Title

Evaluation of Agrivoltaic Systems for Enhanced Agricultural Resource Sustainability

Author

Samara, Omar Alfredo

Publication Date

2024

Peer reviewed|Thesis/dissertation

Evaluation of Agrivoltaic Systems for Enhanced Agricultural Resource Sustainability

By

OMAR ALFREDO SAMARA
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Biological Systems Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Isaya Kisekka, Chair

Md. Shamim Ahamed

Helen Dahlke

Committee in Charge

2024

Acknowledgments

In the name of Allah, the Most Gracious, the Most Merciful. All praise be to Allah.

This has been a long journey, and it is hard to believe it is over. I have so many people to thank and many favors I can never repay. I hope this work can serve as my attempt to show them the struggles were worth it and that this knowledge can benefit society and pay back everyone's help by paying it forward to the world. To any of you reading this who are acknowledged by name or spirit, thank you. And to those of you reading with the intention to learn, I hope this work can benefit you, and you can, in turn, benefit the world. So, before we begin exploring this dissertation, I would like to say my thanks to many of those who helped me get here.

My sincere thanks also go to my committee members, Isaya Kisekka, Shamim Ahamed, and Helen Dahlke. Their support and guidance through this degree have kept me moving through grad school, and the opportunities they provided invaluably contributed to my academic development. The classes they taught me, the classes they let me teach, and the resources for conducting research were all critical in getting this work done, and I am deeply thankful.

I would also like to thank my lab mates over the years: Tom Snowden, Floyid Nicolas, Usama Al-Dughaishi, Kelley Moyers, Chevon Holmes, Iael Raij Hoffman, Abir Ahsan, Charlie Chen, Peter Savchik, Will Lennon, Liyuan Yang, Nick Claypool, Ferisca Putri, Yihe Zou, Felix Ogunmokun, Srini Peddinti, Hayley Amo, Wyatt Northcut, Kinu Koide, Jake Calgari, Jessie Fairbanks, Aaron Fernandez, Kaylie Grundmeyer, Charles Jannsen, Alex Kerlee, Gerardo Medina, John Stepp, Erik Swanson, Craig Valdez, Rishikesh Kaylan, Zoe Wilf, Maxwell Wright, Hayden White,

and John Palomari. Everyone's help and camaraderie were instrumental in getting me to this point.

Specific thanks need to go to Winton Smalley for editing my papers over the years. Without Winton's editing, it is unclear if my writings would be intelligible to the average person.

There have also been undergraduates who have helped me along the way, both with respect to getting work done and in providing me opportunities to improve my abilities as a mentor, and thanks are also due to them: Ayan Behjat, Hanna Wang, Carolin Hallock, Mace Worthington, and Josh Kuik.

Others have also served as advisors and mentors to me, and I would like to thank them for all their help: Dan Frank, Dave Fujino, Victor Duraj, and Heiner Leith. Dan's mentorship was all but short of having a 4th committee member, and I am thankful and honored for his friendship, his mentorship, and the opportunities he provided me. My family is also due infinite thanks. The financial and emotional support provided over the years, particularly from my parents, Yaser Samara and Roxana Nessi, is truly the only reason I was able to get through this Ph.D. There are not enough words or enough pages in this document for me to write my thanks, so I hope the brevity is sufficient. I'd also like to thank my grandparents, Juan and Elena Nessi, for their support. And while my grandmother is no longer with us, I'm sure she's very proud of these accomplishments, and I'm very grateful for all the love and care she gave me while I was growing up. It is truly impossible to express my appreciation and pay back what I owe to my family for all they've given me to get to this point.

Lastly, my friends also deserve thanks. I'm very thankful to say there are too many friends to list here (we'd be writing another 30 or 40 pages with your names). So just know, I know you and you are written in to the spirit of this list. And to you all, it is been a long journey. We've been invested in this together for almost a decade, and I'm proud to say we've finally made it.

Funding sources are also acknowledged: The National Science Foundation Bridge to the Doctorate Program Fellowship which provided me funding for my first two years of grad school. The Verizon Foundation UAP 2016 Project Grant for Sensor and communication technology for plant water stress-based irrigation management in field crops, which funded purchase of solar panels and a temperature sensor. The USDA National Institute of Food and agriculture Sustaining Groundwater and Irrigated Agriculture in the Southwestern United States under a Changing Climate grant, which funded travel to a DSSAT educational conference at a GSR salary. My advisor, Isaya Kisekka, for allowing me to utilize his tomato experiment to conduct my own experiments in a part of the field. Dan Frank, Victor Duraj, Burt Vinucci for allowing me access to fabrication tools, construction equipment, spare parts, and the Western Center for Agricultural Equipment. Mallika Nocco for access to the LICOR Li-6800 Portable Photosynthesis Measurement Device. Mark Lundy for access to infrared net radiometers (of which data was collected but not presented here). And Mathew Gilbert for access to light measurement sensors (also data collected but not presented here).

To conclude, I'd like to dedicate this work to a free Palestine.

Abstract

Climate change and agricultural water use regulation are putting increasing pressure on agricultural systems to become more sustainable while maintaining economic viability. Agrivoltaics (where solar panels are placed over crops) are a novel, proposed method to increase agricultural water and resource sustainability through the dual use of agricultural land. Given how new agrivoltaic systems are, there is a limited (although rapidly expanding) body of work relating to agrivoltaic design or their operations and impacts on agricultural systems. As such, there are no standard methods for designing agrivoltaic systems, and it isn't easy to understand how changing specific design parameters will impact agrivoltaic performance. To contribute to agrivoltaic knowledge, this research investigated the following objectives: 1) evaluate the impact of agrivoltaic system design on tomato productivity under field conditions by measuringplant growth and development (e.g., yield, photosynthesis, crop development), 2) develop a modeling framework for simulation and optimization of agrivoltaic designs, and 3) assess the potential of agrivoltaics in California through geospatial analysis. Chapter 1 of this work addresses Objective 1 through a multi-year agrivoltaics experiment in which processing tomatoes (*Lycopersicon esculentum* Mill.) were grown in a custom-designed agrivoltaic system and seasonal crop growth, photosynthesis rates, yield, and quality data were measured. Chapter 2 aims to address Objective 2 by utilizing findings from Chapter 1 and developing a 3D agrivoltaic crop model to capture the nuances of hourly variations in light distribution through an agrivoltaic system to produce a digital replica model, and then using that model in conjunction with a genetic algorithm to optimize agrivoltaic systems determin-

istically without needing to develop mechanistic driven differentiable equations of agrivoltaic performance. Chapter 3 addresses Objective 3 through geospatial analysisof desirable agrivoltaic performance parameters on a state scale and leveraging existing data sets to calculate agrivoltaic performance (such as suitability, acreage, water savings, energy production, energy savings, and carbon dioxide offsets) and the potential agrivoltaics could have in the future of California.

Chapter 1 results showed that yields under agrivoltaic were significantly reduced in both 2021 ($87.7 \pm 9.5\ t/ha$ for the control and $58.8 \pm 44.5\ t/ha$ for the agrivoltaic treatment) ($35.5 \pm 9.5\ t/ha$ for the control to $23.8 \pm 18\ t/ha$ for the agrivoltaic treatment) and 2022 with a control yield of $72.3 \pm 48.4\ t/ha$ to $51.9 \pm 9.69\ t/ha$ of agrivoltaic yield with a control yield of $72.3 \pm 19.6\ t/ha$ to $21.0 \pm 3.92\ t/ha$ of agrivoltaic yield. Harvest quality of the processing tomatoes was not statistically significant except for color in 2022. Photosynthesis measurements were found to be statistically significant for photosynthetically active radiation levels of 0, 50, 150, 300, 1500, and 2000 $\frac{\mu mol}{m^2 s}$ of PAR.

Chapter 2 used the field data collected in Chapter 1 to construct a 3D crop model using Helios Plant Simulation Software to model agrivoltaic performance at an hourly-time step and also validated the model against experimental data. Agrivoltaic performance parameters (e.g. yield, light use efficiency, power production) were established along with model constraints (e.g. unacceptable yield losses, land use efficiency requirements, self-shading limitations) to rapidly optimize agrivoltaic designs using a genetic algorithm. Numeric simulations showed that the genetic algorithm was able to find an optimized design in just 6 generations of 15 designs per

generation, with only a 6% reduction in yield compared to the reference crop and a 20% reduction in potential photovoltaic power production throughout the season.

Chapter 3 developed a GIS framework for mapping and evaluating suitable locations for agrivoltaics across California. This work determined that under a 1% adoption rate, 129,500 *hectares* could be developed for agrivoltaic production, producing potentially 29,000 GWh/yr, and reducing potentially irrigation demands by 65,374,000 m^3/yr and energy demand from irrigation by 45 GWh/yr.

While the experiment was limited to a specific crop under a specific panel design, the modeling and optimization is done for fixed panel designs, and there is still a lot of uncertainty in geospatial analysis, this work presents a comprehensive analysis of the potential of agrivoltaics in enhancing agricultural resource sustainability in California and beyond.

Contents

Acknowledgments **ii**

Abstract **v**

List of Figures **xii**

List of Tables **xiii**

Introduction **1**

Literature Review 1

Objectives 4

Hypotheses 5

Research Approach 8

Objective 1: Experimental Evaluation of Processing Tomatoes (*Lycopersicon esculentum* Mill.) Grown Under Agrivoltaic Production 8

Objective 2: Genomic Optimization of Fixed Panel Agrivoltaic Systems Using an Agrivoltaic - Responsive Crop Model at an Hourly Timestep 12

Objective 3: Evaluating Agrivoltaic Potential in California 15

Chapter 1: Experimental Evaluation of Processing Tomatoes (*Lycopersicon esculentum* Mill.) Grown Under Agrivoltaic Production **17**

Abstract 17

Introduction 18

Materials and Methods 20

Agrivoltaic Design 20
Description of Crop and Field 21
Data Collection 22
Results 25
Leaf Area Index and Canopy Cover 25
Yield 29
Photosynthesis 32
Discussion 34
Conclusion 37

Genomic Optimization of Fixed Panel Agrivoltaic Systems Using an Agrivoltaic - Responsive Crop Model at an Hourly Timestep 38

Abstract 38
Introduction 39
Methods 41
Crop Modeling 43
Plant Growth 45
Crop Metrics 49
Photovoltaic Modeling 49
Genomic Optimization 52
Cost Function 52
Performance Metrics 53
Penalty Terms 54
Genetic Algorithm 56

Calibration and Validation 59
Numeric Example 61
Crop Production Example in Northern California 62
Results 63
Discussion 68
Findings 68
Future Extensions 69
Conclusions 71

Chapter 3: Evaluating Agrivoltaic Potential in California **73**
Abstract 73
Introduction 74
Methods 75
Evaluation Criteria 76
Analysis Model 78
Model Assumptions 81
Data Sources 87
GIS Analysis Methods 91
Results 92
Discussion and Conclusion 99

Conclusions **105**
Summary 105
Chapter 1 Findings 105

Chapter 2 Findings 107
Chapter 3 Findings 108
Future Work 109
Conclusion 111
Closing Statements 111

Appendix **113**

Chapter 1 113
2021 Harvest Data 113
2022 Harvest Data 114
2021 Photosynthesis Data 115
2022 Photosynthesis Data 116
2021 Canopy Cover Data 117
2022 Canopy Cover Data 119
2021 Leaf Area Index Data 120
2022 Leaf Area Index Data 121
Chapter 2 122
Read Me File 122
Genetic Algorithm Script 122
Manager Script 136
Best Design Script 138
Control Script 152
Agrivoltaic Script 163

References **176**

List of Figures

1 General Relationship Between Light, Photosynthesis and Evapotranspiration 3
2 Agrivoltaic unit used in the experiment in the field with tomatoes growing underneath at the University of California Davis research farm near Davis, California. Photo by [Mistry, 2021] 19
3 Agrivoltaic System Design and Installation 21
4 Leaf Area Index as a Function of Growing Degree Days 27
5 Canopy Cover as a Function of Growing Degree Days 28
6 2021 and 2022 yield results detailing total biomass, fruit biomass, ratio of fruit to total biomass, pH, Brix and color metrics 30
7 Photosynthetic light response of control and agrivoltaic treatments . . 33
8 Agrivoltaic Designs 40
9 Shape of Complex Functions used in Model 48
10 Helios rendering of agrivoltaic design used in validation 61
11 Random Designs 62
12 Control and agrivoltaic crops 64
13 Daily crop performance 65
14 Hourly average crop performance 66
15 Genetic Algorithm Performance 67

16 Input Data Sets - Part 1 (Crop Classifications, GEM Suitability, Water Shortage Vulnerability and Global Horizontal Irradiance) 90
17 Input Data Sets - Part 2 (Evapotranspiration and Ground Water Depth) 91
18 Suitability Map for Agrivoltaics in California 94
19 County Average Suitability Map 101
20 County Land Use Efficiency Map 102
21 County Power Production Map 103
22 County Water Savings Map 104

List of Tables

1 LAI Curve Fitting 26
2 Canopy Cover Curve Fitting 26
3 Outcomes of crop yield and quality for control and agrivoltaic treatments 31
4 Light use efficiency and calculation inputs 31
5 Photosynthetic light response measurment summary 32
6 Photosynthetic light response curve fitting parameters 33
7 Basic information for calibration and validation 60
8 Crop model calibration parameter values used in the numeric example 60
9 Validation results used in evaluating model accuracy and error 61
10 Table with Basic Information for Tomato Production in Northern California 63
11 Table with agrivoltaic design settings 63

12 Summary table of agrivoltaic performance 64
13 Table of Numeric Assumptions Used in the Analysis 82
14 Crop Classification Suitability Scores Used in the Analysis 86
15 Statewide Performance Statistics by Crop Classification 93
16 County Statistics (A-P) 95
17 County Statistics (R-Y) 96
18 2021 Harvest Data 113
19 2022 Harvest Data 114
20 2021 Light Response Measurements 115
21 2022 Light Response Measurements 116
22 2021 Canopy Cover Measurements - Part 1 117
23 2021 Canopy Cover Measurements - Part 2 118
24 2022 Canopy Cover Measurements 119
25 2021 Leaf Area Index Measurements 120
26 2022 Leaf Area Index Measurements 121

Introduction

Literature Review

Agrivoltaics are a novel form of agricultural production where photovoltaic solar panels are placed over agricultural crops [Goetzberger and Zastrow, 1982] for a variety of purposes, primarily to improve land use efficiency [Valle et al., 2017], [Amaducci et al., 2018] and to manage the energy balance of the system to optimally produce crops optimally. While it would seem counter-intuitive to shade crops since less typically sun equals less photosynthesis, photosynthesis (in actuality) is a complex process dictated by light, temperature, and CO_2 levels. Assuming the crop receives sufficient water at high light levels, photosynthesis rates are limited by CO2 concentrations meaning additional sunlight does not result in additional photosynthesis [von Caemmerer et al., 2009], [Senevirathna et al., 2003], [Senevirathna et al., 2008], [Siles et al., 2011]. This means there is some maximum light level, or light saturation point, wherein excess amounts of light will increase heat stress on the plants (reducing photosynthetic efficiency), and increases plant transpiration (Figure 1), resulting in increased irrigation demand which has negative consequences in water-stressed places like California [Chang and Bonnette, 2016], [Prugh et al., 2018]. Simultaneously, climate change is putting increasing pressure on the world to remove CO_2 from the atmosphere [Sroufe and Watts, 2022], [Pauliuk and Heeren, 2021], [Loftus et al., 2015] and placing photovoltaic panels into an agricultural system provides opportunities for renewable energy generation with minimal compromise to the availability of farmland if agrivoltaics can be well de-

signed to support agricultural production. There are many variables and unknowns in how to approach agrivoltaic design, and this research sets out to build a framework by which agrivoltaic system designed can be optimized.

Agrivoltaics have already demonstrated potential to improve yields for some shade-loving crops [Hudelson and Lieth, 2021], [Touil et al., 2021], [Marrou et al., 2013c], [Marrou et al., 2013b] and reduce water usage [Marrou et al., 2013a], [Hassanpour Adeh et al., 2018], [Elamri et al., 2018], [AL-agele et al., 2021]. These findings are generally related to minimal shading levels reducing evapotranspiration requirements while having minimal impact on the crops. Light uniformity relating to shading has been a general concern, particularly in greenhouse production [Kadowaki et al., 2012], even if shading has no theoretical impact on total yield.

There are ongoing efforts to model and optimize agrivoltaic systems [Goetzberger and Zastrow, 1982], [Zohdi, 2021], [Riaz et al., 2022], [Perna et al., 2019], [Kwon et al., 2020], [Imran et al., 2020] but there remains a need to integrate crop modeling into the simulations, as well as adopt optimization schemes which facilitate wider adoption of agrivoltaics into large scale agricultural production. Water use is of particular relevance to Californian agriculture due to scarcity and state regulations for water use. Because agrivoltaics block incoming solar radiation, evaporation in agrivoltaics is directly decreased as net solar radiation is the most significant driver of evaporation [Monteith, 1965]. Given California is striving to reduce agricultural water use under the recently passed Sustainable Groundwater Management Act [California Department of Water Resources, 2014],

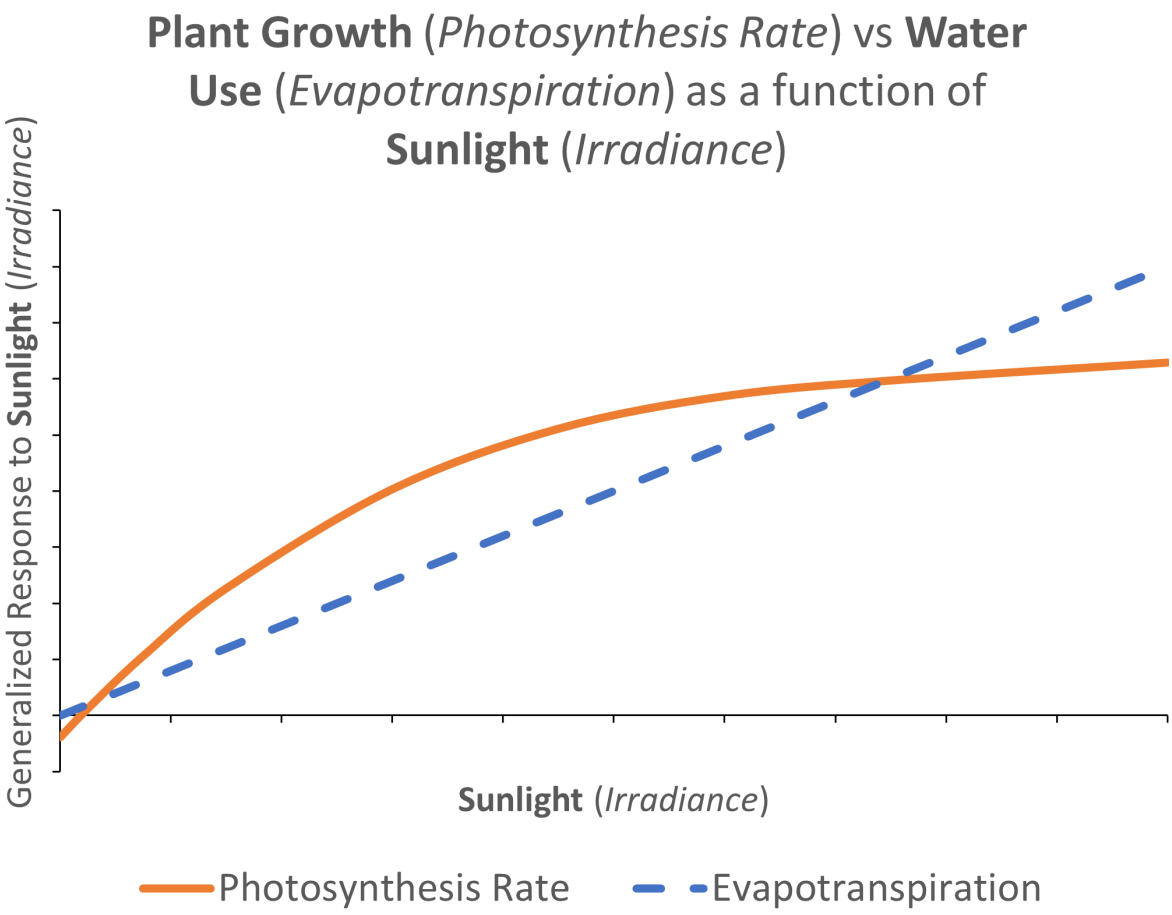


Figure 1: General Relationship Between Light, Photosynthesis and Evapotranspiration

and since much of Californian farmland resides over prime solar energy resources, there could be synergy to help promote high water efficiency farming while also aiding in California's decarbonization goals ([California Energy Commission, 2016]). In addition to improving agricultural water efficiency, agrivoltaics may serve as a pathway to support Californian agricultural resource sustainability through reducing water demand, and subsequent energy associated with pumping and irrigating the reduced water demand, which translates to carbon offsets. However, this potential has not been well explored and needs to be investigated.

Objectives

This research is aimed at contributing to the foundations of agrivoltaic engineering from 3 perspectives: 1) a field experiment is conducted to observe the crop physiology response of processing tomatoes under an agrivoltaics system to aid in understanding the impact that solar panels have on crop growing underneath; 2)) modeling agrivoltaic systems to aid the development of a framework for the engineering of a well-design and optimized agrivoltaics system; and 3) geospatial analysis of potential locations suitable for the installation of agrivoltaic systems in California to improve California's environmental sustainability . To achieve these goals, the following specific objectives were investigated:

Objective 1: Experimental evaluation of processing tomatoes (*Lycopersicon esculentum* Mill.) grown under agrivoltaic through a multi-year field trial by measuring photosynthesis rates to evaluate photosynthetic performance of the crops, non-destructive crop growth metrics such as leaf area index and canopy cover to track

seasonal growth, and measurement of yield and quality at the end of each season.

Objective 2: Genomic optimization of fixed panel agrivoltaic systems using a 3D crop model at an hourly time-step through use of a digital-replica model to simulate the photovoltaic panels and crop in 3D space, definition of optimization parameters useful for agrivoltaic performance modeling, and genomic optimization to determine optimal agrivoltaic designs.

Objective 3: Evaluating agrivoltaic potential in California by conducting a geospatial suitability analysis for agrivoltaic potential of California, then quantification of potential benefits derived from the suitability map.

Hypotheses

Based on review of the literature presented in the introduction, and preliminary analysis (conducted early when scoping the projects), the following hypotheses are proposed:

1. Chapter 1: Experimental Evaluation of Processing Tomatoes (*Lycopersicon esculentum* Mill.) Grown Under Agrivoltaic Production

    (a) Photosynthesis:

        i. H_0: Crops grown under an agrivoltaics system will not have higher photosynthetic carbon assimilation rates than the control

        ii. H_a: Agrivoltaic plants will have higher photosynthetic carbon assimilation rates compared to the control

    (b) Yield:

i. H_0: The control crop will not yield higher than the agrivoltaic crop

ii. H_a: The control crop will have higher yields than the agrivoltaic crop

(c) Quality:

i. H_0: Agrivoltaic crops will not have different quality measurements than the control

ii. H_a: Agrivoltaic plants will have different quality measurements compared to the control

2. Chapter 2: Genomic Optimization of Fixed Panel Agrivoltaic Systems Using an Agrivoltaic - Responsive Crop Model at an Hourly Time step

(a) Digital-replica:

i. H_0: A digital replica of an agrivoltaic crop model coupled with a 3D light model will not be able to simulate crop response to hourly changes in light distribution associated with agrivoltaic light interference

ii. H_a: A digital replica of an agrivoltaic crop model coupled with a 3D light model will be able to simulate crop response to hourly changes in light distribution associated with agrivoltaic light interference

(b) Genomic optimization:

i. H_0: Genomic optimization will not be able to find optimized designs that satisfy given agrivoltaic performance metrics and design constraints

ii. H_a: Genomic optimization will be able to find optimized designs that satisfy given agrivoltaic performance metrics and design constraints

3. Chapter 3: Evaluating Agrivoltaic Potential in California

   (a) Geospatial suitability:

      i. H_0: A geospatial framework for siting agrivoltaics in California will not be able to determine suitable locations for agrivoltaic production in California

      ii. H_a: A geospatial framework for siting agrivoltaics in California will be able to determine suitable locations for agrivoltaic production in California

   (b) Performance:

      i. H_0: A geospatial framework for siting agrivoltaics in California will not be able to calculate performance metrics associated with agrivoltaic adoption across California

      ii. H_a: A geospatial framework for siting agrivoltaics in California will be able to calculate performance metrics associated with agrivoltaic adoption across California

Research Approach

Objective 1: Experimental Evaluation of Processing Tomatoes (*Lycopersicon esculentum* Mill.) Grown Under Agrivoltaic Production

The purpose of this study was to collect relevant crop growth information to inform and evaluate agrivoltaic models and contribute to the growing body of knowledge relating to agrivoltaic crop production. A field experiment was conducted in which processing tomatoes were grown under solar panels near Davis, CA. Crop growth, water usage, and power production were measured in the system. The agrivoltaic system consisted of nine 100 W conventional silicon solar panels installed in a 3 m x 3 m (10 ft x 10 ft) structure, which was placed over two 1.5 m wide processing tomato beds at a height of 1.2 m. The small size and low height of the systems theoretically helped to isolate the microclimate impacts of the agrivoltaics to just that of shading, avoiding complications due to air temperature and wind speed differences that large structures could cause. Given there was no agrivoltaic design software widely available at the time the experimental design was developed, a methodology was proposed to design this agrivoltaic system. The rationale was to maximize shading in the afternoon, reducing heat and water stress and limiting the energy that would be wasted past the light-saturating level for crop photosynthesis while also minimizing shading during other times of the day. To accomplish this, the panels were oriented to be perpendicular to the sun during the day and maximum sunshine hour, which, on average, coincided with the highest water use period over

the last 10 years (June 15th at 13:00 hours). The resulting orientation of the solar panels was 220^{o} azimuth and 19^{o} of tilt determined by using the National Oceanic and Atmospheric Administration solar position calculator [Cornwall et al., 2021] and the perpendicularity of the panels to the sun. The control and agrivoltaic are on the same irrigation line and as such receive the same irrigation, which is determined independent of this experiment through calculations based on California Irrigation Management Information System data points from the nearby Davis station [California Department of Water Resources, 2021]. The experiment was divided into 4 replicates of 2 treatments: agrivoltaic crop had plants located directly below the panels, and the control crop was located away from the panels. In the case of the photosynthesis survey measurements, a third treatment was added by splitting the agrivoltaic treatment plots into sunlit and shaded leave plots. All measurements were taken across treatments in each replicate. Irrigation was the same between the control and agrivoltaics and was determined by the DSSAT crop model, which determined irrigation scheduling. The control plots and agrivoltaic plots were on the same irrigation line and as such, received the same irrigation amount over the growing season, which a crop model determined.

Photosynthesis measurements were critical to understanding the impacts of agrivoltaic on crop growth. Agrivoltaics could theoretically be optimized using the fact that photosynthesis rates are limited by CO_2 levels at higher light levels, meaning additional light at some point will not yield additional benefits for crop yield and may have negative impacts such as extra heat stress or increased water use. The ambient light above the maximum saturation light level could be used for photovoltaic

energy generation. To determine these light levels, photosynthesis experiments were conducted via light response curves, which aide in the characterization of the photosynthesis performance. This data primarily aimed to characterize photosynthesis rates, stomatal conductance, and transpiration of the plants using a LI-COR Li-6800 and to determine if there was a difference between agrivoltaic and conventional crop production. These photosynthesis measurements collected significant information about the gas exchange.

Light response measurements are done using a modified version of the light response curve auto-program which takes the relevant crop physiology measurements at light levels of 0, 50, 150, 300, 600, 900, 1200, 1500 $\frac{\mu mol}{m^2 s}$. The modification was done to add 2000 $\frac{\mu mol}{m^2 s}$ to better represent actual in field light levels which usually reach slightly above 2000 $\frac{\mu mol}{m^2 s}$ in the afternoon.The chamber temperature is set by measuring plant leaf temperature as measured by an infrared net radiometer, and relative humidity and CO2 levels are set to ambient conditions. These measurements provide a curve characterizing light responses in the experimental plants.

Relevant environmental data was collected at the local CIMIS weather station. From the Davis CIMIS weather station, temperature, wind speed, light levels, relative humidity, and vapor pressure deficit were recorded hourly. A thermal unit analysis was conducted by examining the relationship between measured temperatures and plant growth.

The crop growth was measured in a non-destructive way including the leaf area index and canopy cover. Leaf area index will be measured weekly using a Meter Group Ceptometer [Group, 2024] by measuring the light levels across the known

area of the light bar both above and below the crops. This data was also included in the measurement of crop light absorption. Canopy cover was measured by regularly taking photographs on a smartphone at 1.2 m above the ground in each treatment, which were processed using an app called Canopeo [Patrignani and Ochsner, 2015] which automatically calculates the percentage of the image occupied by the crop and produces a canopy cover measurement.

Crop yields were measured using two different methodologies for 2021 and then 2022 at the end of the growing season. During the 2021 harvest, crop yield was determined by hand harvesting four plants in each treatment. The first plant was randomly selected, while the next three plants were plants adjacent to the first one,. These plants were then turned into two groups of two plants, and the entire biomass of the two groups and fruit were weighed. Then the first two plants and fruit were tossed, then the fruits were stripped from the second group, and the fruits were weighed. This methodology was confusing and lead to high variability and was changed. During the 2022 harvest, 10 plants from each control replicate were linearly harvested from a point randomly determined in the field. All 10 plants in each agrivoltaic replicate were harvested. Plants were weighed, then fruits were separated from the plants and weighed. For both years, Samples from each treatment and replicate were sent to a processing facility to measure fruit quality.

Objective 2: Genomic Optimization of Fixed Panel Agrivoltaic Systems Using an Agrivoltaic - Responsive Crop Model at an Hourly Timestep

A digital twin is a virtual simulation that aims to replicate physical systems. In the case of an agrivoltaic system, this involves modeling environmental responses to the photovoltaics and the subsequent impact on crop performance. Key objectives to simulate the parameters, including light, temperature, and wind speed. On the crop side, crop performance is modeled particularly with respect to temperature, photosynthesis, and water responses. The first step for the digital twin operation is to generate a photovoltaic orientation. Following this, a simulation lasting the duration of the crop season is run for light and temperature modification in the system. This is accomplished by building on existing work [Zohdi, 2021]. Once the light/temperature model is run, this information is fed into a crop model. For the purposes of this research, the SIMPLE Crop Model [Zhao et al., 2019a] was used as a basis for modeling as it simulated the core processes of crop growth without being overly complex and was then significantly modified to represent the hourly crop model responses. The model was validated against data collected in Objective 1. These results from the models were analyzed or coupled with other optimization frameworks discussed in the following sections.

Given that agrivoltaics are complex systems, it would be helpful to produce a framework for determining the optimal parameters of agrivoltaic systems to give structure to the agrivoltaic design process. This was done through a liter-

ature review to determine suitable constraints followed by optimization of model parameters. Agrivoltaics should primarily be aimed at improving crop performance; the potential loss of crop due to an agrivoltaic system should not be greater than some threshold [Elborg, 2017], [Deutsches Institut für Normung, 2021], [Massachusetts Department of Energy Resources, 2021], which can be described mathematically in equations 1, 2 and 3. It was also proposed that that the efficiency of the land use in an agrivoltaic system should also be greater than the efficiency of land use in the isolated systems [Valle et al., 2017], [Elamri et al., 2018], [Dupraz et al., 2011], [Mead and Willey, 1980], and self-shading of the panels should be minimized:

$$\frac{Yield_{control} - Yield_{agrivoltaic}}{Yield_{control}} \geq \delta \tag{1}$$

$$\frac{Yield_{agrivoltaic}}{Yield_{control}} + \frac{Power_{agrivoltaic}}{Power_{control}} \geq 1 \tag{2}$$

$$Self\ Shading\ Hours \leq \sigma \tag{3}$$

Considering these constraints, the model will be optimized around solar panel orientation and height to control the amount of light reaching under the panels. Size and reflectivity of the panels can also be considered.

Once the performance of an agrivoltaic system is determined, methods to evaluate this performance in the wider context of engineering is necessary to implement agrivoltaics into society. What the best method for optimization of agrivoltaics is

likely a debate which will rage long beyond a single person's career. There also seems to be a lack of framework for multi-objective agrivoltaics optimization. One potential method is therefore proposed to use a genomic optimization framework built off the Zohdi framework [Zohdi, 2021] be used to study the nature of agrivoltaic optimization with targets based on desirable agrivoltaic performance, and then a cost function was calculated and minimized:

$$min(\Pi) = \omega_1 \frac{Y_c - Y_{av}}{Y_c} + \omega_2 \frac{P_c - P_{av}}{P_c} + \omega_3 \frac{LUE_c - LUE_{av}}{LUE_c} + P_1 + P_2 + P_3 \tag{4}$$

Where $\omega_1, \omega_2, \omega_3$, are relative weights chosen to emphasize crop yield, power production, and/or light use efficiency, respectively (where larger values emphasize greater weight), and P_1, P_2, P_3 are penalty terms if the constraints are not met, as defined below:

$$P_1 = \begin{cases} P_1 & \text{if } \frac{Yield_{av}}{Yield_c} < 1 - \delta \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$P_2 = \begin{cases} P_2 & \text{if } \frac{Yield_{av}}{Yield_c} + \frac{\Sigma Power_{av}}{\Sigma Power_c} < Land\ Equivalent\ Ratio \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$P_3 = \begin{cases} P_3 & \text{if } Self\ Shading\ Hours > \sigma \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Using this framework, agrivoltaic designs can be rapidly compared and optimized to generate designs suited for particular conditions. Outputs from the model can also be used to study agrivoltaic design performance.

Objective 3: Evaluating Agrivoltaic Potential in California

Agrivoltaic systems aim to be implemented as a sustainable solution for agricultural production. This requires work to be done to determine the value of agrivoltaics to society, be it economic or other societal benefit. One such benefit is the role agrivoltaics could play in augmenting farm economics in -stressed regions, particularly regarding water sustainability, as agrivoltaics reduce evaporation through reduced solar radiation and subsequent energy to evaporate water. Saved water correlates to water not pumped or irrigated, which has subsequent water and energy savings that lead to carbon offset potential. To investigate this, a geospatial suitability analysis was conducted to explore the potential areas where agrivoltaics could be deployed in California as well as how these sitings relate to energy production and water savings.

There is currently no established standard method or criteria for determining suitable locations for agrivoltaic ; only a handful of studies attempt this characterization [Kwon et al., 2020], [Ali Khan Niazi and Victoria, 2023], [Coşgun, 2021], and [Majumdar and Pasqualetti, 2018]. This is largely due to the novel nature of agrivoltaic systems and the lack of understanding of what would make a good agrivoltaic site. Through work done in Objective 2, ideas about what properties are desirable in a potential agrivoltaic project can be gleaned, and a framework can be established to evaluate agrivoltaic siting. Given California's conditions and desires to

minimize water stress, produce renewable energy, and maintain agricultural production, a framework is constructed using all three parameters to produce a scale from 0 - 100 ranking particular areas as more or less suitable for agrivoltaic development. The equation is as follows:

$$Agrivoltaic\ Suitability = \frac{Water\ Stress + Photovoltaic\ Potential + Crop\ Type}{3} \tag{8}$$

Once agrivoltaic suitability is determined, the rank is used in calculating different agrivoltaics potential metrics (e.g. water savings, solar energy production, energy savings from water savings etc.) under different adoption rate scenarios. Following this, calculations are made relating to power production, water savings, energy savings, and carbon offsets, all while assuming minimal impacts to agricultural production.

Chapter 1: Experimental Evaluation of Processing Tomatoes (*Lycopersicon esculentum* Mill.) Grown Under Agrivoltaic Production

Abstract

As agrivoltaics gain interest globally, a desire exists to learn how agrivoltaic production may impact crop production. Processing tomatoes were grown for two seasons (2021 and 2022) under agrivoltaic production in Davis, California in a custom agrivoltaic system designed around a principle to excess irradiance and water demand. Non-destructive methods were used to observe growth, specifically leaf area index, canopy cover, and photosynthesis rates. Destructive testing was conducted at the end of each growing season to measure fresh yields and the quality of the harvested fruits. The harvest results showed a significant reduction in yield of 31% (from 85.2 $\pm$ 23.5 $\frac{tons}{hectare}$ to 58.8 $\pm$ 44.5 $\frac{tons}{hectare}$) under agrivoltaic production in 2021 and 70.9% reduction in 2022 (from 178.7 $\pm$ 48.5 $\frac{tons}{hectare}$ to 52.0 $\pm$ 9.7 $\frac{tons}{hectare}$) while the minimal difference in harvested fruit quality were found in both years (with only color difference in 2022 showing statistically significant results). Photosynthetic performance, measured from 0 to 2000 $\frac{\mu mol}{m^2 s}$ of PAR, was found to have been statistically significantly different at the highest and lowest light levels of photosynthetic active radiation (0, 50, 150, 300, 1500 and 2000 $\frac{\mu mol}{m^2 s}$ of PAR), representing typical fully sunlit, and agrivoltaic shaded conditions. Modeling of seasonal crop growth via leaf area index and canopy cover suggests a difference in crop development through the season

as the fitted models suggest different growth curves, but further directed research is needed to investigate the impact of agrivoltaic systems on crop production.

Introduction

Agrivoltaics are a novel form of agricultural production where photovoltaic solar panels are placed over crops to manage the energy environment and optimize the microclimate. Increasing pressures on agricultural production from climate change induced stresses such as reduced water availability and increasing temperatures threatens agricultural sustainability [Raza et al., 2019] while efforts to expand photovoltaic energy production conflicts with agricultural land use [Sargentis et al., 2021]. Agrivoltaic provide an alternative wherein agricultural production and photovoltaic energy generation are synergistically combined to facilitate a multi-benefit land use [Trommsdorff et al., 2021a]. Agrivoltaic experiments across the globe over the last decade have demonstrated potential for agrivoltaic systems to maintain and even improve agricultural lands while also generating green energy, however, the scope of agrivoltaic experiments compared to the diversity of global agricultural is limited [Al Mamun et al., 2022].

This is particularly highlighted by the limited number of agrivoltaics experiments that exist in California despite a seeming confluence of factors favoring agrivoltaic production: large-scale agricultural production, high land value, desire to expand renewable energy production, high heat and water stress, and plentiful sunshine. While it is impossible to quantify the full scope of everything that may or may not exist, two prominent published agrivoltaic projects in California, cover specialized

Figure 2: Agrivoltaic unit used in the experiment in the field with tomatoes growing underneath at the University of California Davis research farm near Davis, California. Photo by [Mistry, 2021]

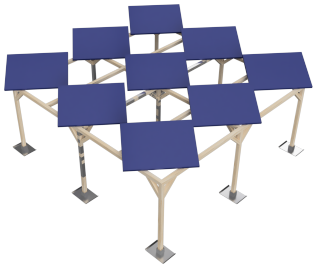
agrivoltaic panel design [Corrado et al., 2013] and agrivoltaic production of potted plants under solar trackers [Hudelson and Lieth, 2021]. In this study, an agrivoltaic experiment growing processing tomatoes was conducted in the west of Davis, CA to understand the dynamics of crop production under an agrivoltaic system (Figure 2). The experiment was conducted with the goal of contributing to the growing body of knowledge of agrivoltaic systems and the implications they have for crop development through measurements of crop yield, quality, and inter-seasonal performance metrics such as leaf area index, photosynthesis rates, and environmental measurements

Materials and Methods

Agrivoltaic Design

A custom agrivoltaic system was designed so that the photovoltaic panels were angled perpendicular to the sun at the time of average peak reference evapotranspiration. Because photosynthesis rates are limited at high sunlight levels [von Caemmerer et al., 2009], light levels can theoretically be limited during times of high light availability. Still, shading should be limited during other times of the day to facilitate photosynthesis. By arranging the panels perpendicular to the sun during the peak reference evapotranspiration hour (almost always in the afternoon) the most direct shading and reduction of light could be provided when light was above the saturation point while during other parts of the day (particularly the morning and late afternoon) the panels are angled such that there is minimal shading to allow for maximum photosynthesis. This approach was chosen because there were no readily available agrivoltaic modeling solutions at the time of this experimental design.

To accomplish this angling, relevant weather data was collected and analyzed to determine the average peak reference evapotranspiration time from the Davis California Irrigation Management Information System station near the field. The analysis used hourly data from the previous decade (2010-2020), sorting it by highest reference evapotranspiration and taking the average date and time of the top 100 values. values. From this it was found that the average peak reference evapotranspiration was June 15th at 13:00 hours. This time was then used in conjunction with the site coordinates into the National Oceanographic and Atmospheric Administration sun



(a) Single Agrivoltaic Unit Design

(b) Agrivoltaic Units in the Field

Figure 3: Agrivoltaic System Design and Installation

angle calculator [Cornwall et al., 2021] to determine the sun angle at this date and time. The panels were then oriented to be perpendicular to this sun angle, which resulted in an azimuth of 220° and 19° altitude.

The agrivoltaic systems consisted of four individual units to increase the statistic power of the experiment as well as to aid in avoiding the potential loss of individual replicates. Each unit was 3 m by 3 m (10 ft by 10 ft) in area with 9 evenly distributed 100 W panels 0.52 m long by 1.07 m wide (1.7 ft long and 3.5 ft wide; ground coverage ratio of 54%). The panels covered two 1.5 m wide tomato and were installed at a height of 1.2 m (4 ft) off the ground to localize the shading on the treatment (Figure 3). The units were randomly placed in the North side of the field. A compass and level were used to verify the correct angles in the field.

Description of Crop and Field

This experiment was conducted in Davis, CA (38.54 N, 121.77 W) between April 28 and September 15 for the 2021 trial and April 21 and September 2 for the 2022 trial using processing tomatoes (*Lycopersicon esculentum* Mill.) [Hartz et al., 2008] with

two different cultivars being grown (Heinz 8504 in 2021, SVTM 9013 in 2022 due to availability) in a Yolo silt loam soil. The experiment was conducted as a subset in the field of a larger experiment by Linker and Kisekka [Linker and Kisekka, 2023]. Both the control and the agrivoltaic treatments were located in the same row on the same subsurface drip irrigation line (30.5 cm (12 in) spacing between drip emitters using a Netafim streamline 3.78 $liters\ per\ hour$ system under 1.5 m (5 ft) tomato beds). Irrigation scheduling was determined based on DSSAT crop model simulations as described in [Linker and Kisekka, 2023]

Data Collection

Crop growth had to be measured non-destructively due to the relatively small number of plants growing under the agrivoltaic system. Two types of measurements were taken to accomplish this: leaf area index (LAI) and canopy cover measurements. At the end of each season, plants were harvested for yield and other plant quality parameters. LAI measurements were taken weekly using a Meter Group LP-80 Ceptometer following the methodology described in the manual [Group, 2024]. These values were then fitted to equation (9) for modeling. Canopy cover measurements were taken weekly using a phone and processed using an app called Canopeo [Patrignani and Ochsner, 2015]. These measurements were also fitted to Equation (10) for modeling. Fitting of both of these models was conducted through minimizing the root mean square error (RMSE) as objective function using the GRG Nonlinear Solver in Excel. To analyze the impact of different models on predicting crop performance, the models were then used to calculate the number of growing

degree days required to reach a specified LAI value.

$$LAI = \frac{A * e^{\frac{-k}{GDD}}}{1 + e^{S - \frac{H}{GDD}}} \tag{9}$$

Where LAI is the leaf area index of the crop (m^2/m^2), A is a fitted shape function parameter ($unitless$), k is is a fitted shape function parameter ($(^\circ C\ day)^{-1}$), S is is a fitted shape function parameter ($unitless$), H is a fitted shape function parameter ($(^\circ C\ day)^{-1}$), and GDD is growing degree days ($^\circ C\ day$) as calculated in equation (12).

