Efficient Solar Provisioning for Net-Zero Internet-Scale Distributed Networks

Vani Gupta†, Prashant Shenoy†, Ramesh K. Sitaraman†‡
†University of Massachusetts, Amherst ‡Akamai Technologies
{vani, shenoy, ramesh}@cs.umass.edu

***Abstract*—Internet-scale Distributed Networks (IDNs) are global distributed systems consisting of a network of hundreds of thousands of servers located in hundreds of data centers around the world. A Content Delivery Network (CDN) is an example of an IDN that delivers content globally through a large network of servers. IDNs consume large amounts of energy and their energy requirements are projected to increase significantly in the future. With carbon emissions from data centers increasing every year, energy solutions for data centers powered by renewables are critical for the sustainability of data centers and for the environment. In this paper, we study the benefits of using solar energy to green IDNs. We study the impact of leveraging global data center locations with high solar potential on the number of solar panels needed to power a CDN. We develop optimal algorithms and heuristics to help minimize the number of solar panels provisioned across the CDN, while making it net-zero. We empirically evaluate our algorithms using extensive load traces from Akamai's global CDN and solar data from PVWatts. Overall, with unrestricted load movement, we can reduce the number of solar panels by 36%, 68%, and 82% for net-zero year, month and week respectively. Our results show that our algorithms can significantly reduce the number of solar panels we need to power our CDN, thereby making sustainability of Internet-scale distributed networks more achievable.**

***Index Terms*—solar, renewables, net-zero, content delivery networks, internet-scale distributed networks, CDN, IDN, green computing**

I. Introduction

Modern Internet services are delivered using Internet-scale Distributed Networks (IDNs) that rely on a global network of hundreds of thousands of servers located all over the world. IDNs include cloud and Internet services such as content delivery networks (CDNs) that deliver web content, applications, and streaming media to clients via hundreds of thousands of servers located in thousands of data center locations throughout the world. Such large Internet-scale systems consume large amounts of energy and consequently incur large energy bills. A research study in 2010 [24] estimated that Google's energy consumption was more than 1,120 GWh and their annual electricity bill exceeded $67M. The environmental impact of these massive IDNs is also significant and is increasing every year. Carbon emissions from data centers are growing at the rate of 15% each year [12].

Given the large energy requirements of IDNs, data centers that are powered using renewables are gaining a lot of traction the industry and the research community. In just six years, Apple's use of renewable energy to power its corporate facilities, retail stores, and data centers worldwide went from 16% in 2010 to 96% in 2016 [1]. Apple has now committed to powering all its facilities world-wide with 100% renewable energy. Google announced this year that it will achieve the milestone of purchasing 100% renewable energy to match consumption for global operations, including its data centers and offices [2]. Google and Apple, use a combination of methods to green their operations. Google is trying to achieve 100% green status by generating some green energy on-site, but mainly by acquiring Renewable Energy Credits (RECs) through their Power Purchase Agreements (PPAs) with renewable energy companies [2]. PPAs are contracts that allow companies to buy power from energy companies at negotiated prices. RECs are a means to keep track of who is using and consuming green energy. Companies create a REC if they generate a MWh of green energy and consume a REC if they consume a MWh of green energy. Google enters into PPAs with renewable energy companies and buys renewable energy from them. Thereafter Google sells the energy back on the grid. This indirect way of generating green energy gets Google RECs that they then use to offset their grid energy use. Apple produces its own green energy where possible, and then it also relies on PPAs and RECs to fully green its operations [1].

There has been a lot of research on making data centers greener by reducing energy consumption or using energy generated from renewable sources. Prior work includes energy reduction using server shutdown or low-power states during off-peak times [15] [18] [5] [25]. There is also work on job scheduling based on predicted solar [11] [10] and load balancing to encourage use of renewable energy [17] [16] [9]. Separately there has been a study [4] for selecting sites for and provisioning green data centers using a follow-the-renewables approach. Greening IDNs is now a necessity for sustainable growth of IDNs, for reducing environmental impact, and for lowering energy costs for companies. When a system produces enough green energy to off-set its brown energy use over a time period, it is said to be '*net-zero*' over that time period. While prior work has addressed greening individual data centers, the problem of greening a large distributed network of data centers, such as an IDN, using renewables, has not received much research attention. To address this issue, we consider the following research question: how can we efficiently provision solar arrays across a global IDN with hundreds of locations to make it net-zero? We address this important question in our study by designing and implementing optimal algorithms and heuristics for provisioning solar panels, and evaluating those algorithms on a real world IDN trace and a year's worth of PVWwatts solar data.

When a system produces enough green energy to off-set its brown energy use over a time period, it is said to be '*net-zero*' over that time period. E.g. if a system produces enough green energy to off-set its brown energy use over a month, we say that it is a '*net-zero month*' system. The energy IDNs consume is dependent on factors like the load they serve, the number of servers that are active at any given point in time, the energy required to cool servers and to run supporting equipment etc. This energy is the *energy demand* in our model. On the other hand, the solar output of the panels is the *energy supply*. In order to be net-zero over a time period, our problem is to match the *energy demand* with the *energy supply* in that time period. In this paper, we study the solar potential of being net-zero over different time periods including a week, a month and a year, while also ensuring that we provision the the minimum number of panels to meet the demand.

There is a significant difference in solar output between different locations on the globe, due to differences in latitude, longitude, and weather. IDNs are deployed as a global network of data centers and the services they provide are replicated across those data centers. Proximity of servers to users is the main reason for global deployment. Therefore, data centers cannot be deployed only in locations where there is high solar output throughout the year, they also need to be close to users. However, replication of services in an IDN allows us to shift workload from one location to another to leverage high solar outputs, and we use this feature extensively to optimize solar panel provisioning and energy usage.

Contributions: Our contributions are listed below:

- *Determining Solar Potential for Global IDNs:* We conduct a comprehensive study to analyze the net-zero solar potential for existing global IDNs with data centers located in hundreds of locations throughout the world. In order to reduce the number of panels provisioned, we leverage global locations that have high solar output. To determine the number of panels needed to be net-zero, we move load in an off-line fashion and ensure that the data center energy demand is matched by solar energy supply for the duration of the time period for which we aim to be net-zero.
- *Algorithm Design:* We design optimal and heuristic algorithms to minimize the number of panels we need to serve the IDN's load by taking advantage of higher levels of solar across various regions on the globe. Firstly, we design our algorithms such that they can be generalized to different net-zero time periods, including net-zero week, month, and year. Secondly, we also design the LP to study the impact of restricting load movement within a certain radius when determining the number of panels to be provisioned. Finally, our LP can also be easily configured to restrict locations where solar panels are installed.
- *Extensive Trace-based Evaluation:* We evaluate our algorithms on an extensive load trace from one of the world's largest CDN. The month-long trace consists of five-minute information from 100,592 servers in 724 global data center locations from Akamai's CDN. We see significant reduction in the number of panels and also find that our heuristic algorithms perform well compared to the optimal. Overall, with unrestricted radius of load movement, we can reduce the number of solar panels by 36% for net-zero year. For net-zero month by about 67% to 68%, and for net-zero week by about 71% and 74% for heuristics, and by 82% for the optimal. Our solution provisions panels by taking advantage of global locations with high solar output. We study the impact of restricting the radius of load movement on the reduction in the number of panels. Even with performance constraints, we can see a significant reduction in the number of panels. For r=500kms, we see a reduction of about 9.7%, 27%, and 53% for net-zero year, month, and week respectively.

