Optimal Design of Efficient Rooftop Photovoltaic Arrays

Madeleine Udell and Oliver Toole

October 12, 2017

Abstract

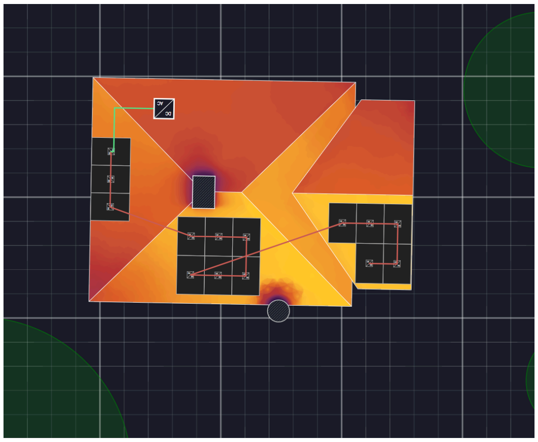
This paper addresses a major challenge in the residential solar industry: automated design of cost-effective, efficient rooftop photovoltaic (PV) installations. Optimal designs choose system components, locations, and wiring to minimize cost while meeting desired energy output and complying with all physical and legal constraints. We present a novel lower bound for the energy produced by a PV installation, which admits efficient optimization via integer linear programming. The resulting algorithm can design systems with a variety of solar hardware, including microinverters, string inverters, and DC optimizers, and optimize for complex shading patterns. Prior to our work, solar installers designed PV installations by hand. Our algorithm automates PV design using OR techniques, and has been used to design over 10,000 installations.

We compare the performance of our optimal designs to designs produced by solar installation experts at the National Renewable Energy Laboratory (NREL). Our algorithm designs faster, cheaper, more energy efficient installations than expert installers, producing designs in tens of seconds where experts require tens of minutes. The optimized designs deliver the required energy output at lower cost in more than 70% of cases, and on average increase the energy produced per dollar invested. These results indicate that US rooftop solar PV installations could produce more than 2% more energy at the same installation cost, or 820 GWh more energy per year.

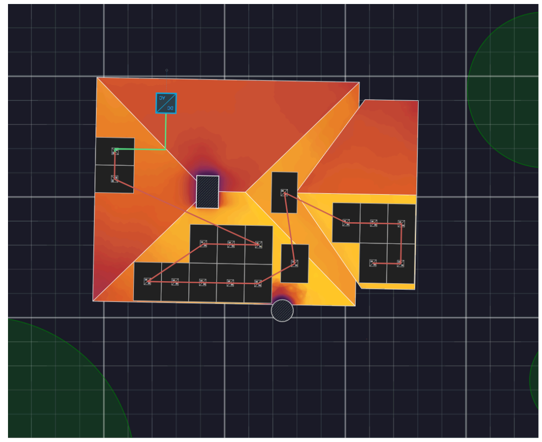
1 Introduction

Distributed production of solar power using photovoltaic (PV) arrays is one compelling option for long term sustainable energy. For many households, the amount of energy incident on their roof each year is more than sufficient to power the household's energy needs Gagnon et al. (2016). Distributed production using rooftop PV arrays reduces the losses associated with power transmission, provides increased energy security to households, and allows households to supplement their income by selling energy back to the grid.

This paper addresses a major challenge in the residential solar industry: automated design of cost-effective, efficient rooftop PV installations. Consider the designs shown in Figure 1. Here the colors shown represent annual solar irradiance: brighter areas (yellows) correspond to higher levels of insolation (ground-level solar radiation), while darker areas (reds to purples) correspond to lower levels. Black rectangles with white squares denote solar panels, and other black shapes show obstructions, which are generally surrounded by a shaded (often purple) area. Panels joined on the same string are connected by a red line, and are connected to a PV inverter by a green line.



(a) Optimized design for sample site in Denver, CO. The cost per kWh is 0.95. DC optimizers are chosen to handle the complex shading. Three nice looking arrays maximize available roof area.



(b) User design for sample site in Denver, CO. The cost per kWh is 1.01. This design is infeasible because some panels overlap ridges. There is also an islanded panel.

Figure 1: Two photovoltaic array designs for sample site in Denver, CO.

We can see that the design produced by an expert installer at the National Renewable Energy Laboratory (NREL) (Figure 1b) has some modules that are not adjacent to any other (resulting in a higher installation cost), and which do not lie entirely on a single roof face (which is not physically allowed). In contrast, the optimal design produced by our algorithm (Figure 1a) supplies the equivalent energy at lower cost and satisfies all constraints.

The PV design problem presents six essential challenges.

1. What kinds of panels and inverters should be used?
2. How many of each component are needed?
3. Where should they be placed?
4. How should they be wired together?
5. How can we estimate (and maximize) the energy produced?
6. How can we estimate (and optimize) the cost of the installation?

This problem is (provably) NP-hard. Yet previously, solar installers had no alternative than to solve these problems by hand, resulting in inefficient designs. To increase the number of photovoltaic installations in the US, while increasing quality and lowering cost, an automated approach to design is critical:

- *Human designs take time.* As the solar industry has matured, hardware costs of solar installations have dropped. By 2012, non-hardware "soft costs", including permitting, financing, and customer acquisition, comprised 64% of the total installation cost for residential solar systems Friedman et al. (2013). Automated system design allows solar installers to quickly assess the economic viability of a potential installation and present a plan to customers, which reduces the soft cost of an installation.

- *Human designs violate constraints.* Many designs created even by experts violate the laws of physics or of man and result in dangerous, nonsensical, or illegal PV systems. For example, our solar installation experts designed systems with inverters too small to cope with the power or current produced by the panels; and placed panels too close to eaves or ridges or chimneys, in violation of fire codes.

- *Humans designs can be improved.* Solar installation experts often fail to consider important but challenging aspects of these PV design problems. For example, they use expensive parts where cheap parts could produce just as much energy; and use heuristics instead of considering the details of the problem, placing panels on a south roof face (to maximize total irradiance) when a western facing array (producing power in peak hours) would be more profitable.

1.1 Contributions

We approach the PV design problem using the fundamental tools of operations research: mathematical modeling, simulation, and optimization.

- We mathematically model the PV design problem for residential solar arrays, and developed a tractable LP-representable approximate formulation.
- We design a fast optimization algorithm, the AutoDesigner, to solve the PV design problem. The AutoDesigner approximates the PV design problem by one that can be globally solved by modern mixed integer linear programming (MILP) solvers, and then refines the approximation to converge to a satisfactory solution.
- We use a detailed simulation of the energy produced by the array to check and refine the linear model, ensuring that the approximation converges to a satisfactory solution.

1.2 Impact

The AutoDesigner algorithm has been implemented by a major solar software provider, Aurora Solar, and is available to all of their customers as a product called the AutoDesigner. It has been used by solar installers to design more than 10,000 installations, saving time, money, and energy. Whereas human experts require about an hour to create a design, the AutoDesigner takes less than a minute.

A third-party comparison of our method with the designs produced by human experts found that the AutoDesigner outperforms hand-tuned designs for more than 70% of test cases, and on average increases the energy produced per dollar invested (Freeman and Simon

2015). Compared to designs produced by human experts, our automated designs are more cost efficient, can be computed more quickly, and consistently respect physical and legal constraints. These results indicate that US rooftop solar PV installations could produce more than 2% more energy at the same installation cost, or 820 GWh more energy per year.

1.3 Related work

While the literature on PV system optimization is large, the AutoDesigner is, to our knowledge, the first automated design tool for rooftop PV arrays.

Rekioua and Matagne (2012) detail techniques used for optimization of solar PV systems, and provide a foundation for our approach.

A major focus of our paper is on designing arrays that function well under conditions of variable shading. Massalha and Appelbaum (2016) consider the effect of variable shading of modules inside solar panels arranged in rows. Nguyen and Lehman (2008) propose a real-time control algorithm to reconfigure PV arrays in order to reduce the effect of variable shading; Patnaik et al. (2011) propose a similar strategy for individual solar panels. In contrast to those proposals, which use a specialized switching circuit, our designs use only standard hardware that is common in modern rooftop PV installations. A substantial literature is devoted to the design of efficient maximum power point trackers (MPPTs); see Esram and Chapman (2007) for a review. Here, we assume that each inverter is equipped with an MPPT, and do not consider the problem of MPPT design.