Functions for canopy cover are calculated using the same equation structure, but with different parameters calculated through the fitting procedure:

$$CC = \frac{A * e^{\frac{-k}{GDD}}}{1 + e^{S - \frac{H}{GDD}}} \tag{10}$$

Where CC is the canopy coverage of the crop ($unitless$), A is a fitted shape function parameter ($unitless$), k is is a fitted shape function parameter ($(^\circ C\ day)^{-1}$), S is is a fitted shape function parameter ($unitless$), H is a fitted shape function parameter ($(^\circ C\ day)^{-1}$), and GDD is growing degree days ($^\circ C\ day$) as calculated in equation(12).

Individual tomato plants were harvested at the end of the season using two different methods for 2021 and 2022. In 2021, biomass samples were collected by harvesting 4 plants. These plants were then weighed with fruit stripped from stems. Fruit weights and biomass sample weights were taken, and then both were dried and weighed on a dry matter basis. The area harvested was known, so the sample mass

was then upscaled to an area of one hectare. This methodology led to questions of the representativeness of the samples, so in 2022 the methodology was changed whereby plants from a 3.0 m (10 ft) strip were harvested. Samples of fruits were also taken in both years and sent to the California Processing Tomato Advisory Board lab for quality analysis using industry-standard quality metrics (pH, color, and brix content). The ratio of fresh yield to total fresh biomass was also calculated via equation (11).

$$FtBR = \frac{Y_{ff}}{Y_{TFB}} \tag{11}$$

Where: $FtBR$ is the fruit to total biomass ratio ($unitless$), Y_{ff} is the fresh fruit biomass ($tons$ / $hectare$), and Y_{TFB} is the total fresh biomass ($tons$ / $hectare$).

Air temperature was measured using a nearby CIMIS weather station in Davis, CA [California Department of Water Resources, 2021] which was approximately 150 m away from the site both years. Growing degree days were calculated from the meteorological data following [Prentice et al., 1992].

$$GDD = \int (T - T_0)dt \tag{12}$$

Where: GDD is growing degree days ($^\circ C/day$), T is the air temperature ($^\circ C$), T_0 is the base temperature for development ($^\circ C$), and dt is the time step (hr).

Photosynthesis measurements were taken using a Licor-6800 portable photosynthesis system [LI-COR, 2024]. Two types of measurements were conducted: Light response and survey measurements. Light response measurements were conducted on

two treatments, the agrivoltaic and the control treatments. Measurements were done by utilizing a modified system light response curve program to include 2000 $\frac{\mu mol}{m^2 s}$, a light value more representative of the maximum PAR experienced in the field and by setting the leaf chamber temperature to a representative temperature as measured using an infrared net radiometer on a leaf. Survey measurements were taken mid-morning, afternoon, late-afternoon, and at sundown (approximately 09:00, 12:00, 16:00 and 19:30, respectively) by measuring light values using the onboard PAR sensor, and leaf temperature and setting the chamber settings to match. The CO_2 setting was set to 400 ppm to reflect ambient conditions, and relative humidity was set to a setting which didn't cause the system to condense and have problems. The light response curve was then modeled using equation (13).

$$A_{net} = \frac{A_{max} k I}{k I + A_{max}} - R_d \tag{13}$$

Where A_{net} is the net photosynthesis rate ($\frac{\mu mol}{m^2 s}$), A_{max} is the maximum photosynthesis rate ($\frac{\mu mol}{m^2 s}$), k is a coefficient ($\frac{m^2 s}{\mu mol}$), I is the incoming PAR irradiance ($\frac{\mu mol}{m^2 s}$), and R_d is the dark respiration rate ($\frac{\mu mol}{m^2 s}$).

Results

Leaf Area Index and Canopy Cover

LAI and canopy cover measurements were modeled using equations 9 and 10 with respect to growing degree days (as seen in Figures 4 and 5). The results of modeling are detailed in table 1 and table 2. The LAI equation was then used to predict the

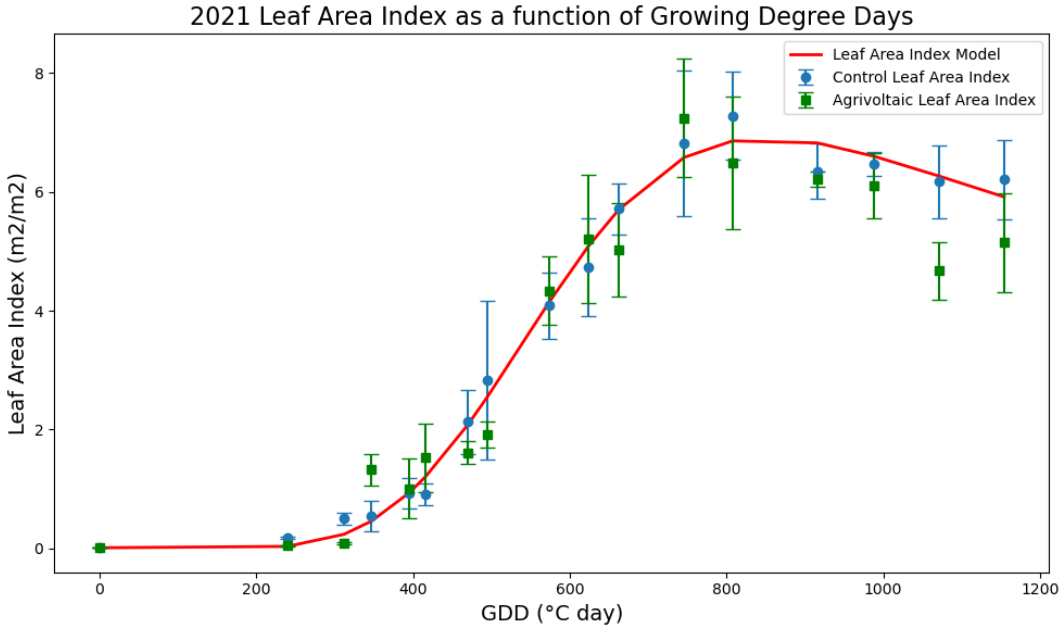
GDD required to reach LAI of 1 and 4 (chosen because an LAI of 1 represents a density of a single layer of leaves covering the entire ground area, and 4 because this represents a dense canopy late season), and the 2021 results showed the time required for the control to reach an LAI of 1 and 4 for the control was 413.4 ± 24.4 GDD and 562.7 ± 46.2 GDD respectively. The agrivoltaic results were 408.5 ± 28.5 GDD and 569.9 ± 17.0 GDD. These results were not statistically significant at $\alpha = 0.05$. The 2022 results showed the time required for the control to reach an LAI of 1 and 4 for the control was 292.8 ± 13.8 GDD and 579.4 ± 66.4 GDD respectively. The agrivoltaic results were 365.9 ± 88.8 GDD and 780.6 ± 110.0 GDD. The LAI = 1 result was not significant at $\alpha = 0.05$, however, the LAI = 4 result was significant at $\alpha = 0.05$, (p = 0.013).

Table 1: LAI Curve Fitting

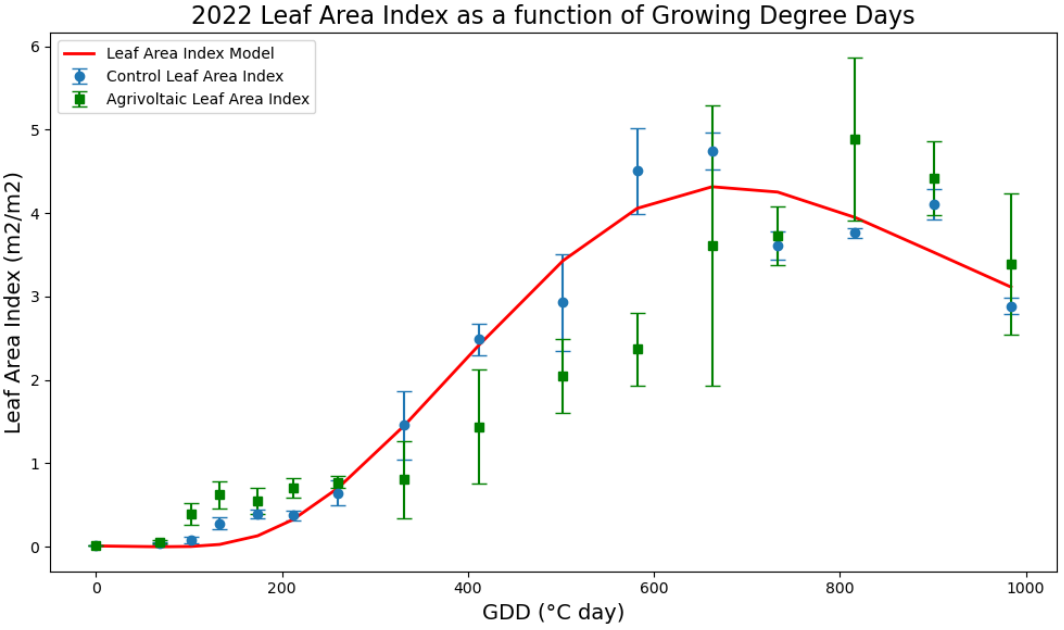
Parameter	Units	Control 2022	Agrivoltaic 2022	Control 2021	Agrivoltaic 2021
A	*unitless*	21.3	55.6	1006	303
k	$(^{\circ}C\ day)^{-1}$	888	1228	2513	2231
S	*unitless*	4.49	15.6	5.51	5.99
H	$(^{\circ}C\ day)^{-1}$	3898	20329	4002	4844
$RMSE$	*none*	0.35	0.53	0.48	0.67
R^2	*none*	0.96	0.90	0.97	0.93

Table 2: Canopy Cover Curve Fitting

Parameter	Units	Control 2022	Agrivoltaic 2022	Control 2021	Agrivoltaic 2021
A	*unitless*	148	148	319	418
k	$(^{\circ}C\ day)^{-1}$	336	359	832	910
S	*unitless*	9.82	6.75	4.20	3.30
H	$(^{\circ}C\ day)^{-1}$	9957	6358	4627	3364
$RMSE$	*none*	3.30	4.52	3.98	3.62
R^2	*none*	0.99	0.98	0.98	0.98

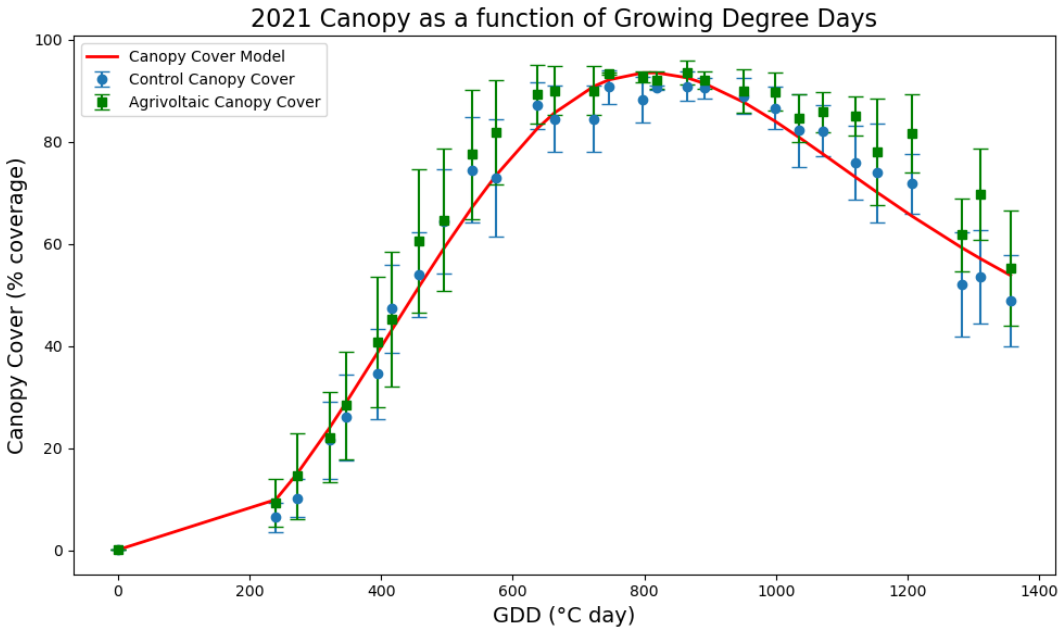


(a) 2021 Leaf Area Index

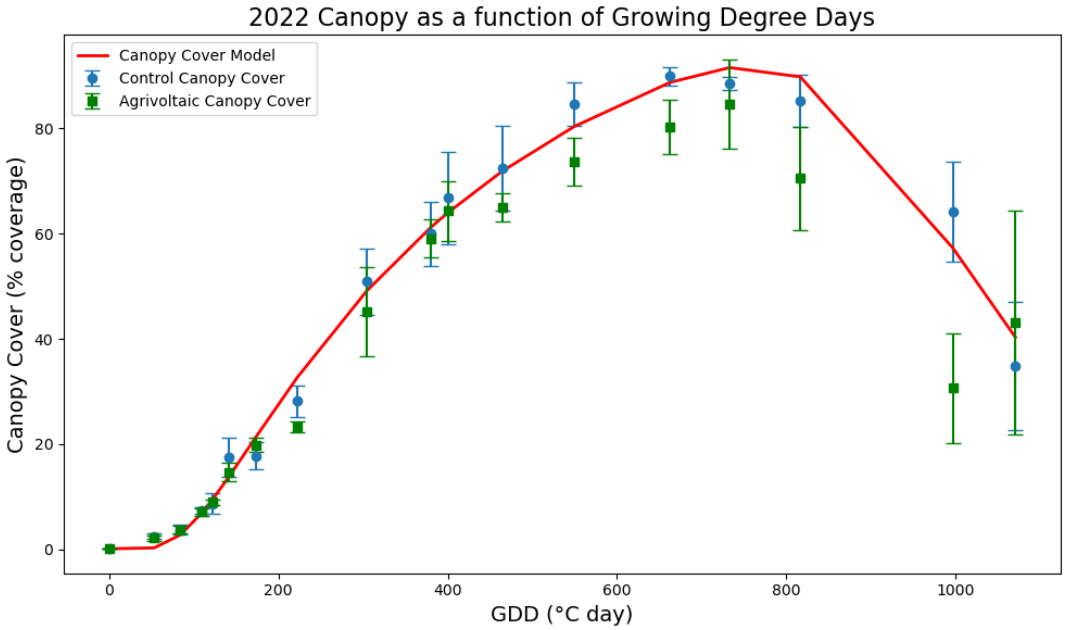


(b) 2022 Leaf Area Index

Figure 4: Leaf Area Index as a Function of Growing Degree Days



(a) 2021 Canopy Cover



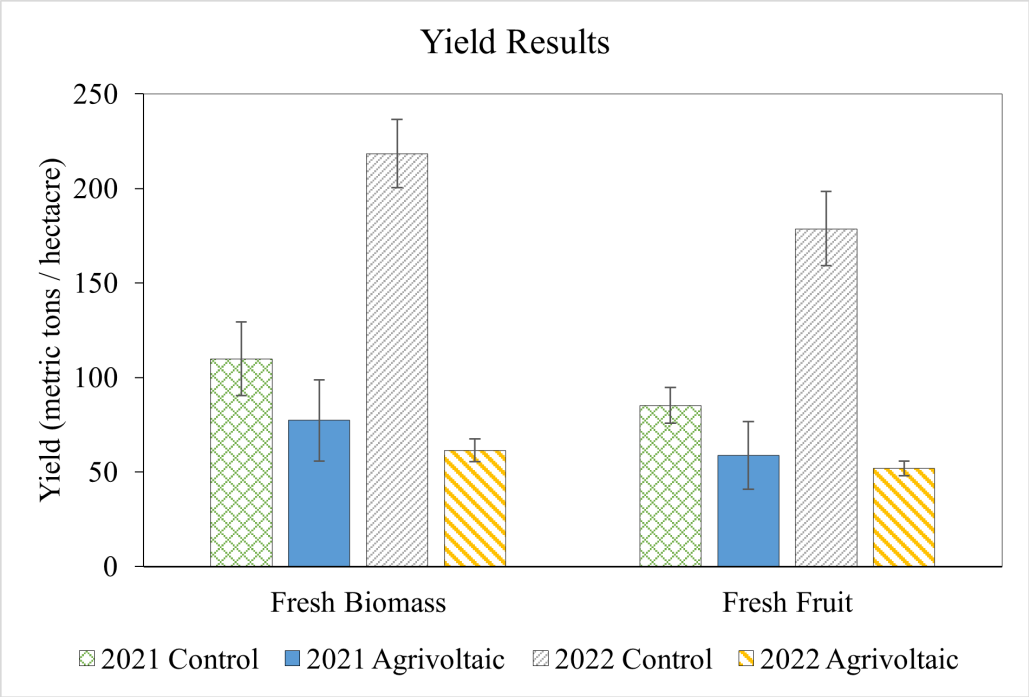
(b) 2022 Canopy Cover

Figure 5: Canopy Cover as a Function of Growing Degree Days

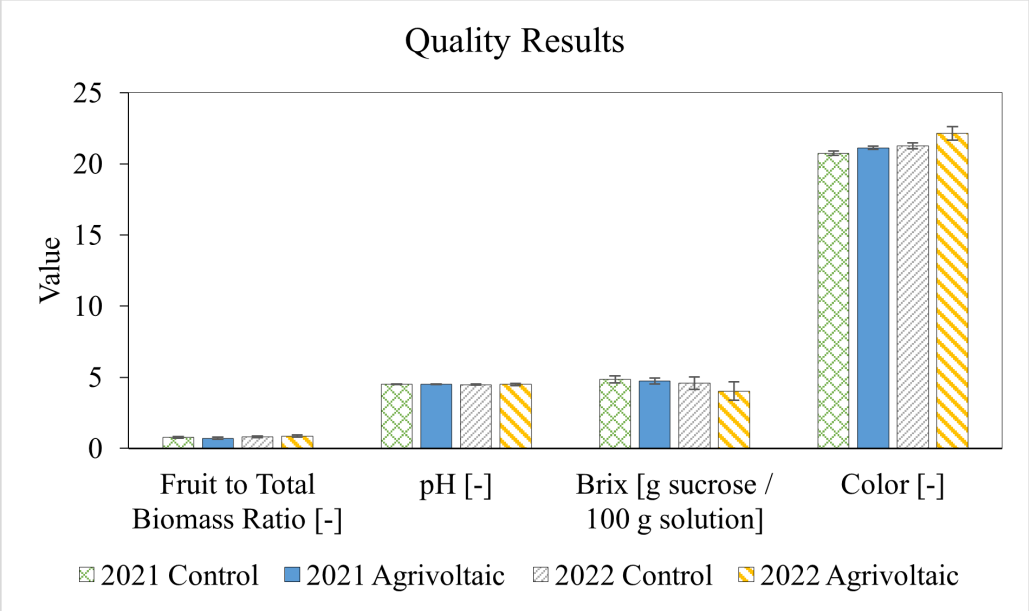
Yield

The yield results found the control to have a fresh fruit biomass of 85.2 ± 23.5 $\frac{tons}{hectare}$ compared to 58.8 ± 44.5 $\frac{tons}{hectare}$ for the agrivoltaic treatment in 2021 and 2022 results showed 179 ± 48.5 $\frac{tons}{hectare}$ and 52.0 ± 9.68 $\frac{tons}{hectare}$ for the control and agrivoltaic treatments respectively. These results were analyzed using a t-test assuming unequal variance in Excel and the 2021 yield difference was found to be significant at $\alpha = 0.01$, and the 2022 yield difference was found to be significant at $\alpha = 0.015$. The 2021 yields for both the control and agrivoltaic are a lower than typical industrial harvest yields, which average around 100-125 $\frac{tons}{hectare}$ [United States Department of Agriculture, 2023], while the 2022 control yield is higher and the 2022 agrivoltaic yield is lower. Total biomass was also measured and the control and agrivoltaic total biomass difference results for both 2021 and 2022 were found to be significant at $\alpha = 0.01$ with values of 110 ± 48.1 $\frac{tons}{hectare}$ compared to 77.3 ± 53.0 $\frac{tons}{hectare}$ in 2021 and 218 ± 44.7 $\frac{tons}{hectare}$ compared to 61.4 ± 14.7 $\frac{tons}{hectare}$ in 2022 (both for the control and agrivoltaic treatments respectively) (fig 6 (a)). Fruit to biomass ratio, pH, color and brix content (fig 6 (b)) were also significance tested to $\alpha = 0.05$ and the only value found to be significant at $\alpha = 0.05$ was color ($unitless$) in 2021 where the difference between the control and agrivoltaic was 20.8 ± 0.25 compared to 21.1 ± 0.22 where the lower value is considered better. Significance testing was not conducted below $\alpha = 0.05$. The significance testing is described in . Summaries of the measured harvest parameters is in table 3.

Energy absorbed by the canopy was also simulated for 3 control and 3 agrivoltaic treatments in 2021 using methodology detailed in Chapter 2 of this dissertation.



(a) Yield Biomass Results



(b) Yield Quality Results

Figure 6: 2021 and 2022 yield results detailing total biomass, fruit biomass, ratio of fruit to total biomass, pH, Brix and color metrics

Table 3: Outcomes of crop yield and quality for control and agrivoltaic treatments

Year	Treatment	Fresh Biomass $\frac{tons}{hectare}$	Fresh Fruit $\frac{tons}{hectare}$	FtBR $\frac{Fresh\ Fruit}{Fresh\ Biomass}$	pH [-]	Color [-]	Brix $\frac{g\ sucrose}{100\ g\ solution}$
2021	control	110 ± 48.1 ***	85.2 ± 23.5 ***	0.77 ± 0.07	4.505 ± 0.03	20.8 ± 0.25 *	4.85 ± 0.17
2021	agrivoltaic	77.3 ± 53.0	58.8 ± 44.5	0.71 ± 0.09	4.52 ± 0.02	21.1 ± 0.22	4.75 ± 0.13
2022	control	219 ± 44.7 ***	179 ± 48.5 **	0.81 ± 0.07	4.48 ± 0.05	21.3 ± 0.43	4.58 ± 0.22
2022	agrivoltaic	61.4 ± 14.7	52.0 ± 9.68	0.86 ± 0.07	4.49 ± 0.07	22.1 ± 0.65	4.03 ± 0.48

*** is significant at $\alpha = 0.01$, ** is significant at $\alpha = 0.015$, * is significant at $\alpha = 0.05$

Given this model has not been validated for accuracy, but is precise enough between simulations for broad exploratory analysis, statistical significance testing is not conducted, but the general trends are observed where the control light use efficiency (LUE) for total biomass in the control is 0.0021 ± 0.0002 $\frac{tons\ /\ hectare}{MJ\ /\ m^2}$ compared the agrivoltaic values of 0.0024 ± 0.0003 $\frac{tons\ /\ hectare}{MJ\ /\ m^2}$, and LUE for fruit biomass in the control is 0.0019 ± 0.0006 $\frac{tons\ /\ hectare}{MJ\ /\ m^2}$ and 0.0018 ± 0.0002 $\frac{tons\ /\ hectare}{MJ\ /\ m^2}$ in the agrivoltaic. Data is shown in table 4. The modeling also suggests the seasonal average absorbed radiation was approximately 43,000 $\frac{MJ}{m^2}$ in the control compared to 22,000 $\frac{MJ}{m^2}$ in the agrivoltaic treatment, which is roughly a 52% decrease in absorbed radiation.

Table 4: Light use efficiency and calculation inputs

Metric	Units	C1	C3	C4	AV1	AV2	AV3
Total Biomass	$\frac{tons}{hectare}$	80.4	100	91.0	44.7	58.4	53.8
Fruit Yield	$\frac{tons}{hectare}$	68.9	118	58.3	39.8	43.5	37.5
Modeled Radiation	$\frac{MJ}{m^2}$	43228	42144	42536	22336	20938	23923
LUE (Total Biomass)	$\frac{tons\ /\ hectare}{MJ\ /\ m^2}$	0.0019	0.0024	0.0021	0.002	0.0028	0.0022
LUE (Fruit Yield)	$\frac{tons\ /\ hectare}{MJ\ /\ m^2}$	0.0016	0.0028	0.0018	0.0018	0.0021	0.0016

Photosynthesis

Photo synthetic light response measurements all demonstrated higher photosynthetic performance for the agrivoltaic crop than the control. However, the differences are only found to be significant at the highest and lowest light levels (PAR = 2000, 1500, 300, 150, 50, 0 ($\frac{\mu mol}{m^2 s}$)). These results are detailed in table 5. It should be noted, these results do not include significance testing below $\alpha = 0.05$, and the significance of these results are highly dependent on the data cleaning methodology. To clean outliers from this data set, unreasonably high and low values (indicating errors in the measurement) were removed by hand (for example, at PAR = 2000 $\frac{\mu mol}{m^2 s}$, values of -7 $\frac{\mu mol}{m^2 s}$ and 60 $\frac{\mu mol}{m^2 s}$ were present when the expected range was around 25-40 $\frac{\mu mol}{m^2 s}$ and thus unreasonable for the light level, so then removed). Values that may have been on the edges of reasonableness were preserved.

Table 5: Photosynthetic light response measurment summary

PAR Level ($\frac{\mu mol}{m^2 s}$)	Control A_{net} ($\frac{\mu mol}{m^2 s}$)	Agrivoltaic A_{net} ($\frac{\mu mol}{m^2 s}$)	Significance Level
2000	26.2 ± 4.8	29.4 ± 4.5	**
1500	26.4 ± 3.2	28.5 ± 3.7	*
1200	24.3 ± 3.3	26.2 ± 5.3	
900	20.6 ± 3.4	23.3 ± 7.0	
600	16.3 ± 2.9	18.2 ± 5.9	
300	8.1 ± 2.2	9.4 ± 1.8	***
150	3.2 ± 1.2	4.0 ± 1.2	**
50	-1.4 ± 1.2	-0.6 ± 1.1	***
0	-3.8 ± 1.3	-3.1 ± 1.1	**

*** is significant at $\alpha = 0.015$, ** is significant at $\alpha = 0.025$, * significant at $\alpha = 0.05$

Fitting was also performed and summaries of the fitting are summarized in table 6 and visualized in figure 7.

Table 6: Photosynthetic light response curve fitting parameters

Parameter	Units	Control	Agrivoltaic
A_{max}	$\frac{\mu mol}{m^2 s}$	39.4	38.9
k	$\frac{m^2 s}{\mu mol}$	0.07	0.07
R_d	$\frac{\mu mol}{m^2 s}$	4.19	4.20
$RMSE$	$none$	5.79	1.28
R^2	$none$	0.77	0.71

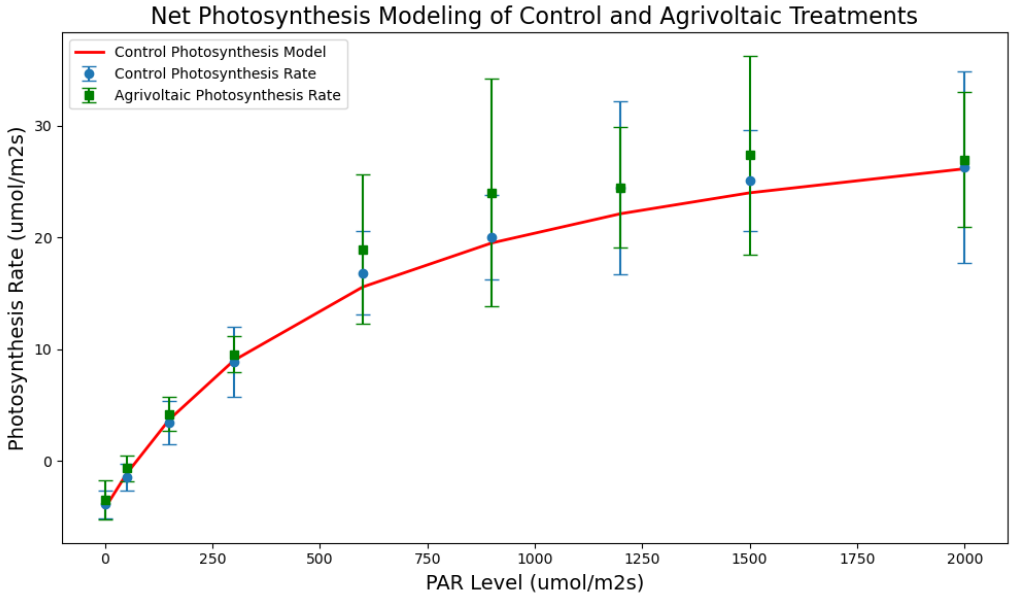


Figure 7: Photosynthetic light response of control and agrivoltaic treatments

Discussion

The significant reduction in yield was unsurprising in this context given the large reduction of solar radiation (roughly a 52% reduction in modeled radiation between the control and agrivoltaic treatments) and the knowledge that processing tomatoes do not respond well to aggressive shading [Abdel-Mawgoud et al., 1995]. Interestingly, changes in fruit quality the agrivoltaic crop production treatment was minimal and were only significantly worse in the 2021 fruit color measurements. However, most fruit quality metrics were lower in the agrivoltaics treatment, albeit not statistically significantly lower than the control. Two other agrivoltaic studies conducted on tomatoes conducted in Italy [Scarano et al., 2024] and [Mohammedi et al., 2023] using the *Solanum lycopersicum* L. species also showed reductions in yield with impacts to quality that could be considered acceptable. Tomatoes are widely considered sun-loving crops, so these results are not surprising. However, tomatoes also commonly grown in greenhouses where solar radiation is decreased due to greenhouse materials and structures. This could suggest there is still some acceptable reduction in light levels that benefit tomato production. Reduced yields in this experiment, then, should not deter further experiments for agrivoltaic tomato systems. Further research and engineering of agrivoltaic systems may eventually yield general design principles for good agrivoltaic tomato performance.

The absorbed radiation modeling suggests the design used in this study prevents about 52% of the incoming solar radiation from being absorbed. Looking at the photosynthesis results from Figure 7 and Table 6, it appears the light saturation point for photosynthetic activity is roughly around 1500 $\frac{\mu mol}{m^2 s}$ of PAR, reaching approximate

75% compared to the typical peak activity of 2000 $\frac{\mu mol}{m^2 s}$ of PAR seen during the day. This by itself may be a misleading metric because that is a per leaf level performance and crop canopies are made up of many leaves absorbing different levels of radiation. However, a review of different agrivoltaic crops [Magarelli et al., 2024] has suggested a 30% shading threshold. This generally supports this notion that potentially limiting tomato shading to no more than 25% as shown in the photosynthesis measurements is a good idea for future experiments. And the 52% reduction in absorbed radiation was likely too aggressive for this design.

The photosynthetic light response model suggests that the saturated photosynthesis rate is slightly higher in the control than in the agrivoltaic treatment. It is important to note this is an empirical equation that would require location-specific parameterization. To make more conclusive determinations, the statistical analysis of photosynthetic performance at different light levels suggests that the agrivoltaic plants performed better at the highest and the lowest light levels, as expected. The agrivoltaic plants were mainly seeing light at extremely high and extremely low light values (either directly sunlit or shaded) and as such there is a change in performance at these points, which could also be influenced by the fact that the physiology of the plant needed to adjust to two extremes. However, what is unclear from these experiments is whether this is due to localized long-term improvements in the microclimate or some physiological change in the plant's performance. It would be advisable to perform a more comprehensive $A - C_i$ curve, and/or conduct thorough experiments to determine FvCB coefficients [Farquhar et al., 1980] and temperature response experiments to understand the plant photosynthetic performance more thoroughly.

This work was not able to conclusively determine if there is a significant difference in the phenological development of the crop within agrivoltaic systems. It appears the performance of the LAI and canopy cover can be modeled differently as demonstrated by the different fitted equations. The actual impacts of the agrivoltaic treatment on GDD calculations guided by phenological targets of an LAI of 1 or 4 $\frac{m^2\ leaf\ area}{m^2\ ground\ area}$, it seems there was only a minimal impact with an average different of 5 GDDs in 2021, but more impactful in 2022 with an average difference of 80 GDDs between the agrivoltaic and control treatment. There are different ways to interpret this result. If a control and agrivoltaic crop are growing simultaneously, the agrivoltaic crop may need to be planted either earlier or harvested later to have comparable results. Another interpretation is, the crop is developing differently, but again, this was not able to be conclusively determined with this experiment, and further research is needed. However, even if the specific results were minimal, even small changes when modeling crop development (which is already highly at risk of error) could have huge implications for modeling agrivoltaic crop performance. A preference would be to use agrivoltaic specific models to model the crop performance of specific crops in agrivoltaic settings. On the other hand, when attempting to undertake a general project such as designing an agrivoltaic system (which many people currently are) for research purposes, where small errors may not have the same compounding impacts as models developed for commercial agricultural practice, it is likely safe to use preexisting crop models to evaluate theoretical crop performance.

Conclusion

This study gleans insights into how processing tomato crops may perform in an agrivoltaic setting compared to a reference control crop. A field experiment was conducted for two years near Davis, CA, using a 3 m x 3 m solar installation over a tomato crop to study light levels, photosynthetic activity, water use, and plant response. Yields were dramatically reduced under the agrivoltaic treatment, as would be intuitively expected with a large reduction in sunlight (52%), but new insights were gained as to the photosynthetic performance of processing tomatoes grown under the agrivoltaic treatment. The experiments suggest the crop adapts to high and low light situations, and modeling crop growth through the season also hints at some potential differences in growth. However, further research should be conducted to better understand the impact of sunlight reduction on tomato growth. Impact of agrivoltaics on fruit quality was minimal suggesting that further refinements in agrivoltaic design could yield functionally negligible impacts, or possibly even positive outcomes of growing processing tomato in agrivoltaic systems.

Genomic Optimization of Fixed Panel Agrivoltaic Systems Using an Agrivoltaic - Responsive Crop Model at an Hourly Timestep

Abstract

As the field of agrivoltaic production advances, there is a need to develop optimization frameworks that allow for the simulation and evaluation of alternative agrivoltaic designs without the need for experimentation. Given the complexities of how different agrivoltaic design parameters will impact the system performance, no single framework or simulation software currently exists that can easily account for all potential variation, which also introduces challenges for system optimization. There does exist, however, models which can evaluate specific performance metrics (such as crop growth, energy production, etc.), and genomic optimization models that can be used to simulate the individual designs and evaluate the performance of many designs to find an optimal solution. Previous attempts have been made to evaluate agrivoltaic performance using models, but many lack the ability to simulate the complexities associated with agrivoltaic production, breaking the spatiotemporal assumptions on which conventional crop models are built. The framework developed in this study attempts to build on previous works by utilizing tested optimization patterns while integrating an agrivoltaic responsive crop model. Specifically, this model integrates sub-leaf level ray tracing to evaluate non-linear photosynthetic performance at an hourly time step to capture the complex dynamics associated with

the influence of agrivoltaic installations on crop performance. The developed model is validated against data collected from a field experiment conducted in Davis, California, where processing tomatoes were grown under a 3 m x 3 m (1.2 m installation height) agrivoltaic system for two years. An optimized design was found with the developed framework consisting of 4 evenly spaced panels (33 cm wide by 115 cm long, at a height of 84 cm)facing southwest (specifically an azimuth of 230° with an elevation of 16°). The optimal design resulted in an agrivoltaic crop yield that was only reduced by 6% compared to the control and only a 20% reduction in the theoretical maximum photovoltaic energy potential throughout the growing season. Other outputs, such as seasonal growth, average crop performance, and power production, were also produced and computed to understand the impact of the design on crop performance.

Introduction

Agrivoltaic systems are an emergent form of agricultural production where photovoltaic solar panels are placed over agricultural crops to increase the efficiency of land use by producing solar energy and crops (Figure 8). Experiments around the world have demonstrated the capacity for agrivoltaics to be a successful form of energy and food production form of production under correct circumstances [Al Mamun et al., 2022], [Hudelson and Lieth, 2021], [Touil et al., 2021], [Marrou et al., 2013c], [Marrou et al., 2013b] and there is a need to develop tools to optimize agrivoltaic designs. However, designing agrivoltaics is complicated as there are no standard methods for evaluating how changes in the system design will

(a) Agrivoltaic design used in [Marrou et al., 2013b]

(b) Agrivoltaic design used in [Trommsdorff et al., 2021b]

Figure 8: Agrivoltaic Designs

impact crop production. Evaluation of a "good" design is an uncertain question as considerations must be given to the performance of both the photovoltaic and the agricultural components, each of which is complicated. Suppose a solar panel is tilted a few degrees in one direction or another, or spaced an extra few centimeters apart. What will the impact be on the system performance in terms of energy, food, and water? While modeling tools exist to evaluate the performance of individual components and even functions deterministically exist to evaluate such components (i.e., water use, crop growth, photovoltaic energy, etc.), there is no commonly accepted "universal" model to evaluate the impacts of changing individual design parameters on agrivoltaic performance. To overcome this, various methods are proposed here for evaluating agrivoltaic performance.

One such approach is utilizing genetic algorithms to deterministically evaluate random agrivoltaic designs, rank and sort them, then continue evaluating designs until an optimal design as defined by some metrics has been found. Two such approaches before have also both demonstrated that vertical photovoltaic designs are optimal solutions [Mengi et al., 2023], [Campana et al., 2021]. However, both these models utilized crop growth models at a daily time scale, which would give pref-

erence to maximizing solar energy in the afternoonwhen there is maximum solar energy available. These findings are interesting given CO2 limited photosynthetic light response curves and the energy balance for water use would suggest this is not an optimal time to maximize light. Additionally, previous work did not incorporate the complications associated with the disruptions of light into the agricultural system, both the crop and the bare soil. Shading causes non-uniform light distribution, which breaks the assumptions of many crop models.

With the need for more complex crop modeling, other considerations to agrivoltaic performance should be considered, such as the desire for agrivoltaics to minimize impact on crop yields [Letternmeier et al., 2021], the desire for efficient land use [Valle et al., 2017], and the desire to minimize the impact of self shading on the photovoltaic panels [Galtieri and Krein, 2015].

This study aims to build on previous works and contribute to the agrivotlaic engineering body of knowledge through the development of a 3D crop and photovoltaic simulation framework capable of simulating the complex considerations of agrivoltaic performance. The simulation and optimization framework developed in this study is then evaluated and optimized through the use of a multi-objective genetic algorithm.

Methods

To design and optimize an agrivoltaic with a crop-responsive model, several considerations were made: sub-daily time steps, modeling of light levels and light distribution underneath an agrivoltaic system, photovoltaic performance, non-linear photosynthetic light response of the plant growth, temperature response and canopy growth

of the plant in an agrivoltaic system, accumulation of biomass in the plant system, evaporation and transpiration components of the system evaluated spatio-temporally, and development of metrics to evaluate the performance of the system.

To manage the light simulation and crop growth models, Helios Simulation Software [Bailey, 2019] was used to simulate light in an agrivoltaic system and generate dynamic crop canopies and photovoltaic panels to understand the spatial influence of shading on the agrivoltaic design as well as capacities to subdivide individual elements in a way which allows for detailed analysis. This analysis was done at an hourly time step but could be reduced or increased depending on need.

A genetic algorithm approach was implemented using previous work from [Mengi et al., 2023] and coded in Python. Additionally, the previous implementation required serial simulation of the agrivoltaic designs, but the proposed implementation in this study allows for a combination of parallel and serial processing (meaning both that multiple simulations are ran at the same time, as well as sequentially) to increase the simulation speed while facilitating control over computer resource usage.

The core concept of this methodology is to: 1) define crop, site and season information and limits for the agrivoltaic design, 2) evaluate control crop and agrivoltaic performance, 3) generate random agrivoltaic designs, 4) simulate agrivoltaic performance for the defined season, 5) evaluate its performance using various performance metrics, 6) ranks the performance of all the simulations, 7) cut the least optimal pairs, generate some new designs as variations of the best designs, then generate new random designs, 8) continue the process until an optimal solution is found.

To accomplish running this simulation, the modeling framework presented be-

low was implemented in parts. The genetic algorithm was coded in Python, and the crop simulation code was coded in C++ using the Helios Simulation Software [Bailey, 2019], with Python code controlling the execution of the simulations. Dynamic multi-threading was also enabled in the software to facilitate running user-defined number of parallel simulations to speed up performance while allowing for control over how much computer resources are utilized.

With respect to Helios, several plugins are used: canopy generator, radiation model, solar position, and visualizer. The radiation model and solar position models were used for ray tracing light (and details of algorithms involved can be found in [Bailey, 2018]),while the visualizer was used for creating visualizations from the results. The canopy generator was used to generate the plant model. The spherical canopy generator was used for 20 plants, 10 plants placed 30 cm (1 ft) apart in 2 rows, which are 1.5 m (5 ft) spaced apart, which is a typical processing tomato planting pattern. This planting pattern can be easily changed for other crops. Periodic boundaries are enforced in the simulation settings such that the crop area is simulated as infinite. Helios primitives (a geometry used for calculating fluxes, an explanation of which can be found in the Helios documentation [Bailey, 2022]) were used to calculate fluxes at the sub-leaf segments, sub-panel segments, and sub-ground segments. Code for the implementation can be found in the Chapter 2 Appendix.

Crop Modeling

Simulation of the crop within this framework consisted of two components: plant growth and photosynthetic performance, while other metrics were calculated in ad-

dition to these two primary components to allow for other relevant performance metric evaluations. Conventional plant growth models are generally simulated at a daily time step, with some models like DSSAT [Hoogenboom et al., 2019] allowing for pseudo-hourly or sub-hourly modeling. This, however, does not facilitate modeling the photosynthetic response to rapidly changing light conditions, which is expected to be a function of photovoltaic shading in the crop system. Additionally, given the large variation in light distribution and its expected impact on the individual plant, conventional models of canopy light attenuation (such as Beer-Lambert) will not produce suitable results [Ponce de León and Bailey, 2019], so the light model should evaluate light on individual leaves. And this implementation of light modeling using Helios allows for the evaluation of sub-leaf segments. Lastly, many crop models evaluate crop growth as a function of air temperature or growing degree days (GDD), which may be fine for macro crop growth, but individual leaves will have large fluctuations in temperature related to the changing light conditions which impact the energy balance and stomatal conductance of individual leaves.

Modeling these light dynamics through the canopy is critically important as photosynthesis rates are limited at high light levels [von Caemmerer et al., 2009], and it is desirable to model this dynamic to capture the non-linear response of photosynthetic performance for light in order to capture an often critically considered component in agrivoltaic crop performance.

Additionally, crop metrics calculated as raw yields can be considered with respect to other parameters, such as light use efficiency and land use efficiency , to explore the implications of non-linear crop responses and other secondary performance char-

acteristics.

Plant Growth

The plant growth model used in this agrivoltaic modeling framework is based on the SIMPLE crop model [Zhao et al., 2019b] because it offers a low complexity framework but diverges significantly as the model is run at a daily timescale and the developed framework is run at an hourly time step. A previous study using this crop model to develop an agrivoltaic optimization [Mengi et al., 2023] highlighted the need to use an hourly crop model to capture non-linear crop dynamics.

This modeling framework is conceptually straight forward, but needs to be bulit by expanding concepts into solvable equations. The first step is to calculate yield as a function of total biomass over the season multiplied by a harvest index for the plant, as defined below:

$$Y = \Sigma BM \times RFtB \tag{14}$$

Where: Y is the crop yield ($tons\ /\ hectare$), ΣBM is the total biomass accumulated over the season ($tons\ /\ hectare$), as defined in equation (15), and $RFtB$ is the ratio of fruit to total biomass ($unitless$ or $\frac{tons\ /\ hectare\ fruit\ biomass}{tons\ /\ hectare\ total\ biomass}$).

The sum of biomass for the whole season can then be described as the evaluation of biomass accumulation at every time step, as defined below:

$$\Sigma BM\ = \int_{t\ Start}^{t\ Harvest} BM_t\ dt \tag{15}$$

Where: $t\ Start$ and $t\ Harvest$ are the starting and harvest time steps ($hours$ or $days$ depending on specific implementation), BM_t is biomass generated at time t as defined in eq. (16), and dt is the time step ($hours$ or $days$ depending on specific implementation).