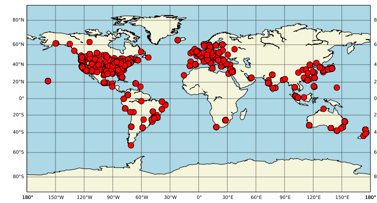


Fig. 1: Plot showing the diversity in geographic locations that can make up a global IDN

In the rest of the paper, we present background and define the problem in Section II and III respectively. We present our optimal and heuristic algorithms in Section IV and experimental methodology in Section V. In Section VI we describe our empirical results. We discuss related work in Section VII and finally present our conclusions in Section VIII.

II. Background

Internet-scale Distributed Networks: Internet-scale distributed networks (IDNs) are large-scale global networks that are comprised of data centers in several locations across the world. Content delivery networks (CDNs) are examples of IDNs and are used to deliver content, streaming audio, video, applications etc. on the web. Figure 1 shows data center locations part of the Akamai CDN. Commercial CDNs use two levels of load-balancing in their systems: *local* and *global*. When a user requests content, the *global* load-balancer assigns the request to a server cluster located 'close-by' to minimize loss and latency [19]. The *local* load-balancer then maps the request to a specific server in the cluster. In order to assign users to nearby data centers and minimize loss and latency, CDNs replicate their services so as to have redundancy in the choice of data centers. This replication is also very useful if load from one data center is assigned to another data center for any other reason, e.g. to leverage a local feature like high solar output.

Energy Consumption Model for an IDN: The primary source of energy consumption in IDNs are the numerous servers deployed in all the various data centers (*server energy*).

The energy consumed by a server is largely dependent on the amount of load it is serving, so we can model the energy consumed by a server as a function of its load (we use normalized load λ, $0 \leq \lambda \leq 1$, which is the actual load as a fraction of its capacity). However, although energy consumed is largely dependent on the load, servers are not energy proportional and still consume some energy, roughly 60%, when they are idle. For our work, we use the standard linear model of server power consumption [3] that defines power consumed by a server as $P_{idle} + (P_{peak} - P_{idle})\lambda$,, where λ is the normalized load on the server, P_{idle} is the power consumed by server that is idle, and P_{peak} is the power consumed by the server that has peak load. We assume that we can move load between servers to consolidate load, and shut down idle servers, so as to use the minimum number of servers needed to serve the load [14]. Our assumption is that such consolidation of load is done at each of the data center locations for each time period. Using the linear model and consolidating load between servers, we then determine the power consumed by the data center. The power consumed (in watts) by the data center in each 5-minute time interval is then multiplied by the number of seconds (5*60) to get the energy consumed by the data center in each time interval (in joules).

In addition to *server energy*, we also need energy to cool them (*cooling energy*). Heat dissipated by servers is a function of the energy they consume. The more heat they dissipate, the more energy is needed to cool them. So cooling energy is proportional to server energy. A recent study of data center energy consumption [20] showed that servers and cooling consumed 56% and 30% of the total energy respectively. Thus, most of the energy consumed by a data center is spent to power and cool servers. We assume a PUE of about 1.8 [27] and scale up server energy consumption to account for cooling energy.

Net-zero Systems: A 'net-zero energy' data center is designed and managed in a manner that uses on-site renewables to entirely offset the use of any non-renewable energy from the grid [8]. That is, on-site renewable energy produced is at least the energy consumed over the period. Extending this basic definition, we define a 'net-zero' IDN for different time periods as below:

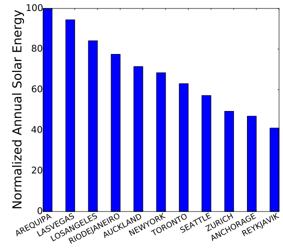
- A *Net-zero Year IDN* is net-zero over a year.
- A *Net-zero Month IDN* is net-zero for every month in a year.
- A *Net-zero Week IDN* is net-zero for every week in a year.

By definition, net-zero week is the most stringent requirement, followed by net-zero month, and finally net-zero year. So a net-zero week IDN will also be net-zero month and net-zero year. A net-zero month IDN will also be net-zero year.

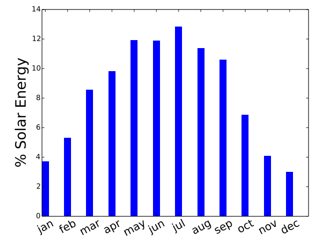
Solar Panels and Factors Affecting Solar Output: A solar panel is an electrical device that converts sunlight into electricity using the photo-voltaic effect [21].

Several factors affect solar output, and we list some of them below:

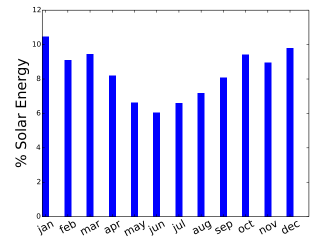
- *Location:* As shown in Figure 2(a), there is a large variation in annual solar output based on location. Locations like Las Vegas have higher annual solar output, while locations like Anchorage have lower solar output.



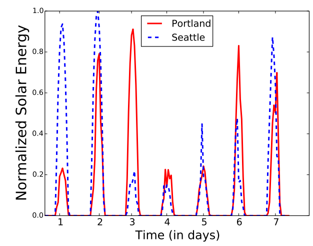
(a) Annual solar output showing large variation across locations



(b) Monthly solar variation by season in a northern hemisphere location (Seattle)



(c) Monthly solar variation by season in a southern hemisphere location (Perth)



(d) Daily solar variation for two close-by locations

Fig. 2: Figure showing rich solar variations across the globe

- *Season:* Solar output also varies by season. Summer months tend to have higher levels of solar output than winter months. This can be seen in Figure 2(b) which shows the monthly solar output for Seattle, WA.
- *Hemisphere:* Figures 2(b) and 2(c) show monthly solar output for Seattle in the northern hemisphere and for Perth in the southern hemisphere. As we see, the trend in the levels of solar output is reversed for these locations, given the northern and southern hemispheres experience opposite seasons.
- *Daily variations:* Solar output goes to zero when the sun is not shining and so the hour of the day affects solar output. As Figure 2(d) shows, for each location there are hours when solar is zero and then it rises steadily, peaks and then falls again once the sun sets.
- *Other factors:* All other factors like location, season, and time of day remaining constant, solar output can still change based on several factors that may include weather, cloud cover, pollution etc. As Figure 2(d) shows, within the same location, season, hemisphere and time of day, we see large variations between solar output from one day of the week to the next. Locations that are close-by e.g. Portland, OR and Seattle, WA also show large variations in solar output, as we can see in the figure.

Given the above analysis, we conclude that solar output is highly variable across time and space and is affected by several diverse factors, and their interplay. While this intermittency and variability is a challenge, it is also an opportunity in the context of a global IDN. Given an IDN has replicated services, we can leverage high levels of solar by moving load to locations that have high solar output. In our study, we leverage these complex variations to reduce the number of panels provisioned.

