



Solar Field Layout and Aimpoint Strategy Optimization

Alexander Zolan,¹ William Hamilton,¹ Michael Wagner,¹
and Kashif Liaqat²

1 National Renewable Energy Laboratory

2 Florida State University

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report
NREL/TP-5700-80596
August 2021



Solar Field Layout and Aimpoint Strategy Optimization

Alexander Zolan,¹ William Hamilton,¹ Michael Wagner,¹
and Kashif Liaqat²

1 National Renewable Energy Laboratory

2 Florida State University

Suggested Citation

Zolan, Alexander, William Hamilton, Michael Wagner, and Kashif Liaqat. 2021. *Solar Field Layout and Aimpoint Strategy Optimization*. Golden, CO: National Renewable Energy Laboratory. NREL/TP-5700-80596. <https://www.nrel.gov/docs/fy21osti/80596.pdf>.

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report
NREL/TP-5700-80596
August 2021

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Solar Energy Technologies Office. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via www.OSTI.gov.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

Contents

Cover Page	i
1 Major Goals and Objectives	1
2 Project Results and Discussion	3
2.1 Thermal Flux Map Characterization	4
2.1.1 Approximation 1: Discretize the Receiver Surface	4
2.1.2 Approximation 2: Translate Flux Maps Without Recalculation	4
2.1.3 Case Studies	6
2.2 Aimpoint Optimization Model	9
2.2.1 Model Formulation	10
2.2.2 Examples Using Gaussian Flux Images	12
2.3 Solution Methods	13
2.3.1 Spatial Decomposition	14
2.3.2 Heuristic Grouping Method	16
2.3.3 Case Study	17
2.4 Layout Improvement Approach	20
2.4.1 Heuristic Improvement Algorithm	21
2.4.2 Case Study	22
2.5 Software Development	22
2.5.1 Optimization Software Package	23
2.5.2 Flux Calculations Within HALOS	24
2.5.3 SolarPILOT Python Interface	24
3 Conclusions and Future Work	26
4 References Cited	28

Abstract

The existing methods that determine heliostat aiming strategies for concentrating solar power (CSP) central receiver plants typically use heuristics and/or are computationally expensive, and they lack flexibility for different desired flux profiles and receiver geometries. Because of the interaction between layout and aimpoint strategy, considering the former without accounting for the latter may yield solutions with superfluous heliostats that cannot be used efficiently without compromising receiver flux constraints. To that end, we develop a software decision tool that uses innovative optimization methods to both optimize aimpoint strategies and improve candidate layouts for the solar collection field of a CSP central receiver plant. A CSP plant's effectiveness relies on the optical efficiency of the solar field, which may be limited by losses due to (i) the cosine effect, (ii) atmospheric attenuation, (iii) interference (i.e., shading and blocking) between heliostats, (iv) spillage as a result of heliostat positioning and geometry, and (v) some heliostats' inability to direct irradiance to the receiver without damage due to excessive thermal flux. The goal of this work is to obtain optimized aiming strategies and improved solar field layouts that reduce capital cost and increase field optical efficiency and utilization, while meeting the power requirements of a given CSP receiver design. We formulate the aimpoint optimization problem as a mixed-integer linear programming model, which we then decompose into submodels that we solve in parallel. The decomposition subdivides the solar field into sections, and aimpoint strategies for each section are obtained independently of the others. To improve existing layouts, we develop a utilization-weighted efficiency metric that we use to relocate heliostats to sections of the solar field with similar efficiency and higher utilization. Finally, to connect our software to high-fidelity flux models, we develop a Python application programming interface for SolarPILOT, a mature software package that characterizes solar field performance and generates the heliostat layouts and flux maps that serve as input to our models.

1 Major Goals and Objectives

This project develops a software decision tool that uses innovative optimization methods to determine the best layout for the solar collection field of a concentrating solar power (CSP) central receiver plant. A CSP plant's effectiveness relies on the optical efficiency of the solar field, which may be limited by losses due to (i) the cosine effect, (ii) atmospheric attenuation, (iii) interference (i.e., shading and blocking) between heliostats, (iv) spillage as a result of heliostat positioning and geometry, and (v) some heliostats' inability to direct irradiance to the receiver without damage due to excessive thermal flux. Design and operations decisions addressed by this project include: (i) the location of each heliostat in the solar field, and (ii) the intended aimpoint of each heliostat to the receiver for each hour, across a representative collection of days. The goal of this work is to reduce the capital cost associated with heliostat production to meet the power requirements of a given CSP receiver design and power block, while increasing the productivity of the system.

Given weather data, heliostat specifications, a topological layout of the proposed design field, and a receiver's size, location, and geometry as input, we formulate a layout optimization methodology that places heliostats in the solar field so as to meet the energy requirements of the power block and storage system at minimum capital cost, while ensuring that the thermal flux profile of the receiver falls within design specifications. The

model limits the inclusion of superfluous heliostats that may not be usable during peak DNI periods without compromising mechanical limits on the receiver due to thermal flux. The software platform builds on existing work by the proposing team and others to make the development tractable.

The success of this project will allow a plant designer to obtain a well-planned solar field layout that accounts for the aimpoint strategy to the receiver, which will:

1. Reduce the capital requirements for the heliostat field by reducing the required land and number of heliostats to send the target amount of irradiance to the receiver;
2. Increase the optical efficiency of the system by reducing the number of inactive heliostats due to thermal flux constraints on the receiver.

The focus of our work is on the development and implementation of optimization models and heuristics, so we rely on SolarPILOT [27] to provide the high-fidelity flux maps that serve as inputs to the tool we develop to obtain aimpoint strategies, as well as starting layouts for our layout improvement algorithm. To facilitate seamless passing of information into our Python-developed tool, in this project we develop a Python application programming interface (API) for SolarPILOT as well [14], which we describe in Section 2.5.3. We then interface with this API to obtain layouts and flux maps for a given solar field.

Our approach to generating an aiming strategy differs from SolarPILOT's in two ways. First, our method seeks a collection of aimpoints that maximizes thermal power delivery to the receiver surface to within a user-specified tolerance of optimality, whereas SolarPILOT uses a heuristic that spreads flux evenly on the surface. Second, our method allows us to restrict flux to within specific limits to prevent damage to the receiver, whereas SolarPILOT's aimpoint strategy imposes no such limits. As a result, we believe our aimpoint strategies are more representative of in situ operations.

Our layout improvement algorithm starts with a given layout, such as the one provided by SolarPILOT that seeks maximum optical efficiency given the receiver and land characteristics. Then, we use the aimpoint optimization model to obtain strategies and heliostat utilization by section. Finally, we relocate heliostats from less-utilized sections to sections with the highest utilization; in other words, to locations in which the relative loss of optical efficiency is lower than the relative increase in utilization.

Project Impact

To our knowledge, the state of the art in using integer programming techniques to obtain aimpoint strategies is the recently published papers by Ashley et al. [2] and Kuhnke et al. [20]. In both cases, the authors use the PS10 plant, which contains 624 heliostats in the solar field, to obtain aimpoint strategies that minimize spillage losses without violating flux constraints on the receiver, an objective analogous to the one in our aimpoint strategy optimization model. This results in this project exhibit solutions for fields five times as large, and we have solved even larger instances, such as the field at Crescent Dunes as a case study we are currently pursuing with Tietronix. We have obtained a similar level of performance gap to that of Kuhnke et al., though they utilize a much coarser receiver surface (i.e., 20 measurement points versus the 400 we incorporate). Our intent is to obtain optimized aimpoints for commercial-scale plants, and we believe that our decomposition technique is well-suited for this application; this allows us to obtain solutions to much larger solar fields on a similar time scale, advancing the state of the art using this solution

technique. The layout improvement method we developed is heuristic in nature, and so we do not present any bounds on further improvement. However, the metric we present for solar field productivity is novel, as it accounts for both optical efficiency and utilization in its criteria for heliostat location selection. Moreover, the productivity improvement we show compared to an existing layout in SolarPILOT is similar to the improvements in optical efficiency presented by Noone et al. [23] when they compared their biomimetic layout to the radial stagger layout. The aimpoint strategy and layout tools can be extended to support alternative geometries for Gen3 receivers currently studied in other projects at the National Renewable Energy Laboratory (NREL), and the aimpoint strategy in particular can be further enhanced to respond to the detection of hotspots on the receiver.

2 Project Results and Discussion

This section presents the full methodology we employed in this study, as well as results obtained for a collection of case studies designed to either (i) measure the accuracy of our tools, or (ii) demonstrate improvement over other methods in the literature. The results begin with a discussion of the inputs we use for our model and the approximations we adopt to ensure that our model instances are both accurate and tractable. Next, we detail the optimization model we implement to obtain thermal-power-maximizing aimpoint strategies, and the optimization methods we implement and develop to obtain solutions to model instances quickly using open-source solvers. We then propose a metric not discussed in the literature to our knowledge, and utilize it as a means to improve layouts for solar fields. Finally, we discuss the software library we developed featuring the optimization model, as well as the development of a Python-based application programming interface for SolarPILOT that improved both the quality of our case study inputs and the accessibility of SolarPILOT for researchers and practitioners.

Literature Review

A recent review by Barbarena et al. [4] shows that the most commonly used approaches to obtain solar field layouts and characterize their performance as part of a full CSP system utilize heuristic methods. TieSOL [1, 17] is a software package that uses particle swarm optimization to obtain layouts and aimpoint strategies [18]. Researchers at the University of Houston developed RCELL [22], which considers staggered radial and north-south oriented layouts. Additional efforts use evolutionary algorithms to obtain heliostat layouts and characterize plant performance via simulation, such as DELSOL3 [19], HELIOS [26], and Campo [7, 8], while others develop heliostat layouts using their own algorithms and then characterize performance using separate software (see, e.g., [3, 11, 21, 30]). These algorithms assume either a radial or spiral orientation, which limits the span of potential designs; this reduces the time required to find solutions, but leaves room for efficiency improvements. Pattern-free optimization methods, like those shown in [6], have been shown to yield improved optical efficiency when compared to fixed geometries. Although all the methods above account for cosine, attenuation, interference, and spillage-related losses, they generally assume that all heliostats are aimed at the center of the receiver during the design process, which in practice would cause excessive thermal flux [2].

Similar to layout optimization, some existing efforts to find an aimpoint strategy that maximizes the delivery of thermal energy to the receiver while adhering to limits on thermal

flux of the receiver rely on evolutionary algorithms, as described in [13]. Belhomme et al. present a metaheuristic based on ant colony optimization to obtain a heliostat aiming strategy for a fixed solar field layout [5]. Cruz et al. propose a two-tiered method that uses a genetic algorithm to select a subset of the heliostats to activate, and then uses a gradient descent method to obtain specific aimpoints [9]; the authors propose a separate, teaching-learning-based optimization algorithm that lends itself well to parallel implementation [10]. To our knowledge, the only aimpoint optimization methodology that characterizes the model as a mixed-integer linear program was developed by Ashley et al. [2] and, more recently, Kuhnke et al. [20]. To achieve this formulation, the authors pre-specify a discrete grid of candidate aimpoints, then allocate heliostats to the grid with a goal of minimizing the flux gradient across the surface of the receiver. All of the works above assume that the layout is fixed; co-optimization of aimpoint strategy and solar field layout is an open question that we begin to address in this project.

