

Bachelor-Thesis Studiengang Digital Construction in Architecture

Digital unterstützte PV-Planung und Optimierung Fassade

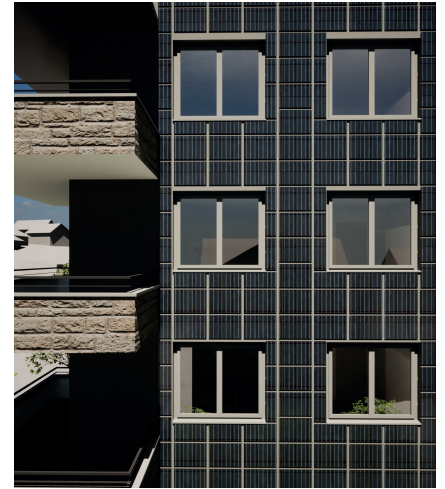
Visualisierung automatische Anordnung



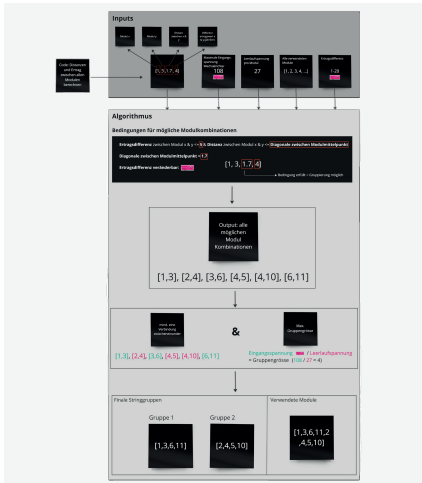
Frontansicht Photovoltaik-Fassade



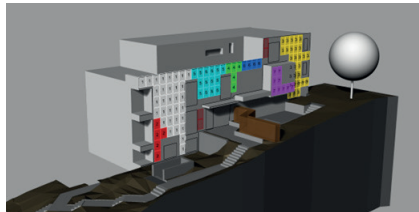
Optimiert auf minimalen Verschchnitt



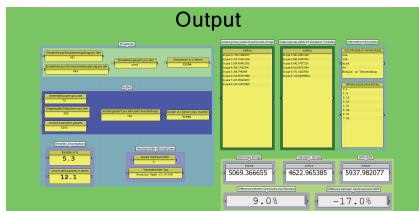
Logik des Optimierungsalgorithmus



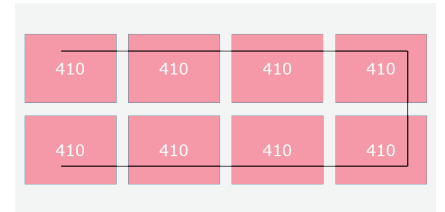
Berechnete String-Verschaltung & Ertrag in Rhino



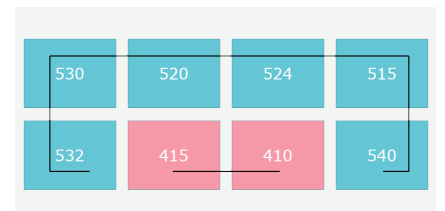
Output der berechneten Lösung



Ausgangsproblem Reihenschaltung



Optimierte Gruppierung mit Grasshopper-Skript



Problemstellung

Die Photovoltaik ist massgebend für die zukünftige nachhaltige Energieproduktion. Dafür müssen alle potenziellen Flächen wie die Gebäudefassaden ebenfalls genutzt werden. Da diese Flächen meist stark verschattet sind, mindert sich deren Ertrag stark im Gegensatz zu den Dachflächen. Weil die PV-Stränge immer in Reihe geschaltet sind, hat ein verschattetes Modul einen sehr hohen Einfluss auf den Ertrag des gesamten Strangs.

Um dieses Problem in den Griff zu kriegen werden heutzutage elektrische Leistungsoptimierer verwendet. Diese haben jedoch im Vergleich zu den PV-Modulen eine relativ niedrige Lebensdauer und verursachen hohe zusätzliche Kosten auf Grund der Installation, Wartung und dem Ersatz. Ausserdem müssen Fassadenlifte und Gerüste für den Unterhalt installiert werden.

Grundsätzlich läge das gesamte PV-Potenzial der Schweiz bei ca. 100 TWh, wobei Fassadenanlagen bis zu 18% des Gesamtertrages ausmachen könnten.

Lösungskonzept

Es wurden verschiedene Skripts erarbeitet, welche in jeder Planungsphase angewendet werden können. Dabei hat der Nutzer die Möglichkeit seine PV-Anlage selbst zu planen und anschliessend zu berechnen oder er lässt sich die Modulplatzierung automatisch vom Tool erstellen.

Der erarbeitete Algorithmus kann die bestmöglichen Gruppierungen von Modulen für die String-Verschaltung identifizieren. Dadurch kann der Gesamtertrag erhöht werden.

Zuerst wird die Einstrahlungssituation analysiert. Anschliessend werden die Modulerträge sowie die Positionen der Module abgeglichen und darauf folgende String-Gruppen generiert. Dabei wird darauf geachtet, dass die Installation nicht beeinträchtigt wird.

Damit die Resultate verglichen werden können, wird eine Wirtschaftlichkeitsrechnung der Option mit elektronischen Optimierern (MLPE), der Option ohne MLPE und einer «standard»-Planungsweise

erstellt. Der Output enthält neben den Ertragswerten auch Aussagen über die Rendite und die Amortisationsdauer.

Viele Parameter, wie: PV-Modultyp, Optimizer-Kosten, Wechselrichter-Typ, Panelausrichtung, Angaben zu Einspeisevergütung, Stromkosten, usw. können vom Nutzer selbst eingegeben werden. Dadurch wurde ein interaktives Tool geschaffen, welches dem Nutzer die Möglichkeit gibt, einfach und in verschiedenen Planungsphasen verschiedene PV-Situationen durchzurechnen und diese zu optimieren.

Samuel Henseler
Marco Kohler

Betreuer:
Markus Weber
Michal Rontsinsky

Kooperationspartner:
Amr Elesawy, Kostezer Daniel, Eric Langenskiöld
Basler & Hofmann AG

Hochschule Luzern
Technik & Architektur

Fachartikel zur Bachelor-Thesis

Im Studiengang

«Digital Construction in Architecture»

Digital unterstützte PV-Planung und Optimierung Fassade

Verfasser: Marco Kohler
Samuel Henseler

Dozierende: Markus Weber
Michal Rontsinsky

Auftraggeber: Hochschule Luzern – Technik & Architektur
(Industriepartner: Amr Elesawy, Basler & Hofmann AG)

Abgabedatum: 09. Juni 2023

Digital unterstützte PV-Planung und Optimierung Fassade

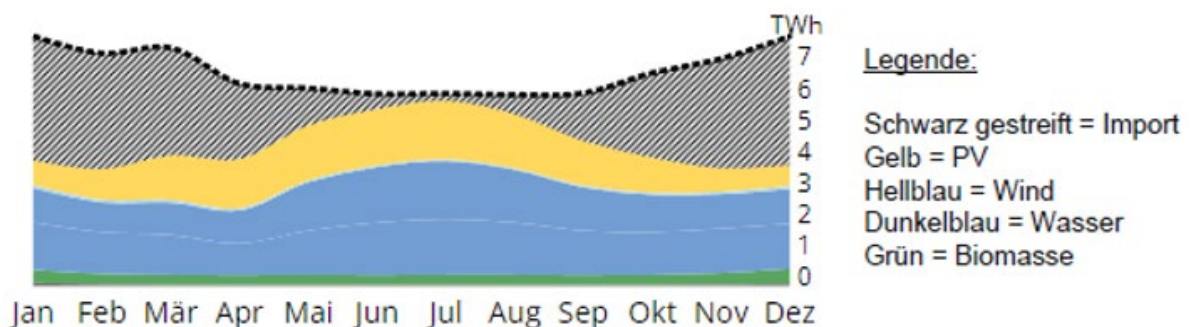
Um die Energiewende zu schaffen, muss in Zukunft auf Photovoltaikanlagen gesetzt werden. Aktuell sind auf dem Schweizer Markt vor allem Aufdachanlagen stark vertreten, da sie den grössten Ertrag versprechen. Eine Investition in Fassadenanlagen kann sich ebenfalls lohnen. Um Ertragseinbussen auf Grund der Verschattung zu minimieren, werden heutzutage elektronische Leistungsoptimierer eingesetzt. Eine intelligente Verschaltung der Strings kann ohne Leistungsoptimierer ähnliche Ertragsleistungen erzielen bei geringeren Investitions- und Unterhaltskosten.

Aufgabenstellung

Die Aufgabenstellung gliedert sich in mehrere Hauptpunkte. Zu Beginn wird eine theoretische Recherche über den Markt der Photovoltaik Anlagen und verschiedene Lösungsansätze zum Verschattungsproblem angestellt. Anschliessend muss der herkömmliche Berechnungsvorgang des Industriepartners Basler & Hofmann analysiert werden. Bei Punkt drei wird durch ein erstelltes Grasshopper Skript die Ertragsberechnung automatisiert. Dabei werden Teile der Planung automatisiert sowie die Wirtschaftlichkeitsberechnung ergänzt. Die String-Verschaltung wird auf Basis eines Algorithmus optimiert. Als Validierung wird abschliessend ein Vergleich zwischen dem Ertrag der herkömmlichen String-Verschaltung, der String-Verschaltung mit Optimierer und der erarbeiteten optimierten String-Verschaltung angestellt.

Energiesituation Schweiz und Potenzial PV-Anlagen

Die Grafik zeigt, wie die Energiesituation mit der heutigen Ausbaustrategie des Stromnetzes bis 2050 aussieht.



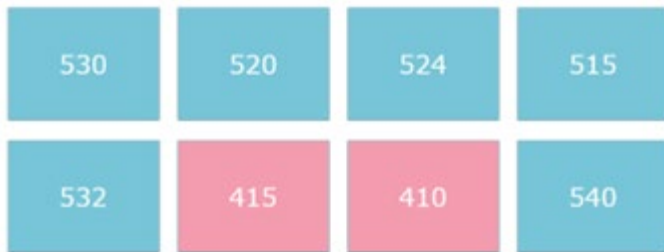
1 Energiesituation bis 2050

In der Schweiz wurden im Referenzjahr 2021 ca. 27'000 Netzverbundanlagen installiert. Davon waren nur rund 100 Anlagen Fassadenanlagen. Diese tragen schweizweit ca. 1 % zur Gesamtleistung aller Photovoltaikanlagen bei. Das Potenzial für PV-Dächer und PV-Fassaden

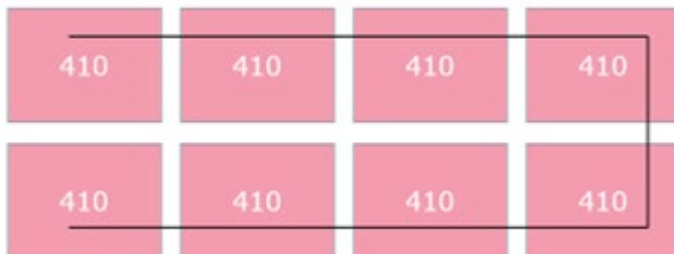
in der Schweiz liegt ungefähr bei 76 Terawattstunden pro Jahr, was 10 % mehr als unserem heutigen Stromverbrauch entspricht. Würden auch die Alpenflächen genutzt werden, so stiege dieses Potential auf 100 Terawattstunden pro Jahr. Das ausschöpfbare Potenzial von Fassadenanlagen entspricht ca. 18 % des gesamten PV-Potenzials der Schweiz, was somit etwa 18 Terawattstunden pro Jahr ausmachen würde. Die potenziellen Energiequellen werden jedoch noch nicht vollständig ausgenutzt. Die Stärke der PV-Anlage an der Fassade im Vergleich zur Anlage auf dem Dach ist die Stromproduktion an Vor- und Nachmittagen sowie im Winter. Dadurch würden diese Ertragslücken geschlossen werden.

String-Verschaltung

PV-Module müssen für eine möglichst hohe Spannung in Reihe, sogenannte Strings, geschaltet werden. Das Problem dieses Systems ist ähnlich wie bei einer Lichterkette: Fällt ein Element aus oder lässt weniger Spannung durch, entsteht zum einen ein lokaler Widerstand. Zum anderen führt die verminderte Spannung zu einem kleineren Ertrag. Diese Situation entsteht bei der Verschattung von einzelnen Modulen.



2 Einstrahlungssituation



Leistung der Fassade:
 $8 \text{ Module} \times \text{Modulnennleistung } 0,15 = 1,2 \text{ kWp}$
Ertrag:
 $1,2 \text{ kWp} \times 80\% \text{ PR} \times 410 \text{ kWh/m}^2\text{a} = 394 \text{ kWh/a}$

3 Berechnung String Verschaltung ohne Optimierer

Damit die verschatteten Module durch die vom Widerstand ausgelöste entstandene Hitze nicht schaden nehmen, werden elektronische Bypass-Dioden verwendet. Sie können einzelne verschattete Teile des Panels ausschalten und so den Strom ungehindert durch das Panel leiten. In aktuellen Modulen sind sie standardmässig verbaut. Elektronische Leistungsoptimierer können zusätzlich in der Anlage installiert werden. Sie sind dazu da, den jeweiligen Ertrag pro Modul zum Gesamtertrag des Strings beizusteuern. Der grösste Nachteil solcher Geräte ist die beschränkte Lebensdauer sowie die zusätzlichen Installations- und Unterhaltskosten. Bei kleineren oder stark verschatteten Anlagen kann ein Einsatz in Bezug auf die Wirtschaftlichkeit sogar kontraproduktiv sein. Mit Hilfe einer optimalen String-Verschaltung kann der Ertrag gesteigert, die Kosten jedoch im Vergleich zur Anlage mit Leistungsoptimierern tief gehalten werden.

Verschattung

Bei sonnigen, wolkenlosen Tagen und einer optimalen Ausrichtung zur Sonne liegt der Spitzenertrag in der Zentralschweiz bei ungefähr 1000 W/m². An trübigen Tagen oder bei vollständiger Verschattung wirkt ausschliesslich die diffuse Einstrahlung, was zu einer Ertragsminimierung von bis zu 80 % führen kann. Vor allem bei Fassadenanlagen ist die Verschattung ein grosses Thema, da beispielsweise durch Nachbargebäude und umliegende Bäume der Schattenanteil grösser ist als bei Aufdachanlagen. Aufgrund der Zusammensetzung des Gesamtertrags: kleinster Ertrag pro String mal die Anzahl Module im String vermindert sich der Ertrag stark auch bei Verschattung eines einzelnen Modules.

Optimierung String-Verschaltung

Um die optimale Verschaltung der Module auf einer spezifischen Anlage zu erreichen, wurde in Rhino / Grasshopper mit Python ein Algorithmus geschrieben, welcher diese Arbeit übernimmt. Der Algorithmus kann die bestmöglichen Gruppierungen von Modulen für die String-Verschaltung identifizieren. Dies wird erreicht, indem verschiedene Regeln definiert werden, welche festlegen ob die jeweiligen Module gruppiert werden können. Dabei wird darauf geachtet, dass der Gesamtertrag erhöht, jedoch die Installation nicht beeinträchtigt wird. Wichtige Faktoren sind dabei die Ertragsdifferenz zwischen den Modulen, die Distanz zwischen den Modulen und die Gruppengrösse.

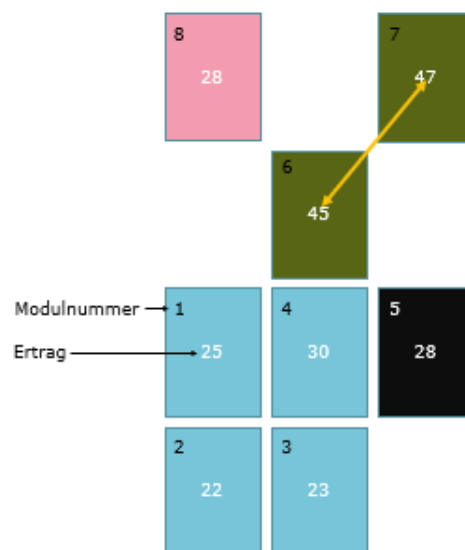
Um in derselben Gruppe eingeteilt zu werden, müssen folgende Bedingungen erfüllt werden:
Ertragsdifferenz ≤ 5 & Distanz \leq Diagonale zwischen Modulmittelpunkten / Gruppengrösse ≤ 4 (maximale Eingangsspannung) / Gruppengrösse ≤ 2 Module

String 1: Module 1, 2, 3, 4

Nicht verwendet: Modul 5 \rightarrow Blindmodul /

Modul 6, 7 \rightarrow Blindmodule / Modul 8 \rightarrow

Blindmodul

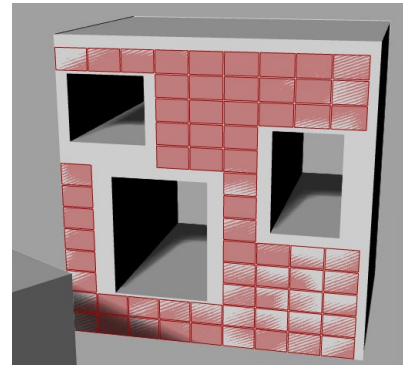


4 Logik Algorithmus

Die Logik des Algorithmus ist anpassbar auf verschiedene Optimierungsvorgänge. Die String-Verschaltung kann auf das gesamte Jahr oder einzelne Zeitabschnitte optimiert werden. Für die Optimierung der verschiedenen Zeitabschnitte wurde ein weiterer Algorithmus entwickelt. Dieser ist in der Lage, aufgrund der der spezifischen Zeiten die bestmögliche String-Verschaltung zu eruieren.

Automatische Modulgenerierung

Neben der manuellen, Erstellung von PV-Modulen und Berechnung des Ertrags gibt es im erarbeiteten Skript die Möglichkeit, die Module automatisch auf einer Fassade anordnen zu lassen und direkt deren Ertrag zu berechnen. Dieses Skript bietet in der Zukunft die Möglichkeit den optimalen Photovoltaik-Panel-Typ für spezifische Fassaden zu eruieren.



5 Automatische Modulplatzierung

Outcome

Bezüglich des Ertrags lohnt sich in den meisten Situationen der Einsatz von elektronischen Optimierern. Im getesteten Modell lag der Unterschied des Ertrags zwischen dem durch den Algorithmus optimierten System und dem System mit elektronischen Optimierern bei -12 %, was ca. 1'080 kWh/a entspricht. Im Vergleich zur nicht-optimierten Situation liegt die Verbesserung jedoch bei +13 %, was ca. 1'100 kWh/a entspricht. Der Vorteil durch das mit dem Algorithmus optimierten System sind die geringeren Investitionskosten, welche im Vergleich 17 % kleiner ausgefallen sind. Ebenso die Unterhaltskosten, welche ohne elektronische Optimierer um 17 % geringer sind. Einen grossen Unterschied ist bei den zyklischen Investitionskosten sichtbar, welche bei der Anlage mit elektronischen Optimierern nach 30 Jahren um 61 % höher ausfallen als bei der Anlage ohne elektronische Optimierer. Das liegt daran, weil die Lebensdauer für die elektronischen Optimierer bei ca. 15 Jahren liegen und diese deshalb die zyklischen Investitionskosten stark erhöhen.

Investitionskosten:

Ohne MLPE: 36'000.-
 Mit MLPE: 42'120.- **+17%**

Kosten/Jahr:

Ohne MLPE: 540.-
 Mit MLPE: 632.- **+17%**

Kosten/30Jahre*:

Ohne MLPE: 34'200.-
 Mit MLPE: 55'320.- **+61%**

Ertrag:

Ohne MLPE: 8'827 kWh/a
 Mit MLPE: 9'909 kWh/a **+12%**
 Standard: 7'704 kWh/a **-13%**

Einnahmen ohne MLPE:

Einspeisevergütung: 680.-
 Stromkosteneinsp.: 1'192.-
 Gesamt/Jahr: **1'872.-**
 Gesamt/30J: **56'160.-**

Einnahmen mit MLPE:

Einspeisevergütung: 764.-
 Stromkosteneinsp.: 1'338.-
 Gesamt/Jahr: **2'103.-**
 Gesamt/30J: **63'090.-**



Zusammenfassung ohne MLPE:

Einnahmen /30J: 56'160.-
 Kosten /30J: -34'200.-
 Gesamteinnahmen/30J: 21'960.-

Zusammenfassung mit MLPE:

Einnahmen /30J: 63'090.-
 Kosten /30J: -55'320.-
 Gesamteinnahmen/30J: 7'770.- **-35%**

*Die angenommene Lebensdauer der Wechselrichter und Optimizer ist 15 Jahre. Deshalb wird alle 15 Jahre neu investiert.

6 Vergleich optimierte String-Verschaltung & Stringverschaltung mit elektrischen Optimierern (MLPE)

University of Lucerne
Technology & Architecture

Abstract for the bachelor thesis

In the course

«Digital Construction in Architecture»

Digitally supported PV planning and optimization of the facade

Author: Marco Kohler
Samuel Henseler

Lecturers: Markus Weber
Michal Rontsinsky

Client: University of Lucerne – Technology & Architecture
(Industry partner: Amr Elesawy, Basler & Hofmann AG)

Submission date: June 09, 2023

Digitally supported PV planning and optimization of the facade

This work deals with the automated design of photovoltaic systems on facades in order to maximize their yield without electronic optimizers. Currently, shaded photovoltaic modules are equipped with electronic power optimizers, which, however, have a limited lifetime and cause high costs. Without power optimizers, a large potential of PV systems remains unused. The goal of this work is to use a Grasshopper script to automate and optimize string interconnection and module placement so that yield can be maximized. The optimization is done based on an economic calculation implemented in the script. The central research question includes increasing the yield by optimizing the strings and fully automating the PV planning process. The work includes theoretical research about the photovoltaic market in Switzerland, about technical basics about mounting and about the shading problem. The conventional planning process is analyzed. A Rhino / Grasshopper script automatically places modules on the facade and calculates the optimal string connection based on an irradiation analysis. The data is then evaluated in an economic calculation. The result is compared with a non-optimized system to validate the yield increase. This work results in four different scripts, each of which can be used for one planning phase. The results finally show the potential of automation in the field of PV planning.