III. Problem Statement

Data centers consume energy to maintain, run, and cool servers and other equipment. For a net-zero data center, energy supply needs to be matched by the demand by using energy generated from renewables, like solar. There is a large variation in solar output across global locations, with certain locations being excellent for solar generation. In this paper, we address the problem of efficient solar panel provisioning for global net-zero IDNs. To provision panels efficiently, we move load to locations with high solar output. While defining the problem, we make two simplifying assumptions. First, we assume that it is possible to install as many solar panels as we need in any location. Second, we assume that it is possible to deploy as many servers as we need at any location.

Our provisioning problem can be broadly stated as follows: *Using load movement, how can we efficiently provision solar panels for a global IDN so it is net-zero over a given time period such as a year, a month, or a week.* Importantly, being net-zero over a certain period implies total energy user in that period across the IDN equal total solar output for the IDN. It does not require instant power usage to be fully met by solar production at that instant.

Specifically, we study the following research questions:

- *How to achieve net-zero IDNs with and without performance constraints:* Firstly, in order to determine our full solar potential, we analyze how much reduction we can see in the number of panels if we allow load to be assigned to any location on the globe, without worrying about performance. This scenario yields best case results for reduction in the number of panels, and becomes a point of comparison for results under more constrained scenarios. Secondly, in order to reduce latency, it is important to serve load from locations close to users. Therefore, we place constraints on the radius within with we must operate while moving load to a data center with higher solar output.
- *How do results change if panels are assigned to top k locations only:* Solar panel installation may only be possible in large data centers in major metro areas. We analyze the number of solar panels needed to make IDNs net-zero if the panels could only be deployed in large data centers in major cities. For an IDN data centers sizes vary by location and population. Generally speaking, areas with larger population tend to have larger data centers so servers can be proximal to users in order to reduce latency. We use the number of servers as an indicator for the size of the data center. In this scenario, we sort our data centers by the number of servers they have and consider only the top k locations as candidates for installing panels, ensuring panels are installed at bigger data centers. We study how the number of panels provisioned change as we vary k.

dcid/time	1	2...	n
1	s_{11}	s_{12}...	s_{1n}
2	s_{21}	s_{22}...	s_{2n}
..	..	..	..
m	s_{m1}	s_{m2}...	s_{mn}

TABLE I: Supply Matrix

dcid/time	1	2...	n
1	l_{11}	l_{12}...	l_{1n}
2	l_{21}	l_{22}...	l_{2n}
..	..	..	..
m	l_{m1}	l_{m2}...	l_{mn}
total load	l_1	l_2...	l_n

TABLE II: Demand Matrix

IV. Algorithms for Solar Provisioning in IDNs

We begin with an LP formulation to solve the solar provisioning problem under performance constraints. While the LP formulation is optimal, it is also computationally intensive. Therefore, to reduce run-time complexity in the no performance constraint scenario, we also define two heuristic algorithms that run faster while yielding comparable results.

Before defining the algorithm, we first define the main inputs to the algorithm.

- *Demand and Supply Matrices:* We set up the inputs as two matrices: one a demand matrix (shown in Table II) and the other a supply matrix (shown in Table I). The values in the demand matrix 'l_{ij}' represent the energy used by the data center at the corresponding time. The values in the supply matrix 's_{ij}' represent the solar energy available at that location per panel. Both matrices have 'm' rows corresponding to data centers and 'n' columns corresponding to time periods.
- *Neighbors of a data center 'i':* We define $N_{i\delta}$ to be the set of all data centers within a radius of δkms from i.

A. Optimal LP Formulation with Performance Constraints

We formulate an LP, which we refer to as LP_{perf}, to determine the minimum number of panels we can provision to meet demand, given the solar energy available at various locations. In addition to the inputs defined above, for each data center i, we define variables l_{ijt} to be the load moved from data center i to data center j, at time t, $\forall j \in N_{i\delta}$. We also define p_i as the number of panels installed at data center i. Given this setup, we define the LP as shown below. We minimize the total number of solar panels provisioned in the objective function as below:

$$\text{Min:} \sum_{i=1}^{m} p_i \tag{1}$$

We then define constraints as below: Incoming load should be less than or equal to the solar supply:

$$\text{s.t.:} \sum_{i \in N_{j\delta}} l_{ijt} \leq s_{jt} * p_j, \qquad \forall j, t \tag{2}$$

Total outgoing load, including load moved from the data center to itself should be equal to the starting load:

$$\sum_{j \in N_{i\delta}} l_{ijt} = l_{it}, \qquad \forall i, t \tag{3}$$

In addition to these constraints, we also have non-negative constraints for each of the variables defined:

$$l_{ijt} \geq 0, \qquad \forall i \in N_{j\delta}, \forall j, t \tag{4}$$
$$l_{it} \geq 0, \qquad \forall i, t \tag{5}$$
$$s_{jt} \geq 0, \qquad \forall j, t \tag{6}$$
$$p_j \geq 0, \qquad \forall j \tag{7}$$
$$\delta \geq 0 \tag{8}$$

Without performance constraints, i.e. for an unlimited radius of load movement, where load can potentially be assigned from a data center to any other data center, the number of l_{ijt} variables, and the number of their associated constraints, becomes very large. For net-zero month, for m data centers, the number of l_{ijt} type variables is $m * m * 12$. For our roughly 1800 data centers and a net-zero month scenario, this number of these variable is a huge number: 1800*1800*12 = 38,880,000. Therefore, we device a alternative LP formulation for the no performance scenario.

B. Optimal LP Formulation With No Performance Constraints

In this section, we do not consider performance constraints in our formulation. In addition to the inputs and variables defined earlier, we define in_{it} is the load moved into data center i at time t. Similarly out_{it} is the load moved out of data center i at time t. Given this setup, we define the LP, referred to as LP_{nperf}, as shown below. We minimize the total number of solar panels installed:

$$\text{Min:} \sum_{i=1}^{m} p_i \tag{9}$$

Subject to four types of constraints: Total incoming load minus the outgoing load should be zero:

$$\text{s.t.:} \sum_{i=1}^{m} in_{it} - \sum_{i=1}^{m} out_{it} = 0, \qquad \forall i, t \tag{10}$$

Incoming load should be less than or equal to the solar supply:

$$l_{it} + in_{it} - out_{it} \leq s_{it} * p_i, \qquad \forall i, t \tag{11}$$

Total outgoing load should be less than or equal to the sum of the starting load and any incoming load:

$$l_{it} + in_{it} - out_{it} \geq 0, \qquad \forall i, t \tag{12}$$

In addition to these constraints, we also have non-negative constraints for each of the variables defined:

$$l_{it} \geq 0, \qquad \forall i, t \tag{13}$$
$$s_{it} \geq 0, \qquad \forall i, t \tag{14}$$
$$p_i \geq 0, \qquad \forall i \tag{15}$$
$$in_{it} \geq 0, \qquad \forall i, t \tag{16}$$
$$out_{it} \geq 0, \qquad \forall i, t \tag{17}$$

The LP_{nperf} does away with the l_{ijt} type variables that represent load moved from data center i to data center j at time t. Instead it only models load moved in and out of each data center at time t using variables in_{it} and out_{it}. So for the net-zero month case for m data centers for LP_{perf}, where we had to contend with $12m^2$ variables, we now only have to work with $12m$ variables for the LP_{nperf} formulation.