Biomass at any time step can then be calculated by evaluating the photosynthetic flux as a function of incoming light times a temperature response function over the area of a leaf sub-division times the time step, evaluated for every sub leaf segment for every leaf in the system, all multiplied by a resource use efficiency, as defined below:

$$BM_t = (\left\{ \frac{A(Q_{PAR,t})}{A_{sL}} dt \times sL \mid sL \in L \right\} \text{ for each } L \in nL) \times RUE \tag{16}$$

Where: BM_t is the accumulated biomass ($tons\ /\ hectare$) at time t, $A(Q_{t,PAR})$ is the net photosynthetic light response rate at time t ($\mu mol\ CO_2\ /\ m^2 s$) as a function of of the photosynthetic active radiation (PAR) ($\mu mol\ photons/\ m^2 s$) flux (Q) as defined in eq. (17), $A_s L$ is the subdivided leaf area (m^2), sL is the leaf subdivision ($unitless$), L is the individual leaf ($unitless$), and nL is the total number of leaves ($unitless$) as define in eq. (18), and RUE is the resource use efficiency ($\frac{tons\ /\ hectare}{\mu mol\ CO_2}$).

The photosynthetic flux can be defined as a function of incoming photosynthetic active radiation in a non-linear function, as defined below:

$$A(Q_{PAR}) = \frac{A_{max} k Q_{PAR}}{k Q_{PAR} + A_{max}} - R_d \tag{17}$$

Where: $A(Q_{PAR})$ is the net photosynthetic light response rate ($\mu mol\ CO_2\ /\ m^2 s$) as

a function of the photosynthetic active radiation (PAR) ($\mu mol\ CO_2\ /\ m^2 s$) flux (Q), A_{max} is the maximum photosynthetic rate ($\mu mol\ CO_2\ /\ m^2 s$), k ($m^2 s\ /\mu mol\ photons$) is a shape function, Q_{PAR} ($\mu mol\ photons/\ m^2 s$) is the incoming PAR, and R_d is the dark respiration rate ($\mu mol\ CO_2\ /\ m^2 s$). The shape of this function is described in fig. 9.

Given this model evaluates photosynthesis on individual leaves, it is important to calculate the actual number of leaves as a function of the leaf area index, which is a measure of how much leaf area there is per unit ground area, in conjunction with the area of individual leaves, as defined below:

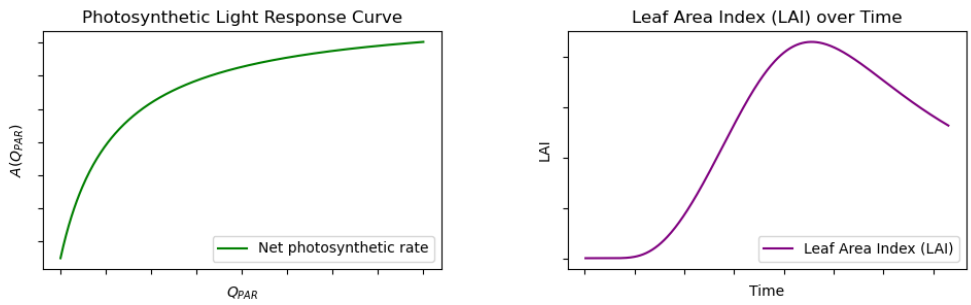
$$nL = LAI \times \frac{A_g}{A_L} \tag{18}$$

Where: nL is the number of leaves per plant ($leaves\ /\ plant$), LAI is the leaf area index of the plant ($m^2\ leaves\ /\ m^2\ plant$) as calculated in equation (19), A_g is the ground area per plant ($m^2\ /\ plant$), A_L is the area per individual leaf ($m^2\ /\ leaf$).

The leaf area index itself can be calculated as a fitted function of the cumulative GDD to evaluate how the crop is growing through time in response to the environment, as defined below:

$$LAI = \frac{Ce^{-k/TT(t)}}{1 + e^{S-(H/TT(t))}} \tag{19}$$

Where: LAI is the leaf area index of the plant ($m^2\ leaves\ /\ m^2\ plant$), C is a shape function coefficient ($m^2\ leaves\ /\ m^2\ plant$), k is another shape function coefficient ($^\circ C\ day$), $TT(t)$ is accumulated thermal time ($^\circ C\ day$) evaluated at time t ($hours$ or



(a) Net Photosynthesis Shape (b) Leaf Area Index Shape

Figure 9: Shape of Complex Functions used in Model

days depending on implementation) as defined by equation (20), S is another shape function coefficient (*unitless*), and H is another shape function coefficient ($^{\circ}C$ *day*). The shape of this function is described in Figure 9.

Given the leaf area index is a function of growing degree days, it is important to evaluate growing degree days up until the point in the season it's being evaluated at, as defined below:

$$TT(t) = \int_{t\ Start}^{t} GDDdt \tag{20}$$

Where: $TT(t)$ is accumulated thermal time ($^{\circ}C$ *day*) evaluated at time t (*hours*), GDD is the growing degree days ($^{\circ}C$ *day*) at time t as defined in eq. (21), and dt is the time step (*days*) (which will be (1/24 of a day to calculate at an hourly timestep).

Evaluating GDD accumulation at specific time steps is a simple calculation of the temperature difference between the air and the base temperature for growth, assuming the temperature is greater than the base temperature and below a maximum

temperature, as defined below:

$$GDD = \begin{cases} (T_{t,air} - T_{base}) & \text{if } T_{t,air} > T_{base} \text{ and } T_{t,air} < T_{hot} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Where: GDD is growing degree days ($^{\circ}C\ day$), $T_{t,air}$ is the air temperature ($^{\circ}C$) evaluated at time t ($hours$), T_{base} is the base air temperature for growth($^{\circ}C$), and $T_{t,hot}$ is the maximum air temperature for growth ($^{\circ}C$).

Crop Metrics

Since agrivoltaic crop performance cannot easily be described by a linear model, the yield of the system used in relation to the total amount of photosynthetic active radiation, as defined below:

$$LUE = \frac{Y}{Q_{SRad,\Sigma Season}} \tag{22}$$

Where: LUE is the light use efficiency of the crop ($\frac{tons\ /\ hectare}{kWh\ /\ m^2}$), Y is the yield ($tons\ /\ hectare$) as defined by eq. (14), and $Q_{SRad,\Sigma Season}$ is the sum total of solar radiation into the crop system ($kWh\ /\ m^2$).

Photovoltaic Modeling

Photovoltaics are the second half of agrivoltaic systems (agri- being agriculture and -voltaic being photovoltaic). Modeling their performance using various metrics is critically important, but given this implementation is focused on agricultural model,

the performance metrics are limited compared to what they could theoretically be, but the framework can support more complexity if desired. The first metric is to track how much power is produced through the entire seasons, as defined below:

$$\Sigma P = \int_{t\ Start}^{t\ Harvest} P_t\ dt \tag{23}$$

Where: ΣP is the total power ($kWh\ electricity$) absorbed by the solar panels evaluated over the entire season, and P_t is the power of the solar panels ($kW\ electricity$) evaluated at time t ($hours$).

Calculating the power of the photovoltaic panels is a function of the incoming solar radiation flux, the efficiency of each panel, and the reflectivity of each panel over the area of the panel times the time step, as defined below:

$$P_t = \left\{ \frac{Q_{SRad,abs} \times \eta_\rho \times \hat{n}}{A_{s\rho}} dt \mid s\rho \in \rho \right\} \text{ for each } \rho \in n\rho \tag{24}$$

Where: P_t is the electrical power produced by the solar panels ($kWh\ electricity$) evaluated at time t ($hours$), $Q_{SRad,abs}$ is the absorbed radiation flux of total solar radiation ($W\ /\ m^2$), η_ρ is the efficiency of the solar panels ($W\ electricity\ /\ W\ Solar\ Radiation$), $A_{s\rho}$ is the area of the subdivided solar panel (m^2), $s\rho$ is the subdivided solar panel ($division\ /\ solar\ panel$), ρ is the individual solar panel, and $n\rho$ is the total number of solar panels generated in the design.

Given the evaluation of power is done at a subdivided panel scale, the differences in light absorption can be calculated. Theoretically, the light on the panel should be uniform unless there's an external influence, like self-shading of the panels. Un-

derstanding how frequently this occurs is important to evaluating the photovoltaic performance, particularly within the context of the genetic algorithm which can produce random designs that cause significant self shading. This constraint is defined below:

$$SSH = \int_{t\ Start}^{t\ Harvest} \zeta dt \tag{25}$$

Where: SSH is the sum of self-shading hours ($hour$) on the solar panels evaluated over the entire season, ζ is the individual evaluation of self-shading of the panels, as defined in equation (26).

If a self shading condition is encountered, which causes problems for the energy production of the solar panel, there would be some difference between the maximum and minimum power values across the panel, as defined below:

$$\zeta(t) = \left\{ \begin{cases} 1 & \text{if } P_{min,s\rho}(t) \leq P_{max,s\rho}(t) \times d_3 \\ 0 & \text{otherwise} \end{cases} \mid \rho \in n\rho \right\} \tag{26}$$

Where: ζ is the evaluation of if self-shading has occurred at time t by comparing if the maximum power value at any panel subdivision ($P_{max,s\rho}(t)$) on panel ρ times a percent difference threshold (d_3) that is not greater than the minimum power ($P_{min,s\rho}(t)$) on the same panel ρ, indicating shading is causing non-uniform power distribution on the panels.

Genomic Optimization

A genetic algorithm for optimization was used to evaluate the agrivoltaic designs using methodology previously used in agrivoltaic performance optimization [Mengi et al., 2023]. To facilitate the genetic algorithm, a cost function for evaluating the performance of an individual design was needed, as well as a system for generating and evaluating strings.

Cost Function

A cost function must be developed to evaluate agrivoltaic performance, and several considerations should be made. Firstly, the cost function should be constructed in a way so it optimizes and evaluates the genetic algorithm in the same way. In this framework, the genetic algorithm seeks to find a design with the lowest cost. Since the performance metrics will also have different units and scales, it is helpful to evaluate them in a non-dimensional and normalized way so that no term will intrinsically overshadow any other term. It may be desirable to consider certain metrics with greater weight than others, and as such a user-defined weight term can be multiplied by each normalized cost. Additionally, there may be constraints that are necessary to enforce in the design. The overall form of the cost function that was used in this specific implementation is defined by equation 27:

$$\Pi = \omega_1 \alpha + \omega_2 \beta + \omega_3 \eta_L + P_1 + P_2 + P_3 \tag{27}$$

Where: ω_1, ω_2 and ω_3 are weights defined to assign greater or lesser influence to the

α, β, and η_L terms, which are defined in the next section.

Performance Metrics

The first metric for evaluating the performance of the agrivoltaics is the difference in crop yield between the agrivoltaic and control crops normalized by the control yield. As the agrivoltaic yield increases, the value of this term approaches 0, which is ideal in this framework, and if the agrivoltaic yield is higher relative to the control crop, the value will become negative, having a stronger influence in minimizing the rest of the function. It is defined below:

$$\alpha = \frac{Y_c - Y_{av}}{Y_c} \tag{28}$$

Where: α is the crop cost parameter defined as the the percent difference between the control crop and the agrivoltaic corp (denoted by subscripts c and av respectively) as calculated by eq. (14).

It is also desirable to evaluate agrivoltaic performance with respect to how the photovoltaic element performs in the system. The process for this evaluation is the same for the rest of the metrics, and is described below:

Crop performance evaluations can also help in assessing the performance of good agrivoltaic designs. The first metric evaluated is the difference in light use efficiencies of the control and agrivoltaic crops, as defined below:

$$\eta_L = \frac{LUE_c - LUE_{av}}{LUE_c} \tag{29}$$

Where: η_L is the light use efficiency cost parameter defined as the percent difference of the control light use efficiency and the agrivoltaic light use efficiency (denoted by subscripts c and av respectively) as calculated by equation. (22).

Penalty Terms

To enforce constraints on the designs, penalty terms can be added to the cost function to add arbitrarily high costs to designs that fail to meet certain constraints even if specific metrics are performing well. While any specific metrics can be added, within this implementation of the framework, 3 particular constraints are considered: 1) the agrivoltaic design has too much negative impact on crop yield, 2) agrivoltaic production too much compromise to either the crop or photovoltaic performance compared to single land use of the land for solely agricultural or photovoltaic use, resulting in poor land use efficiency and 3) if there is too much self-shading occurring on the panels which have negative impacts on the electricity production.

The first penalty looks at the ratio of agrivoltaic crop yield compared to the control and ensures the agrivoltaic crop is not having too much negative impact on crop yield, defined by an acceptable crop loss threshold, defined below:

$$P_1 = \begin{cases} P_1 & \text{if } \frac{Y_{av}}{Y_c} < 1 - \delta \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

Where: P_1 is a penalty term imposed if the ratio of the agrivoltaic yield to the control yield (both calculated via equation (14)) is less than δ, which is the acceptable loss percentage.

The second penalty term looks at land use efficiency. Theoretically, the land producing as much photovoltaic energy as possible or producing as much crop as possible is 100% efficient compared to itself. However, when the two systems are integrated, it is assumed some compromise has to be made, i.e., some of the crops has to be removed for installing the solar panels, and the solar panels have to be spaced further apart to facilitate sufficient lighting for the crop. However, if the compromise in production is low enough, i.e., only 30% of the crop is compromised, and 30% of the photovoltaic energy production is compromised, both of these systems are still 70% efficient uses of land by themselves, and it means the land is producing 140% of what resources would be produced in either system singularly. To facilitate this, a constraint is added to define what the desirable land use efficiency, or land use equivalent ratio, should be, as defined below:

$$P_2 = \begin{cases} P_2 & \text{if } \frac{Y_{av}}{Y_c} + \frac{\Sigma P_{av}}{\Sigma P_c} < LER \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

Where: P_2 is a penalty term imposed if the land equivalent ratio (LER) (the sum of ratios of agrivoltaic crop yield to control crop yield as calculated in equation (14) and agrivoltaic power production to control power production in equation (23)).

Photovoltaic performance is also greatly negatively affected by self shading, especially in designs with fixed panels that cannot react to the self shading. It is desirable to enforce a constraint which attempts to minimize the time the photovoltaic panels self shade, as defined below:

$$P_3 = \begin{cases} P_3 & \text{if } \ SSH_{av} > \sigma \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

Where: P_3 is a penalty term imposed if the number of SSH of the agrivoltaic system is greater than to σ, which is the acceptable number of self shading hours in the design.

Genetic Algorithm

The genetic algorithm is the heart of the genomic optimization. While there are many different approaches available, the approach by [Mengi et al., 2023] was used because it is simple to implement and had already been shown to work for this type of optimization.

The genetic algorithm operates by first simulating the performance of the control crop and photovoltaic design to have a reference, then generating random designs to be evaluated, ranked, and sorted, then continuing to generate new designs as functions of previous best designs and repeating until the best design is found. At the core of the genetic algorithm are the individual design strings, as defined below:

$$\mathbf{\Lambda}^i = \{\Lambda_1^i, ..., \Lambda_N^i\} = \{n\rho, l, w, h, \theta_a, \theta_e, \eta_\rho, \hat{n}, x, y\} \tag{33}$$

Where: $\mathbf{\Lambda}^i$ is the total design string, Λ_j^i denotes an individual design string parameter, $n\rho$ is the number of solar panels, l is the length of the individual solar panel (m), w is the width of an individual solar panel (m), h is the height of the center of the

individual solar panel (m), Θ_a is the azimuth angle of the panel ($degrees$), $Theta_e$ is the elevation angle of the panel ($degrees$), η_ρ is the power conversion efficiency of the solar panel ($kW\ electricity\ /\ kW\ solar\ energy$), $\hat{n}$ is the reflectivity of an individual panel ($kW\ reflected\ /\ kW\ solar\ energy$), x is half the total footprint of the photovoltaic design within the crop system (m) in the x direction, and y is half the total footprint of the photovoltaic design within the crop system (m) in the y direction.

The individual arrays then need to be aggregated into a generational array of the design strings, as defined below:

$$\mathbf{\Lambda} = (\mathbf{\Lambda}^1, ..., \mathbf{\Lambda}^i, ..., \mathbf{\Lambda}^S) \tag{34}$$

Where: $\mathbf{\Lambda}$ is the array of design strings from 1 to S, and S is the total number of design strings per generation of the genetic algorithm.

The designs then need to be evaluated and ranked and sorted such that the designs with minimum costs are top ranked, as defined below:

$$\min \left\{ \Pi(\mathbf{\Lambda}^i) \right\} \forall\ i \tag{35}$$

Where: the cost Π is evaluated for all $\mathbf{\Lambda^i}$ in i and the minimum value is chosen.

From this, the top ranked designs are then used to create random permutations of the top two designs in an effort to locate local minimums.

$$\Lambda^{(ci)} = \begin{pmatrix} n\rho^{pi}\phi_1 + n\rho^{p(i+1)}(1-\phi_1) \\ \cdots \\ \cdots \\ \eta_\rho^{pi}\phi_8 + \eta_\rho^{p(i+1)}(1-\phi_8) \end{pmatrix} \tag{36}$$

Where: Λ^{ci} is the new design string matrix containing all the parameters defined in equation 33, and each row of the matrix is parameter i (as described in equation 33) multiplied by random number ϕ_i, then added to parameter $i+1$ multiplied by random number $(1-\phi_i)$. This process generates child variations of the best parent design strings.

The same procedure is then slightly modified to replicate this process but introduce more random variation to locate local minimums.

$$\Lambda^{(c(i+1))} = \begin{pmatrix} n\rho^{p(i+1)}\hat{\phi}_1 + n\rho^{pi}(1-\hat{\phi}_1) \\ \cdots \\ \cdots \\ \eta_\rho^{p(i+1)}\hat{\phi}_8 + \eta_\rho^{pi}(1-\hat{\phi}_8) \end{pmatrix} \tag{37}$$

$\Lambda^{c(i+1)}$ is the new design string matrix containing all the parameters defined in equation 33, and each row of the matrix is parameter $i+1$ (as described in equation 33) multiplied by random number $\hat{\phi}_i$, then added to parameter i multiplied by random number $(1-\hat{\phi}_i)$. This process generates child variations of the best parent design strings.

Calibration and Validation

An image of the Helios Rendering can be seen in figure 10.

Calibration of the modeling framework is first performed using an average model of LAI and photosynthetic light response using values calculated from Chapter 1 (see Tables, values referenced in tables 7 and 8. The model described in previous sections was implemented using code shown in the Chapter 2 Appendix and was used to calculated the total accumulated biomass. This was used in conjunction with the ratio of fruit to total biomass to predict a crop yield. This crop yield was then divided by the accumulated CO_2 through the season (as calculated through the model described above) to determine how much biomass is generated per μmol / $hectare$ of CO_2 assimilated. This metric was chosen for calibration because it was not measured in the experiments, and had a significant impact on the results. The resulting number from this procedure was 1.33^{-7} $\frac{tons\ /\ hectare}{\mu mol\ CO_2}$ was calculated, as shown in equation 38. This value is then validated.

$$RUE = \frac{\Sigma BM \times RtFB}{\Sigma A} \tag{38}$$

Where RUE is the resource use efficiency ($\frac{tons\ /\ hectare}{\mu mol\ CO_2}$), ΣBM is the sum of biomass through the season ($tons$ / $hectare$), $RFtB$ is the ratio of fruit to biomass ($\frac{tons/hectare\ fresh\ yield}{tons/hectare\ total\ yield}$), and ΣA is the accumulated carbon through the season (μmol / $hectare$).

Validation was conducted using the parameters for three control and three agrivoltaic treatments from 2021 (C1, C3, C4, AV1, AV2 and AV3) described in Table 9

Table 7: Basic information for calibration and validation

Julian Start Date	Starting Year	Season Length	UTC	Latitude	Longitude
118	2021	140	7	38.53	121.77

Table 8: Crop model calibration parameter values used in the numeric example

Parameter	Units	Value	Parameter	Units	Value
LAI_a	$\frac{m^2\ leaves}{m^2\ plant}$	27.8	A_{max}	$\frac{\mu mol\ CO_2}{m^2 s}$	39.4
LAI_k	$^\circ C\ day$	345	A_k	$\frac{\mu mol\ CO_2}{m^2 s}$	0.07
LAI_S	–	4.56	A_{rd}	$\frac{\mu mol\ CO_2}{m^2 s}$	4.19
LAI_H	$^\circ C\ day$	1243	RUE	$\frac{tons\ /\ hectare}{\mu mol\ CO_2}$	1.33×10^{-7}
T_{base}	$^\circ C$	20	$RFtB$	$\frac{tons\ /\ hectare\ fresh\ yield}{tons\ /\ hectare\ total\ yield}$	0.77
T_{min}	$^\circ C$	20	dt	s	3600
T_{max}	$^\circ C$	45			

in conjunction with the RUE value of 1.33^{-7} $\frac{tons\ /\ hectare}{\mu mol\ CO_2}$ calculated earlier in equation 38. For agrivoltaic treatments, panels are simulated as demonstrated in figure 10. From this, the measured and simulated yields are compared, and a percent difference is also calculated. Absorbed radiation was also calculated for calculations done in Chapter 1.

The validation seems to suggest the measured and simulated yields for the control had a percent difference of 26.9% ± 7.75% while the agrivoltaic simulated and measured yields had a percent difference of 9.64% ± 7.92%. These are not particularly great results, and suggest the model should likely not be used when agronomically accurate results are needed, but these results are also not the worst possible results. Further refinement will likely be necessary. Possible sources of error and potential improvements will be discussed in the discussion section.

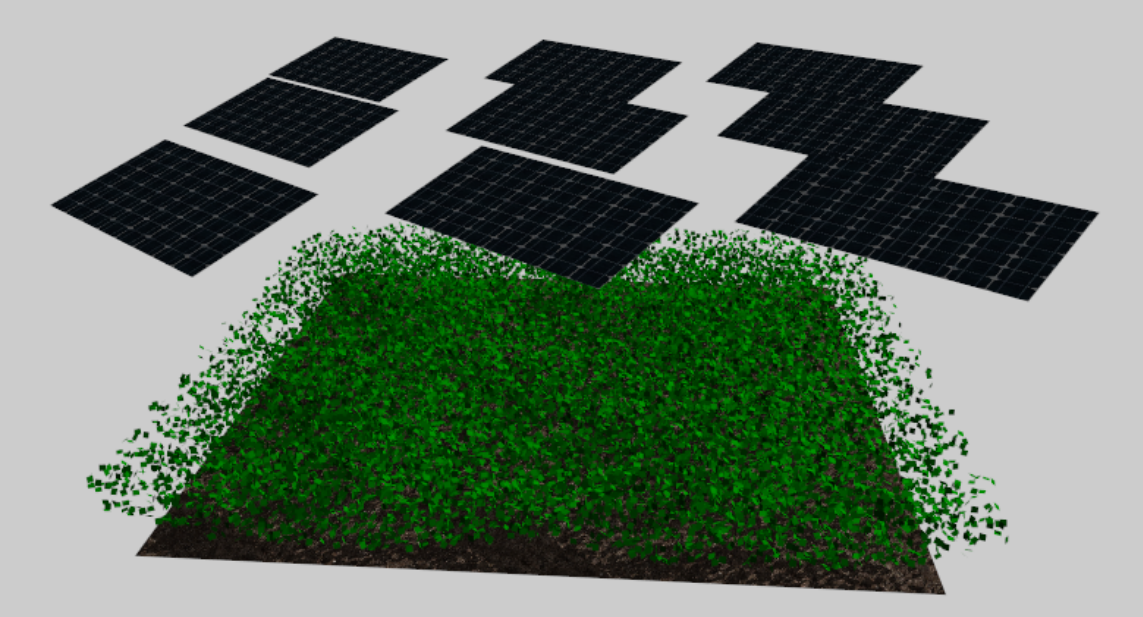


Figure 10: Helios rendering of agrivoltaic design used in validation

Table 9: Validation results used in evaluating model accuracy and error

Parameter	Units	C1	C3	C4	AV1	AV2	AV3
LAI_a	$\frac{m^2\ leaves}{m^2\ plant}$	318	181	264	71.7	488	349
LAI_k	C $^\circ C\ day$	2516	2171	2326	1659	2728	2307
LAI_S	[-]	4.69	5.11	5.59	6.29	6.75	4.92
LAI_H	$^\circ C\ day$	3726	4345	4524	6045	5155	3334
$RFtB$	$\frac{tons/hectare\ fresh\ yield}{tons/hectare\ total\ yield}$	0.65	0.82	0.76	0.73	0.74	0.64
A_{max}	$\frac{\mu mol\ CO_2}{m^2 s}$	24.0	22.00	32.0	32.4	43.7	30.8
A_k	$\frac{\mu mol\ CO_2}{m^2 s}$	0.05	0.06	0.05	0.08	0.10	0.05
A_{rd}	$\frac{\mu mol\ CO_2}{m^2 s}$	2.83	2.54	3.32	3.47	4.41	2.08
Measured Yield	$\frac{tons}{hectare}$	68.9	118	58.3	39.8	43.5	37.5
Simulated Yield	$\frac{tons}{hectare}$	52.1	82.2	69.2	32.6	43.3	34.5
Percent Difference	%	27.8	36.0	17.1	19.8	0.51	8.58
Average	%	26.9			9.64		
Standard Deviation	%	7.75			7.92		
Absorbed Radiation	$\frac{MJ}{m^2}$	43228	42144	42536	22336	20938	23923

Numeric Example

A numeric example is presented to demonstrate the capabilities of this model, designing an agrivoltaic system to produce a tomato crop in Northern California. Example

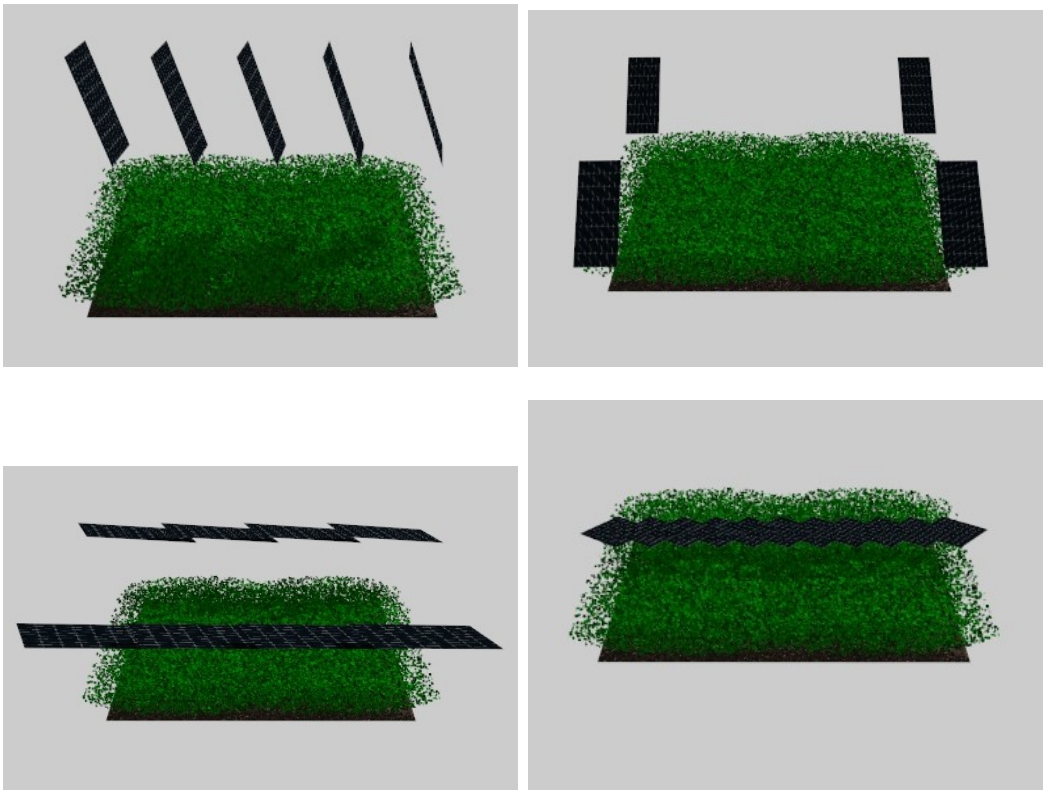


Figure 11: Random Designs

random designs from the simulation are shown in figure 11.

Crop Production Example in Northern California

As a demonstration, a simulation is run to optimize an agrivoltaic design for the processing tomato crop grown in Chapter 1 with the model under climate and light conditions representative of Davis, California. The crop was grown assuming the crop is fully irrigated, well fertilized, and grown during a typical growing season for processing tomatoes. Specific numbers are shown in 10, 8 and 11.

Table 10: Table with Basic Information for Tomato Production in Northern California

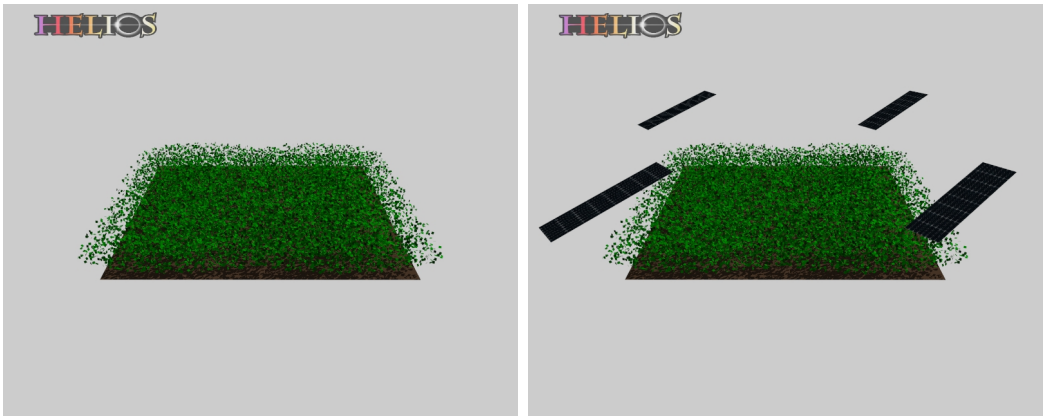
G	S	K	P	Julian Start	Start Year	Season Length	UTC	Latitude	Longitude
15	15	4	2	15	2022	130	7	35.2	119.1

Table 11: Table with agrivoltaic design settings

Parameter	Units	Value	Description
w_1	–	1	Weight for crop performance
w_2	–	1	Weight for photovoltaic performance
w_3	–	1	Weight for LUE performance
p_1	–	1000	Crop loss penalty
p_2	–	1000	LER failure penalty
p_3	–	1000	Self Shading penalty
d_1	–	0.7	Acceptable loss threshold %
d_2	hr	30	Acceptable self shading hours per season
LER	–	1	Land equivalent ratio
nP	–	[1,16)	Number of solar panels
L	cm	[30,150)	Length of solar panels
W	cm	[30,150)	Width of solar panels
H	cm	[50,150)	Height of solar panels
Az	degrees	[0,360)	Azimuth of solar panels
El	degrees	[0,91)	Elevation angles of panels
Tr	–	[0,0]	Translucency of solar panels
Eff	–	[0.2,0.2]	Efficiency of solar panels
$xLimit$	m	1.52	Length from (0,0) in x-direction
$yLimit$	m	1.52	Length from (0,0) in y-direction

Results

For the example crop described, the best agrivoltaic design found was a single panel design with panels of 76 cm by 106 cm size, placed at the height of 241 cm off the ground, oriented 136 ° from North with an azimuth of 39°, as illustrated in Figure 12. Specific simulation performance and crop meta performance are highlighted in Figures 12, 13, 14, and 15. This crop was found to have had a reduction of 14.5%



(a) Control Crop Simulation (b) Agrivoltaic Crop Simulation

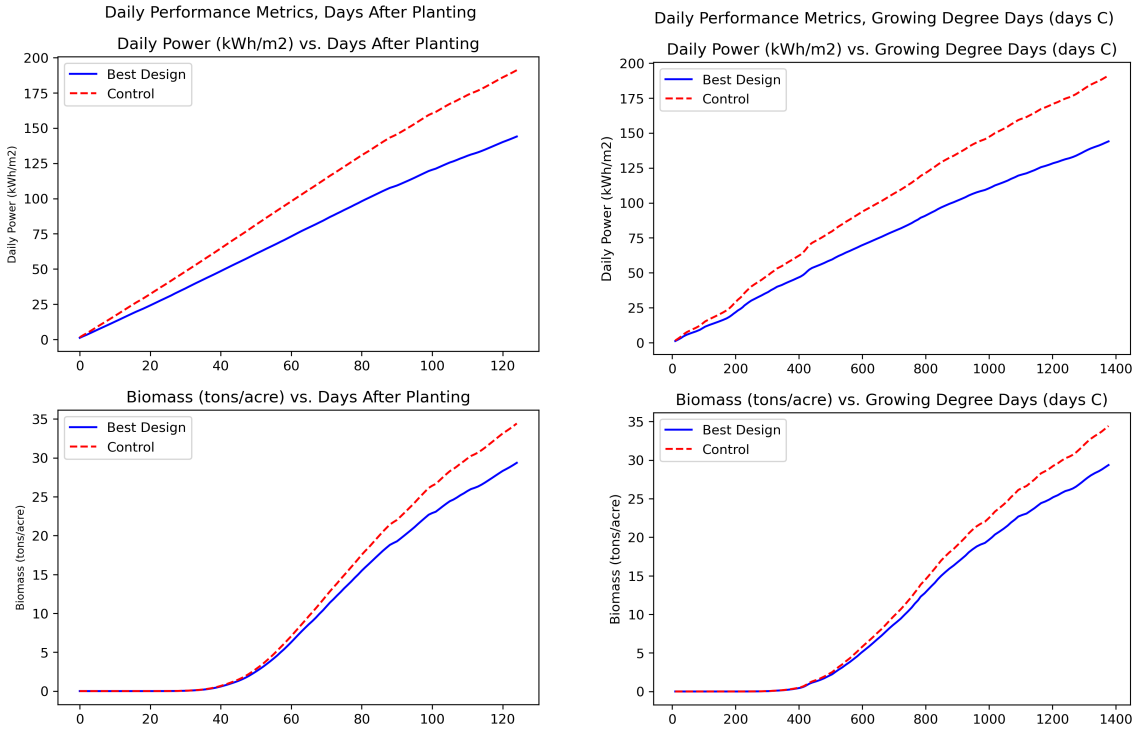
Figure 12: Control and agrivoltaic crops

in crop yield compared to the control, a 24.6% reduction in potential photovoltaic energy produced through the season, and a 7.69% reduction in light use efficiency from the crop.

Table 12: Summary table of agrivoltaic performance

	Yield	Power	Light Use Efficiency
Treatment	$\frac{tons}{hectare}$	$\frac{kWh}{season}$	$\frac{tons/hectare}{MJ/m^2/season}$
Control	55.8	191	1.66×10^{-3}
Agrivoltaic	65.5	144	1.80×10^{-3}
% Difference	14.5%	24.6%	7.69%

Figure 14 and 13 show how the photovoltaic panels produced power and the crop developed throughout the season, both with respect to days after planting and growing degree days, as well as the average hourly performance of photovoltaic energy production and photosynthesis (Figure 14). The results of the biomass simulations suggest the main difference in crop growth between the agrivoltaic and control treatment seems to increase later in the season as LAI is increasing and the canopy be-



(a) Days After Planting Performance (b) Growing Degree Days Performance

Figure 13: Daily crop performance

comes denser. Also, analysis of the average hourly performance of power production shows the power generation is lowest during the early day and maximizes during the afternoon, which is expected given the orientation of the panels. The photosynthesis rate seems to mirror the same pattern as the average photosynthesis rates seem to dip as the solar panels start to absorb power. Figure 15 also shows some interesting trends in the genetic algorithm. Firstly, an optimal solution was found for 6 of the 15 simulated generations, while it appears to have only 3 generations for the system to find average solutions which minimized the penalties of bad-performing designs. The optimization seemed to be mainly driven by the algorithm finding designs that

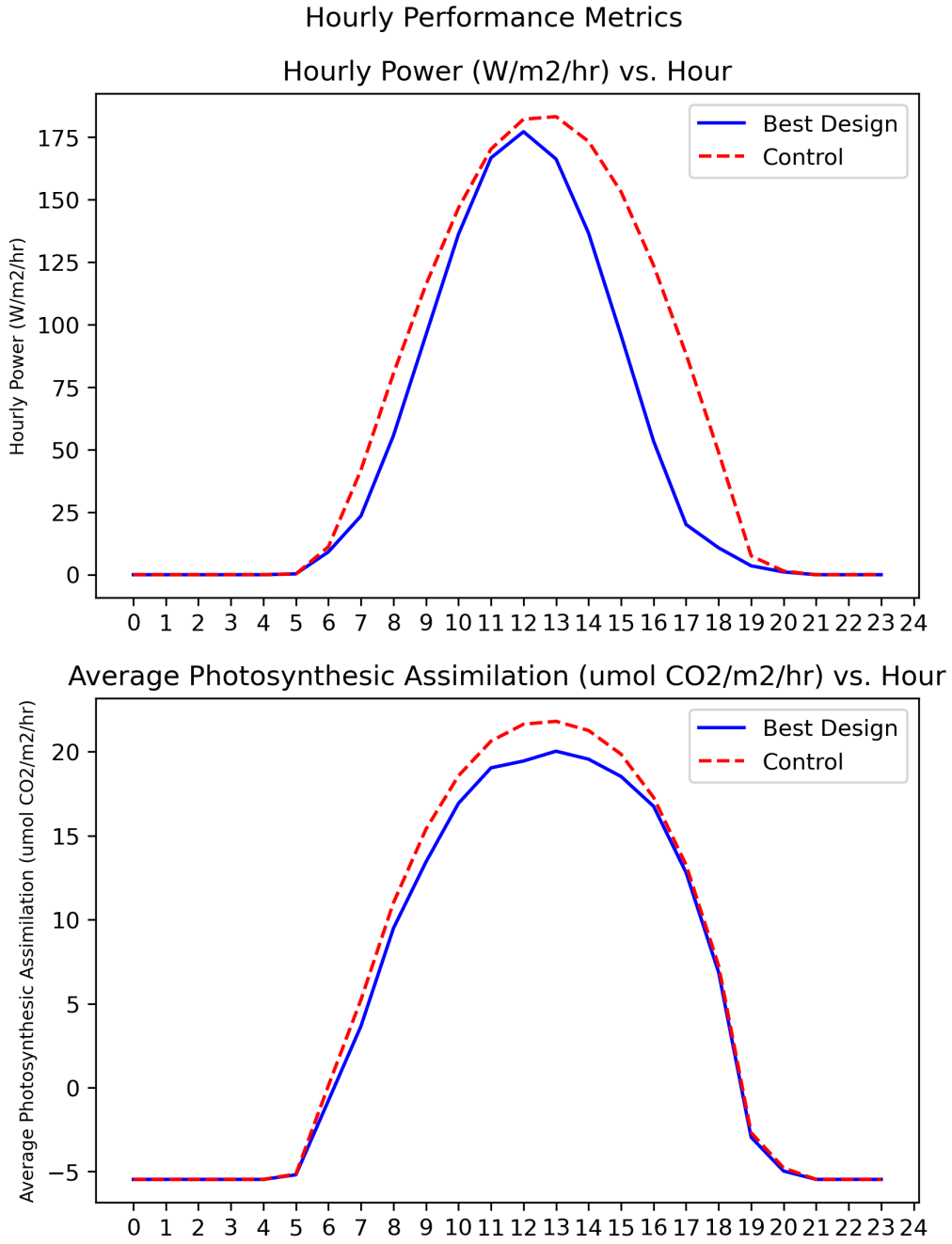
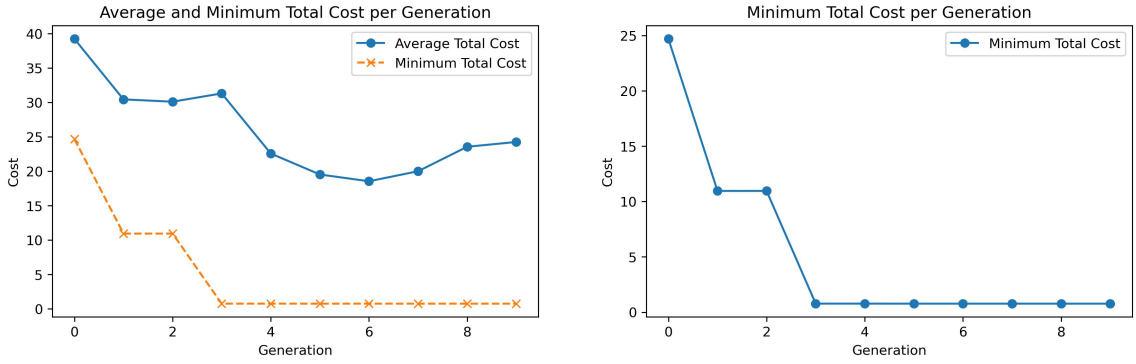
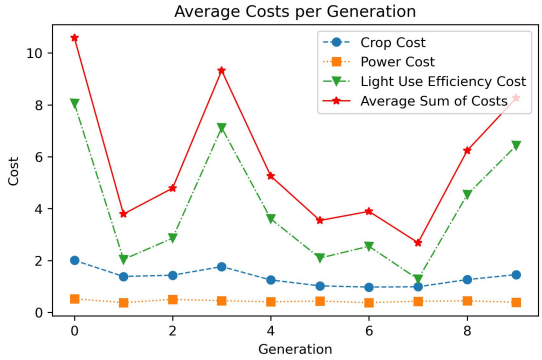


Figure 14: Hourly average crop performance



(a) Generation Costs

(b) Minimum Cost



(c) Individual Costs

Figure 15: Genetic Algorithm Performance

minimized crop losses and improved light use efficiency. The relatively low and quick minimization of the cost suggests the algorithm could be run more intensively to find even more optimum designs.

Discussion

Findings

The developed modeling and optimization framework was capable of finding relatively quickly good agrivoltaic designs. Upon analysis of the genetic algorithm performance, it appears little optimization is occurring, but in reality, the detailed numeric analysis suggests that good designs are found quickly through the parallel simulations and then are fine-tuned through successive iterations. This suggests more aggressive optimization parameters could be introduced to further optimize system designs.

The developed framework provides high-resolution simulations of agrivoltaic crop performance at sub-daily time steps (specifically tested for hourly time steps here) to capture the nuances of the spatio-temporal light distribution and subsequent plant responses. The framework also possesses an ability to integrate other time responses if desired. A large range of metrics are also captured, allowing for complex evaluation criteria to be calculated when evaluating agrivoltaic performance. The genetic algorithm approach allows for the complexity of non-linear and non-discrete responses in agrivoltaic performance to be modeled effectively and optimized.

In particular, previous work by the author in [Mengi et al., 2023] highlighted the importance of using an hourly time step in the crop model and formulating the cost function in a manner which facilitates non-linear crop performance. Both [Mengi et al., 2023] and [Campana et al., 2021] used a daily crop model, which had the effect that the simulated agrivoltaic optimums resulted in vertical photovoltaic

panels as it seemed to prioritize maximizing energy in the cropping system without respect to the nuances of how that energy is actually being used. Through the utilization of an hourly crop model and subsequent cost parameters for light use, it was found that designs that had more flat solar panel angles performed better, limiting light during peak times while still maintaining high average levels of photosynthesis as shown in Figure 14.

As highlighted by the validation of the modeling framework with field-observed data, there is room for model improvement. The developed framework only uses a single calibrated coefficient (the RUE value) based on a single average. It also includes certain assumptions about the crop development and canopy geometry which may need refinement. A number of these important assumptions include, shape and location of leaves, distribution of leaves, the structure of stems within the plant, and key phenological moments in the plant life-cycle. While this limits the applications of the model, the model still serves as a framework to demonstrate the potential of these types of approaches for future works.

Future Extensions

Because this simulation framework utilizes the Helios software system, which both requires CUDA [NVIDIA Corporation, 2023] (a software system which requires specific NVDIA graphics cards to run raytracing and other mathematical operations on the multitude of cores on the graphics card instead of the few cores located on the main computer CPU) and simulates potentially millions of individual primitive geometries (which include the leaves, solar panels and ground in this case) with iter-

ative calculations, the simulation requires relatively high-performance hardware, and can easily take over 24 hours to run (depending on a combination of factors). It is unclear whether this ultra-high resolution data provides better results than less computationally intensive approaches, as this analysis does not conduct a performance analysis of different computational methods. In particular, many operations within Helios, like ray tracing of diffuse light through translucent surfaces and solving for energy balance, require iterative solutions. In individual simulations, it may not be a major issue, but when attempting to run hundreds of simulations involving millions of individual primitives for every hour of every day of a defined crop season, the iterative calculations add significant time to calculations. From this, alternative methods of canopy simulation should be investigated and considered in an effort to balance the simulation resolution with computational resources.