Hochschule Luzern
Technik & Architektur

Bachelorarbeit

Im Studiengang

«Digital Construction in Architecture»

Digital unterstützte PV-Planung und Optimierung Fassade

Verfasser:

Marco Kohler

Matrikelnummer: 20-274-486

marco.kohler@stud.hslu.ch

Samuel Henseler

Matrikelnummer: 20-273-025

samuel.henseler@stud.hslu.ch

Bachelor-Thesis an der Hochschule Luzern - Technik & Architektur

Titel **Digital unterstützte PV-Planung und Optimierung Fassade**

Diplomandin/Diplomand Kohler, Marco

Bachelor-Studiengang **Bachelor Digital Construction**

Diplomandin/Diplomand Henseler, Samuel

Bachelor-Studiengang **Bachelor Digital Construction**

Semester **HS23**

Dozentin/Dozent **Weber, Markus & Rontsinsky, Michal**

Expertin/Experte **Demkow-Bartlomé, Joanna**

Abstract Deutsch

Diese Arbeit beschäftigt sich mit der automatisierten Planung von Photovoltaikanlagen an Fassaden, um deren Ertrag ohne elektronische Optimizer zu maximieren. Aktuell werden verschattete Photovoltaikmodule mit elektronischen Leistungsoptimierern ausgestattet, welche jedoch eine begrenzte Lebensdauer haben und hohe Kosten verursachen. Ohne Leistungsoptimierer bleibt ein grosses Potential der PV-Anlagen ungenutzt. Ziel der Arbeit ist es, mit einem Grasshopper-Skript die String Verschaltung sowie die Modulplatzierung zu automatisieren und optimieren, sodass der Ertrag maximiert werden kann. Die Optimierung wird auf Grundlage einer im Skript implementierten Wirtschaftlichkeitsrechnung angestellt. Die zentrale Forschungsfrage umfasst die Ertragssteigerung durch Optimierung der Strings und eine vollständige Automatisierung des PV-Planungsprozesses. Die Arbeit umfasst eine theoretische Recherche über den Photovoltaikmarkt der Schweiz, über technische Grundlagen zu Montage und über das Verschattungsproblem. Der herkömmliche Planungsprozess wird analysiert. Ein Rhino / Grasshopper-Skript platziert Module automatisiert auf der Fassade und berechnet auf Grundlage einer Einstrahlungsanalyse die optimale Stringverschaltung. Die Daten werden anschliessend in einer Wirtschaftlichkeitsrechnung ausgewertet. Das Ergebnis wird mit einer nicht-optimierten Anlage verglichen, um die Ertragssteigerung zu validieren. Aus dieser Arbeit entstehen drei verschiedene Skripts, welche für jeweils eine Planungsphase genutzt werden können. Die Ergebnisse zeigen schlussendlich das Potential einer Automatisierung im Bereich der PV-Planung auf.

Ort, Datum

Luzern, 09.06.2023

© Kohler, Marco & Henseler, Samuel, Hochschule Luzern – Technik & Architektur

Alle Rechte vorbehalten. Die Arbeit oder Teile davon dürfen ohne schriftliche Genehmigung der Rechteinhaber weder in irgendeiner Form reproduziert noch elektronisch gespeichert, verarbeitet, vervielfältigt oder verbreitet werden.

Sofern die Arbeit auf der Website der Hochschule Luzern online veröffentlicht wird, können abweichende Nutzungsbedingungen unter Creative-Commons-Lizenzen gelten. Massgebend ist in diesem Fall die auf der Website angezeigte Creative-Commons-Lizenz.

Inhalt

1	Einleitung	1
1.1	Thema.....	1
1.2	Erkenntnisinteresse	1
1.3	Ausgangslage	1
1.4	Fragestellungen	2
1.5	Zielsetzung.....	2
1.6	Hypothesen.....	2
1.7	Methodenwahl.....	3
2	Rolle der PV-Fassaden	4
2.1	Solarstrom für die Zukunft Luzern	4
2.2	Potenzial Fassadenanlagen.....	5
3	Technologische Grundlagen	8
3.1	Energiequelle Sonne	8
3.2	Panels	8
3.2.1	Paneltypen.....	8
3.2.2	Ausrichtung	9
3.2.3	Verschattung	10
3.3	Wechselrichter	11
3.4	Konstruktionen.....	12
3.5	MLPE & Bypass Dioden.....	14
4	Wartung	15
5	PV-Planungsvorgehen Industriepartner	16
6	Hindernisse	19
6.1	Verschattung / String-Verschaltung.....	19
6.2	Problemstellungen seitens Industriepartner	21
7	Das Projekt	22
7.1	Optimierung der Planung.....	22
7.2	Grundlagen.....	24
7.3	Vorgehen.....	27
7.4	Level 1 Ertragsberechnung Fläche.....	30
7.4.1	Benutzerinteraktion	30
7.4.2	Plugins und Components.....	32
7.4.3	Funktionsweise.....	32
7.4.4	Output.....	33
7.5	Level 2 Berechnung vorgegebener Strings	34
7.5.1	Benutzerinteraktion	34
7.5.2	Plugins und Components.....	35
7.5.3	Funktionsweise.....	36

7.5.4	Output.....	36
7.6	Level 3 Optimierung String-Verschaltung.....	38
7.6.1	Benutzerinteraktion	38
7.6.2	Plugins und Components.....	39
7.6.3	Algorithmus Optimierung String-Verschaltung.....	40
7.6.4	Stündliche Ertragsberechnung	45
7.6.5	Wirtschaftlichkeitsberechnung.....	46
7.6.6	Output.....	48
7.7	Level 4 Automatische PV Modul Generierung und Optimierung String Verschalung	50
7.7.1	Benutzerinteraktion	50
7.7.2	Plugins und Components.....	50
7.7.3	Automatische Modulgenerierung.....	51
7.7.4	Output.....	55
7.8	Forschung und mögliches weiteres Vorgehen	57
7.8.1	String-Verschaltung Optimierung mit Azimut Werten.....	57
7.8.2	Automatische Modulplatzierung.....	60
7.8.3	Bypass Implementierung.....	62
7.9	Rechenleistung	64
8.	Erkenntnisse.....	65
8.1	Kosten/ Erträge/ Differenzen	66
8.2	Jährlich-/ Monatliche-/ Stündliche Berechnung	69
8.3	String-Verschaltung Optimierung mit Azimut Werten.....	70
8.4	Bypass Erträge für eine Stunde	72
9.	Diskussion und Interpretation	74
10.	Reflexion.....	76
	Tabellenverzeichnis	78
	Abbildungsverzeichnis	79
	Literaturverzeichnis	81

1 Einleitung

1.1 Thema

Diese Arbeit untersucht die Möglichkeit, mittels automatisiertem Planungsvorgang den optimalen Ertrag einer Photovoltaikanlage an einer Fassade zu erhalten, ohne dafür elektronische Optimizer oder Module-level power electronics (MLPE) zu nutzen.

1.2 Erkenntnisinteresse

Basler & Hofmann stellt ihr Problem der zeitaufwändigen, manuellen String-Verschaltung Planung vor und zeigt, dass ein grosses Potenzial zur Beschleunigung dieses Prozesses über die Automatisierung erreichbar wäre. Ebenfalls geht durch die manuelle Planung ein grosser Teil an Ertrag verloren. Die Bachelorarbeit zur «Digital unterstützten PV-Planung und Optimierung Fassade» greift dieses Problem auf und es wird ein Grasshopper-Skript entwickelt, welches grosse Teile der String-Verschaltung Planung und der dazugehörigen Sonnenanalysen, Modulplatzierung auf der Fassade, Optimierungen der String-Verschaltung sowie der Ertragsberechnung automatisiert.

1.3 Ausgangslage

Einzelne verschattete Photovoltaikmodule beeinflussen den Energieertrag der gesamten Photovoltaikanlage und mindern so deren Effizienz und Ertrag. Um dieses Problem zu lösen, werden heutzutage MLPE verwendet. Diese haben jedoch eine kurze Lebensdauer und vergleichsweise hohe Unterhaltskosten. Werden die PV-Module untereinander optimal verschaltet, können die MLPE weggelassen werden. Dieser Planungsschritt wird heutzutage manuell erledigt, wobei keine direkte Optimierung der Verschaltung angestrebt wird. Dies soll in Zukunft automatisiert werden, damit die Effizienz der Anlagen hoch und der Planungsaufwand sowie die Unterhaltskosten tief gehalten werden können.

1.4 Fragestellungen

Es stellt sich die zentrale Forschungsfrage, wie der Ertrag und der Planungsprozess einer Photovoltaikanlage, ohne MLPE verbessert werden kann. Es werden folgende Teilfragen gestellt, um dieser Forschungsfrage nachzugehen.

F.1: Kann durch die Optimierung der String-Verschaltung eine Erhöhung des Ertrags erzielt werden?

F.2: Kann der gesamte Photovoltaikplanungsteil von Simulation über Analyse und Planung bis hin zur Ertragsrechnung automatisiert werden?

1.5 Zielsetzung

Das Ziel dieser vorliegenden Arbeit ist es aufzuzeigen, wie durch die Automatisierung eine optimale String-Verschaltung der PV-Module umgesetzt werden kann. Ausserdem soll der Planungsprozess automatisiert werden. Als Output wird eine Ertragsrechnung der optimierten String-Verschaltung erstellt. Neben der Ertragsrechnung soll das Skript ebenfalls als Output angesehen werden. Die Wirtschaftlichkeitsrechnung stellt den Vergleich der herkömmlichen und der Variante mit MLPE zur optimierten Variante der String-Verschaltung dar.

1.6 Hypothesen

Die forschungsleitenden Vermutungen lauten:

H1: «Durch die Optimierung der String-Verschaltung kann der Ertrag einer Photovoltaikanlage an einer Fassade um 10 % erhöht werden.»

H2: «Der Arbeitsablauf der Photovoltaikplanung an Fassaden kann ebenfalls vollständig automatisiert werden.»

1.7 Methodenwahl

Zu Beginn wird eine theoretische Recherche über den Markt der Photovoltaik-Anlagen angestellt. Ausserdem wird die Wichtigkeit der Photovoltaik für die Zukunft der Energiegewinnung aufgezeigt. Zudem wird der herkömmliche Prozess der Ertragsberechnung einer Photovoltaik Anlage analysiert. Anschliessend wird ein Grasshopper-Skript erstellt, welches die grundsätzlichen Ertragsberechnungen automatisiert. Dieses Skript wird erweitert mit der automatischen Modulverteilung auf einer gegebenen Fassade. Zum Schluss wird mittels eines Algorithmus die optimale String-Verschaltung der Module untereinander ermittelt. Aus dem Resultat wird ein Vergleich zu einer nicht-optimierten Anlage sowie zu einer Anlage mit MLPE gezogen, damit die Ertragssteigerung validiert werden kann. Die Ertrags- sowie Wirtschaftlichkeitsberechnung, welche in den Planungsprozess von Basler & Hofmann implementiert werden kann, wird abschliessend im Skript ergänzt.

2 Rolle der PV-Fassaden

Dieses Kapitel befasst sich zuerst mit dem Solarstrom als Energiequelle der Zukunft am Beispiel Luzern. Danach wird das ausschöpfbare Potential von Dach- sowie Fassadenanlagen aufgezeigt. Dies dient dazu, die Bedeutung der Forschung im Bereich der PV-Fassaden aufzuzeigen.

2.1 Solarstrom für die Zukunft Luzern

Gemäss dem Stadtrat Luzern (Stadtrat Stadt Luzern, 2021, S. 4) und ihrer ausgearbeiteten Energiestrategie bis zum Jahre 2030 soll der Primärenergieverbrauch bis 2050 auf 2'000 Watt Dauerleistung pro Kopf der Bevölkerung gesenkt werden. Dabei sollen die Treibhausgasemissionen bis 2040 auf 0 Tonnen reduziert werden (Stadtrat Stadt Luzern, 2021, S. 4). Dazu muss stark in den Ausbau des Solarstromnetzes investiert werden. Ziel ist es laut Stadtrat Luzern (ebd.) «bis 2050 ein Viertel des voraussichtlich zunehmenden Stromverbrauchs abzudecken». Neben dem Ausbau der Solarstromanlagen auf städtischen Liegenschaften, sollen auch private Liegenschaften von einem Ausbau profitieren. Es werden gemäss Stadtrat (ebd., S.105) «Förderprogramme ab dem Jahr 2023 umgesetzt, welche Boni für verschiedene Eigenschaften vergeben wie: hohe Winterstromproduktion, Kombination Dachbegrünung und Photovoltaik, hoher Dachflächennutzungsanteil, ...». Der Vorgehensplan zeigt eindeutig die Wichtigkeit und Unumgänglichkeit dieser nachhaltigen Energiebeschaffung in Bezug auf das Klimaziel 2050 auf. Daraus lässt sich auch schliessen, dass das Potenzial der Solarstromanlagen noch nicht ausgenutzt wird, einerseits auf der Seite der Politik, wie aber auch auf Seite der technischen Umsetzung. Fassadenanlagen werden in dem vorliegenden Dokument nur spärlich aufgegriffen, doch ist deren Potenzial ebenfalls ein wichtiges Asset, um das Klimaziel 2050 Netto-Null möglichst schnell zu erreichen (ebd., S.44) (siehe Kapitel 2.2).

Ein kompletter Umsturz der Energiegewinnung, sodass die Primärenergie bis 2050 nicht mehr gebraucht wird und der gesamte Energiebedarf durch erneuerbare Sekundärenergie gedeckt werden kann, ist gemäss Stadtrat Luzern im Moment nicht möglich. Dies aufgrund der ebenfalls limitierten Verfügbarkeit von erneuerbaren Energieressourcen

bedingt durch «Knappheit an Landressourcen, Ansprüche an Natur-, Landschafts-, Ortsbild- oder Denkmalschutz ...» (ebd., S.50).

Trotzdem ist es möglich, einen hohen Anteil an heutiger Primärenergie mittels technologischer Lösungen und politischen Anreizen in CO₂-arme Sekundärenergie umzuwandeln.

2.2 Potenzial Fassadenanlagen

Grundsätzlich kann zwischen vier verschiedenen Arten von Photovoltaikanlagen unterschieden werden: Inselanlage, Bodenanlagen, Aufdachanlagen und gebäudeintegrierte Anlagen (Roberts & Guariento, 2009, S. 10). Aufdachanlagen und gebäudeintegrierte Anlagen, unter welche auch Fassadenanlagen dazugehören, sind meist netzgekoppelt. Das bedeutet es existiert kein Batteriespeicher und der überschüssige, nicht im Eigenverbrauch genutzte Strom, wird zu einem vom Netzanbieter festgelegten Netzpreis verkauft. Ausserdem befindet sich die stromerzeugende Anlage laut Roberts (ebd.) «bei den Gebäuden, die den Strom nutzen, im Gegensatz zu den grossen Kraftwerken, deren zentral erzeugter Strom über lange Hochspannungsleitungen zu den Nutzern gelangt».

Gebäudeintegrierte Anlagen, welche mit Metallkonstruktionen am Gebäude fix befestigt werden, sind Stand 2009 nicht sehr verbreitet. So trugen sie gerade mal 1 Prozent (ebd., S.11) zur gesamten Erzeugungsleistung in Deutschland bei. Zum Vergleich wurden im Referenzjahr 2021 in der Schweiz ca. 27'000 Netzverbundanlagen installiert, davon waren nur rund 100 Anlagen Fassadenanlagen (Swissolar, 2022, S. 8). Sie trugen also ebenso wie in Deutschland nur ca. 1 % zur Gesamtleistung durch Photovoltaikanlagen bei.

Insgesamt liegt das realisierbare Potenzial für PV-Dächer und Fassaden in der Schweiz gemäss Swissolar (ebd., S.17) bei «67 Terawattstunden pro Jahr», was «10 Prozent mehr als unserem heutigen Stromverbrauch» entspricht. Insgesamt würde, inklusive Nutzung der Alpenflächen, ein Gesamtpotential der PV-Anlagen in der Schweiz von ca. 100TWh/Jahr vorliegen. Laut Giddey¹ liegt das auf Fassadensysteme bezogene ausschöpfbare Potenzial bei ca. 18 % des gesamten PV-Potentials der Schweiz, was ungefähr 18 TWh/Jahr bedeuten würde. EFH und MFH machen Stand 2021 fast die Hälfte

¹ Gemäss Vortrag von Giddey, F.an der ETH vom 28.02.2023

aller installierten Anlagen aus (ebd., S.14). Die zweite Hälfte der Anlagen befindet sich auf Industrie- und Gewerbebauten. Für eine Erreichung des Ziels von 100TWh/Jahr ist die vollständige Erschließung aller restlichen Gebäude essenziell.

Auch diese müssten noch ausgebaut werden. Ziel sollte eine kombinierte PV-Leistung von 2000MW/a sein (ebd.).

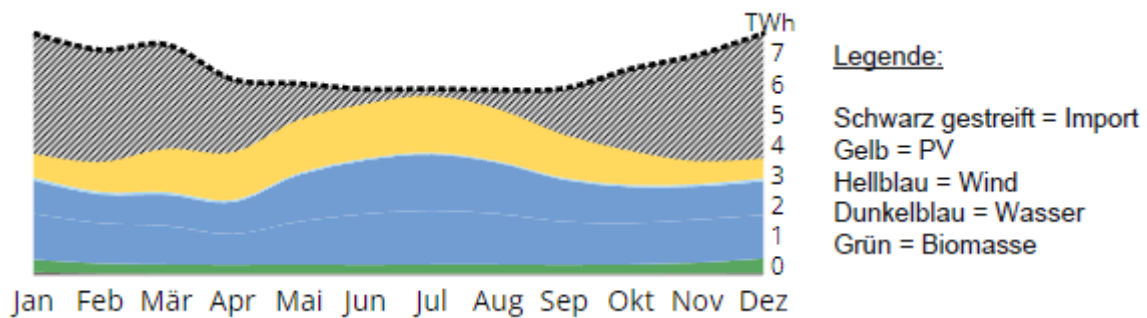


Abb. 1: "Powerswitcher" Axpo Szenario aktuelles Ausbautempo bis 2050 (Axpo, 2023)

Auf der Abbildung 1 wird deutlich, wie die Energiesituation mit der heutigen Ausbaustrategie des Stromnetzes bis 2050 aussieht. PV-Anlagen leisten im Winterhalbjahr relativ wenig, was bedeutet, dass viel Strom importiert werden muss (schwarz gestreift). Daraus lässt sich ableiten, dass das Netz der PV-Anlagen zum einen ausgebaut werden muss, da auch im Sommerhalbjahr noch Strom importiert wird, und zum anderen die Optimierung mehr auf den Winter ausgerichtet werden muss. Das würde bedeuten, dass mehr Ost- und Westfassaden jedoch weniger Südfassaden mit hohem Peak Wert für die Installation von PV-Anlagen genutzt werden müssen. Ausserdem wirkt sich die Optimierung für die Wintermonate auf die Ausrichtung der Module nach Azimut Wert aus.

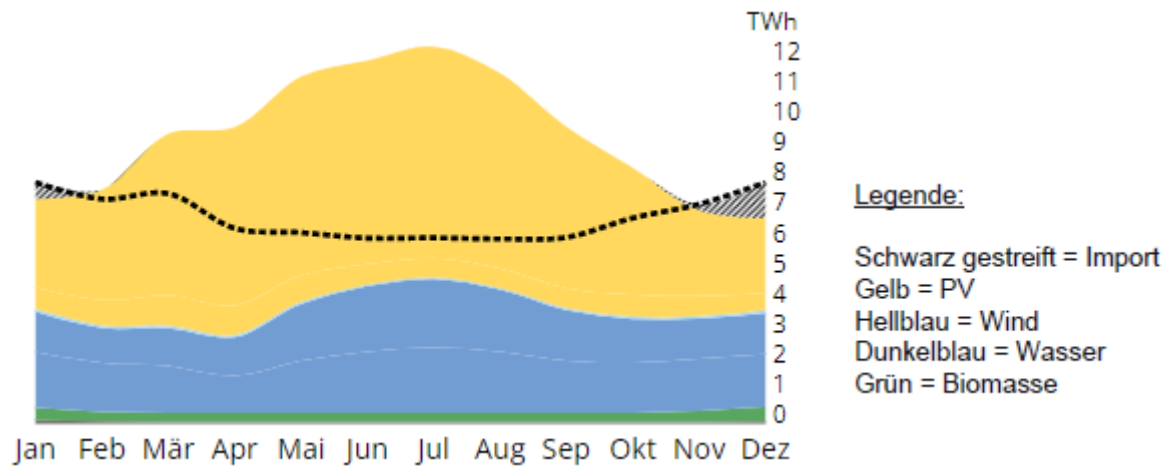


Abb. 2: Szenario 2050 Ausbau PV x5.5 (Axpo, 2023)

Die Abbildung 2 zeigt das Potenzial der PV-Anlagen, wenn diese bis 2050 um das 5.5-Fache ausgebaut würden. Es ist ersichtlich, dass der eingekaufte, importierte Strom fast vollständig vermieden werden kann. Der massive Überfluss an Energie in den Sommermonaten könnte mithilfe von Stauseen teilweise gespeichert und in den Monaten November, Dezember und Januar wieder verwendet werden. So liesse sich auch diese Energie wiederverwenden. Wie in (Anhang A1) ersichtlich, ist die Nachfrage nach PV-Fassaden gemäss Eric Langenskiöld (25.04.23) in Zukunft steigend, da ihr Potenzial in der Stromproduktion am Vor- und Nachmittag ist. Dies dient als ausgleichendes System im Vergleich zu den Dachanlagen, deren Stärke vor allem die Mittagssonne ist. Ausserdem wird laut Langenskiöld (ebd.) ein Marktpotential der PV-Fassaden auf ungefähr 30 % der Neubauten geschätzt.

3 Technologische Grundlagen

In diesem Kapitel wird zuerst auf die elektrische Energiequelle «Sonne» eingegangen. Anschliessend befasst sich die Arbeit mit den gängigsten Paneltypen, wobei auch die herkömmlichen Befestigungsmethoden erklärt werden. Dann wird auf die Verschattung als Hauptproblem der Fassadenanlagen eingegangen. Zum Schluss werden die Wechselrichter als wichtiges Element der PV-Anlage hervorgehoben und die Möglichkeiten elektronischer Optimierer wie Bypass-Dioden und MLPE aufgezeigt.

3.1 Energiequelle Sonne

Die Sonne als grösster Energielieferant der Erde sendet laut Weller (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 11) Energie in Form von elektromagnetischer Strahlung auf die Erde. Diese Strahlung kann in elektrischen Strom umgewandelt werden. Dafür muss zuerst die Strahlungsleistung einer optimalen, senkrechten Einstrahlung auf eine Fläche von einem Quadratmeter gemessen werden. «Die sogenannte Solarkonstante beträgt etwa $1367\text{W}/\text{m}^2$ » (ibd.). Jedoch wird diese Solarkonstante reduziert durch Luftmoleküle und Aerosole und beläuft sich deshalb auf maximal $1000\text{ W}/\text{m}^2$ (ibd.).

3.2 Panels

3.2.1 Paneltypen

Bei den kommerziell angebotenen Solarzellen kann zwischen zwei Hauptgruppen unterschieden werden: kristalline Zellen aus Silizium und neuere Dünnschichtzellen aus Halbleitermaterialien (ibd., S.14). Die kristallinen Zellen können weiter in monokristallin und polykristallin unterteilt werden, wobei monokristalline Zellen einen 20-30 % höheren Wirkungsgrad besitzen, jedoch auch kostenintensiver sind (ibd.). Deshalb bestehen laut Weller (ibd., S.15) «fast 90 % der weltweit eingesetzten Solarzellen aus kristallinem Silizium.»