2.1 Thermal Flux Map Characterization

Our goal is to obtain a layout and aimpoint strategy that maximizes the thermal power delivered to the receiver, given terrain characteristics, the receiver design, and a pre-specified number of heliostats to position in the solar field as input. Because high-fidelity flux maps can be computationally expensive to generate, even when provided a (fixed) collection of heliostat and aimpoint locations as input, we have developed two approximations that will maintain a high degree of accuracy while allowing instances of our model to be generated within a few minutes.

2.1.1 Approximation 1: Discretize the Receiver Surface

Our first approximation discretizes the receiver surface into a collection of *measurement points* and *aimpoints*, the former of which are used to estimate total thermal power delivery to the receiver and ensure that flux limits are met, and the latter of which may be targeted by a heliostat. This addresses the non-convex nature of flux incident to a measurement point as a function of the aimpoint location, and allows for us to solve an instance of the aimpoint strategy model to proven optimality by establishing a valid upper bound on thermal power delivery. We note that for some receiver designs, not all aimpoints will be accessible to every heliostat; e.g., a heliostat may only access a column of aimpoints on a cylindrical receiver. Figure 1 displays an example receiver surface discretization.

2.1.2 Approximation 2: Translate Flux Maps Without Recalculation

The second approximation translates a thermal flux image in the place of recalculation for any non-central aimpoint. Specifically, given a heliostat location and sun location as input, we calculate the flux incident to each measurement point on the receiver, which we call a *flux map*, when a central aimpoint is selected. Then, for every other aimpoint, we use that image and shift the flux map by the distance between the central and new aimpoints to obtain an estimated flux map. Let H be the number of heliostat locations, and let A be the number of aimpoints. Using our approximate method, we only generate H flux maps as input to the optimization model, instead of generating $H \cdot A$ flux maps when we recalculate flux for every aimpoint.

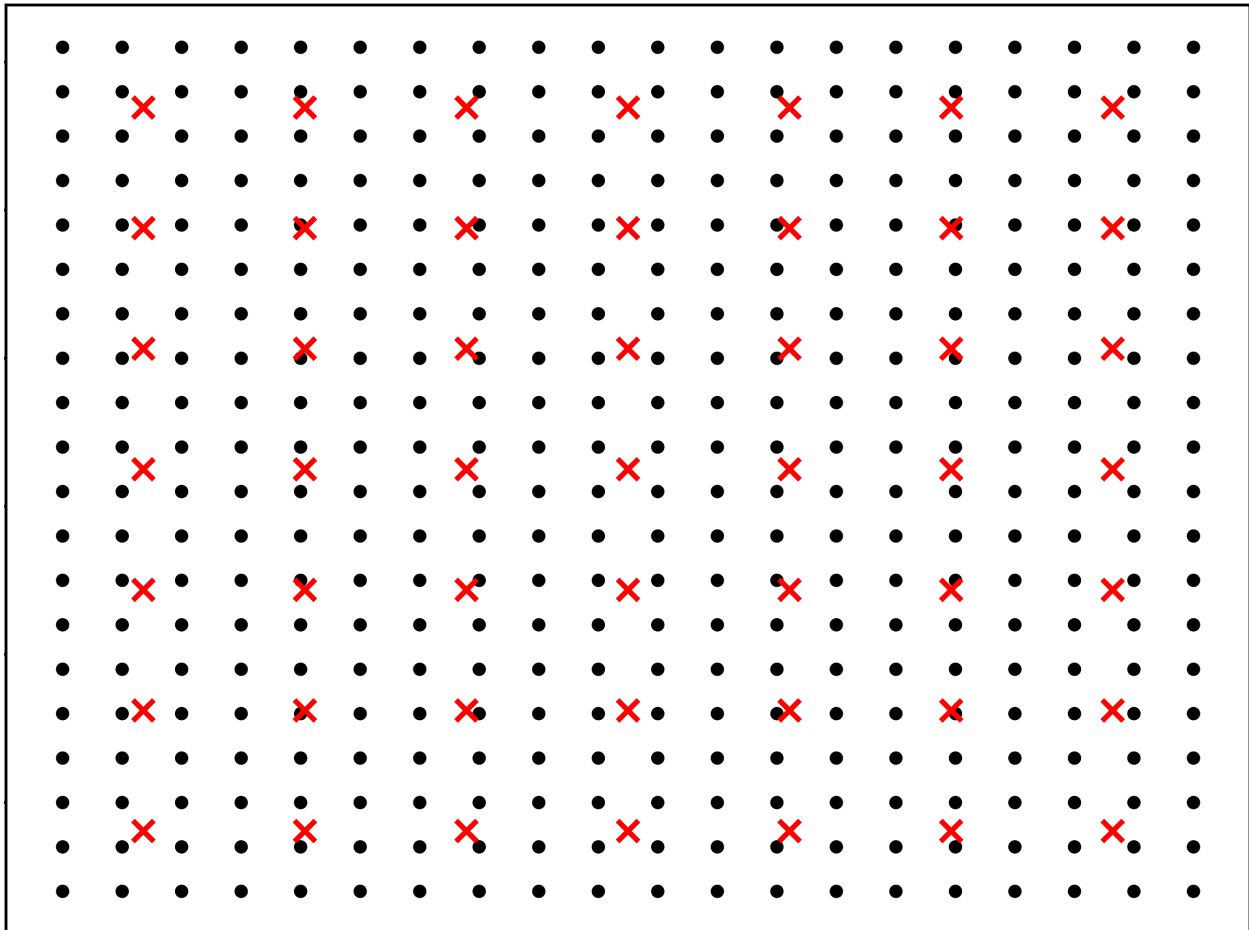


Figure 1: An example discretization of a square, flat-plate receiver surface into a collection of 400 measurement points and 49 aimpoints, represented by black dots and red x's, respectively

2.1.3 Case Studies

We assess the quality of our estimation method by comparing the calculated flux incident to each measurement point by generating H flux maps and translating images in our method described above to that incident to the same measurement points by generating $H \cdot A$ flux maps as the higher-fidelity (but more computationally expensive) standard. Let M be the number of measurement points on the receiver that we will evaluate for the case study. Then, we calculate mean absolute percent error (MAPE), our measure of interest, in the following way:

1. For $m = 1, \dots, M$, let o_m be the thermal flux incident to measurement point m when using the approximate method that generates H flux maps, or one for each heliostat, aimed at the center of the receiver surface.
2. For $m = 1, \dots, M$, let e_m be the thermal flux incident to measurement point m when using the higher-fidelity method that generates $H \cdot A$ flux maps, or one for each heliostat-aimpoint pairing.
3. Then, for $m = 1, \dots, M$, the ratio $\delta_m = |o_m - e_m|/e_m$ is the absolute percent error for measurement point m , and $\epsilon = \sum_{m=1}^M \delta_m / M$ is the MAPE for the case study.

Separately, we calculate the difference in thermal power delivered via the approximation to that of the higher-fidelity method.

Location-Variant Cases

We develop a collection of four case studies, and our criteria for an accurate approximation is a MAPE of less than 10% when compared to the baseline of a high-resolution map of the receiver without image shifting. Each case study utilizes 2,500 measurement points (i.e., 50 rows of 50 points each) and 25 candidate aimpoints. Our cases studies take place in two different locations (Daggett, California, and Ouarzazate, Morocco), each of which is paired with a 50-MW receiver and a 250-MW receiver. Similar to the previous test, we implement Gaussian flux images.

The solar field used in each case study was generated by SolarPILOT [27] after the receiver characteristics and location were updated from default settings. Figure 2 displays the solar field generated for one of the case studies, specifically that of the 50-MW, flat-plate receiver in Daggett, CA. Each heliostat was randomly assigned to one of the 25 candidate aimpoints, which were distributed equally across the flat-plate receivers.

Table 1 displays the results of the studies, and shows that all four case studies presented exhibit a MAPE well within 10% and a total field error of less than 1%. After repeating the random assignment of heliostats to aimpoints 30 different times we found that the confidence interval on MAPE was significantly lower than 10%, as was the maximum MAPE across all cases and randomly generated instances. Figure 3 displays a map of the relative error by measurement point for each case. We observe that error is low in general and mostly confined to the edges in the corners for the 250-MW cases, whereas error is highest near the bottom edge for the 50-MW cases. We believe the likely source of error reduction for the larger plants is the presence of mirrors casting larger flux images on the receiver from a longer distance, which are subject to a smaller angular difference (and hence, a reduced alteration of the image) when moving to an off-center aimpoint.

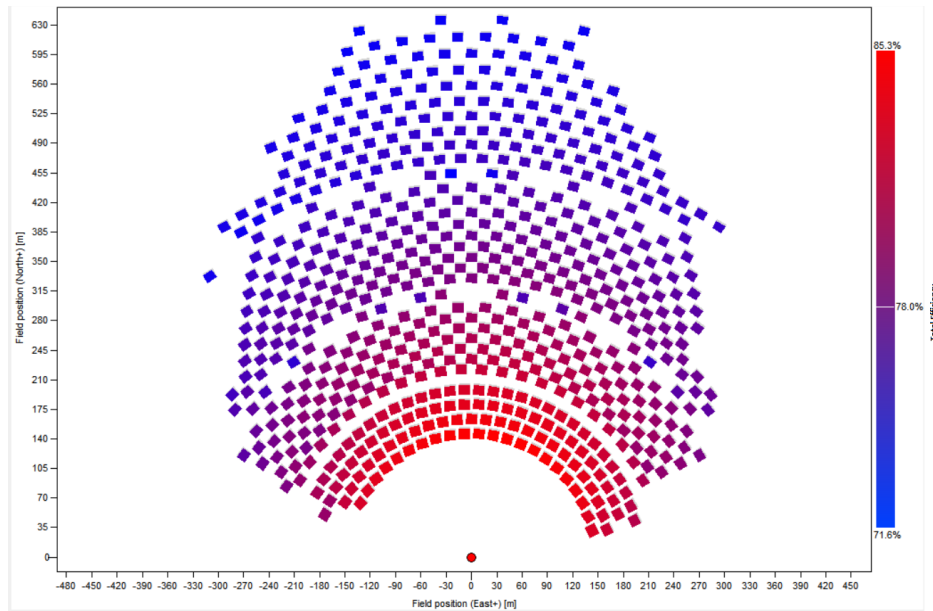


Figure 2: Solar field layout used in the case study using a 50-MW, flat-plate receiver in Daggett, CA. The dot at the bottom-center of the image represents the receiver, and the rectangles represent heliostats. The legend on the right maps the color of each heliostat to its optical efficiency.

