of them fixed to be the "central grid" shared by all demands. A generalization to the case of arbitrary load balancing vectors remains an interesting open problem, as well as a more practical simulation study, either on real data (if such data for microgrids becomes available) or with synthetically generated inputs.

References

[1] S. Biswas, "India struggles to deliver enough power," 2012. [Online]. Available: http://www.bbc.co.uk/news/world-asia-india-19063241

[2] V. Bajaj, "Ten interesting things about India power," 2012. [Online]. Available: http://www.nytimes.com/2012/04/20/business/global/india-struggles-to-deliver-enough-electricity-for-growth.html

[3] A. Lodi, S. Martello, and M. Monaci, "Two-dimensional packing problems: A survey," *European Journal of Operational Research*, vol. 141, no. 2, pp. 241–252, 2002.

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

[5] H. Saitoh and J. Toyoda, "A new concept of electric power network for the effective transportation of small power of dispersed generation plants," *IEEE J. Trans. PE*, vol. 115, no. 6, pp. 568–575, 1995.

[6] R. Abe, H. Taoka, and D. McQuilkin, "Digital grid: Communicative electrical grids of the future," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 399–410, 2011.

[7] "World's first digital grid router controls electricity packets, boosts grid access for renewables in Japan," 2013. [Online]. Available: http://www.japanfs.org/en/pages/032474.html

[8] A. Kesselman and K. Kogan, "Nonpreemptive scheduling of optical switches," *IEEE Transactions on Communications*, vol. 55, no. 6, pp. 1212–1219, 2007.

[9] T. Carpenter, S. Singla, P. Azimzadeh, and S. Keshav, "The impact of electricity pricing schemes on storage adoption in Ontario," in *e-Energy*, 2012, p. 18.

[10] A. K. Mishra, D. E. Irwin, P. J. Shenoy, J. Kurose, and T. Zhu, "SmartCharge: cutting the electricity bill in smart homes with energy storage," in *e-Energy*, 2012, p. 29.

[11] S. Paudyal, C. A. Canizares, and K. Bhattacharya, "Optimal operation of distribution feeders in smart grids," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4495–4503, 2011.

[12] M. A. Bonuccelli, I. S. Gopal, and C. K. Wong, "Incremental time-slot assignment in SS/TDMA satellite systems," *IEEE Transactions on Communications*, vol. 39, no. 7, pp. 1147–1156, 1991.

[13] P. Crescenzi, X. Deng, and C. H. Papadimitriou, "On approximating a scheduling problem," *J. Comb. Optim.*, vol. 5, no. 3, pp. 287–297, 2001.

[14] B. Towles and W. J. Dally, "Guaranteed scheduling for switches with configuration overhead," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 835–847, 2003.

[15] M. Chan and M. Hamdi, "An active queue management scheme based on a capture-recapture model," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 4, pp. 572–583, 2003.