Theorem 1. *LP_{perf} with unlimited δ and LP_{nperf} are equivalent.*

Proof: (Part 1) *A solution of LP_{perf} with unlimited δ is also a solution of LP_{nperf}*
Suppose we have a solution for unlimited δ case of LP_{perf} called S_{perf}:

$$S_{perf} = \{l_{ijt} | \forall i, j, t\} \cup \{p_i | \forall i\}$$

Using this solution, we construct another solution S as below:

$$S = \{in_{jt} | \forall j, t\} \cup \{out_{jt} | \forall j, t\} \cup \{p_j | \forall j\}$$

Where we define in_{it} and out_{it} in terms of l_{ijt} variables as below:

$$in_{jt} = \sum_{i=1}^{m} l_{ijt}$$

$$out_{jt} = \sum_{k=1}^{m} l_{jkt}$$

We now show that S is a feasible solution of LP_{nperf} by showing it satisfies all its constraints (10 through 17):

- *Constraint 10 of LP_{nperf}:*

$$\sum_{j=1}^{m} in_{jt} - \sum_{j=1}^{m} out_{jt}$$
$$= \sum_{j=1}^{m}\sum_{i=1}^{m} l_{ijt} - \sum_{j=1}^{m}\sum_{k=1}^{m} l_{jkt}$$

  Given every outgoing load has a corresponding incoming load, we can cancel all the terms pairwise in the above difference. Therefore,

$$\sum_{j=1}^{m} in_{jt} - \sum_{j=1}^{m} out_{jt} = 0$$

- *Constraint 11 of LP_{nperf}:*

$$\begin{aligned} & l_{jt} + in_{jt} - out_{jt} \\ & = l_{jt} - out_{jt} + in_{jt} \\ & = l_{jt} - \sum_{k=1}^{m} l_{jkt} + \sum_{i=1}^{m} l_{ijt} && \text{From def of } out_{jt}, in_{jt} \\ & = 0 + \sum_{i=1}^{m} l_{ijt} && \text{From Eq 3 of } LP_{perf} \\ & \leq s_{jt} * p_j && \text{From Eq 2 of } LP_{perf} \end{aligned}$$

- *Constraint 12 of LP_{nperf}:*

$$\begin{aligned} & l_{jt} + in_{jt} - out_{jt} \\ & = 0 + in_{jt} && \text{From Eq 3 of } LP_{perf} \\ & = \sum_{i=1}^{m} l_{ijt} && \text{By definition} \\ & \geq 0 && \text{From Eq 4 of } LP_{perf} \end{aligned}$$

- *Non-negativity Constraints 13 through 17 of LP_{nperf}:* All non-negativity constraints also hold. Constraints 13 through 15 are also constraints in LP_{perf}, so they get ported directly. Constraint 16 and 17 are true by definition, as they are sums of quantities greater than or equal to zero.

So given S satisfies all the constraints of LP_{nperf}, it is also a solution of LP_{nperf}. Therefore, for unlimited δ, every solution S_{perf} of LP_{perf} can be reduced to a feasible solution S of LP_{nperf}.

■

Proof: (Part 2) *Proving that a solution of LP_{nperf} is also a solution of LP_{perf}*

Suppose we have a solution S_{nperf} for LP_{nperf} as defined below:

$$S_{nperf} = \{in_{jt} | \forall j, t\} \cup \{out_{jt} | \forall j, t\} \cup \{p_j | \forall j\}$$

Using this solution, we construct another solution S below by defining l_{ijt} values in terms of in_{it} and out_{it} values:

$$S = \{l_{ijt} | \forall i, j, t\} \cup \{p_i | \forall i\}$$

We first consider the net inflow and out flow for each location, given by the following difference:

$$in_{jt} - out_{jt}, \forall j, t$$

Respecting these net inflows and outflows, we define our l_{ijt} by matching inflows with outflows, ensuring we only pick non-negative values for each l_{ijt}. Note this matching can always be done because constraint 10 is satisfied, meaning net inflow and net outflow in the system are the same. We note that multiple such sets of l_{ijt} values can satisfy the condition, but we only need to pick one of them to prove the result. We also note that for each location i, l_{iit} is counted both as incoming load and outgoing load. With this, we can see that the starting load in a location is the sum total of the outgoing load. Also the net ending load at a location is the sum of the incoming load:

$$in_{jt} = \sum_{i=1}^{m} l_{ijt} \tag{18}$$

$$out_{jt} = \sum_{k=1}^{m} l_{jkt} \tag{19}$$

$$l_{jt} = out_{jt} \tag{20}$$

With this definition of l_{ijt} variables, we show that all the constraints 2 through 8 of LP_{perf} are satisfied:

- *Constraint 2 of LP_{perf}:*

$$\begin{aligned} & \sum_{i=1}^{m} l_{ijt} \\ & = in_{jt} + 0 && \text{Using Eq 18} \\ & = in_{jt} + l_{jt} - out_{jt} && \text{Using Eq 20 and expanding 0} \\ & \leq s_{jt} * p_j && \text{Using Eq 11 of } LP_{nperf} \end{aligned}$$

- *Constraint 3 of LP_{perf}:*

$$\sum_{k=1}^{m} l_{jkt} = l_{jt}, \forall j, t$$

This constraint is satisfied because of how we defined our l_{ijt} variables (as shown in equations 19 and 20).

- *Non-negativity constraints 4 through 8 of LP_{perf}:* Constraint 4 is satisfied by the way l_{ijt} were chosen. Constraints 5 through 7 can be directly ported from LP_{nperf}. Constraint 8 is also satisfied as δ is unlimited in this case.

So given S satisfies all the constraints of LP_{perf}, it is also a solution of LP_{perf}. Therefore, every solution S_{nperf} of LP_{nperf} can be reduced to a feasible solution S of LP_{perf}.

■

Given the proofs of Part 1 and Part 2 above, the two LP formulations LP_{perf} (for unlimited δ) and LP_{nperf} are equivalent.

C. Greedy Heuristic Algorithms

Optimal LP formulations may be computationally very expensive for large system sizes, so we also develop efficient heuristics. We describe below our heuristic algorithms that are inspired at a high level by greedy approximation algorithms to the the set-covering problem [6]. We loosely consider the load to be served as the set to be covered. The different amounts of load we can serve using energy generated from solar panels at various locations are like the subsets that can cover the original set.

1) Max Solar Per Panel Heuristic (MSP): We now define a greedy heuristic algorithm that performs comparably and runs faster than the LP. For example, for the no performance net-zero week case, the heuristics took a few seconds to complete, while the LP_{nperf} took over 2.25 hours, LP_{perf} for r=200kms ran for over 10.5 hours, and LP_{perf} for r=700kms did not complete even in 28 days. We use the same demand and supply matrices that we defined earlier in Section IV. In order to minimize the number of panels, we note that we need to assign as much load as we can to a location that has the highest solar output. This would help us to cover the maximum load with the minimum number of panels for a given time slot. We greedily pick the maximum solar per panel location across time and space (i.e. across all time periods and all data centers), and assign the entire load for the time period to that location. Using the solar per panel value and the load, we determine the number of panels to place at that location. Once we install panels at a location, these panels can then be used to serve demand for other time periods as well, so we accordingly

dcid/time	1	2...	n
1	p_{11}	p_{12}...	p_{1n}
2	p_{21}	p_{22}...	p_{2n}
3	p_{31}	p_{32}...	p_{3n}
..	..	..	..
..	..	..	..
m	p_{m1}	p_{m2}...	mn

TABLE III: Number of Panels Matrix

adjust the demand values to reflect the extra supply for all other time periods. We continue to place panels in this way until we satisfy the entire demand in all time periods. For pseudocode, please see Algorithm 1.