3.2.2 Ausrichtung

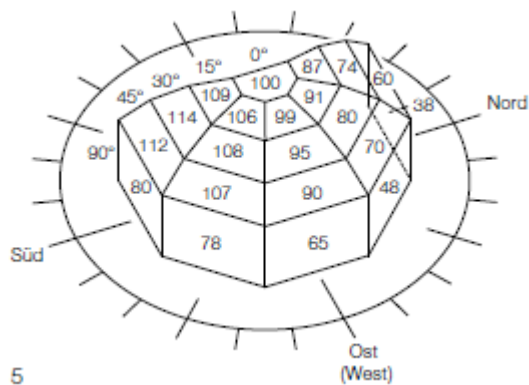


Abb. 3: jährliche Einstrahlung auf unterschiedliche Dachflächenorientierungen im Vergleich zur Horizontalen (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 12)

Die Sonnenstrahlung setzt sich grundsätzlich aus der direkten, der diffusen und der reflektierenden Strahlung zusammen. «Solarzellen können alle drei Formen nutzen, wobei die direkte Strahlung am meisten Energie enthält.» (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 12). Da sich die maximale Leistung der Solar konstanten aus einer Fläche senkrecht zur Einstrahlung bezieht, kann daraus abgeleitet werden, dass eine Dachfläche mehr Energie produziert, je mehr sie der Sonne zugewandt ist. Die obenstehende Grafik Abbildung 3 zeigt auf, wie sich der Ertrag in Prozent im Vergleich zur Horizontalen in unseren Breitengraden verhält. Es ist ersichtlich, dass eine 30 Grad angewinkelte und südlich ausgerichtete Fläche ungefähr 15 % mehr Ertrag einbringt als eine horizontale Ausrichtung. Fassadenflächen im Süden mit 90 Grad Ausrichtung bringen ca. 20 % weniger Ertrag. Trotz nicht optimaler Ausrichtung der Modulflächen auf West- und Ostseite können selbst Fassadenflächen noch akzeptable Energieerträge von 55 % - 65 % im Vergleich zu einer optimal ausgerichteten Dachfläche leisten (ebd., S.13). Stark geneigte Flächen in Nordrichtung sind für Photovoltaikmodule nicht optimal, so werden sie trotzdem oft aufgrund von architektonischer Konsistenz gebaut. Der höchste Ertrag kann mithilfe von «dem Sonnenstand selbst ausrichtenden Panels» erreicht werden. Dies lohnt sich aufgrund des Mehraufwandes «jedoch hauptsächlich in Südeuropa, wo die Direktstrahlung hoch ist ...» (ebd.).

3.2.3 Verschattung

Wie vorgängig erwähnt, kann der Spitzenwert an sonnigen, wolkenlosen Tagen bei optimaler Ausrichtung bei $1000\text{W}/\text{m}^2$ liegen. An trüben Tagen wirkt ausschliesslich die diffuse Strahlung, was bei der gleichen Ausrichtung meist einen nicht höheren Spitzenwert als $50\text{W}/\text{m}^2$ zur Folge hat (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 12). Durch die Verschattung von PV-Anlagen werden vor allem kristalline Module stark beeinträchtigt. Das verschattende Objekt verhindert die direkte Sonneneinstrahlung, sodass nur noch die diffuse Strahlung auf die entsprechende Fläche trifft. Energieerträge minimieren sich so um bis zu 80 % (ebd., S.13).

Um das Problem der Verschattung schon frühzeitig aufzugreifen, müssen Planer die Schattenwanderung auf dem betreffenden Gebäude analysieren. Dabei können technische Analysen, wie sie in Rhino/Grasshopper oder spezifischen Sonnenanalysetools möglich sind, helfen. Jeder Zeitpunkt besitzt einen spezifischen Sonnenstand, welcher in einem Sonnenbahndiagramm abgebildet werden kann. Dieser wird definiert über den Azimut Winkel (Himmelsrichtung) sowie dem Höhenwinkel zwischen dem Sonnenstand und dem Horizont (ebd.). Grundsätzlich gilt: Je mehr Zeitpunkte in die Analyse aufgenommen werden, desto genauer ist die Sonnenanalyse. Dies ist wichtig, um eine möglichst gute Annäherung an die effektiven Einstrahlungswerte zu erlangen. Dabei können wenige Minuten einen grossen Einfluss auf den Verschattungsgrad der PV-Anlage haben und infolgedessen Ertrag deutlich minimieren. Als Faustregel gilt, dass der Horizont oberhalb des Höhenwinkels von 15 Grad weitestgehend frei ist, damit die Solarstrahlung im Sommerhalbjahr gut genutzt werden kann. In der Abbildung 4 ist dies visuell dargestellt.

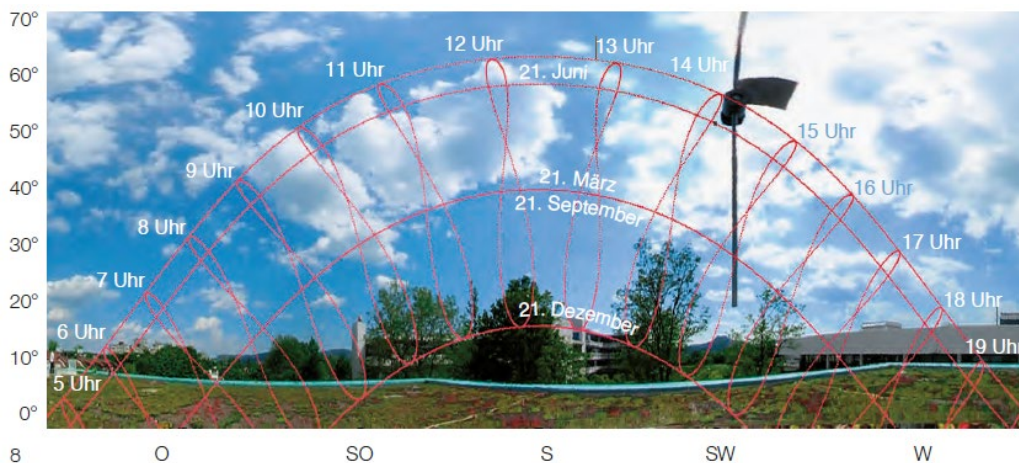


Abb. 4: Minimierung der Verschattung in Sommermonaten (ebd.)

3.3 Wechselrichter

Damit der Solarstrom im Alltag genutzt werden kann, benötigt ein PV-System einen Wechselrichter. Dieser fungiert als Energiemanager und wandelt in erster Linie den erzeugten Gleichstrom in verwendbaren Wechselstrom um. Der überschüssige Strom fließt über den Wechselrichter ins öffentliche Stromnetz (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 25). Ausserdem kann der Wechselrichter je nach Modelltyp einen oder mehrere Maximum Power Point Tracker (MPP) verwalten. Diese sind zuständig, um «bei ständig wechselnden Einstrahlungs- und Temperaturverhältnissen stets die maximale Leistung in das Stromnetz einzuspeisen ...» (ebd., S.27). Dazu gleicht ein MPP-Regler die Wechselrichterspannung der Generatorspannung (Spannung der PV-Module) an. Heutzutage angewandte, gute Wechselrichter besitzen einen Wirkungsgrad von bis zu 96 % (ebd.). Es gilt: je mehr MPP-Eingänge vorhanden sind, desto mehr PV-Stränge können von einem Wechselrichter verwaltet werden. Die Auswahl des passenden Wechselrichters hängt von der Grösse, dem Nutzen und der örtlichen Möglichkeiten des Systems ab. Dabei kann zwischen dem Zentral-, dem Strang- und dem Modulwechselrichterkonzept unterschieden werden. Für grosse Industrieanlagen eignet sich das Zentralwechselrichterkonzept am besten, da nur ein leistungsstarker Wechselrichter zum Einsatz kommt. Laut Weller (ebd.) stehen der aufwendigen Verkabelung die Preisvorteile und der höhere Wirkungsgrad gegenüber. Ist die betreffende Anlage des Öfteren teilweise verschattet, eignet sich das Strangwechselrichtersystem. Bei diesem wird im Falle der Verschattung nur der betroffene Strang ertragstechnisch beeinflusst, nicht aber die gesamte Anlage. Das letzte Konzept der Modulwechselrichter ist aufgrund der individuellen «Mini-Wechselrichter» pro Modul vergleichsweise kostenintensiv und kann sich wegen «... des erhöhten Wartungs- und Überwachungsaufwands ...» bisher nicht durchsetzen (ebd.).

Bei der Wahl des passenden Wechselrichters gilt es folgende Eigenschaften zu beachten:

- Maximale Eingangsspannung / Anzahl MPP-Tracker
- Wirkungsgrad
- Kosten Gerät/Installation

3.4 Konstruktionen

Dieses Kapitel bezieht sich auf die Befestigung von Photovoltaikmodulen an Fassaden. Bei der Montage von rahmenlosen PV-Modulen kommen oft konventionelle Glasbau-techniken zum Einsatz.

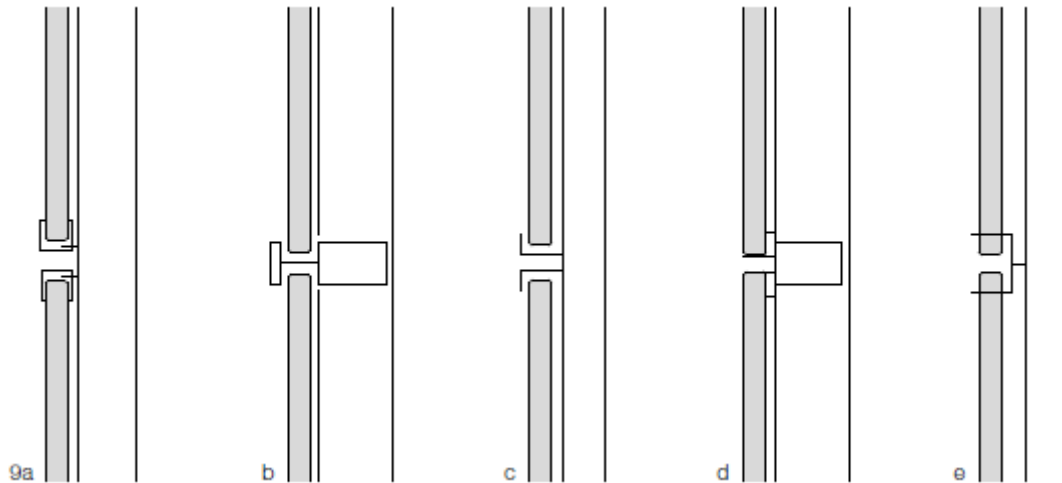


Abb. 5: Montagetechniken rahmenloser PV-Module (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 52)

Die Abbildung 5 zeigt lineare Befestigungsoptionen für rahmenlose PV-Module. Es wird bei der Befestigung zwischen Verschraubung, Klemmung, Klammerung, Klebung und Bohrung unterschieden (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 53). Die Klebung sollte aufgrund der kreislaufwirtschaftlichen Möglichkeiten möglichst vermieden werden. Wichtig zu beachten ist, dass in Deutschland ab acht Meter Höhe «zusätzlich (...) eine mechanische Sicherung der Glasscheiben (...) erforderlich ist» (ebd.). In anderen europäischen Ländern ist diese Regelung meist nichtig und daher eine zusätzliche Sicherung nicht nötig. Dies sollte jedoch vorgängig mit den Behörden am Standort abgeklärt werden.

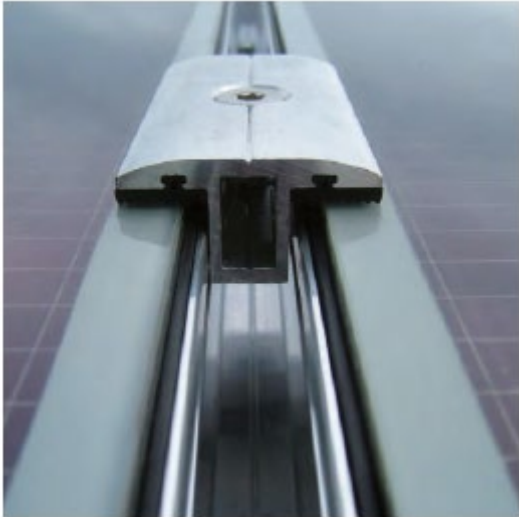


Abb. 6: Punktförmige Halterungen (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 54)

Vorteil einer punktförmigen Halterung (siehe Abb. 6) ist die geringe eigen Verschattung sowie die geringe Schmutzansammlung an den Rändern. Diese können die Leistung des betroffenen Panels stark minimieren. Das architektonische Erscheinungsbild kann ausserdem verfeinert werden. Für Punkthalterungssysteme werden in vielen europäischen Ländern VSG-Gläser benötigt. Dadurch sind nicht alle Module auf diese Befestigungsmethode anwendbar (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 54).

3.5 MLPE & Bypass Dioden

Da die Photovoltaikzellen meist in Reihe geschaltet sind, wirkt sich die Verschattung eines Modules stark auf den gesamten String aus. Das verschattete Modul wird so zum «Stromverbraucher» und lässt den Ertrag der anderen Module im String minimieren oder gar zum Erliegen kommen. Bei dieser Situation kann sich das betroffene Modul stark erhitzen, was die Zellen oder das Einbettungsmaterial beschädigen kann (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 23). Bypass Dioden werden hier eingesetzt, um den Strom ohne Blockade durch das verschattete Modul hindurchzuleiten. Bypass Dioden sind heutzutage in allen gängigen Photovoltaikmodulen verbaut. Zum Teil sogar mehrere pro Modul, um möglichst unterschiedliche Verschattungsszenarien zu umgehen. Wo die Bypass Dioden einfache «Substrings» innerhalb des Modules je nach Ertrag ein oder ausschalten, können MLPE den Ertrag jedes Modules nutzen und so zu einem höheren Endertrag führen. Sie haben die Möglichkeit neben dem Abschalten einzelner Substrings die Ausgangsspannung des betroffenen Modules stufenlos einzustellen, damit der String der Nenn-Eingangsspannung des Wechselrichters entspricht und damit einen möglichst hohen Ertrag aus den Modulen zu holen.

Moderne Wechselrichter besitzen integrierte Maximum-Power-Point Tracker (MPP-Tracker) welche dieselbe Funktion wie MLPE haben, jedoch nicht direkt am Modul sitzen und dadurch nur die Spannung des gesamten Strings optimieren können, nicht aber die der einzelnen Module.

Die Installation der MLPE erfolgt hinter jedem einzelnen Modul (Orduz, Solorzano, Egido, & Roman, 2011, S. 359). Diese Installationen sind jedoch mit weiteren Kosten verbunden (Energie-Experten, 2022) welche in der Amortisations- und Wirtschaftlichkeitsrechnung mit einbezogen werden müssen.

Es sollte darauf geachtet werden, dass nicht in jeder Situation eine Lösung mit MLPE mehr Ertrag oder eine bessere Rendite & Amortisation bringt als mit den standardmäßigen Bypass Dioden. In einer stark verschatteten Situation hilft die Umplatzierung der Anlage oder eine Optimierung der String-Verschaltung, damit nicht-verschattete Module zumindest nicht von den verschatteten beeinflusst werden (siehe Kapitel 7 «Das Projekt»).

4 Wartung

In diesem Abschnitt wird darauf eingegangen, welche Bestandteile wartungsrelevant sind, welche Kosten diese verursachen und in welchem Intervall diese getätigt werden müssen.

Damit die PV-Anlage die bestmögliche Leistung erbringen kann und Fehlfunktionen der Anlage frühzeitig erkannt werden können, sollte ein Wartungsplan erstellt werden. Gemäss den Empfehlungen von Simon Roberts und Nicolo Guariento (Roberts & Guariento, 2009, S. 59) sollten folgende Punkte im Wartungsplan inkludiert werden: die Prüfung und Aufzeichnung der elektrischen Leistung im monatlichen Intervall, die Sichtprüfung des Solarfelds im jährlichen Intervall und die Reinigung der PV-Module im jährlichen Intervall. Durch die monatliche Prüfung der elektrischen Leistung kann ein breites Spektrum an Problemen verhindert werden. Diese Probleme reichen von defekten Leitungen über korrodierte Anschlüsse bis hin zum Ausfall eines Wechselrichters (ebd., S.60). Im Normalfall muss bei einer Photovoltaikfassade keine Reinigung vollzogen werden, da die Reinigung in einem gewissen Grad vom ablaufenden Regenwasser übernommen wird. In Gebieten, wo die Luftqualität sehr schlecht ist und die Schmutzpartikel auf den PV-Modulen zurückbleiben, was ihre Leistung beeinträchtigt, sollte die Fassade mit PV-Modulen genauso oft gereinigt werden wie die anderen Glas- oder Metallfassaden der Gebäude. Für den Austausch der Fassadenmodule wird eine Fassadenbefahr Anlage benötigt. Für den Austausch von grösseren und schwereren Modulen müssen für den Austausch spezielle Verfahren entwickelt werden (ebd., S.74).

Neben den eigentlichen PV-Modulen, müssen auch die Wechselrichter und falls vorhanden, die MLPE gewartet werden. Ihre erwartete Lebensdauer liegt nicht wie bei den Modulen bei 20 – 25 Jahren, sondern müssen nach ca. 10 – 15 Jahren ausgetauscht werden (Energie-Experten, 2021). Meist werden die Kosten für die Wartung über einen Wartungsvertrag mit einem PV-Unternehmen gedeckt. Diese Mehrkosten, vor allem für die MLPE, muss in die Wirtschaftlichkeitsberechnung mit einbezogen werden. Für den Entscheid für oder gegen die MLPE sollte ein Fachmann beigezogen werden.

5 PV-Planungsvorgehen Industriepartner

In diesem Kapitel wird das «State-of-the-Art» Planungsverfahren von PV-Anlagen am Beispiel des Industriepartners Basler & Hofmann analysiert.

Als erster Schritt wird ein 3D Modell auf die Plattform PV Syst geladen, um die jährlichen Einstrahlungswerte in kWh/m²a auf den einzelnen Fassadenflächen zu erhalten (siehe Abb. 7).

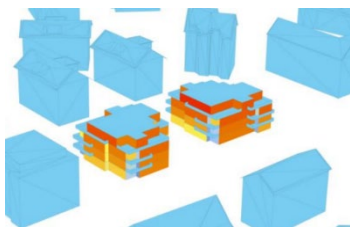


Abb. 7: Sonneneinstrahlung
B&H (PV Syst)

Als nächster Schritt werden die Flächen in einzelne Zonen unterteilt (siehe Abb.8). Diese Unterteilung ist nach der Hausnummer und Fassadenausrichtung gegliedert. Den einzelnen Flächen werden pro Ausrichtung Nummern vergeben. Die Einstrahlungswerte kWh/m²a werden dann manuell pro Zone im Excel eingetragen (siehe Tab. 1).

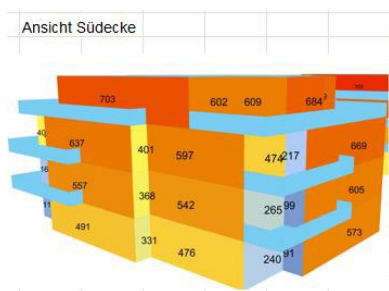


Abb. 8: Zoneneinteilung B&H (PV Syst)

	Haus A					
	Stock	SO 1	SO 2	SO 3	SO 4	SO 5
Einstrahlung						
Aus Simulation	EG (unten)	331	240	573	430	n.a.
(kWh/m²a)	OG 1 (unten)	368	265	605	471	n.a.
	OG 2 (oben)	401	474	669	560	n.a.
	OG 3 (attika)	602	684	669	495	645

Tabelle 1 Einstrahlungswerte B&H (Excel A10)

Um die Ertragsleistung der PV-Fassaden mit und ohne MLPE miteinander zu vergleichen, werden zwei verschiedene Varianten erstellt. Bei der ersten Variante werden die erhaltenen Jahresertragswerte von 4 Unterteilungen auf 3 reduziert. Für den Wert «EG (unten)» und den Wert «OG1 (unten)», wird der kleinere Ertragswert übernommen und

im Wert «unten» zusammengefasst. Dies aus dem Grund, weil die Module mit dem geringeren Einstrahlungswert den Stromfluss definieren. Die Werte «OG2 (oben)» und «OG3 (attika)» werden neu als Werte «oben» und «attika» wiedergegeben (siehe Tab. 2 & 3).

		Haus A			
		Stock	SO 1	SO 2	SO 3
Einstrahlung					
Aus Simulation	EG (unten)		331	240	573
(kWh/m2a)	OG 1 (unten)		368	265	605
	OG 2 (oben)		401	474	669
	OG 3 (attika)		602	684	669

Tabelle 2: Grundlage Einstrahlungswerte (Excel A10)

Var. 1 mit 3 Stringebenen	unten	331	240	573
(kWh/m2a)	oben	385	370	637
	attika	602	684	669

Tabelle 3: Berechnete Einstrahlungswerte (Variante 1) (Excel A10)

Bei der zweiten Variante ist das Vorgehen unterschiedlich. Es wird versucht, die Optimierung des Ertrages mit MLPE in der Berechnung der zweiten Variante wiederzugeben. Von den Einstrahlungswerten «EG (unten)» und «OG1 (unten)» wird der Mittelwert berechnet und neu als Wert «unten» definiert. Um die Einstrahlungsoptimierung mit dem Wert «OG2 (oben)» zu simulieren, wird folgende Berechnung vollzogen:

$$(OG2 (oben) * 2 + OG1 (unten) * 1) : 3 = Wert «oben»$$

Diese Berechnung wird nun dem Wert «oben» zugeteilt. Der Wert «OG3 (attika)» wird ohne Berechnung neu dem Wert «attika» zugeteilt. Daraus resultierten die Werte für die Variante zwei, welche in der Tabelle 5 ersichtlich sind. Die Tabelle 4 enthält die Ausgangswerte.

		Haus A			
		Stock	SO 1	SO 2	SO 3
Einstrahlung					
Aus Simulation	EG (unten)		331	240	573
(kWh/m2a)	OG 1 (unten)		368	265	605
	OG 2 (oben)		401	474	669
	OG 3 (attika)		602	684	669

Tabelle 4: Grundlage Einstrahlungswerte (Excel A10)

Var. 2 mit Moduloptimizern (kWh/m2a)	unten	350	253	589
	oben	390	404	648
	attika	602	684	669

Tabelle 5: Berechnete Einstrahlungswerte (Variante 2) (Excel A10)

Die Leistung der Module wird in der Tabelle 5 mit der folgenden Formel berechnet.

$$\text{Anzahl Module} * \text{Modulnennleistung}(kWp/\text{Modul}) = \text{Leistung}$$

Um einen Vergleichswert zwischen den beiden Varianten zu generieren, wird die Leistungsdifferenz ermittelt:

$$(\text{Var.1 (kWh/m2a)} - \text{Var.2 (kWh/m2a)}) : \text{Var.2 (kWh/m2a)} = \text{Leistungsdifferenz (\%)}$$

Der Ertrag pro Jahr wird mit folgender Formel ermittelt:

$$kWh/m2a * \text{Leistung (kWp)} * PR (\%) = \text{Ertrag (kWh/a)}$$

Die Leistungs- und Ertragsberechnung wird bei beiden Varianten identisch berechnet (siehe Tab. 6). Anschliessend werden bei beiden Varianten die einzelnen Ertragswerte pro Fassadenfeld miteinander addiert. Im Endergebnis ist die Ertragsdifferenz von ca. 1000 kWh/a ersichtlich. Dieser Wert sollte mittels genauerer Berechnung überprüft werden.

Ertrag	unten	1'858	1'343	587	1'822	5'610		1'667	1'067	409	1'226	4'369		9'980
Var. 1 mit 3 Stringebenen (kWh/a)	oben	2'583	1'660	693	2'194	7'131		1'766	1'322	724	1'896	5'708		12'839
	attika	249	557	70	531	1'407		176	436	64	577	1'252		2'660
	total	4'691	3'560	1'350	4'548	14'149		3'609	2'825	1'197	3'699	11'329		25'478
Var. 2 mit Moduloptimizern (kWh/a)	unten	1'926	1'442	622	1'947	5'937		1'691	1'147	452	1'382	4'671		10'608
	oben	2'641	1'700	706	2'235	7'282		1'783	1'354	770	2'015	5'922		13'204
	attika	249	557	70	531	1'407		176	436	64	577	1'252		2'660
	total	4'817	3'698	1'397	4'713	14'626		3'650	2'937	1'285	3'974	11'845		26'471

Tabelle 6: Gesamtertragswerte (kWh/a) Variante 1 und Variante2 (Excel A10)

Die String-Verschaltungsplanung wird bis anhin manuell erstellt. Des Weiteren wird für die Ermittlung des Ertrags pro String das Modul mit dem geringsten Ertragsjahreswert auf den gesamten String hochgerechnet (siehe Anhang A1). Dabei ist zu beachten, dass diese Berechnung nur eine Annäherung ist. Die genaue Berechnung müsste wie im Kapitel 7.6.4 mit den stündlichen Werten berechnet werden.