Our approach makes heavy use of modern optimization techniques, including the approximation of complex non-linear objective by a piecewise linear objective, mixed integer linear programming, bisection search for black-box optimization, and model reduction via clustering. For the reader unfamiliar with these techniques, we recommend the discussion of bisection methods for quasiconvex optimization in Boyd and Vandenberghe (2004), and the introduction to linear optimization techniques in Bertsimas and Tsitsiklis (1997).

Below, we provide definitions of the important hardware components of modern rooftop PV installations. We recommend Parida et al. (2011) for a more thorough treatment.

2 PV arrays

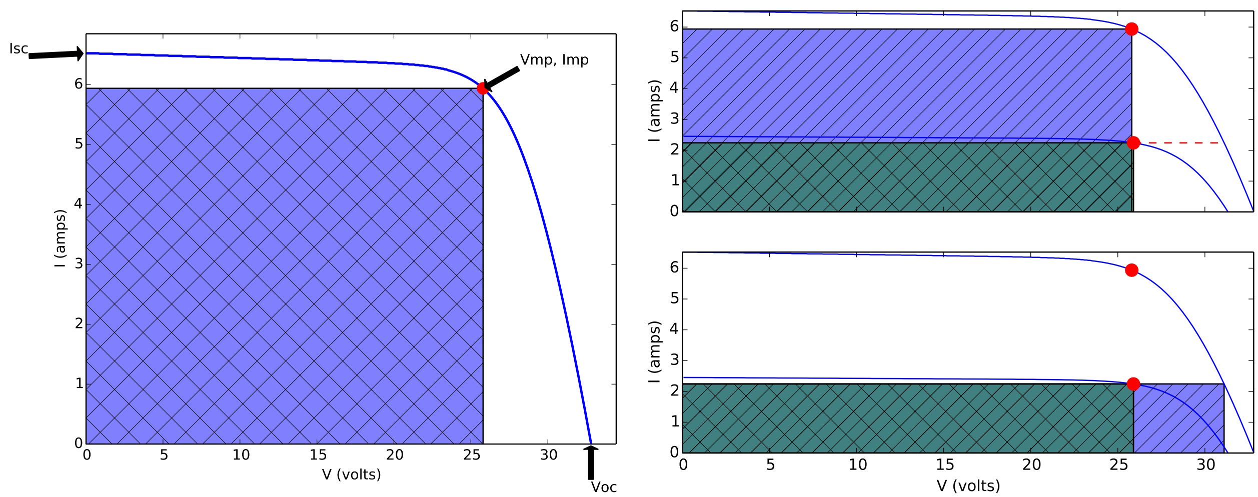
In this section we describe the characteristics of PV arrays and how design choices affect the energy and cost of these arrays. Much of this material is standard; see Rekioua and Matagne (2012) for an introductory treatment. The electronic components encompassed by our model include solar panels, string inverters, microinverters, and DC optimizers.

We leave to future work the problem of finding optimal solar arrays featuring batteries, bypass diodes, and other more advanced electronics, as they are less commonly used in practice and lead to very different modeling challenges from the ones we focus on here. Our choice to model only solar panels, string inverters, microinverters, and DC optimizers ensures that each I-V curve has a unique optimum, but makes the design problem harder, since advanced electronics can reduce the impact of shading on solar array performance.

2.1 PV panels

Solar panels are the building blocks for solar arrays. For a given irradiance and temperature, a solar panel can operate at a range of currents and voltages. This range is best visualized with an I-V curve as shown in Figure 2a (De Soto et al. 2006). The panel can operate at any point (I, V) on the curve; at any point, the energy produced is the product of the current I and the voltage V.

There are four notable points on each I-V curve. The short circuit current I_{sc} is the current through the panel when the voltage is zero. The open circuit voltage V_{oc} is the voltage across the panel when the current is zero. The current I_{mp} and voltage V_{mp} give the *maximum power point* of the panel, the current and voltage at which the panel produces the most power. A useful approximation to the behavior of PV panels states that as irradiance increases, so does I_{sc}; and as temperature increases, V_{oc} decreases (but only slightly) (De Soto et al. 2006). When a *string* of panels is connected in series, the voltage across the string increases with the number of panels. When strings of panels are connected in parallel, the current at the output increases with the number of strings. The orientation of a panel relative to the sun determines the effective irradiance on the panel. To produce maximum power, the normal to the plane of the panel should point directly at the sun.



(a) Typical current vs voltage (I-V) curve

(b) Current vs voltage (I-V) curves for a shaded (upper curve) and unshaded (lower curve) panel. Shaded areas represent the maximum energy produced (upper plot) and lower bound on the energy produced (lower plot).

Figure 2: Current vs voltage (I-V) curves and bounds on energy produced.

2.2 PV inverters

The simplest PV array consists of a collection of solar panels connected to inverters that convert the DC energy produced by a solar panel into AC energy that can be consumed locally or by the electric grid. The energy produced by an array is determined by the irradiance incident on the panels, the panel efficiency, and the efficiency of the inverters.

Three different inverter configurations are conventionally used to wire together inverters with solar panels, which have remarkably different efficiency characteristics.

Microinverters. Microinverter configurations use a separate inverter for each solar panel to convert the power from each panel individually. The power lost in this conversion is generally less than 2%. However, the cost of purchasing a separate inverter for each panel is often prohibitive.

String inverters. String inverter configurations use a single inverter to convert the power from an array of panels. The simplest array consists of panels wired together in series. More complex arrays connect multiple strings to the same inverter. To understand how this affects efficiency, it is important to understand how inverters control their inputs.

Each inverter contains one (or more) power electronic devices called maximum power point trackers (MPPT) that can adjust the current flowing through them. The MPPT is wired in series with one (or more) strings of panels so the current through each panel is controlled by the current through the MPPT. The voltage across each panel is determined by the incident irradiance and the current through the panel according to the characteristic I-V curve for that panel. The DC power produced by the panel is the product of the current and voltage across the panel. The MPPT adjusts the current in order to produce the maximum instantaneous power possible for the array given the irradiance on each panel.

Figure 2b illustrates this relation. It shows the current vs voltage (I-V) curve for two panels, one unshaded (the upper curve) and one shaded (the lower curve). The power produced by a single panel is the product of a current and voltage pair lying on that panel's I-V curve. The MPPT finds the current (red dot) that maximizes the power in the string, which is given by the product of the current and voltage chosen, represented as the shaded area on the plot. The upper plot shows the current voltage pair producing the maximal power for each panel independently. The lower plot shows that the optimal current for the shaded panel is feasible for the unshaded panel, and produces at least as much power in the unshaded panel as in the shaded one. This observation gives rise to an approximation to the true maximum power produced by a string of panels: the power produced is always greater than the minimum power in any panel, times the number of panels. We make use of this approximation below.

DC optimizers. DC optimizer configurations use power electronics mounted on each panel to maximize the energy produced by each string by separately controlling the voltage on each panel. The power electronics used on each panel are cheaper than microinverters while producing similar efficiencies even on heavily shaded roofs, making them an attractive option. These units began to appear in residential installations around 2009 (Grana and Shiao 2014), but are still quite expensive compared to traditional string inverter configurations. By

2013, DC optimizer configurations represented only 2.5% of total installed rooftop PV arrays (Grana and Shiao 2014).

3 Problem statement

In this section, we will define the components of the problem our algorithm solves to produce a PV design. In brief, the *PV design problem* is to find a valid PV array design meeting a certain desired energy and minimizing the cost of the array. A valid design obeys a variety of constraints on the panel locations and on the wiring of the panels to the inverters. We describe our model for these constraints, for the energy produced by a design, and for the cost of a design in the remainder of this section, and defer a formal problem description for now.

Other forms of the optimization problem may also be interesting. For example, one might want to maximize energy while controlling cost. It is easy to solve the maximum energy problem using just a logarithmic (in the inverse error tolerance) number of calls to a black-box solver for the minimum cost problem (Aravkin et al. 2016).

In this paper we will concentrate on the minimum cost problem formulation for simplicity.

3.1 Location constraints

Solar panels generally should not overlap, and are usually placed together in regular arrays. Panels may only be placed in certain areas of roofs to obey fire codes and other zoning constraints. Usually, these restrictions prevent panels from being placed too close to a roof edge or to an obstruction such as a chimney. In this paper, we restrict panels to be placed parallel to the roof face, and leave to future work the complications of panels with compound tilt.