Algorithm 1 MSP Heuristic Pseudocode

```
function SPHEURISTIC( )
    time ← [t_1, t_2, ..., t_n]                        ▷ time periods
    spp ← [s_11, s_12, ..., s_mn]                      ▷ solar output
    load ← [l_1, l_2, ..., l_n]                        ▷ load for time period
    provpanels ← []                                    ▷ provisioned panels
    for t_y in time do
        s_xy ← min s_ij s.t. s_ij ∈ spp                ▷ pick min solar
        p_xy ← l_y/s_xy                                ▷ assign panels
        provpanels ← provpanels ∪ [p_xy]               ▷ add to provisioned panels
        for i in [1, 2, ..., m] do
            l_i ← |l_i − l_y|                          ▷ adjust other loads
        load ← load − l_y                              ▷ delete assigned load
    return provpanels
```

2) Min Number of Panels Heuristic (MNP): We now describe our second heuristic algorithm. The basic structure of this algorithm is the same as the MSP algorithm, except we now use a different heuristic to make a decision on where to place panels. We first determine the number of panels for each location for each time period, by dividing the load for the time period for all locations, by the solar per panel for the corresponding time and location. This gives us the 'Number of Panels Matrix' shown in Table III. We then pick the lowest number of panels value and install those many panels at the corresponding location and time period. Like before, once any panels are installed at a location, they are also available for other time slots. So we accordingly adjust the demand to reflect the extra supply. We recompute the number of panels matrix for the adjusted loads, and start over. We do this exercise until all the demand is met. Pseudocode for this algorithm is detailed in Algorithm 2.

V. Experimental Methodology

We conduct experiments on an extensive Akamai load trace spanning a month. The trace consists of load information from 100,592 servers in 724 global data center locations as shown in Figure 1. The dataset includes load, requests, and bytes served by each server every five minutes over a month-long trace. Further, it has detailed information about every data center, including the number of deployed servers, total server capacity, and the location of the data center including its latitude, longitude, city, state, and country.

Algorithm 2 MNP Heuristic Pseudocode

```
function NPHEURISTIC( )
    time ← [t_1, t_2, ..., t_n]                        ▷ time periods
    spp ← [s_11, s_12, ..., s_mn]                      ▷ solar output
    load ← [l_1, l_2, ..., l_n]                        ▷ load for time period
    origpanels ← []                                    ▷ original panels
    provpanels ← []                                    ▷ provisioned panels
    for s_ij in spp do
        op_ij ← l_i/s_ij                               ▷ determine original num panels
        origpanels ← origpanels ∪ [op_ij]              ▷ add to original panels
    for t_y in time do
        op_xy ← min op_ij s.t. op_ij ∈ origpanels      ▷ pick min panels
        p_xy ← l_y/s_xy                                ▷ assign panels
        provpanels ← provpanels ∪ [p_xy]               ▷ add to provisioned panels
        for i in [1, 2, ..., m] do
            l_i ← |l_i − l_y|                          ▷ adjust other loads
        load ← load − l_y                              ▷ delete assigned load
        time ← time − t_y                              ▷ delete time column
        origpanels ← []                                ▷ reset original panels
        for s_ij in spp do
            op_ij ← l_i/s_ij                           ▷ determine num panels
            origpanels ← origpanels ∪ [op_ij]          ▷ add to panels
    return provpanels
```

Parameter	Value
Loss %	14
System Capacity	0.275 kW
Module Type	Standard
Timeframe	Hourly
Azimuth	180 deg for northern hemisphere and 0 for southern
Tilt	Absolute value of latitude
Dataset	'TMY2' for US Locations and 'Intl' for others

TABLE IV: Parameters for PVWatts Data

For solar energy data, we use the PVWatts [22] hourly data of AC energy generation from solar radiation for a year. Assuming the power rating for solar panels is between 200 watts and 350 watts [7], we take an average value of 275 watts as the power rating per panel. Therefore, we use the system system capacity as the 0.275 kW for PVWatts in order to get the output generated by a single panel. For simplification, for all other required solar parameters, we use the values listed under 'Default Values' on page 3 in the PVWatts version 5 manual [23]. The required parameters used for downloading PVWatts data are detailed in Table IV.

The trace has 5-minute readings, whereas solar data is hourly. So we make an assumption that solar data does not change much during the hour and use an hour's reading for each of the five minutes that fall within that hour. Also, we have the solar data for the year, but we have the load trace for one month only. We assume that the load pattern for the CDN repeats monthly for the year. For each five minute interval, we convert the load and solar output power (in watts) into energy units (joules).

For baseline comparison, we use the number of panels we need to serve the IDN load without any load movement. E.g. for net-zero week, for each week, we divide the week's load for the data center by the corresponding week's sum of solar per panel values. We then determine the number of original panels by taking the maximum of all the weekly number of panels.

Through our experiments, we study different scenarios under which we move load to leverage solar: The first scenario enforces no performance constraints and allows unrestricted movement of load to take advantage of high levels of solar in different global locations. The second scenario restricts load movement within a certain radius, and we consider a number of such radii. Given load can end up in remote locations with an unlimited radius of load movement where it may not be possible to install solar panels, we also study the scenario where we restrict panel provisioing to larger data centers in major cities. We use the number of servers as an indicator for large data centers and constraint locations for panel provisioning to only be the top k locations sorted by the number of servers. In addition, for each scenario above, we repeat the study for net-zero week, net-zero month, and net-zero year.

With the above setup, our experimental evaluation seeks to answer the following questions for each of the three scenarios above, for each net-zero time period:

- How much of a reduction can we see in the number of panels when compared to not moving load anywhere to take advantage of high solar output?
- Which net-zero time period requires the most panels and why?
- How do the algorithms compare in their ability to reduce the number of panels?
- With performance constraints, how do the above results vary with different radii?
- With performance constraints, how does the number of locations selected vary with different radii of load movement?
- For the top k scenario, how to the results change with different radii?
- For the top k scenario, how does the number of locations selected vary with k?

VI. Empirical Results

The following paragraphs describe our results for scenarios where we restrict load movement within certain fixed radii (*imposing performance constraints*), where we allow unrestricted load movement (*without any performance constraints*), and when the panel provisioning is restricted to top k locations.

A. With and Without Performance Constraints

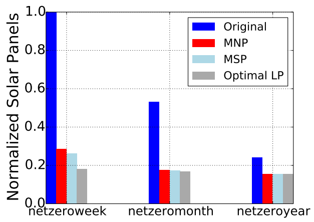
The goal of this experiment was to study the impact of load movement within a radius on the reduction in the number of panels when compared to the two extremes of unrestricted load movement and no load movement at all.