Case	Location	Rating (MW)	# Heliostats	MAPE (%)	Field Error (%)
1	Daggett	50	656	2.66	0.60
2	Daggett	250	3,322	0.35	0.06
3	Ouarzazate	50	650	3.14	0.74
4	Ouarzazate	250	3,250	0.11	0.03

Table 1: Mean Absolute Percent Error (MAPE) Summary for Case Studies

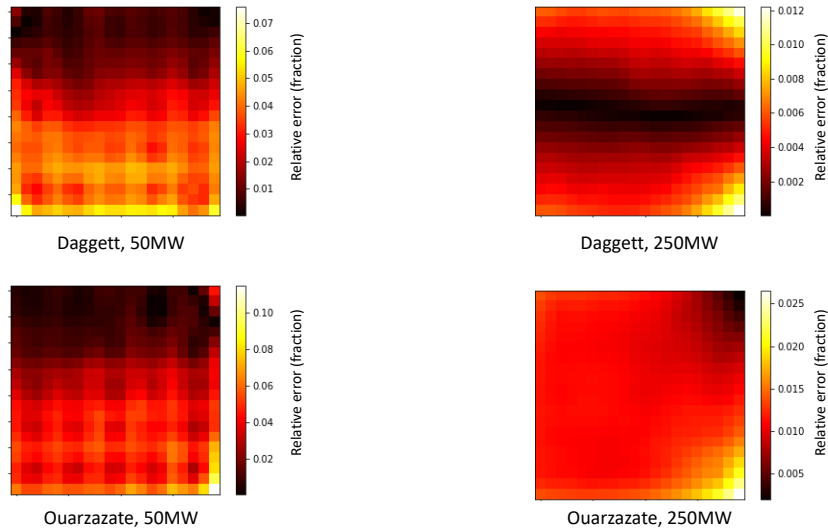


Figure 3: Heat maps displaying the relative difference in flux incident to each measurement point for each of the rectangular flat-plate case studies described above. Labels below each heat map denote the case study, and the legend on the right maps color to absolute fraction of error at each measurement point. The x-axis position and y-axis positions on each map are analogous to the horizontal and vertical locations on the receiver, respectively.

Annual Assessment

We extended our assessment of the image translation approximation for a full year instead of using a single time period. Specifically, we have extended cases 1 and 2 for a large collection of hours to demonstrate that the accuracy of our approximation is robust with respect to changes in the time of day and year.

For our year-long analysis, we only consider hours in which the DNI is greater than or equal to 500 (W/m²). The weather data used in these case studies have around 2,700 such hours, with wide day-to-day and seasonal variation. To keep the computation time of this analysis manageable, we select every 15th hour that meets the minimum DNI threshold for one year. Figure 4 displays the hours selected over the course of the year, demonstrating that our analysis includes a wide representation of both DNI and solar angles.

The results obtained from the case studies are presented using empirical cumulative distribution function (ECDF) in Figure 5. The graphs show that the maximum value for MAPE is found to be 2.5% and 0.3% for 50 MW and 250 MW, respectively, both of which fall below the goal threshold of 10%. The trend of a larger field having lower MAPE than the smaller field is consistent with the findings from the last quarter, and confirms that our approximation is scalable.

Figure 6 displays the MAPE by the hour of the day and month of the year. The results show that the error is highest in the morning and evening, and is lowest at noon, when the DNI is highest under a clear-sky model. The diurnal nature of approximation error is due to the increased image distortion when the heliostats are oriented to capture sunlight at

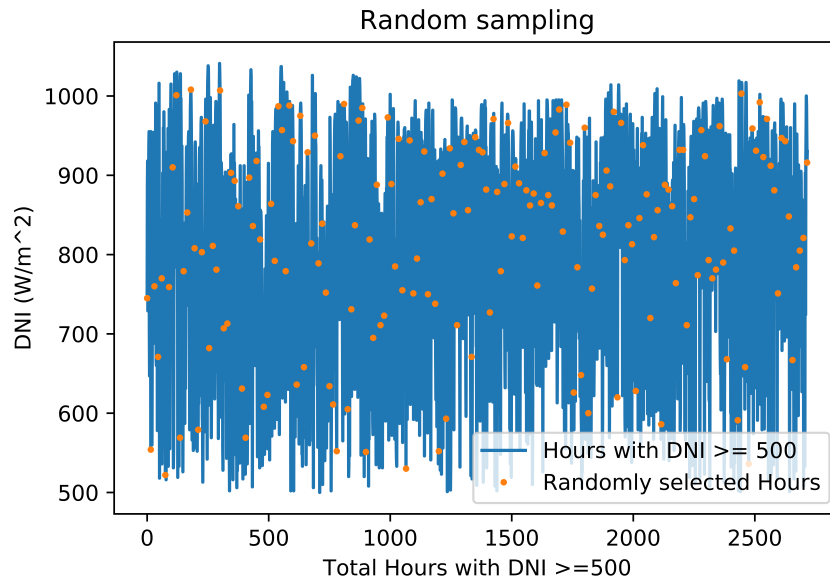


Figure 4: A time series of DNI by hour for all hours of the year in which DNI is at least 500 W/m^2 . Orange dots represent the hours selected for the yearly analysis, demonstrating that our analysis includes a wide representation of both DNI and solar angles.

sunrise and sunset. Shifting the aimpoint of the image in the early and late hours leads to further distortion of the flux image, which is not captured by our image-shifting approach. As a result, the approximation error is greater in the early and late hours when compared to noon when using the approach we adopt. This diurnal trend is consistent for each month of the year. Additionally, similar to the results presented in Figure 3, the highest error occurred in the corners of the receiver, where aimpoints are sufficiently far from the measurement point that image distortion would have a greater impact than at the center of the receiver surface.

2.2 Aimpoint Optimization Model

Our aimpoint strategy optimization model assumes that heliostat locations are fixed, and we consider a single time period in which the aimpoint strategy is optimized. The optimized aimpoint strategies then serve as input to the layout optimization method that we discuss later in Section 2.4. Flux maps serve as inputs to the model, using the discretization and image-shifting approximations that we developed in Section 2.1. The developments in Sections 2.2–2.3.1 are described further in [31].

In what follows, we describe the aimpoint strategy optimization model in general form. We assume that we can choose any flux calculation method to pre-compute the flux delivered to each measurement point given a heliostat's selected aimpoint, which serves as a parameter in our model. Our notational convention is adapted from the best practices established in [24] and uses lowercase letters to denote decision variables, capital letters to represent parameters, subscripts to denote indices, and caligraphic letters to denote sets.

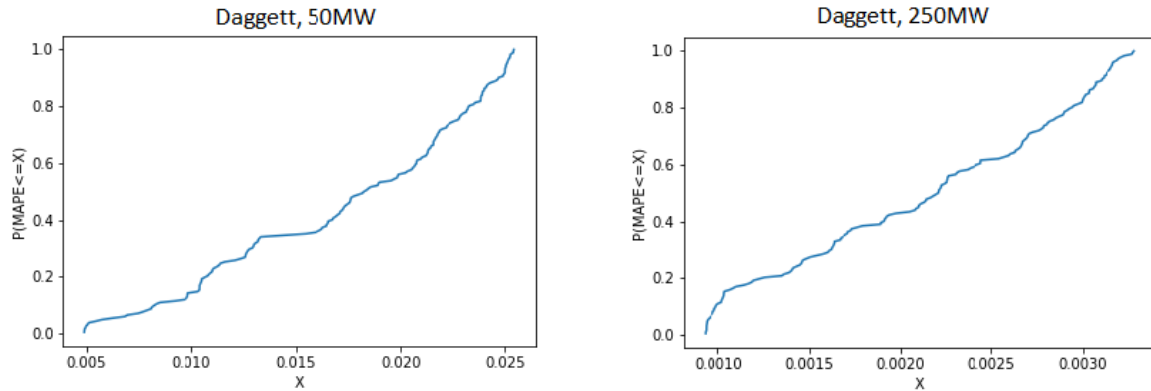


Figure 5: Yearly ECDF plots of selected hours with DNI greater than or equal to 500 W/m². The x -axis denotes the MAPE level as a fraction, and the y -axis denotes the proportion of cases that exhibit a MAPE at or below the level on the x -axis.

2.2.1 Model Formulation

Sets and Indices

$h \in \mathcal{H}$:	heliostats
$m \in \mathcal{M}$:	receiver measurement points
$m \in \hat{\mathcal{M}}_m$:	measurement points that neighbor measurement point m
$a \in \mathcal{A}$:	receiver aimpoints

Parameters

		Units
C_{hma} :	thermal flux incident to measurement point m when heliostat h aims at point a on the receiver	[W/m ²]
S_m :	surface area allocated to measurement point m	[m ²]
B_m :	upper limit on thermal flux incident to measurement point m	[W/m ²]
$D_{mm'}$:	upper limit on difference in flux incident to measurement points m and m'	[W/m ²]

Decision Variables

		Units
y_{ha} :	1 if heliostat h focuses on aimpoint a , and 0 otherwise	[unitless]
x_m :	thermal flux delivered to measurement point m	[W/m ²]

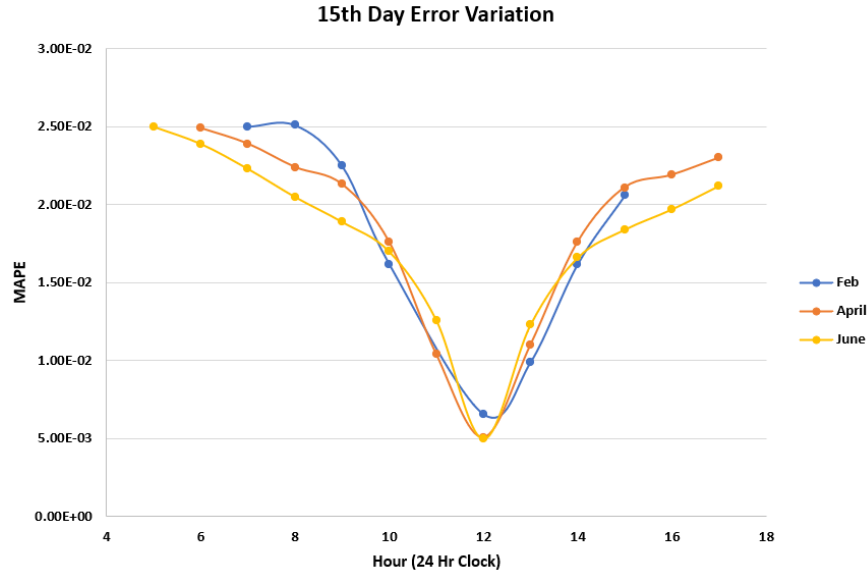


Figure 6: Summaries of average MAPE results for each hour of the day, for the 50-MW Daggett case study described above. Each line represents the summary MAPE for a different month.