Additionally, while the crop modeling framework is highly responsive to the types of impacts agrivoltaics are expected to have on crop performance, 3D crop modeling is developing, and there are a lot of considerations to 3D crop development that are not considered within this framework. In particular, because this framework utilizes Helios' canopy generator for generic crops, it doesn't capture the nuance that may be expected of leaf distribution, timing, shade response stretching, etc., associated with agrivoltaic crop development. Within the context of a generic crop framework, this is a functional first step in optimizing agrivoltaic designs and learning the potential impacts of complex design choices on agrivoltaic performance e. More specific crop models would be beneficial, however, and Helios can model some of these behaviors if the right work is done to validate.

The model, as presented, assumes panels are of fixed location and angle. This approach is possibly good for low-complexity photovoltaic designs. However, many photovoltaic systems rotate in one or two axes and are a critical part of photovoltaic infrastructure. Rotating panels also follow certain performance algorithms. To facilitate these algorithms, this framework would need to be extended to facilitate panel rotation and accept different rotational algorithms, which could theoretically be designed around optimizing agrivoltaic performance. Given the complexity of solutions for generating and rotating photovoltaic patterns, this genetic algorithm approach is still likely a good candidate as gradient-based optimizations are not necessary, and these complexities can be optimized through the stochastic methods presented here.

Lastly, there is often a desire for economic performance when discussing practical applications of agrivoltaic systems. While this model can handle pseudo-economic performance indicators by looking at some of the penalties and thresholds as economic performance metrics, this model does not specifically have economic analysis built in. Other models do integrate economics, and integrating economic models would be a meaningful addition when considering the complexities of agrivoltaic design. Therefore, adding economic assessment options would be a valuable and low-complexity extension for future versions of this model.

Conclusions

A modeling framework was developed to optimize the design of the agrivoltaic system. This framework expanded on previous work presented in [Mengi et al., 2023] by capturing some of the nuances of crop production and the non-linear photosynthesis

created by photovoltaics by simulating crop growth at an hourly time scale while also improving model performance through parallel processing and utilization of CUDA for graphics card accelerated ray tracing. A genetic algorithm was used to explore complex design spaces based on first-order models. Genetic algorithms present a powerful opportunity to study various design scenarios and understand what "good" agrivoltaic design may be. Through the use of a generic crop model and capable of simulation crop growth for most crops with the accounting for non-linearity in plant responses, it is demonstrated that agrivoltaic design should be done at an hourly time step rather than the daily time step that most models are using. The modeling framework can function as a survey method to understand how certain agrivoltaic designs will impact the crops grown and build the knowledge base for agrivoltaic engineers as this field develops. While there are still opportunities for improving the model, it serves as a framework for the future of agrivoltaic design and optimization.

Chapter 3: Evaluating Agrivoltaic Potential in California

Abstract

As agrivoltaic technologies develop globally, there is a natural interest in evaluating their potential at scale. Evaluating agrivoltaic potential, however, is an open-ended concept that can be approached from many different angles, and one such angle is what the potential of large-scale deployment of agrivoltaic systems in California agriculture could be. This study had two objectives: 1) evaluating the potential of agrivoltaic systems by first developing an agrivoltaic suitability model to evaluate California farmlands suitable for agrivoltaic development, and 2) conducting a statewide geospatial analysis evaluating the impacts of agrivoltaic deployment (for a generic agrivoltaic system which provides 15% shading and is 15% efficiency) with respect to land use, water demand, energy production and irrigation energy demand, carbon dioxide offsets, and land use efficiency for the State of California. The results showed a capacity of 128,000 km^2 of California's farmland has agrivoltaic potential (approximately 80,000 km^2 of which are grazing land, and the rest conventional cropping land), could reduce annual water use by 470 million m^3/yr, and has the potential to offset 819 $kt\ CO_2/yr$. Agrivoltaic production has the capacity to generate 2.86 million GWh/yr of electricity while reducing irrigation power demand by 4,540 GWh/yr. Under a 1% adoption rate scenario, assuming 1,280 km^2 would be developed, 29,000 GWh/yr of electricity could be produced, reducing irrigation demand by 65 million m^3/yr and reducing energy demand for irrigation by 45 GWh/yr with

a corresponding CO_2 offset of 8.19 $kt\ CO_2/yr$.

Introduction

Interest in agrivoltaic systems, a novel form of agricultural production where photovoltaic panels are placed over agricultural crops, is increasing across the world [Al Mamun et al., 2022], and there is a natural question of where are best locations to site agrivoltaic farms. Estimating agrivoltaic performance and opportunities for their development are complex questions that have not been researched in detail to date. Depending on how agrivoltaics can be strategically developed, from sustainability tools supported at the governmental level to individual farms adopting performance enhancing technology, there is much ambiguity in determining where agrivoltaics should be placed. As such, it is helpful not to propose one single "optimal" analysis for evaluating the potential of agrivoltaics but rather develop a framework for exploring the potential of agrivoltaics in regions with great photovoltaic potential such as California.

Previous studies geospatially analyzing agrivoltaic potential [Kwon et al., 2020], [Coşgun, 2021], [Majumdar and Pasqualetti, 2018], and [Ali Khan Niazi and Victoria, 2023] are insightful in analysis, particularly with respect to evaluating photovoltaic potential globally, but not specifically answer how to explore suitable locations for agrivoltaics in California. Given the unique Mediterranean climate in California [Pathak et al., 2018] aand the competition between agricultural land and land requirements for large-scale photovoltaic energy production [Sargentis et al., 2021], [Havrysh et al., 2022], as well as water stress

impacting agricultural production [Escriva-Bou et al., 2017] and California's drive to increase renewable energy production [Jafarinejad et al., 2023], agrivoltaic systems provide opportunities to mitigate land conflicts between agricultural and photovoltaic land uses and improving agricultural resource sustainability in light of climate change.

This study will evaluate the potential of deploying agrivoltaics in California by developing criteria for identification of suitable locations for agrivoltaic systems in California. A suitability model is calculated considering crop classification, photovoltaic suitability, and water shortage vulnerability as, and used to calculate a maximum potential for different parameters (area, power production, water demand reduction, irrigation energy demand reduction, carbon dioxide offsets, land use efficiency) associated with agrivoltaic production in California

Methods

A suitability model is developed for Californian farmland using a GIS approach and different geospatial datasets (described in detail in the following sections) to evaluate the potential of agrivoltaic systems. Agrivoltaic performance metrics (suitable area, power production, water demand reduction, irrigation energy demand reduction, carbon dioxide offsets, land use efficiency) are also calculated geospatially, then multiplied by the suitability model (which is ranking suitability using a scale from 0-100) to calculate a maximum potential for these performance metrics.

Evaluation Criteria

To evaluate the suitability of farmland in California for agrivoltaic production, the following logic is proposed to start conceptualizing a model. Land is more suitable for agrivoltaic development when:

- The land is of a compatible crop classification
- The land is suitable for photovoltaic energy development
- The land is vulnerable to water shortages

Land being of a compatible classification is reasoned by proposing certain types of land classification will be more (negatively) or less (positively) impacted by agrivoltaic development, and these impacts could result from scientific concepts or legal concepts. The land being suitable for photovoltaic development is also a critically important concept because if the land is not suitable for photovoltaic panels, then it is less likely that photovoltaic panels would be installed, which is intuitive but also should be made explicit decision criteria. The land vulnerable to water shortages is reasoned into this analysis because agriculture depends on water access. If a farm is vulnerable to water shortages, this increases the need to implement strategies to reduce water demand. Given photovoltaic panels can shade the land and reduce water usage, there is a synergy here in which agrivoltaic systems could help generate electricity, promote agricultural production, and reduce the water demand of the farm. These concepts are all relevant to the California context and are thus used as a framework for analyzing the potential of agrivoltaic production in California.

From this conceptual framework, an equation was formulated to evaluate the suitability of land for agrivoltaic development (on a scale from $0-100$), which is defined as:

$$AVS = \frac{w_1 CCS + w_2 PVS + w_3 WSV}{3} \quad where\ w_1 + w_2 + w_3 = 3 \tag{39}$$

Where AVS is the Agrivoltaic Suitability Score, CCS is the Crop Compatibility Score (as defined in table 14 and described in the model assumptions section), PVS is the photovoltaic suitability score as defined in the data sources as the data source for photovoltaic suitability,WSV is the water stress score as defined in the data sources for water shortage vulnerability as a metric of water stress, and $w_1,\ w_2,\ w_3$ are subjective weights used to emphasize certain parameters as desired for analysis. All three metrics are evaluated on a scale from $0-100$, then divided by 3 to ensure the resulting score is also on a scale of $0-100$.

From this index, information about the impacts of agrivoltaics may have can be calculated by looking at the following metrics relating to agrivoltaic performance:

- Photovoltaic energy generation of the agrivoltaic systems
- Acreage of agrivoltaic installations
- Change in water demand (evapotranspiration) on agrivoltaic land
- Pumping and irrigation energy reductions associated with changes in evapotranspiration

- Carbon Dioxide offsets from photovoltaic energy generation as well as pumping and irrigation energy demand reduction
- Land use efficiency of agrivoltaic production

Analysis Model

To estimate if a given land parcel will adopt agrivoltaic production, the maximum potential is estimated by assuming a percentage of agrivoltaic production in relation to the farm's suitability (or, alternatively, a capacity for a field to adopt agrivoltaic development) is the maximum capacity of adoption of agrivoltaic systems.

$$P(A) = \frac{AVS}{100} \tag{40}$$

Where $P(A)$ is the probability of adoption for a given field ($unitless$), and AVS is the agrivoltaic suitability score for a given field ($unitless$).

Power production is also considered as the sum of direct and diffuse irradiance per area and per day, multiplied by the density of agrivoltaics (in relation to how much power they absorb), the efficiency of the panels, the area of production at a given probability of adoption, times the number of days in the year.

$$P_{AV} = DNI \times A \times P(A) \times \rho_{AV} \times 365 \tag{41}$$

Where P_{AV} is the agrivoltaic power production at a given probability of adoption (kWh), DNI and GHI are the direct normal irradiation and global horizontal irradiation ($kWh/m^2/day$), respectively, A_{tot} is the area of agrivoltaic production (km^2),

as calculated in equation 42, ρ_{AV} is the density of agrivoltaic panels ($unitless$), η_{AV} is the efficiency of the photovoltaic panels ($unitless$), AVS is the agrivoltaic suitability score ($unitless$), $P(A)$ is the percent probability of adoption of the specific scenario ($unitless$), and 365 is the number of days in non-leap years.

The area of agrivoltaic production is calculated by looking at the area of an individual farm and multiplying it by the probability of adoption.

$$A_{tot} = A_{field} \times P(A) \tag{42}$$

Where A_{tot} is the total area of agrivoltaic production (km^2), A_{field} is the area of an individual field (km^2), and $P(A)$ is the probability of adoption ($unitless$).

The reduction in water use, or water savings, is calculated as the estimated reduction of evapotranspiration times the area of the evapotranspiration to calculate a volume of water savings.

$$\Delta V_{AV} = (ET_{Control} - ET_{AV}) \times A \tag{43}$$

Where ΔV_{AV} is the change in volume of water demand (m^3), $ET_{Control}$ and ET_{AV} (m) are the evapotranspiration amounts associated with the control and agrivoltaic production, and A_{tot} is the area of agrivoltaic production (km^2).

Once water savings are calculated, the energy associated with those water savings can be calculated. First the saved energy associated with pumping is estimated, which is the volume of water saved times the density of water, gravity, the pumping depth, and the proportion of irrigation pumped from groundwater.

$$\Delta E_{pump} = \rho \times g \times h \times \Delta V_{AV} \times PGP \tag{44}$$

Where ΔE_{pump} is the energy reduction associated with reduced pumping demand (J, then converted to kWh for the analysis), ρ is the density of water (kg/m^3), g is the acceleration of gravity on Earth (m/s^2), h is the pumping depth of ground water for a given field (m in the equation, but converted from $ft.$ in the data), ΔV_{AV} is the change in volume of water demand associated with agrivoltaic production (m^3), and PGP is the proportion of water demand that is pumped ground water ($unitless$), as explained in the model assumptions section.

Irrigation energy can also be calculated by looking at the reduction in water volume times the pressure difference times the proportion of pressure irrigated agriculture.

$$\Delta E_{irrag} = \Delta V_{AV} \times \Delta P \times PIA \tag{45}$$

Where ΔE_{irrag} is the energy reduction associated with reduced pressurized irrigation demand (J, then converted to kWh for the analysis), ΔV_{AV} is the change in volume of water demand associated with agrivoltaic production (m^3), ΔP is the amount of pressurization needed (Pa, converted from PSI), and PIA is the proportion of water demand that is delivered via pressurized irrigation ($unitless$), as explained in the model assumptions section.

Carbon Dioxide offsets can also be calculated by looking at the power produced times the carbon offset associated with power production multiplied by the irrigation

energy savings and the carbon offset associated with irrigation.

$$\Delta CO_2 = \gamma \times P_{AV} + \lambda \times (\Delta E_{pump} + \Delta E_{irrag}) \tag{46}$$

Where ΔCO_2 is the CO_2 offset associated with agrivoltaic production at $P(A)$ ($tons\ CO_2$, γ is the carbon emission factor associated with photovoltaic energy production ($tons\ CO_2/kWh$), P_{AV} is the agrivoltaic power production at a given probability of adoption (kWh), λ is the carbon emission factor associated with agricultural water demand ($tons\ CO_2/kWh$), and $\Delta E_{pump} + \Delta E_{irrag}$ is the energy reduction associated with reduced agricultural water demand (kWh).

Lastly, it is helpful to use another metric that looks at the efficiency of land use with how much power is produced and how much energy is saved per unit of land.

$$LUE = \frac{P_{AV} + \Delta E_{pump} + \Delta E_{irrag}}{A_{AV}} \tag{47}$$

Where LUE is the land use efficiency of agrivoltaic production at a given adoption probability (kWh/m^2), P_{AV} is the agrivoltaic power production at a given probability of adoption (kWh), $\Delta E_{pump}+\Delta E_{irrag}$ is the energy reduction associated with reduced agricultural water demand (kWh) and A_{AV} is the agrivoltaic production acreage at a given probability of adoption (m^2).

Model Assumptions

Agrivoltaic systems are an emerging technology that offers highly customizable systems that can be adapted to any site requirement. However, California agriculture

Table 13: Table of Numeric Assumptions Used in the Analysis

Parameter	Value	Units	Source
Percentage of Pumped Irrigation	80%	unitless	approximated from [Jha et al., 2022]
Percentage of Pressure Irrigation	70%	unitless	approximated from [Jha et al., 2022]
Shading/Light Absorption	15%	unitless	derrived from [Evans, 2014], [Magarelli et al., 2024]
Photovoltaic Efficiency	15%	$\frac{kWh\ Energy}{kWh\ Irradiance}$	estimated from [NREL, 2012]
CO_2 Offset of Natural Gas	52.91	$\frac{kg\ CO_2}{million\ Btu}$	[EIA, 2023]
CO_2 Offset of Gasoline	67.39	$\frac{kg\ CO_2}{million\ Btu}$	[EIA, 2023]
CO_2 Cost of Photovoltaic Panels	40.0	$\frac{gCO_2\ equivalent}{kWh}$	[NREL, 2012]
$w_1,\ w_2,\ w_3$	1	unitless	chosen

is one of the most productive agricultural production systems in the world, hence, adoption of agrivoltaics cannot diminish agronomic outputs reached by commercial production. To accommodate these realities, assumptions are made to conceptualize and reasonably estimate the scale of agrivoltaic potential. Assumptions with mathematical parameters are listed in Table 13 alongside their sources.

The first assumption is the probability of agrivoltaic adoption, which is the combination of how suitable the land is for agrivoltaic development multiplied by a target adoption rate. This means if a given field is 85% suitable for agrivoltaic development, at a 100% adoption rate, it will only become 85% developed for agrivoltaic, as defined by equation (40). Given the complexities of agrivoltaic adoption being unknown, as well as not all forms of crop production being fully compatible with agrivoltaics and other potential discrepancies like higher photovoltaic density on some parts of the land and not others, some of the power going to non-grid purposes, etc., as well as the general inertia towards adopting such a radical change to agricultural production, this aims to model these discrepancies.

The next assumption is that agrivoltaics will have acceptable impacts (both positive and negative) on crop production in a specific agrivoltaic system. The reasons

for this assumption are: Californian agriculture encompasses over 400 types of crops [California Department of Food and Agriculture, 2022], meaning it is incredibly difficult to model every specific crop impact at this time, and it is possible crops grown can shift and production practices can adapt to agrivoltaic production. This assumption has two impacts on the modeling: 1) there is no modeling of the impact of agrivoltaics on statewide food production, and 2) it is assumed that agrivoltaics will absorb 15% of the energy for a given plot. Because there are no large-scale examples of well optimized agrivoltaic systems to use as case studies for establishing an average shading level, greenhouses serve as a substitute of a well-established form of agricultural production with positive yields within the context of agriculture in light reduced systems, and typical greenhouse materials typically reduce energy into the systems between 2-30% [Evans, 2014]. A recent literature review of agrivoltaic crops also suggested limiting shading to no more than 30% [Magarelli et al., 2024]. And while some crops may desire greater or lesser shading, assuming 15% is a conservative estimate for the "average" acceptable shading in agrivoltaic system.

The next assumption is that the reduction in evapotranspiration is linearly reduced by shading, i.e. the 15% shading assumed will result in a 15% reduction in evapotranspiration, as defined in equation (48). While evapotranspiration is a complex process with many variables, one of the well-established energy balance approaches, the Penman-Montieth model [Allen, 2005], shows a linear relationship between solar radiation and evapotranspiration. Agrivoltaics also have mixed impacts on temperature (another important factor in evapotranspiration) in an agrivoltaic system [Al Mamun et al., 2022]. Large-scale agrivoltaics also likely have an impact

on wind profiles in agricultural fields, but this is not well-documented. These temperature and wind speed effects, however, are difficult to model with this particular framework, but the impacts of agrivoltaics can be assumed to reduce the amount of evapotranspiration through the reduction in available solar radiation. On the flip side, the presence of the photovoltaic panels generates some degree of heat and prevents some energy from leaving the ground, which would increase the evapotranspiration potential of agrivoltaics. This means this assumption likely results in a conservative estimate of the reduction of evapotranspiration in agrivoltaic systems at the state scale.

$$ET_{AV} = ET_{Control} \times (1 - \rho_{AV}) \tag{48}$$

The next set of assumptions is related to pumping and irrigation energy demand. The first of these assumptions is that pumping depth at the time of measurement is representative of typical pumping depth. This is a challenging assumption as pumping depth varies throughout the year, but without a more robust data set and doing the analysis on a shorter time scale, it is difficult to get a more accurate measurement. Whether this over or underestimates energy demand is also difficult to ascertain. However, given the analysis is conducted as the relative difference between control and agrivoltaic scenarios, and the purpose of this analysis is primarily getting scales of potential, the assumption is contextually acceptable. It should still be noted the values provided through this analysis should be interpreted as scales of magnitude rather than specific, accurate results. Also relating to pumping, 80% of the irrigation demand is assumed to come from groundwater pumping. Depending on the region

and time of year in California, up to 100% of water may be supplied by ground water [California State Water Resources Control Board, 2022]. WWhile a lot of this pumping occurs in agricultural regions during periods of the year with high water demand, it is difficult to confirm without a comprehensive spatio-temporal water budget of every field in the state. As such, this assumption helps generate some reasonable scale but cannot be taken as a specific number without conducting a comprehensive water budget of California, which is beyond the scope of this analysis. Inrelation to irrigation systems, most but not all of Californian agriculture is irrigated with pressurized systems [Jha et al., 2022], and as such, an approximate value of 70% of agriculture is assumed to be irrigated in this analysis. Pressurized irrigation systems also require different levels of pressurization to function correctly, depending on their type. To aggregate all the different potential irrigation systems, an average value of 10 PSI of pressurization is chosen, as some will require more and others less, and some forms of irrigation, like flooding, do not require pressurization.

To model CO_2 offsets, a series of assumptions need to be made relating the carbon costs of photovoltaic production as well as the energy it will replace. While photovoltaic energy can be compared to offsets for coal for maximum carbon accounting, California's energy grid does not utilize significant coal production but does contain significant amounts of natural gas [California Energy Commission, 2022]. Additionally, California has legislated demands to expand electric vehicle usage [California Air Resources Board, 2022a]. Given electrons on the grid are functionally untraceable, and carbon offsetting has to be compared to potential emissions of these untraceable electrons, which would no longer exist, it is assumed the energy

Table 14: Crop Classification Suitability Scores Used in the Analysis

Crop Classification	**Score**
Grazing land	100
Farmland of Statewide Importance	80
Farmland of Local Importance	60
Unique Farmland	40
Prime Farmland	20

grid mix being replaced by agrivoltaic energy generation is half natural gas for the general grid, and half gasoline for electric vehicle networks as it is been suggested agrivoltaics could support electric vehicle networks [Steadman and Higgins, 2022].

Lastly, several assumptions were made regarding the crop suitability for agrivoltaics described in Table 14. Grazing land is given the highest suitability score because the low densities of grazing crops lend well to not being dramatically impacted by agrivoltaic shading and not being harvested, meaning grazing land yields are not of concern for the market nor are accommodating farming equipment. Farmland of statewide importance is given the next highest score on the rationale that it is farmland important for the State to allocate resources to both protect agricultural production as well as expand renewable energy deployments and could deploy resources to develop those lands. Farmland of local importance is given the next lower score for a similar reason as it is possible local governments may choose to invest and allocate resources, but could also be more protective of conventional crop production and generally have fewer resources to mobilize than the State. The unique farmland category is next and is given its score because it is possible for specific or niche applications to adopt, but also the uniqueness of certain plots might make them more resistant to change. Lastly, the prime farmland category gets the lowest score as

this land is defined as land that has sufficient water and is highly productive. This suggests prime farmland may have the lowest incentives to change farming practices just to adopt novel technologies.

Data Sources

Land use and land cover data comes from the California Department of Water Resources 2021 Statewide Crop Mapping data set [CADOC, 2024]. It is processed by only selecting and then extracting classifications for G, S, L, P, and U, which correspond grazing land, farmland of statewide importance, local importance, prime farmland, and unique farmland. The prepared input is shown in Figure 16.

The photovoltaic suitability data set comes from the Argonne National Laboratory Geospatial Energy Mapper data set for utility-scale photovoltaic solar power suitability across the United States [Argonne National Laboratory, 2023].This data set is designed for strategic planning of energy systems in the United States, emphasizing renewable energy technologies. The exact specifics of the model are available through the tool [Argonne National Laboratory, 2023], but in summary, it considers the slope of the land, the land cover classification, the population density, the habitat, the land protection status, the solar energy potential, and the distance to substations in evaluating suitable areas for utility-scale photovoltaic development. Agrivoltaics, while not having specifically been deployed at a large scale for use case analysis, are likely going to be flexible in their connection to the grid, meaning some systems will be connected, and others may be less connected for local power generation. However, when considering the large-scale potential of agrivoltaic implementation, it is helpful

to consider them as a utility-scale installation. As such, this data set is useful. The data set is a GeoTIFF raster file by default and was converted to a vector shapefile of the agricultural regions of interest within California as shown in Figure 16.

The water stress dataset comes from the California Department of Water Resources Water Shortage Vulnerability Scoring and Tool data set [California Department of Water Resources, 2023b].]. This data set was developed under a legislative mandate to identify water supplies at risk of water stress and drought among small water suppliers and rural communities. This includes significant agricultural lands and gives direct insight into water stress across California. It considers physical and social vulnerabilities, impacts of climate change (such as temperature change, saltwater intrusion, and wildfire risk), hazardous events and conditions to water supplies, infrastructure, and other metrics relevant to California's water stresses. Agrivoltaics block sunlight, which directly reduces evapotranspiration in crops. Given California is water stressed, as is evident through this data set, it is reasonable to assume that higher levels of water stress and drought risk would incentivize agrivoltaic development as it would remove a small part of agricultural production (reducing water demand) and shade the remaining crop, also reducing water demand. This data set was provided as a shapefile and clipped to the relevant study as shown in Figure 16.

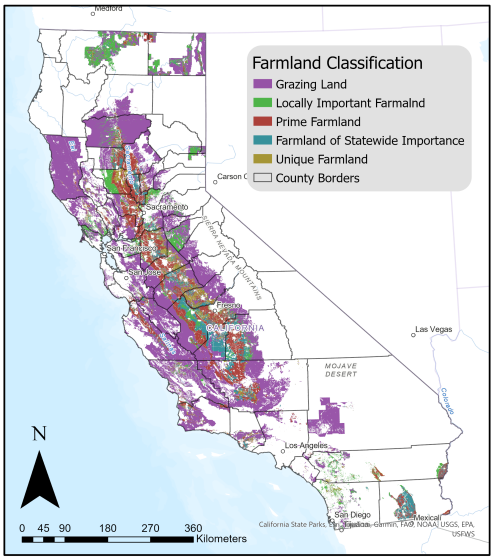
The solar power data sets come from the National Renewable Energy Laboratory Direct Normal Irradiance and Global Horizontal Irradiance data sets in the National Solar Radiation Database [Sengupta et al., 2018]. It is a datasets which show annual irradiance on a grid basis. This data set was selected because it provides a way to

calculate photovoltaic power production across California. The file was provided as shapefile and clipped to the study area as shown in Figure 16.

The evapotranspiration data set comes from the OpenET 2016-2021 Evapotranspiration database [Melton et al., 2021]. To evaluate the potential of agrivoltaics to reduce water demand in California, it will be helpful to know what the water demand of agriculture is. The OpenET data set provides high resolution actual evapotranspiration across the state using different methodologies for calculation over several years, meaning reasonable estimates of actual evapotranspiration can be calculated. The data set is a 30m raster grid which of a large portion of the United States, which was resampled to 500 m and clipped to California and reprojected using Google Earth Engine, then converted into a vector shape file for analysis, with the processed data used as an input being shown in Figure 17.

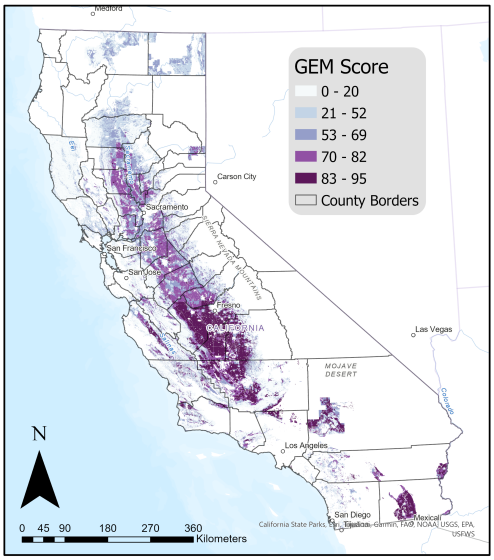
The groundwater depth data comes from the California Department of Water Resources Seasonal Groundwater Depth Points data set [California Department of Water Resources, 2023a]. This is a dataset designed to assist in understanding water levels throughout the state and contains data on depth to water levels in monitored wells in most parts of California, excluding the Imperial Valley at the time of writing. The data is provided as point data, which was then Kriged to create an interpolated raster map across the study area, and then converted to polygon data and clipped to the study area, as shown in Figure 17.

Input: Farmland Classifications



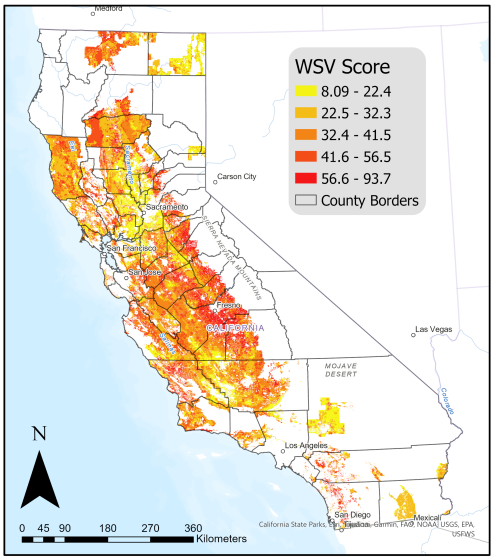
Projection: NAD 1983 California (Teale) Albers (Meters)
Data Sources Used in Analysis: CADWR Water Shortage Vulnerability Score, NREL Direct Normal Irradiance, ANL Photovoltaic Suitability Map, OpenET CONUS Ensemble Evapotranspiration Data, CADWR Groundwater Depth, CALFIRE County Boundaries, CADOC Important Farmland Categories

Input: Photovoltaic Suitability



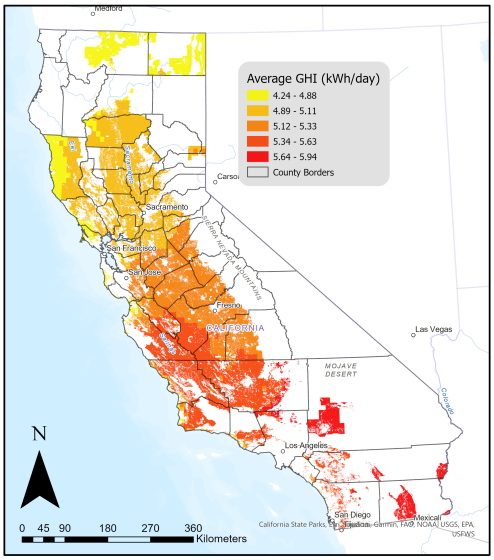
Projection: NAD 1983 California (Teale) Albers (Meters)
Data Sources Used in Analysis: CADWR Water Shortage Vulnerability Score, NREL Direct Normal Irradiance, ANL Photovoltaic Suitability Map, OpenET CONUS Ensemble Evapotranspiration Data, CADWR Groundwater Depth, CALFIRE County Boundaries, CADOC Important Farmland Categories

Input: Water Shortage Vulnerability Score



Projection: NAD 1983 California (Teale) Albers (Meters)
Data Sources Used in Analysis: CADWR Water Shortage Vulnerability Score, NREL Direct Normal Irradiance, ANL Photovoltaic Suitability Map, OpenET CONUS Ensemble Evapotranspiration Data, CADWR Groundwater Depth, CALFIRE County Boundaries, CADOC Important Farmland Categories

Input: Global Horizontal Irradiance



Projection: NAD 1983 California (Teale) Albers (Meters)
Data Sources Used in Analysis: CADWR Water Shortage Vulnerability Score, NREL Direct Normal Irradiance, ANL Photovoltaic Suitability Map, OpenET CONUS Ensemble Evapotranspiration Data, CADWR Groundwater Depth, CALFIRE County Boundaries, CADOC Important Farmland Categories

Figure 16: Input Data Sets - Part 1 (Crop Classifications, GEM Suitability, Water Shortage Vulnerability and Global Horizontal Irradiance)

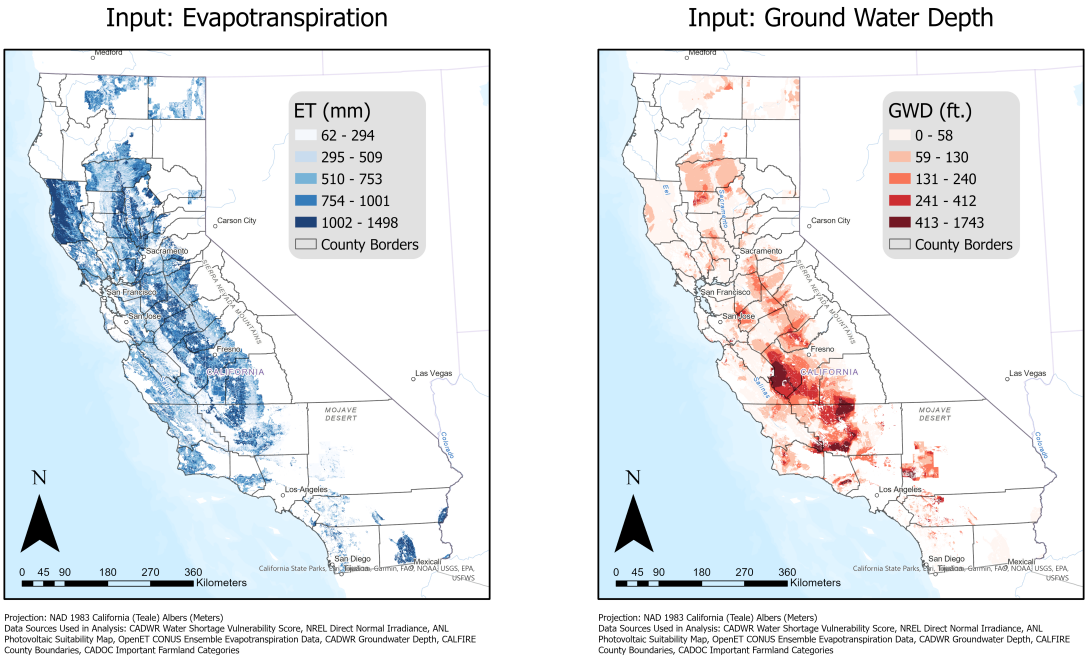


Figure 17: Input Data Sets - Part 2 (Evapotranspiration and Ground Water Depth)

GIS Analysis Methods

The suitability analysis was conducted in ArcGIS Pro 3.1.3 using built-in geoprocessing tools for the whole state of California. All data was projected and analyzed using the NAD 1983 California (Teale) Albers Projection due to the nature of the study area. Data sources,originally rasters, were converted to polygon shape files for analysis. The depth to groundwater table data was point data, that was interpolated using the default kriging settings in ArcGIS Pro. The resulting raster was converted to a vector (polygon) dataset and then clipped to the study area of California. The global horizontal irradiation data, evapotranspiration data, and pho-

tovoltaic suitability data were then clipped to the crop classification data set. Upon completion of these clips, spatial joins were conducted to link the newly clipped data to the individual fields. This data set was then clipped again by the water shortage vulnerability vector (converted from the raster data set), to produce an ultra-high resolution vector data set, which was then again spatially joined. Duplicates were searched for at each step and removed using the Duplicate Feature and Delete Identical tools. Validation was conducted at each step by analyzing the summary statistics and identifying unexpected and null results. If problems were identified, the steps were redone to ensure accuracy. Once all the data was accumulated in a single polygon layer, analysis was conducted by calculating fields using the modeling explained previously.

Results

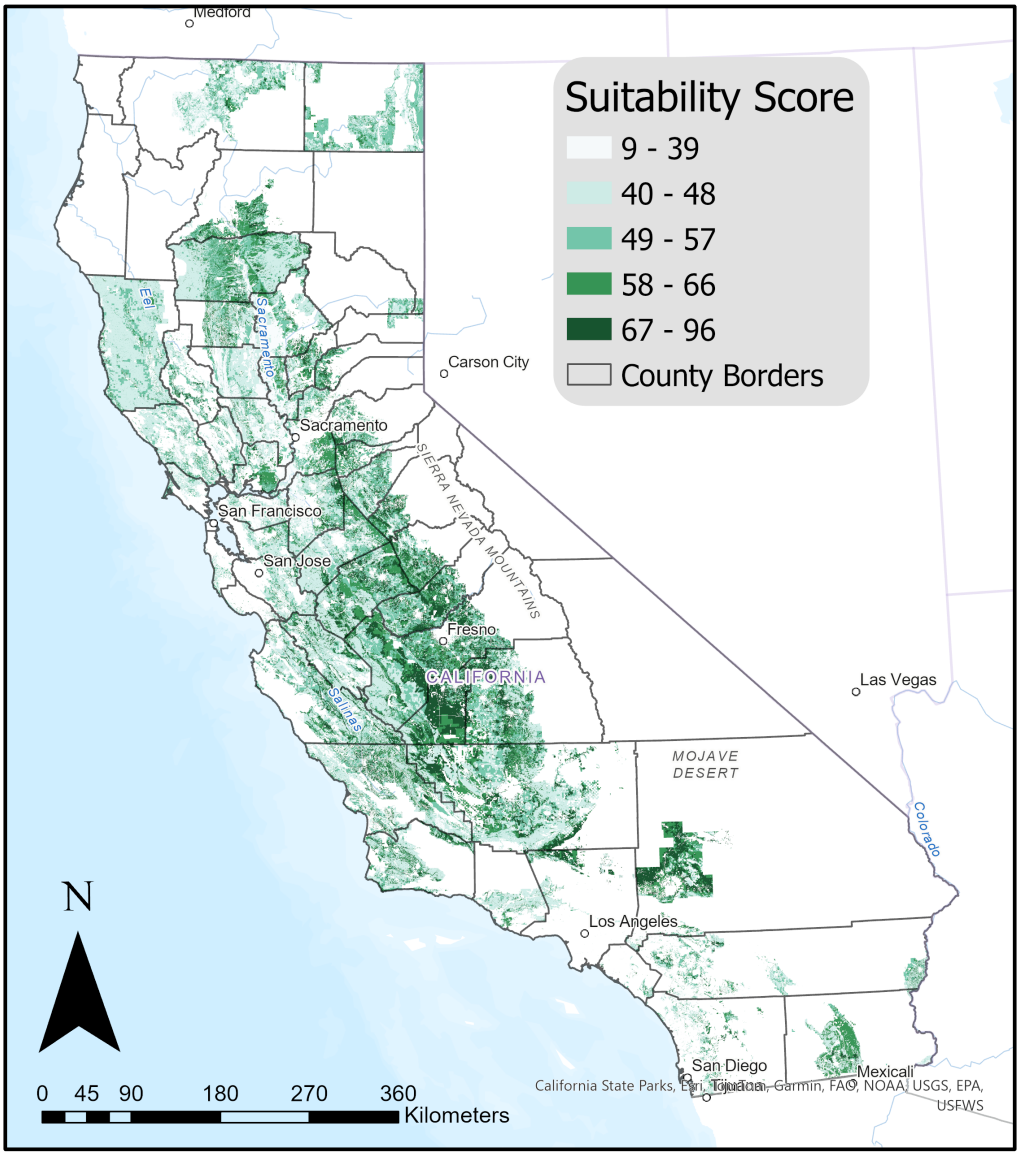
Across the state, this modeling shows a maximum county average agrivoltaic suitability score of 64.6 out of 100 in Kings County and a minimum average score of 44.3 in Colusa County. The analysis suggests an estimated 127,000 km^2 of farmland are suitable for agrivoltaic development, with almost 80,000 of those km^2 being grazing land (which is found to have a low average photovoltaic suitability score of 37, as well as an average total agrivoltaic suitability score of 57), and the remaining being of conventional farmland. Prime farmland and farmland of Statewide importance also both have high photovoltaic suitability of 71 and 74, respectively; however, prime farmland only has an average agrivoltaic suitability of 43 compared to the Statewide important farmland, which has an average agrivoltaic suitability

Table 15: Statewide Performance Statistics by Crop Classification

Classification	Suitability Score, Mean	GEM Score, Mean	Area	Power Production	Water Savings	Energy Savings	CO_2 Offset	Mean LUE
Units	-	-	km^2	GWh/yr	m^3/yr	GWh/yr	$t\ CO_2/yr$	$kWh/m^2/yr.$
Grazing Land	57	37	78695	1827118	3631914888	0	192	24
Farmland of Local Importance	49	50	12263	245514	203035	1124.00	16604682	2791
Prime Farmland	43	71	19952	380666	1176237091	1429	258004	1542
Farmland of Statewide Importance	64	74	10042	290367	208897	1157	19629060	2592
Unique Farmland	47	61	5678	116924	353193950	826	149078	8384
Total	52	59	126630	2860590	6518533687	4536	819208	3067

score of 64, which is an improvement over the grazing land. Stemming from this, there is an estimated 6,520,000,000 m^3/yr of water savings estimated from agrivoltaic shading across the entire state annually. Power production estimates to 2,860,000 GWh/yr, and associated reductions in irrigation energy of 4,500 GWh/yr There is an estimated maximum 193,000,000 $Mt\ CO_2/yr$ of carbon that could be offset by wide-scale agrivoltaic adoption. These values are summarized in Tables 15, 16 and 17.

California Agrivoltaic Suitability Map



Projection: NAD 1983 California (Teale) Albers (Meters)
Data Sources Used in Analysis: CADWR Water Shortage Vulnerability Score, NREL Direct Normal Irradiance, ANL Photovoltaic Suitability Map, OpenET CONUS Ensemble Evapotranspiration Data, CADWR Groundwater Depth, CALFIRE County Boundaries, CADOC Important Farmland Categories

Figure 18: Suitability Map for Agrivoltaics in California

Table 16: County Statistics (A-P)

County Name	Suitability Score, Mean	Acres	Power Production	Water Savings	Energy Savings	CO_2 Offset	Mean LUE
Units	-	km^2	GWh/yr	m^3/yr	GWh/yr	$ktCO_2/yr$	$kWh/m^2/yr$
Alameda	51.20	2,683.49	20,305.85	97,863,134	3.45	0.62	26.61
Amador	58.10	2,043.99	19,014.91	157,302,345	9.57	1.73	73.85
Butte	50.10	215.00	49,708.93	12,401,535	59.40	10.73	1,966.34
Calaveras	53.90	7,814.82	41,405.30	521,055,561	73.31	13.24	590.23
Colusa	44.30	884.97	39,226.72	55,932,626	109.68	19.80	2,904.75
Contra Costa	46.60	1,329.08	19,176.77	76,715,912	28.69	5.18	1,244.77
El Dorado	56.10	3,021.77	23,563.67	203,012,523	55.77	10.07	885.78
Fresno	59.20	8,670.27	218,313.03	498,689,982	506.51	91.47	2,635.85
Glenn	53.10	564.58	44,120.16	44,247,836	126.55	22.85	1,248.21
Humboldt	46.70	3,313.94	4.05	181,171,263	-	0.00	18.10
Imperial	53.90	1,611.43	55,376.61	82,783,362	137.17	24.77	578.83
Kern	56.50	1,148.29	268,059.03	59,531,030	251.31	45.40	581.65
Kings	64.60	791.36	92,608.11	51,238,240	151.01	27.27	2,249.19
Lake	54.60	608.50	24,765.21	29,979,529	30.19	5.45	1,304.52
Lassen	50.50	5,422.49	3,141.30	220,589,813	1.61	0.29	55.19
Los Angeles	56.80	3,226.67	29,831.41	197,805,319	7.36	1.33	3,951.59
Madera	60.10	1,032.27	78,705.41	74,593,125	221.71	40.04	1,518.74
Marin	44.50	1,263.87	10,157.39	60,950,535	23.65	4.27	394.23
Mariposa	61.80	1,911.12	41,025.44	110,862,967	0.21	0.04	26.73
Mendocino	48.30	4,603.63	145,490.21	194,838,628	17.69	3.20	264.97
Merced	57.10	10,915.22	112,192.16	470,457,637	285.66	51.58	2,842.85
Modoc	48.40	207.18	84,805.25	8,642,811	107.34	19.39	2,077.62
Monterey	53.80	4,392.25	118,896.03	161,046,615	109.58	19.80	1,138.06
Napa	47.40	2,305.00	20,050.62	126,430,655	51.80	9.36	3,800.85
Nevada	60.90	2,236.69	13,835.00	103,715,753	11.61	2.10	717.45
Orange	49.90	2,816.49	3,635.00	142,640,003	3.88	0.70	1,357.77
Placer	50.90	1,351.63	12,669.98	72,425,852	47.05	8.50	114.83
Plumas	49.30	154.91	8,148.63	8,116,472	14.11	2.55	407.85

Table 17: County Statistics (R-Y)

County Name	Suitability Score, Mean	Acres	Power Production	Water Savings	Energy Savings	CO_2 Offset	Mean LUE
Units	-	km^2	GWh/yr	m^3/yr	GWh/yr	$ktCO_2/yr$	$kWh/m^2/yr$
Riverside	51.70	2,096.39	50,365.25	97,791,806	172.69	31.19	1,584.07
Sacramento	51.60	1,407.52	30,440.41	73,993,254	77.31	13.96	609.22
San Benito	53.80	4,586.28	60,114.45	260,554,619	19.74	3.57	71.24
San Bernardino	60.70	1,019.43	108,285.24	43,430,961	8.94	1.63	861.61
San Diego	50.90	2,947.25	30,482.98	166,341,299	130.68	23.60	931.97
San Joaquin	49.40	1,646.04	59,902.46	81,607,909	203.23	36.70	1,729.10
San Luis Obispo	56.40	1,153.27	129,733.12	41,253,307	97.04	17.54	1,537.40
San Mateo	46.50	7,100.60	3,853.14	350,597,275	4.74	0.86	50.30
Santa Barbara	53.60	1,019.62	65,349.50	50,110,964	72.49	13.10	2,503.76
Santa Clara	51.70	402.48	34,716.79	15,787,833	22.05	3.98	963.39
Santa Cruz	44.40	1,985.61	2,809.46	92,527,879	14.01	2.53	14,692.54
Shasta	58.30	0.22	40,852.27	16,018	20.49	3.70	179.18
Sierra	49.20	1,733.48	4,255.46	107,090,777	8.21	1.48	73.10
Siskiyou	47.40	1,423.32	82,589.01	75,377,543	177.64	32.08	4,132.28
Solano	50.90	3,667.50	29,505.08	69,425,940	40.04	7.23	904.11
Sonoma	46.30	0.64	43,538.74	37,596	97.77	17.66	1,617.96
Stanislaus	51.70	618.08	72,284.59	31,868,487	206.92	37.37	4,010.08
Sutter	46.00	1,050.06	25,142.69	65,181,039	69.48	12.55	497.60
Tehama	53.60	5,192.48	150,956.61	230,800,826	130.87	23.64	198.93
Trinity	45.80	-	11.63	-	-	0.00	18.28
Tulare	58.60	2,088.52	128,852.79	107,580,909	324.01	58.51	1,152.93
Tuolumne	62.90	2,551.38	26,913.16	149,352,459	0.41	0.08	27.13
Ventura	46.30	5,098.52	25,962.96	328,655,543	77.61	14.02	1,564.05
Yolo	46.30	7,100.60	36,229.29	350,597,275	82.36	14.87	191.20
Yuba	53.10	164.62	19,211.18	10,407,731	31.26	5.65	131.35
Total	53.60	147.68	2,860,590.44	3,656,333	4,535.83	819.21	198.93

Since agricultural production is geographically distributed, looking at agrivoltaic production in smaller geographic regions can be helpful. Comparative analysis across

counties was conducted, and top counties by different metrics were identified. With respect to average suitability by county (as shown in Figure 19), the 5 most suitable counties in order are Kings, Tuolumne, Mariposa, Nevada, and San Bernardino counties with county-wide average agrivoltaic suitability scores of 64.6, 62.9, 61.9, 60.9 and 60.7, respectively. The least suitable counties are Colusa, Santa Cruz, Marin, Trinity, and Sutter counties, with scores of 44.3, 44.4, 44.5, 45.8, and 46.0 respectively.