3.2 Inverter constraints

Each inverter has a safe operating range of currents I, voltages V and powers P given by constraint sets

$$(I, V, P) \in \mathcal{C}_i. \tag{1}$$

Inverter manufacturers specify the constraint sets $\mathcal{C}_i$ by defining maximum and minimum values for the current, voltage and power, so that the sets $\mathcal{C}_i$ may be specified in the form of affine inequality constraints on I, V, and P.

To ensure safety of the PV system, a design should never connect more panels and strings to an inverter than it can safely handle even at maximal irradiance. Hence these constraints restrict the number of panels and strings that can be served by a single inverter.

3.3 Energy output

The power produced by a PV array can be calculated by computing the maximum power point for each MPPT in the array given the irradiances on the panels, and summing the powers produced in each panel at that current and irradiance. This function is difficult to compute, since finding the MPPT requires solving a partial differential equation (PDE) (Rekioua and Matagne 2012). We do not address the problem of computing the MPPT in this paper. Instead, our algorithm relies on another method to compute the power produced by a given PV array. Our method requires access to these values but not to gradients of the value, which allows us to use any reliable PDE solver in our optimization routine. In numerical results below, we use an simulator developed at Aurora Solar to solve this PDE. This simulator has been independently evaluated by the National Renewable Energy Laboratory (NREL) and found to perform within 6% (or $\leq 2\%$ if we discard one outlier test site) of the true values on standard installations (Freeman and Simon 2015).

3.4 Cost

In this paper we approximate the cost of a PV array as the sum of the costs of the inverters, panels, and DC optimizers chosen. Other forms for the cost objective are worth considering: for example, inverters last around 10 years, while solar panels can last up to 30 years with today's technology. So tripling the cost of the inverters might provide a more accurate estimate of the lifetime capital cost of the installation. This change increases the incentive to find designs that save money on inverters, which is often achieved by selecting string inverters in place of microinverters.

We neglect installation cost (including labor) and the cost of wires and other system components, which usually form a very small part of the cost of residential solar installations. Modeling and optimizing designs for these costs is an interesting challenge we defer to future work.

4 Algorithm

The algorithm presented here addresses each of the five difficulties.

1. Enumerate the panels and inverters that can be used in conjunction; encode each as an indicator variable.

2. Encode capacity constraints on each system component as a linear constraint to determine how many are needed.

3. Use an indicator variable that matches each solar panel to a string on some roof face to encode wiring decisions.

4. Estimate the energy produced by a particular wiring configuration using a simple linear lower bound.

5. Estimate the cost of the installation as an affine function of the indicator variables corresponding to the panels and inverters chosen.

That is, using a linear lower bound on the energy produced by a design described below, the PV design problem can be encoded as a MILP that computes a feasible design with energy *at least as great* as the input desired energy. This design MILP is described below. To refine the design when the true energy produced is excessive, we use a bisection method presented as Algorithm 1.

Algorithm 1 Bisection algorithm for PV array optimization

input: problem data, convergence tolerance ϵ
Choose potential panels and inverters.
Set $\bar{E}^{\text{des}} \leftarrow E^{\text{des}}$, $\bar{E}^{\text{high}} \leftarrow E^{\text{des}}$, $\bar{E}^{\text{low}} \leftarrow 0$.
repeat
  Solve design MILP (5) to determine assignment of panels to faces, string lengths to faces, string lengths to inverters, and wiring of panels to strings.
  Simulate design to determine true energy output $\hat{E}$ of design.
  if $\hat{E} \geq E^{\text{des}}$ **then**
    $\bar{E}^{\text{high}} \leftarrow \bar{E}^{\text{des}}$
  else
    $\bar{E}^{\text{low}} \leftarrow \bar{E}^{\text{des}}$
  end if
  $\bar{E}^{\text{des}} \leftarrow \frac{1}{2}(\bar{E}^{\text{low}} + \bar{E}^{\text{high}})$
until $\bar{E}^{\text{high}} - \bar{E}^{\text{low}} < \epsilon$
Refine design with local search.
output: strings of panels, inverter wiring

In the remainder of this section we describe each of the steps in Algorithm 1 in turn.

The design MILP (5) always underestimates the energy that will be produced by a given stringing of those panels, so $\bar{E}^{\text{high}}$ and $\bar{E}^{\text{low}}$ always produce designs with true output energy bracketing E^{des}. By adjusting the desired aggregate output of the panels used in the assignment MILP, this algorithm converges to a design with energy close to E^{des}.

The design space of the problem is discrete, so we cannot expect to achieve a design producing energy *exactly* E^{des}. Instead, when $\bar{E}^{\text{high}}$ and $\bar{E}^{\text{low}}$ are sufficiently close, all input energies to the MILP in between these two values will produce the same design as either $\bar{E}^{\text{high}}$ or $\bar{E}^{\text{low}}$. For example, if the set of panels $\mathcal{P}^{\text{low}}$ are used in the design that produces energy $\bar{E}^{\text{low}}$ and panel p is the next best panel, then we often eventually find that the following equality holds:

$$\bar{E}^{\text{low}} + e_p = \bar{E}^{\text{high}},$$

where e_p is the annual energy produced by the next best panel p. Hence no energy between $\bar{E}^{\text{low}}$ and $\bar{E}^{\text{high}}$ can be achieved without perturbing the locations of the panels. Alternatively, we could move panels into different (suboptimal) locations to hit E^{des} exactly, but that would lower the efficiency (in \$/kWh) of the design.

Notation. To avoid confusion, we use a consistent notation for indexing panels, inverters, roof faces, strings, and time. Panels are indexed by $p \in \mathcal{P}$, inverters are indexed by $i \in \mathcal{I}$, roof faces are indexed by $f \in \mathcal{F}$, strings are indexed by $s \in \mathcal{S}$, and time is indexed by $t \in \mathcal{T}$. Feasible string lengths for inverter $i \in \mathcal{I}$ are indexed by $l \in \mathcal{L}_i$. The feasible string lengths will generally be a range of integers of the form $\mathcal{L}_i = \{m_i, \ldots, M_i\}$ where $m_i \in \mathbf{Z}$ and $M_i \in \mathbf{Z}$ are determined by the minimum and maximum allowable voltage for inverter i. The set of all allowable string lengths in the installation $\mathcal{L}$ satisfies

$$\mathcal{L} = \cup_{i \in \mathcal{I}} \mathcal{L}_i.$$

The set of panels on face f is denoted by $\mathcal{P}_f$; the sets $\mathcal{P}_f$ form a partition of $\mathcal{P}$.

We will use these indices to differentiate among variables and parameters corresponding to different system components. For example, e_{pt} will be the energy of panel p at time t, while e_{st} will be the energy of string s at time t.

4.1 Potential panels and inverters

We assume that a set of panel and inverter types is given as input to the algorithm.

Potential panels. The design MILP requires as input a set of potential panels $p \in \mathcal{P}$. A potential panel p is a panel of a specified type at a specified location on a roof. The complexity of the design MILP increases with the number of potential panels $|\mathcal{P}|$, but the MILP will be infeasible (or produce a poor solution) if too few panels are allowed. Most residential designs use a single panel type, and arrange panels on each roof face in a regular array; other designs are generally considered ugly. Hence to produce a set of potential panels $\mathcal{P}$, our algorithm chooses a single type of panel and tiles each roof face so as to fit as many panels as possible in a regular array. Examples are shown in Figure 3.

Potential inverters. A potential inverter is an instance of an inverter of a specified type. The set of potential inverters $\mathcal{I}$ is chosen to contain as many replicas of each inverter type as may be necessary to service the PV array. The complexity of the design MILP increases with the number of inverters $|\mathcal{I}|$, but the MILP will be infeasible (or produce a poor solution) if too few inverters are allowed.

We pick inverters to use by solving a very simple optimization problem. For each type of inverter i, let the variable n_i be the number of inverters of type i, and recall that m_i is the minimum length of a string connected to an inverter of type i. Let n_p be a lower bound on the number of panels needed to meet the desired energy constraint $\bar{E}^{\mathrm{des}}$. We can find a set of inverters needed to service n_p panels by solving the following optimization problem:

$$\begin{array}{lll} \text{minimize} & \sum_{i \in \mathcal{I}} c_i n_i & \text{(cost)} \\ \text{subject to} & n_p \leq \sum_{i \in \mathcal{I}} n_i m_i & \text{(capacity)} \end{array} \tag{2}$$

with variables $n_i \in \mathbb{Z}$.