Formulation

$$z^* = \max \sum_{m \in \mathcal{M}} S_m \cdot x_m \quad (1a)$$

$$\text{subject to } x_m = \sum_{h \in \mathcal{H}} \sum_{a \in \mathcal{A}} C_{hma} \cdot y_{ha}, \quad \forall m \in \mathcal{M} \quad (1b)$$

$$x_m \leq B_m, \quad \forall m \in \mathcal{M} \quad (1c)$$

$$x_m - x_{m'} \leq D_{mm'}, \quad \forall (m, m') \in \mathcal{M} \times \mathcal{M} \quad (1d)$$

$$x_m \geq 0, \quad \forall m \in \mathcal{M} \quad (1e)$$

$$y_{ha} \in \{0, 1\}, \quad \forall h \in \mathcal{H}, a \in \mathcal{A}. \quad (1f)$$

Discussion

Through the objective in (1a), we seek a collection of aimpoints for the heliostats such that the maximum power is delivered to the receiver, which we define in our approximate model as the sum of each measurement point's incident flux, times its allocated surface area. Constraint (1b) ensures that only the selected aimpoint for each heliostat is considered when calculating the total flux incident to each measurement point. Constraint (1c) limits the flux incident to each measurement point to its pre-specified bounds. Similarly, constraint (1d) restricts the thermal gradient on the receiver to specific limits. Although the flux profile on the receiver can impact its thermal efficiency [15], we do not consider efficiency gains; rather, we assume that these bounds may be used to restrict operations to those permissible within the receiver's design specifications. Constraints (1e) and (1f) provide non-negativity and binary restrictions, respectively, which ensures that variables maintain physically and logically realistic values.

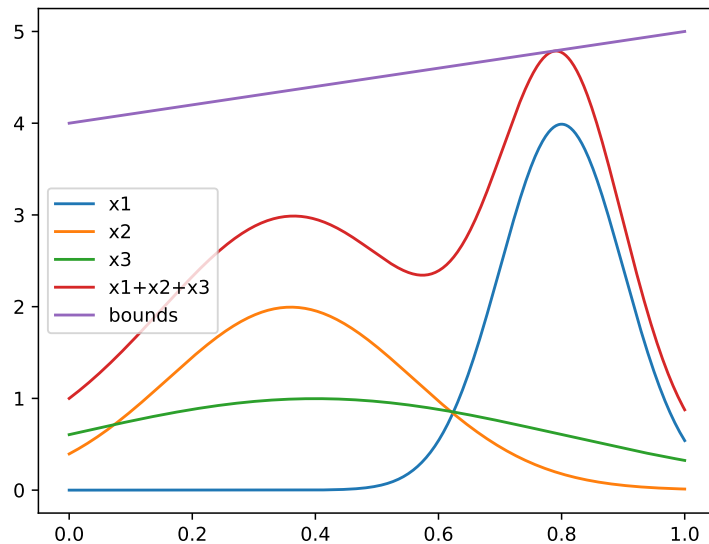


Figure 7: The optimal solution to an instance of model (1) in which the three one-dimensional flux images are Gaussian with unique variances. Here, the objective is to maximize the area under the curves captured within the interval $[0,1]$ by selecting aimpoints, which lie at the center of each curve.

2.2.2 Examples Using Gaussian Flux Images

We illustrate the value of using an optimization model to obtain aimpoints by developing two case studies that use Gaussian images. In our first case study, we consider three univariate Gaussian images on a one-dimensional receiver surface. In this example, the receiver spans the $[0,1]$ interval, the upper limit on thermal flux is defined as

$$B(y) = 4 + y$$

for $y \in [0,1]$, and the Gaussian images have standard deviations of $s_1 = 0.1$, $s_2 = 0.2$, and $s_3 = 0.4$. In this setting, the decision maker's goal is to select the aimpoint (i.e., center) of each of the curves. For now, only the upper limit on flux and bounds of $[0,1]$ for the candidate aimpoints are considered; we do not consider lower limits on flux incident on any measurement point, nor do we include the restrictions on flux gradients described in equation (1d).

Figure 7 displays the optimal solution under these conditions, with positions set to 0.79, 0.37, and 0.43, respectively. Without the upper limit, all three curves would be centered at 0.50 to maximize the total flux delivered to the interval; however, the limit pushes the more focused image to the upper limit, while the two less-focused images are shifted to the left to ensure that the flux limit is within the specified range.

Case 2: Two-Dimensional Gaussian Images on a Flat-Plate Receiver

Our second case study considers the optimized aimpoints for four Gaussian images to a flat-plate receiver surface that is 25×25 meters. Relative to the south-facing receiver surface, the heliostats are placed at the following locations:

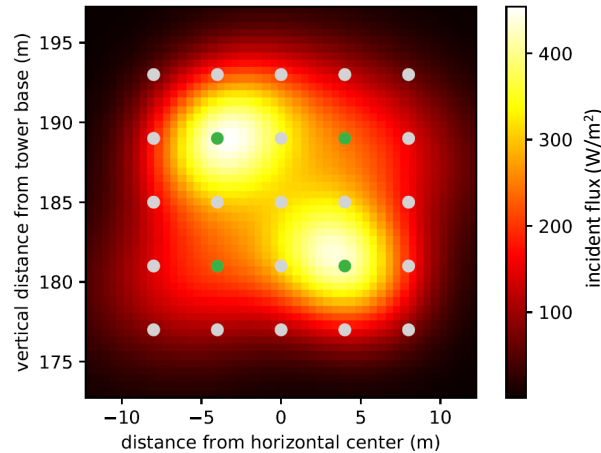


Figure 8: The optimal solution to an instance of model (1) in which four heliostats are aimed at a 25m-by-25m flat-plate receiver. Green points denote aimpoints selected by one of the four heliostats, and gray dots represent available aimpoints that were not selected. The color map on the right denotes the flux (W/m^2) incident to each measurement point on the receiver.

1. 500 meters south and 500 meters west
2. 500 meters south and 250 meters west
3. 500 meters south and 250 meters east
4. 500 meters south and 500 meters east,

and the error level for each (single-point) heliostat is 5 milliradians.

Similar to the previous example, the objective of Case 2 is to obtain aimpoint selections for these four heliostats that maximize the total power delivered to the receiver surface, subject to an upper bound on incident flux that is the same for all measurement points on the receiver. As in Case 1, we do not impose the lower bounds on incident flux in constraint (1c), nor do we place any restrictions on the thermal flux gradient per constraint (1d), at any point on the receiver.

Figure 8 displays the optimal solution under these conditions; green dots represent selected aimpoints, while gray dots represent aimpoints that were not selected for any of the heliostats. The results show that even though the center aimpoint would yield the highest power delivery for each heliostat, the flux limits preclude a strategy in which multiple images focus on the center of the receiver; moreover, in the optimal solution, none of the heliostats are aimed at the center.

2.3 Solution Methods

Because solar fields may exceed 10,000 heliostats for commercial-scale plants, these model instances can become computationally intractable with off-the-shelf solvers. To that end, we employ two optimization methods to quickly obtain high-quality solutions to the models that we develop in Section 2.2 using open-source solvers. In the first, we present a spatial decomposition of the optimization model that subdivides the problem into subproblems that we may solve in parallel. In the second, we implement an existing

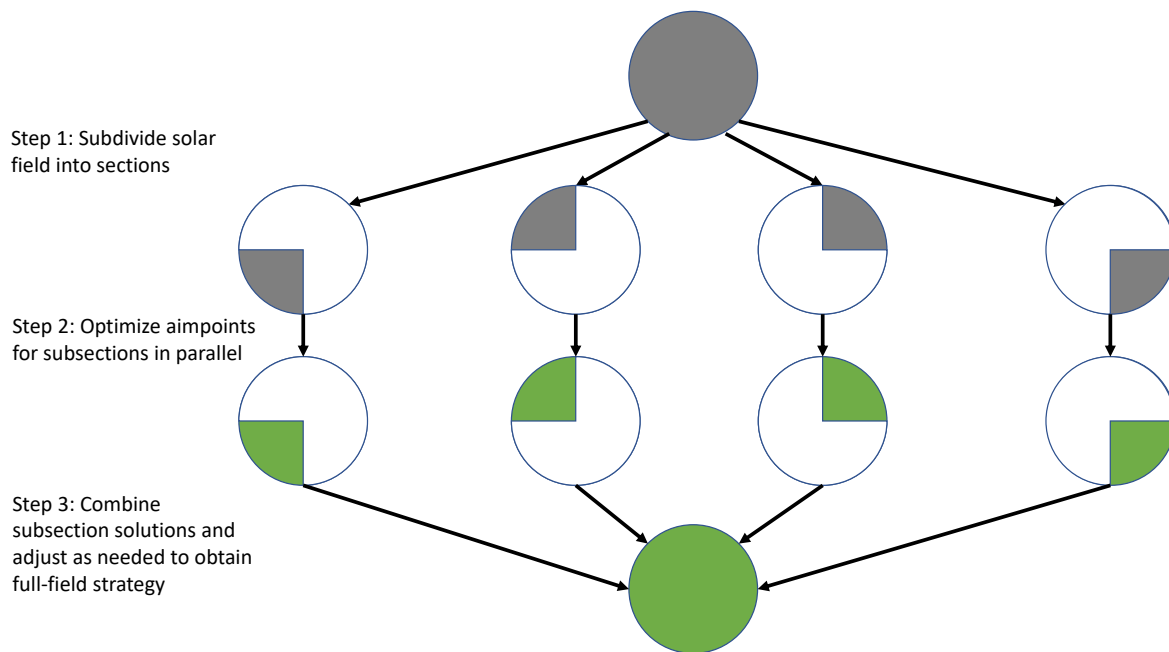


Figure 9: Flowchart describing the approach to solving large-scale layout and aimpoint strategy optimization model instances. The method will allow for subproblems to be solved in parallel, which will provide solutions more quickly than solving the large-scale instance directly.

heuristic from the literature that creates groups of heliostats which share an aimpoint.

2.3.1 Spatial Decomposition

This section describes a spatial decomposition of the aimpoint strategy optimization model. Figure 9 displays a brief overview of the methodology we adopt, which subdivides the solar field into smaller sections that for which we optimize aimpoints in parallel, and then combine the sections to obtain a solution for the entire solar field. The method allows for solutions to be obtained more quickly than solving the large-scale model instances directly. The approach we use to separate the larger optimization models that we solve in parallel is similar to the temporal decomposition described in [33], but our decomposition is spatial in nature.

In what follows, we first define the decomposed optimization model, which is equivalent to the model in Section 2.2. We then present a reformulation of the model that allows the problem to separate naturally by field; this is done by adding a restriction on the allowable flux limits by section in a manner that allows the section-specific solutions to provide a feasible aimpoint strategy. Our notation uses uppercase letters for parameters, lowercase letters for variables, and subscripts to denote elements of a set which, in turn, is represented by an uppercase, calligraphic letter.