Average suitability scores do not give a full picture, so other analyses are conducted. Land use efficiency (as calculated in equation (47), and shown in Figure 20) is a metric that highlights the power production and irrigation energy offset on an annual per area basis for easy comparison. The five most land-use efficient counties per this analysis are: Santa Cruz, Siskiyou, Stanislaus, Los Angeles, and Napa counties, with land-use efficiency scores of 14,692, 4,132, 4,010, 3,951, and 3,800 $kWh/m^2/yr$, and the least efficient counties are Humboldt, Trinity, Alameda, Mariposa, and Tuolumne counties, with values of 18.1, 18.3, 26.6, 26.7, and 27.1 $kWh/m^2/yr$, respectively.

Looking at absolute metrics also helps understand the potential for agrivoltaics. With respect to power production (see Figure 21), the top five counties are Kern, Fresno, Tehama, Mendocino and San Luis Obispo counties with maximum photovoltaic power production estimates of 268,059, 218,313, 150,956, 145,490 and 129,733 GWh/yr, while the five least agrivoltaic power producing counties are Humboldt, Trinity, Santa Cruz, Lassen and Orange counties with maximum estimates of 4.05, 11.6, 2,809, 3,141 and 3,635 GWh/yr, respectively.

Water savings is another absolute metric of critical importance to agricultural production. Water savings per county was calculated (Figure 22) and the top five water saving counties are Mendocino, Fresno, Kern, Tehama, and Tulare counties with maximum estimated savings of 521,055,561, 498,689,982, 470,457,637, 350,597,275 and 328,655,542 m^3/yr and the counties with the least estimated water savings are Humboldt, Trinity, Lassen, Santa Cruz, and Sierra counties with water savings estimates of 16,018, 37,595, 3,656,332, 8,116,471, and 8,642,811 m^3/yr, respectively. Associated with this is irrigation energy savings, where the top five irrigation energy saving counties are: Fresno, Tulare, Merced, Kern, and Madera counties with estimated maximum energy savings of 506, 324, 285, 251, and 221 GWh/yr while the counties with the least estimated savings are: Humboldt, Trinity, Mariposa, Tuolumne, and Lassen counties with estimates of 0.00, 0.00, 0.21, 0.41 and 1.61 GWh/yr, respectively.

Lastly, CO_2 offsets are ranked on a county basis, which could theoretically be nebulous as energy on the grid is not tracked on an electron basis, and knowing which electrons are photovoltaically produced and then theoretically offset energy sources that were produced carbon-based is challenging. However, from a carbon accounting perspective, it may be helpful to to analyze which counties are offsetting CO_2. The top 5 counties with CO2 offsets are: Kern, Fresno, Tehama, Mendocino, and San Luis Obispo counties with maximum estimated offsets of 268,059, 218,313, 150,956, 145,490, and 129,733 $Mt\ CO_2/yr$, while the 5 counties producing the least offsets are: Humboldt, Trinity, Santa Cruz, Lassen and Orange counties with maximum offset estimates of 0.00, 0.00, 0.19, 0.21 and 0.25 $Mt\ CO_2/yr$, respectively.

Discussion and Conclusion

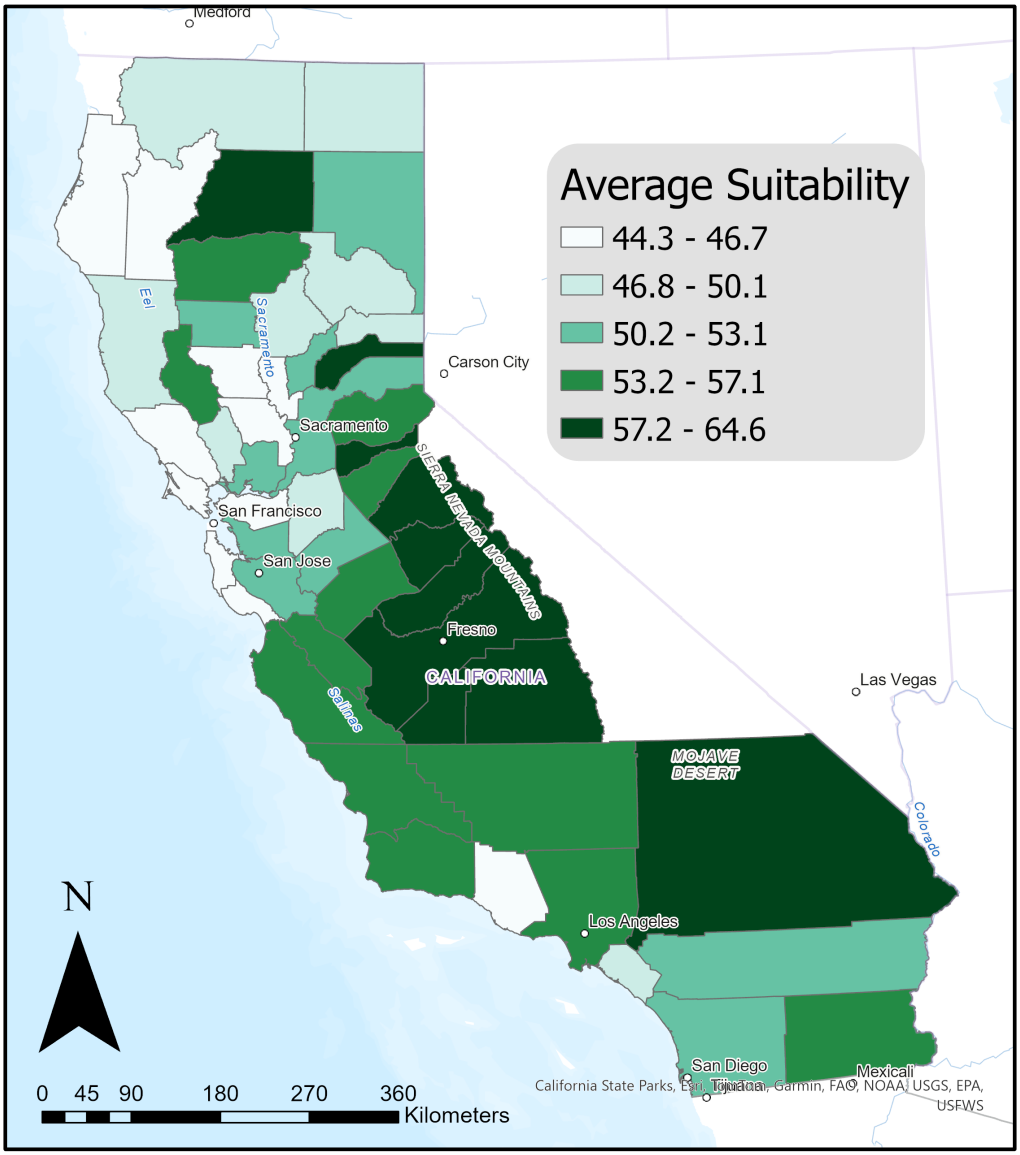
While results represent theoretical maximums, California is very unlikely to transition to 100% agrivoltaic adoption, and certainly not overnight. For this discussion, a 1% adoption rate was used as this is not an impossible goal, even if it is ambitious given the current state of agrivoltaics. With respect to how much land would be transitioned, a 1% adoption rate suggests up to 1,300 km^2 of farmland would need to be converted for agrivoltaic production. This analysis also estimates an average annual agricultural water demand reduction of 65,200,000 m^3 at 1% adoption. These values are not particularly large, but in this context but represent significant investments and expansions of photovoltaic development.

California energy generation in 2022 was 287,000 GWh for the year, with approximately 40,500 GWh of solar energy produced [California Energy Commission, 2022]. This analysis estimates a potential for agrivoltaic energy generation of 29,000 $GWh/yr.$ and irrigation energy reductions of 45 $GWh/yr.$. When considering the potential energy reductions, this represents nearly a 10% increase in California's energy generation and almost 60% increase in California's solar energy production. Associated with this is the CO_2 offsets, and California generated 369 million metric tons of CO_2 equivalents [California Air Resources Board, 2022b]. This analysis shows agrivoltaics have a potential to reduce CO_2 emissions by 819 $t\ CO_2/yr$, representing a theoretical reduction of California's annual CO_2 emissions by less than 1%.

Ultimately, how important the scale of these metrics is in the big picture is likely a matter of subjective interpretation to policymakers who may consider how aggres-

sively to push for wide-scale implementation of agrivoltaic development. However, for parties interested in attempting to independently expand agrivoltaic deployments, either for their farm or for finding farmers, this analysis highlights locations across the state that may be more suitable. This all being said, agrivoltaics have the potential to be hugely transformative to California's agricultural landscapes and result in significant power production and water use reductions with minimal impact on agriculture.

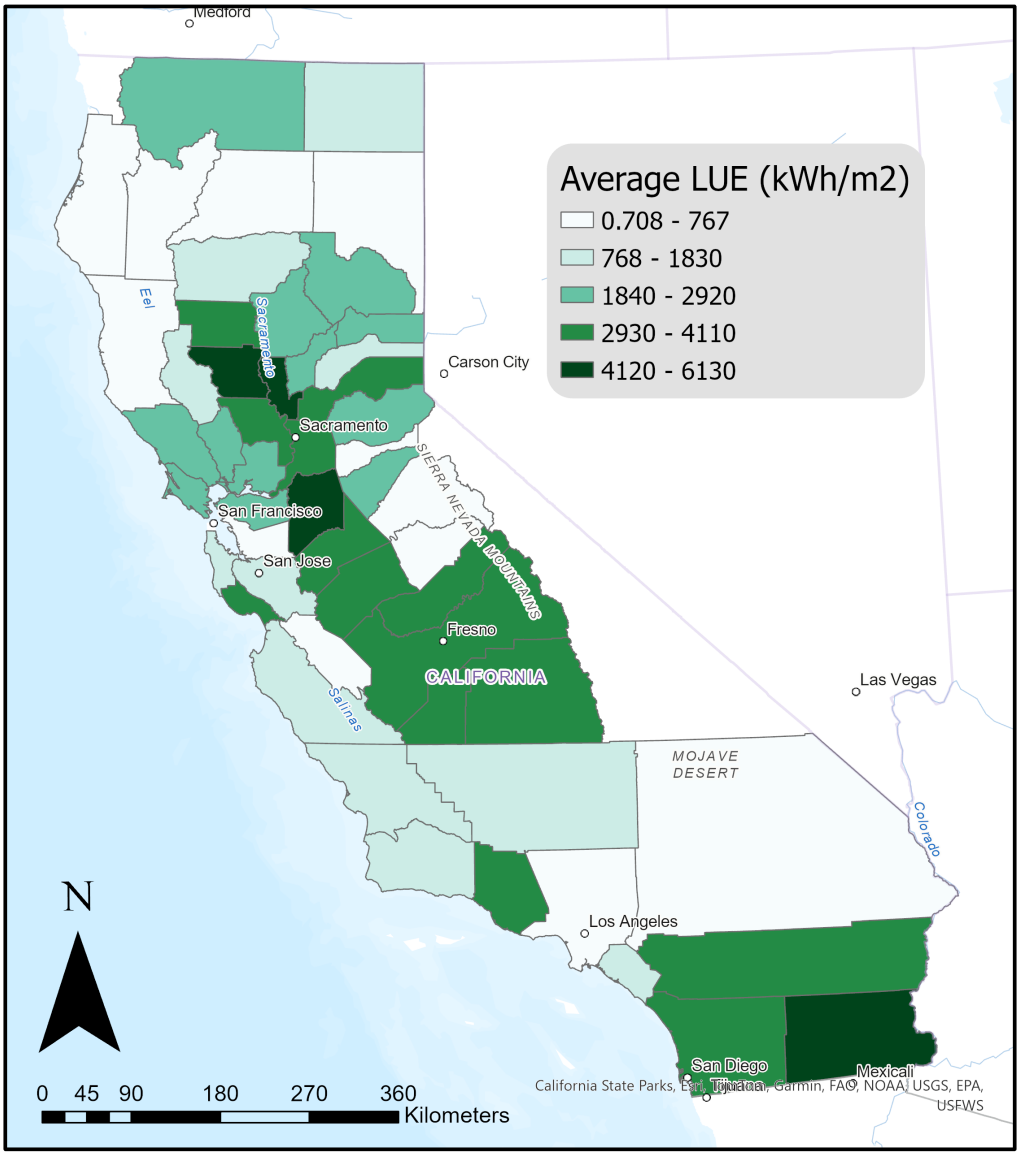
County Average Suitability Score



Projection: NAD 1983 California (Teale) Albers (Meters)
Data Sources Used in Analysis: CADWR Water Shortage Vulnerability Score, NREL Direct Normal Irradiance, ANL Photovoltaic Suitability Map, OpenET CONUS Ensemble Evapotranspiration Data, CADWR Groundwater Depth, CALFIRE County Boundaries, CADOC Important Farmland Categories

Figure 19: County Average Suitability Map

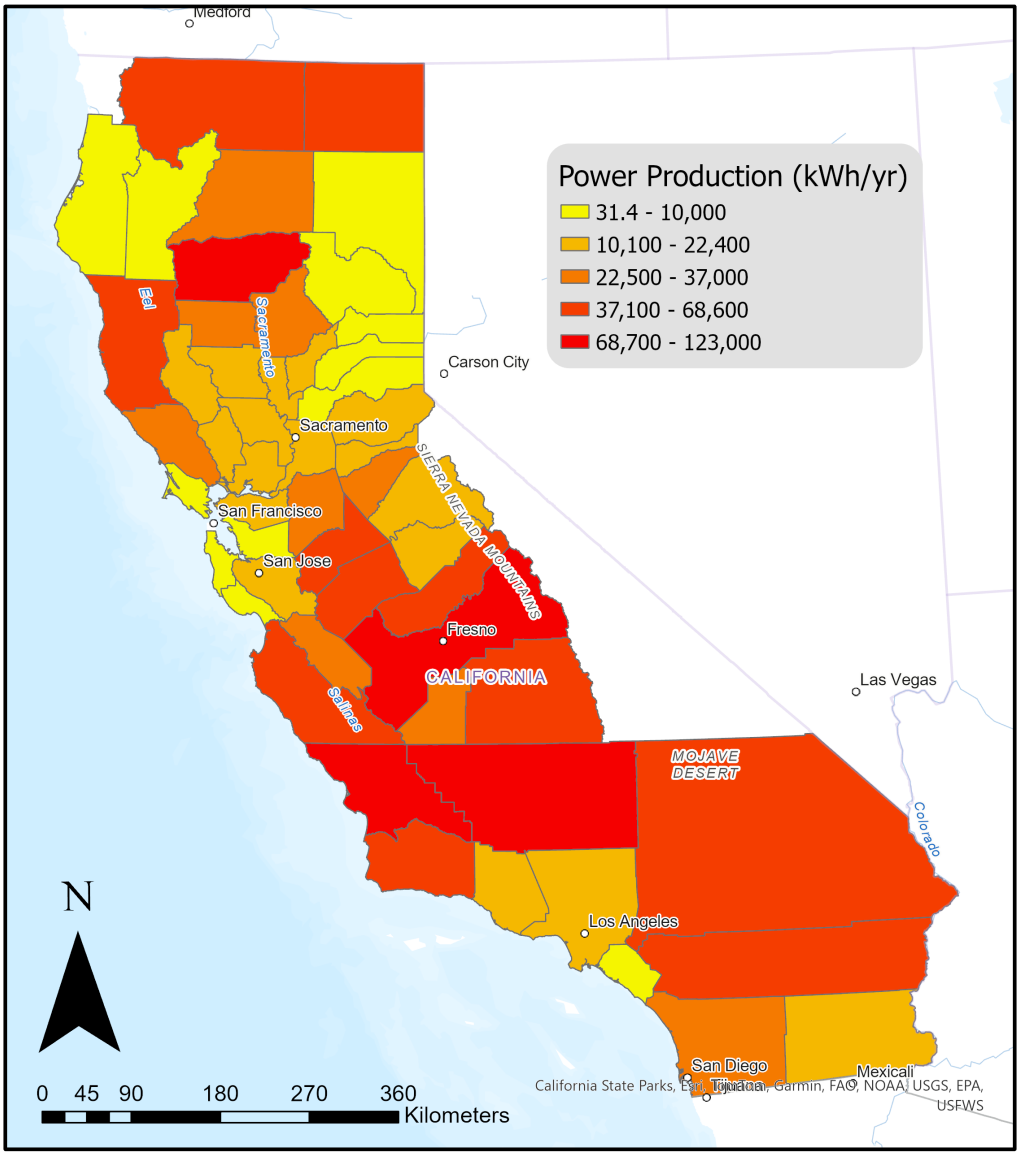
County Average Land Use Efficiency



Projection: NAD 1983 California (Teale) Albers (Meters)
Data Sources Used in Analysis: CADWR Water Shortage Vulnerability Score, NREL Direct Normal Irradiance, ANL Photovoltaic Suitability Map, OpenET CONUS Ensemble Evapotranspiration Data, CADWR Groundwater Depth, CALFIRE County Boundaries, CADOC Important Farmland Categories

Figure 20: County Land Use Efficiency Map

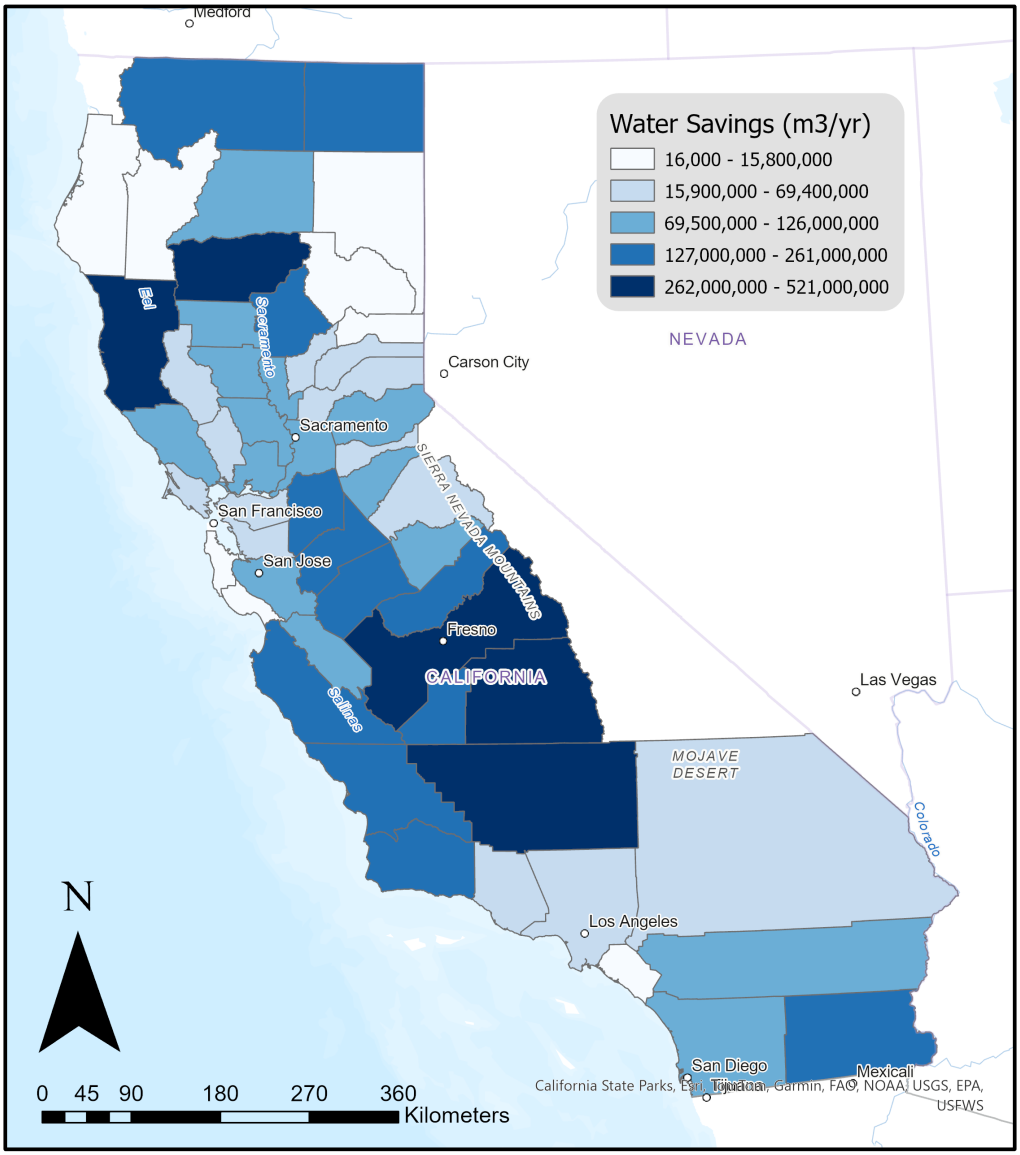
County Probable Annual Power Production



Projection: NAD 1983 California (Teale) Albers (Meters)
Data Sources Used in Analysis: CADWR Water Shortage Vulnerability Score, NREL Direct Normal Irradiance, ANL Photovoltaic Suitability Map, OpenET CONUS Ensemble Evapotranspiration Data, CADWR Groundwater Depth, CALFIRE County Boundaries, CADOC Important Farmland Categories

Figure 21: County Power Production Map

County Probable Water Savings



Projection: NAD 1983 California (Teale) Albers (Meters)
Data Sources Used in Analysis: CADWR Water Shortage Vulnerability Score, NREL Direct Normal Irradiance, ANL Photovoltaic Suitability Map, OpenET CONUS Ensemble Evapotranspiration Data, CADWR Groundwater Depth, CALFIRE County Boundaries, CADOC Important Farmland Categories

Figure 22: County Water Savings Map

Conclusions

Summary

This research has conducted a comprehensive agrivoltaic suitablility analysis. A multi-year field experiment was conducted to explore the impacts of agrivoltaic production on crop performance for processing tomatoes. The experimental data was then leveraged to create a model that simulated agrivoltaic crop performance and subsequently analyzed to optimize agrivoltaic designs for maximum solar energy production and minimum yield loss. The combined lessons learned from the field experiment and modeling results were used to conduct a state-wide GIS-based analysis to determine suitable locations and potential benefits of adopting agrivoltaic systems throughout California.

Chapter 1 Findings

Chapter 1 aimed to understand the impacts of agrivoltaics on processing tomato production. The key findings are as follows:

1. Agrivoltaic plants will have an improved photosynthetic carbon assimilation rate compared to their control relating to an improved microclimate. However, they will have a lower daily net photosynthesis compared to the control due to increased shading, which will correspond to lower yields. Modeling will suggest that agrivoltaic crops grow differently due to the presence of the photovoltaic panels. Agrivoltaic plants showed statistically significantly improved photosynthetic rates at light levels representing sunlight and shaded conditions. The

large reduction in absorbed solar radiation contributed to lower net photosynthetic rates and reduced yields

2. The agrivoltaic yields were reduced due to reduced light; however, measured crop quality was minimally impacted. The modeled impacts on crop performance were inconclusive at this time. The growing degree day calculations suggest the 2022 crop showed different results compared to, the 2021 crop, leading to an assessment that the impact of agrivoltaic production may be enough to negatively impact conventional crop models, but not as significantly as to prevent work on the agrivoltaic model developed in Chapter 2

These results were surprising but not completely unexpected. Plants, like all living things, adapt to their environments so to see them adapting to agrivoltaic circumstances makes perfect sense. And while significantly more work needs to be done to further understand how crops behave when grown in agrivoltaic systems, both generally and under specific scenarios, powerful insights were gleaned through these experiments. The impacts on photosynthetic rates under full light and shaded light levels was particularly interesting as it seems to suggest the plant is adapting to the light conditions of the agrivoltaic system. More work will need to be focused determining the specific realities of phenological development of crops in agrivoltaic production. How significant is plant adaptation in agrivoltaic systems? It is unclear and needs to be studied.

Chapter 2 Findings

In Chapter 2, the goal was to create a modeling framework that models agrivoltaic performance at an hourly time-step, moving past a constraint of conventional crop models, which typically account for daily light integrals rather than accounting for the nuances associated with shifting light conditions that occurs during the day in agrivoltaic systems. Leveraging this framework allowed genomic optimization of agrivoltaic designs. Agrivoltaic systems are complex, and there not enough information to determine how changing individual parameters could impact crop performance. The findings of the study are described concisely as follows:

1. A digital replica model was produced, which was capable of simulating light distribution in an agrivoltaic system that was coupled to a crop growth capable of simulating plant growth and yield at an hourly time-step throughout the season

2. The genomic algorithm-based optimization strategy was found to be effective at rapidly identifying good agrivoltaic designs while also outputting useful information to understand how the design impacts crop performance

The model framework proposed here shows potential but also an opportunity for improvement in both the 3D crop modeling as well as agrivoltaic modeling. The genetic algorithm used in the framework for optimization the computationally expensive nature. Of particular concern is the high computational cost of ray tracing conducted by the Helios software for every hour of every day across tens to hundreds of thousands of potential leaf segments via CUDA. While a powerful tool, it

is hard to run without expensive computer hardware. Nevertheless, the developed framework allows deterministic modeling of crop performance with stochastic optimization, which facilitates finding designs in numeric spaces where gradient-based optimizations are not traditionally possible.

Chapter 3 Findings

The goal of Chapter 3 was to use some of the insights gained from Chapters 1 and 2 to develop a suitability framework to explore the wider potential of agrivoltaic production in California. While findings from the previous chapters are not directly utilized here, the broad insights served to formulate an approach. The GIS framework for the potential siting of agrivoltaic systems in California determined a variety of locations in Central and Southern California as viable agrivoltaic sites. Calculations demonstrate that agrivoltaic systems have meaningful potential to provide environmental benefits relating to power production, water savings, energy savings, and carbon offset in California. Main key takeaway messages of this work include:

1. A geospatial analysis framework for the potential siting of agrivoltaic systems in California was developed and produced a map which assists in determining the suitability of agrivoltaic development across the entire state

2. Calculations demonstrated that agrivoltaic systems have meaningful potential to improve land use efficiency, generate power, reduce water demand, reduce irrigation energy demands, and offset carbon dioxide emissions

This framework found a large potential for land adoption, power production,

water savings, and CO2 offsets. While 100% agrivoltaic adoption is not likely, even an ambitious 1% adoption through the State of California presents an opportunity to significantly impact California's agricultural and energy landscapes.

Future Work

Future work can be summarized broadly through more refinement. Agrivoltaic production is a new field, and there are so many opportunities to learn things. In terms of priorities, determining the impacts of agrivoltaic systems on crop development is likely a very high priority. It is likely not feasible to wait for every crop and every agrivoltaic design to be tested before determining how to design agrivoltaic systems. Studies that seek to understand how crops adapt will help provide meaningful insights to make informed decisions on designing agrivoltaic systems for crop production. These sorts of studies will also help expand crop modeling capabilities.

Currently, crop models are generally done as single point models assuming large, uniform fields and done at a daily time-step as that is the most commonly available data type for practical applications. However, agrivoltaics are complex systems that challenge assumptions relating to the dynamics of light distribution assumed in most crop models. The nature of shading in agrivoltaic systems is deeply complex, and more work needs to be done to understand these impacts.

The 3-dimensional crop modeling work here is also a foundation but not the definitive process for conducting agrivoltaic crop modeling. It serves as an interesting base for generating deterministic crop models, but given that 3D crop modeling is deeply complex and also a developing field, more work will have to be done to refine

the model. In particular, there are a lot of dynamics relating to how the leaves are distributed and along the plant or stems, their leaf size, shape, etc., which are not modeled here. Considerations of specific growth stages and other processes like water and heat stress are also not modeled in the presented version. These are all opportunities to continue refining 3D crop modeling and agrivoltaic performance, which could open the door to optimizing not just the panel designs but also the crop orientation, spacing, irrigation, etc.

With respect to agrivoltaic engineering and design, more optimization methods should be explored. While the approach taken here was very powerful, it also required powerful hardware to run and is extremely computationally expensive. Even running simulations on a $10,000 computer and maxing out performance through multi-threading, simulations routinely took between 20 and 30 hours. Work must be done to find more efficient optimization models for the agrivoltaic space. There's also the opportunity to integrate economic modeling and study other facets of agrivoltaic performance beyond the scientific performance of crop production and electricity generation.

Lastly, the geospatial modeling was very interesting, but also just one potential approach that makes broad agrivoltaic performance assumptions. The model presented here was very broad to cover a large study area (the entire State and all of Californian agriculture) and conduct a preliminary scoping, and there still remains the opportunity to refine the analysis. This sort of large-scale modeling is largely speculative at this stage, but integrating crop-specific designs alongside specific crop maps of geographies, or looking at surveys to possibly predict adoption rates, or

maybe even considering local regulations could all be future projects to integrate into geospatial analyses for agrivoltaic adoption.

Conclusion

Agrivoltaic systems provide an interesting opportunity for the future of sustainable agriculture and renewable energy generation. The potential to reduce water demand with minimal impact on crop performance and enhance crop performance in some circumstances through blending of photovoltaic technology with climate-smart farming is high if the technology is properly developed.

To accomplish developing agrivoltaic technologies, this dissertation has studied crop performance, developed agrivoltaic models, and then explored the benefits to the whole state of California. We have found impacts on photosynthesis to be beneficial to crop development and worthy of further study. This research has also shown that agrivoltaic models can benefit from hourly light modeling to capture the nuances of crop development under dynamic shading and that genomic optimization provides a pathway to complex optimizations. And lastly, geospatial mapping was shown to highlight suitable areas for agrivoltaic development while also facilitating calculations for projecting the potential benefits of wide scale agrivoltaic adoption.

Closing Statements

This work seemed impossible several years ago, but through hard work and a lot of help, I was able to accomplish all this. I set out to do a meaningful Ph.D., and now I think I've helped contribute to a new field of agriculture and engineering: agrivoltaic

engineering. I've learned so much along the way, and now have done so much with that. It is my hope that this work can be further developed by me and by others also interested in building a better future.

In the name of Allah, the Most Gracious, the Most Merciful. All praise be to Allah. Thank you for reading the last 7 years of my life, and I wish you great benefits from this work.

Appendix

Chapter 1

2021 Harvest Data

Table 18: 2021 Harvest Data

Treatment	Fresh Weight (4 Plants)	Fresh Weight (2 Plant Sample)	Fruit Sample	Sample Area	Yield	pH	Color	Brix
Units	kg	kg	kg	ft^2	$\frac{tons}{acre}$	-	-	-
C1	55.86	29.57	22.44	35.00	27.93	4.53	20.50	4.80
C2	47.13	21.51	17.74	20.00	38.64	4.54	20.50	5.00
C3	76.26	39.96	33.03	30.00	47.96	4.47	21.00	5.00
C4	67.82	29.06	18.92	35.00	23.55	4.48	21.00	4.60
AV1	26.05	13.97	9.70	30.00	14.08	4.54	21.00	4.80
AV2	24.76	5.68	3.62	35.00	4.50	4.51	21.00	4.80
AV3	26.86	12.04	6.57	40.00	7.16	4.50	21.00	4.80
AV4	41.98	8.10	5.42	40.00	5.91	4.53	21.50	4.50
AV1-2		10.86	8.35	10.00	36.38			
AV2-2		12.47	10.58	10.00	46.10			
AV3-2		7.22	5.32	10.00	23.17			
AV4-2		15.62	12.20	10.00	53.13			

2022 Harvest Data

Table 19: 2022 Harvest Data

Treatment	Total Biomass	Fruit Biomass	Sample Area	Yield	pH	Color	Brix
Units	kg	kg	ft^2	$\frac{tons}{acre}$	-	-	-
C1	147.16	106.48	50.00	92.77	4.50	22.00	4.40
C2	103.28	74.62	50.00	65.01	4.45	21.00	4.30
C3	89.36	73.04	50.00	63.63	4.42	21.00	4.90
C4	53.70	31.44	50.00	27.39	4.60	25.50	4.20
AV1	31.50	25.16	50.00	21.92	4.44	23.00	3.40
AV2	17.72	16.70	50.00	14.55	4.42	21.50	4.20
AV3	36.26	28.76	50.00	25.06	4.60	21.50	4.70
AV4	28.68	25.95	50.00	22.61	4.51	22.50	3.80

2021 Photosynthesis Data

Table 20: 2021 Light Response Measurements

Control											
Light Level	7/24/2021	7/24/2021	7/24/2021	7/24/2021	8/10/2021	8/10/2021	8/10/2021	8/10/2021	8/23/2021	8/23/2021	8/23/2021
2000.00	20.75	14.94	21.32	21.17	19.87	17.39	23.66	31.73	21.02	15.40	26.94
1500.00	26.70	23.59	23.35	23.21	14.77	18.08	23.96	31.01	22.63	16.03	24.76
1200.00	25.27	25.85	20.86	21.57	13.42	16.73	23.59	29.25	21.61	15.49	21.37
900.00	20.57	21.72	18.90	20.37	12.87	15.08	21.65	26.03	19.55	14.52	17.22
600.00	16.37	17.69	16.07	17.77	11.61	26.62	18.14	20.15	16.01	12.99	12.03
300.00	8.98	9.92	8.47	10.32	6.98	7.93	10.40	9.62	8.98	8.74	5.99
150.00	3.84	4.35	3.74	4.76	2.87	3.18	9.95	3.34	3.10	4.13	1.30
50.00	-0.27	-0.04	-0.43	0.16	-1.59	-1.26	-0.62	-1.66	-1.16	-0.19	-2.89
0.00	-2.14	-2.26	-2.81	-2.07	-3.93	-3.91	-3.07	-4.18	-3.36	-2.73	-5.16
Agrivoltaic											
Light Level	7/24/2021	7/24/2021	7/24/2021	7/24/2021	8/10/2021	8/10/2021	8/10/2021	8/10/2021	8/23/2021	8/23/2021	8/23/2021
2000.00	19.99	32.03	25.33	19.13	21.54	27.57	16.61	31.88	21.25	26.72	16.89
1500.00	26.01	31.36	26.58	25.36	20.59	27.09	18.39	30.96	21.68	13.20	15.99
1200.00	27.50	29.77	23.63	24.73	18.79	26.18	17.35	30.02	22.28	12.71	14.70
900.00	22.46	25.14	22.54	22.95	15.95	23.56	14.76	26.55	21.54	11.94	13.21
600.00	19.92	20.30	19.27	18.76	13.49	35.79	11.67	20.65	17.94	10.60	11.04
300.00	10.48	11.22	11.51	10.62	8.00	10.28	8.24	10.24	10.54	6.80	6.55
150.00	5.65	5.04	5.38	4.97	3.30	3.83	4.09	8.77	4.47	3.16	2.70
50.00	0.59	0.48	0.72	0.49	-1.43	-1.01	0.98	-1.04	0.13	-0.58	-1.51
0.00	-1.67	-2.09	-1.67	-1.70	-3.87	-3.40	-2.27	-3.60	-2.24	-2.69	-3.97

2022 Photosynthesis Data

Table 21: 2022 Light Response Measurements

Control												
Light Level	7/4/2022	7/4/2022	7/4/2022	7/4/2022	7/11/2022	7/11/2022	7/11/2022	7/11/2022	7/18/2022	7/18/2022	7/18/2022	7/18/2022
2000.00	57.50	30.67	23.61	29.50	28.15	30.91	36.30	26.14	28.15	30.80	25.52	23.73
1500.00	28.24	27.57	23.47	27.11	28.92	30.94	33.91	28.48	26.42	27.85	23.74	22.95
1200.00	25.98	23.61	20.32	23.48	28.52	30.00	31.10	54.59	22.82	24.26	21.21	21.44
900.00	21.35	18.35	15.39	17.62	26.22	26.77	26.39	22.85	18.80	20.51	18.70	19.27
600.00	15.19	12.57	10.77	22.49	21.25	20.61	19.39	17.96	14.23	15.93	15.93	15.87
300.00	6.50	5.34	4.91	4.34	11.50	21.06	9.10	9.50	8.05	9.15	9.64	8.95
150.00	1.10	1.09	1.22	-0.04	4.77	4.19	2.92	3.62	3.87	3.97	4.71	3.77
50.00	-3.29	-3.92	-2.78	-4.29	-0.76	-0.85	-1.95	-1.55	-0.58	-0.84	-0.47	-1.16
0.00	-5.52	-6.09	-5.77	-6.85	-3.11	-3.29	-4.38	-4.34	-3.31	-3.88	-3.03	-3.77
Agrivoltaic												
Light Level	7/4/2022	7/4/2022	7/4/2022	7/4/2022	7/11/2022	7/11/2022	7/11/2022	7/11/2022	7/18/2022	7/18/2022	7/18/2022	7/18/2022
2000.00	30.62	29.89	32.29	31.35	33.54	34.45	33.04	32.39	20.08	19.13	31.37	33.27
1500.00	29.18	28.15	59.45	29.29	32.50	33.52	32.39	31.17	19.97	18.33	27.91	30.77
1200.00	26.86	25.84	25.05	26.37	31.17	31.53	30.50	29.37	18.65	17.41	24.18	28.74
900.00	48.01	22.17	19.27	22.64	58.67	27.58	27.00	25.88	17.18	16.08	21.20	25.72
600.00	17.03	17.05	14.44	17.54	22.66	20.72	20.77	39.73	15.36	14.25	16.97	20.44
300.00	7.81	8.39	6.63	8.37	11.93	10.53	10.61	10.83	10.35	9.51	9.62	11.02
150.00	1.93	2.38	1.72	2.30	4.99	4.11	4.05	4.73	5.44	5.01	4.35	4.87
50.00	-2.79	-2.26	-2.59	-2.41	-0.25	-1.00	-1.17	-0.27	0.77	0.65	-0.82	-0.42
0.00	-5.14	-10.03	-5.14	-5.04	-2.88	-3.65	-3.93	-3.03	-1.78	-2.31	-3.98	-3.45

2021 Canopy Cover Data

Table 22: 2021 Canopy Cover Measurements - Part 1

Date	C1	C2	C3	C4	AV1	AV2	AV3	AV4
5/24/2021	3.00	10.50	7.60	4.70	2.10	11.90	14.50	8.50
5/27/2021	5.70	15.70	11.60	7.90	1.40	19.50	23.60	13.60
6/1/2021	11.70	29.00	28.70	16.80	7.30	30.70	24.30	26.30
6/3/2021	13.70	36.80	29.00	24.60	10.60	37.90	30.70	34.30
6/7/2021	23.90	45.00	41.40	28.00	19.00	49.50	44.10	50.20
6/10/2021	32.30	52.90	51.60	52.30	23.20	53.00	47.60	57.10
6/14/2021	47.30	66.10	57.40	45.10	36.90	64.50	67.10	73.70
6/17/2021	47.20	73.40	69.30	67.70	41.20	69.70	70.60	77.30
6/21/2021	57.30	84.10	79.60	77.00	55.90	81.70	84.40	88.10
6/24/2021	58.10	89.00	67.20	77.50	64.60	84.80	87.60	90.50
6/29/2021	81.10	90.50	92.20	84.50	80.30	88.30	94.50	94.30
7/1/2021	74.70	82.60	91.70	88.80	82.20	90.00	93.60	94.30
7/6/2021	74.70	82.60	91.70	88.80	82.20	90.00	93.60	94.30
7/8/2021	85.50	92.50	94.30	90.60	93.50	93.00	93.40	93.10

Table 23: 2021 Canopy Cover Measurements - Part 2

Date	C1	C2	C3	C4	AV1	AV2	AV3	AV4
7/13/2021	92.40	80.70	90.30	89.30	94.30	92.70	91.50	92.20
7/15/2021	90.80	89.90	90.60	90.80	94.20	90.80	89.90	93.40
7/19/2021	92.80	89.00	87.30	94.40	97.10	91.70	91.20	94.30
7/21/2021	92.70	87.50	92.00	90.00	94.70	90.70	90.50	92.20
7/26/2021	93.80	84.10	89.80	87.90	97.30	87.90	86.40	88.20
7/30/2021	90.80	80.10	86.20	89.40	95.10	90.30	89.20	84.60
8/2/2021	85.90	70.40	83.40	89.20	92.20	82.10	84.00	79.90
8/5/2021	87.10	77.70	76.50	87.20	92.70	82.90	83.90	83.70
8/9/2021	86.50	65.90	75.20	75.90	90.90	80.50	85.70	82.90
8/12/2021	89.90	73.30	65.80	66.60	86.40	61.20	87.20	77.50
8/16/2021	79.80	63.10	72.50	71.70	89.10	69.60	87.50	80.40
8/23/2021	53.60	34.80	60.30	59.50	63.90	52.90	72.00	58.20
8/26/2021	59.60	41.50	48.50	64.90	77.10	56.40	78.40	67.00
8/30/2021	55.40	38.90	41.40	59.80	64.50	37.80	65.80	52.70

2022 Canopy Cover Data

Table 24: 2022 Canopy Cover Measurements

Date	C1	C2	C3	C4	AV1	AV2	AV3	AV4
5/10/2022	3.00	1.90	2.00	2.90	2.10	1.40	2.40	2.80
5/14/2022	4.40	2.40	4.80	3.40	4.30	2.80	3.30	4.50
5/17/2022	7.90	6.80	8.20	6.80	7.00	5.90	7.70	7.80
5/20/2022	6.50	9.30	7.30	11.50	8.40	8.30	9.80	9.20
5/24/2022	11.90	22.20	18.40	17.70	13.50	13.30	14.10	17.80
5/27/2022	18.20	14.20	17.30	21.50	19.80	17.90	20.00	21.60
5/31/2022	26.20	30.80	24.20	31.40	24.40	23.60	21.50	23.70
6/7/2022	60.10	47.70	43.10	52.80	59.30	43.30	37.20	40.90
6/14/2022	57.70	65.30	66.00	51.10	57.80	57.70	55.70	65.10
6/16/2022	79.40	61.30	70.00	56.50	63.60	65.70	56.00	72.00
6/21/2022	67.30	65.80	70.80	86.10	61.60	68.60	66.70	63.10
6/28/2022	90.80	85.70	80.10	81.90	78.60	76.10	66.40	73.80
7/8/2022	90.50	88.40	92.60	88.40	72.40	79.60	82.30	86.60
7/15/2022	90.00	89.30	86.80	88.10	72.20	81.50	92.00	92.90
7/22/2022	88.50	78.00	83.30	91.10	68.70	71.00	57.40	84.90
8/6/2022	69.00	68.10	71.60	47.90	42.90	19.80	20.90	39.00
8/12/2022	39.40	45.30	40.30	14.00	40.40	70.60	49.90	11.50

2021 Leaf Area Index Data

Table 25: 2021 Leaf Area Index Measurements

Date	C1	C2	C3	C4	AV1	AV2	AV3	AV4
4/29/2021	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
5/24/2021	0.17	0.18	0.21	0.17	1.80	2.61	2.36	2.09
5/31/2021	0.35	0.63	0.49	0.53	2.91	1.39	2.29	2.01
6/3/2021	0.27	0.95	0.54	0.39	1.19	1.08	1.70	0.63
6/7/2021	0.54	1.27	1.00	0.91	1.71	0.64	0.68	0.78
6/10/2021	0.62	1.09	0.93	1.01	2.27	0.88	1.43	0.89
6/15/2021	1.60	2.90	1.66	2.36	1.60	1.39	1.85	2.92
6/17/2021	1.84	5.11	2.40	1.96	1.86	1.67	2.22	3.06
6/24/2021	3.67	5.03	3.69	3.95	3.95	3.91	5.15	4.16
6/28/2021	3.94	5.72	3.89	5.38	4.31	4.60	6.72	3.23
7/1/2021	5.30	6.04	6.23	5.28	5.35	5.77	3.95	2.46
7/8/2021	5.76	6.31	6.31	8.90	7.09	8.54	6.10	5.62
7/14/2021	7.86	6.26	8.11	6.89	7.89	5.14	6.43	0.23
7/23/2021	6.77	5.64	6.20	6.77	6.36	6.05	6.22	5.29
7/29/2021	6.74	6.19	6.39	6.57	5.45	6.05	6.80	3.37
8/5/2021	7.06	6.41	5.61	5.59	5.28	4.62	4.11	5.16
8/12/2021	6.94	5.11	6.45	6.32	4.93	4.26	6.26	3.37

2022 Leaf Area Index Data

Table 26: 2022 Leaf Area Index Measurements

Date	C1	C2	C3	C4	AV1	AV2	AV3	AV4
5/1/2022	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
5/12/2022	0.06	0.02	0.05	0.05	0.05	0.05	0.05	0.09
5/16/2022	0.15	0.08	0.06	0.03	0.19	0.49	0.53	0.36
5/23/2022	0.21	0.40	0.24	0.28	0.80	0.76	1.12	1.19
5/27/2022	0.39	0.47	0.38	0.32	1.26	0.42	1.74	1.96
5/30/2022	0.44	0.39	0.28	0.40	1.13	1.31	0.78	0.71
6/3/2022	0.70	0.44	0.84	0.61	0.87	0.79	0.67	0.78
6/10/2022	2.08	1.57	1.18	1.01	0.89	1.53	0.38	0.43
6/17/2022	2.69	2.46	2.20	2.59	2.49	1.29	0.58	1.40
6/24/2022	3.88	2.93	2.52	2.39	1.88	2.30	1.42	2.59
7/1/2022	5.28	4.43	3.85	4.46	2.78	2.74	0.62	2.24
7/8/2022	4.89	4.66	5.00	4.42	1.12	3.15	5.59	4.58
7/15/2022	3.81	3.75	3.40	3.49	3.21	3.97	4.13	3.62
7/22/2022	3.68	3.77	3.84	3.77	3.91	4.14	6.38	5.14
7/29/2022	4.07	3.91	4.04	4.41	4.49	3.68	4.84	4.68
8/5/2022	2.77	2.90	3.04	2.84	4.86	2.89	2.99	2.84

Chapter 2

Read Me File

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

Hello! If you're reading this, it means I've finished working on my Ph.D. and you're curious enough to read what I've done! Thank you for your interest in this work!