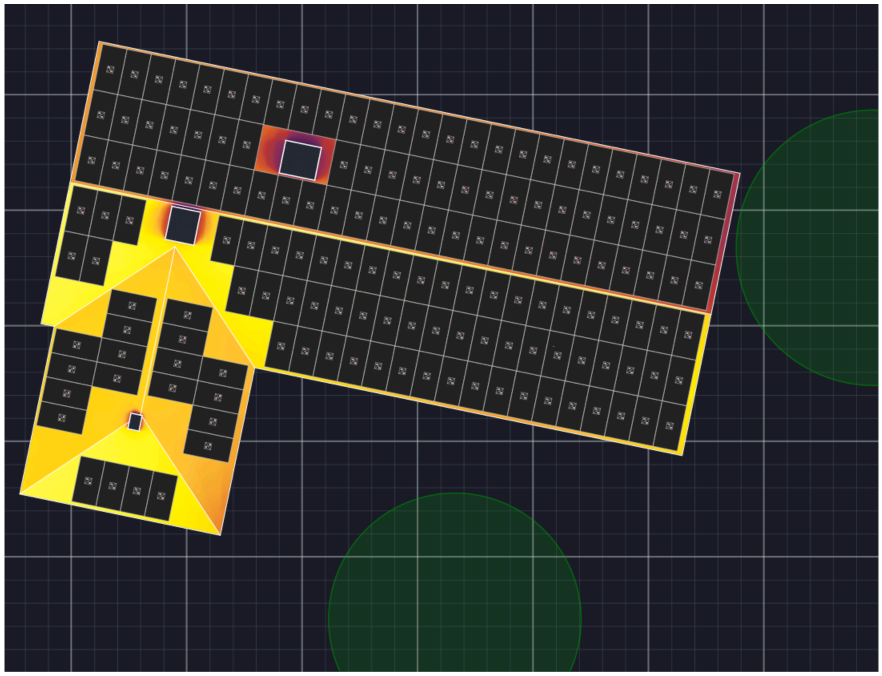


Figure 3: Each roof face is tiled to find feasible panel locations.

Potential strings. Similarly, to solve the design MILP we must first choose a set of potential strings $\mathcal{S}$. Each potential string $s \in \mathcal{S}$ is described by its length l_s. Just as with the panels and inverters, the complexity of the MILP increases with the number of strings $|\mathcal{S}|$, but the MILP is infeasible if too few are chosen. Using too many strings can present a substantial challenge to MILP solvers. Let us postpone the question of how to choose potential strings $\mathcal{S}$; we will soon see an alternative formulation of the design problem that does not require enumeration of $\mathcal{S}$.

4.2 Design MILP

We are now ready to describe the MILP formulation of the PV design problem.

Problem data. The problem data for the design MILP consists of the following:

- costs of panels c_p for $p \in \mathcal{P}$
- costs of inverters c_i for $i \in \mathcal{I}$
- annual energy of panels e_p for $p \in \mathcal{P}$
- hourly energy of panels e_{pt} for $p \in \mathcal{P}, t \in \mathcal{T}$ (usually $\mathcal{T}$ ranges over every hour in a year)
- desired system energy $\bar{E}^{\mathrm{des}}$

- constraints on inverter power, current, and voltage $\mathcal{C}_i$ for $i \in \mathcal{I}$
- maximum number of strings that can be wired to each inverter N_i for $i \in \mathcal{I}$
- string lengths l_s for $s \in \mathcal{S}$

We treat microinverter, string inverter, and DC optimizer configurations using the same framework; the difference between these configurations is simply encoded in the inverter constraints $\mathcal{C}_i$. For example, a microinverter configuration will have $N_i = 1$ for every $i \in \mathcal{I}$ and will constrain the voltage, power, and current wired to the inverter so that only one panel can be wired to a single inverter. We will assume in the exposition below that each inverter has only one MPPT. The case of multiple MPPTs adds only notational complexity.

Variables. We introduce integer variables to denote which components are used and how they are wired together:

- $z_p \in \{0, 1\}$ for $p \in \mathcal{P}$ is 1 if panel p is used
- $z_i \in \{0, 1\}$ for $i \in \mathcal{I}$ is 1 if inverter i is used
- $z_s \in \{0, 1\}$ for $s \in \mathcal{S}$ is 1 if string s is used
- $z_{sp} \in \{0, 1\}$ for $s \in \mathcal{S}$, $p \in \mathcal{P}$ is 1 if panel p is assigned to string s
- $z_{li} \in \{0, 1\}$ for $i \in \mathcal{I}$, $l \in \mathcal{L}_i$ is 1 if any string of length l is wired to inverter i
- $n_{li} \in \mathbb{Z}$ for $i \in \mathcal{I}$, $l \in \mathcal{L}_i$ denotes the number of strings of length l wired to inverter i
- $n_{lf} \in \mathbb{Z}$ for $f \in \mathcal{F}$, $l \in \mathcal{L}$ denotes the number of strings of length l on face f

We introduce continuous variables to denote the energy produced by the components:

- $e_{st} \in \mathbb{R}$ for $s \in \mathcal{S}$, $t \in \mathcal{T}$ denotes the energy produced by string s at time t

Capacity constraint. To ensure that these variables have the correct interpretations, we say the capacity constraint

$$(z_p, z_i, z_s, z_{li}, z_{sp}, n_{li}, n_{lf}) \in \mathcal{C}^{\text{cap}}$$

holds if

$$\begin{array}{lll}
\sum_{s\in\mathcal{S}} z_s = \sum_{f\in\mathcal{F}} n_{lf}, & l \in \mathcal{L} & \text{(every string is on some face)} \\
\sum_{s\in\mathcal{S}} z_s = \sum_{i\in\mathcal{I}} n_{li}, & l \in \mathcal{L} & \text{(every string is wired to some inverter)} \\
\sum_{p\in\mathcal{P}} z_p = \sum_{s\in\mathcal{S}} l_s z_s, & & \text{(every panel is on some string)} \\
\sum_{s\in\mathcal{S}} z_{sp} = z_p, & p \in \mathcal{P} & \text{(no panel is on more than one string)} \\
\sum_{p\in\mathcal{P}} z_{sp} = l_s z_s, & s \in \mathcal{S} & \text{(strings have correct lengths)} \\
\sum_{l\in\mathcal{L}_i} n_{li} \le N_i z_i, & i \in \mathcal{I} & \text{(strings fit on inverters)} \\
\sum_{l\in\mathcal{L}_i} z_{li} \le 1, & i \in \mathcal{I} & \text{(inverters have a single MPPT)} \\
n_{li} \le N_i z_{li}, & i \in \mathcal{I}, l \in \mathcal{L}_i & \text{(indicator true if number is positive).}
\end{array} \qquad (3)$$

Note that the capacity constraint is representable as an affine inequality constraint in the problem variables.

Inverter constraint. The maximum current, voltage, and power across an inverter are linear in the problem variables, with constants of proportionality that depend on the characteristics of the type of panel chosen.

- Current is proportional to $\sum_{l \in \mathcal{L}_i} n_{li}$, the number of strings wired to the inverter.
- Voltage is proportional to $\max_{l \in \mathcal{L}_i} l z_{li}$, the maximum length of any string.
- Power is proportional to $\sum_{l \in \mathcal{L}_i} l n_{li}$, the total number of panels wired to the inverter.

For ease of notation, we assume that the constants of proportionality have been incorporated into the constraint sets $\mathcal{C}_i$ from (1), so

$$\left(\sum_{l \in \mathcal{L}_i} n_{li},\ \max_{l \in \mathcal{L}_i} l z_{li},\ \sum_{l \in \mathcal{L}_i} l n_{li} \right) \in \mathcal{C}_i$$

if and only if the configuration is within the safe operating range for inverter i.

Energy. We use a simple approximation to the energy output of an array based on the irradiance and wiring to encourage similarly shaded panels to be strung together into strings. Letting $\mathcal{P}_s$ denote the set of panels in string s, and e_{pt} the energy of panel p and time t, the approximation to the energy e_{st} produced by string s at time t is

$$e_{st} = |\mathcal{P}_s| \min_{p \in \mathcal{P}_s} e_{pt}.$$

This approximation is an underestimate of the true energy. The MPPT will choose a current through the string which is feasible (less than the maximum current allowable for any panel in the string), and which causes the string to produce the greatest energy possible. The current which produces the maximal power in the panel with the least irradiance is always feasible for the other panels. It produces power $\min_{p \in \mathcal{P}_s} e_{pt}$ in the panel with the least irradiance, and produces at least that much power in every other panel in the string. Figure 2b demonstrates this approximation graphically.