New Sets and Indices	
$h \in \mathcal{H}_f$:	heliostats in field section f
New Parameters	
B_{mf} :	upper limit on thermal flux incident to measurement point m from field section f [W/m ²]
$D_{mm'f}$:	upper limit on difference in flux incident to measurement points m and m' from field section f [W/m ²]
New Decision Variables	
x_{mf} :	total thermal flux incident to measurement point m from field section f [W/m ²]

Decomposed Formulation

$$z^* = \max \sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}} S_m \cdot x_{mf} \quad (2a)$$

$$\text{subject to } x_{mf} = \sum_{h \in \mathcal{H}_f} \sum_{a \in \mathcal{A}_h} C_{mha} \cdot y_{ha}, \quad \forall m \in \mathcal{M}, f \in \mathcal{F} \quad (2b)$$

$$\sum_{a \in \mathcal{A}_h} y_{ha} \leq 1, \quad \forall h \in \mathcal{H}_f, f \in \mathcal{F} \quad (2c)$$

$$\sum_{f \in \mathcal{F}} x_{mf} \leq \sum_{f \in \mathcal{F}} B_{mf}, \quad \forall m \in \mathcal{M} \quad (2d)$$

$$\sum_{f \in \mathcal{F}} (x_{mf} - x_{m'f}) \leq \sum_{f \in \mathcal{F}} D_{mm'f}, \quad \forall m' \in \hat{\mathcal{M}}_m, m \in \mathcal{M} \quad (2e)$$

$$x_{mf} \geq 0, \quad \forall m \in \mathcal{M}, f \in \mathcal{F} \quad (2f)$$

$$y_{ha} \in \{0, 1\}, \quad \forall h \in \mathcal{H}, a \in \mathcal{A}_h. \quad (2g)$$

Discussion

Model 2 is equivalent to model 1, but utilizes the cloned flux decision variables in each field section, as well as the flux and gradient bound parameters that have been redefined to be section-specific. Except for the summations on both sides of constraints (2d) and (2e), model (2) separates the model naturally by field section. In the following, we present a reformulation that is more restrictive, but allows a full separation of the model by field section.

Separable Aimpoint Strategy Optimization Model Formulation

$$\underline{z}^* = \max_{\mathbf{x}, \mathbf{y}} \sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}} S_m \cdot x_{mf} \quad (3a)$$

$$\text{subject to } x_{mf} = \sum_{h \in \mathcal{H}_f} \sum_{a \in \mathcal{A}_h} C_{mha} \cdot y_{ha}, \quad \forall m \in \mathcal{M}, f \in \mathcal{F} \quad (3b)$$

$$\sum_{a \in \mathcal{A}_h} y_{ha} \leq 1, \quad \forall h \in \mathcal{H}_f, f \in \mathcal{F} \quad (3c)$$

$$x_{mf} \leq B_{mf}, \quad \forall m \in \mathcal{M}, f \in \mathcal{F} \quad (3d)$$

$$(x_{mf} - x_{m'f}) \leq D_{mm'f}, \quad \forall m' \in \hat{\mathcal{M}}_m, m \in \mathcal{M}, f \in \mathcal{F} \quad (3e)$$

$$x_{mf} \geq 0, \quad \forall m \in \mathcal{M}, f \in \mathcal{F} \quad (3f)$$

$$y_{ha} \in \{0, 1\}, \quad \forall a \in \mathcal{A}_h, h \in \mathcal{H}_f, f \in \mathcal{F}. \quad (3g)$$

Discussion

Model (3) is analogous to model (2), but separates the section-specific flux and gradient limits in constraints (3d) and (3e) for each field section. This is a restriction on the original problem, which only places a single limit for the entire field in each of the two constraints. We select field section-specific flux limits that are proportional to the field's productivity:

$$B_{mf} = B_m \cdot \frac{\sum_{h \in \mathcal{H}_f} \sum_{m' \in \mathcal{M}} C_{m'ha^c}}{\sum_{h \in \mathcal{H}} \sum_{m' \in \mathcal{M}} C_{m'ha^c}}, \quad \forall m \in \mathcal{M}, f \in \mathcal{F} \quad (4)$$

$$D_{mm'f} = D_{mm'} \cdot \frac{\sum_{h \in \mathcal{H}_f} \sum_{m'' \in \mathcal{M}} C_{m''ha^c}}{\sum_{h \in \mathcal{H}} \sum_{m'' \in \mathcal{M}} C_{m''ha^c}}, \quad \forall m' \in \hat{\mathcal{M}}_m, m \in \mathcal{M}, f \in \mathcal{F}, \quad (5)$$

in which a^c is the central aimpoint. Equations (4) and (5) proportionally scale the flux and gradients, respectively, by the ratio of the flux contribution sum for the heliostats within a given solar field section divided by the total flux contribution of the entire field, assuming all heliostats are aimed at the center of the receiver. With these new limits imposed for each field section, the model naturally separates by section, and so each section may be solved separately and in parallel.

2.3.2 Heuristic Grouping Method

We adopted a modification to the aimpoint strategy optimization model that reduces solution time significantly with limited impact on solution quality. Specifically, we pre-specify a group size, then subdivide each solar field into groups, and impose a condition that each group of heliostats must be directed to the same aimpoint on the receiver. Let $g \in \mathcal{G}$ be the set of heliostat groups, and let $h \in \mathcal{H}$ be the collection of heliostats in the solar field, as in the mathematical formulation of the aimpoint strategy optimization model discussed in previous reporting. Finally, let $\mathcal{H}_g \subseteq \mathcal{H}$ be the collection of heliostats in group g , $g \in \mathcal{G}$. Then, we include the following set of constraints in the aimpoint strategy optimization

model:

$$y_{ha} = y_{h'a}, \quad \forall a \in \mathcal{A}, (h, h') \in \mathcal{H}_g \times \mathcal{H}_g, g \in \mathcal{G} : h < h', \quad (6)$$

where y_{ha} is a binary variable that is 1 when heliostat h is directed at aimpoint a , and zero otherwise. Constraint (6) requires that all heliostats in a group focus on the same aimpoint index a . By implementing constraint (6), the revised model is equivalent to one that only has $|\mathcal{G}| \times |\mathcal{A}|$ aimpoint selection decision variables; i.e., the model size is reduced by a factor equal to the group size. Computational experiments indicate that the model solution times are reduced significantly when combined with the decomposition method described in the previous quarter; we present these experiments below in Section 2.3.3.

2.3.3 Case Study

The following case study measures the energy delivered to the receiver over the course of a representative collection of hours using the software tool we develop in this project, and compares the annual energy delivery to what is achieved when using the aimpoint strategy from SolarPILOT. We employ the following method to compare the two strategies when provided a solar field and receiver characteristics as input:

1. Let $t \in \mathcal{T}$ be the collection of representative time periods in the year for an annual analysis.
2. For each period $h \in \mathcal{H}$, let p_t be the thermal power delivered to the receiver from the aimpoint optimization model that we develop.
3. For each hour $h \in \mathcal{H}$, let q_t be the thermal power delivered to the receiver from SolarPILOT's default aimpoint strategy. SolarPILOT does not consider flux limitations; therefore, we employ the following corrective action as needed.
 - (a) Check for flux violations on the receiver at each measurement point. If no flux violations exist, return p_h and terminate. Otherwise, let \hat{m} be the measurement point of maximum flux violation.
 - (b) Let \hat{h} be the heliostat index that contributes the greatest flux to measurement point \hat{m} . Defocus heliostat \hat{h} . Return to step (3a).
4. Calculate the factor of improvement:

$$z = \frac{\sum_{t \in \mathcal{T}} (p_t - q_t)}{\sum_{t \in \mathcal{T}} q_t}.$$

We employ two case studies to demonstrate the value of our solution method. In the first, we utilize a 50-MW, flat-plate receiver, and in the second, we employ a 250-MW, cylindrical receiver. Both cases use Daggett, CA, as the candidate location and use SolarPILOT to generate the solar field layout with a design point of 950 W/m² DNI. Figure 10 depicts the solar field layout for each study; each color denotes a unique field section for which we obtain optimized aimpoints separately and in parallel, using the decomposition technique described in previous quarterly reports. We employ 8 sections for the 50-MW case and 16 sections for the 250-MW case, leveraging the 8 threads available on the laptop on which we conduct these case studies. The grouping heuristic was implemented with a group size of 4, and heliostats were grouped together by index in SolarPILOT, which is analogous to grouping them together according to distance from the receiver.

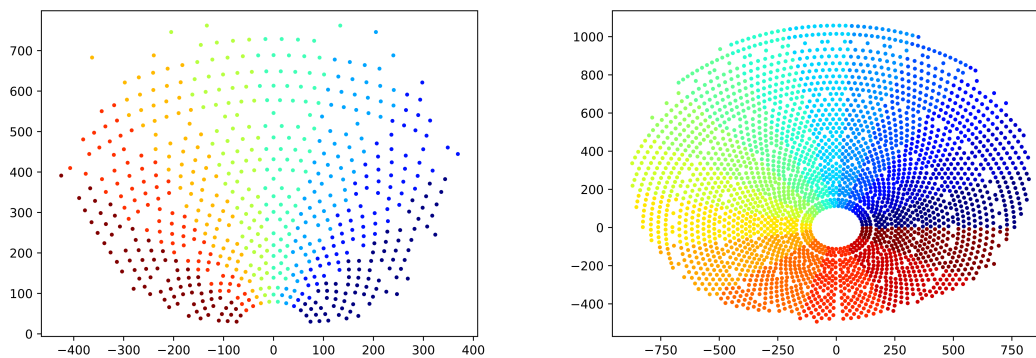


Figure 10: Layouts for the two case studies we employ to compare performance of our model vs. SolarPILOT. The layout on the left is paired with a 50-MW, flat-plate receiver, and the layout on the right is paired with a 250-MW, cylindrical receiver. Horizontal and vertical axis values correspond to distance from the center of the receiver, in meters. Each unique color denotes a field section that we solve separately (and in parallel) to obtain aimpoint strategies quickly, using the decomposition approach described in previous reports. SolarPILOT was used for flux map and layout generation in both cases.