1) Number of Panels: Our results are shown in Figures 3(a) and 3(b) We list our observations below:

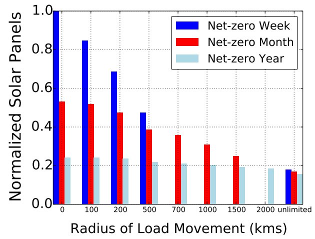
- *Heuristics perform comparably:* From Figure 3(a), we see that for an unlimited radius of load movement, the heuristic algorithms perform comparably with the LP.
- *Load movement helps reduce the number of required panels dramatically:* Figure 3(a) shows that with an unlimited radius, for net-zero year, number of panels decrease by 36% for all algorithms. For net-zero month, the MNP decreases panels by 66.94%, the MSP by 67.03%, and the LP by 68%. For net-zero week, the MNP decreases panels by 71.4%, the MSP by 73.7%, the LP by 82%. Figure 3(b) shows that even with performance constraints, we can see a significant reduction in the number of panels. For r=500kms, we see a reduction of about 9.7%, 27%, and 53% for net-zero year, month, and week respectively.
- *Number of panels is inversely proportional to size of net-zero time window:* Figure 3(b) shows that the number of panels provisioned is highest for net-zero week, followed by net-zero month, followed by net-zero year. This is intuitive given we are averaging over a larger time period for net-zero year as compared to net-zero month. In the case of a net-zero year, we must match demand with supply over the entire year. For net-zero month we must match demand with supply for *each* month, however, low our supply maybe and however high our demand may be for various months. Therefore, we must satisfy our net-zero condition for the 'worst' month in our list. Similarly for net-zero week. Therefore, the number of panels increase as we move from net-zero year, to net-zero month, to net-zero week.

2) Number of Locations: We study the number of locations where solar panels are provisioned by radii for different net-zero time periods. Our results are show in Figures 3(c) and 3(d). Our main observations are:

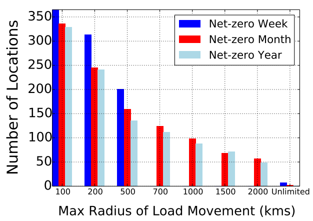
- *Number of locations decrease with increase in radius*: Figure 3(c) shows that with an increase in the radius of load movement, the number of locations where solar panels are allocated decreases for each net-zero time period. This is because as the radius increases, load converges to locations that are globally high for solar output. Figure 4 shows the locations that are chosen for various values of max radius for net-zero month. With load movement, we see the locations shrink and converge to the hubs for solar generation.
- *Number of locations is inversely proportional to the size of net-zero time period:* Figure 3(c) also shows that the number of locations are the maximum for net-zero week followed by net-zero month, and then net-zero year. Net-zero month and net-zero year are closer in the number of



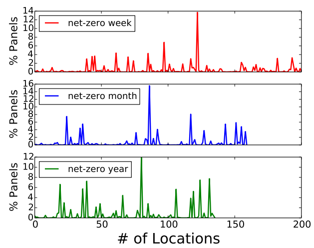
(a) For an unlimited radius, we see a dramatic reduction in the number of panels. Heuristics perform nearly as well as the optimal LP



(b) Number of panels decrease as the radius of load movement increases. Number of panels is also inversely proportional to the time period over which we aim to be net-zero



(c) Number of locations where panels are allocated decreases with increase in radius, because load converges to global locations with high solar output



(d) Distribution of panels for r=500kms for net-zero time periods. Number of locations is largest for net-zero week, then for net-zero month, and finally for net-zero year

Fig. 3: Results with and without performance constraints

locations they pick. Figure 3(d) shows the distribution of panels for a radius of 500kms for different net-zero time periods. We observe that for all the net-zero time periods, panels are fairly evenly distributed with a few high peaks lying between about 12% and 16%.

With unconstrained load movement we observe that, for all the algorithms, load is moved to locations that are high in solar output. However, with these choices we find that load can end up in remote locations where solar panel installation may not be feasible. To address this issue, we restrict solar panel installation to locations that have large data centers in major cities.

3) Understanding Location Choices: In this section, we discuss why the heuristic and optimal algorithms pick certain locations for net-zero year, net-zero month and net-zero week under unrestricted load movement. Overall, locations are picked for their high solar output regardless of other characteristics. However, the choices may seem non-intuitive at first given solar output must be considered for different time periods, and not just for the whole year. We explain these choices in detail in this subsection.

For net-zero year, all algorithms favor the location that has the maximum total annual solar output, and place all panels in that location. For the locations we considered, Arequipa, Peru was the location that topped the list for annual solar output. Therefore, as Table V shows, all the algorithms assigned the sum total of the load to Arequipa.

For net-zero month, the percentage break-up by locations is detailed in Table VI. We see that the algorithms tend to favor locations that have consistently high monthly solar output, with low variance. If we normalize monthly solar output values based on the max of all the monthly solar output values across all considered locations, we find that Arequipa performs quite well. As Figure 5(a) shows, the normalized monthly solar never falls below the 80% level, except slightly for the second month. Given the above, we observe that LP favors Arequipa and places over 96% panels there. The heuristic algorithms also place the majority of their panels (over 80%) in Arequipa.

Location	MSP	MNP	LP
AREQUIPA, Peru, South America	100	100	100

TABLE V: % Panels by location for net-zero year without performance constraints

Location	MSP	NP	LP
AREQUIPA, Peru, South America	80.05	84.01	96.07
ALBUQUERQUE, NM, United States, North America	1.97	2.26	0
PAROW, South Africa	14.23	5.43	0
JERUSALEM, Israel	3.75	1.57	0
YELLOWKNIFE, Canada	0	2.92	0
MONTEREY, CA, United States	0	3.81	0
PERTH, Australia	0	0	3.93

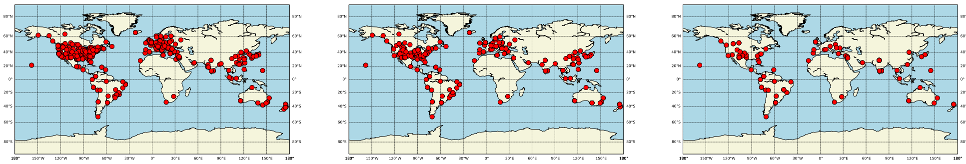
TABLE VI: % Panels by location for net-zero month without performance constraints

For net-zero week, the percentage break-up of the number of panels by location is detailed in Table VII. Locations that have a high weekly solar output tend to get picked. Figure 5(b) shows the normalized weekly solar output for Arequipa. We observe that for most weeks, Arequipa has a higher than 70% output. However, there are a few weeks where Arequipa does not do so well (e.g. for week 10 the output falls below 60%). The LP assigns more than half the panels to Arequipa. The heuristic algorithms, however, pick Winnipeg, Canada for assigning over 40% of panels. Winnipeg is not one of the top locations for annual solar output. However, it has extremely high solar output during one of its weeks. From this analysis, we learn that the LP tends to pick more robust locations that have consistently high solar output, where as the heuristic algorithms may pick locations that have a few weeks where their solar output is the maximum of any location.