This approximation has the advantage that it can be represented with linear constraints. If for some $M \in \mathbf{R}$,

$$e_{st} \leq e_{pt} l_s + M(1 - z_{sp}), \quad s \in \mathcal{S}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{4}$$

then e_{st} is a lower bound on the energy produced in string s at time t.

- If $z_{sp} = 1$, then $e_{st} \leq e_{pt} l_s$.
- If $z_{sp} = 0$, then $e_{st} \leq e_{pt} l_s + M$.

If M is large enough, then e_{st} is not constrained by e_{pt} for panels p not chosen to be in string s. Choosing

$$M \geq \left(\max_{l \in \mathcal{L}} l\right) \left(\max_{p \in \mathcal{P}, t \in \mathcal{T}} e_{pt}\right)$$

is sufficient.

This linear approximation is exact when, at each time t, every panel in a given string has exactly the same energy. In fact, this happens surprisingly often, since at any time of day, the energies of all the panels on a given roof face can be well approximated by one of two values: $e_{pt} \approx \alpha_t$ or $e_{pt} \approx \beta_t\}$ for every $p \in \mathcal{P}$. This simplification is due to the binary impact of shading: panels either produce a high energy (when in direct sunlight) or a low energy (using diffuse light from the blue sky). Using this linear approximation in the optimization problem induces a clustering of panels into strings that groups panels by the energy they produce, and hence, one that frequently segregates shaded panels from unshaded panels, making the approximation exact at the solution for many times t.

Design MILP. We're ready to put together all of the parts into the design MILP. Our problem is

$$\begin{array}{llll}
\text{minimize} & \sum_{i\in\mathcal{I}} c_i z_i + \sum_{p\in\mathcal{P}} c_p z_p & & \text{(cost)} \\
\text{subject to} & E^{\text{des}} \leq \sum_{s\in\mathcal{S}} \sum_{t\in\mathcal{T}} e_{st} & & \text{(energy)} \\
& e_{st} \leq e_{pt} l_s + M(1 - z_{sp}) & s \in \mathcal{S}, t \in \mathcal{T}, p \in \mathcal{P} & \text{(linear approximation (4))} \\
& e_{st} \leq M z_s & s \in \mathcal{S} & \text{(no energy from unused string)} \\
& (\sum_{l\in\mathcal{L}_i} z_{li}, \max_{l\in\mathcal{L}_i} l z_{li}, \sum_{l\in\mathcal{L}_i} l n_{li}) \in \mathcal{C}_i & i \in \mathcal{I} & \text{(inverters (1))} \\
& (z_p, z_i, z_s, z_{li}, z_{sp}, n_{li}, n_{lf}) \in \mathcal{C}^{\text{cap}} & & \text{(capacity (3))}
\end{array} \tag{5}$$

with variables $z_p \in \{0,1\}$, $z_i \in \{0,1\}$, $z_s \in \{0,1\}$, $z_{li} \in \{0,1\}$, $z_{sp} \in \{0,1\}$, $n_{li} \in \mathbb{Z}$, and $n_{lf} \in \mathbb{Z}$, for $p \in \mathcal{P}$, $i \in \mathcal{I}$, $f \in \mathcal{F}$, $s \in \mathcal{S}$, and $l \in \mathcal{L}$.

The solution to the design MILP gives the minimum cost set of panels z_p and inverters z_i together with an assignment of panels to strings z_{sp} and of strings to faces n_{lf} with the properties that

1. the panels can be wired safely to inverters; and
2. the linear approximation to the energy produced by the strings of panels exceeds the desired energy $\bar{E}^{\text{des}}$.

The design MILP is a mixed integer linear program that can be solved by any standard solver; the experiments in this paper all use the Gurobi solver (Optimization 2012).

4.3 Refinement

The solution of the design MILP gives us an assignment of panels to strings in the form of z_{sp}. Before the design is complete, we must find an ordering of panels on strings such that they can be installed quickly and cheaply. In general, panels that are located far apart cost more to connect because extra wire is needed, and additional time is needed for the installation. We would like to find an ordering of panels that minimizes the amount of wire necessary to connect them.

We solve this problem by reducing it to a (very small) traveling salesman problem. Define a graph in which each panel in a string is a node. Let d_{ij} be the distance between panel p_i and p_j, which might depend on the physical distance between the panels, or on the cost of wiring the two panels together. Define variable $c_{ij} \in \{0, 1\}$ for each pair of panels i and j so $c_{ij} = c_{ji} = 1$ if p_i is connected to p_j in the string and 0 otherwise. We search for a solution c that connects every $p \in s$ in a single tour in order to minimize the total distance of the tour,

$$(1/2) \sum_{i \in s} \sum_{j \in s} d_{ij} c_{ij}.$$

In residential solar designs, there are generally no more than a dozen or so panels in a string, so a reasonable solution to this traveling salesman problem can be found quickly with standard methods.

The solution c will connect the panels in a loop. There are many ways to transform this into a string with distinct start and endpoints. We might remove the edge with the longest distance d_{ij}, or the edge between two panels closest to the inverter to which string s is wired.

5 Improving performance

The design MILP (5) produces a design that satisfies the requirements of the PV design problem. A particularly nice property of the design MILP is that the energy produced by the design always exceeds the input desired energy $\bar{E}^{\text{des}}$. However, the size of the problem is too large to solve in a reasonable time for all but the most simple problems. In this section we describe a few approximations that reduce the time needed to produce a good solution to the PV design problem to less than a minute.

5.1 Assignment and wiring

One important change to the design MILP which reduces computation time is to split it into two separate optimization problems which we call the assignment MILP and the wiring MILP. The assignment MILP is independent of the stringing of the panels, while the wiring MILP performs the stringing. We describe both below. Definitions for the problem data and variables used in these problem descriptions were previously introduced above.

Capacity constraints. The capacity constraint for the assignment problem now takes a different form: it enforces constraints on the string lengths, but no longer enforces any constraints on the assignment of panels to strings (which we leave to the wiring MILP). To ensure that the remaining variables have the correct interpretations, we say that the capacity constraint for the assignment problem

$$(z_p, z_i, z_{li}, n_{li}, n_{lf}) \in \mathcal{C}^{\text{asg}}$$

holds if

$$
\begin{array}{lll}
\sum_{p\in\mathcal{P}_f} z_p = \sum_{l\in\mathcal{L}} l n_{lf}, & f\in\mathcal{F} & \text{(every panel is on some face)}\\
\sum_{l\in\mathcal{L}_i} n_{li} \le N_i z_i, & i\in\mathcal{I} & \text{(every string is wired to some inverter)}\\
\sum_{i\in\mathcal{I}} n_{li} = \sum_{f\in\mathcal{F}} n_{lf}, & l\in\mathcal{L} & \text{(every string is on some face).}
\end{array}
\qquad (6)
$$

Note that the capacity constraint for the assignment problem is representable as an affine inequality constraint in the problem variables.

Assignment MILP. The assignment MILP assigns panels to roof faces and to inverters, respecting the capacity constraints by solving the problem

$$
\begin{array}{llll}
\text{minimize} & \sum_{i\in\mathcal{I}} c_i z_i + \sum_{p\in\mathcal{P}} c_p z_p & & \text{(cost)}\\
\text{subject to} & \sum_{p\in\mathcal{P}} e_p z_p \ge E^{\text{des}} & & \text{(energy)}\\
& (\sum_{l\in\mathcal{L}_i} z_{li}, \max_{l\in\mathcal{L}_i} l z_{li}, \sum_{l\in\mathcal{L}_i} l n_{li}) \in \mathcal{C}_i & i\in\mathcal{I} & \text{(inverters (1))}\\
& (z_p, z_i, z_{li}, n_{li}, n_{lf}) \in \mathcal{C}^{\text{asg}} & & \text{(capacity (6))}
\end{array}
\qquad (7)
$$

with variables $z_p \in \{0,1\}$, $z_i \in \{0,1\}$, $z_{li} \in \{0,1\}$, $n_{li} \in \mathbb{Z}$, and $n_{lf} \in \mathbb{Z}$ for $p \in \mathcal{P}$, $i \in \mathcal{I}$, $f \in \mathcal{F}$, and $l \in \mathcal{L}$. The solution to this problem gives the minimum cost set of panels and inverters with the properties that

1. the panels can be assigned to faces and wired safely to inverters, and
2. the sum of the annual energies of the panels exceeds the desired energy E^{des}.