The case studies utilize a collection of 28 time periods throughout the year that correspond to those used in SolarPILOT to estimate the annual performance of the solar field; Table 2 summarizes the time of year and DNI in each period. An open-source solver, CoinCBC, was used to obtain aimpoints with a termination criterion of either 300 seconds runtime or a gap of 0.5% between the best possible solution and the best solution found per sub-problem; solution times averaged less than 5 seconds for the 50-MW case and less than 100 seconds for the 250-MW case, and the 300-second limit was never met for any of the time periods in either case. Figure 11 displays the thermal power delivered to the receiver in both cases; the blue lines denote power delivered using our own tool, and the orange lines denote power delivered using SolarPILOT’s aimpoint strategy. The results show that our tool, Heliostat Aimpoint and Layout Optimization Software (HALOS) [32], outperforms the SolarPILOT aimpoint strategy in every period, and delivers an average of 6.6% and 6.1% more thermal energy to the receiver for the 50-MW and 250-MW cases, respectively. Notable outliers in this case are periods 9 and 21, in which HALOS outperforms SolarPILOT significantly; because no defocusing occurs during these periods, one likely explanation is that SolarPILOT’s aiming strategy exhibits greater spillage in these hours due to (i) SolarPILOT’s tendency to spread aimpoints evenly on the receiver surface, and (ii) the increased image distortion during the early-morning hours. When the outlier in hour 21 is removed, relative improvement in energy delivery remains above 5% for each case.

Figure 12 shows the aimpoints selected for one of the time periods in the case study. We note that the cylindrical receiver restricts aimpoint selection to a vertical column on the receiver for each heliostat, with the horizontal position centered on the receiver surface to maximize intercept efficiency. The results do not display a specific pattern for the aimpoint strategy, but most aimpoints are at or near the center of the receiver to minimize spillage. As DNI increases, more aimpoints spread to the outer edges of the receiver, and

Table 2: Summary of Time Periods Used for Case Study

Period	Day	Time	DNI (W/m^2)
1	Feb 2	11am	587
2	Feb 2	12pm	840
3	Feb 2	2pm	674
4	Feb 2	3pm	631
5	May 4	11am	887
6	May 4	12pm	777
7	May 4	1pm	680
8	May 4	2pm	664
9	Aug 3	6am	634
10	Aug 3	7am	841
11	Aug 3	8am	814
12	Aug 3	9am	823
13	Aug 3	10am	865
14	Aug 3	11am	947
15	Aug 3	12pm	970
16	Aug 3	1pm	813
17	Aug 3	2pm	919
18	Aug 3	3pm	861
19	Aug 3	4pm	671
20	Aug 3	5pm	716
21	Nov 3	7am	651
22	Nov 3	8am	821
23	Nov 3	9am	789
24	Nov 3	10am	785
25	Nov 3	11am	511
26	Nov 3	12pm	805
27	Nov 3	1pm	784
28	Nov 3	3pm	537

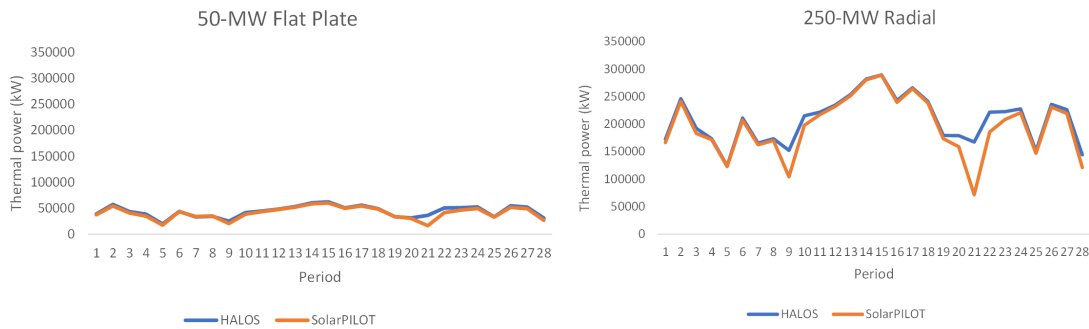


Figure 11: Thermal power delivery comparison by time period between our software tool (HALOS) and SolarPILOT for two case studies in Daggett, CA, with SolarPILOT-generated fields. The chart on the left corresponds to the 50-MW, flat-plate receiver, and the chart on the right corresponds to the 250-MW, cylindrical receiver.

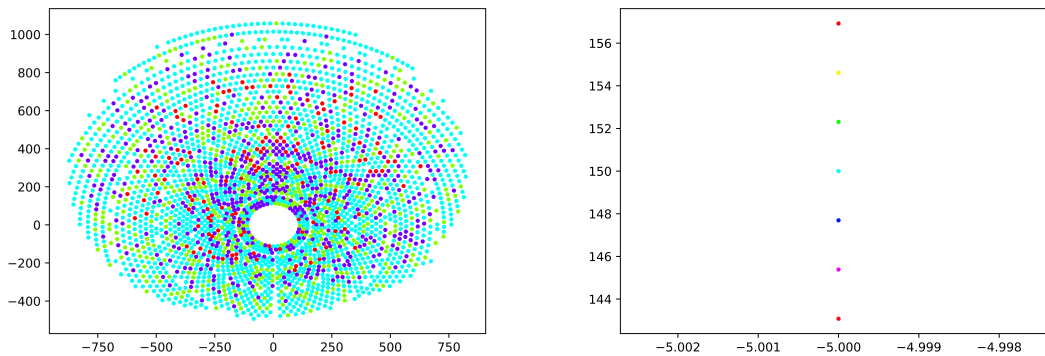


Figure 12: Aimpoints selected by our optimization model for the 250-MW case study. The layout on the left corresponds to the aimpoints selected, and the figure on the right presents the mapping between each color on the left-hand figure and its vertical position on the receiver.

defocusing starts to take place.

2.4 Layout Improvement Approach

SolarPILOT [27] includes two common options for solar field layouts, including the radial stagger, a commonly deployed layout for tower systems. Several works have attempted to improve on the radial stagger and an existing solar field, PS10 in Spain [6, 11, 28]. In recent years, a biomimetic layout has been proposed by Noone et al. [23] and studied by Zhang et al. [29] as an alternative to the radial stagger; however, the upside has been fairly limited when attempting to improve the optical efficiency alone, as Noone et al. describe a 0.5% relative improvement in optical efficiency. Moreover, while the works above explore the tradeoff between atmospheric attenuation and the impacts of blocking and shading to obtain an optimal field density, to our knowledge, no works in the literature account for utilization as a factor in the total efficiency of a heliostat. To that end, we have developed a layout improvement method that takes as input an existing layout and improves it by utilizing the aimpoint optimization model to find commonly defocused heliostats, and then

reposition them in locations that may be less optically efficient, but would be utilized more frequently, resulting in more energy delivered to the receiver.

Unlike our approach for aimpoint strategy optimization, which has been designed in a general framework and implemented in an object-oriented manner to facilitate the future implementation of alternative geometries, our layout improvement algorithm relies on specific features of SolarPILOT. Specifically, SolarPILOT first builds a radial stagger layout, then removes less efficient heliostats from the layout until a specific target energy delivery is reached. We take advantage of this feature by storing the locations of removed heliostats and their productivity, so that they may be recovered in our layout improvement scheme. Figure 13 depicts an example of the output from our procedure, whose goal is to improve the productivity of the layout by trading optical efficiency for utilization.

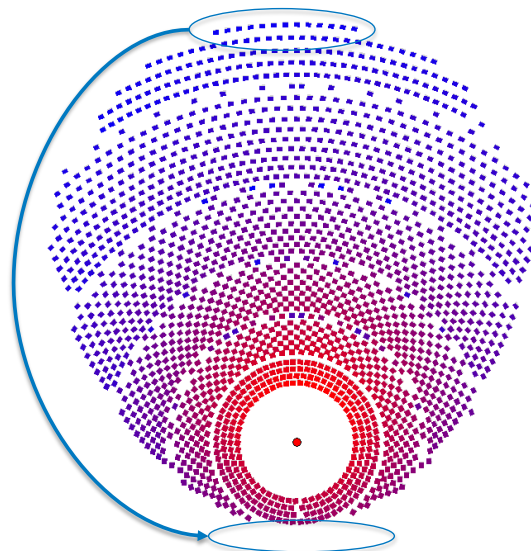


Figure 13: An example solar field layout developed by SolarPILOT, with a proposed moving of heliostats from the northern, less utilized side to the southern side, where utilization increases at the cost of optical efficiency

2.4.1 Heuristic Improvement Algorithm

Let δ and ϵ be the thresholds of the fraction of heliostats defocused in a section to qualify for heliostat removal and addition, respectively. Let p be the proportion of the defocused heliostats in the worst-case time period to relocate from a section. An overview of our procedure for layout improvement is as follows:

1. Build a radial stagger layout in SolarPILOT for a given location, and store the location and productivity of each removed heliostat.
2. Use the aimpoint strategy optimization model to obtain aimpoints in a collection of representative time periods for an annual assessment. Track heliostat utilization (i.e., the number of defocused heliostats) in each time period.
3. If heliostat utilization is at most $(1-\delta)$ in some sections and at least $(1-\epsilon)$ in others, remove the $p \cdot \delta \cdot |\mathcal{H}|$ least efficient heliostats from each section with less than 100% utilization, and add an equal number of heliostats to sections with $(1-\epsilon)$ utilization, starting with the most productive heliostats that were removed.

4. Repeat steps 2 and 3 until utilization is at least $(1-\delta)$ in every field section.

2.4.2 Case Study

Similar to the case study we employed to demonstrate the performance improvements versus SolarPILOT in Section 2.3.3, we develop a case study here to assess the performance of our layout improvement method. Specifically, we employ the following method after obtaining a solar field with a radial stagger and rejected heliostats as input from SolarPILOT:

1. Let $t \in \mathcal{T}$ be the collection of representative time periods in the year for an annual analysis.
2. For each period $h \in \mathcal{H}$, let q_t be the thermal power delivered to the receiver from the aimpoint optimization model that we develop and the field generated by SolarPILOT without any modifications.
3. Using the method described in Section 2.4, move heliostats from less-utilized sections to fully (or nearly fully) utilized sections, using $\delta = \epsilon = 0.005$.
4. For each hour $h \in \mathcal{H}$, let p_t be the thermal power delivered to the receiver from the aimpoint optimization model using the revised layout.
5. Calculate the factor of improvement:

$$z = \frac{\sum_{t \in \mathcal{T}} (p_t - q_t)}{\sum_{t \in \mathcal{T}} q_t}.$$

We use a 250-MW, cylindrical receiver in Daggett, CA, for our case study, with a design point of 850 W/m^2 ; the design point is reduced from the value of 950 W/m^2 used in the other case studies to allow SolarPILOT to oversize the field slightly. Our heuristic used groups of 4 heliostats each.

Figure 14 displays the original and revised layouts to the left and right, respectively, with red dots denoting replaced heliostats and blue dots representing new heliostats; the majority of heliostats were removed from the southern sections of the field and redistributed to the western and northern sections. In total, our layout improvement algorithm relocated 60 heliostats, or about 2% of the field, resulting in an increase in energy production of 0.25%, a level of improvement similar to that described in [23]. We expect that results would improve if the field is oversized; this is an opportunity for future work.