B. Restricting Panels to Top K Data Centers

We use the number of servers in a data center as the proxy for data centers that are large and are located in non-remote places with large populations. We restrict panel provisioning to top k locations sorted by the number of servers. Figure 6 shows the change in the number of panels across different values of k. We list our observations below:

- *Number of panels provisioned varies inversely with k:* First of all, we see that the number of panels provisioned increases when we restrict ourselves to fewer locations. This is intuitive considering that we are operating with more constraints, and therefore are not able to extract as much reduction from solar output as we could in an

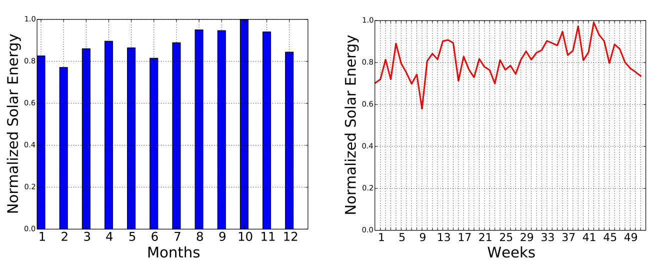


a) Max radius = 100 (b) Max radius = 500 (c) Max radius = 1500

Fig. 4: Locations where the LP with performance constraints places solar panels for net-zero month

Location	MSP	NP	LP
REGINA, Canada	4.09	1.54	0
AREQUIPA, Peru, South America	27.06	5.04	50.34
SANTIAGO, Chile	8.16	7.66	0
PAROW, South Africa	7.58	9.80	0
WINNIPEG, Canada	47.05	43.22	0
CANBERRA, Australia	6.05	0	0.31
LAGRANDE, OR, United States	0	0.53	0
BEND, OR, United States	0	0.61	0
LISBON, Portugal	0	0.36	0
RENO, NV, United States	0	1.62	0
YELLOWKNIFE, NT, Canada	0	4.14	0
JERUSALEM, Israel	0	0.03	0
EL PASO, TX, United States	0	0.18	0
SUDBURY, ON, Canada	0	0.14	0
BUENOS AIRES, Argentina	0	0.39	0
SAN LUIS OBISPO, CA, United States	0	1.21	0
CANBERRA, Australia	0	7.04	0
LA PAZ, Bolivia	0	0.85	0
THEBARTON, Australia	0	0.39	0
TUCSON, AZ, United States	0	0.91	0
SAINT GEORGE, UT, United States	0	0.49	0
MONTEREY, CA, United States	0	12.51	0
ALBUQUERQUE, NM, United States	0	1.66	0
PERTH, Australia	0	0	5.17
HAYS, KS, United States	0	0	7.06
ALAMOGORDO, NM, United States	0	0	14.78
PRETORIA, South Aftrica	0	0	14.62
CEDAR CITY, UT, United States	0	0	7.73

TABLE VII: % Panels by location for net-zero week without performance constraints



(a) Plot shows that Arequipa, Peru has a high solar output consistently across all the months of the year

(b) Plot shows that Arequipa, Peru has a high solar output for most weeks of the year

Fig. 5: Plots showing Arequipa monthly and weekly solar output

unconstrained setting. The larger the k, the more locations are in play for extracting solar savings.

- *Number of panels vary inversely with the size of the net-zero time window:* Once again, we see that the number of panels provisioned is the most for net-zero week, followed by net-zero month, and finally net-zero year. This trend is preserved across different values of k. See Section VI-A1 for a detailed explanation.
- *k=500 balances both panel optimization and top-k requirements:* For k=500, the number of panels are very close to the no performance panel provisioning scenario. Therefore, restricting to the top 500 data centers is a good middle ground for installing a near-optimal number of panels at non-remote locations.
- *Discussion on location choices under top k restrictions:* As expected, with top k restrictions, we find that the locations selected for the majority of panels with smaller values of k tend to be larger cities. E.g. for k=5 net-zero year selects Dallas, TX versus for k=1200, the location picked is Arequipa, Peru. Similarly, for net-zero month, for k=5, the location picked is Dallas, TX again and for k=1200, Arequipa and Perth are picked. For net-zero week, for k=5, Atlanta is picked, and for k=1200, we see almost half of the panels provisioned in Arequipa. Tables VIII, IX and X show the details of the locations selected.

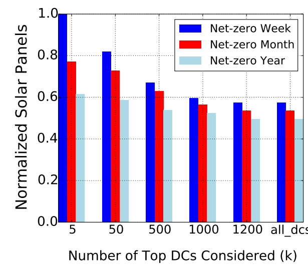


Fig. 6: Number of panels provisioned are highest for net-zero week, followed by net-zero month and then net-zero year

K	Location	Percentage Panels
5	DALLAS, TX, United States, North America	100.0
50	LOSANGELES, CA, United States, North America	100.0
500	SCOTTSDALE, AZ, United States, North America	100.0
1000	HENDERSON, NV, United States, North America	100.0
1200	AREQUIPA, Peru, South America	100.0

TABLE VIII: Locations under Top K Restrictions for Net-zero Year

VII. Related Work

Recently, there has been a lot of research on reducing or greening energy consumption in data centers, including solutions focused on shutting down servers or clusters during

K	Location	Percentage Panels
5	DALLAS, TX, United States, North America	100.0
50	MIAMI, FL, United States, North America	88.83
	LOSANGELES, CA, United States, North America	11.17
500	SCOTTSDALE, AZ, United States, North America	61.08
	AUCKLAND, New Zealand, Oceania	34.
	SYDNEY, NSW, Australia, Oceania	4.37
1000	RANDBURG, South Africa, Africa	51.80
	LASVEGAS, NV, United States, North America	31.88
	PERTH, WA, Australia, Oceania	16.32
1200	AREQUIPA, Peru, South America	96.07
	PERTH, Australia, Oceania	3.93

TABLE IX: Locations under Top K Restrictions for Net-zero Month

K	Location	Percentage Panels
5	ATLANTA, GA, United States, North America	100.0
50	DALLAS, TX, United States, North America	50.4
	LOSANGELES, CA, United States, North America	49.66
500	RIODEJANEIRO, RJ, Brazil, South America	31.36
	SCOTTSDALE, AZ, United States, North America	21.01
	SANTOS, SP, Brazil, South America	13.14
	LOSANGELES, CA, United States, North America	11.88
	AUCKLAND, New Zealand, Oceania	9.55
	CAIRO, Egypt, Africa	8.62
	SYDNEY, NSW, Australia, Oceania	3.06
	DENVER, CO, United States, North America	1.38
1000	DUBAI, United Arab Emirates, Asia	39.40
	PERTH, Australia, Oceania	19.46
	RANDBURG, South Africa, Africa	16.49
	AMMAN, Jordan, Asia	10.08
	LASVEGAS, NV, United States, North America	7.07
	BUENOSAIRES, Argentina, South America	3.27
	ADELAIDE, SA, Australia, Oceania	2.15
	BRISBANE, QLD, Australia, Oceania	1.58
	SCOTTSDALE, AZ, United States, North America	0.50
1200	AREQUIPA, Peru, South America	50.20
	ELPASO, TX, United States, North America	14.41
	RANDBURG, South Africa, Africa	11.80
	SAINTGEORGE, UT, United States, North America	11.75
	PERTH, WA, Australia, Oceania	6.87
	LASVEGAS, NV, United States, North America	2.70
	CANBERRA, ACT, Australia, Oceania	2.28