6 Hindernisse

In diesem Kapitel wird das Verschattungsproblem in Bezug auf spezifische String-Verschaltungen aufgezeigt. Es stellt das Ausgangswissen bereit für das anschliessend umgesetzte Projekt.

6.1 Verschattung / String-Verschaltung

Bei Gebäuden innerhalb einer Überbauung ist die Wahrscheinlichkeit gross, dass diese gegenseitig Schattenwürfe generieren. In der Planung von PV-Anlagen wird deshalb darauf geachtet, dass die PV-Anlage meist auf der Nordseite des Geländes platziert werden. Dies aus dem Grund, weil diese Bereiche längere Zeit unbebaut bleiben. Nichtsdestotrotz kann die Verschattung nicht ganzheitlich vermieden werden. Viele Faktoren beeinflussen die Leistungsfähigkeit von PV-Anlagen, die Verschattung hat dabei den grössten Anteil (Roberts & Guariento, 2009, S. 36). Die Panels werden so weit als möglich in Reihe zu einem String zusammen geschaltet, dadurch kann sich ihre Spannung addieren. Es fliesst durch jedes Modul, welches sich im gleichen String befindet, exakt der gleiche Strom (ebd., S.37). Dies bedeutet, wenn ein Modul verschattet ist, können die Module im selben String ihr Potenzial nicht ausschöpfen und produzieren gleich viel Strom wie das verschattete Modul.

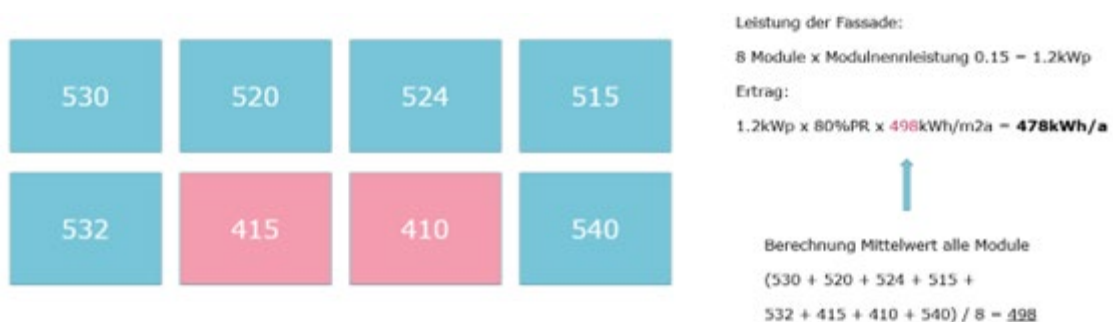


Abb. 9: Herkömmliche Berechnung ohne String-Verschattung

In der Abbildung 9 ist die herkömmliche Berechnung von B&H ersichtlich. In der Abbildung 10 ist die selbe Situation, mittels der genauen Berechnung abgebildet. Der Ertragsunterschied dieser beiden Berechnungsmethoden ist enorm und zeigt auf, dass die Verwendung des Mittelwerts nicht zielführend ist.

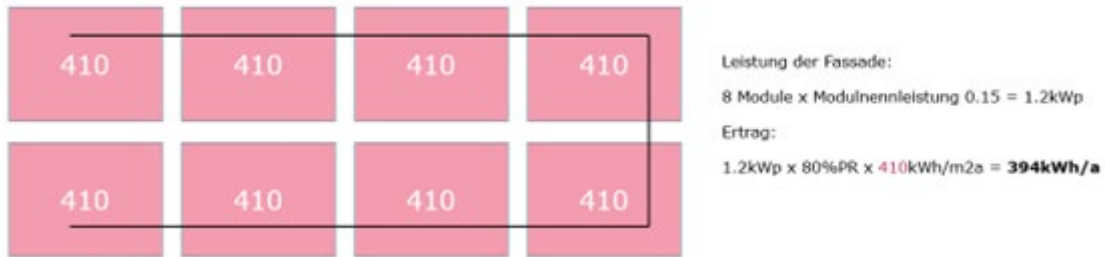


Abb. 10: Genaue Berechnung String-Verschaltung

Ein einzelnes Modul sorgt so für einen Leistungsverlust im gesamten String. Aufgrund dieses Problems sind die heutigen PV-Module mit Bypass-Dioden versehen. Der Leistungsverlust kann aufgrund dieser verringert werden, das verschattete Modul kann überbrückt werden und der Leistungsverlust fällt geringer aus. Bei komplexen Verschattungssituationen ist diese Überbrückung nicht ausreichend. Für dieses Problem sollen in Zukunft Simulationen Abhilfe schaffen (Theele, 2016).

Jede Stunde im Jahr besitzt einen spezifischen Sonnenstand, welcher sich über den Azimut Winkel sowie dem Höhenwinkel zwischen dem Sonnenstand und dem Horizont definiert. Daraus resultiert, dass der Schattenwurf und die Einstrahlungsintensität in jeder Stunde verschieden ist und dadurch schlussfolgernd auch der Ertrag der Module (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 13). Um eine genaue Berechnung des Ertrags pro String zu erhalten, muss die Berechnung in einem stündlichen Intervall ausgeführt werden. In der Abbildung 11 ist ein String simuliert, welcher aus zwei Modulen besteht. Mit der jährlichen Berechnung liegt der Ertrag bei 105 kWh/a, mit der monatlichen Berechnung bei 101 kWh/m2a. Dies aus dem Grund, da pro Monat immer der kleinere Ertragswert mal die Anzahl Module im String gerechnet wird. Bei der Berechnung mit den stündlichen Werten wäre der Gesamtertrag nochmals um einiges geringer, da anstatt mit 12 Ertragswerten, mit 8760 Werten gerechnet wird.



Abb. 11: Ertragsberechnung pro Jahr und pro Monat

6.2 Problemstellungen seitens Industriepartner

Wie im Kapitel 3.2.3 beschrieben, haben einzelne verschattete Module einen grossen Einfluss auf die Ertragswerte. Der Einsatz von MLPE ermöglicht die Überwachung und Optimierung des Energieertrags jedes einzelnen Moduls. Dadurch wird sichergestellt, dass verschattete Module die Leistung der nicht verschatteten Module nicht beeinträchtigen und somit der Gesamtertrag einer PV-Anlage nicht reduziert wird (Energie-Experten, 2022). Gemäss Eric Langenskiöld ist bei den MLPE die Lebensdauer ein grosser Nachteil. Im Vergleich zu den dauerhaften Photovoltaik-Modulen, welche eine Lebensdauer von ca. 40 Jahren aufweisen, liegt die Ausfallquote bei den MLPE bereits nach 15 Jahren bei ca. 50 %. An den Fassaden sind zudem die Auswechslungen dieser MLPE erschwert, da die Zugänglichkeit meist nur mit einer Hebebühne gewährleistet ist. Des Weiteren sind diese MLPE hinter jedem Modul verbaut, für die Auswechslung müsste deswegen das gesamte Photovoltaik-Modul demontiert werden. Daraus lässt sich schlussfolgern, dass die Unterhaltskosten nach dem 15 Lebensjahr stark ansteigen (siehe Anhang A1).

Wie im Kapitel 5 beschrieben, wird die Ertragsberechnung mittels Excel Liste und die String-Verschaltung manuell geplant. Diese Prozesse sind sehr zeitaufwändig, die String-Verschaltung Planung ist nicht optimiert und die Ertragsberechnungen nicht exakt. Der Industriepartner erhofft sich durch den Output dieser Arbeit eine Minimierung des Planungs-Aufwands und ein optimiertes Vorgehen in der String-Verschaltungsplanung, welche den Einsatz von MLPE ablösen soll.

7 Das Projekt

In diesem Kapitel wird das erarbeitete Projekt vorgestellt. Zuerst wird die Optimierung der Planung erläutert. Anschliessend werden alle Grundlagen vorgestellt und das Vorgehen beschrieben. Darauf folgen dann die vier verschiedenen Levels der Scripts von Level 1 bis 4. Pro Level werden jeweils die spezifischen Eigenschaften aufgezeigt und näher erklärt. Zum Schluss wird die erforderliche Rechenleistung für den Betrieb des Skripts aufgezeigt.

7.1 Optimierung der Planung

Das vorliegende Projekt optimiert den gesamtheitlichen Prozess der PV-Fassaden Planung von Basler & Hofmann. Darin enthalten sind alle Planungsphasen von der Machbarkeit bis zur Projektierung. Diese Planungsphasen werden mittels den Scripts Level 1-4 abgebildet.

Die Machbarkeit wird mit dem Skript Level 1 abgedeckt. Die Zielgruppen für das Skript Level 1 sind Entwicklungsfirmer, Investoren, Makler, Banken und Architekten. Die definierten Zielgruppen haben die Möglichkeit, ihre Gebäude- und Umgebungsmodelle in das Programm Rhino zu laden und die einzelnen Fassadenflächen zu analysieren. In diesem Skript ist die Ertragsberechnung und Wirtschaftlichkeitsberechnung mittels einer groben Annäherung inkludiert, basierend auf jährlichen Ertragswerten. Dadurch kann eruiert werden, welche Flächen eines Gebäudes genügend Ertrag erbringen und somit durch eine PV-Anlage nutzbar wären. Weiter kann manuell überprüft werden, welcher Modultyp am meisten Ertrag erbringt und welche Rendite damit erreicht werden kann.

Die Vorprojekt Phase wird mit dem Skript Level 2 abgedeckt. Die Zielgruppen für das Skript Level 2 sind PV-Planer und Gebäudetechnikplaner. Die Stufe Skript Level 2 ist detaillierter gegliedert, wobei die Planer die Möglichkeit haben, ihre Gebäude- und Umgebungsmodelle inklusive vordefinierter PV-Module und Strings in das Programm Rhino zu laden. Sie haben die Möglichkeit, verschiedene Varianten der Modultypen und der String-Verschaltung zu analysieren. In diesem Skript ist die Ertragsberechnung und Wirtschaftlichkeitsberechnung mittels einer genaueren Berechnung inkludiert, basierend

auf monatlichen Ertragswerten. Dadurch kann eruiert werden, welcher Modultyp, welche Modulanordnung und welche String-Verschaltung vorteilhaft ist. Im Output ist ersichtlich, wie viel Ertrag und welche Rendite die überprüften Varianten einbringen.

Die Phase Projektierung wird mit dem Skript Level 3 abgedeckt. Dabei sind die Zielgruppen für das Skript Level 3 PV-Planer und Gebäudetechnikplaner. In der Stufe Skript Level 3 besteht die Möglichkeit für die Planer ihre Gebäude- und Umgebungsmodelle inklusive vordefinierten PV-Modulen in das Programm Rhino zu laden. Die Planer haben die Möglichkeit, die optimale String-Verschaltung für ihre PV-Anlage zu generieren und den Wechselrichtertyp darauf abzustimmen. In diesem Skript ist die Ertrags- und Wirtschaftlichkeitsberechnung mittels einer genaueren Berechnung inkludiert, basierend auf monatlichen oder stündlichen Ertragswerten. Es gibt die Möglichkeit, die bestmögliche String-Verschaltung und den darauf abgestimmten Wechselrichter mit manuellen Parametereinstellungen zu eruieren. Dieser Schritt kann auch automatisch mittels eines Algorithmus erstellt werden. Im Output ist ersichtlich, wie viel Ertrag und welche Rendite die verschiedenen String-Verschaltungen einbringen. Zusätzlich ist die prozentuale Differenz zu einer herkömmlichen String-Verschaltung und der Variante mit MLPE ersichtlich.

Das Skript Level 4 ist auf die Phase des Vorprojekts und der Projektierung ausgelegt. Die Zielgruppen für das Skript «automatische Modul Generierung» sind Architekten, PV-Planer und Gebäudetechnikplaner. Das Skript bietet die Möglichkeit für die Planer, ihre Gebäude- und Umgebungsmodelle in das Programm Rhino zu laden. Auf den ausgewählten Fassadenflächen werden anschliessend automatisch PV-Module generiert. Weiter ist es möglich, verschiedene Varianten mit unterschiedlichen Modultypen zu erstellen. Die erstellten Module sind mit der Optimierung der String-Verschaltung und der Auswahl des Wechselrichtertyps von Skript Level 3 verknüpft. Im Output ist gleich wie im Skript 3 ersichtlich, wie viel Ertrag und welche Rendite die verschiedenen String-Verschaltungen einbringen. Zusätzlich ist die prozentuale Differenz zu einer herkömmlichen String-Verschaltung und der Variante mit MLPE ersichtlich.

7.2 Grundlagen

In diesem Kapitel wird erläutert, auf welchen Grundlagen-Daten die Berechnungen in den Scripts basieren. Als Grundlage für die Berechnung des Ertrages dient die Excel Tabelle, welche von Basler & Hofmann (B&H) zur Verfügung gestellt wird (siehe Anhang A10). Sie stellt die ursprüngliche Berechnungsweise von B&H dar, welche anschliessend im Skript automatisiert werden soll. Die Leistungsdaten und Abmessungen der Modultypen, welche in den Skripts hinterlegt sind, werden vom Datenblatt (siehe Anhang A11) entnommen. Die hinterlegten Wechselrichterdaten sind in den Datenblättern (siehe Anhang A 17-19) ersichtlich, dabei wird darauf geachtet, dass Wechselrichter mit möglichst unterschiedlichen Eigenschaften verwendet werden.

Die Grundlage für die Kosten der MLPE ist die Tabelle «SolarEdge P-Serie», welche in der Abbildung 12 ersichtlich sind. Es wird für die Berechnung ein Durchschnittswert von 60 CHF pro MLPE angenommen.

SolarEdge P-Serie	Anwendung	Brutto-Preise pro Stück
P300	für Module mit 60 Zellen	47,67 €
P350	für Hochleistungsmodule mit 60 Zellen und 72 Zellen	50,49 €
P370	für leistungstärkere 60- und 72-Zellen Module	-
P404	für Module mit 60 Zellen und 72 Zellen, kurze Stränge	55,90 €
P405	für Dünnschichtmodule	57,07 €
P500	für Module mit 96 Zellen	55,00 €
P600	für 2 PV Module mit 60 Zellen	62,00 €
P700	für 2 PV Module mit 72 Zellen	67,32 €
P800p	Parallel-Verschaltung von 2 Modulen mit 96 5"-Zellen	81,50 €
P800s	Reihen-Verschaltung von 2 bifacial- oder Hochleistungsmodulen	-

Abb. 12: Übersicht Preise elektrische Optimizer (Energie-Experten, 2022)

Als Grundlage zur Überprüfung der Skripts werden eigene, abstrahierte 3D-Modelle verwendet. Diese sind ausschliesslich in Allplan entstanden. Es handelt sich dabei je nach Level des Skripts um einfache Kuben oder um komplexere Gebäude mit realen Kontext-Geometrien. Die verwendeten Modelle sind in den unteren Abbildungen visualisiert (siehe Abb. 13, 14, 15 & 16).

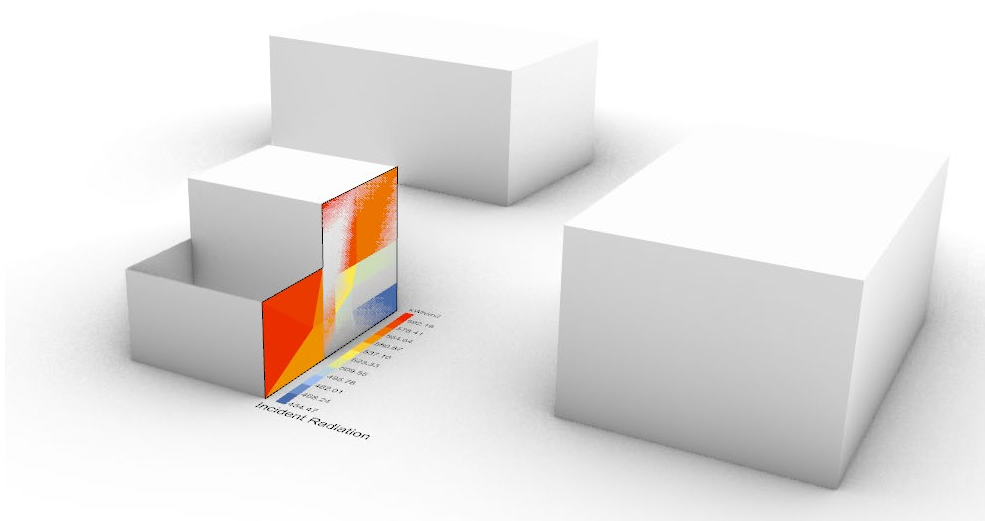


Abb. 13: Modell für Skript Level 1

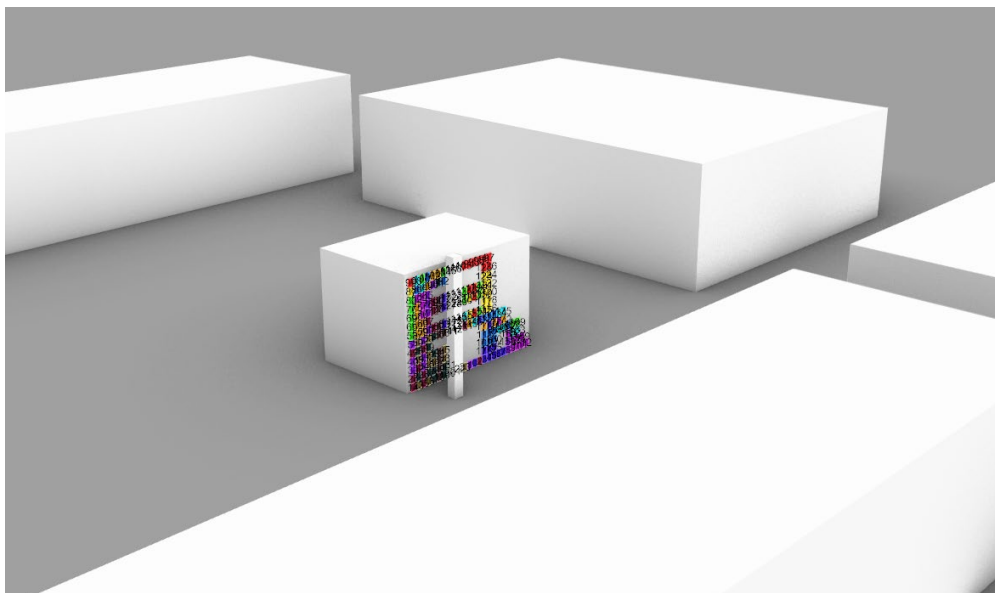


Abb. 14: Modell für Skript Level 2

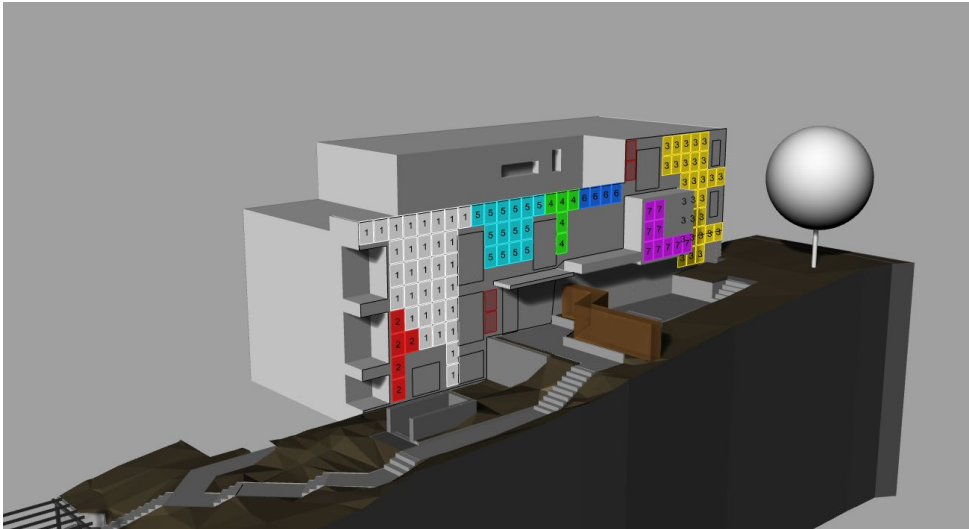


Abb. 15: Modell für Skript Level 3

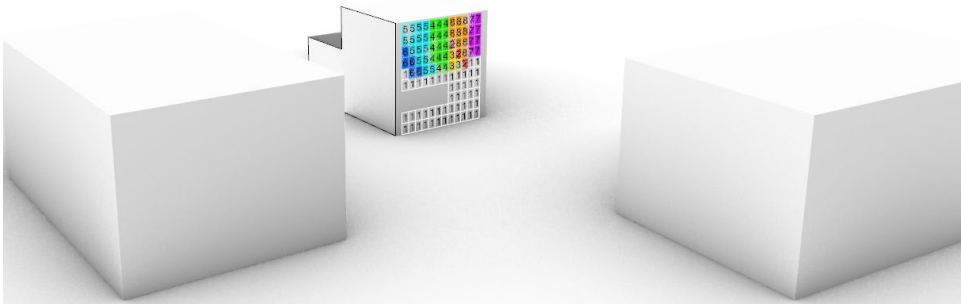


Abb. 16: 22 Modell für Skript Level 4

7.3 Vorgehen

Zunächst wurde das Planungsvorgehen von Basler & Hofmann analysiert, wie es im Kapitel 5 detailliert beschrieben ist. Dies war von Bedeutung, um ein grundlegendes Verständnis für die Planung zu erlangen. Zusätzlich wurde eine Recherche zu den einzelnen Komponenten einer PV-Anlage durchgeführt, um den Zusammenhang zwischen ihnen zu verstehen und sie in unser Tool integrieren zu können. Diverse Gespräche mit Experten sorgten für eine grobe Übersicht darüber, welche Faktoren bei den Berechnungen und der Optimierung beachtet werden sollten. Für jedes Projekt erstellt Basler & Hofmann ein detailliertes Excel Sheet, um die Ertragswerte für die verschiedenen Fassadenflächen zu berechnen. Dieses Excel Sheet wurde analysiert und alle Input-, Output-Daten und Berechnungsformeln aufgelistet.

Nach den Analyse- und Recherchearbeiten wurde als erster Schritt in Rhino ein simples Testobjekt erstellt. Dieses Testobjekt konnte für die Sonnenanalyse mit dem Programm Grasshopper und dem Plugin LadyBug verwendet werden (Ladybug, 2023). Das «Standard»-Planungsvorgehen basiert auf Annahmen gestützt auf Visualisierungen der geplanten String-Verschaltungen durch Basler & Hofmann (siehe Abbildung 17) und den gemessenen Einstrahlungswerten auf einen einfachen Kubus. Die Annahme ist, dass die Strings aufgrund der aufsteigenden Einstrahlungsleistung von Gebäudesockel bis Dachkante immer in horizontaler Richtung verkabelt werden. Dieses Standardvorgehen wird in dieser Arbeit als Vergleichsvariante zur erarbeiteten Optimierung verwendet (siehe Abbildung 18).