2.5 Software Development

This section describes the software we implement to encompass the models and solutions developed in earlier sections that will allow the CSP research community and industry partners to gain awareness of, and interest in, the solutions we developed in this project. Specifically, we describe the software framework for the package we develop to obtain optimized solar field layouts and aimpoints when provided a receiver, location, and heliostat characteristics as input. Additionally, we describe the Python API we developed for SolarPILOT, which both allow us to directly access its mature, high-fidelity flux characterization methods within the optimization package, as well as increase the ability of SolarPILOT users to directly access the myriad open-source software libraries that enable data analysis and visualization in the Python community.

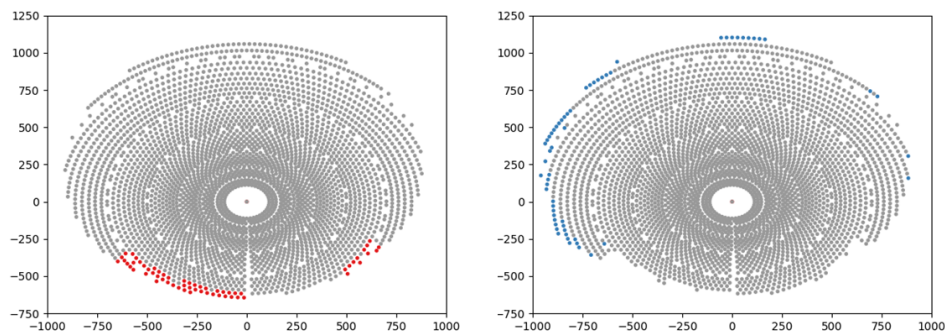


Figure 14: Original (left) and improved (right) layouts, the latter of which was obtained using the layout improvement scheme described in Section 2.4.1. Red dots denote replaced heliostats, and blue dots denote newly added heliostats when compared to the original design from SolarPILOT.

2.5.1 Optimization Software Package

We developed a Python-based software package that encompasses the entire procedure—from the reading of inputs and building of the receiver object, to the creation and solution of the optimization model, and then finally, output analysis. We created a private GitHub repository under the name Heliostat Aimpoint and Layout Optimization Software (HALOS) [32] to retain the code in a version-controlled environment; the source code is available at <https://www.github.com/NREL/HALOS>.

The software is implemented in the open-source Python programming language, which we have chosen for several reasons, including:

1. Python is an open-source language, which will allow us to make our software package open-source without additional licensing restrictions;
2. Pyomo [16], the modeling language we use to implement the models in Section 2.2, is written in Python, which allows us to seamlessly process inputs, obtain an aimpoint strategy, and generate outputs for analysis in the same language;
3. Python includes libraries that can generate high-quality visuals for output analysis, such as Plotly; and,
4. Several existing software packages, such as the System Advisor Model developed at NREL [25], include a Python API that will allow us to call other software packages directly to obtain flux profiles.

Because SolarPILOT [27] was developed at NREL and is considered a standard in solar power tower performance characterization, we have modeled the software architecture of our library to contain modules analogous to the objects in SolarPILOT. Table 3 includes a listing of each module in our library to date, plus a short description of the objects and methods contained within, and a reference to an analogous SolarPILOT object.

The scope of our project is limited to the development of a decision tool that obtains an optimized layout and aimpoint strategies for the solar field, whereas SolarPILOT performs much more detailed characterization of the system, and so we do not include all the SolarPILOT objects that are not addressed in our library; rather, we access those methods

HALOS	SolarPILOT	Notes
flux_model	ambient	calls methods from geometry, flux_method, mirror_model, and field to generate inputs for the optimization model; we currently read in weather data as a member of the flux_model object
geometry	receiver	generates different receiver geometries, including the location and normal vector of each measurement point on the receiver
flux_method	fluxsim	calculator of flux at a measurement point given heliostat location and aim point
mirror_model	heliostat	contains shape and error characteristics for a heliostat
field	SolarField	contains coordinates for candidate heliostat locations
optimize	optimize	contains the optimization model in Pyomo
plotting	NA	used for plotting results; SolarPILOT handles this through its own user interface

Table 3: Description of Modules in the HALOS Library, Including a Reference to Analogous Objects in SolarPILOT

directly through the SolarPILOT Python API that we developed and describe below in Section 2.5.3.

Populating HALOS Instances With SolarPILOT

With weather, solar field and receiver characteristics serving as input, HALOS may obtain flux maps from SolarPILOT to serve as input to the optimization models described in 2.2. SolarPILOT may also be used to generate a solar field if heliostat coordinates are not available. HALOS interacts with SolarPILOT via the Python API that we summarize below in Section 2.5.3 and describe in detail in [14]. The connection is one-way; i.e., SolarPILOT currently cannot call HALOS to obtain an aimpoint strategy at this time. Developing this connection may be considered for future work.

2.5.2 Flux Calculations Within HALOS

HALOS may also be used to directly create the flux maps that serve as input to the aimpoint strategy optimization model. Figure 15 displays a collection of the base and derived objects currently in the tool. This hierarchical structure allows for a potential user to add new receiver geometries, mirror models, or other new features into our tool as a standalone object and then use our tool to obtain optimized layouts and aimpoint strategies. Currently, only Gaussian flux maps are available within HALOS for cylindrical and flat-plate receivers.

2.5.3 SolarPILOT Python Interface

SolarPILOT is an open-source, C++ software tool that generates and characterizes power tower CSP systems. SolarPILOT can simulate receiver flux distributions using two methods: (i) a Hermite expansion technique (analytical) and (ii) a ray-tracing technique called SolTrace. The Hermite method enables SolarPILOT to accurately simulate large heliostat

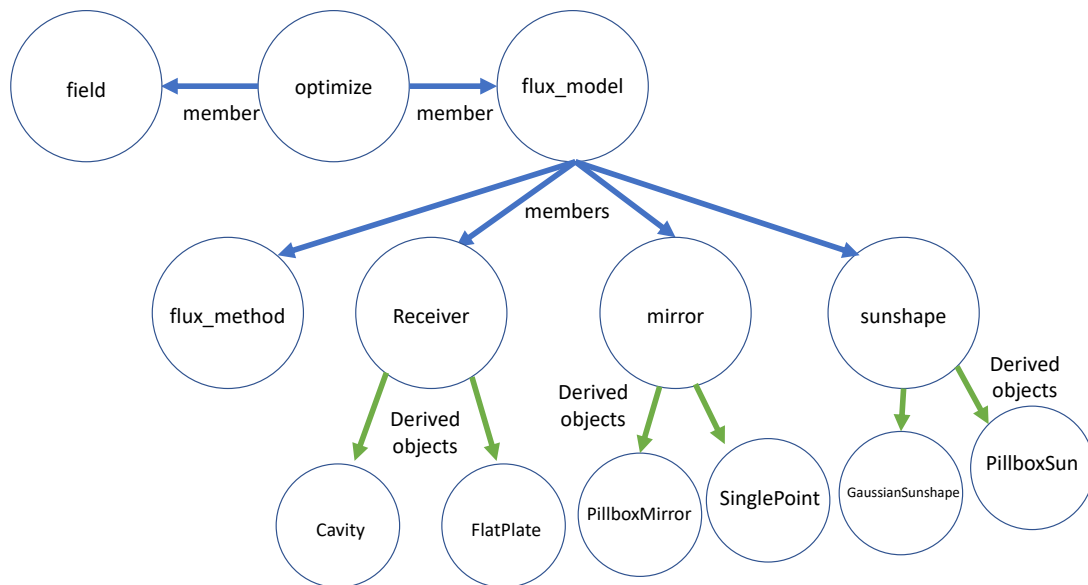


Figure 15: Flowchart describing the structure of our tool when used to generate flux maps internally. The structure allows users to create new features, such as a new receiver geometry, and then use the tool to obtain optimized layouts and aimpoint strategies.

fields in a quick and computationally efficient manner, while SolTrace provides a robust, Monte-Carlo-based, ray-tracing method that allows for the cross-comparison of results and the analysis of more complex geometries.

SolarPILOT is a well-known and frequently used software in the CSP research community to simulate power tower systems. Previously, SolarPILOT users were limited to interacting with the software through either the graphical user interface (GUI) or a Language Kit (LK) application programming interface (API), shown in Figure 16 on the left. The GUI provides an interactive visual method for users to update variables and explore results; however, creating multiple cases and performing large parametric analysis can be cumbersome and time consuming. The LK-API overcomes these challenges by providing the ability to create scripts that execute SolarPILOT’s computational methods. However, LK is a domain-specific language that requires SolarPILOT’s GUI to operate in the background, has a limited documentation and user support network, and has very limited capabilities for data analysis and visualization. Users can utilize SolarPILOT’s computational methods through the System Advisor Model (SAM), which enables receiver fluid flow path calculations; however, this interface is even more restrictive than the above two methods.

To address these issue and increase accessibility, we have developed and tested a SolarPILOT Python API, which we hereafter refer to as Co-Pylot (shown in Figure 16 on the right with the new API link highlighted in red). For Co-Pylot’s development, we created new C++ source code to export SolarPILOT’s functions from its dynamic-link library (DLL). In addition to being accessed by Python through the Co-Pylot API, these exported functions may be accessed by other scripting languages, e.g., MATLAB[®], by creating an API within the specific language to handle data type conversions. While we did not create

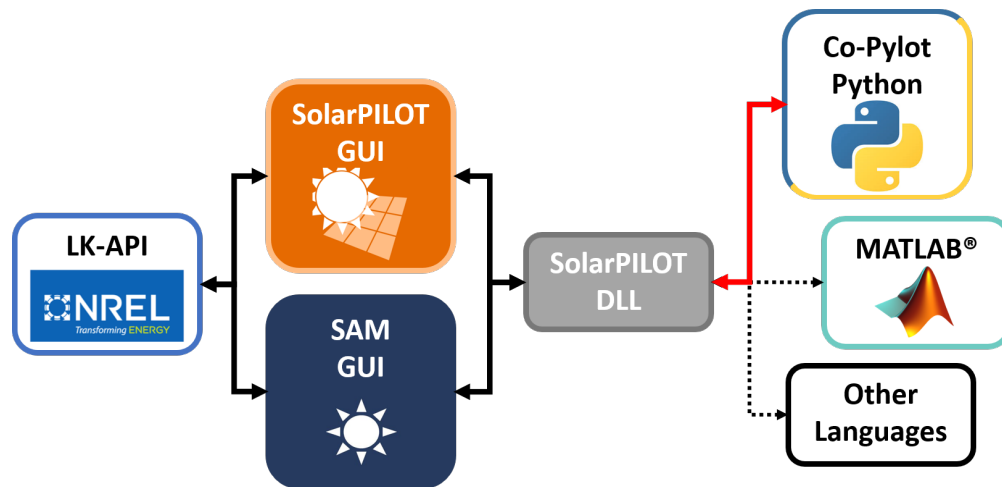


Figure 16: A diagram presenting user interfaces for SolarPILOT's dynamic-link library (DLL)

these links within this project, the development of these APIs would be straightforward with the existing C++ source code, and could be developed in future projects.