TABLE X: Locations under Top K Restrictions for Net-zero Week

off-peak periods and/or using low-power consumption states instead of powering them off in order to prevent wear and tear [15] [18] [5] [25]. These solutions provide significant savings in energy, they do not deal directly with the use of green energy to power data centers. Separately, a lot of work has been done in the area of renewables for data centers. Previous work has also looked into providing a solution for selecting sites for and provisioning green data centers using a follow-the-renewables approach [4]. However, their work focuses on setting up a data centers from scratch, where as our work explores how to use renewables to green and existing IDN. Work has also been done on job scheduling within a data center based on predicted solar and brown energy prices [11] [10]. Previous work has also modeled the potential of using renewable energy for data centers located in colder locations [26], while [13] propose a solution for data center expansion using modular solar panels and distributed battery systems to have near-zero environmental impact. While this work takes advantage of renewables to reduce energy consumption, it does not deal with efficient provisioning of solar panels for an existing global IDN. Moving load across data centers to increase the use of renewables has been studied before. Studies have also been done [17] [16] on how and to what extent geographical load balancing can encourage use of renewable energy and reduce the use of brown energy. Their distributed algorithm offers significant savings in cost (including energy cost and delay cost). Work has also been done on green solutions that control user traffic and direct it to different data center locations based on changes workload and carbon footprint [9]. However, all these works [17] [16] [9] do not focus on efficient provisioning of solar panels.

VIII. Conclusions

We studied the problem of efficient solar panel provisioning for net-zero IDNs. Using our heuristic and optimal algorithms, we are able to significantly reduce the number of solar panels needed for creating net-zero IDNs. Overall, with unrestricted radius of load movement, for net-zero year, we can reduce the number of solar panels by 36%, between 67% to 68% for net-zero month, and for net-zero week about 71% and 74% using heuristics, and 82% using the optimal. We also show that if we restrict the radius of load movement, we can achieve a significant reduction in the number of panels for all net-zero time periods. For instance, for r=500kms, we see a reduction of about 9.7%, 27%, and 53% for net-zero year, month, and week respectively. We also saw that restricting to the top 500 data centers is a good middle ground for achieving near-optimal number of panels installed at major data center locations. In conclusion, we demonstrated that by leveraging locations with high solar output, we can significantly reduce the number of panels needed to serve load and achieve net-zero status for global IDNs. As part of future work, we plan to study if we can combine renewables like solar and open air cooling to develop even more efficient solutions for greening IDNs.

References

[1] *'Apple Data Centers'*. https://images.apple.com/environment/pdf/Apple_Environmental_Responsibility_Report_2017.pdf.

[2] *'Google Data Centers'*. https://www.google.com/about/datacenters/renewable/.

[3] L. Barroso and U. Holzle. The case for energy-proportional computing. *IEEE Computer*, 2007.

[4] Josep Lluis Berral, Iñigo Goiri, Thu D. Nguyen, Ricard Gavaldà, Jordi Torres, and Ricardo Bianchini. Building green cloud services at low cost. In *IEEE 34th International Conference on Distributed Computing Systems, ICDCS 2014, Madrid, Spain, June 30 - July 3, 2014*, pages 449–460, 2014.

[5] Yiyu Chen, Amitayu Das, Wubi Qin, Anand Sivasubramaniam, Qian Wang, and Natarajan Gautam. Managing server energy and operational costs in hosting centers. In *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '05, pages 303–314, New York, NY, USA, 2005. ACM.

[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, third edition, 2009.

[7] *Solar Panel Power Rating*. https://www.energysage.com/solar/buyers-guide/solar-panel-efficiency/.

[8] M. Arlitt et al. Towards the design and operation of net-zero energy data centers. In *ITherm*, 2012.

[9] Peter Xiang Gao, Andrew R. Curtis, Bernard Wong, and Srinivasan Keshav. It's not easy being green. *SIGCOMM Comput. Commun. Rev.*, 42(4):211–222, August 2012.

[10] Íñigo Goiri, William Katsak, Kien Le, Thu D. Nguyen, and Ricardo Bianchini. Parasol and greenswitch: Managing datacenters powered by renewable energy. *SIGARCH Comput. Archit. News*, 41(1):51–64, March 2013.

[11] Íñigo Goiri, Kien Le, Md. E. Haque, Ryan Beauchea, Thu D. Nguyen, Jordi Guitart, Jordi Torres, and Ricardo Bianchini. Greenslot: Scheduling energy consumption in green datacenters. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 20:1–20:11, New York, NY, USA, 2011. ACM.
[12] J. Koomey. Growth in data center electricity use 2005 to 2010. 2010.
[13] Chao Li, Yang Hu, Juncheng Gu, Yuan Jingling, and Tao Li. Oasis: Scaling out datacenter sustainably and economically. In *IEEE Transactions on Parallel and Distributed Systems, vol. , no. , pp. 1, 5555*, 2016.
[14] M. Lin, A. Wierman, L. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *Proc. IEEE INFOCOM*, 2011.
[15] Minghong Lin, Adam Wierman, Lachlan L. H. Andrew, and Eno Thereska. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Trans. Netw.*, 21(5):1378–1391, October 2013.
[16] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H. Low, and Lachlan L. H. Andrew. Geographical load balancing with renewables. *SIGMETRICS Performance Evaluation Review*, 39(3):62–66, 2011.
[17] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H. Low, and Lachlan L.H. Andrew. Greening geographical load balancing. In *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '11, pages 233–244, New York, NY, USA, 2011. ACM.
[18] Vimal Mathew, Ramesh K. Sitaraman, and Prashant J. Shenoy. Energy-efficient content delivery networks using cluster shutdown. In *International Green Computing Conference, IGCC 2013, Arlington, VA, USA, June 27-29, 2013, Proceedings*, pages 1–10, 2013.
[19] E. Nygren, R.K. Sitaraman, and J. Sun. The Akamai Network: A platform for high-performance Internet applications. *ACM SIGOPS Operating Systems Review*, 44(3):2–19, 2010.
[20] S. Pelley, D. Meisner, T. Wenisch, and J. VanGilder. Understanding and abstracting total data center power. In *Workshop on Energy-Efficient Design*, 2009.
[21] *PV Education Solar Cell*. http://www.pveducation.org/pvcdrom/solar-cell-structure.
[22] *PVWatts*. https://developer.nrel.gov/docs/solar/pvwatts-v5/.
[23] *PVWatts V5 API*. http://www.nrel.gov/docs/fy14osti/62641.pdf.
[24] A. Quereshi. Power-demand routing in massive geo-distributed systems. 2010.
[25] Navin Sharma, Sean Barker, David Irwin, and Prashant Shenoy. Blink: Managing server clusters on intermittent power. *SIGARCH Comput. Archit. News*, 39(1):185–198, March 2011.
[26] Enida Sheme, Neki Frashri, Simon Holmbacka, Sbastien Lafond, and Drazen Luzanin. Datacenters powered by renewable energy: A case study for 60 degrees latitude north. In *24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, page 15, 2016.
[27] *The Average PUE is 1.8: Uptime Institute*. http://bit.ly/1eEqN9d.