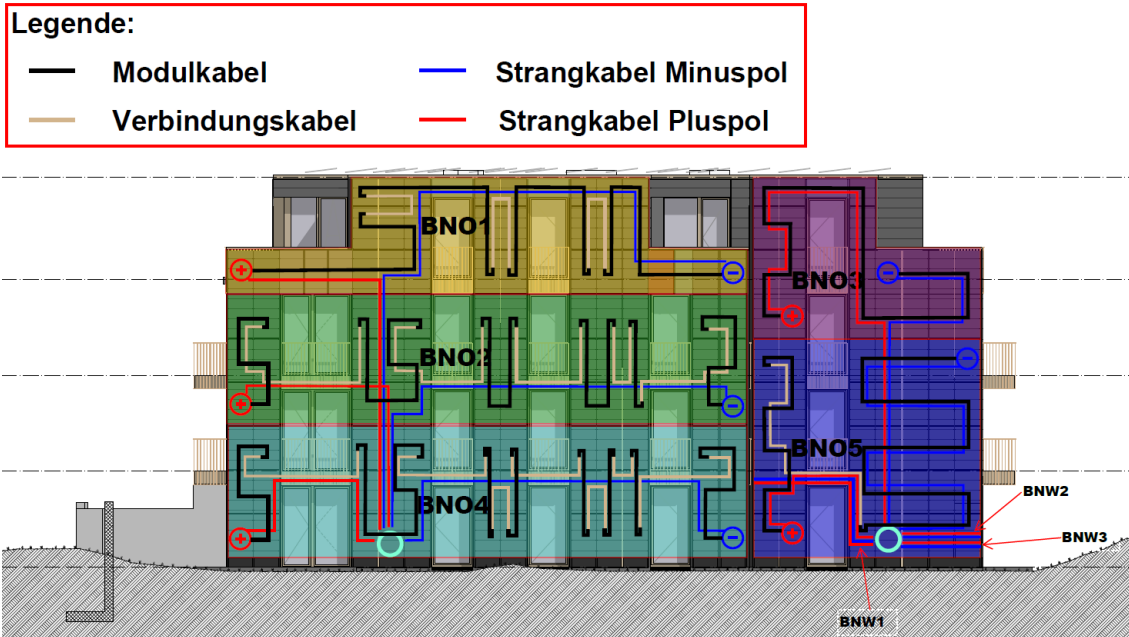


Abb. 17: Vorlage Basler & Hofmann (Anhang A12)

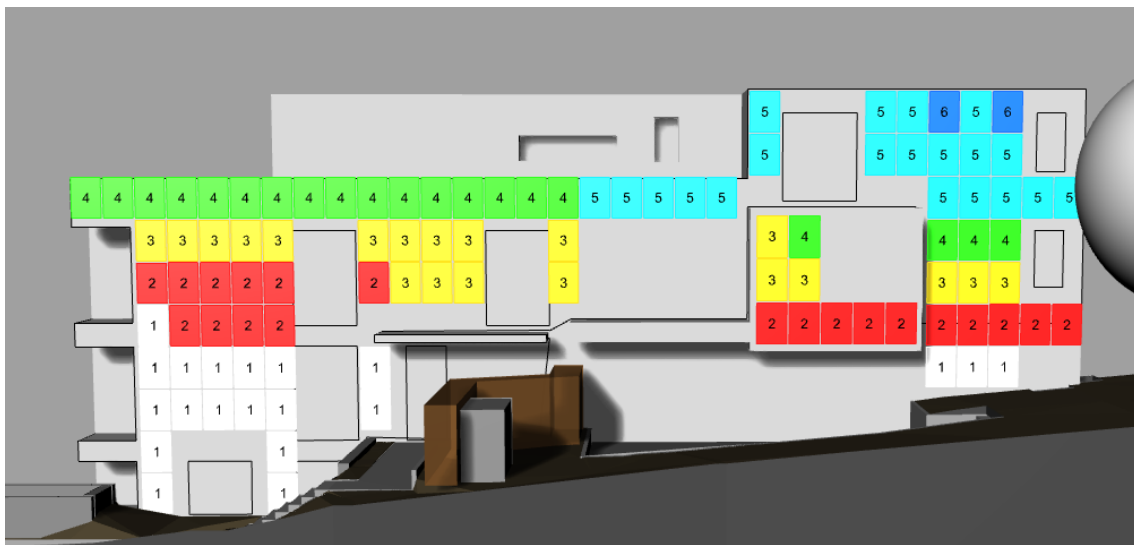


Abb. 18: "Standard" Auslegung der Strings

Bei der String-Verschaltungsoptimierung wird darauf geachtet, dass die Logik darauf ausgelegt ist, alle Stufen der Detaillierung, der Ertragsberechnung und der String-Verschaltungsoptimierung zu bewältigen. Dies war bei der iterativen Verbesserung der String-Verschaltungsoptimierung essenziell. Des Weiteren kann die Logik der String-Verschaltung gewährleisten, dass eine detaillierte Berechnung und die bestmögliche String Kombination mit stündlichen Werten funktioniert. Bei der Erarbeitung war es essenziell, die Detaillierung in verschiedene Stufen einzuteilen. Zum einen bietet das

Plugin Tool Lady Bug nur die Grundfunktion jährliche Einstrahlungswerte oder einzelne eingestellte Zeiträume auszugeben (Ladybug, 2023). So war es im ersten Schritt nur möglich, das Skript mit der Stufe 1 zu optimieren und zu berechnen. Durch das Vorlagenskript von Amr Elesawy (siehe Anhang A25), konnten Anpassungen am Ausgangsskript mit Lady Bug Integration vorgenommen werden. Dadurch war es möglich Einstrahlungsdaten in monatlichen, verschiedenen Azimut Unterteilungen und in stündlichen Gliederungen auszugeben (Chris, 2022). Dank dieser Anpassungen im Skript ist es möglich die Stufe 2-5 zu implementieren.

Level of Detail Ertragsberechnung / String-Verschaltung Optimierung

- Stufe 1:** Ertragsberechnung mit jährlichen Ertragswerten / String-Verschaltung Optimierung mit jährlichen Ertragswerten
- Stufe 2:** Ertragsberechnung mit monatlichen Ertragswerten / String-Verschaltung Optimierung mit jährlichen Ertragswerten
- Stufe 3:** Ertragsberechnung mit stündlichen Ertragswerten / String-Verschaltung Optimierung mit jährlichen Ertragswerten
- Stufe 4:** Ertragsberechnung mit stündlichen Ertragswerten / String-Verschaltung Optimierung mit monatlichen Ertragswerten
- Stufe 5:** Ertragsberechnung mit stündlichen Ertragswerten / String-Verschaltung Optimierung mit verschiedenen Azimut Ertragswerten

7.4 Level 1 Ertragsberechnung Fläche

7.4.1 Benutzerinteraktion

Dieses Skript ist grundsätzlich in sehr frühen Phasen der Designanalyse und von der in Kapitel 7.1 definierten Zielgruppe zu verwenden. Auf Basis einer Sonnenanalyse und den manuellen Wirtschaftlichkeitsfaktoren, wird die zu erwartende Rendite sowie die Amortisationsdauer einer Fassade berechnet.

Neben dem Wetterfile, welches in der Schweiz meist auf dem Standort Genf basiert, soll vom Benutzer im Rhino-Fenster die zu berechnende Fläche angegeben werden. Anschliessend kann der Nutzer die Eingabefläche überprüfen, um sicherzustellen, dass die Fläche korrekt ausgerichtet ist. Für die Berechnung der Einstrahlung müssen alle relevanten Kontext-Geometrien, welche einen Schattenwurf auf die Fassade generieren, angewählt werden. Die Kontextdaten können zum Beispiel von der Plattform «Swisstopo» abgerufen werden. Des Weiteren hat der Nutzer die Auswahlmöglichkeit, mit welchem Photovoltaik Modultyp die Ertragsberechnung durchgeführt werden soll (siehe Abb. 19).

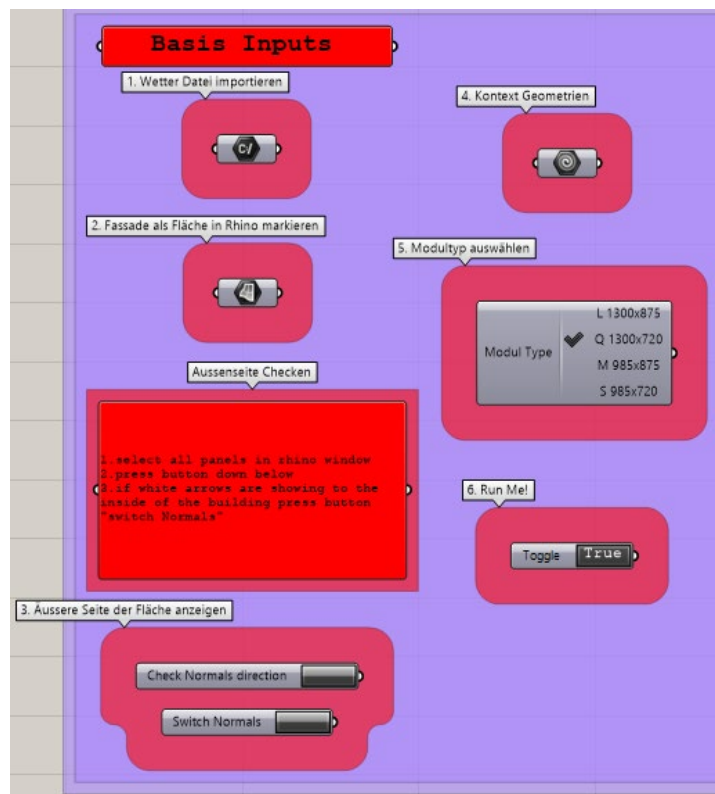


Abb. 19: Basis Inputs Level 1

Neben den oben genannten Auswahlmöglichkeiten, hat der Nutzer die Möglichkeit mittels des Wirtschaftlichkeit-Abschnitts die Berechnung zu beeinflussen. Dieses umfasst den Eigenverbrauch in Prozent des Gesamtertrages, die Investitions- und Instandhaltungskosten, den Förderbeitrag der Investitionskosten, den Strompreis, die Eigenvergütung, sowie die zu berechnenden Jahre (siehe Abb. 20).

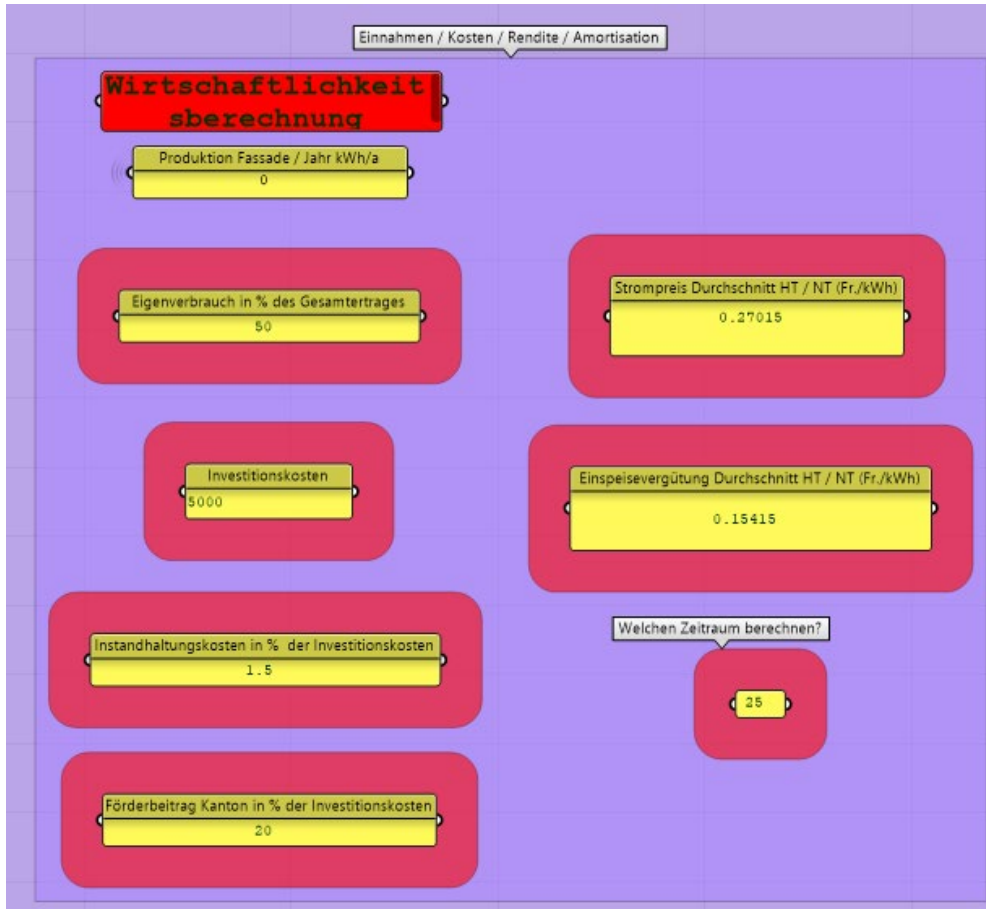


Abb. 20: Wirtschaftlichkeit Inputs

7.4.2 Plugins und Components

Für dieses Skript verwendete Plugins:

- LunchBox | ver: 2020.6.30.0 (Miller, 2023)
- MetaHopper | ver: 1.2.2 (andheum, 2019)
- Ladybug Tools | ver: 1.6.0 (Ladybug Tools, 2023)

Für dieses Skript verwendete Components:

(siehe Anhang A2)

7.4.3 Funktionsweise

Grundsätzlich wird für die Einstrahlungsberechnung das Tool «LB Incident Radiation» aus dem Ladybug Plugin (Ladybug Tools, 2023) verwendet. Die Inputs für diese Berechnung werden im Kapitel 7.4.1 durch den Benutzer vorgängig angegeben. Über die Eingabe der Modultypen kann die Performance Ratio (PR) sowie die Modulkapazität in kWp/Modul verändert werden. Ausserdem hat der Modultyp Einfluss auf die Rastergrösse, mit welcher die Einstrahlungsberechnung durchgeführt wird. Es wird jedes Modul als eine einzelne Rasterfläche verwendet. Dadurch wird am Ende für jedes Panel ein einzelner Einstrahlungswert pro Jahr ausgegeben. Mit der Multiplikation aus dem Gesamtleistungswert der Fassade in kWp, dem summierten Einstrahlungswert aller Module sowie dem PR-Faktor, wird der endgültige Ertrag pro Jahr in kWh/a ausgegeben (siehe Anhang A23). Die Wirtschaftlichkeitsrechnung wird auf Grundlage des Berechnungsdokumentes der RESiQ (Kutschera, 2014) in Python erstellt (siehe Anhang A22) und ist im Level 3 (siehe Kapitel 7.6.5) genauer erläutert.

7.4.4 Output

Der Output ist so gegliedert, dass zuerst die relevanten Einnahmequellen sowie die Gesamteinnahmen pro Jahr und für die zu berechnenden Jahre angegeben werden. Danach werden im selben Format die Kosten ausgegeben. Die Rendite und die Amortisationsdauer werden separat ausgegeben. Rechts neben den Kosten und Einnahmen wird der Gesamtenergieertrag der Fassade auf ein Jahr ausgegeben.

Der Output dieses Skripts stellt eine einfache Berechnung des möglichen Zustandes der Fassade mit den angegebenen Modulen an diesem Standort dar. Es wird keine zyklische Erneuerung von gewissen Modulen oder anderen elektronischen Komponenten wie dem Wechselrichter mit in die Kosten einbezogen (siehe Abb. 21).

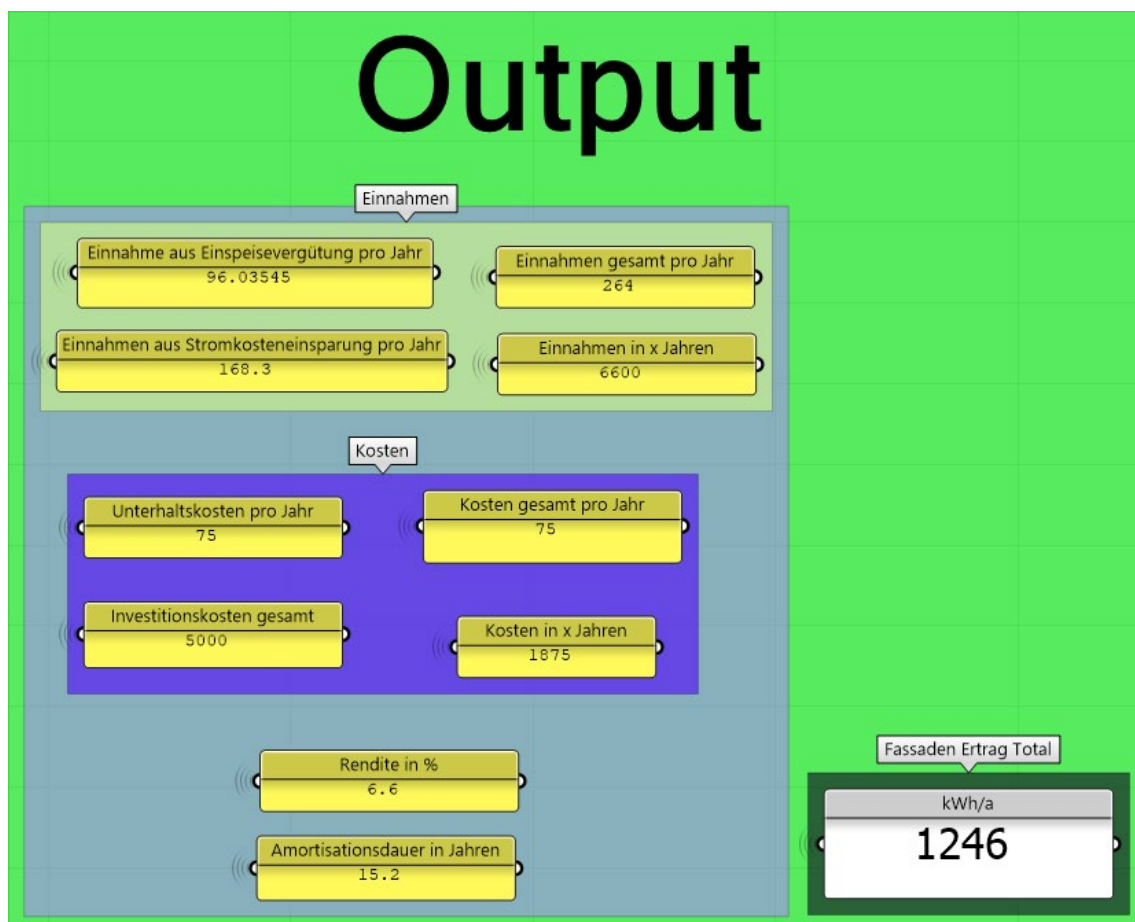


Abb. 21: Output Skript Level 1

7.5 Level 2 Berechnung vorgegebener Strings

7.5.1 Benutzerinteraktion

Dieses Skript ist vorwiegend auf die Vorprojektphase und die in Kapitel 7.1 definierten Zielgruppen ausgelegt. Im Gegensatz zum Level 1 (siehe Kapitel 7.4) bringt das Level 2 die Möglichkeit, mit Panel Geometrien zu arbeiten. Neben den Standardinputs wie in Level 1 beschrieben kann im Level 2 zusätzlich die String Gruppierung selbst definiert und daraus die Ertrags- sowie Wirtschaftlichkeitsberechnung abgeleitet werden. Wichtig zu beachten ist, dass der Modultyp die Dimensionen der importierten Panel-Geometrien aufweist (siehe Abbildung 22).

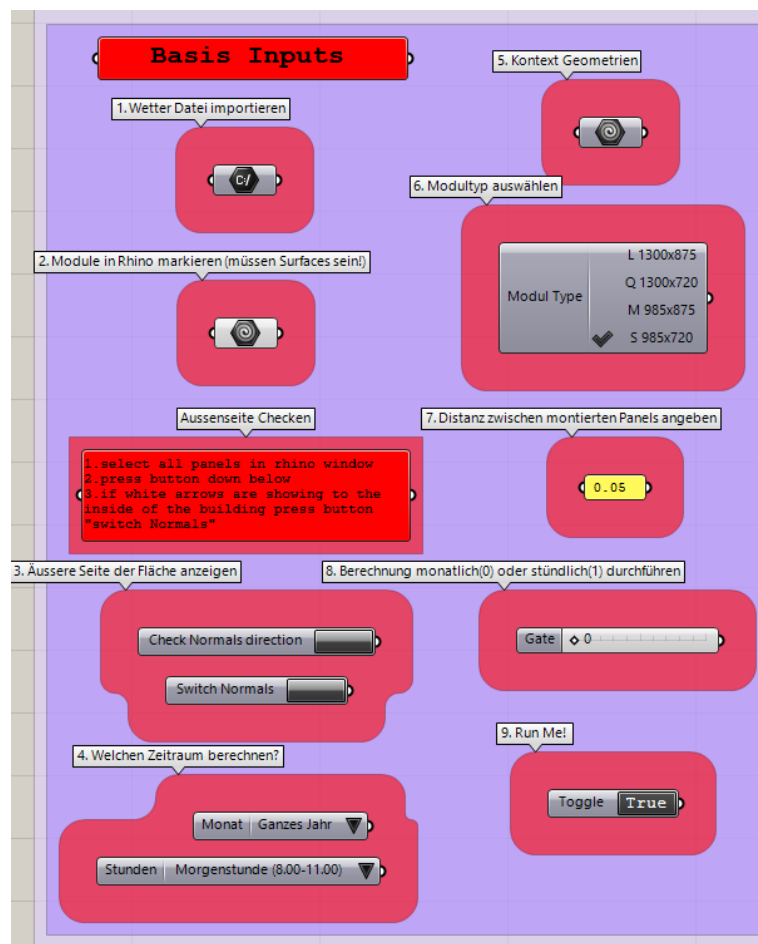


Abb. 22: User Interface Skript Level 2 und

Da in diesem Skript mit Panels gearbeitet wird, besteht die Möglichkeit, die String Gruppen, der Ertrag pro Modul und die Panelnummer zu visualisieren. Dazu können im Abschnitt «Visualisierungstools» die verschiedenen Regler gemäss Beschreibung im Skript verschoben werden. Dies hat kein Einfluss auf die Berechnung (siehe Abb. 23).