Co-Pylot enables users access to all of SolarPILOT's functionality through a Python scripting interface, which is the most popular programming language for data science and analytics. Co-Pylot's functionality includes:

- Creating and destroying an instance of SolarPILOT that includes a callback functionality to propagate messages from SolarPILOT back to Python users
- Getting and setting variables within SolarPILOT including importing custom land boundaries for field layout
- Creating receiver and heliostat objects for analysis of fields with multiple receivers and/or heliostat templates
- Generating, assigning, and modifying heliostat field layouts including the ability to set individual heliostat locations, aimpoints, soiling, and reflectivity
- Simulating heliostat field performance
- Returning SolarPILOT detailed results for individual heliostat performance, aggregated field performance, and receiver solar flux distribution
- Saving the python created SolarPILOT instance to either a .csv file with all variable values or a SolarPILOT .spt file that can be read by SolarPILOT's GUI.

In addition to Co-Pylot's functionality with SolarPILOT, Python users have access to the more than 100,000 open-source libraries to develop, analyze, optimize, and visualize CSP tower research utilizing SolarPILOT's computation engines. Co-Pylot's functionality is limited to working with the standalone SolarPILOT library, and does not currently allow for interfacing with the System Advisor Model [12] or its other submodels. More details on the API are described in [14].

3 Conclusions and Future Work

This work detailed in this report outlines the outcomes that we have achieved in developing a software package that obtains optimized aimpoint strategies and uses the resulting

utilization metrics to improve solar field layouts.

Specifically, we have:

1. Developed a mixed-integer linear programming model that obtains optimized aimpoint strategies for CSP tower plants
2. Implemented a high-quality approximation of the high-resolution images developed by packages like SolarPILOT to reduce the computing time required to generate instances of our model
3. Improved solution times for model instance via a spatial decomposition approach that subdivides large-scale plants into sections, for which optimized aimpoint strategies may be obtained separately and in parallel
4. Demonstrated significant improvement in annual energy delivered to the receiver using our aimpoint optimization software, when compared to SolarPILOT's aiming strategies, in multiple case studies
5. Identified a novel metric for layout improvement that weights heliostat production by its utilization
6. Created an algorithm that uses utilization statistics by section to relocate heliostats in the solar field, increasing overall energy production
7. Implemented the optimization methods in a publicly available software package
8. Developed a Python API for SolarPILOT to expand the package's accessibility to both our software package and open-source data analysis tools.

Recommendations for future work include expanding the collection of geometries that can be generated in HALOS, additional analysis of aimpoint strategies under different operating conditions (e.g., partial cloud cover), extension of the optimization model to alter existing aimpoints in response to receiver hotspots, active transition of the software to the public domain, and further refinement and documentation of the software package to enhance functionality and usability.

The following manuscripts were produced as part of the efforts described in this report:

- Hamilton, W., Wagner, M., and Zolan, A. (2021). Demonstrating SolarPILOT's Python API Through Optimal Aimpoint Strategy Use Case. *Proceedings of ASME 15th International Conference on Energy Sustainability (ES2021)*, Accepted.
- Zolan, A., Hamilton, W., Liaqat, K., and Wagner, M. (2021). A Spatial Decomposition Approach to Optimizing Aimpoint Strategies for Commercial-Scale Concentrating Solar Power Tower Plants. *In preparation*.
- Zolan, A. and Wagner, M. (2021). A Utilization-weighted Efficiency Metric for Solar Field Layout Improvement. *In preparation*.

The source code for HALOS is available at <https://www.github.com/NREL/HALOS>.

Acknowledgments

In addition to the co-authors, the following participants have contributed significantly to the composition of this report and research accomplishments herein:

- Dr. Janna Martinek, National Renewable Energy Laboratory
- Alex Mikulich, Colorado School of Mines.

The authors gratefully acknowledge Bill Schreckengost, Michel Izygon, and Fabian Gross,

who provided feedback on our optimization methods and software tool.

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number 35930. This paper was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

4 References Cited

- [1] Armstrong, P. and M. Izygon (2014). An innovative software for analysis of sun position algorithms. *Energy Procedia* 49, 2444–2453.
- [2] Ashley, T., E. Carrizosa, and E. Fernández-Cara (2017). Optimisation of aiming strategies in solar power tower plants. *Energy* 137, 285–291.
- [3] Atif, M. and F. A. Al-Sulaiman (2015). Optimization of heliostat field layout in solar central receiver systems on annual basis using differential evolution algorithm. *Energy Conversion and Management* 95, 1 – 9.
- [4] Barberena, J., A. M. Larrayoz, M. Sánchez, and A. Bernardos (2016). State-of-the-art of heliostat field layout algorithms and their comparison. *Energy Procedia* 93, 31–38.
- [5] Belhomme, B., R. Pitz-Paal, and P. Schwarzbözl (2014). Optimization of heliostat aim point selection for central receiver systems based on the ant colony optimization metaheuristic. *Journal of solar energy engineering* 136(1), 011005.
- [6] Carrizosa, E., C. Domínguez-Bravo, E. Fernández-Cara, and M. Quero (2015). A heuristic method for simultaneous tower and pattern-free field optimization on solar power systems. *Computers & Operations Research* 57, 109 – 122.
- [7] Collado, F. J. and J. Guallar (2012). Campo: Generation of regular heliostat fields. *Renewable Energy* 46, 49 – 59.
- [8] Collado, F. J. and J. Guallar (2013). A review of optimized design layouts for solar power tower plants with campo code. *Renewable and Sustainable Energy Reviews* 20, 142–154.
- [9] Cruz, N., J. Álvarez, J. Redondo, M. Berenguel, and P. Ortigosa (2018). A two-layered solution for automatic heliostat aiming. *Engineering Applications of Artificial Intelligence* 72, 253 – 266.

- [10] Cruz, N., J. L. Redondo, J. D. Álvarez, M. Berenguel, and P. M. Ortigosa (2017). A parallel teaching–learning-based optimization procedure for automatic heliostat aiming. *The Journal of Supercomputing* 73(1), 591–606.
- [11] Cruz, N., S. Salhi, J. Redondo, J. Álvarez, M. Berenguel, and P. Ortigosa (2018). Hector, a new methodology for continuous and pattern-free heliostat field optimization. *Applied Energy* 225, 1123 – 1131.
- [12] Freeman, J. M., N. A. DiOrio, N. J. Blair, T. W. Neises, M. J. Wagner, P. Gilman, and S. Janzou (2018). System Advisor Model (SAM) general description (version 2017.9.5). Technical report, National Renewable Energy Laboratory (NREL), Golden, CO (United States).
- [13] Grobler, A. and P. Gauché (2014). A review of aiming strategies for central receivers. In *Proceedings of the second Southern African Solar Energy Conference*.
- [14] Hamilton, W. T., M. J. Wagner, and A. J. Zolan (2021). Demonstrating SolarPILOT's Python API through heliostat optimal aimpoint strategy use case. *Proceedings of ASME 15th International Conference on Energy Sustainability (ES2021)*, Accepted.
- [15] Harris, J. A. and T. G. Lenz (1985). Thermal performance of solar concentrator/cavity receiver systems. *Solar Energy* 34(2), 135–142.
- [16] Hart, W. E., C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Sirola (2017). *Pyomo-optimization modeling in python*, Volume 67. Springer.
- [17] Izygon, M., P. Armstrong, C. Nilsson, and N. Vu (2011). TieSOL—a GPU-based suite of software for central receiver solar power plants. *Proceedings of SolarPACES*.
- [18] Izygon, M., K. McMurtrie, and N. Vu (2018). Particle swarm optimization of the layout of a heliostat field. In *AIP Conference Proceedings*, Volume 2033, pp. 040018. AIP Publishing.
- [19] Kistler, B. L. (1986). A user's manual for DELSOL3: A computer code for calculating the optical performance and optimal system design for solar thermal central receiver plants. *Sandia National Laboratories, Sandia Report No. SAND86-8018*.
- [20] Kuhnke, S., P. Richter, F. Kepp, J. Cumpston, A. M. Koster, and C. Büsing (2019). Robust optimal aiming strategies in central receiver systems. *Renewable Energy*.
- [21] Li, C., R. Zhai, H. Liu, Y. Yang, and H. Wu (2018). Optimization of a heliostat field layout using hybrid pso-ga algorithm. *Applied Thermal Engineering* 128, 33 – 41.
- [22] Lipps, F. and L. Vant-Hull (1978). A cellwise method for the optimization of large central receiver systems. *Solar Energy* 20(6), 505–516.
- [23] Noone, C. J., M. Torrilhon, and A. Mitsos (2012). Heliostat field optimization: A new computationally efficient model and biomimetic layout. *Solar Energy* 86(2), 792–803.

- [24] Teter, M. D., A. M. Newman, and M. Weiss (2016). Consistent notation for presenting complex optimization models in technical writing. *Surveys in Operations Research and Management Science* 21(1), 1–17.
- [25] Turchi, C. S. and G. A. Heath (2013). Molten salt power tower cost model for the system advisor model (SAM). Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).
- [26] Vittitoe, C. and F. Biggs (1981). A user's guide to HELIOS: A computer program for modeling the optical behavior of reflecting solar concentrators. *Sandia National Laboratories, Albuquerque, NM, SAND81-1180*.
- [27] Wagner, M. and T. Wendelin (2018). SolarPILOT: A power tower solar field layout and characterization tool. *Solar Energy* 171, 185–196.
- [28] Wang, J., L. Duan, and Y. Yang (2018). An improvement crossover operation method in genetic algorithm and spatial optimization of heliostat field. *Energy* 155, 15 – 28.
- [29] Zhang, M., L. Yang, C. Xu, and X. Du (2016). An efficient code to optimize the heliostat field and comparisons between the biomimetic spiral and staggered layout. *Renewable Energy* 87, 720–730.
- [30] Zhou, Y. and Y. Zhao (2014). Heliostat field layout design for solar tower power plant based on GPU. *IFAC Proceedings Volumes* 47(3), 4953 – 4958. 19th IFAC World Congress.
- [31] Zolan, A., W. Hamilton, K. Liaqat, and M. Wagner (2021). A spatial decomposition approach to optimizing aimpoint strategies for commercial-scale concentrating solar power tower plants. In preparation.
- [32] Zolan, A., W. Hamilton, K. Liaqat, M. Wagner, and USDOE Office of Energy Efficiency and Renewable Energy (2021). HALOS (Heliostat Aimpoint and Layout Optimization Software).
- [33] Zolan, A. J., M. S. Scioletti, D. P. Morton, and A. M. Newman (2021). Decomposing loosely coupled mixed-integer programs for optimal microgrid design. *INFORMS Journal on Computing*.