This problem has fewer variables than the design MILP 5, and can be solved much more efficiently.

Wiring MILP. The assignment MILP assigns string lengths to roof faces, but does not select which panels to wire together into strings. Its estimate of the energy produced by a design with k_f panels on each face $f \in \mathcal{F}$ is simply the sum of the top k_f annual energies of panels on face f. Since some energy is lost in the conversion from DC to AC power, this overestimates the energy produced, particularly for string inverter configurations: when a shaded panel is strung in series with an unshaded panel, the performance of the string suffers significantly. The wiring MILP uses instead the linear approximation to the energy of a string described above.

The solution to (7) determines the number of strings of each length on each face. Given these strings $s \in S$ each with length l_s, we solve the following problem for each face to find the best way to wire the panels together:

$$
\begin{array}{llll}
\text{maximize} & \sum_{s\in\mathcal{S}} \sum_{t\in\mathcal{T}} e_{st} & & \text{(energy)}\\
\text{subject to} & e_{st} \le e_{pt} l_s + M(1 - z_{sp}) & s\in\mathcal{S}, t\in\mathcal{T}, p\in\mathcal{P} & \text{(linear approximation (4))}\\
& \sum_{p\in\mathcal{P}} z_{sp} = l_s & s\in\mathcal{S} & \text{(string length)}\\
& \sum_{s\in\mathcal{S}} z_{sp} \le 1 & p\in\mathcal{P} & \text{(one string per panel)}
\end{array}
\qquad (8)
$$

with variables $z_{sp} \in \{0,1\}$ and e_{st}.

5.2 Binary representation

We can exploit other properties of our problem data to achieve a more refined solution. As we noted earlier, at any given time, panels on the same roof face either produce a high energy (when in direct sunlight) or a low energy (using diffuse light from the blue sky). Hence the energy of the panels on a particular roof face can be well approximated as

$$e_{pt} \approx \alpha_t(1 - w_{pt}) + \beta_t w_{pt},$$

where $w_{pt} \in \{0, 1\}$, and α_t (β_t) is the average energy of an unshaded (shaded) panel at time t. The binary constraints on the problem variables allows ILP solvers to restrict the search space significantly, leading to faster convergence.

5.3 Clustering times

The complexity of Problem 8 grows significantly with $|\mathcal{T}|$. To make the problem smaller, we can cluster times to find a subset of times that still captures the shading information. The clustering method described here builds on De Rubira and Toole (2015).

Recall that we defined $e_t \in \mathbf{R}^{N_p}$ to be the vector of panel energies at time t. Suppose that we have a partition $\mathcal{T}_1, \dots, \mathcal{T}_k$ of the times $\mathcal{T}$ so that e_t is the same for every $t \in \mathcal{T}_i$, for each $i = 1, \dots, k$. Pick a set of index times $t_1 \in \mathcal{T}_1, \dots, t_k \in \mathcal{T}_k$. Then Problem 8 reduces to

$$\begin{array}{llll} \text{maximize} & \sum_{s\in\mathcal{S}} \sum_{i=1}^k |\mathcal{T}_i| e_{st_i} & & \text{(energy)} \\ \text{subject to} & e_{st_i} \le e_{pt_i} l_s + M(1 - z_{sp}) & s \in \mathcal{S}, i = 1, \dots, k, p \in \mathcal{P} & \text{(linear approximation (4))} \\ & \sum_{p\in\mathcal{P}} z_{sp} = l_s & s \in \mathcal{S} & \text{(string length)} \\ & \sum_{s\in\mathcal{S}} z_{sp} \le 1 & p \in \mathcal{P} & \text{(one string per panel).} \end{array} \tag{9}$$

Problem 9 uses many fewer variables and constraints than Problem 8. Hence this clustered problem formulation can typically be solved in less than a second, several orders of magnitude faster than the original formulation.

For more general problems, we may still wish to cluster times so that e_t is approximately equal for every time in the cluster. To achieve this, first pick the number of time clusters k. This number will be chosen to ensure Problem 9 can be solved efficiently. We aim to find sets $\mathcal{T}_1, \dots, \mathcal{T}_k$ that partition $\mathcal{T}$, as well as representative times for each cluster $t_i \in \mathcal{T}_i$ for $i = 1, \dots, k$. To do this, we solve

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^k \sum_{t\in\mathcal{T}_i} \|e_t - e_{t_i}\|, \\ \text{subject to} & \{\mathcal{T}_i\} \text{ partition the set } \mathcal{T} \\ & t_i \in \mathcal{T}_i, \quad i = 1, \dots, k \end{array}$$

with variables $\mathcal{T}_i$ and t_i.

An alternative approach, which we employ in our numerical experiments, is to approximate the energy vectors e_t in a cluster by the cluster centroid b_i. We solve

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{k} \sum_{t \in \mathcal{T}_i} \|e_t - b_i\| \\ \text{subject to} & \{\mathcal{T}_i\}_{i=1}^{k} \text{ partition the set } \mathcal{T} \\ & b_i \in \{0,1\}^{|\mathcal{P}|}, \quad i = 1, \ldots, k. \end{array}$$

This problem is a standard k-means clustering problem. Note that the cluster centroid vector of energies b_i may never be (exactly) achieved at any time.

Shading patterns are periodic with only slight variations daily. Hence a good choice of the number of clusters k can greatly reduce the problem size without introducing significant errors.

If we use the binary representation introduced above, then instead of clustering energies $e_t \in \mathbf{R}^{|\mathcal{P}|}$, we can instead cluster the boolean vectors $w_t \in \{0,1\}^{|\mathcal{P}|}$. Given this clustering, we solve Problem 9 with the objective (energy) replaced by

$$\sum_{s \in \mathcal{S}} \sum_{i=1}^{k} |\mathcal{T}_i| e_{st_i}.$$

Eliminate islands. Often times the design that produces the most energy is too expensive to install, or aesthetically awkward. One common problem is that panels may be placed far away from any other panels. We call these *islanded panels*. These islanded panels increase the cost of the installation, since a separate *rack* (beams attached to the roof) must be installed for each islanded panel. They also reduce the visual symmetry of the PV array, and may be considered an eyesore.

To remove islanded modules from the design produced by the design MILP, we perform a local search around the given design by iteratively moving islanded panels to available locations with the most adjacent panels, so long as the move does not reduce the energy of the installation unacceptably. Let n_l be the number of adjacent panels to location l, and e_l be the annual energy of a panel at location l. The desirability of location l is given by the function

$$f(l) = n_l + \epsilon e_l,$$

where $\epsilon < 1/\max_l e_l$ trades off between our annoyance at including islanded modules and our annoyance at reducing the energy of the installation. We iteratively select the filled location l with the smallest value of $f(l)$, and move that panel to the unfilled location l' with the largest value of $f(l)$, until no unfilled location has a higher value of f than any filled location.

6 Numerical results

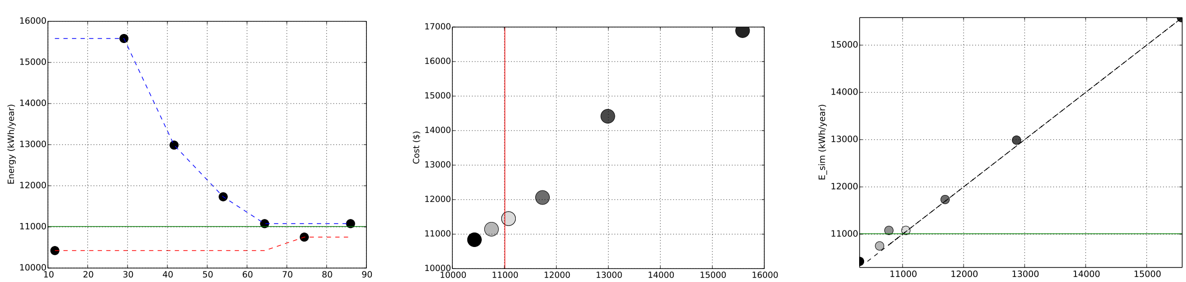
6.1 Convergence

In Figure 4 we show a typical convergence plot for the AutoDesigner, applied to the roof face shown in Figure 5e, for a desired energy output (red line) of 11KWh/year. Each iteration shown is an outer iteration of the bisection method obtained by solving the assignment MILP and the wiring MILP with clustered times. In each figure, the green line shows the desired energy E^{des}.