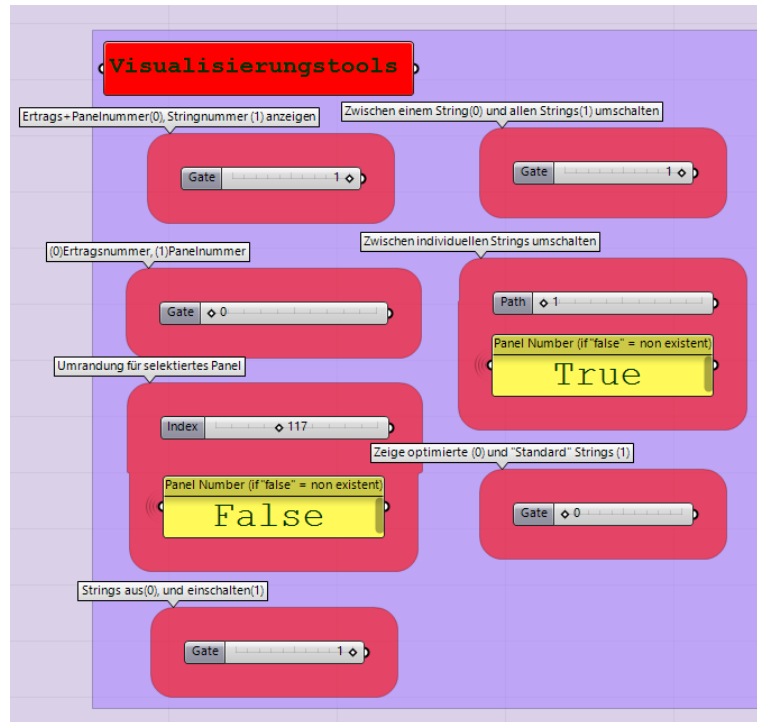


Abb. 23: Visualisierungstool

7.5.2 Plugins und Components

Für dieses Skript verwendete Plugins:

- LunchBox | ver: 2020.11.2.0 (Miller, 2023)
- MetaHopper | ver: 1.2.2 (andheum, 2019)
- Ladybug Tools | ver: 1.6.0 (Ladybug Tools, 2023)

Für dieses Skript verwendete Components:

(siehe Anhang A3)

7.5.3 Funktionsweise

Im Vergleich zum Skript Level 1, wird die Berechnung nicht auf Basis eines Grids, sondern aufgrund der importierten Photovoltaik Module getätigt. Die eigentliche Ertrags- sowie Wirtschaftlichkeitsberechnung ist jedoch grundsätzlich dieselbe wie in Skript Level 1 (siehe Kapitel 7.4.3).

Für die Visualisierung werden verschiedene Kombinationen von «Stream Filter» Nodes verwendet, welche die Möglichkeit bieten, zwischen verschiedenen Inputs für ein Skript auszuwählen. Damit die Ertragszahlen sowie die Panelnummern angezeigt werden können, wird das Node «Evaluate Surface» verwendet. Dadurch wird die Zahl immer am Mittelpunkt der jeweiligen Fläche «befestigt». Die Umrandung der gewählten Panels wird im ähnlichen Verfahren ausgeführt. Dabei wird jedoch nicht die Fläche erkannt, sondern die Aussenkanten der Flächen.

Um die Panels gemäss ihren String Gruppen einzufärben, wird der Input in diesem Beispielformat: [[30, 28, 19], [56, 73, 80]] an den eckigen Klammern, welche die einzelnen String Gruppen darstellen, geteilt und pro Gruppe eingefärbt. Dazu wird wie in der Beschriftung die Fläche jedes Panels evaluiert.

Für die Berechnung der Ertragswerte pro Gruppe wird der minimale Ertrag jeder Gruppe mit der Anzahl Modulen in der Gruppe multipliziert. Der Gesamtertrag ist schlussendlich die Summe aus den Ertragswerten aller String Gruppen.

7.5.4 Output

Auch im Level 2 werden die Einnahmen, Kosten, Rendite, Ertrag und Amortisationsdauer berechnet. Zusätzlich wird in einem separaten Output der Ertrag pro Gruppe ausgegeben. Mit diesen Werten und der Möglichkeit die Gruppen im Rhino Fenster zu visualisieren, kann zeitnah festgestellt werden, welche String Gruppen den höchsten und welche den tiefsten Ertrag pro Jahr einbringen (siehe Abb. 24).

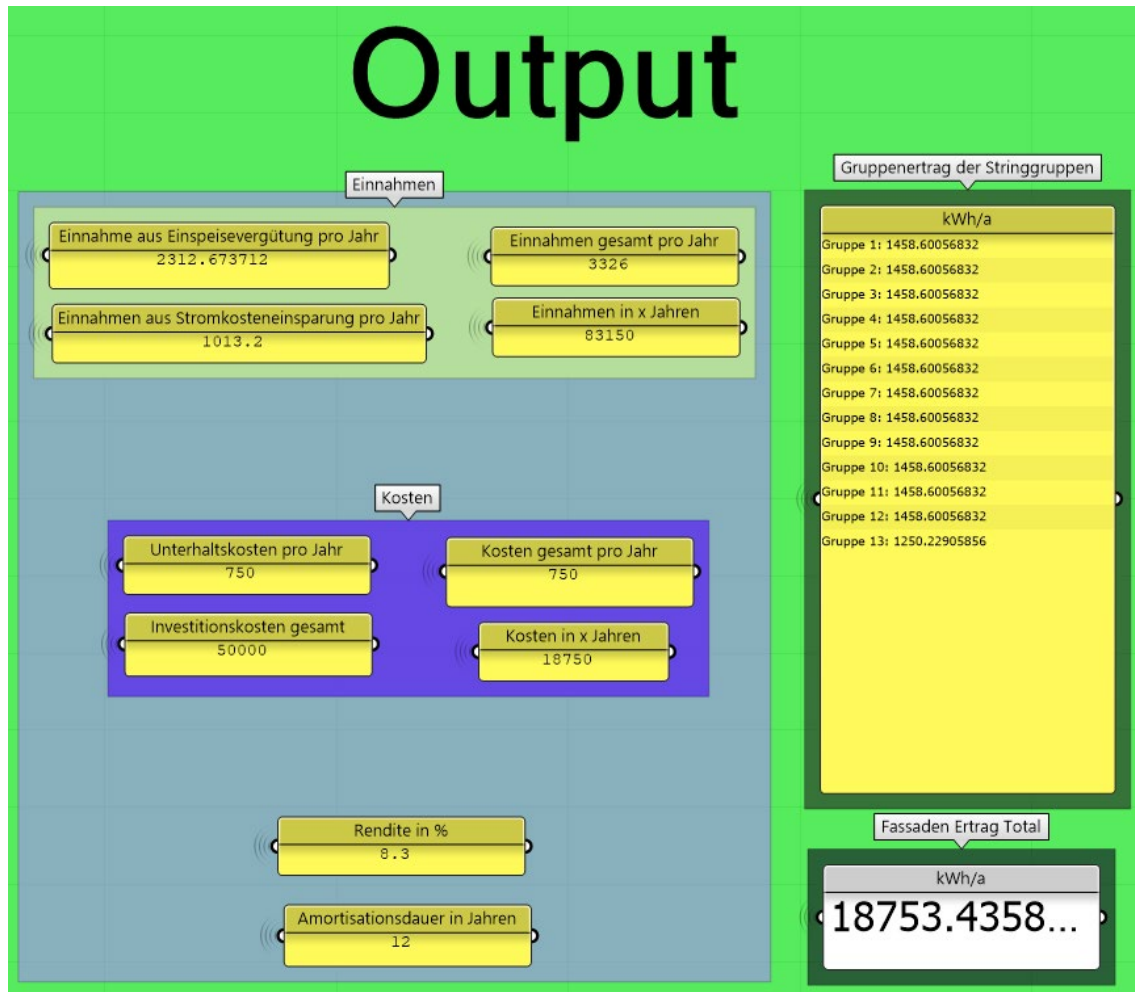


Abb. 24: Output Skript Level 2

7.6 Level 3 Optimierung String-Verschaltung

7.6.1 Benutzerinteraktion

Das Skript Level 3 ist auf die Phase der Projektierung ausgelegt und spricht die Zielgruppen PV-Planer und Gebäudetechnikplaner an. Die Grundfunktionen und die Nutzereingaben dieses Skripts sind dieselben wie beim Skript Level 2. (siehe Kapitel 7.5.1) Nebst der Ertragsberechnung und der Wirtschaftlichkeitsberechnung, bietet dieses Skript dem Nutzer die Möglichkeit, die String-Verschaltung der Panels zu optimieren. Mittels zwei verschiedenen Galapagos Algorithmen (siehe Abbildung 25) kann der Nutzer die String-Verschaltung entweder auf das gesamte Jahr oder auf einzelne vordefinierte Stunden optimieren (siehe Abbildung 26). Die Interaktion der Optimierung mit den Galapagos Algorithmen wird im «Tutorial-Video Script Level 3» detailliert erläutert (siehe Anhang A8).

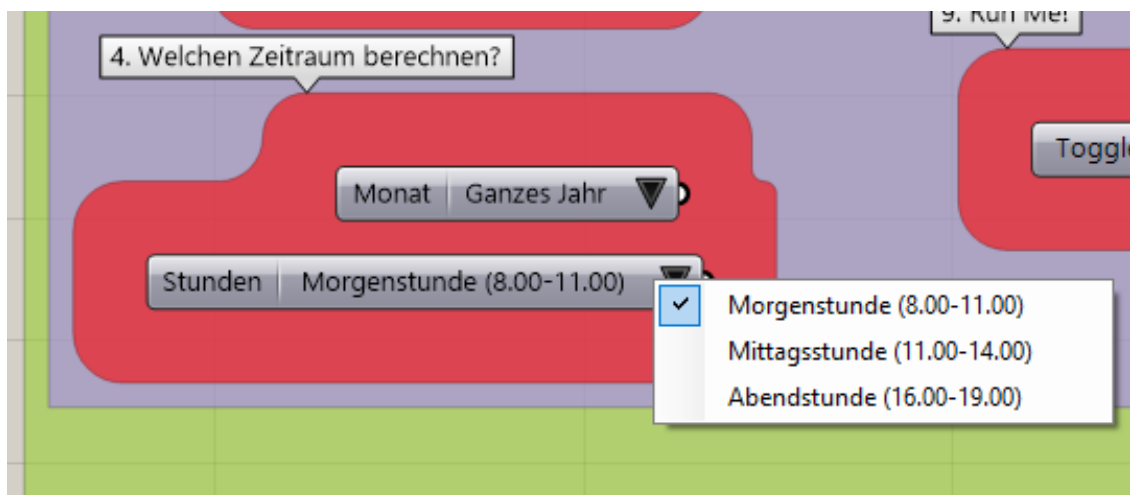


Abb. 26: Nutzerauswahl (Optimierung jährlich / vordefinierte Stunden)

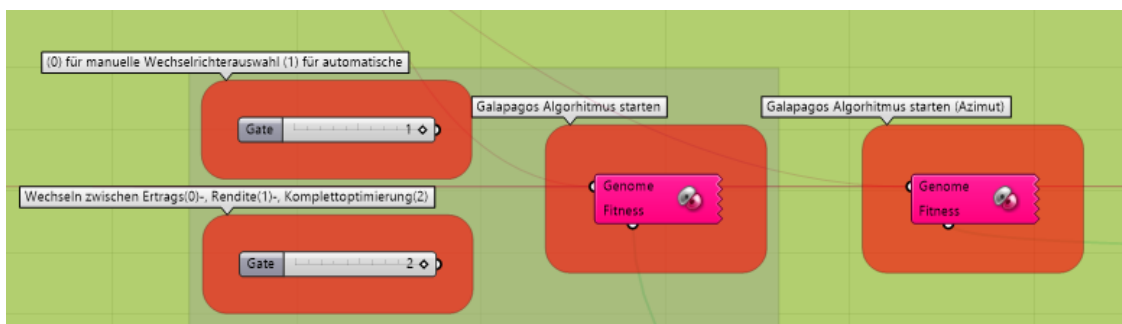


Abb. 25: Galapagos Nodes (Optimierung jährlich / vordefinierte Stunden)

7.6.2 Plugins und Components

Für dieses Skript verwendete Plugins:

- LunchBox | ver: 2020.11.2.0 (Miller, 2023)
- MetaHopper | ver: 1.2.2 (andheum, 2019)
- Ladybug Tools | ver: 1.6.0 (Ladybug Tools, 2023)

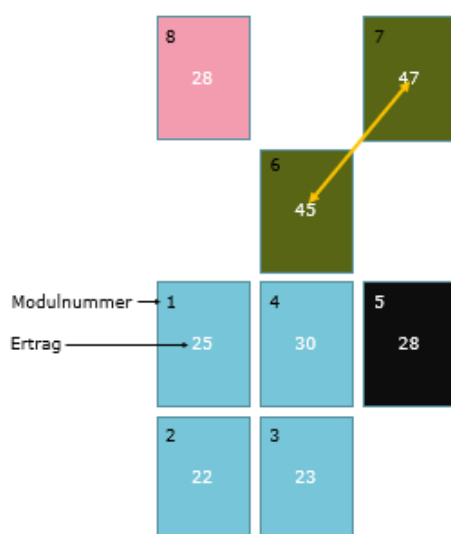
Für dieses Skript verwendete Components:

(siehe Anhang A4)

7.6.3 Algorithmus Optimierung String-Verschaltung

Die Logik des Algorithmus der String-Verschaltung wird in einem ersten Teil grob beschrieben, um ein Grundverständnis für die Funktion des Algorithmus zu erhalten. In einem zweiten Schritt wird die Funktion des Codes Schritt für Schritt erläutert.

Grundsätzlich hat der Algorithmus die Funktion, die bestmögliche Gruppierung von Modulen für die String-Verschaltung zu identifizieren. Dies wird erreicht, indem verschiedene Regeln definiert werden, welche den Ertrag der Strings beeinflussen und gleichzeitig die Installation der Panels nicht beeinträchtigen. Wichtige Faktoren sind dabei die Ertragsdifferenz zwischen den Modulen, die Distanz zwischen den Modulen und die Gruppengröße. Der Algorithmus bietet die Möglichkeit automatisch oder manuell verstellbar, die Ertragsdifferenz zwischen den Modulen zu verändern, damit nur diejenigen Module miteinander gruppiert werden, welche keine zu grosse Ertragsdifferenz aufweisen. Des Weiteren wird die Distanz zwischen den Modulen automatisch berechnet, damit nur Module gruppiert werden können, welche maximal den Abstand einer Diagonale zwischen dem Mittelpunkt der Module aufweisen. Weiter sorgt der Regelsatz der Gruppengröße dafür, dass die Gruppengröße (Anzahl Module in der Gruppe) die maximale Eingangsspannung der MPP Eingänge des Wechselrichter nicht übersteigt (siehe Kapitel 3.3, MPP-Tracker). Im letzten Schritt wird überprüft, ob die Gruppengröße grösser gleich zwei ist, um zu verhindern, dass String Gruppen mit weniger gleich zwei Modulen entstehen, da diese einen grossen Einfluss auf die Kosten haben. Sie brauchen einen separaten MPP Eingang, erbringen aber im Verhältnis nur



einen geringen Ertrag. In der Abbildung 27 ist ein Beispiel der Logik des Algorithmus ersichtlich.

Um in derselben Gruppe eingeteilt zu werden, müssen folgende Bedingungen erfüllt werden: Ertragsdifferenz ≤ 5 & Distanz \leq Diagonale zwischen Modulmittenpunkten / Gruppengröße ≤ 4 (maximale Eingangsspannung) / Gruppengröße ≤ 2 Module

Abb. 27: Bsp. Logik String-Verschaltung

String 1: Module 1, 2, 3, 4

Nicht verwendet: Modul 5 → Blindmodul / Modul 6, 7 → Blindmodule / Modul 8 → Blindmodul

Im folgenden Abschnitt wird der Algorithmus Schritt für Schritt erläutert. Zum einen wird auf die Inputs eingegangen und zum anderen auf die Funktion des Codes.

Im ersten Schritt wird ein Code generiert, welcher die Ertragsdifferenz und die Distanz zwischen allen Modulen berechnet. Der Output dieses Codes ist wie in der Abbildung 28 ersichtlich gegliedert. Der erste und zweite Wert des Outputs sind die Modulnummern, von denen die Ertragsdifferenz und Distanz berechnet werden. Der dritte Wert steht für die berechnete Distanz und der vierte Wert für die berechnete Ertragsdifferenz zwischen diesen Modulen. Weitere Inputs, welche für den Algorithmus benötigt werden sind: die maximale Eingangsspannung des Wechselrichters, die Leerlaufspannung pro Modul, alle verwendeten Module und die Ertragsdifferenz.

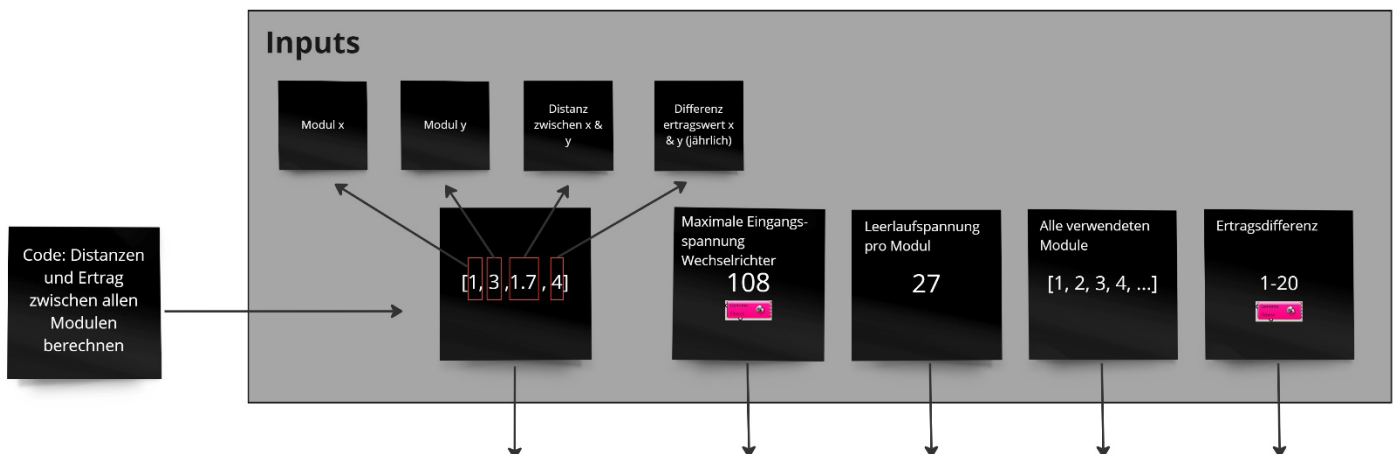


Abb. 28: Inputs Algorithmus

Der erste Schritt, welcher im Algorithmus durchlaufen wird, ist die Selektion der Module, welche miteinander in einer Gruppe gruppiert werden könnten. Der Code fügt die Module in eine Liste hinzu, wenn diese die Bedingung der maximalen Ertragsdifferenz und der maximalen Distanz erfüllen. Um die Gruppierung zu steuern und somit zu einem besseren Ergebnis zu gelangen, kann die Ertragsdifferenz manuell oder durch den Galapagos Algorithmus verändert werden. Der Output dieses Abschnittes ist eine Liste mit allen möglichen Modulkombinationen (siehe Abbildung 29).

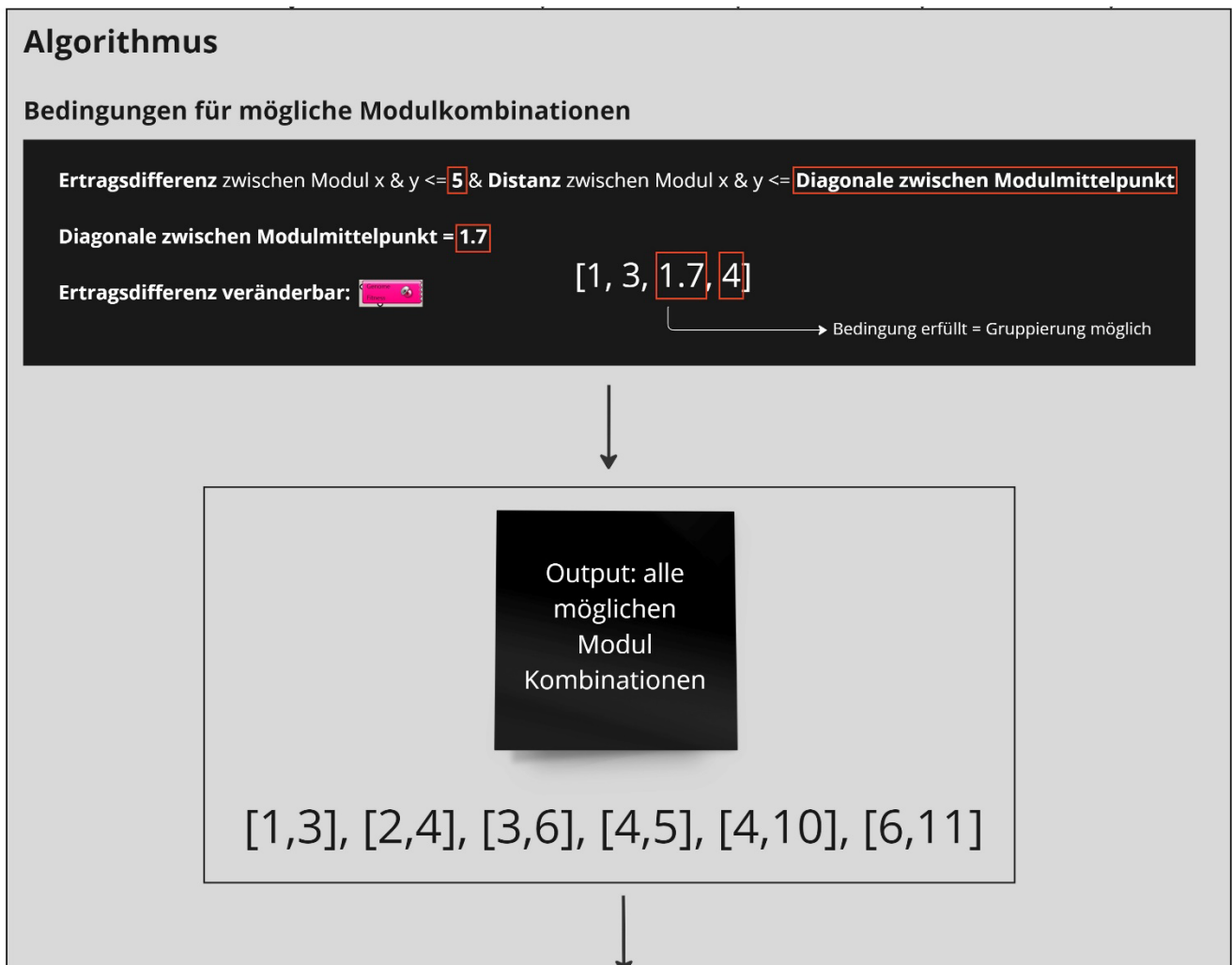


Abb. 29: Filterung mögliche Modulkombinationen

Die Liste der möglichen Kombinationen wird anschliessend mittels einer Schleife auf mögliche Gruppenbildungen geprüft. Jedes Gruppenpaar durchläuft die in Abbildung 30 aufgezeichnete Schleife. Es werden diejenigen Module in eine Gruppe übertragen, zwischen welchen mindestens eine Verbindung besteht und die Gruppengrösse nicht grösser ist als die Eingangsspannung des Wechselrichters geteilt durch die Leerlaufspannung der Module. Des Weiteren sind in der Abbildung 30 die finalen String Gruppen ersichtlich und alle verwendeten Modulnummern.