I will start by saying this code base is a mess. I am not a professional coder, and I'm probably already working on either refactoring or completely rewriting this model in to something new. So with that in mind, please do not contact me if you notice errors, improvements, optimizations, confusing things, etc. This code is as it is and served as the base of my Ph.D. efforts and I'm now moving on. If you can read this and learn, great! If you'd like to collaborate on something, also great! Feel free to reach me wherever I might be. However, I am no longer maintaining this version of the model and will not make efforts to do so. If you would like to do something with the model, please start by refering to the Helios documentation (https://baileylab.ucdavis.edu/software/helios/). Also understand that being able to operate Helios correctly is prerequisite to being able to use this software. I think this makes running the model difficult and am looking in to ways to reduce that complexity, but it's what I've got in the mean time. Unfortunately, I cannot promise to help you learn Helios. If getting Helios working is too difficult, then rebuilding and running this model will also probably be too difficult. And as I've said, I'm working on new versions and will not be supporting this long term. Hopefully those will be simpler.

If you can run Helios, you'll have to rebuild all the C++ programs. For both storage size and removing some of the hardcoded file paths (one of my major programming sins that will be fixed in future models), I have deleted build folders. Rebuilding them should be simple, but it will need to be done to protect yourself from missing recoding some of the paths in the scripts.

Thank you for taking the time to read this! I wish you luck on your agrivoltaic journey!
```

Listing 1: ReadMe.txt

Genetic Algorithm Script

```
#THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

import subprocess
import time
import os
import numpy as np
import pandas as pd
```

```
import math
import matplotlib.pyplot as plt

#Variable Definitions
#Genetic Algorithm Variables
Generations = 10 #number of generations
Strings = 10 #number of design strings
K = 4 #number of strings for breeding, must be even
P = 2 #number of strings surviving for breeding
nDesignVariables = 10 #number of variables per design string
nEvaluationVariables = 6 #number of variables per string for evaluation
nSummaryVariables = 12 #number of variables per generation to summarize results (minimum and average)
threads = 4 #generate the number of threads for parallel processing
# seedValue = 100
# np.random.seed(seedValue)
defaultCost = 1

#make sure values are legal
if K % 2 != 0:
    raise ValueError("K must be even")
if K + P >= Strings:
    raise ValueError("K + P must be less than Strings")

#FilePaths
rootDirectory = #put your root directory path here
inputFilePath = rootDirectory + r"\geneticAlgorithm\inputs"
outputFilePath = rootDirectory + r"\geneticAlgorithm\outputs"
outputFilePathGraphs = rootDirectory + r"\geneticAlgorithm\outputs\graphs"
outputFilePathAlt = rootDirectory + r"\geneticAlgorithm\temp"
managerLocation = rootDirectory + r"\geneticAlgorithm\c++manager\manager\build"
agrivoltaicLocation = rootDirectory + r"\geneticAlgorithm\agrivoltaic\agrivoltaic\build"
controlLocation = rootDirectory + r"\geneticAlgorithm\control\control\build"
bestDesignLocation = rootDirectory + r"\geneticAlgorithm\bestDesign\bestDesign\build"
isDoneFilePath = rootDirectory + r"\geneticAlgorithm\outputs\isFinished.done"
weatherFileCSVPath = rootDirectory + r"\geneticAlgorithm\inputs\cimis_2021.csv"

#check to see if weather file exists
if not os.path.isfile(weatherFileCSVPath):
    raise ValueError("Weather Data file not found. Please provide weather data path.")

#Site information
julStart = 118 #starting julian date
yearStart = 2021 #starting year
seasonLength = 140 #number of days in the season
UTC = 7 #UTC timezone
latitude = 38.53 #latitude, north is positive
longtitude = 121.77 #longtitude, west is positive
```

```
55
56 #Crop growth variables
57 #LAI Variables
58 #Validation
59 LAI_a = 238.8
60 LAI_k = 587.0
61 LAI_S = 4.383
62 LAI_H = 849.6
63
64 #Photosynthesis Variables
65 A_max = 30.8
66 A_k = 0.05
67 A_rd = 2.08
68
69 #Transpiration Variables
70 Trans_max = 40
71 Trans_k = 0.07
72 Trans_rd = 5
73
74 #Temperature response function variables
75 T_base = 20
76 T_min = 20
77 T_max = 45
78 T_r = 30
79 T_q = 0.1
80
81 #Other Variables
82 RUE = 0.00000011055 #resource use efficiency
83 HI = 0.64 #harvest index
84 dt = 3600 #timestep in seconds
85 I = 100000000 #hourly irrigation (mm/hr), currently not being used
86
87 #Cost Function Definitions
88 #Function weights
89 w1 = 1 #weight for crop cost
90 w2 = 1 #weight for power cost
91 w3 = 0 #weight for water cost, currently not being used
92 w4 = 1 #weight for light use efficiency cost
93 w5 = 0 #weight for water use efficiency cost, currently not being used
94 p1 = 5 #penalty value yield loss threshold
95 p2 = 5 #penalty value land equivalent ratio
96 p3 = 5 #penalty value self shading hours
97 d1 = 0.7 #crop loss delta
98 d2 = 30 #self shading delta
99 ler = 1 #land equivalent ratio
100
101 #Function evaluations
```

```
pi = 0 #initialize total cost function
alpha = 0 #initialize alpha
beta = 0 #intitialize beta
nuL = 0 #initialize nuL
nuW = 0 #initialize nuW

#Design Variable Limits (low and high limits)
#first limit is inclusive, second limit is exclusive
numberPanelLimits = [1,16] #number of panels
lengthPanelLimits = [30,150] #length of panels (cm)
widthPanelLimits = [30,150] #width of panels (cm)
heightPanelLimits = [50,150] #height of panels (cm)
azimuthPanelLimits = [0,360] #azmimuth of panels (degrees)
elevationPanelLimits = [0,91] #elevation angle of panels (degrees)
translucencyPanelLimits = [0.01,0.01] #translucency of panels
efficiencyPanelLimits = [0.2,0.2] #efficiency of panels
xfootprintLimits = 1.52 #total footprint of panels in x dimension (m)
yfootprintLimits = 1.52 #total footprint of panels in y dimension (m)

#Function for calculating cost function, and subfunctions
#Inputs: control and agrivoltaic solar radiation, crop yield, power and ET
#Outputs:
def costFunction(controlRadiation, agrivoltaicRadiation, controlYield, agrivoltaicYield, controlPower,
     agrivoltaicPower,
                 controlET, agrivoltaicET, controlSelfShadingHours, agrivoltaicSelfShadingHours):
 # Default cost values when division by zero occurs
    default_cropCost = w1 * defaultCost
    default_powerCost = w2 * defaultCost
    default_waterCost = w3 * defaultCost
    default_lightUseEfficiencyCost = w4 * defaultCost
    default_waterUseEfficiencyCost = w5 * defaultCost

    #Calculate costs
    try:
        cropCost = w1 * ((controlYield - agrivoltaicYield) / controlYield)
        print(f"Crop Cost (calculated): {cropCost}")
    except ZeroDivisionError:
        cropCost = default_cropCost
        print(f"Crop Cost (default due to division by zero): {cropCost}")

    try:
        powerCost = w2 * ((controlPower - agrivoltaicPower) / controlPower)
        print(f"Power Cost (calculated): {powerCost}")
    except ZeroDivisionError:
        powerCost = default_powerCost
        print(f"Power Cost (default due to division by zero): {powerCost}")

```

```
    #temporary removal
    waterCost = 0

    #temporary removal
    # try:
    #     waterUseEfficiencyCost = w5 * (((controlYield / controlET) - (agrivoltaicYield / agrivoltaicET)) / (
     controlYield / controlET))
    #     print(f"Water Use Efficiency Cost (calculated): {waterUseEfficiencyCost}")
    # except ZeroDivisionError:
    #     waterUseEfficiencyCost = default_waterUseEfficiencyCost
    #     print(f"Water Use Efficiency Cost (default due to division by zero): {waterUseEfficiencyCost}")
    waterUseEfficiencyCost = 0

    try:
        waterUseEfficiencyCost = w5 * (((controlYield / controlET) - (agrivoltaicYield / agrivoltaicET)) / (
     controlYield / controlET))
        print(f"Water Use Efficiency Cost (calculated): {waterUseEfficiencyCost}")
    except ZeroDivisionError:
        waterUseEfficiencyCost = default_waterUseEfficiencyCost
        print(f"Water Use Efficiency Cost (default due to division by zero): {waterUseEfficiencyCost}")

    #Calculate penalties
    try:
        if(agrivoltaicYield / controlYield < d1):
            lossThreshold = p1
            print(f"Loss Threshold (penalty): {lossThreshold}")
        else:
            lossThreshold = 0
            print(f"Loss Threshold (no penalty): {lossThreshold}")
    except ZeroDivisionError:
        lossThreshold = p1
        print(f"Loss Threshold (default due to division by zero): {lossThreshold}")

    try:
        if((agrivoltaicYield / controlYield) + (agrivoltaicPower / controlPower) < ler):
            landEquivalentRatio = p2
            print(f"Land Equivalent Ratio (penalty): {landEquivalentRatio}")
        else:
            landEquivalentRatio = 0
            print(f"Land Equivalent Ratio (no penalty): {landEquivalentRatio}")
    except ZeroDivisionError:
        landEquivalentRatio = p2
        print(f"Land Equivalent Ratio (default due to division by zero): {landEquivalentRatio}")

    if(agrivoltaicSelfShadingHours > d2):
        selfShading = p3
        print(f"Self Shading (penalty): {selfShading}")
```

```
    else:
        selfShading = 0
        print(f"Self Shading (no penalty): {selfShading}")

    totalCost = cropCost + powerCost + lightUseEfficiencyCost + lossThreshold + landEquivalentRatio +
     selfShading    print(f"Total Cost: {totalCost}")

    return totalCost, cropCost, powerCost, waterCost, lightUseEfficiencyCost, waterUseEfficiencyCost

#Function for generating random strings
#Inputs: none
#Outputs: number of panels, length of panels, width of panels, height of panels,
#azimuth of panels, elevation of panels, reflectivity of panels, efficiency of panels
#x dimension footprint of design, y dimension footprint of design
def randomStringGenerator():
    nPanels = np.random.randint(numberPanelLimits[0],numberPanelLimits[1])
    lPanels = np.random.randint(lengthPanelLimits[0],lengthPanelLimits[1])
    wPanels = np.random.randint(widthPanelLimits[0],widthPanelLimits[1])
    hPanels = np.random.randint(heightPanelLimits[0],heightPanelLimits[1])
    aPanels = np.random.randint(azimuthPanelLimits[0],azimuthPanelLimits[1])
    ePanels = np.random.randint(elevationPanelLimits[0],elevationPanelLimits[1])
    rPanels = np.random.uniform(translucencyPanelLimits[0],translucencyPanelLimits[1])
    effPanels = np.random.uniform(efficiencyPanelLimits[0],efficiencyPanelLimits[1])
    #These two are set to the default values for now because they don't do anything yet
    xPanels = xfootprintLimits
    yPanels = yfootprintLimits
    return nPanels, lPanels, wPanels, hPanels, aPanels, ePanels, rPanels, effPanels, xPanels, yPanels

#Begin GA
#Print information about time
print("Begining Program")
initialTime = int(round(time.time()))
print("Current time is: ", time.strftime("%H:%M:%S", time.localtime()))

#Create information files for Helios
#start with creating arrays
gaData = [Generations, Strings, K, P, threads]
gaNumbersArray = np.array(gaData)

informationData = [julStart, yearStart, seasonLength, UTC, latitude, longtitude]
informationArray = np.array(informationData)

cropData = [LAI_a, LAI_k, LAI_S, LAI_H, A_max, A_k, A_rd, Trans_max, Trans_k, Trans_rd, T_base, T_min, T_max,
     T_r, T_q, RUE, HI, dt, I]
cropArray = np.array(cropData)

weatherFileCSVPath = weatherFileCSVPath.replace("\\", "/")
```

```
weatherArray = [weatherFileCSVPath]

#Save information files
gaNumbersFilePath = inputFilePath + r"\gaNumbersFile.csv"
np.savetxt(gaNumbersFilePath, gaNumbersArray[None,:], fmt = '%f', delimiter = ",")
informationFilePath = inputFilePath + r"\informationFile.csv"
np.savetxt(informationFilePath, informationArray[None,:], fmt = '%f', delimiter = ",")
cropDataFilepath = inputFilePath + r"\cropDataFile.csv"
np.savetxt(cropDataFilepath, cropArray[None,:], fmt = '%.10f', delimiter = ",")
weatherDataFilepath = inputFilePath + r"\weatherDataFilepath.csv"
np.savetxt(weatherDataFilepath, weatherArray, fmt = "%s")

#Create initial arrays
designArray = np.zeros((Strings,nDesignVariables))
summaryArray = np.zeros((Generations,nSummaryVariables + 1))

sgArray = np.zeros((Generations*Strings,2))
costVariablesArray = []

#%%

#Run control simulation
print("Running Helios Simulations for control crop. Please Wait...")
startTime = int(round(time.time()))
controlEXE = controlLocation + r"/control.exe"
result = subprocess.run([controlEXE],capture_output=True,text=True,cwd=controlLocation)
controlArray = []
controlArray = np.loadtxt(outputFilePath + r"\outputFileControl.csv", delimiter = ",",skiprows=1)
print("Output:", result.stdout)
print("Errors:", result.stderr)
endTime = int(round(time.time()))
print("Control simulation finished in " + str(round((endTime-startTime)/60)) + " minutes.")
print("Now begining Genetic Algorithm for " + str(Strings) + " strings over " + str(Generations) + " generations
      .")
print("Estimated total simulation time is " + str(round((2*(endTime-startTime)*(math.ceil((Generations*Strings)/
      threads)))/60)) + " minutes.")

#%%

#Assign initial values
for i in range(Strings):
    designArray[i,:] = randomStringGenerator()

#Run simulations
for i in range(Generations):
    for j in range(Strings):
    # Generate array with locations of the panels over the footprint of panels
```

```
        panel_locations_results = []
        panelLocations = []
        numberPanels = int(designArray[j,0])
        z = designArray[j,3]

        # If there is only one panel, place it at the center
        if numberPanels == 1:
            panelLocations.append((0, 0, z))
        else:
            # Find factors of numberPanels that are closest to each other for grid dimensions
            sqrt_val = int(np.sqrt(numberPanels))
            for rows in range(sqrt_val, 0, -1):
                if numberPanels % rows == 0:
                    cols = numberPanels // rows
                    break
            else:
                # In case it's a prime number, we set rows to 1 and columns to the number itself
                rows, cols = 1, numberPanels

            # Generate locations for multiple panels
            xInterval = np.linspace(-xfootprintLimits, xfootprintLimits, num=cols) if cols > 1 else [0]
            yInterval = np.linspace(-yfootprintLimits, yfootprintLimits, num=rows) if rows > 1 else [0]
            for y in yInterval:
                for x in xInterval:
                    # Ensure we don't create more locations than panels
                    if len(panelLocations) < numberPanels:
                        panelLocations.append((x, y, z))

        # Make sure the value in the array matches the actual number of panels
        designArray[j,0] = len(panelLocations)

        # Create array of the design string to save
        designString = [(designArray[j,0], designArray[j,1], designArray[j,2], designArray[j,3],
                         designArray[j,4], designArray[j,5], designArray[j,6], designArray[j,7])]

        panel_locations_results.append((panelLocations, designString))

        #Saving design and coordinate files
        panelCoordinatePath = inputFilePath + r"\coordinateFile" + str(j) + r".csv"
        np.savetxt(panelCoordinatePath, panelLocations, delimiter = ",", newline = '\n', fmt = '%f')
        panelDesignPath = inputFilePath + r"\designFile" + str(j) + r".csv"
        np.savetxt(panelDesignPath, designString, delimiter = ",", newline = '\n',  fmt = '%f')

    #Run Helios and wait for it to finish
    print("Running Helios Simulations for Generation " + str(i+1) + ". Please Wait...")
    startTime = int(round(time.time()))
    managerEXE = managerLocation + r"/manager.exe"
```

```
    result = subprocess.run([managerEXE],capture_output=True,text=True,cwd=agrivoltaicLocation)
    print("Output:", result.stdout)
    print("Errors:", result.stderr)
    endTime = int(round(time.time()))
    print("Simulations for Generation " + str(i+1) + " finished in " + str(round((endTime-startTime)/60))
          + " minutes. Now Calculating costs and generating next generation.")

    #create cost array for saving data
    costArray = np.zeros((Strings, nEvaluationVariables))

    #Start processing results and create cost function array
    for j in range(Strings):
        tempArray = []
        tempArray = np.loadtxt(outputFilePath + r"\outputFile" + str(j) + r".csv", delimiter = ",")

        print("Calculating Costs for Generation " + str(i+1) + ", String " + str(j+1) + ":")
        costArray[j,:] = costFunction(controlArray[0], tempArray[0], controlArray[1], tempArray[1],
                                      controlArray[2], tempArray[2], controlArray[3], tempArray[3],
                                      controlArray[4], tempArray[4])

        sgArray[i*Strings+j,0] = i
        sgArray[i*Strings+j,1] = j

    costVariablesArray.append(costArray.copy())

    #Generate evaluation array then evaluate
    evaluationArray = np.append(designArray,costArray,axis=1)

    #Rank and sort results
    rankColumn = evaluationArray[:,10]
    rankedIndex = np.argsort(rankColumn)
    evaluationArray = evaluationArray[rankedIndex]

    #Summarize Results
    summaryArray[i,:] = (i,
                         np.min(costArray[:,0]), np.mean(costArray[:,0]), np.min(costArray[:,1]), np.mean(
     costArray[:,1]),
                         np.min(costArray[:,2]), np.mean(costArray[:,2]), np.min(costArray[:,3]), np.mean(
     costArray[:,3]),
                         np.min(costArray[:,4]), np.mean(costArray[:,4]), np.min(costArray[:,5]), np.mean(
     costArray[:,5]))

    #Save Best Designs
    designArray[:P, :] = evaluationArray[:P, :10]

    #Breed new best pairs
    for j in range(P, P+int(K/2)):
```

```
        phiArray = np.random.rand(10)
        designArray[j,0] = int(round((evaluationArray[0,0]*phiArray[0])+(evaluationArray[1,0]*(1-phiArray[0]))))
        designArray[j,1] = int(round((evaluationArray[0,1]*phiArray[1])+(evaluationArray[1,1]*(1-phiArray[1]))))
        designArray[j,2] = int(round((evaluationArray[0,2]*phiArray[2])+(evaluationArray[1,2]*(1-phiArray[2]))))
        designArray[j,3] = int(round((evaluationArray[0,3]*phiArray[3])+(evaluationArray[1,3]*(1-phiArray[3]))))
        designArray[j,4] = int(round((evaluationArray[0,4]*phiArray[4])+(evaluationArray[1,4]*(1-phiArray[4]))))
        designArray[j,5] = int(round((evaluationArray[0,5]*phiArray[5])+(evaluationArray[1,5]*(1-phiArray[5]))))

    for j in range(P+int(K/2), P+K):
        phiArray = np.random.rand(10)
        designArray[j,0] = int(round((evaluationArray[1,0]*phiArray[0])+(evaluationArray[0,0]*(1-phiArray[0]))))
        designArray[j,1] = int(round((evaluationArray[1,1]*phiArray[1])+(evaluationArray[0,1]*(1-phiArray[1]))))
        designArray[j,2] = int(round((evaluationArray[1,2]*phiArray[2])+(evaluationArray[0,2]*(1-phiArray[2]))))
        designArray[j,3] = int(round((evaluationArray[1,3]*phiArray[3])+(evaluationArray[0,3]*(1-phiArray[3]))))
        designArray[j,4] = int(round((evaluationArray[1,4]*phiArray[4])+(evaluationArray[0,4]*(1-phiArray[4]))))
        designArray[j,5] = int(round((evaluationArray[1,5]*phiArray[5])+(evaluationArray[0,5]*(1-phiArray[5]))))

    #Generate new random designs
    rows = len(designArray)
    for j in range(P+K,rows):
        designArray[j,:] = randomStringGenerator()

    print("Cost for Generation " + str(i+1) + " finished.")

#Save best result to a file
headerstring = "Number of Panels,Panel Length (cm),Panel Width (cm),Panel Height (cm),Panel Azimuth (degrees),
     Panel Elevation (degrees),Panel Reflectivity,Panel Efficiency,not used,not used,Total Cost,Crop Cost (Yield
     /Yield),Power Cost (W/W),Water Cost(mm/mm),Light Use Efficiency Cost,Water Use Efficiency Cost"
bestDesignPath = outputFilePath + r"\bestDesign.csv"
bestDesign = evaluationArray[0:1]
np.savetxt(bestDesignPath, bestDesign, fmt = '%f', newline = "\n", delimiter = ",", header = headerstring)

print("Genetic Algorithm Finished. Current time is: ", time.strftime("%H:%M:%S", time.localtime()))
finalTime = int(round(time.time()))
simulationTime = round((finalTime - initialTime)/60)
if simulationTime == 1:
    print("Simulation time was: " + str(simulationTime) + " minute")
else:
    print("Simulation time was: " + str(simulationTime) + " minutes")

#Run best design
print("Running Helios Simulations for Best Design. Please Wait...")
startTime = int(round(time.time()))
bestDesignEXE = bestDesignLocation + r"\bestDesign.exe"
result = subprocess.run([bestDesignEXE],capture_output=True,text=True,cwd=bestDesignLocation)
print("Output:", result.stdout)
print("Errors:", result.stderr)
```

```
endTime = int(round(time.time()))
print("Best Design simulation finished in " + str(round((endTime-startTime)/60)) + " minutes. Current time is: "
      , time.strftime("%H:%M:%S", time.localtime()))
finalTime = int(round(time.time()))
simulationTime = round((finalTime - initialTime)/60)
if simulationTime == 1:
    print("Simulation time was: " + str(simulationTime) + " minute")
else:
    print("Simulation time was: " + str(simulationTime) + " minutes")
print("Now graphing results...")

#%%

costArrayUnpacked = np.vstack(costVariablesArray)
costsArrayGraphing = np.append(sgArray, costArrayUnpacked,axis=1)

bestDesignHourlyString = outputFilePath + r"\hourlyDataBestDesign.csv"
controlHourlyString = outputFilePath + r"\hourlyDataControl.csv"

hourly_best_design_data = pd.read_csv(bestDesignHourlyString)
hourly_control_data = pd.read_csv(controlHourlyString)

bestDesignDailyString = outputFilePath + r"\dailyDataBestDesign.csv"
controlDailyString = outputFilePath +  r"\dailyDataControl.csv"

daily_best_design_data = pd.read_csv(bestDesignDailyString)
daily_control_data = pd.read_csv(controlDailyString)

# Group by the 'Hour' column and compute the mean for each hour for both datasets
hourly_best_design_grouped = hourly_best_design_data.groupby('Hour').mean().reset_index()
hourly_control_grouped = hourly_control_data.groupby('Hour').mean().reset_index()

# List of metrics to plot for hourly data
hourly_metrics = [
    'Hourly Power (W/m2/hr)',
    'Average Photosynthesic Assimilation (umol CO2/m2/hr)',
]

# Plotting the metrics for hourly data in the style of daily data
plt.figure(figsize=(6.5, 9))
plt.suptitle("Hourly Performance Metrics", y=0.94)
for idx, metric in enumerate(hourly_metrics, 1):
    plt.subplot(len(hourly_metrics), 1, idx)
    plt.plot(hourly_best_design_grouped['Hour'], hourly_best_design_grouped[metric], label="Best Design", color=
     "blue")
    plt.plot(hourly_control_grouped['Hour'], hourly_control_grouped[metric], label="Control", color="red",
     linestyle="--")
```

```
    plt.ylabel(metric,fontsize = 8)
    plt.title(metric + " vs. Hour")
    plt.legend()
    plt.xticks(range(0, 25, 1))  # Ensure hourly ticks for clarity
filename = outputFilePathGraphs + r"\hourly_comparison.png"
if os.path.exists(filename):
    os.remove(filename)
plt.savefig(filename, dpi=300, bbox_inches='tight')
plt.close()

# List of metrics to plot
daily_metrics = [
    "Daily Power (W/m2)",
    "Biomass (tons/acre)",
]

# Function to scale the values and adjust labels
def adjust_metric_values(data, metric):
    if metric == "Daily Power (W/m2)":
        return data / 1000, "Daily Power (kWh/m2)"
    return data, metric

# Plotting the metrics against "Days After Planting"
plt.figure(figsize=(6.5, 9))
plt.suptitle("Daily Performance Metrics, Days After Planting", y=0.94)

for idx, metric in enumerate(daily_metrics, 1):
    plt.subplot(len(daily_metrics), 1, idx)

    # Adjust metric values and label
    best_design_values, adjusted_label = adjust_metric_values(daily_best_design_data[metric], metric)
    control_values, _ = adjust_metric_values(daily_control_data[metric], metric)

    plt.plot(daily_best_design_data["Days After Planting"], best_design_values, label="Best Design", color="blue
     ")
    plt.plot(daily_control_data["Days After Planting"], control_values, label="Control", color="red", linestyle=
     "--")

    plt.ylabel(adjusted_label, fontsize=8)
    plt.title(f"{adjusted_label} vs. Days After Planting")
    plt.legend()

filename = outputFilePathGraphs + r"\daily_DOP.png"
if os.path.exists(filename):
    os.remove(filename)
plt.savefig(filename, dpi=300, bbox_inches='tight')
plt.close()
```

```
508
509 # Plotting the metrics against "Growing Degree Days (days C)"
510 plt.figure(figsize=(6.5, 9))
511 plt.suptitle("Daily Performance Metrics, Growing Degree Days (days C)", y=0.94)
512
513 for idx, metric in enumerate(daily_metrics, 1):
514     plt.subplot(len(daily_metrics), 1, idx)
515
516     # Adjust metric values and label
517     best_design_values, adjusted_label = adjust_metric_values(daily_best_design_data[metric], metric)
518     control_values, _ = adjust_metric_values(daily_control_data[metric], metric)
519
520     plt.plot(daily_best_design_data["Growing Degree Days (days C)"], best_design_values, label="Best Design",
        color="blue")
521     plt.plot(daily_control_data["Growing Degree Days (days C)"], control_values, label="Control", color="red",
        linestyle="--")
522
523     #plt.xlabel("Growing Degree Days (days C)")
524     plt.ylabel(adjusted_label)
525     plt.title(f"{adjusted_label} vs. Growing Degree Days (days C)")
526     plt.legend()
527
528 filename = outputFilePathGraphs + r"\daily_GDD.png"
529 if os.path.exists(filename):
530     os.remove(filename)
531 plt.savefig(filename, dpi=300, bbox_inches='tight')
532 plt.close()
533
534 evaluationColumns = [
535     "Generation",
536     "Design",
537     "Total Cost",
538     "Crop Cost",
539     "Power Cost",
540     "Water Cost",
541     "Light Use Efficiency Cost",
542     "Water Use Efficiency Cost"
543 ]
544
545 df = pd.DataFrame(costsArrayGraphing, columns=evaluationColumns)
546
547 #including this temporarily to suppress water outputs
548 df = df[[
549     "Generation",
550     "Design",
551     "Total Cost",
552     "Crop Cost",
```

```
    "Power Cost",
    "Light Use Efficiency Cost"
]]

# Calculate average and minimum 'Total Cost' per generation
avg_total_cost = df.groupby(df.iloc[:, 0]).mean().iloc[:, 2]
min_total_cost = df.groupby(df.iloc[:, 0]).min().iloc[:, 2]

# Plot
plt.figure(figsize=(6.5, 4))
avg_total_cost.plot(label='Average Total Cost', marker='o',linestyle='-')
min_total_cost.plot(label='Minimum Total Cost', marker='x',linestyle="--")
plt.title('Average and Minimum Total Cost per Generation')
plt.xlabel('Generation')
plt.ylabel('Cost')
plt.legend()
plt.grid(False)
filename = outputFilePathGraphs + r"\GenerationCosts.jpg"
if os.path.exists(filename):
    os.remove(filename)
plt.savefig(filename, dpi = 300, bbox_inches = 'tight')
plt.close()

plt.figure(figsize=(6.5, 4))
min_total_cost.plot(label='Minimum Total Cost', marker='o')
plt.title('Minimum Total Cost per Generation')
plt.xlabel('Generation')
plt.ylabel('Cost')
plt.legend()
plt.grid(False)
filename = outputFilePathGraphs + r"\MinimumCosts.jpg"
if os.path.exists(filename):
    os.remove(filename)
plt.savefig(filename, dpi = 300, bbox_inches = 'tight')
plt.close()

# Calculate the average of each cost per generation
avg_values = df.groupby(df.iloc[:, 0]).mean()

#temporarily suppressing water outputs
avg_crop_cost = avg_values.iloc[:, 3]
avg_power_cost = avg_values.iloc[:, 4]
avg_water_cost = 0
avg_light_use_efficiency_cost = avg_values.iloc[:, 5]
avg_water_use_efficiency_cost = 0

# Calculate the sum of the average values of the 5 costs for each generation
```

```
avg_sum_cost = avg_crop_cost + avg_power_cost + avg_water_cost + avg_light_use_efficiency_cost +
    avg_water_use_efficiency_cost

# Plot
plt.figure(figsize=(6.5, 4))
avg_crop_cost.plot(label='Crop Cost', marker='o', linestyle='--', linewidth=1)
avg_power_cost.plot(label='Power Cost', marker='s', linestyle=':', linewidth=1)
avg_light_use_efficiency_cost.plot(label='Light Use Efficiency Cost', marker='v', linestyle='-.', linewidth=1)
avg_sum_cost.plot(label='Average Sum of Costs', marker='*', linestyle='-', linewidth=1, color='red')

plt.title('Average Costs per Generation')
plt.xlabel('Generation')
plt.ylabel('Cost')
plt.legend()
plt.grid(False)
filename = outputFilePathGraphs + r"\IndividualCosts.jpg"
if os.path.exists(filename):
    os.remove(filename)
plt.savefig(filename, dpi = 300, bbox_inches = 'tight')
plt.close()

print("Plots finished. Plots can be found in: " + outputFilePath)
print("Best Design can be found in: " + bestDesignPath)
```

Listing 2: genetic$_a$*lgorithm.py*

Manager Script

```
//THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
    TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

#include <iostream>
#include <thread>
#include <vector>
#include <fstream>
#include <sstream>
#include <string>

using namespace std;

string gaNumbersFilepath = //your filepath here /geneticAlgorithm/inputs/gaNumbersFile.csv";
string outputsFilepath = //your filepath here /geneticAlgorithm/outputs";
string agrivoltaicFilepath = //your filepath here /geneticAlgorithm/agrivoltaic/agrivoltaic/cmake-build-debug-
    visual-studio";
```

```
15 string controlFilepath = //your filepath here /geneticAlgorithm/control/control/cmake-build-debug-visual-studio
       ";
16
17 vector<int> readValuesFromCSV(const string& filename) {
18     ifstream file(filename);
19     vector<int> values;
20     if (!file.is_open()) {
21         cerr << "Failed to open " << filename << "." << std::endl;
22         return values;  // Empty vector
23     }
24     string line;
25     getline(file, line);  // Read the single line
26     stringstream ss(line);
27     string token;
28     while (getline(ss, token, ',')) {  // Split by comma
29         values.push_back(stoi(token));
30     }
31     file.close();
32     return values;
33 }
34
35 void launchReceiver(int startIndex, int endIndex, int Generations, int numStrings, int K, int P){
36     for(int index = startIndex; index < endIndex; ++index){
37         string command = agrivoltaicFilepath + "/agrivoltaic.exe " + to_string(index) + " "
38                          + to_string(Generations) + " " + to_string(numStrings) + " "
39                          + to_string(K) + " " + to_string(P);
40         if (system(command.c_str()) != 0) {
41             cerr << "Failed to execute agrivoltaic.exe with index: " << index << ", Generations: "
42                  << Generations << ", Strings: " << numStrings << ", K:" << K << ", P:" << P << "." << endl;
43         }
44     }
45 }
46
47 int main(void){
48     vector<int> values = readValuesFromCSV(gaNumbersFilepath);
49     if (values.empty()) {
50         return 1;  // Exit with error
51     }
52     cout << ", Generations: "<< values[0] << ", Strings: " << values[1]
53          << ", K:" << values[2] << ", P:" << values[3] << ", Threads: " << values[4] << "." << endl;
54
55     int numThreads = values[4];
56     int numStrings = values[1];
57
58     int baseStringsPerThread = numStrings / numThreads;
59     int extraStrings = numStrings % numThreads;
60
```

```
    int startIndex = 0;
    vector<thread> threads;
    for (int i = 0; i < numThreads; ++i) {
        int stringsForThisThread = baseStringsPerThread + (i < extraStrings ? 1 : 0);
        int endIndex = startIndex + stringsForThisThread;
        threads.push_back(thread(launchReceiver, startIndex, endIndex, values[0], values[1], values[2], values[3]));
        cout << "Launching thread: " << i << " for strings [" << startIndex << ", " << endIndex - 1 << "]." << endl;
        startIndex = endIndex;
    }

    // Wait for all threads to complete
    for (auto& t : threads) {
        t.join();
    }
    return 0;
}
```

Listing 3: Manager.cpp

Best Design Script

```
//THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

#include "Context.h"
#include "EnergyBalanceModel.h"
#include "PhotosynthesisModel.h"
#include "StomatalConductanceModel.h"
#include "Visualizer.h"
#include "RadiationModel.h"
#include "SolarPosition.h"
#include "CanopyGenerator.h"
#include <iostream>
#include <thread>
#include <vector>
#include <fstream>
#include <sstream>
#include <string>

using namespace helios;
using namespace std;

string rootDirectory = //your root directory here;
```

```
22
23 string informationFilepath = rootDirectory + "/geneticAlgorithm/inputs/informationFile.csv";
24 string cropDataFilepath = rootDirectory + "/geneticAlgorithm/inputs/cropDataFile.csv";
25 string weatherDataFilepath = rootDirectory + "/geneticAlgorithm/inputs/weatherDataFilepath.csv";
26 string outputsFilepath = rootDirectory + "/geneticAlgorithm/outputs";
27 string inputPath = rootDirectory + "/geneticAlgorithm/inputs";
28 const char* panelTexture = "F:/Helios/Helios/projects/geneticAlgorithm/inputs/solarpanel.png";
29
30 //custom enum for the smart csv reading
31 enum DataType {
32     TYPE_INT,
33     TYPE_FLOAT,
34     TYPE_STRING
35 };
36
37 struct Point {
38     float x;
39     float y;
40     float z;
41 };
42
43 //function for reading csv files
44 void readCSVRow(const string &filepath, vector<pair<void *, DataType>> &destinations) {
45     ifstream fileStream(filepath);
46     if (!fileStream) {
47         cerr << "Failed to open file." << endl;
48         return;
49     }
50     string line;
51     getline(fileStream, line);
52     istringstream iss(line);
53     string token;
54
55     for (auto &dest : destinations) {
56         getline(iss, token, ',');
57         switch (dest.second) {
58             case TYPE_INT:
59                 *static_cast<int *>(dest.first) = stoi(token);
60                 break;
61             case TYPE_FLOAT:
62                 *static_cast<float *>(dest.first) = stof(token);
63                 break;
64             case TYPE_STRING:
65                 *static_cast<string *>(dest.first) = token;
66                 break;
67         }
68     }
```

```
}

vector<Point> readLocationCSV(const string& filepath) {
    vector<Point> data;
    ifstream fileStream(filepath);

    if (!fileStream) {
        cerr << "Failed to open file." << endl;
        return data; // Return empty vector
    }

    string line;
    while (getline(fileStream, line)) {
        istringstream iss(line);
        Point pt;
        char delimiter;

        iss >> pt.x >> delimiter >> pt.y >> delimiter >> pt.z;

        data.push_back(pt);
    }

    return data;
}

//Crop growth functions
//function for calculating the normalized temperature response function
//temporarily set to 1
float normalizedTResponse(float Tleaf, float Tmin, float Tmax, float Tr, float q){
    //return pow((Tleaf - Tmin) / (Tr - Tmin), q) * ((Tmax - Tleaf) / (Tmax - Tr));
    return 1;
}

//function for photosynthetic response
//Based on numeric solution
float photosynthesisResponse(float PAR, float Amax, float Ak, float Ard){
    return ((Amax * Ak * PAR) / ((Ak * PAR) + Amax)) - Ard;
}

//LAI function
float laiFunction(float GDD, float LAIa, float LAIk, float LAIS, float LAIH){
    return (LAIa * exp(-LAIk/GDD)) / (1 + exp(LAIS - (LAIH / GDD)));
}

//Function to calculate hydraulic conductivity
//Note: Actual Max Theta = ThetaMax + ThetaMin, adjust ThetaMax accordingly
//temporarily set to 1
```

```
float hydrualicConductivityFunction( float Theta, float ThetaMax, float bcoefficicient, float ThetaMin){
    //return (ThetaMax * exp(-bcoefficicient * Theta)) + ThetaMin;
    return 1;
}

//Function to calculate bare soil evaporation
//Uses Priestley-Taylor equation with certain assumptions (but can be overridden)
//temporarily set to 0
float evaporation(float Rn, float T, float p = 101.3, float alpha = 1.0, float G = 0){
    float delta = 4098 * (0.6108 * exp((17.27 * T)/(T+237.3))) / pow(T + 237.3,2);
    float gamma = 0.665 * pow(10,-3) * p;
    float E = alpha * delta * (Rn - G) / (delta + gamma);
    //return E;
    return 0;
}

//Function to calculate leaf transpiration
//Based on numeric solution
//temporarily set to 0
float transpiration(float PAR, float Transmax, float k, float Transd){
    //return ((Transmax * k * PAR) / ((k * PAR) + Transmax)) - Transd;
    return 0;
}

float growingDegreeDayFunction(float Tair, float Tbase, float Tmax, float dt){
    if((Tair > Tbase) && (Tair < Tmax)){
        return (Tair - Tbase) * (dt / 86400); //dt / seconds per day
    }
    else{
        return 0;
    }
}

int main(void){
    Context context;
    Visualizer visualizer(800);

    // Values in informationData
    int julStart, yearStart, seasonLength, UTC;
    float latitude, longitude;

    // Values in cropData
    float LAI_a, LAI_k, LAI_S, LAI_H, A_max, A_k, A_rd, Trans_max, Trans_k, Trans_rd,
        T_base, T_min, T_max, T_r, T_q, RUE, HI, dt, I;

    // Values in DesignData
    float numberPanels, lengthPanels, widthPanels, heightPanels, azimuthPanels, elevationPanels;
```

```
    float reflectivityPanels, efficiencyPanels;