Panel 4a shows the (energy, time) pairs obtained as the algorithm converges to the desired energy E^{des}.

Panel 4b shows the (energy, cost) pairs obtained as the algorithm progresses as dots whose shading corresponds to algorithm progress: dark dots correspond to early iterations, and lighter dots appear at later iterations. We see that the algorithm first finds (energy, cost) points that bracket the desired energy; as the algorithm makes progress, it finds points closer and closer to the desired energy. The cost of the final (lightest) iterate is minimal among feasible points, *i.e.*, points that lie to the right of the red line.

Panel 4c shows the relation between the simulated energy E of the design and the parameter $\bar{E}^{\text{des}}$ used in the outer bisection algorithm. We plot points $(E, \bar{E}^{\text{des}})$ for each iteration as dots whose shading corresponds to algorithm progress as above. The dashed line shows the line $y = x$. We see that the algorithm first chooses $\bar{E}^{\text{des}}$ so that the simulated energy E brackets the desired energy E^{des}, and then bisects to find the lowest $\bar{E}^{\text{des}}$ producing a design with energy at least E^{des}. Deviations of iterates from the dashed line show that sometimes large changes in $\bar{E}^{\text{des}}$ cause only small changes in the simulated energy E.



(a) Energy converges to E^{des}. (b) Cost and energy converge. (c) Simulated energy converges.

Figure 4: Convergence of our algorithm for a typical roof.

6.2 Comparison to human-generated designs

The AutoDesigner was validated by NREL on residential test sites across the country. An expert designer produced designs for a range of sample sites, and the AutoDesigner produced designs for the same sites. A few of the resulting designs are shown in Figure 5.

We considered the following three performance metrics:

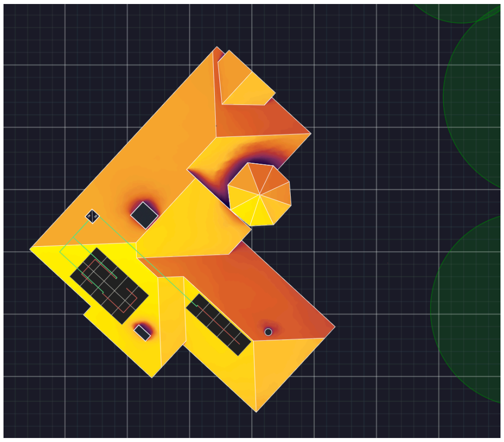
- **cost** is the cost of panels, inverters, and DC optimizers used in the design ($),
- **energy** is the simulated annual energy produced by a design (kWh), and
- **savings** is the simulated annual bill savings from the design. ($/year).

A design can sometimes save money even while producing less energy by producing energy at times of day when electricity is most expensive.

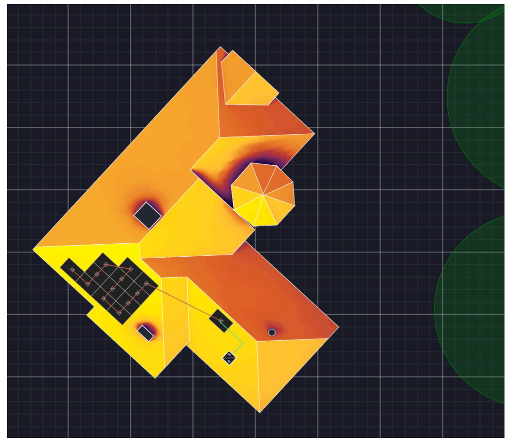
We say that one design is better than another if it has a higher *value* (energy to cost ratio), or if it has a higher *return on investment* (ROI) (savings to cost ratio). In more than 73% of our tests, the optimizer was found to produce designs that performed better than the designs produced by a PV expert with respect to either value or ROI.

To conduct this test, we selected 30 test sites located across the country. Staff at Aurora Solar created models for the roofs and trees at these sites, and provided a 60 minute training to a human designer, an NREL staff member with experience designing residential solar arrays. The designer was given an annual energy target for each site (80% of estimated consumption) and instructed to create the cheapest design possible. At the designer's disposal were 2 types of panels, 14 types of string inverters, 2 types of DC optimizers, and 2 types of micro inverters. The designer spent 9 hours creating designs for the 30 sites, approximately 18-20 minutes per site. Independently, we ran the AutoDesigner for each site with the same energy targets and components as inputs.

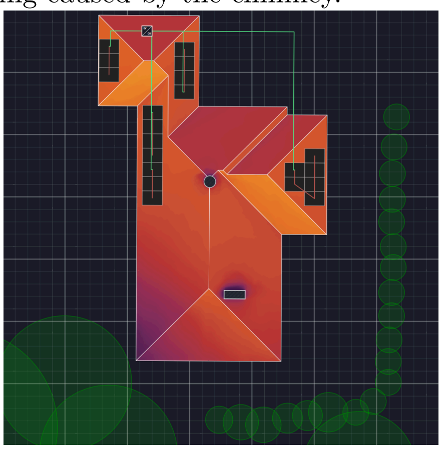
Designs created by human experts tend to fall within around 5% of the nominal energy production for the installation (Figure 6a). Our algorithm produces designs with energy production within 5% of the nominal values in less than two minutes for 92% of sites in our study, and in less than 3 minutes for every site (Figure 6b).



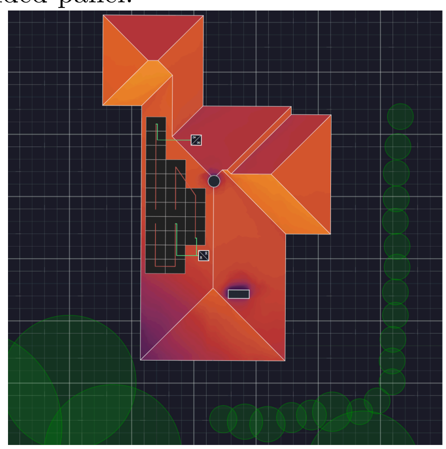
(a) Optimized design for sample site 1, in Millbrae, CA. The cost per kWh is 0.72. The strings in the left array reduce the impact of shading caused by the chimney.



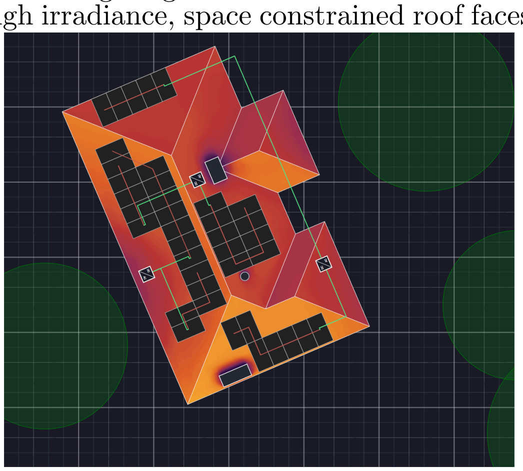
(b) User design for sample site 1, in Millbrae, CA. The cost per kWh is 0.89. This design uses expensive DC optimizers and an islanded panel.



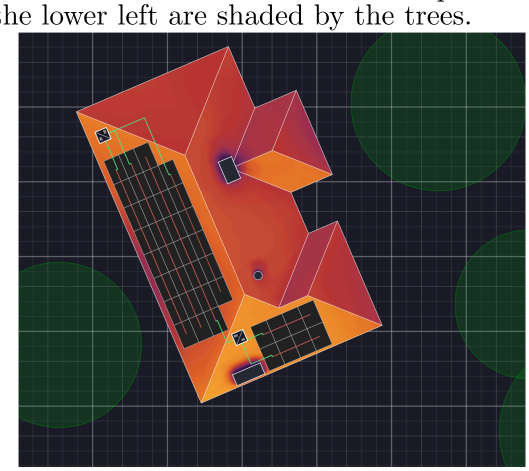
(c) Optimized design for sample site 4, in Portland, OR. The cost per kWh is 1.07. The string lengths are chosen to utilize the high irradiance, space constrained roof faces.



(d) User design for sample site 4, in Portland, OR. The cost per kWh is 1.17. One large roof face is used but some of the panels in the lower left are shaded by the trees.