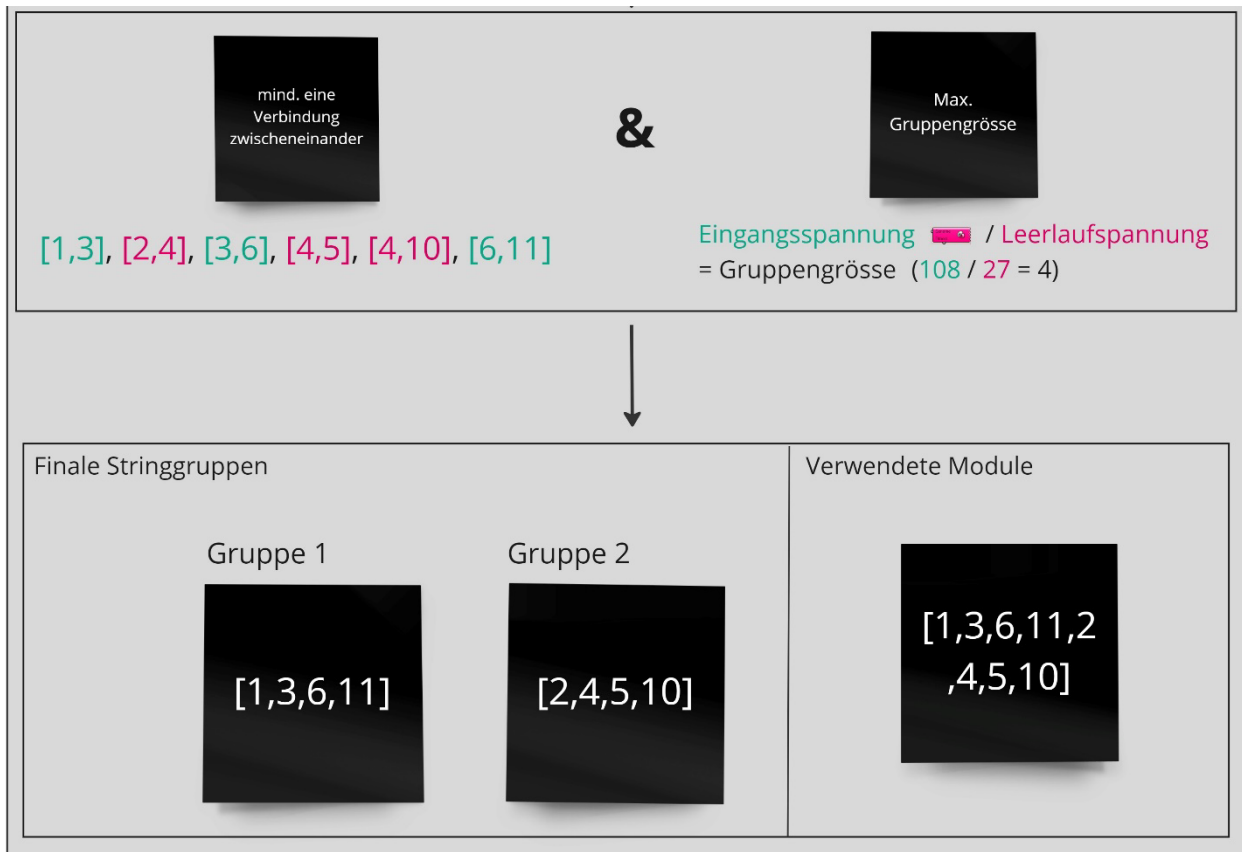


Abb. 30: Gruppierung Module / Finale String Gruppen

Die Liste der verwendeten Modulnummer dient dazu, dass einzelne Module nicht mehrmals verwendet werden. Die obere Schleife der Gruppierung überprüft bei jedem Durchlauf zusätzlich, ob die einzelnen Module der Modulpaare sich bereits in der Liste der verwendeten Module befindet. Falls dies der Fall ist, kann dieses Modul nicht ein zweites Mal verwendet werden und es ist sichergestellt, dass alle Module nur einmal verwendet werden.

Für die weitere Optimierung der String-Verschaltung wird der Algorithmus Galapagos verwendet. Dieser Algorithmus kann entweder Formen oder Verknüpfungen, mittels eines benutzerdefinierten Zieles optimieren. Um die optimale Verknüpfung herauszufiltern, benötigt Galapagos eine Reihe von veränderbaren Optionen, mit welchen er das benutzerdefinierte Endziel, den sogenannten Fitnessindex optimieren kann (Krymsky, 2014). In diesem Skript sind die veränderbaren Parameter, die Ertragsdifferenz, die Auswahl des Wechselrichters, welcher die Eingangsspannung definiert und ein Minusfaktor, welcher mit dem Eingangsspannungswert verknüpft ist. Der Minusfaktor wird zusätzlich benötigt, um die String Gruppengröße flexibel zu definieren, damit nicht automatisch der maximal Wert für die Optimierung verwendet wird. In der Abbildung 31 sind die veränderbaren Slider für die Ertragsdifferenz und für den Minusfaktor ersichtlich. In der Abbildung 32 ist die veränderbare Auswahl zwischen den Wechselrichter ersichtlich und in der Abbildung 33 die dazugehörigen Eingangsspannungswerte.

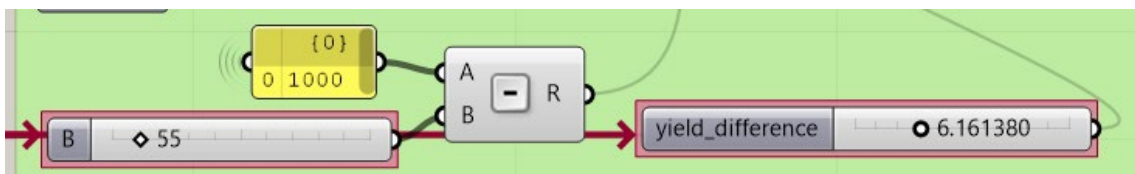


Abb. 31: Slider Ertragsdifferenz & Minus Faktor

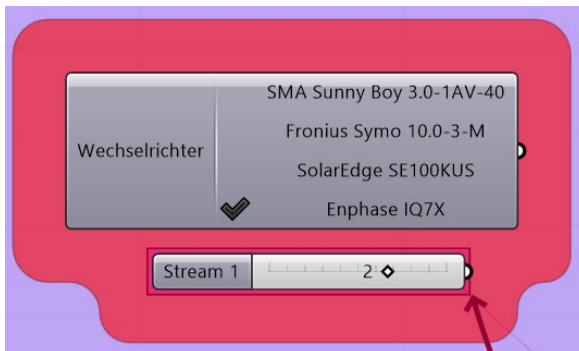


Abb. 33: Slider Wechselrichter

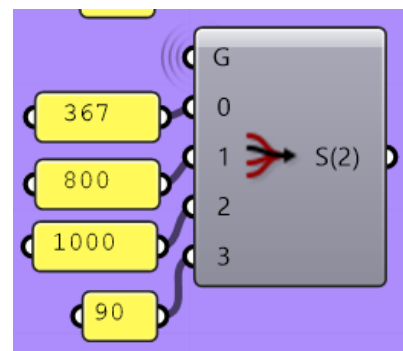


Abb. 32: Eingangsspannungswerte der Wechselrichter

Als Ziel der Optimierung werden drei verschiedene Fitnessfaktoren definiert, welche vom Benutzer manuell eingestellt werden können. Der erste Fitnessfaktor bezieht sich auf den Ertrag der PV-Anlage, der zweite Fitnessfaktor auf die Rendite und der dritte ist eine Kombination aus dem Ertrag und der Rendite (siehe Anhang A14). Der vollständige Code des Algorithmus ist im Anhang A13 ersichtlich.

7.6.4 Stündliche Ertragsberechnung

Für die Berechnung des Ertrags Outputs hat der Nutzer die Möglichkeit zwischen einer monatlichen oder einer stündlichen Berechnung auszuwählen. Um den Ertrag möglichst genau zu eruieren, ist die stündliche Berechnung am besten geeignet (siehe Kapitel 6.1). Bei beiden Berechnungen werden separate Sonnenanalysen erstellt, um die jeweiligen Werte zu erhalten. Des Weiteren wird ein Code erstellt für beide Berechnungen, welcher pro Stunde oder pro Monat die jeweiligen Werte der einzelnen Strings berechnet. In der Abbildung 34 ist ein Beispiel der Funktion des Codes ersichtlich. In der stündlichen Berechnung werden normalerweise 8760 Stunden (Stunden im Jahr) pro String berechnet. In diesem Beispiel werden zur Vereinfachung drei Stunden mit zwei Strings à drei Modulen berechnet. Als erster Schritt wird der Ertrag für jedes Modul pro Stunde berechnet. Damit die Ertrags-Werte weiterverwendet werden können, werden mittels des Codes «Umwandlung Format» den Ertragswerten Modulnummern zugeteilt. Der Code «Austausch Werte / String Gruppen Ertrag» erhält als Input die String Gruppen des String Verschaltung Algorithmus und die Ertragswerte mit den zugehörigen Modulnummern. Im Code werden die einzelnen Ertragswerte pro Zeitabschnitt den Strings zugewiesen. Als letzter Schritt wird bei jedem String der kleinste Ertragswert eines Moduls mal die Gruppenlänge multipliziert. Der Ablauf der monatlichen Berechnung ist identisch. Die Berechnung wird einzig statt mit 8760 Ertragswerten nur mit 12 Ertragswerten (Monate im Jahr) berechnet (siehe Anhang A24).

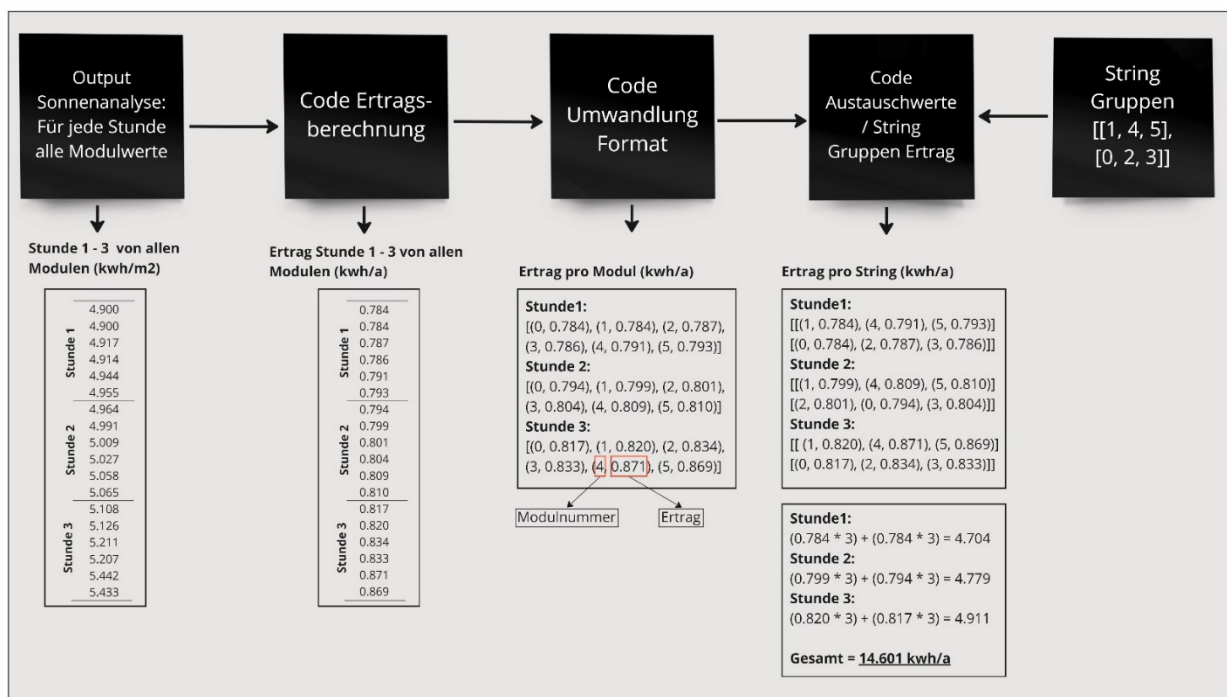


Abb. 34: Stündliche Ertragsberechnung

7.6.5 Wirtschaftlichkeitsberechnung

Die Wirtschaftlichkeitsrechnung wird auf Grundlage des Berechnungsdokumentes der RESiQ (Kutschera, 2014) in Python erstellt. Zuerst werden die benötigten Parameter definiert. Im Skript verwendete Parameter sind:

- Produktion der Fassade / Jahr (kWh/a) → durch Skript berechnet
- Kosten pro MLPE
- Eigenverbrauch pro Jahr in % der Produktion
- Investitionskosten PV-Module
- Wechselrichtertyp (Kosten, Anzahl Eingangsstrings, max. Spannung)
- Instandhaltungskosten in % der Investitionskosten
- Förderbeitrag Kanton in % der Investitionskosten
- Strompreis Durchschnitt Hochtarif / Niedertarif (Fr./kWh)
- Einspeisevergütung Durchschnitt Hochtarif / Niedertarif (Fr./kWh)
- Anzahl zu berechnender Jahre
- Zyklische Investition durch Erneuerung der Wechselrichter / MLPE
- Lebensdauer Wechselrichter / MLPE → 15 Jahre

Es wurde im Rahmen dieser Arbeit aufgrund der zur Verfügung stehender Dokumente angenommen, dass diese Parameter den grössten Einfluss auf das Endresultat haben. Parameter wie Speicherlösungen, Reinigung, Installation, Steuervorteile, Strompreisentwicklung, Wechselrichtereffizienz (ebd., S.3) und dergleichen wurden in der Arbeit nicht berücksichtigt. Diese können je nach Situation eine Differenz zum erhaltenen Resultat zur Folge haben.

Ziel der Wirtschaftlichkeitsrechnung ist es, eine Aussage über die wirtschaftliche Leistung der erarbeiteten PV-Systeme treffen zu können. Daher sind im Output die Teilkosten- und Einnahmen aufgelistet, aus welchen die Kosten- und Einnahmen pro Jahr und pro Anzahl angegebenen Jahre zusammengesetzt sind.

Als Ergebnis wird ein eindeutiges Resultat als Rendite und Amortisationsdauer definiert, welches die wirtschaftliche Sinnhaftigkeit der Anlage beschreibt.

In diesem Projekt wird grundsätzlich mit dem statischen Berechnungsmodell ohne Berücksichtigung des Zeitwerts des Geldes (Kutschera, 2014, S. 5) gearbeitet. Die zyklische Investition findet immer genau nach Ablauf der Lebensdauer der elektronischen Geräte statt. Das bedeutet, dass die Investition für Wechselrichter und MLPE in der Berechnung für 30 Jahre schon für die nächsten 15 Jahre bis ins Jahr 45 mitgerechnet wurde.

Die **Einnahmen** werden mit folgender Formel berechnet (ebd., S.18):

$$\mathbf{Einnahmen_Jahr} = \text{Einspeisevergütung} \times (\text{Produktion} - \text{Eigenverbrauch}) + (\text{Eigenverbrauch} * \text{Strompreis})$$

$$\mathbf{Einnahmen_X_Jahre} = \text{Einnahmen_Jahr} * \text{Anzahl_Jahre}$$

Die **Kosten** werden mit folgender Formel berechnet:

$$\mathbf{Kosten_Jahr} = (\text{Investitionskosten} + \text{Kosten_Wechselrichter}) * \text{Instandhaltungskosten_prozent}$$

$$\mathbf{Kosten_X_Jahre} = \text{Kosten_Jahr} * \text{Anzahl_Jahre} + \text{Kosten_Wechselrichter} + (\text{Kosten_Wechselrichter} * (\text{Anzahl_Jahre}/15)) + \text{Kosten_Optimizer} + (\text{Kosten_Optimizer} * (\text{Anzahl_Jahre}/15))$$

Die **Rendite** wird mit folgender Formel berechnet:

$$\mathbf{Rendite} = (\text{Einnahmen_Jahr} * 100) / (\text{Investitionskosten_PV} + \text{Kosten_Wechselrichter} + \text{Kosten_Optimizer})$$

Die **Amortisationsdauer** wird mit folgender Formel berechnet:

$$\mathbf{Amortisationsdauer} = (\text{Investitionskosten} - \text{Förderbeitrag} + \text{Kosten_Wechselrichter} + \text{Kosten_Optimizer}) / \text{Einnahmen_Jahr}$$

(Source code siehe Anhang A20 & A21)

7.6.6 Output

Neben dem Gesamtertrag, dem Gruppenertrag und der Wirtschaftlichkeitsberechnung, wird im Level 3 ebenfalls ausgegeben, welche Module vom Skript als «Blindmodule» klassifiziert wurden (siehe Kapitel 7.6.3). Ausserdem wird ein Vergleich zwischen dem Output mit und ohne MLPE, sowie dem Ertrag aus einer «Standard»-Planungsweise (siehe Kapitel 5) und der optimierten String-Verschaltung ausgegeben (siehe Abb. 35, 36 & 37).

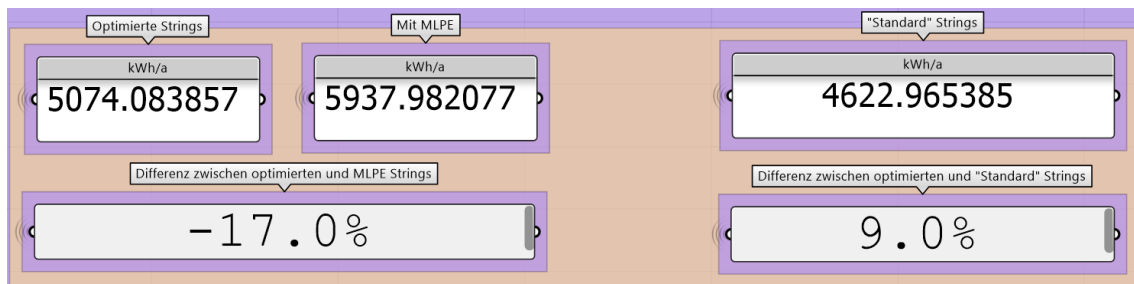


Abb. 35: Output Skript Level 3, Vergleiche

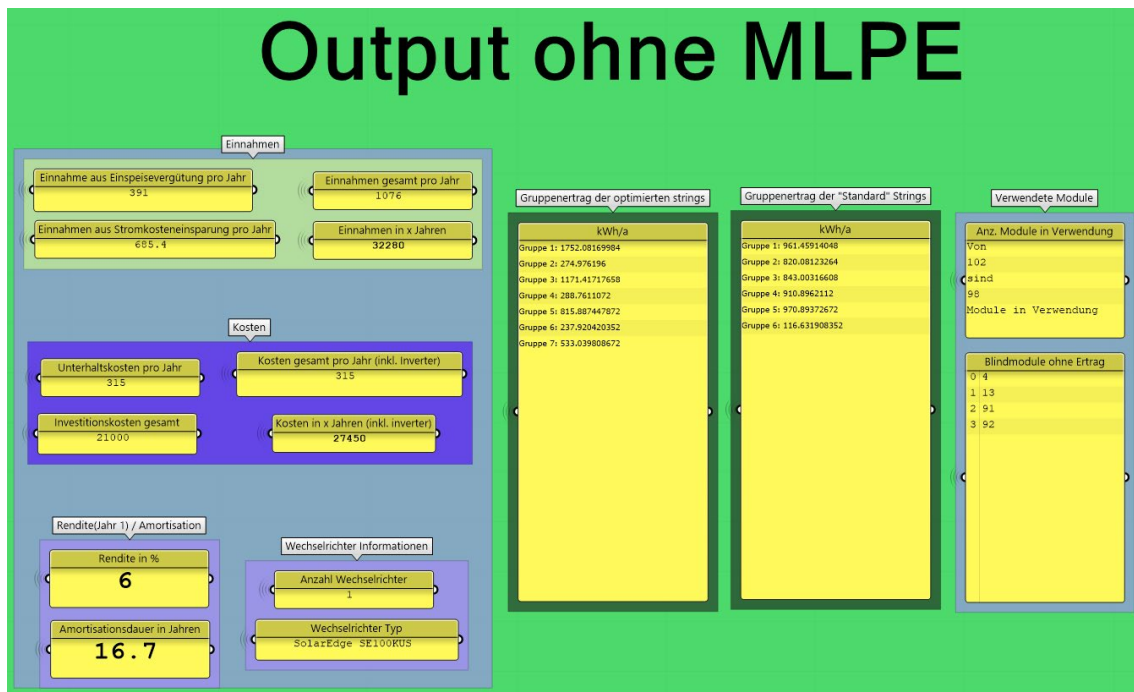


Abb. 36: Output Skript Level 3, ohne MLPE

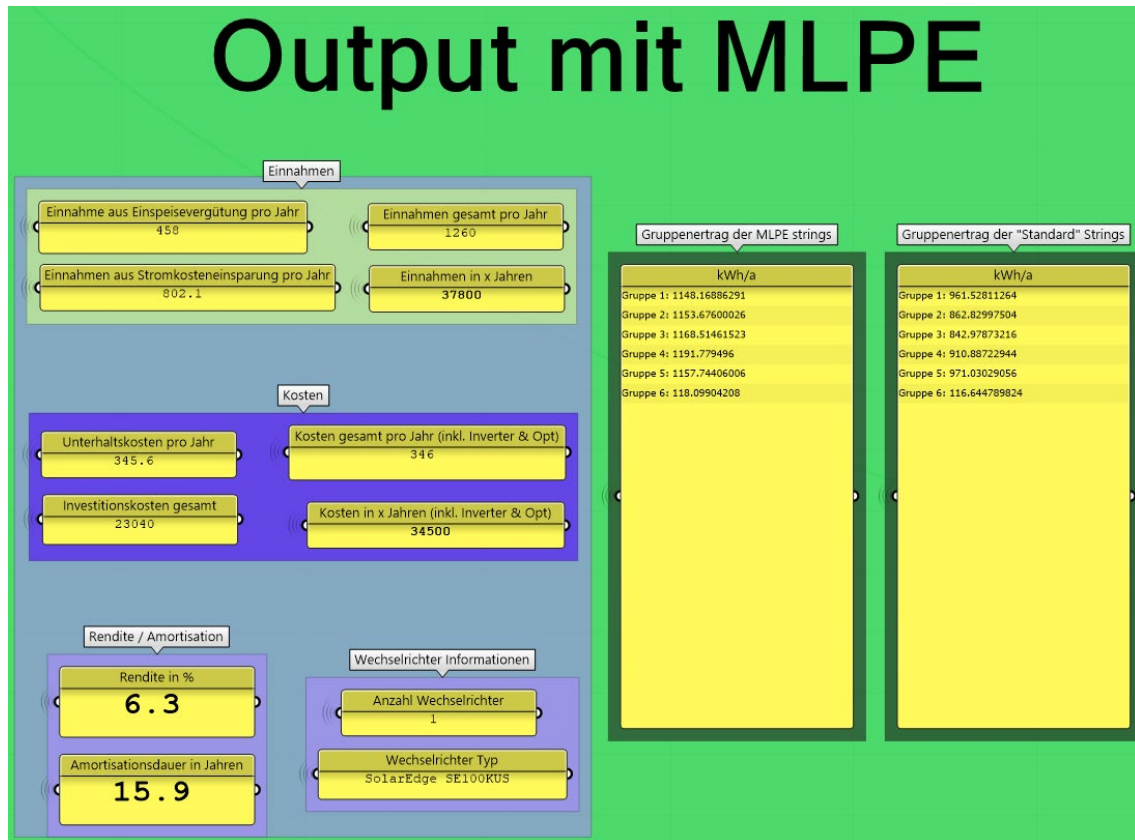


Abb. 37: Output Skript Level 3, mit MLPE

7.7 Level 4 Automatische PV Modul Generierung und Optimierung String Verschaltung

7.7.1 Benutzerinteraktion

Die Benutzereingabe verhält sich gleich wie in Skript Level 3 (siehe Kapitel 7.6). Da die Panels im Level 4 automatisch auf der definierten Fassadenfläche generiert werden, kann zusätzlich die Panelausrichtung (horizontal / vertikal) definiert werden. Nachdem ausgewählt wurde, ob die Berechnung mit monatlichen oder stündlichen Einstrahlungswerten auf die Panels durchgeführt werden soll, kann das Skript mit einem Klick auf den «Run me» - Button gestartet werden. Nach diesem Schritt sollte im Output Panel der Ertrag sowie alle Wirtschaftlichkeitsparameter ausgegeben werden. Wie in Level 3 kann auch in Level 4 die String-Verschaltung noch weiter optimiert werden. Dafür kann die monatliche oder die stündliche Berechnungsweise angewandt werden. Getestet ist die monatliche Variante, da die Dauer für eine stündliche Berechnung stark von der Rechenleistung des Computers abhängt. Grundsätzlich funktioniert die Methode jedoch für beide Varianten. Sobald der Galapagos Algorithmus einen optimierten Wert ausgibt, kann er gestoppt und dieser Wert übernommen werden. Nun sollte im Output der vom Algorithmus definierte Wechselrichtertyp, dessen eingesetzte Anzahl sowie alle Ertrags- und Wirtschaftlichkeitswerte ausgegeben werden.