    // Data in weatherFilepathData
    string weatherCSVPath;

    string designFile = inputPath + "/designFile0.csv";
    string coordinateFile = inputPath + "/coordinateFile0.csv";

    vector<pair<void *, DataType>> informationData = {
            {&julStart, TYPE_INT},
            {&yearStart, TYPE_INT},
            {&seasonLength, TYPE_INT},
            {&UTC, TYPE_INT},
            {&latitude, TYPE_FLOAT},
            {&longitude, TYPE_FLOAT}
    };

    vector<pair<void *, DataType>> cropData = {
            {&LAI_a, TYPE_FLOAT},
            {&LAI_k, TYPE_FLOAT},
            {&LAI_S, TYPE_FLOAT},
            {&LAI_H, TYPE_FLOAT},
            {&A_max, TYPE_FLOAT},
            {&A_k, TYPE_FLOAT},
            {&A_rd, TYPE_FLOAT},
            {&Trans_max, TYPE_FLOAT},
            {&Trans_k, TYPE_FLOAT},
            {&Trans_rd, TYPE_FLOAT},
            {&T_base, TYPE_FLOAT},
            {&T_min, TYPE_FLOAT},
            {&T_max, TYPE_FLOAT},
            {&T_r, TYPE_FLOAT},
            {&T_q, TYPE_FLOAT},
            {&RUE, TYPE_FLOAT},
            {&HI, TYPE_FLOAT},
            {&dt, TYPE_FLOAT},
            {&I, TYPE_FLOAT}
    };

    vector<pair<void *, DataType>> designData = {
            {&numberPanels, TYPE_INT},
            {&lengthPanels, TYPE_FLOAT},
            {&widthPanels, TYPE_FLOAT},
            {&heightPanels, TYPE_FLOAT},
            {&azimuthPanels, TYPE_FLOAT},
            {&elevationPanels, TYPE_FLOAT},
            {&reflectivityPanels, TYPE_FLOAT},
```

```
            {&efficiencyPanels, TYPE_FLOAT},
    };

    vector<pair<void *, DataType>> weatherFilePathData = {
            {&weatherCSVPath, TYPE_STRING}
    };


    readCSVRow(informationFilepath, informationData);
    readCSVRow(cropDataFilepath, cropData);
    readCSVRow(designFile,designData);
    readCSVRow(weatherDataFilepath, weatherFilePathData);

    vector<Point> coordinates = readLocationCSV(coordinateFile);

    // Print the values for confirmation
    cout << "Information Data:\n";
    cout << "Julian Start: " << julStart << ", Start Year: " << yearStart << ", Season Length: " << seasonLength
         << ", UTC: " << UTC << ", Latitude: " << latitude << ", Longtitude: " << longitude << endl;

    cout << "\nCrop Data:\n";
    cout << "LAI a: " << LAI_a << ", LAI k: " << LAI_k << ", LAI S: " << LAI_S
         << ", LAI H: " << LAI_H << ", A max: " << A_max << ", A k: " << A_k
         << ", A rd: " << A_rd << ", Transpiration max: " << Trans_max << ", Transpiration k: " << Trans_k
         << ", Transpiration rd: " << Trans_rd << ", T base: " << T_base << ", T min: " << T_min
         << ", T max: " << T_max << ", T r: " << T_r << ", T q: " << T_q
         << ", RUE: " << RUE << ", HI: " << HI << ", dt: " << dt << ", Irrigation: " << I << "." << endl;

    //add irrigation, and other water related parameters
    cout << "\nWeather File Path:\n" << weatherCSVPath << endl;

    context.loadTabularTimeseriesData(weatherCSVPath,{"CIMIS"},",");

    vector<double> dailyRow;
    vector<double> hourlyRow;

    vector<vector<double>> dailyData;
    vector<vector<double>> hourlyData;

    float dailyDayNumber;
    float hourNumber;
    float growingDegreeDaysNumber;
    float dailyA;
    float dailyLAI;
    float dailyET;
    float hourlyDayNumber;
    float hourlyA;
```

```
    float hourlyET;

    //variables for generating the ground
    vec3 origin(0,0,0);
    vec2 extent(3,3);
    int2 nTiles(3,3);
    int2 subpatches(25,25);

    Date julDay;
    Time time;

    float pressure = 101300; //atmospheric pressure (Pa)
    float temperature; //air temperature (K)
    float humidity;    //relative humidity (-)
    float precipitation; //precipiation (mm)
    float turbidity = 0.05;  //atmospheric turbidity coeff (-)
    float Rflux;
    float fdiff;
    float RDirect;
    float RDiffuse;
    float reduction; //variable for water stress impact
    float hourlyPower = 0;
    float dailyPower = 0;
    float selfShadingHour = 0;
    float dailySelfShadingHours = 0;
    float pvEfficiency = 0.2; //temporarily constant value, change later to read from data

    //temporary variable for calibration
    float accumulatedA = 0;

    //values for self shading hours search
    float min_val;
    float max_val;
    float ave_val;
    float ssh_ave;

    float recompile_test_ignore;

    //values for output file
    float radiationSumOutput;
    float yieldOutput;
    float powerOutput;
    float ETOutput;
    float selfShadingHoursOutput;

    context.cropDomainX(origin.x + make_vec2(-0.5 * extent.x, 0.5 * extent.x));
    context.cropDomainY(origin.y + make_vec2(-0.5 * extent.y, 0.5 * extent.y));
```

```
304
305     //build canopy and ground
306     CanopyGenerator cgen(&context);
307     SphericalCrownsCanopyParameters params; //build spherical canopies
308     float GDD = 0;
309     float GDD_Sum = 0;
310     float leafAreaIndex = laiFunction(GDD_Sum,LAI_a,LAI_k,LAI_S,LAI_H);
311     cgen.seedRandomGenerator(10);
312     params.plant_spacing = make_vec2(1.52, 0.3);
313     params.plant_count = make_int2(2,10);
314     params.crown_radius = make_vec3(1,0.17,.2);
315     params.leaf_area_density = leafAreaIndex;
316     cgen.buildCanopy(params);
317     //cgen.buildGround(origin,extent,nTiles,subpatches,"plugins/canopygenerator/textures/dirt.jpg");
318
319     //variables for UUIDs
320     vector<uint> groundUUIDs = context.addTile(make_vec3(0,0,0),extent,nullrotation, make_int2(10,10), RGBcolor
        (.341,.255,.184));
321     vector<uint> leafUUIDs;
322     vector<uint> panelUUIDs;
323     vector<uint> allUUIDs;
324
325     leafUUIDs = cgen.getLeafUUIDs();
326     groundUUIDs = cgen.getGroundUUIDs();
327
328     //Generate solar panels
329     vec2 panelSize = make_vec2(lengthPanels/100,widthPanels/100);
330     cout << "Panel length: " << lengthPanels/100 << " m, Panel width: " << widthPanels/100
331          << " m, Panel height: "<< heightPanels/100 << " m, Panel elevation: " << elevationPanels
332          << " degrees, azimuth: " << azimuthPanels << " degrees." << endl;
333     float elevationRad = 2*M_PI*(elevationPanels/360);
334     float azimuthRad = 2*M_PI*(azimuthPanels/360);
335     SphericalCoord panelAngle = make_SphericalCoord(elevationRad,azimuthRad);
336     int2 panelSubDivisions = make_int2(10,10);
337     for (const Point& pt : coordinates) {
338         vec3 panelLocation = make_vec3(pt.x,pt.y,pt.z/100);
339         cout << "Panel location is: " << panelLocation << endl;
340         panelUUIDs = context.addTile(panelLocation, panelSize, panelAngle, panelSubDivisions, panelTexture);
341     }
342
343     context.setPrimitiveData(groundUUIDs,"twosided_flat",uint(0));
344
345     //initialize solar position
346     SolarPosition solarposition( UTC, latitude, longitude, &context );
347
348     // Get into radiation model
349     RadiationModel radiation(&context);
```

```
350
351     uint SunSource = radiation.addCollimatedRadiationSource(); //adding the source, direction added later
352
353     radiation.addRadiationBand("SRad");
354     radiation.setDirectRayCount("SRad",1000);
355     radiation.disableEmission("SRad");
356
357     radiation.enforcePeriodicBoundary("xy");
358
359     context.setPrimitiveData(groundUUIDs,"twosided_flag",uint(0));
360
361     radiation.updateGeometry();
362
363     float biomass = 0;
364     float groundDayFlux = 0;
365     float leafDayFlux = 0;
366     float leafTFlux = 0;
367     float groundEFlux = 0;
368
369     float daysAfterPlanting = 0;
370
371     //start the loop
372     cout << "Simulating for " << seasonLength << " days." << endl;
373     for(int day = julStart; day < julStart + seasonLength; day++) {
374         float dayF = static_cast<float>(day);
375         julDay = make_Date(day,yearStart);
376         context.setDate(julDay);
377
378         context.deletePrimitive(leafUUIDs);
379         params.leaf_area_density = leafAreaIndex;
380         cgen.buildCanopy(params);
381         leafUUIDs = cgen.getLeafUUIDs();
382         radiation.updateGeometry();
383
384         allUUIDs = context.getAllUUIDs();
385
386         for(int hour = 0; hour < 24; hour++){
387             context.setTime(0,0,hour); //set time
388             vec3 sunDirection = solarposition.getSunDirectionVector(); //gets direction from the plugin
389             radiation.setSourcePosition(SunSource, sunDirection); //set the source and direction
390
391             //generate direct and diffuse radiation at timestep
392             time = context.getTime();
393             Rflux = context.queryTimeseriesData("net_radiation", julDay, time);
394             temperature = context.queryTimeseriesData("air_temperature", julDay, time);
395             humidity = context.queryTimeseriesData("air_humidity", julDay, time);
396             precipitation = context.queryTimeseriesData("precipitation",julDay,time);
```

```
397
398             context.setPrimitiveData(allUUIDs,"air_temperature",temperature);
399             context.setPrimitiveData(allUUIDs,"air_humidity",humidity);
400
401             fdiff = solarposition.getDiffuseFraction(pressure,temperature,humidity,turbidity); //this is a
    fraction of total flux
402             RDirect = Rflux*(1-fdiff);
403             RDiffuse = Rflux*fdiff;
404             cout << "Day: " << julDay << ". Time: " << time << endl;
405             cout << "CIMIS SRad: " << Rflux << " W/m2. CIMIS Air Temp: " << temperature << " C. CIMIS Relative
    Humidity: " << humidity << endl;
406             cout << "Diffuse fraction is: " << fdiff << ". Direct flux is: " << RDirect << " W/m2. Diffuse flux
    is: " << RDiffuse << " W/m2." << endl;
407             GDD = growingDegreeDayFunction(temperature,T_base,T_max,dt);
408             cout << "Air temperature is: " << temperature << ". Base temperature is: " << T_base
409             << ". Growing Degree Days are: " << GDD << " days C" << endl;
410             radiation.setSourceFlux(SunSource,"SRad",RDirect);
411             radiation.setDiffuseRadiationFlux("SRad",RDiffuse);
412
413             context.setPrimitiveData(leafUUIDs, "relfectivity_SRad",0.05f);
414             context.setPrimitiveData(leafUUIDs,"transmissivity_SRad",0.05f);
415
416             context.setPrimitiveData(groundUUIDs,"reflectivity_SRad",0.15f);
417             context.setPrimitiveData(groundUUIDs,"reflectivity_SRad",0.35f);
418
419             radiation.runBand("SRad");
420
421             float A_canopy = 0;
422             float T_canopy = 0;
423             for(uint UUID: leafUUIDs){
424                 float leafTotal, leafDirect, leafDiffuse, A, T;
425                 float area = context.getPrimitiveArea(UUID);
426                 context.getPrimitiveData(UUID, "radiation_flux_SRad", leafTotal);
427                 A = photosynthesisResponse(leafTotal * 2.5,A_max,A_k,A_rd) * area * dt;
428                 context.setPrimitiveData(UUID, "Photosynthesis", A);
429                 A_canopy += A;
430                 T = transpiration(leafTotal * 2.5,Trans_max,Trans_k,Trans_k) * 0.000018016 * 1000000 * area;
431                 context.setPrimitiveData(UUID, "Transpiration", T);
432                 T_canopy += T;
433             }
434
435             accumulatedA += A_canopy;
436             biomass = accumulatedA * RUE;
437
438             float AverageT;
439             context.calculatePrimitiveDataAreaWeightedMean(leafUUIDs, "Transpiration",AverageT);
440             float AverageA;
```

```
            context.calculatePrimitiveDataAreaWeightedMean(leafUUIDs,"Photosynthesis",AverageA);

            float E_ground = 0;
            for(uint UUID: groundUUIDs){
                float rad_Ground, Evap, alpha;
                float area = context.getPrimitiveArea(UUID);
                context.getPrimitiveData(UUID, "radiation_flux_SRad", rad_Ground);
                Evap = evaporation(rad_Ground,temperature) * area;
                context.setPrimitiveData(UUID, "Evaporation", Evap);
                E_ground += Evap;
            }

            float AverageE;
            context.calculatePrimitiveDataAreaWeightedMean(groundUUIDs,"Evaporation",AverageE);

            float AWater_canopy = A_canopy * 0.000018016 * 1000000;
            float WaterDemand = E_ground + T_canopy + AWater_canopy;
            float AvailableWater = I + precipitation;

            if(WaterDemand > AvailableWater){
                if(WaterDemand - AvailableWater < 0){
                    reduction = 0;
                }
                else{
                    reduction = 1 - ((WaterDemand - AvailableWater)/ WaterDemand);
                }
                GDD = GDD * reduction;
                A_canopy = A_canopy * reduction;
                AverageA = AverageA * reduction;
                AverageE = AverageE * reduction;
                AverageT = AverageT * reduction;
            }

            hourlyPower = 0;
            vector<float> SSHcalc;
            for(uint UUID: panelUUIDs){
                float rad_Panel, power;
                float area = context.getPrimitiveArea(UUID);
                context.getPrimitiveData(UUID, "radiation_flux_SRad", rad_Panel);
                power = rad_Panel * efficiencyPanels;
                context.setPrimitiveData(UUID, "Power", power);
                SSHcalc.push_back(power);
            }

            context.calculatePrimitiveDataAreaWeightedMean(panelUUIDs,"Power",hourlyPower);

            min_val = SSHcalc[0];
```

```
488             max_val = SSHcalc[0];
489             ave_val = 0;
490
491             for(float value: SSHcalc){
492                 min_val = min(min_val,value);
493                 max_val = max(max_val,value);
494                 ave_val += value;
495             }
496
497             ssh_ave = ave_val / static_cast<float>(SSHcalc.size());
498
499             selfShadingHour = 0;
500             if(min_val < 0.25 * max_val){
501                 selfShadingHour += 1;
502             }
503
504             leafTFlux += T_canopy;
505             groundEFlux += E_ground;
506             GDD_Sum = GDD_Sum + GDD;
507
508             dailyPower += hourlyPower;
509             dailySelfShadingHours += selfShadingHour;
510
511             float hourF = static_cast<float>(hour);
512             hourlyRow = {daysAfterPlanting,dayF,hourF,hourlyPower,AverageA,AverageE,AverageT,
513                          AverageE+AverageT,selfShadingHour};
514             hourlyData.push_back(hourlyRow);
515
516             float leafHourFlux;
517             context.calculatePrimitiveDataAreaWeightedSum(leafUUIDs,"radiation_flux_SRad",leafHourFlux);
518             leafDayFlux += leafHourFlux * 3600 / 1e6 / (1 / (4 / (3 * M_PI * 0.5f*0.5f*0.5f)));
519             float groundHourFlux;
520             context.calculatePrimitiveDataAreaWeightedSum(groundUUIDs,"radiation_flux_SRad",groundHourFlux);
521             groundDayFlux += groundHourFlux * 3600 / 1e6 / 9;
522             cout << "Hourly flux on leafs is: " << leafHourFlux << " W. Hourly flux on ground is: " <<
    groundHourFlux << " W." << endl;
523             cout << "Daily flux on leafs is: " << leafDayFlux << " MJ/m2. Daily flux on ground is: " <<
    groundDayFlux << " MJ/m2." << endl;
524         }
525
526         //add values to the vector
527         dailyRow = {daysAfterPlanting,dayF,GDD_Sum,dailyPower,accumulatedA,biomass,groundEFlux,leafTFlux,
528                     leafTFlux+groundEFlux,dailySelfShadingHours};
529         dailyData.push_back(dailyRow);
530
531         leafAreaIndex = laiFunction(GDD_Sum,LAI_a,LAI_k,LAI_S,LAI_H);
532         cout << "GDD Sum = " << GDD_Sum << endl;
```

```
        cout << "LAI = " << leafAreaIndex << endl;

        daysAfterPlanting += 1;
    }

    radiationSumOutput = leafDayFlux + groundDayFlux;
    yieldOutput = biomass * HI;
    powerOutput = dailyPower;
    ETOutput = groundEFlux + leafTFlux;
    selfShadingHoursOutput = dailySelfShadingHours;

    // Visualize results
    visualizer.buildContextGeometry(&context);
    visualizer.setLightingModel(Visualizer::LIGHTING_PHONG_SHADOWED);
    visualizer.setCameraPosition(make_SphericalCoord(6,0.45,3.14), make_vec3(0,0,0));
    visualizer.plotUpdate();
    string joinedFilepath = outputsFilepath + "/bestDesignVisualization.jpg";
    const char* imageFilepath = joinedFilepath.c_str();
    visualizer.printWindow(imageFilepath);
    visualizer.closeWindow();

    cout << "Total Plant and Ground Radiation is: " << radiationSumOutput
    << " MJ/m2. Total Yield is: " << yieldOutput
    << " tons/acre. Total Power Production is: " << powerOutput
    << " W/m2. Total ET is: " << ETOutput
    << " mm/acre. Total Self Shading Hours is: " << selfShadingHoursOutput
    << " hours." << endl;

    //Print simulation outputs
    float values[] = {radiationSumOutput,yieldOutput,powerOutput,ETOutput,selfShadingHoursOutput};
    string performanceOutputs = outputsFilepath + "/outputFileBestDesign.csv";
    ofstream outputFile(performanceOutputs);
    if (outputFile.is_open()) {
        outputFile << "Radiation Sum (MJ/m2),Yield (tons/acre),Power (W/m2/season),ET (mm),Self Shading Hours (
     hrs)\n";
        for (int i = 0; i < sizeof(values) / sizeof(values[0]); ++i) {
            outputFile << values[i];
            if (i < sizeof(values) / sizeof(values[0]) - 1) {
                outputFile << ",";
            }
        }
        outputFile.close();
        cout << "Best Design performance outputs written to: " << performanceOutputs << endl;
    }
    else{
        cerr << "Error opening " << performanceOutputs << endl;
    }
```

```
579
580     string hourlyOutputs = outputsFilepath + "/hourlyDataBestDesign.csv";
581     ofstream hourlyFile(hourlyOutputs);
582     if(hourlyFile.is_open()){
583         hourlyFile << "Days After Planting,Julian Day,Hour,Hourly Power (W/m2/hr),"
584                       "Average Photosynthesic Assimilation (umol CO2/m2/hr),"
585                       "Average Evaporation (mm/hr),Average Transpiration (mm/hr),"
586                       "Average Evapotranspiration (mm/hr),Self Shading Hour (hr/hr)\n";
587         for(const auto& hourlyRow : hourlyData){
588             for(size_t i = 0; i < hourlyRow.size(); ++i){
589                 hourlyFile << fixed << setprecision(3) << hourlyRow[i];
590                 if(i < hourlyRow.size() - 1){
591                     hourlyFile << ",";
592                 }
593             }
594             hourlyFile << "\n";
595         }
596         hourlyFile.close();
597         cout << "Best Design hourly outputs written to: " << hourlyOutputs << endl;
598     }
599     else{
600         cerr << "Error opening " << hourlyOutputs << endl;
601     }
602
603     string dailyOutputs = outputsFilepath + "/dailyDataBestDesign.csv";
604     ofstream dailyFile(dailyOutputs);
605     if(dailyFile.is_open()){
606         dailyFile << "Days After Planting,Julian Day,Growing Degree Days (days C),"
607                      "Daily Power (W/m2),Accumulated C (umol CO2),Biomass (tons/acre),Evaporation Flux (mm),
     Transpiration Flux (mm),"
608                      "Evapotranspiration (mm),Self Shading Hours (hr/day)\n";
609         for(const auto& dailyRow : dailyData){
610             for(size_t i = 0; i < dailyRow.size(); ++i){
611                 dailyFile << fixed << setprecision(3) << dailyRow[i];
612                 if(i < dailyRow.size() - 1){
613                     dailyFile << ",";
614                 }
615             }
616             dailyFile << "\n";
617         }
618         dailyFile.close();
619         cout << "Best Design daily outputs written to: " << dailyOutputs << endl;
620     }
621     else{
622         cerr << "Error opening " << dailyOutputs << endl;
623     }
624
```

```
625     cout << "Best Design simulation finished." << endl;
626 
627     return 0;
628 }
```

Listing 4: bestDesign.cpp

Control Script

```
1  //THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
       TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
2  
3  #include "Context.h"
4  #include "EnergyBalanceModel.h"
5  #include "PhotosynthesisModel.h"
6  #include "StomatalConductanceModel.h"
7  #include "Visualizer.h"
8  #include "RadiationModel.h"
9  #include "SolarPosition.h"
10 #include "CanopyGenerator.h"
11 #include <iostream>
12 #include <thread>
13 #include <vector>
14 #include <fstream>
15 #include <sstream>
16 #include <string>
17 
18 using namespace helios;
19 using namespace std;
20 
21 string rootDirectory = //your root directory here;
22 
23 string informationFilepath = rootDirectory + "/geneticAlgorithm/inputs/informationFile.csv";
24 string cropDataFilepath = rootDirectory + "/geneticAlgorithm/inputs/cropDataFile.csv";
25 string weatherDataFilepath = rootDirectory + "/geneticAlgorithm/inputs/weatherDataFilepath.csv";
26 string outputsFilepath = rootDirectory + "/geneticAlgorithm/outputs";
27 
28 //custom enum for the smart csv reading
29 enum DataType {
30     TYPE_INT,
31     TYPE_FLOAT,
32     TYPE_STRING
33 };
34 
35 //function for reading csv files
```

```
void readCSVRow(const string &filepath, vector<pair<void *, DataType>> &destinations) {
    ifstream fileStream(filepath);
    if (!fileStream) {
        cerr << "Failed to open file." << endl;
        return;
    }
    string line;
    getline(fileStream, line);
    istringstream iss(line);
    string token;

    for (auto &dest : destinations) {
        getline(iss, token, ',');
        switch (dest.second) {
            case TYPE_INT:
                *static_cast<int *>(dest.first) = stoi(token);
                break;
            case TYPE_FLOAT:
                *static_cast<float *>(dest.first) = stof(token);
                break;
            case TYPE_STRING:
                *static_cast<string *>(dest.first) = token;
                break;
        }
    }
}

//Crop growth functions
//function for calculating the normalized temperature response function
//temporarily set to 1
float normalizedTResponse(float Tleaf, float Tmin, float Tmax, float Tr, float q){
    //return pow((Tleaf - Tmin) / (Tr - Tmin), q) * ((Tmax - Tleaf) / (Tmax - Tr));
    return 1;
}

//function for photosynthetic response
//Based on numeric solution
float photosynthesisResponse(float PAR, float Amax, float Ak, float Ard){
    return ((Amax * Ak * PAR) / ((Ak * PAR) + Amax)) - Ard;
}

//LAI function
float laiFunction(float GDD, float LAIa, float LAIk, float LAIS, float LAIH){
    return (LAIa * exp(-LAIk/GDD)) / (1 + exp(LAIS - (LAIH / GDD)));
}

//Function to calculate hydraulic conductivity
```

```
//Note: Actual Max Theta = ThetaMax + ThetaMin, adjust ThetaMax accordingly
//temporarily set to 1
float hydrualicConductivityFunction( float Theta, float ThetaMax, float bcoefficicient, float ThetaMin){
    //return (ThetaMax * exp(-bcoefficicient * Theta)) + ThetaMin;
    return 1;
}

//Function to calculate bare soil evaporation
//Uses Priestley-Taylor equation with certain assumptions (but can be overridden)
//temporarily set to 0
float evaporation(float Rn, float T, float p = 101.3, float alpha = 1.0, float G = 0){
    float delta = 4098 * (0.6108 * exp((17.27 * T)/(T+237.3))) / pow(T + 237.3,2);
    float gamma = 0.665 * pow(10,-3) * p;
    float E = alpha * delta * (Rn - G) / (delta + gamma);
    //return E;
    return 0;
}

//Function to calculate leaf transpiration
//Based on numeric solution
//temporarily set to 0
float transpiration(float PAR, float Transmax, float k, float Transd){
    //return ((Transmax * k * PAR) / ((k * PAR) + Transmax)) - Transd;
    return 0;
}

float growingDegreeDayFunction(float Tair, float Tbase, float Tmax, float dt){
    if((Tair > Tbase) && (Tair < Tmax)){
        return (Tair - Tbase) * (dt / 86400); //dt / seconds per day
    }
    else{
        return 0;
    }
}

int main(void){
    Context context;
    Visualizer visualizer(800);

    // Values in informationData
    int julStart, yearStart, seasonLength, UTC;
    float latitude, longitude;

    // Values in cropData
    float LAI_a, LAI_k, LAI_S, LAI_H, A_max, A_k, A_rd, Trans_max, Trans_k, Trans_rd,
        T_base, T_min, T_max, T_r, T_q, RUE, HI, dt, I;

```

```
    // Data in weatherFilepathData
    string weatherCSVPath;

    vector<pair<void *, DataType>> informationData = {
            {&julStart, TYPE_INT},
            {&yearStart, TYPE_INT},
            {&seasonLength, TYPE_INT},
            {&UTC, TYPE_INT},
            {&latitude, TYPE_FLOAT},
            {&longitude, TYPE_FLOAT}
    };

    vector<pair<void *, DataType>> cropData = {
            {&LAI_a, TYPE_FLOAT},
            {&LAI_k, TYPE_FLOAT},
            {&LAI_S, TYPE_FLOAT},
            {&LAI_H, TYPE_FLOAT},
            {&A_max, TYPE_FLOAT},
            {&A_k, TYPE_FLOAT},
            {&A_rd, TYPE_FLOAT},
            {&Trans_max, TYPE_FLOAT},
            {&Trans_k, TYPE_FLOAT},
            {&Trans_rd, TYPE_FLOAT},
            {&T_base, TYPE_FLOAT},
            {&T_min, TYPE_FLOAT},
            {&T_max, TYPE_FLOAT},
            {&T_r, TYPE_FLOAT},
            {&T_q, TYPE_FLOAT},
            {&RUE, TYPE_FLOAT},
            {&HI, TYPE_FLOAT},
            {&dt, TYPE_FLOAT},
            {&I, TYPE_FLOAT}
    };

    vector<pair<void *, DataType>> weatherFilePathData = {
            {&weatherCSVPath, TYPE_STRING}
    };

    julStart = 90;
    seasonLength = 120;
    T_base = 20;

    readCSVRow(informationFilepath, informationData);
    readCSVRow(cropDataFilepath, cropData);
    readCSVRow(weatherDataFilepath, weatherFilePathData);

    // Print the values for confirmation
```

```
177     cout << "Information Data:\n";
178     cout << "Julian Start: " << julStart << ", Start Year: " << yearStart << ", Season Length: " << seasonLength
179          << ", UTC: " << UTC << ", Latitude: " << latitude << ", Longtitude: " << longitude << endl;
180
181     cout << "\nCrop Data:\n";
182     cout << "LAI a: " << LAI_a << ", LAI k: " << LAI_k << ", LAI S: " << LAI_S
183          << ", LAI H: " << LAI_H << ", A max: " << A_max << ", A k: " << A_k
184          << ", A rd: " << A_rd << ", Transpiration max: " << Trans_max << ", Transpiration k: " << Trans_k
185          << ", Transpiration rd: " << Trans_rd << ", T base: " << T_base << ", T min: " << T_min
186          << ", T max: " << T_max << ", T r: " << T_r << ", T q: " << T_q
187          << ", RUE: " << RUE << ", HI: " << HI << ", dt: " << dt << ", Irrigation: " << I << "." << endl;
188
189     //add irrigation, and other water related parameters
190     cout << "\nWeather File Path:\n" << weatherCSVPath << endl;
191
192     context.loadTabularTimeseriesData(weatherCSVPath,{"CIMIS"},",");
193
194     vector<double> dailyRow;
195     vector<double> hourlyRow;
196
197     vector<vector<double>> dailyData;
198     vector<vector<double>> hourlyData;
199
200     float dailyDayNumber;
201     float hourNumber;
202     float growingDegreeDaysNumber;
203     float dailyA;
204     float dailyLAI;
205     float dailyET;
206     float hourlyDayNumber;
207     float hourlyA;
208     float hourlyET;
209
210     //variables for generating the ground
211     vec3 origin(0,0,0);
212     vec2 extent(3,3);
213     int2 nTiles(3,3);
214     int2 subpatches(25,25);
215
216     Date julDay;
217     Time time;
218
219     float pressure = 101300; //atmospheric pressure (Pa)
220     float temperature; //air temperature (K)
221     float humidity;    //relative humidity (-)
222     float precipitation; //precipiation (mm)
223     float turbidity = 0.05;  //atmospheric turbidity coeff (-)
```

```
    float Rflux;
    float fdiff;
    float RDirect;
    float RDiffuse;
    float reduction; //variable for water stress impact
    float hourlyPower = 0;
    float dailyPower = 0;
    float selfShadingHour = 0;
    float pvEfficiency = 0.2; //temporarily constant value, change later to read from data

    //temporary variable for calibration
    float accumulatedA = 0;

    context.cropDomainX(origin.x + make_vec2(-0.5 * extent.x, 0.5 * extent.x));
    context.cropDomainY(origin.y + make_vec2(-0.5 * extent.y, 0.5 * extent.y));

    //build canopy and ground
    CanopyGenerator cgen(&context);
    SphericalCrownsCanopyParameters params; //build spherical canopies
    float GDD = 0;
    float GDD_Sum = 0;
    float leafAreaIndex = laiFunction(GDD_Sum,LAI_a,LAI_k,LAI_S,LAI_H);
    cgen.seedRandomGenerator(10);
    params.plant_spacing = make_vec2(1.52, 0.3);
    params.plant_count = make_int2(2,10);
    params.crown_radius = make_vec3(1,0.17,.2);
    params.leaf_area_density = leafAreaIndex;
    cgen.buildCanopy(params);


    //variables for UUIDs
    vector<uint> groundUUIDs = context.addTile(make_vec3(0,0,0),extent,nullrotation, make_int2(10,10), RGBcolor
     (.341,.255,.184));
    vector<uint> leafUUIDs;
    vector<uint> allUUIDs;

    leafUUIDs = cgen.getLeafUUIDs();
    groundUUIDs = cgen.getGroundUUIDs();

    context.setPrimitiveData(groundUUIDs,"twosided_flat",uint(0));

    //initialize solar position
    SolarPosition solarposition( UTC, latitude, longitude, &context );

    // Get into radiation model
    RadiationModel radiation(&context);

```

```
    uint SunSource = radiation.addCollimatedRadiationSource(); //adding the source, direction added later

    radiation.addRadiationBand("SRad");
    radiation.setDirectRayCount("SRad",1000);
    radiation.disableEmission("SRad");

    radiation.enforcePeriodicBoundary("xy");

    context.setPrimitiveData(groundUUIDs,"twosided_flag",uint(0));

    radiation.updateGeometry();

    float biomass = 0;
    float groundDayFlux = 0;
    float leafDayFlux = 0;
    float leafTFlux = 0;
    float groundEFlux = 0;

    float daysAfterPlanting = 0;

    //start the loop
    cout << "Simulating for " << seasonLength << " days." << endl;
    for(int day = julStart; day < julStart + seasonLength; day++) {
        float dayF = static_cast<float>(day);
        julDay = make_Date(day,yearStart);
        context.setDate(julDay);

        context.deletePrimitive(leafUUIDs);
        params.leaf_area_density = leafAreaIndex;
        cgen.buildCanopy(params);
        leafUUIDs = cgen.getLeafUUIDs();
        radiation.updateGeometry();

        allUUIDs = context.getAllUUIDs();

        for(int hour = 0; hour < 24; hour++){
            context.setTime(0,0,hour); //set time
            vec3 sunDirection = solarposition.getSunDirectionVector(); //gets direction from the plugin
            radiation.setSourcePosition(SunSource, sunDirection); //set the source and direction

            //generate direct and diffuse radiation at timestep
            time = context.getTime();
            Rflux = context.queryTimeseriesData("net_radiation", julDay, time);
            temperature = context.queryTimeseriesData("air_temperature", julDay, time);
            humidity = context.queryTimeseriesData("air_humidity", julDay, time);
            precipitation = context.queryTimeseriesData("precipitation",julDay,time);

```

```
317             context.setPrimitiveData(allUUIDs,"air_temperature",temperature);
318             context.setPrimitiveData(allUUIDs,"air_humidity",humidity);
319 
320             fdiff = solarposition.getDiffuseFraction(pressure,temperature,humidity,turbidity); //this is a
    fraction of total flux
321             RDirect = Rflux*(1-fdiff);
322             RDiffuse = Rflux*fdiff;
323             cout << "Day: " << julDay << ". Time: " << time << endl;
324             cout << "CIMIS SRad: " << Rflux << " W/m2. CIMIS Air Temp: " << temperature << " C. CIMIS Relative
    Humidity: " << humidity << endl;
325             cout << "Diffuse fraction is: " << fdiff << ". Direct flux is: " << RDirect << " W/m2. Diffuse flux
    is: " << RDiffuse << " W/m2." << endl;
326             GDD = growingDegreeDayFunction(temperature,T_base,T_max,dt);
327             cout << "Air temperature is: " << temperature << ". Base temperature is: " << T_base
328             << ". Growing Degree Days are: " << GDD << " days C" << endl;
329             radiation.setSourceFlux(SunSource,"SRad",RDirect);
330             radiation.setDiffuseRadiationFlux("SRad",RDiffuse);
331 
332             context.setPrimitiveData(leafUUIDs, "relfectivity_SRad",0.05f);
333             context.setPrimitiveData(leafUUIDs,"transmissivity_SRad",0.05f);
334 
335             context.setPrimitiveData(groundUUIDs,"reflectivity_SRad",0.15f);
336             context.setPrimitiveData(groundUUIDs,"reflectivity_SRad",0.35f);
337 
338             radiation.runBand("SRad");
339 
340             float A_canopy = 0;
341             float T_canopy = 0;
342             for(uint UUID: leafUUIDs){
343                 float leafTotal, leafDirect, leafDiffuse, A, T;
344                 float area = context.getPrimitiveArea(UUID);
345                 context.getPrimitiveData(UUID, "radiation_flux_SRad", leafTotal);
346                 A = photosynthesisResponse(leafTotal * 2.5,A_max,A_k,A_rd) * area * dt;
347                 context.setPrimitiveData(UUID, "Photosynthesis", A);
348                 A_canopy += A;
349                 T = transpiration(leafTotal * 2.5,Trans_max,Trans_k,Trans_k) * 0.000018016 * 1000000 * area;
350                 context.setPrimitiveData(UUID, "Transpiration", T);
351                 T_canopy += T;
352             }
353 
354             accumulatedA += A_canopy;
355             biomass = accumulatedA * RUE;
356 
357             float AverageT;
358             context.calculatePrimitiveDataAreaWeightedMean(leafUUIDs, "Transpiration",AverageT);
359             float AverageA;
360             context.calculatePrimitiveDataAreaWeightedMean(leafUUIDs,"Photosynthesis",AverageA);
```

```
361
362             float E_ground = 0;
363             for(uint UUID: groundUUIDs){
364                 float rad_Ground, Evap, alpha;
365                 float area = context.getPrimitiveArea(UUID);
366                 context.getPrimitiveData(UUID, "radiation_flux_SRad", rad_Ground);
367                 Evap = evaporation(rad_Ground,temperature) * area;
368                 context.setPrimitiveData(UUID, "Evaporation", Evap);
369                 E_ground += Evap;
370             }
371
372             float AverageE;
373             context.calculatePrimitiveDataAreaWeightedMean(groundUUIDs,"Evaporation",AverageE);
374
375             float AWater_canopy = A_canopy * 0.000018016 * 1000000;
376             float WaterDemand = E_ground + T_canopy + AWater_canopy;
377             float AvailableWater = I + precipitation;
378
379             if(WaterDemand > AvailableWater){
380                 if(WaterDemand - AvailableWater < 0){
381                     reduction = 0;
382                 }
383                 else{
384                     reduction = 1 - ((WaterDemand - AvailableWater)/ WaterDemand);
385                 }
386                 GDD = GDD * reduction;
387                 A_canopy = A_canopy * reduction;
388                 AverageA = AverageA * reduction;
389                 AverageE = AverageE * reduction;
390                 AverageT = AverageT * reduction;
391             }
392
393             leafTFlux += T_canopy;
394             groundEFlux += E_ground;
395             GDD_Sum = GDD_Sum + GDD;
396
397             hourlyPower = Rflux * .2;
398             dailyPower += hourlyPower;
399
400             float hourF = static_cast<float>(hour);
401             hourlyRow = {daysAfterPlanting,dayF,hourF,hourlyPower,AverageA,AverageE,AverageT,
402                          AverageE+AverageT,0};
403             hourlyData.push_back(hourlyRow);
404
405             float leafHourFlux;
406             context.calculatePrimitiveDataAreaWeightedSum(leafUUIDs,"radiation_flux_SRad",leafHourFlux);
407             leafDayFlux += leafHourFlux * 3600 / 1e6 / (1 / (4 / (3 * M_PI * 0.5f*0.5f*0.5f)));
```

```
            float groundHourFlux;
            context.calculatePrimitiveDataAreaWeightedSum(groundUUIDs,"radiation_flux_SRad",groundHourFlux);
            groundDayFlux += groundHourFlux * 3600 / 1e6 / 9;
            cout << "Hourly flux on leafs is: " << leafHourFlux << " W. Hourly flux on ground is: " <<
    groundHourFlux << " W." << endl;
            cout << "Daily flux on leafs is: " << leafDayFlux << " MJ/m2. Daily flux on ground is: " <<
    groundDayFlux << " MJ/m2." << endl;
        }
        //add values to the vector
        dailyRow = {daysAfterPlanting,dayF,GDD_Sum,dailyPower,accumulatedA,biomass,groundEFlux,leafTFlux,
                    leafTFlux+groundEFlux,0};
        dailyData.push_back(dailyRow);

        leafAreaIndex = laiFunction(GDD_Sum,LAI_a,LAI_k,LAI_S,LAI_H);
        //leafAreaIndex = 5;
        cout << "GDD Sum = " << GDD_Sum << endl;
        cout << "LAI = " << leafAreaIndex << endl;

        daysAfterPlanting += 1;
    }

    // Visualize results
    visualizer.buildContextGeometry(&context);
    visualizer.setLightingModel(Visualizer::LIGHTING_PHONG_SHADOWED);
    visualizer.setCameraPosition(make_SphericalCoord(6,0.45,3.14), make_vec3(0,0,0));
    visualizer.plotUpdate();
    string joinedFilepath = outputsFilepath + "/controlVisualization.jpg";
    const char* imageFilepath = joinedFilepath.c_str();
    visualizer.printWindow(imageFilepath);
    visualizer.closeWindow();

    float radiationSum = leafDayFlux + groundDayFlux;
    float yield = biomass * HI;
    float power = dailyPower;
    float ET = groundEFlux + leafTFlux;
    float selfShadingHours = 0;

    cout << "Total Plant and Ground Radiation is: " << radiationSum
    << " MJ/m2. Total Yield is: " << yield
    << " tons/acre. Total Power Production is: " << power
    << " W/m2. Total ET is: " << ET
    << " mm/acre. Total Self Shading Hours is: " << selfShadingHours
    << " hours." << endl;

    //Print simulation outputs
    float values[] = {radiationSum,yield,power,ET,selfShadingHours};
    string performanceOutputs = outputsFilepath + "/outputFileControl.csv";
```

```
453     ofstream outputFile(performanceOutputs);
454     if (outputFile.is_open()) {
455         outputFile << "Radiation Sum (MJ/m2),Yield (tons/acre),Power (W/m2/season),ET (mm),-----\n";
456         for (int i = 0; i < sizeof(values) / sizeof(values[0]); ++i) {
457             outputFile << values[i];
458             if (i < sizeof(values) / sizeof(values[0]) - 1) {
459                 outputFile << ",";
460             }
461         }
462         outputFile.close();
463         cout << "Control performance outputs written to: " << performanceOutputs << endl;
464     }
465     else{
466         cerr << "Error opening " << performanceOutputs << endl;
467     }
468 
469     string hourlyOutputs = outputsFilepath + "/hourlyDataControl.csv";
470     ofstream hourlyFile(hourlyOutputs);
471     if(hourlyFile.is_open()){
472         hourlyFile << "Days After Planting,Julian Day,Hour,Hourly Power (W/m2/hr),"
473                       "Average Photosynthesic Assimilation (umol CO2/m2/hr),"
474                       "Average Evaporation (mm/hr),Average Transpiration (mm/hr),"
475                       "Average Evapotranspiration (mm/hr),Self Shading Hour (hr/hr)\n";
476         for(const auto& hourlyRow : hourlyData){
477             for(size_t i = 0; i < hourlyRow.size(); ++i){
478                 hourlyFile << fixed << setprecision(3) << hourlyRow[i];
479                 if(i < hourlyRow.size() - 1){
480                     hourlyFile << ",";
481                 }
482             }
483             hourlyFile << "\n";
484         }
485         hourlyFile.close();
486         cout << "Control hourly outputs written to: " << hourlyOutputs << endl;
487     }
488     else{
489         cerr << "Error opening " << hourlyOutputs << endl;
490     }
491 
492     string dailyOutputs = outputsFilepath + "/dailyDataControl.csv";
493     ofstream dailyFile(dailyOutputs);
494     if(dailyFile.is_open()){
495         dailyFile << "Days After Planting,Julian Day,Growing Degree Days (days C),"
496                      "Daily Power (W/m2),Accumulated C (umol/m2),Biomass (tons/acre),Evaporation Flux (mm),
     Transpiration Flux (mm),"
497                      "Evapotranspiration (mm),Self Shading Hours (hr/day)\n";
498         for(const auto& dailyRow : dailyData){
```

```
            for(size_t i = 0; i < dailyRow.size(); ++i){
                dailyFile << fixed << setprecision(3) << dailyRow[i];
                if(i < dailyRow.size() - 1){
                    dailyFile << ",";
                }
            }
            dailyFile << "\n";
        }
        dailyFile.close();
        cout << "Control daily outputs written to: " << dailyOutputs << endl;
    }
    else{
        cerr << "Error opening " << dailyOutputs << endl;
    }

    cout << "Control simulation finished." << endl;

    return 0;
}
```

Listing 5: control.cpp

Agrivoltaic Script

```
//THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

#include "Context.h"
#include "EnergyBalanceModel.h"
#include "PhotosynthesisModel.h"
#include "StomatalConductanceModel.h"
#include "Visualizer.h"
#include "RadiationModel.h"
#include "SolarPosition.h"
#include "CanopyGenerator.h"
#include <iostream>
#include <thread>
#include <vector>
#include <fstream>
#include <sstream>
#include <string>

using namespace helios;
using namespace std;
string rootDirectory = //your root directory here;
```

```
21
22 string informationFilepath = rootDirectory + "/geneticAlgorithm/inputs/informationFile.csv";
23 string cropDataFilepath = rootDirectory + "/geneticAlgorithm/inputs/cropDataFile.csv";
24 string weatherDataFilepath = rootDirectory + "/geneticAlgorithm/inputs/weatherDataFilepath.csv";
25 string outputsFilepath = rootDirectory + "/geneticAlgorithm/outputs";
26 string inputPath = rootDirectory + "/geneticAlgorithm/inputs";
27 const char* panelTexture = "F:/Helios/Helios/projects/geneticAlgorithm/inputs/solarpanel.png";
28
29 //custom enum for the smart csv reading
30 enum DataType {
31     TYPE_INT,
32     TYPE_FLOAT,
33     TYPE_STRING
34 };
35
36 struct Point {
37     float x;
38     float y;
39     float z;
40 };
41
42 //function for reading csv files
43 void readCSVRow(const string &filepath, vector<pair<void *, DataType>> &destinations) {
44     ifstream fileStream(filepath);
45     if (!fileStream) {
46         cerr << "Failed to open file." << endl;
47         return;
48     }
49     string line;
50     getline(fileStream, line);
51     istringstream iss(line);
52     string token;
53
54     for (auto &dest : destinations) {
55         getline(iss, token, ',');
56         switch (dest.second) {
57             case TYPE_INT:
58                 *static_cast<int *>(dest.first) = stoi(token);
59                 break;
60             case TYPE_FLOAT:
61                 *static_cast<float *>(dest.first) = stof(token);
62                 break;
63             case TYPE_STRING:
64                 *static_cast<string *>(dest.first) = token;
65                 break;
66         }
67     }
```