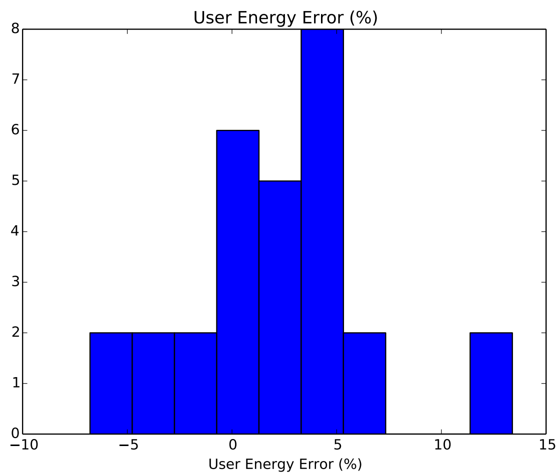


(e) Optimized design for sample site 5, in Staten Island, NY. The cost per kWh is 1.08. The panels and strings on the left roof face are chosen to minimize the impacts of shading from the tree.

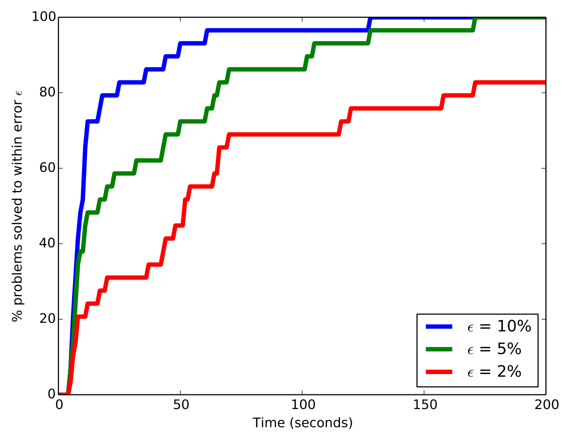


(f) User design for sample site 5, in Staten Island, NY. The cost per kWh is 1.12. Two large arrays are used but the stringing does not account for the shading caused by the trees or chimneys.

Figure 5: Optimized designs vs user designs.



(a) Relative error in energy production for experts' designs.



(b) CDF of relative error in energy production for MILP design for different error tolerances ϵ.

Figure 6: Relative errors in user and optimized designs.

7 Discussion

We have presented an algorithm for approximately minimizing the cost of an PV installation that produces a certain desired energy, and we have demonstrated that the algorithm produces excellent designs in practice. This algorithm produces a design that is optimal up to the modeling approximations we have chosen. To produce a better design, with respect to these metrics, would require either

- a more accurate model of cost of installation,
- a more accurate model of energy produced, or
- modeling a more rich set of potential hardware components.

For example, instead of forming a grid of potential panel locations at the start of the algorithm, one might want to allow a continuum of panel locations. Of course, allowing more panel locations increases the complexity of the algorithm.

Our approach to modeling hardware components was to enumerate each possible component. A more powerful model might parametrize these components and choose from the parameters; however, this addition is not particularly useful for most solar installers, who must choose from pre-existing hardware components. Other hardware components give rise to substantively different problems: for example, allowing a battery adds a control problem (how to best operate the battery) to the design problem (how to size a battery to achieve the best ROI).

Another natural and important extension of our work is to investigate robust formulations of the problem to ensure the array continues to perform well in a variety of weather conditions, despite measurement errors in problem parameters, and despite changing conditions: for example, one might want a design that continues to perform well even as trees around the roof grow taller and cast more shade. More consequentially, one might want to choose a design to be robust to plausible regulatory changes that affect the price and incentive structure for residential solar energy.

8 Conclusion

In this paper, we proposed an automatic solution for the design of rooftop solar photovoltaic arrays. We showed that our algorithm, the AutoDesigner, was capable of producing designs of quality comparable or better than those produced by a human expert in a small fraction of the time: our experiments showed that the AutoDesigner outperforms hand-tuned designs for more than 70% of test cases.

Technically, the approach of the AutoDesigner is to approximate the PV design problem by one that can be globally solved by modern mixed integer linear programming (MILP) solvers, and then to refine the approximation until the method converges to a satisfactory solution. The AutoDesigner algorithm consists of three phases:

1. a MILP formulation to find the optimal panel locations and array wiring,
2. a bisection refinement routine to tune the linear model, and
3. a local search routine to improve the results of the linear approximation.

We believe there is an opportunity for more sophisticated optimization techniques to further improve the solution accuracy and solution speed, as well as to target a more detailed model of the problem that includes the cost of wiring and installation. From a practical perspective, it is also important to find designs which are aesthetically appealing, as homeowners are loathe to install ugly arrays.

From a broader perspective, this work presents a detailed case study of the application of a variety of optimization techniques, including outer approximation of a PDE-constrained objective by a linear objective, mixed integer linear programming, bisection search for black-box optimization, and model reduction via clustering, to a complex and realistic problem in engineering design. More generally, this work showcases the efficacy of OR techniques to solve an important and complex problem in engineering design. We hope that this case study can serve as a guide for other important applications in engineering design.

This work also has important implications for the development of the solar industry. Automated design of rooftop solar has the potential to lower the "soft costs" of installation, which comprise more than 64% of total installed cost. As such, we believe this problem deserves attention by the research community, and this project takes an important first step in that direction. This project has already had significant impact in practice. We hope this project inspires further contributions from the OR community to secure our energy future through efficient distributed solar energy.

Acknowledgements

This work was developed with support from Aurora Solar and the DOE Sunshot Initiative. Aurora Solar also provided access to their shading simulation, which computes the objective function for the AutoDesigner.

References

Aravkin A, Burke J, Drusvyatskiy D, Friedlander M, Roy S (2016) Level-set methods for convex optimization. *arXiv preprint arXiv:1602.01506* .

Bertsimas D, Tsitsiklis J (1997) *Introduction to Linear Optimization*, volume 6 (Athena Scientific Belmont, MA).

Boyd S, Vandenberghe L (2004) *Convex Optimization* (Cambridge University Press).

De Rubira TT, Toole O (2015) Clustering techniques for photovoltaic system simulations under nonuniform irradiance conditions. *2015 IEEE Photovoltaic Specialist Conference (PVSC)*, 1–5 (IEEE).

De Soto W, Klein S, Beckman W (2006) Improvement and validation of a model for photovoltaic array performance. *Solar Energy* 78–88.

Esram T, Chapman PL (2007) Comparison of photovoltaic array maximum power point tracking techniques. *IEEE Transactions on energy conversion* 22(2):439–449.

Freeman J, Simon J (2015) Analysis of aurora's performance simulation engine for three systems URL `http://www.nrel.gov/docs/fy15osti/64213.pdf`.

Friedman B, Ardani K, Feldman D, Citron R, Margolis R, Zuboy J (2013) Benchmarking non-hardware balance-of-system (soft) costs for U.S. photovoltaic systems using a bottom-up approach and installer survey — second edition URL `http://www.nrel.gov/docs/fy14osti/60412.pdf`.

Gagnon P, Margolis R, Melius J, Phillips C, Elmore R (2016) Rooftop solar photovoltaic technical potential in the united states: A detailed assessment URL `http://www.nrel.gov/docs/fy16osti/65298.pdf`.

Grana P, Shiao M (2014) The microinverter and DC optimizer landscape 2014: The evolution from market niche to market success URL `http://www.greentechmedia.com/research/report/the-microinverter-and-dc-optimizer-landscape-2014`.

Massalha Y, Appelbaum J (2016) The effect of interconnecting solar cells in the pv module on the power output. *Science of Electrical Engineering (ICSEE), IEEE International Conference on the*, 1–4 (IEEE).

US Energy Information Administration (2016) Electric power monthly with data for March 2016. URL `http://www.eia.gov/electricity/monthly/pdf/epm.pdf`.

Nguyen D, Lehman B (2008) An adaptive solar photovoltaic array using model-based reconfiguration algorithm. *IEEE Transactions on Industrial Electronics* 55(7):2644–2654.

Optimization G (2012) Gurobi optimizer reference manual URL `http://www.gurobi.com`.

Parida B, Iniyan S, Goic R (2011) A review of solar photovoltaic technologies. *Renewable and sustainable energy reviews* 15(3):1625–1636.

Patnaik B, Sharma P, Trimurthulu E, Duttagupta S, Agarwal V (2011) Reconfiguration strategy for optimization of solar photovoltaic array under non-uniform illumination conditions. *Photovoltaic Specialists Conference (PVSC), 2011 37th IEEE*, 001859–001864 (IEEE).

Rekioua D, Matagne E (2012) *Optimization of photovoltaic power systems: modelization, simulation and control* (Springer Science & Business Media).