```
}

vector<Point> readLocationCSV(const string& filepath) {
    vector<Point> data;
    ifstream fileStream(filepath);

    if (!fileStream) {
        cerr << "Failed to open file." << endl;
        return data; // Return empty vector
    }

    string line;
    while (getline(fileStream, line)) {
        istringstream iss(line);
        Point pt;
        char delimiter;

        iss >> pt.x >> delimiter >> pt.y >> delimiter >> pt.z;

        data.push_back(pt);
    }

    return data;
}

//Crop growth functions
//function for calculating the normalized temperature response function
//set to 1 for now
float normalizedTResponse(float Tleaf, float Tmin, float Tmax, float Tr, float q){
    //return pow((Tleaf - Tmin) / (Tr - Tmin), q) * ((Tmax - Tleaf) / (Tmax - Tr));
    return 1;
}

//function for photosynthetic response
//Based on numeric solution
float photosynthesisResponse(float PAR, float Amax, float Ak, float Ard){
    return ((Amax * Ak * PAR) / ((Ak * PAR) + Amax)) - Ard;
}

//LAI function
float laiFunction(float GDD, float LAIa, float LAIk, float LAIS, float LAIH){
    return (LAIa * exp(-LAIk/GDD)) / (1 + exp(LAIS - (LAIH / GDD)));
}

//Function to calculate hydraulic conductivity
//Note: Actual Max Theta = ThetaMax + ThetaMin, adjust ThetaMax accordingly
//set to 1 for now
```

```
115 float hydrualicConductivityFunction( float Theta, float ThetaMax, float bcoefficicient, float ThetaMin){
116     //return (ThetaMax * exp(-bcoefficicient * Theta)) + ThetaMin;
117     return 1;
118 }
119 
120 //Function to calculate bare soil evaporation
121 //Uses Priestley-Taylor equation with certain assumptions (but can be overridden)
122 //set to 0 for now
123 float evaporation(float Rn, float T, float p = 101.3, float alpha = 1.0, float G = 0){
124     float delta = 4098 * (0.6108 * exp((17.27 * T)/(T+237.3))) / pow(T + 237.3,2);
125     float gamma = 0.665 * pow(10,-3) * p;
126     float E = alpha * delta * (Rn - G) / (delta + gamma);
127     //return E;
128     return 0;
129 }
130 
131 //Function to calculate leaf transpiration
132 //Based on numeric solution
133 //set to 0 for now
134 float transpiration(float PAR, float Transmax, float k, float Transd){
135     //return ((Transmax * k * PAR) / ((k * PAR) + Transmax)) - Transd;
136     return 0;
137 }
138 
139 float growingDegreeDayFunction(float Tair, float Tbase, float Tmax, float dt){
140     if((Tair > Tbase) && (Tair < Tmax)){
141         return (Tair - Tbase) * (dt / 86400); //dt / seconds per day
142     }
143     else{
144         return 0;
145     }
146 }
147 
148 int main(int argc, char* argv[]) {
149     if (argc < 6) {  // We need 5 arguments: program name + 5 values
150         cerr << "Please provide five values." << endl;
151         return 1;
152     }
153 
154     int threadNumber = atoi(argv[1]);
155     int Generations = atoi(argv[2]);
156     int Strings = atoi(argv[3]);
157     int K = atoi(argv[4]);
158     int P = atoi(argv[4]);
159     cout << "Received values for Thread: " << threadNumber << ", Generations: " << Generations
160          << ", Strings: " << Strings << ", K: " << K << ", P: " << P << endl;
161 
```

```
    //strings for solar design files
    string designFile = inputPath + "/designFile" + to_string(threadNumber) + ".csv";
    string coordinateFile = inputPath + "/coordinateFile" + to_string(threadNumber) + ".csv";
    cout << "Thread " << threadNumber << " is running " << designFile << endl;


    Context context;

    // Values in informationData
    int julStart, yearStart, seasonLength, UTC;
    float latitude, longitude;

    // Values in cropData
    float LAI_a, LAI_k, LAI_S, LAI_H, A_max, A_k, A_rd, Trans_max, Trans_k, Trans_rd,
            T_base, T_min, T_max, T_r, T_q, RUE, HI, dt, I;

    // Values in DesignData
    float numberPanels, lengthPanels, widthPanels, heightPanels, azimuthPanels, elevationPanels;
    float reflectivityPanels, efficiencyPanels;

    // Data in weatherFilepathData
    string weatherCSVPath;

    vector<pair<void *, DataType>> informationData = {
            {&julStart, TYPE_INT},
            {&yearStart, TYPE_INT},
            {&seasonLength, TYPE_INT},
            {&UTC, TYPE_INT},
            {&latitude, TYPE_FLOAT},
            {&longitude, TYPE_FLOAT}
    };

    vector<pair<void *, DataType>> cropData = {
            {&LAI_a, TYPE_FLOAT},
            {&LAI_k, TYPE_FLOAT},
            {&LAI_S, TYPE_FLOAT},
            {&LAI_H, TYPE_FLOAT},
            {&A_max, TYPE_FLOAT},
            {&A_k, TYPE_FLOAT},
            {&A_rd, TYPE_FLOAT},
            {&Trans_max, TYPE_FLOAT},
            {&Trans_k, TYPE_FLOAT},
            {&Trans_rd, TYPE_FLOAT},
            {&T_base, TYPE_FLOAT},
            {&T_min, TYPE_FLOAT},
            {&T_max, TYPE_FLOAT},
            {&T_r, TYPE_FLOAT},
```

```
            {&T_q, TYPE_FLOAT},
            {&RUE, TYPE_FLOAT},
            {&HI, TYPE_FLOAT},
            {&dt, TYPE_FLOAT},
            {&I, TYPE_FLOAT}
    };

    vector<pair<void *, DataType>> designData = {
            {&numberPanels, TYPE_INT},
            {&lengthPanels, TYPE_FLOAT},
            {&widthPanels, TYPE_FLOAT},
            {&heightPanels, TYPE_FLOAT},
            {&azimuthPanels, TYPE_FLOAT},
            {&elevationPanels, TYPE_FLOAT},
            {&reflectivityPanels, TYPE_FLOAT},
            {&efficiencyPanels, TYPE_FLOAT},
    };

    vector<pair<void *, DataType>> weatherFilePathData = {
            {&weatherCSVPath, TYPE_STRING}
    };

    julStart = 90;
    seasonLength = 120;
    T_base = 20;

    readCSVRow(informationFilepath, informationData);
    readCSVRow(cropDataFilepath, cropData);
    readCSVRow(designFile,designData);
    readCSVRow(weatherDataFilepath, weatherFilePathData);

    vector<Point> coordinates = readLocationCSV(coordinateFile);

    // Print the values for confirmation
    cout << "Information Data:\n";
    cout << "Julian Start: " << julStart << ", Start Year: " << yearStart << ", Season Length: " << seasonLength
         << ", UTC: " << UTC << ", Latitude: " << latitude << ", Longtitude: " << longitude << endl;

    cout << "\nCrop Data:\n";
    cout << "LAI a: " << LAI_a << ", LAI k: " << LAI_k << ", LAI S: " << LAI_S
         << ", LAI H: " << LAI_H << ", A max: " << A_max << ", A k: " << A_k
         << ", A rd: " << A_rd << ", Transpiration max: " << Trans_max << ", Transpiration k: " << Trans_k
         << ", Transpiration rd: " << Trans_rd << ", T base: " << T_base << ", T min: " << T_min
         << ", T max: " << T_max << ", T r: " << T_r << ", T q: " << T_q
         << ", RUE: " << RUE << ", HI: " << HI << ", dt: " << dt << ", Irrigation: " << I << "." << endl;

    //add irrigation, and other water related parameters
```

```
    cout << "\nWeather File Path:\n" << weatherCSVPath << endl;

    context.loadTabularTimeseriesData(weatherCSVPath,{"CIMIS"},",");

    vector<double> dailyRow;
    vector<double> hourlyRow;

    vector<vector<double>> dailyData;
    vector<vector<double>> hourlyData;

    float dailyDayNumber;
    float hourNumber;
    float growingDegreeDaysNumber;
    float dailyA;
    float dailyLAI;
    float dailyET;
    float hourlyDayNumber;
    float hourlyA;
    float hourlyET;

    //variables for generating the ground
    vec3 origin(0,0,0);
    vec2 extent(3,3);
    int2 nTiles(3,3);
    int2 subpatches(25,25);

    Date julDay;
    Time time;

    float pressure = 101300; //atmospheric pressure (Pa)
    float temperature; //air temperature (K)
    float humidity;    //relative humidity (-)
    float precipitation; //precipiation (mm)
    float turbidity = 0.05;  //atmospheric turbidity coeff (-)
    float Rflux;
    float fdiff;
    float RDirect;
    float RDiffuse;
    float reduction; //variable for water stress impact
    float hourlyPower = 0;
    float dailyPower = 0;
    float selfShadingHour = 0;
    float dailySelfShadingHours = 0;
    float pvEfficiency = 0.2; //temporarily constant value, change later to read from data

    //temporary variable for calibration
    float accumulatedA = 0;
```

```
303
304     //values for self shading hours search
305     float min_val;
306     float max_val;
307     float ave_val;
308     float ssh_ave;
309
310     //values for output file
311     float radiationSumOutput;
312     float yieldOutput;
313     float powerOutput;
314     float ETOutput;
315     float selfShadingHoursOutput;
316
317     context.cropDomainX(origin.x + make_vec2(-0.5 * extent.x, 0.5 * extent.x));
318     context.cropDomainY(origin.y + make_vec2(-0.5 * extent.y, 0.5 * extent.y));
319
320     //build canopy and ground
321     CanopyGenerator cgen(&context);
322     SphericalCrownsCanopyParameters params; //build spherical canopies
323     float GDD = 0;
324     float GDD_Sum = 0;
325     float leafAreaIndex = laiFunction(GDD_Sum,LAI_a,LAI_k,LAI_S,LAI_H);
326     cgen.seedRandomGenerator(10);
327     params.plant_spacing = make_vec2(1.52, 0.3);
328     params.plant_count = make_int2(2,10);
329     params.crown_radius = make_vec3(1,0.17,.2);
330     params.leaf_area_density = leafAreaIndex;
331     cgen.buildCanopy(params);
332
333     //variables for UUIDs
334     vector<uint> groundUUIDs = context.addTile(make_vec3(0,0,0),extent,nullrotation, make_int2(10,10), RGBcolor
        (.341,.255,.184));
335     vector<uint> leafUUIDs;
336     vector<uint> panelUUIDs;
337     vector<uint> allUUIDs;
338
339     leafUUIDs = cgen.getLeafUUIDs();
340     groundUUIDs = cgen.getGroundUUIDs();
341
342     //Generate solar panels
343     vec2 panelSize = make_vec2(lengthPanels/100,widthPanels/100);
344     cout << "Panel length: " << lengthPanels/100 << " m, Panel width: " << widthPanels/100
345          << " m, Panel height: "<< heightPanels/100 << " m, Panel elevation: " << elevationPanels
346          << " degrees, azimuth: " << azimuthPanels << " degrees." << endl;
347     float elevationRad = 2*M_PI*(elevationPanels/360);
348     float azimuthRad = 2*M_PI*(azimuthPanels/360);
```

```
    SphericalCoord panelAngle = make_SphericalCoord(elevationRad,azimuthRad);
    int2 panelSubDivisions = make_int2(10,10);
    for (const Point& pt : coordinates) {
        vec3 panelLocation = make_vec3(pt.x,pt.y,pt.z/100);
        cout << "Panel location is: " << panelLocation << endl;
        panelUUIDs = context.addTile(panelLocation, panelSize, panelAngle, panelSubDivisions, panelTexture);
    }

    context.setPrimitiveData(groundUUIDs,"twosided_flat",uint(0));

    //initialize solar position
    SolarPosition solarposition( UTC, latitude, longitude, &context );

    // Get into radiation model
    RadiationModel radiation(&context);
    radiation.disableMessages();

    uint SunSource = radiation.addCollimatedRadiationSource(); //adding the source, direction added later

    radiation.addRadiationBand("SRad");
    radiation.setDirectRayCount("SRad",1000);
    radiation.disableEmission("SRad");

    radiation.enforcePeriodicBoundary("xy");

    context.setPrimitiveData(groundUUIDs,"twosided_flag",uint(0));

    radiation.updateGeometry();

    float biomass = 0;
    float groundDayFlux = 0;
    float leafDayFlux = 0;
    float leafTFlux = 0;
    float groundEFlux = 0;

    float daysAfterPlanting = 0;

    //start the loop
    cout << "Simulating for " << seasonLength << " days." << endl;
    for(int day = julStart; day < julStart + seasonLength; day++) {
        float dayF = static_cast<float>(day);
        julDay = make_Date(day,yearStart);
        context.setDate(julDay);

        context.deletePrimitive(leafUUIDs);
        params.leaf_area_density = leafAreaIndex;
        cgen.buildCanopy(params);
```

```
396         leafUUIDs = cgen.getLeafUUIDs();
397         radiation.updateGeometry();
398 
399         allUUIDs = context.getAllUUIDs();
400 
401         for(int hour = 0; hour < 24; hour++){
402             context.setTime(0,0,hour); //set time
403             vec3 sunDirection = solarposition.getSunDirectionVector(); //gets direction from the plugin
404             radiation.setSourcePosition(SunSource, sunDirection); //set the source and direction
405 
406             //generate direct and diffuse radiation at timestep
407             time = context.getTime();
408             Rflux = context.queryTimeseriesData("net_radiation", julDay, time);
409             temperature = context.queryTimeseriesData("air_temperature", julDay, time);
410             humidity = context.queryTimeseriesData("air_humidity", julDay, time);
411             precipitation = context.queryTimeseriesData("precipitation",julDay,time);
412 
413             context.setPrimitiveData(allUUIDs,"air_temperature",temperature);
414             context.setPrimitiveData(allUUIDs,"air_humidity",humidity);
415 
416             fdiff = solarposition.getDiffuseFraction(pressure,temperature,humidity,turbidity); //this is a
    fraction of total flux
417             RDirect = Rflux*(1-fdiff);
418             RDiffuse = Rflux*fdiff;
419             cout << "Day: " << julDay << ". Time: " << time << endl;
420             cout << "CIMIS SRad: " << Rflux << " W/m2. CIMIS Air Temp: " << temperature << " C. CIMIS Relative
    Humidity: " << humidity << endl;
421             cout << "Diffuse fraction is: " << fdiff << ". Direct flux is: " << RDirect << " W/m2. Diffuse flux
    is: " << RDiffuse << " W/m2." << endl;
422             GDD = growingDegreeDayFunction(temperature,T_base,T_max,dt);
423             cout << "Air temperature is: " << temperature << ". Base temperature is: " << T_base
424                  << ". Growing Degree Days are: " << GDD << " days C" << endl;
425             radiation.setSourceFlux(SunSource,"SRad",RDirect);
426             radiation.setDiffuseRadiationFlux("SRad",RDiffuse);
427 
428             context.setPrimitiveData(leafUUIDs, "relfectivity_SRad",0.05f);
429             context.setPrimitiveData(leafUUIDs,"transmissivity_SRad",0.05f);
430 
431             context.setPrimitiveData(groundUUIDs,"reflectivity_SRad",0.15f);
432             context.setPrimitiveData(groundUUIDs,"reflectivity_SRad",0.35f);
433 
434             radiation.runBand("SRad");
435 
436             float A_canopy = 0;
437             float T_canopy = 0;
438             for(uint UUID: leafUUIDs){
439                 float leafTotal, leafDirect, leafDiffuse, A, T;
```

```
            float area = context.getPrimitiveArea(UUID);
            context.getPrimitiveData(UUID, "radiation_flux_SRad", leafTotal);
            A = photosynthesisResponse(leafTotal * 2.5,A_max,A_k,A_rd) * area * dt;
            context.setPrimitiveData(UUID, "Photosynthesis", A);
            A_canopy += A;
            T = transpiration(leafTotal * 2.5,Trans_max,Trans_k,Trans_k) * 0.000018016 * 1000000 * area;
            context.setPrimitiveData(UUID, "Transpiration", T);
            T_canopy += T;
        }

        accumulatedA += A_canopy;
        biomass = accumulatedA * RUE;

        float AverageT;
        context.calculatePrimitiveDataAreaWeightedMean(leafUUIDs, "Transpiration",AverageT);
        float AverageA;
        context.calculatePrimitiveDataAreaWeightedMean(leafUUIDs,"Photosynthesis",AverageA);

        float E_ground = 0;
        for(uint UUID: groundUUIDs){
            float rad_Ground, Evap, alpha;
            float area = context.getPrimitiveArea(UUID);
            context.getPrimitiveData(UUID, "radiation_flux_SRad", rad_Ground);
            Evap = evaporation(rad_Ground,temperature) * area;
            context.setPrimitiveData(UUID, "Evaporation", Evap);
            E_ground += Evap;
        }

        float AverageE;
        context.calculatePrimitiveDataAreaWeightedMean(groundUUIDs,"Evaporation",AverageE);

        float AWater_canopy = A_canopy * 0.000018016 * 1000000;
        float WaterDemand = E_ground + T_canopy + AWater_canopy;
        float AvailableWater = I + precipitation;

        if(WaterDemand > AvailableWater){
            if(WaterDemand - AvailableWater < 0){
                reduction = 0;
            }
            else{
                reduction = 1 - ((WaterDemand - AvailableWater)/ WaterDemand);
            }
            GDD = GDD * reduction;
            A_canopy = A_canopy * reduction;
            AverageA = AverageA * reduction;
            AverageE = AverageE * reduction;
            AverageT = AverageT * reduction;
```

```
            }

            hourlyPower = 0;
            vector<float> SSHcalc;
            for(uint UUID: panelUUIDs){
                float rad_Panel, power;
                float area = context.getPrimitiveArea(UUID);
                context.getPrimitiveData(UUID, "radiation_flux_SRad", rad_Panel);
                power = rad_Panel * efficiencyPanels;
                context.setPrimitiveData(UUID, "Power", power);
                SSHcalc.push_back(power);
            }

            context.calculatePrimitiveDataAreaWeightedMean(panelUUIDs,"Power",hourlyPower);

            min_val = SSHcalc[0];
            max_val = SSHcalc[0];
            ave_val = 0;

            for(float value: SSHcalc){
                min_val = min(min_val,value);
                max_val = max(max_val,value);
                ave_val += value;
            }

            ssh_ave = ave_val / static_cast<float>(SSHcalc.size());

            selfShadingHour = 0;
            if(min_val < 0.25 * max_val){
                selfShadingHour += 1;
            }

            leafTFlux += T_canopy;
            groundEFlux += E_ground;
            GDD_Sum = GDD_Sum + GDD;

            dailyPower += hourlyPower;
            dailySelfShadingHours += selfShadingHour;

            float hourF = static_cast<float>(hour);
            hourlyRow = {daysAfterPlanting,dayF,hourF,hourlyPower,AverageA,AverageE,AverageT,
                         AverageE+AverageT,selfShadingHour};
            hourlyData.push_back(hourlyRow);

            float leafHourFlux;
            context.calculatePrimitiveDataAreaWeightedSum(leafUUIDs,"radiation_flux_SRad",leafHourFlux);
            leafDayFlux += leafHourFlux * 3600 / 1e6 / (1 / (4 / (3 * M_PI * 0.5f*0.5f*0.5f)));
```

```
            float groundHourFlux;
            context.calculatePrimitiveDataAreaWeightedSum(groundUUIDs,"radiation_flux_SRad",groundHourFlux);
            groundDayFlux += groundHourFlux * 3600 / 1e6 / 9;
            cout << "Hourly flux on leafs is: " << leafHourFlux << " W. Hourly flux on ground is: " <<
    groundHourFlux << " W." << endl;
            cout << "Daily flux on leafs is: " << leafDayFlux << " MJ/m2. Daily flux on ground is: " <<
    groundDayFlux << " MJ/m2." << endl;
        }

        //add values to the vector
        dailyRow = {daysAfterPlanting,dayF,GDD_Sum,dailyPower,accumulatedA,biomass,groundEFlux,leafTFlux,
                    leafTFlux+groundEFlux,dailySelfShadingHours};
        dailyData.push_back(dailyRow);

        leafAreaIndex = laiFunction(GDD_Sum,LAI_a,LAI_k,LAI_S,LAI_H);
        cout << "GDD Sum = " << GDD_Sum << endl;
        cout << "LAI = " << leafAreaIndex << endl;

        daysAfterPlanting += 1;
    }

    radiationSumOutput = leafDayFlux + groundDayFlux;
    yieldOutput = biomass * HI;
    powerOutput = dailyPower;
    ETOutput = groundEFlux + leafTFlux;
    selfShadingHoursOutput = dailySelfShadingHours;

    //Print simulation outputs
    float values[] = {radiationSumOutput,yieldOutput,powerOutput,ETOutput,selfShadingHoursOutput};
    string outputs = outputsFilepath + "/outputFile" + to_string(threadNumber) + ".csv";
    ofstream outputFile(outputs);
    if (outputFile.is_open()) {
        for (int i = 0; i < sizeof(values) / sizeof(values[0]); ++i) {
            outputFile << values[i];
            if (i < sizeof(values) / sizeof(values[0]) - 1) {
                outputFile << ",";
            }
        }
        outputFile.close();
    }
    else{
        cerr << "Error opening the file." << endl;
    }
    cout << "simulation " << threadNumber << " finished." << endl;

    Visualizer visualizer(800);
    visualizer.buildContextGeometry(&context);
```

```
    visualizer.setLightingModel(Visualizer::LIGHTING_PHONG_SHADOWED);
    visualizer.setCameraPosition(make_SphericalCoord(10,0.45,3.14), make_vec3(0,0,0));
    visualizer.plotUpdate();
    string imagePathString = outputsFilepath + "/design_" + to_string(threadNumber) + "_Visualization.jpg";
    const char* imagePath = imagePathString.c_str();
    visualizer.printWindow(imagePath);
    visualizer.closeWindow();
}
```

Listing 6: agrivoltaic.cpp

References

[Abdel-Mawgoud et al., 1995] Abdel-Mawgoud, A., El-Abd, S., Singer, S., Abou-Hadid, A., and Hsiao, T. (1995). Effect of shade on the growth and yield of tomato plants. *Strategies for Market Oriented Greenhouse Production 434*, pages 313–320.

[AL-agele et al., 2021] AL-agele, H. A., Proctor, K., Murthy, G., and Higgins, C. (2021). A case study of tomato (solanum lycopersicon var. legend) production and water productivity in agrivoltaic systems. *Sustainability*, 13(5):2850.

[Al Mamun et al., 2022] Al Mamun, M. A., Dargusch, P., Wadley, D., Zulkarnain, N. A., and Aziz, A. A. (2022). A review of research on agrivoltaic systems. *Renewable and Sustainable Energy Reviews*, 161:112351.

[Ali Khan Niazi and Victoria, 2023] Ali Khan Niazi, K. and Victoria, M. (2023). Comparative analysis of photovoltaic configurations for agrivoltaic systems in europe. *Progress in Photovoltaics: Research and Applications*.

[Allen, 2005] Allen, R. (2005). Penman–monteith equation. In Hillel, D., editor, *Encyclopedia of Soils in the Environment*, pages 180–188. Elsevier, Oxford.

[Amaducci et al., 2018] Amaducci, S., Yin, X., and Colauzzi, M. (2018). Agrivoltaic systems to optimise land use for electric energy production. *Applied energy*, 220:545–561.

[Argonne National Laboratory, 2023] Argonne National Laboratory (2023). Geospatial energy mapper (gem).

[Bailey, 2022] Bailey, B. (2022). Helios simulation software documentation. `https://baileylab.ucdavis.edu/software/helios/_overview.html`. Accessed: November 2, 2023.

[Bailey, 2018] Bailey, B. N. (2018). A reverse ray-tracing method for modelling the net radiative flux in leaf-resolving plant canopy simulations. *Ecological Modelling*, 368:233–245.

[Bailey, 2019] Bailey, B. N. (2019). Helios: A scalable 3d plant and environmental biophysical modeling framework. *Frontiers in Plant Science*, 10.

[CADOC, 2024] CADOC (2024). Important farmland. Accessed: 2024-06-11.

[California Air Resources Board, 2022a] California Air Resources Board (2022a). Advanced clean cars ii. An overview of the Advanced Clean Cars II program, its regulations, emphasis on environmental justice and equity, incentive programs, and the background of CARB's efforts to reduce motor vehicle emissions.

[California Air Resources Board, 2022b] California Air Resources Board (2022b). Current california ghg emission inventory data.

[California Department of Food and Agriculture, 2022] California Department of Food and Agriculture (2022). California agricultural production statistics 2022. Statistics related to California's top agricultural commodities, cash receipts, agricultural exports, and organic product sales.

[California Department of Water Resources, 2014] California Department of Water Resources (2014). Sustainable groundwater management act: Ab 1739, sb 1168, sb 1319. Government Publication.

[California Department of Water Resources, 2021] California Department of Water Resources (2021). California irrigation managment information system.

[California Department of Water Resources, 2023a] California Department of Water Resources (2023a). Groundwater depth seasonal points.

[California Department of Water Resources, 2023b] California Department of Water Resources (2023b). Water shortage vulnerability technical methods.

[California Energy Commission, 2016] California Energy Commission (2018, 2006, 2016, 2016). Ab 3232, ab 32, sb 32, ab 197. Government Publication.

[California Energy Commission, 2022] California Energy Commission (2022). 2022 total system electric generation. This dataset presents the total utility-scale electric generation for California in 2022, detailing the energy mix, solar generation, renewable resources, hydroelectric generation, and more.

[California State Water Resources Control Board, 2022] California State Water Resources Control Board (2022). Groundwater issue: Supply. An overview of groundwater's role in California, its challenges, and the initiatives undertaken by the Water Boards to ensure its sustainable management.

[Campana et al., 2021] Campana, P. E., Stridh, B., Amaducci, S., and Colauzzi, M. (2021). Optimisation of vertically mounted agrivoltaic systems. *Journal of Cleaner Production*, 325:129091.

[Chang and Bonnette, 2016] Chang, H. and Bonnette, M. R. (2016). Climate change and water-related ecosystem services: impacts of drought in california, usa. *Ecosystem Health and Sustainability*, 2(12):e01254.

[Cornwall et al., 2021] Cornwall, C., Horiuchi, A., and Lehman, C. (2021). Noaa solar position calculator.

[Corrado et al., 2013] Corrado, C., Leow, S. W., Osborn, M., Chan, E., Balaban, B., and Carter, S. A. (2013). Optimization of gain and energy conversion efficiency using front-facing photovoltaic cell luminescent solar concentratordesign. *Solar Energy Materials and Solar Cells*, 111:74–81.

[Coşgun, 2021] Coşgun, A. E. (2021). The potential of agrivoltaic systems in turkey. *Energy Reports*, 7:105–111. 2021 6th International Conference on Advances on Clean Energy Research.

[Deutsches Institut für Normung, 2021] Deutsches Institut für Normung (2021). Agri-photovoltaic systems – requirements for primary agricultural use. Standard. English translation of DIN SPEC 91434:2021-05.

[Dupraz et al., 2011] Dupraz, C., Marrou, H., Talbot, G., Dufour, L., Nogier, A., and Ferard, Y. (2011). Combining solar photovoltaic panels and food crops for optimising land use: Towards new agrivoltaic schemes. *Renewable energy*, 36(10):2725–2732.

[EIA, 2023] EIA (2023). Carbon dioxide emissions coefficients. Accessed: 2024-06-11.

[Elamri et al., 2018] Elamri, Y., Cheviron, B., Lopez, J.-M., Dejean, C., and Belaud, G. (2018). Water budget and crop modelling for agrivoltaic systems: Application to irrigated lettuces. *Agricultural water management*, 208:440–453.

[Elborg, 2017] Elborg, M. (2017). Reducing land competition for agriculture and photovoltaic energy generation–a comparison of two agro-photovoltaic plants in japan. In *Proceedings of the International Conference on Sustainable and Renewable Energy Development and Design (SREDD2017)*, volume 3, page 5.

[Escriva-Bou et al., 2017] Escriva-Bou, A., Gray, B., Green, S., Harter, T., Howitt, R., MacEwan, D., and Seavy, N. (2017). Water stress and a changing san joaquin valley. *Public Policy Institute of California. https://www. ppic. org/content/pubs/report/R_0317EHR. pdf.*

[Evans, 2014] Evans, M. R. (2014). Common greenhouse glazing materials. A comprehensive overview of various greenhouse glazing materials, detailing their properties, advantages, and disadvantages.

[Farquhar et al., 1980] Farquhar, G. D., von Caemmerer, S., and Berry, J. A. (1980). A biochemical model of photosynthetic co2 assimilation in leaves of c3 species. *Planta*, 149(1):78–90.

[Galtieri and Krein, 2015] Galtieri, J. and Krein, P. T. (2015). Designing solar arrays to account for reduced performance from self-shading. In *2015 IEEE Power and Energy Conference at Illinois (PECI)*, pages 1–8. IEEE.

[Goetzberger and Zastrow, 1982] Goetzberger, A. and Zastrow, A. (1982). On the coexistence of solar-energy conversion and plant cultivation. *International Journal of Solar Energy*, 1(1):55–69.

[Group, 2024] Group, M. (2024). *LP-80 Manual*. Meter Group. Accessed: 2024-06-11.

[Hartz et al., 2008] Hartz, T., Miyao, G., Mickler, J., Lestrange, M., Stoddard, S., Nuñez, J., and Aegerter, B. (2008). Processing tomato production in california - ucanr.edu.

[Hassanpour Adeh et al., 2018] Hassanpour Adeh, E., Selker, J. S., and Higgins, C. W. (2018). Remarkable agrivoltaic influence on soil moisture, micrometeorology and water-use efficiency. *PloS one*, 13(11):e0203256.

[Havrysh et al., 2022] Havrysh, V., Kalinichenko, A., Szafranek, E., and Hruban, V. (2022). Agricultural land: Crop production or photovoltaic power plants. *Sustainability*, 14(9):5099.

[Hoogenboom et al., 2019] Hoogenboom, G., Porter, C. H., Boote, K. J., Shelia, V., Wilkens, P. W., Singh, U., White, J. W., Asseng, S., Lizaso, J. I., Moreno, L. P., et al. (2019). The dssat crop modeling ecosystem. In *Advances in crop modelling for a sustainable agriculture*, pages 173–216. Burleigh Dodds Science Publishing.

[Hudelson and Lieth, 2021] Hudelson, T. and Lieth, J. H. (2021). Crop production in partial shade of solar photovoltaic panels on trackers. In *AIP Conference Proceedings*, volume 2361. AIP Publishing.

[Imran et al., 2020] Imran, H., Riaz, M. H., and Butt, N. Z. (2020). Optimization of single-axis tracking of photovoltaic modules for agrivoltaic systems. In *2020 47th IEEE Photovoltaic Specialists Conference (PVSC)*, pages 1353–1356. IEEE.

[Jafarinejad et al., 2023] Jafarinejad, S., Hernandez, R. R., Bigham, S., and Beckingham, B. S. (2023). The intertwined renewable energy–water–environment (rewe) nexus challenges and opportunities: A case study of california. *Sustainability*, 15(13):10672.

[Jha et al., 2022] Jha, G., Nicolas, F., Schmidt, R., Suvočarev, K., Diaz, D., Kisekka, I., Scow, K., and Nocco, M. A. (2022). Irrigation decision support systems (idss) for california's water-nutrient-energy nexus. *Agronomy*, 12(8).

[Kadowaki et al., 2012] Kadowaki, M., Yano, A., Ishizu, F., Tanaka, T., and Noda, S. (2012). Effects of greenhouse photovoltaic array shading on welsh onion growth. *Biosystems Engineering*, 111(3):290–297.

[Kwon et al., 2020] Kwon, O., Kang, J., Trommsdorff, M., and Lee, K. (2020). Sensitivity analysis for optimized agrivoltaic designs: An inquiry on the trade-off between homogenous light conditions and electrical yield. In *2020 47th IEEE Photovoltaic Specialists Conference (PVSC)*, pages 1220–1225. IEEE.

[Letternmeier et al., 2021] Letternmeier, F., Gimbel, E., Schlaak, A., Schindele, S., Keinath, T., Steinhuser, A., Trommsdorff, M., Lindenmeyer, C., Mair, I. G., Lachhammer, A., Hilber, F., Balz, D.-I. F. M., Schlaich, P. D. M., Bohn, M., , Weinrebe, D. G., Eisel, M. D. F., Heintze, M. G., Huttmayer, T., Mack, J., Mayer, J., Bruckner, J., Lenck, J., Wagner, D.-I. F. J., Kameoka, F., Hogy, a. P. D. P., Pataczek, L., and Zikeli, D. S. (2021). Agri-photovoltaic systems – Requirements for primary agricultural use English translation of DIN SPEC 91434:2021-05. Standard, Deutsches Institut für Normung, Berlin, Germany.

[LI-COR, 2024] LI-COR, I. (2024). *LI-6800 Portable Photosynthesis System Instruction Manual*. LI-COR Biosciences. Accessed: 2024-06-11.

[Linker and Kisekka, 2023] Linker, R. and Kisekka, I. (2023). Model-based simulation-optimization of irrigation scheduling – a field evaluation with processing tomatoes. *Smart Agricultural Technology*, 4:100234.

[Loftus et al., 2015] Loftus, P. J., Cohen, A. M., Long, J. C., and Jenkins, J. D. (2015). A critical review of global decarbonization scenarios: what do they tell us about feasibility? *Wiley Interdisciplinary Reviews: Climate Change*, 6(1):93–112.

[Magarelli et al., 2024] Magarelli, A., Mazzeo, A., and Ferrara, G. (2024). Fruit crop species with agrivoltaic systems: A critical review. *Agronomy*, 14(4).

[Majumdar and Pasqualetti, 2018] Majumdar, D. and Pasqualetti, M. J. (2018). Dual use of agricultural land: Introducing 'agrivoltaics' in phoenix metropolitan statistical area, usa. *Landscape and Urban Planning*, 170:150–168.

[Marrou et al., 2013a] Marrou, H., Dufour, L., and Wery, J. (2013a). How does a shelter of solar panels influence water flows in a soil–crop system? *European Journal of Agronomy*, 50:38–51.

[Marrou et al., 2013b] Marrou, H., Guilioni, L., Dufour, L., Dupraz, C., and Wery, J. (2013b). Microclimate under agrivoltaic systems: Is crop growth rate affected in the partial shade of solar panels? *Agricultural and Forest Meteorology*, 177:117–132.

[Marrou et al., 2013c] Marrou, H., Wéry, J., Dufour, L., and Dupraz, C. (2013c). Productivity and radiation use efficiency of lettuces grown in the partial shade of photovoltaic panels. *European Journal of Agronomy*, 44:54–66.

[Massachusetts Department of Energy Resources, 2021] Massachusetts Department of Energy Resources (2021). Solar massachusetts renewable target (smart) program regulation 225 cmr 20.00.

[Mead and Willey, 1980] Mead, R. and Willey, R. (1980). The concept of a 'land equivalent ratio'and advantages in yields from intercropping. *Experimental agriculture*, 16(3):217–228.

[Melton et al., 2021] Melton, F., Huntington, J., Grimm, R., Herring, J., Hall, M., Rollison, D., Erickson, T., Allen, R., Anderson, M., Fisher, J., Kilic, A., Senay, G., Volk, J., Hain, C., Johnson, L., Ruhoff, A., Blankenau, P., Bromley, M., Carrara, W., and Anderson, R. (2021). Openet: Filling a critical data gap in water management for the western united states. *JAWRA Journal of the American Water Resources Association*, 58.

[Mengi et al., 2023] Mengi, E., Samara, O. A., and Zohdi, T. I. (2023). Crop-driven optimization of agrivoltaics using a digital-replica framework. *Smart Agricultural Technology*, 4:100168.

[Mistry, 2021] Mistry, N. (2021). Photo of agrivoltaic unit installed in the field.

[Mohammedi et al., 2023] Mohammedi, S., Dragonetti, G., Admane, N., and Fouial, A. (2023). The impact of agrivoltaic systems on tomato crop: A case study in southern italy. *Processes*, 11(12):3370.

[Monteith, 1965] Monteith, J. L. (1965). Evaporation and environment. In *Symposia of the society for experimental biology*, volume 19, pages 205–234. Cambridge University Press (CUP) Cambridge.

[NREL, 2012] NREL (2012). Life cycle greenhouse gas emissions from solar photovoltaics. Technical report, National Renewable Energy Laboratory, Golden, CO. NREL/FS-6A20-56487.

[NVIDIA Corporation, 2023] NVIDIA Corporation (2023). NVIDIA CUDA.

[Pathak et al., 2018] Pathak, T. B., Maskey, M. L., Dahlberg, J. A., Kearns, F., Bali, K. M., and Zaccaria, D. (2018). Climate change trends and impacts on california agriculture: a detailed review. *Agronomy*, 8(3):25.

[Patrignani and Ochsner, 2015] Patrignani, A. and Ochsner, T. E. (2015). Canopeo: A powerful new tool for measuring fractional green canopy cover. *Agronomy Journal*, 107(6):2312–2320.

[Pauliuk and Heeren, 2021] Pauliuk, S. and Heeren, N. (2021). Material efficiency and its contribution to climate change mitigation in germany: A deep decarbonization scenario analysis until 2060. *Journal of Industrial Ecology*, 25(2):479–493.

[Perna et al., 2019] Perna, A., Grubbs, E. K., Agrawal, R., and Bermel, P. (2019). Design considerations for agrophotovoltaic systems: maintaining pv area with increased crop yield. In *2019 IEEE 46th Photovoltaic Specialists Conference (PVSC)*, pages 0668–0672. IEEE.

[Ponce de León and Bailey, 2019] Ponce de León, M. A. and Bailey, B. N. (2019). Evaluating the use of beer's law for estimating light interception in canopy architectures with varying heterogeneity and anisotropy. *Ecological Modelling*, 406:133–143.

[Prentice et al., 1992] Prentice, I. C., Cramer, W., Harrison, S. P., Leemans, R., Monserud, R. A., and Solomon, A. M. (1992). Special paper: a global biome model based on plant physiology and dominance, soil properties and climate. *Journal of biogeography*, pages 117–134.

[Prugh et al., 2018] Prugh, L. R., Deguines, N., Grinath, J. B., Suding, K. N., Bean, W. T., Stafford, R., and Brashares, J. S. (2018). Ecological winners and losers of extreme drought in california. *Nature Climate Change*, 8(9):819–824.

[Raza et al., 2019] Raza, A., Razzaq, A., Mehmood, S. S., Zou, X., Zhang, X., Lv, Y., and Xu, J. (2019). Impact of climate change on crops adaptation and strategies to tackle its outcome: A review. *Plants*, 8(2):34.

[Riaz et al., 2022] Riaz, M. H., Imran, H., Alam, H., Alam, M. A., and Butt, N. Z. (2022). Crop-specific optimization of bifacial pv arrays for agrivoltaic food-energy production: The light-productivity-factor approach. *IEEE Journal of Photovoltaics*, 12(2):572–580.

[Sargentis et al., 2021] Sargentis, G.-F., Siamparina, P., Sakki, G.-K., Efstratiadis, A., Chiotinis, M., and Koutsoyiannis, D. (2021). Agricultural land or photovoltaic parks? the water–energy–food nexus and land development perspectives in the thessaly plain, greece. *Sustainability*, 13(16):8935.

[Scarano et al., 2024] Scarano, A., Semeraro, T., Calisi, A., Aretano, R., Rotolo, C., Lenucci, M. S., Santino, A., Piro, G., and De Caroli, M. (2024). Effects of the agrivoltaic system on crop production: The case of tomato (solanum lycopersicum l.). *Applied Sciences*, 14(7):3095.

[Senevirathna et al., 2003] Senevirathna, A., Stirling, C., and Rodrigo, V. (2003). Growth, photosynthetic performance and shade adaptation of rubber (hevea brasiliensis) grown in natural shade. *Tree Physiology*, 23(10):705–712.

[Senevirathna et al., 2008] Senevirathna, A., Stirling, C., and Rodrigo, V. (2008). Acclimation of photosynthesis and growth of banana (musa sp.) to natural shade in the humid tropics. *Experimental Agriculture*, 44(3):301–312.

[Sengupta et al., 2018] Sengupta, M., Xie, Y., Lopez, A., Habte, A., Maclaurin, G., and Shelby, J. (2018). The national solar radiation data base (nsrdb). *Renewable and Sustainable Energy Reviews*, 89:51–60.

[Siles et al., 2011] Siles, P., Bustamante, O., Valdivia, E., Burkhardt, J., and Staver, C. (2011). Photosynthetic performance of banana ('gros michel', aaa) under a natural shade gradient. In *VII International Symposium on Banana: ISHS-ProMusa Symposium on Bananas and Plantains: Towards Sustainable Global Production 986*, pages 71–77.

[Sroufe and Watts, 2022] Sroufe, R. and Watts, A. (2022). Pathways to agricultural decarbonization: Climate change obstacles and opportunities in the us. *Resources, Conservation and Recycling*, 182:106276.

[Steadman and Higgins, 2022] Steadman, C. L. and Higgins, C. W. (2022). Agrivoltaic systems have the potential to meet energy demands of electric vehicles in rural oregon, us. *Scientific Reports*, 12(1):4647.

[Touil et al., 2021] Touil, S., Richa, A., Fizir, M., and Bingwa, B. (2021). Shading effect of photovoltaic panels on horticulture crops production: A mini review. *Reviews in Environmental Science and Bio/Technology*, 20(2):281–296.

[Trommsdorff et al., 2021a] Trommsdorff, M., Kang, J., Reise, C., Schindele, S., Bopp, G., Ehmann, A., Weselek, A., Högy, P., and Obergfell, T. (2021a). Combining food and energy production: Design of an agrivoltaic system applied in arable and vegetable farming in germany. *Renewable and Sustainable Energy Reviews*, 140:110694.

[Trommsdorff et al., 2021b] Trommsdorff, M., Kang, J., Reise, C., Schindele, S., Bopp, G., Ehmann, A., Weselek, A., Högy, P., and Obergfell, T. (2021b). Combining food and energy production: Design of an agrivoltaic system applied in arable and vegetable farming in germany. *Renewable and Sustainable Energy Reviews*, 140:110694.

[United States Department of Agriculture, 2023] United States Department of Agriculture, N. A. S. S. (2023). *2023 California Processing Tomato Report*. USDA NASS. Accessed: 2024-06-11.

[Valle et al., 2017] Valle, B., Simonneau, T., Sourd, F., Pechier, P., Hamard, P., Frisson, T., Ryckewaert, M., and Christophe, A. (2017). Increasing the total productivity of a land by combining mobile photovoltaic panels and food crops. *Applied energy*, 206:1495–1507.

[von Caemmerer et al., 2009] von Caemmerer, S., Farquhar, G., and Berry, J. (2009). Biochemical model of c3 photosynthesis. In *Photosynthesis in silico: understanding complexity from molecules to ecosystems*, pages 209–230. Springer.

[Zhao et al., 2019a] Zhao, C., Liu, B., Xiao, L., Hoogenboom, G., Boote, K. J., Kassie, B. T., Pavan, W., Shelia, V., Kim, K. S., Hernandez-Ochoa, I. M., et al. (2019a). A simple crop model. *European Journal of Agronomy*, 104:97–106.

[Zhao et al., 2019b] Zhao, C., Liu, B., Xiao, L., Hoogenboom, G., Boote, K. J., Kassie, B. T., Pavan, W., Shelia, V., Kim, K. S., Hernandez-Ochoa, I. M., Wallach, D., Porter, C. H., Stockle, C. O., Zhu, Y., and Asseng, S. (2019b). A simple crop model. *European Journal of Agronomy*, 104:97–106.

[Zohdi, 2021] Zohdi, T. (2021). A digital-twin and machine-learning framework for the design of multiobjective agrophotovoltaic solar farms. *Computational Mechanics*, 68(2):357–370.