7.7.2 Plugins und Components

Für dieses Skript verwendete Plugins:

- LunchBox | ver: 2020.11.2.0 (Miller, 2023)
- MetaHopper | ver: 1.2.2 (andheum, 2019)
- Ladybug Tools | ver: 1.6.0 (Ladybug Tools, 2023)

Für dieses Skript verwendete Components:

(siehe Anhang A5 & A14)

7.7.3 Automatische Modulgenerierung

Was das Skript Level 4 von den anderen Levels unterscheidet, ist die automatische Modulplatzierung auf einer vordefinierten Fassade. Dieser erste Teilbereich des Skripts kann in sechs Untergruppen unterteilt werden. In einem ersten Schritt wird ein Raster über die gesamte Fassade erstellt. Die Rastergröße hängt von der Auswahl des PV-Modultypen ab. Der Rasterstartpunkt muss im aktuellen Stand des Projektes manuell in den Inputs angegeben werden (siehe Abb. 38).

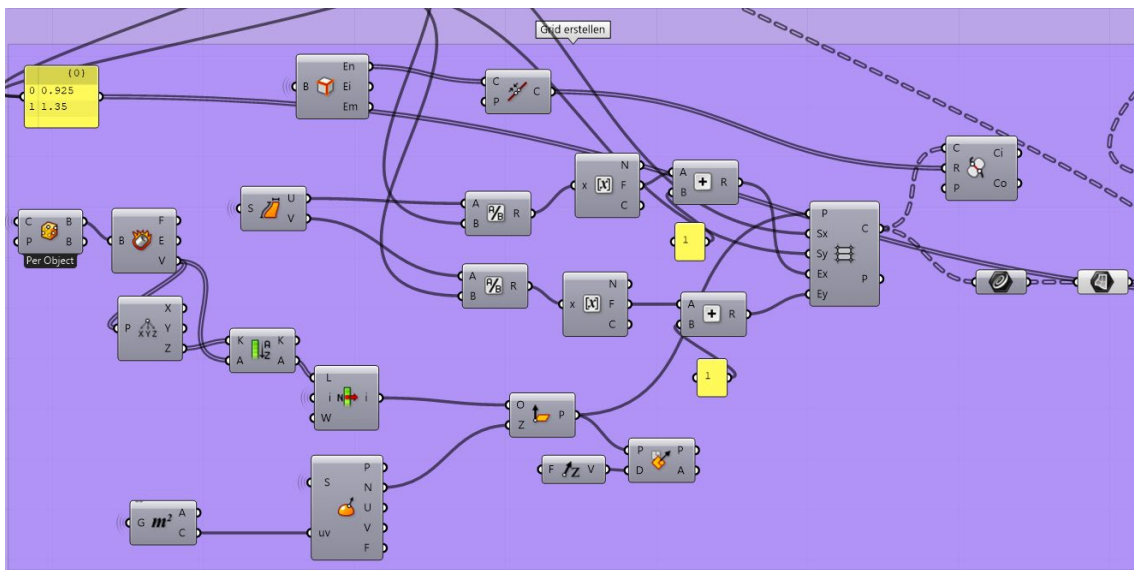


Abb. 38: Raster auf Fassade erstellen

Die Module einer PV-Anlage haben bei der Montage immer einen gewissen Abstand zwischen einander. Dieser wird im zweiten Teilbereich erstellt. Dazu werden aus dem Raster ohne Abstände die Mittelpunkte der Rasterflächen entnommen und als Mittelpunkt für neue Modulflächen verwendet. Der Abstand zwischen den Modulen wird ebenfalls manuell in den Inputs angegeben (siehe Abb. 39).

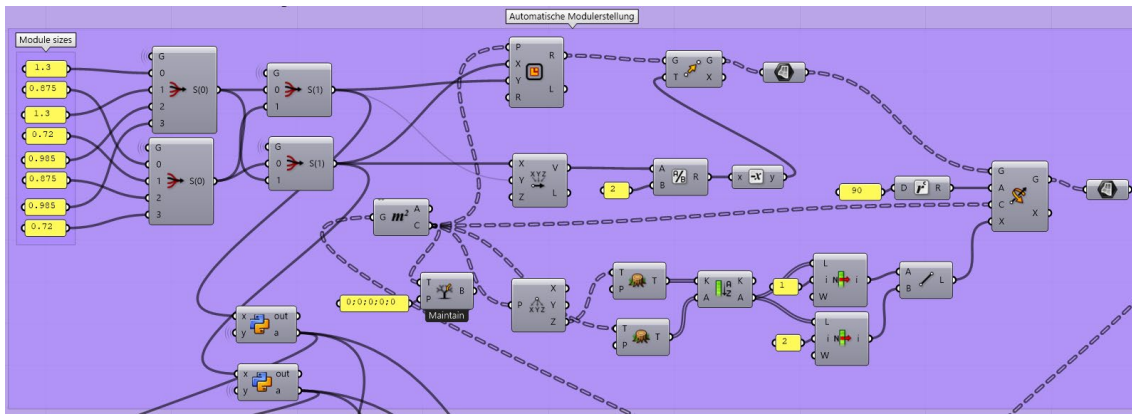


Abb. 39: Neue Module mit Abstand erstellen

Damit alle Module, welche sich über einer Fenster- oder Türöffnung befinden, entfernt werden, müssen diese Öffnungen zuerst auf der Fläche gefunden werden. Im Teilbereich 3 werden aus der definierten Fassadenfläche alle Kanten extrahiert. Alle zusammenhängenden Kanten, welche kürzer als die längste Kante sind, definieren grundsätzlich die Öffnungen in einer Fläche. Dieser Ansatz ist jedoch relativ ungenau. Sobald die Öffnung in der Fläche eine spezielle Form besitzt und dadurch dessen Aussenkante grösser, als die der Umrandung der Fassaden ist, funktioniert die Erkennung nicht mehr korrekt. Die Wahrscheinlichkeit des Eintretens eines solchen Falles ist jedoch relativ gering und deshalb für diese Arbeit vernachlässigbar (siehe Abb. 40).

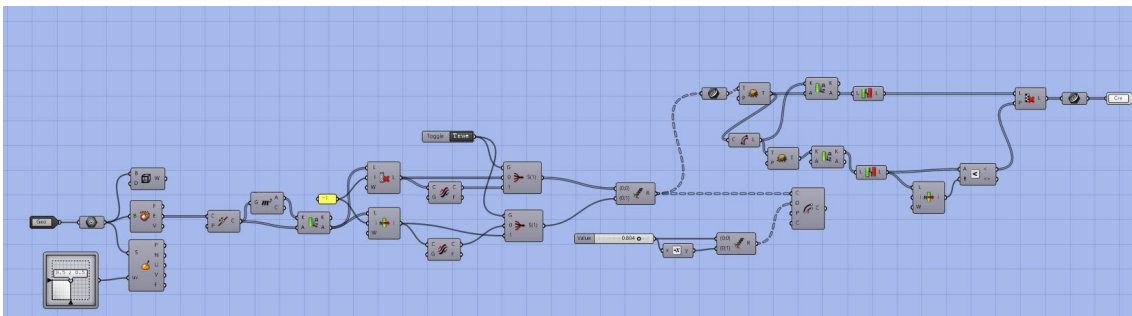


Abb. 40: Erkennung von Öffnungen

Sobald die Kanten der Öffnungen feststehen, werden die Module von diesen Kanten verschnitten. Alle verschnittenen Module innerhalb der Öffnungen werden dabei entfernt. Die ganzen Module innerhalb der Öffnungen werden in einem nächsten Schritt zusätzlich entfernt.

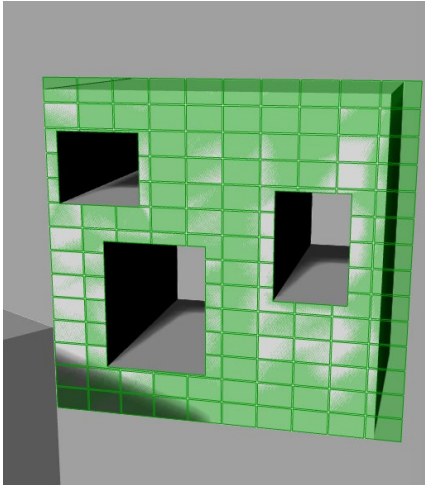


Abb. 41: Module innerhalb Öffnungen entfernt

Auch die Aussenkante der Fassadenfläche wird als Schnittkante verwendet und verschneidet alle Module, welche sie passieren (siehe Abb. 41).

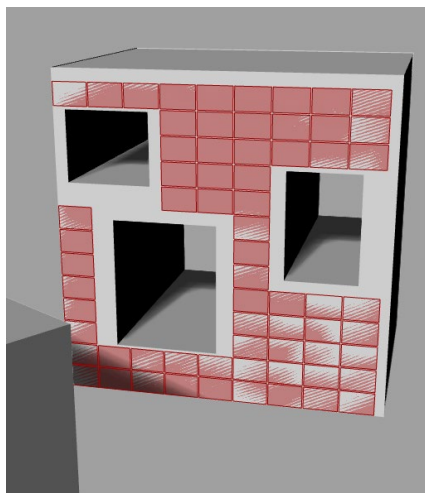


Abb. 42: Verschnittene Module entfernt

Module, welche nach dem Verschneiden nicht mehr der Ausgangsgrösse entsprechen, werden im nächsten Schritt entfernt. Dadurch entsteht die endgültige Modulplatzierung (siehe Abb. 42).

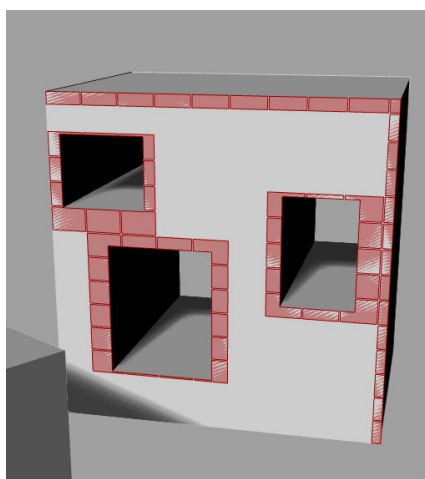


Abb. 43: Verschnittene Module = Blindmodule

Die verschnittenen Module können über den Slider ersichtlich gemacht werden (siehe Abb. 43). Sie werden nun als «Blindmodule» bezeichnet, da es wirtschaftlich keinen Sinn macht, diese als Ertragsproduzierende Module zu konstruieren. Ausserdem wird im Output angegeben, welche Fläche die verschnittenen Module, sowie die generierten Module haben. Dies sollte die Bestellung dieser Blindmodule vereinfachen.

Sollten die Modultypen nicht vorgegeben sein, kann nach der ersten Analyse mithilfe des Galapagos Algorithmus die Ausnutzung der Fassadenfläche optimiert werden. Dazu wird der Solver auf den maximalen Flächenwert der erstellten Panels gestartet. Dies sollte wie vorgängig beschrieben im Output über die Flächenwerte ersichtlich sein.

Damit die Einstrahlungsberechnung anschliessend durchgeführt werden kann, müssen die «Normals», also die rechten Seiten oder Aussenseiten der Modulflächen, evaluiert werden. Dazu werden alle Module an ihren spezifischen Aussenseiten extrudiert. Daraus entstehen verschiedene Quader mit spezifischen Mittelpunkten. Sollten sich die Mittelpunkte dieser Quader innerhalb der Gebäudegeometrie befinden, werden deren Aussenseiten gewechselt. Somit wird sichergestellt, dass die Sonnenanalyse anschliessend immer korrekt durchgeführt werden kann (siehe Abb. 44).

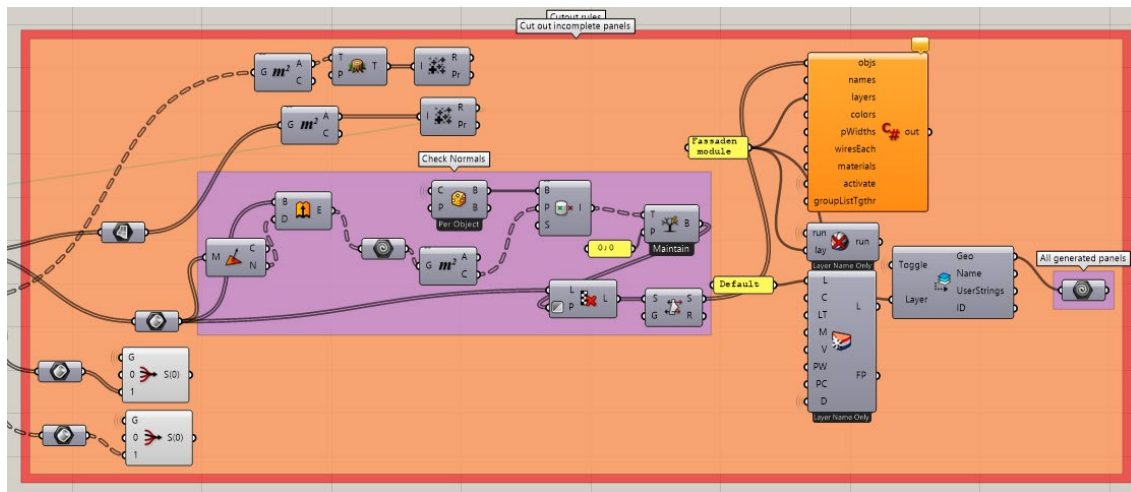


Abb. 44: Offsetkurve Aussenkante & Normals direction

Die generierte PV-Modulgeometrie kann nur in gebackenem Zustand der Sonnenanalyse übergeben werden. Ausserdem sollte die Möglichkeit bestehen, die generierten Module vor der Ertragsberechnung zu löschen und mit einer anderen Variante weiterzuarbeiten (siehe Abb. 45).

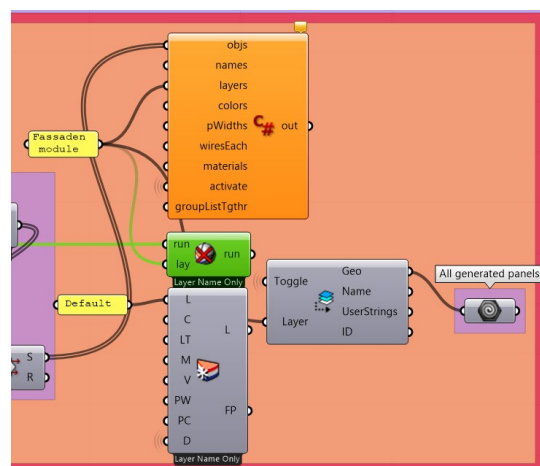


Abb. 45: "Backen" auf spezifischen Layer

Deshalb werden die Module im letzten Teilbereich 6 auf den Layer «Fassadenmodule» gebacken (siehe Anhang A14). Mit dem Node «Delete on Layer» können auf Knopfdruck alle Elemente auf dem Layer «Fassadenmodule» wieder gelöscht werden. Für die weitere Berechnung werden alle Elemente auf demselben Layer an einen Geometrie-Node übergeben.

7.7.4 Output

Der Output für Level 4 ist gleich gegliedert wie der Output für Level 3 und ist in den Abbildungen 46, 47 und 48 ersichtlich (siehe Kapitel 7.6).

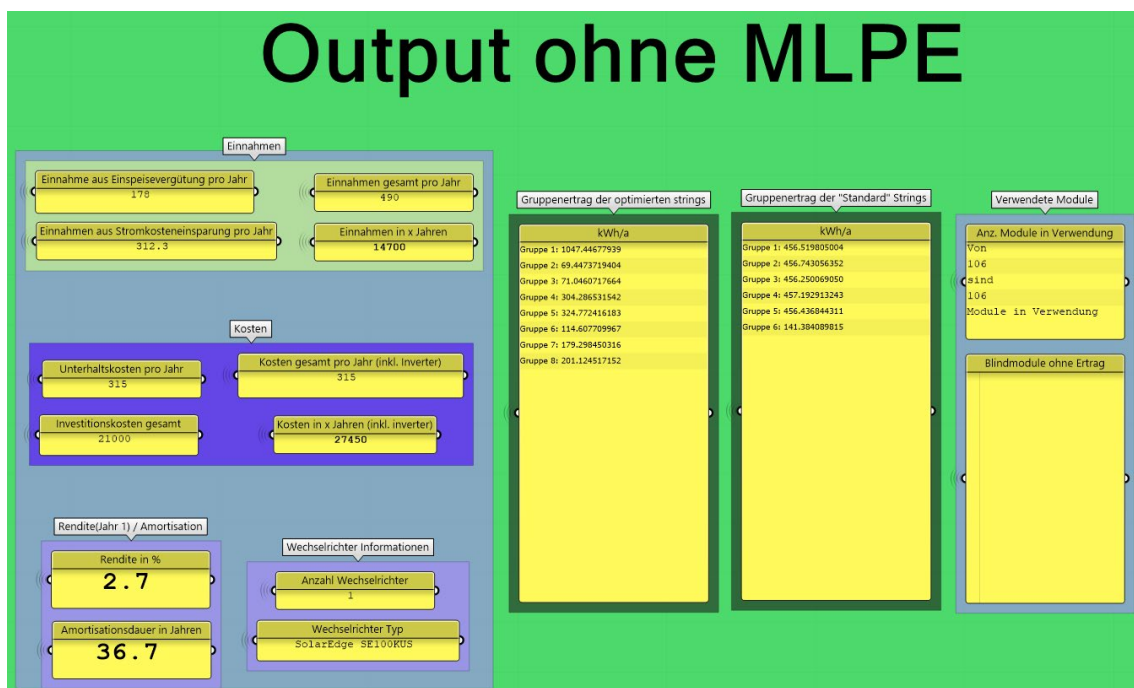


Abb. 46: Output Skript Level 4, ohne MLPE

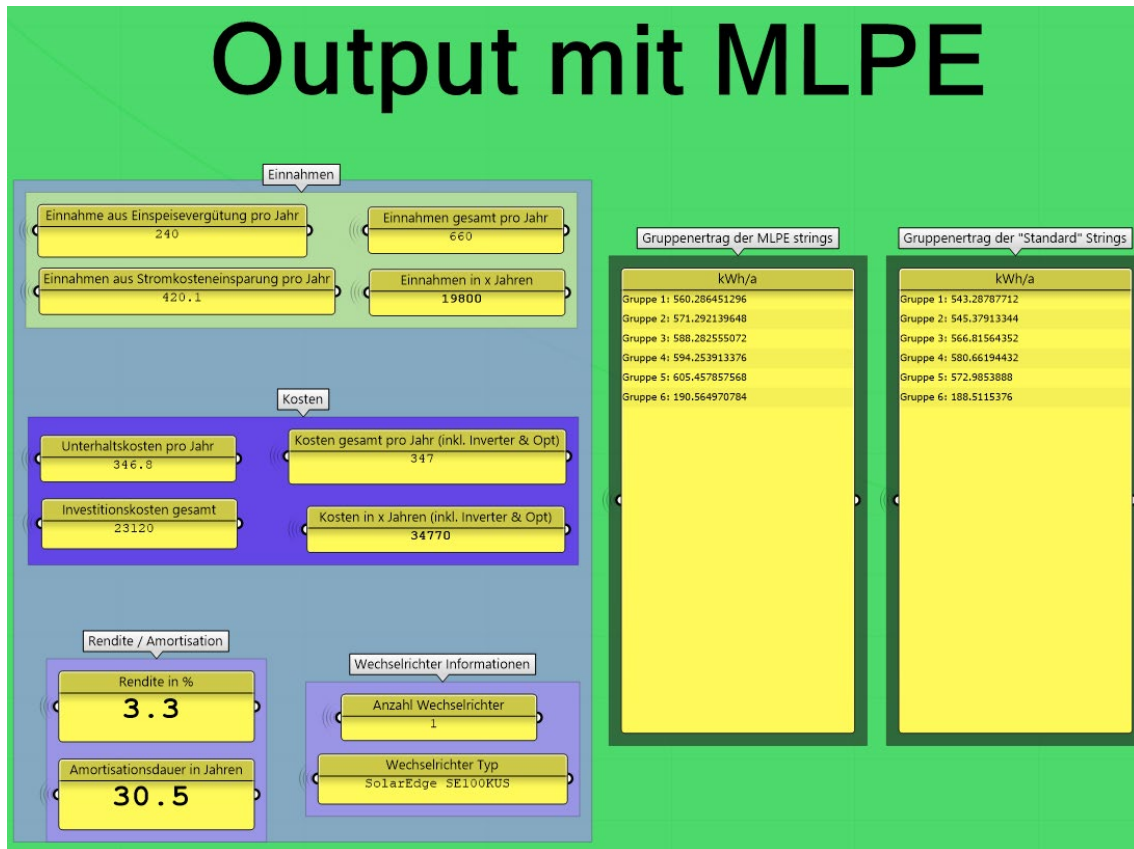


Abb. 47: Output Skript Level 4, mit MLPE

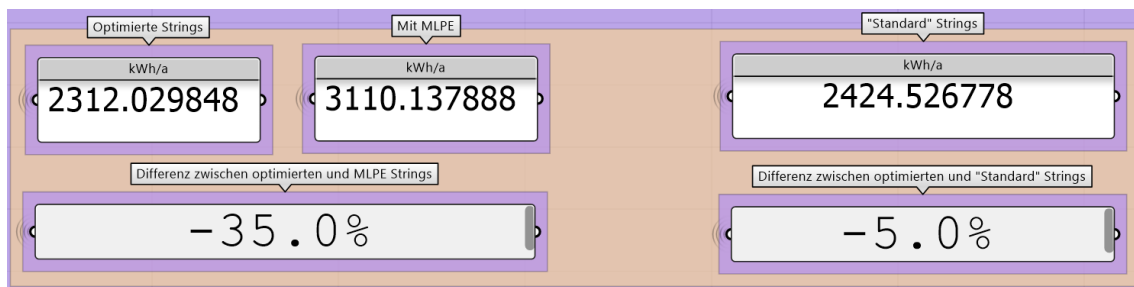


Abb. 48: Output Skript Level 4, Vergleiche

7.8 Forschung und mögliches weiteres Vorgehen

7.8.1 String-Verschaltung Optimierung mit Azimut Werten

Im Script Level 3 und 4 wird die String Optimierung auf das gesamte Jahr optimiert und die Ertragsberechnung mittels stündlichen Berechnungen angestellt. Gemäss dem Level of Detail, welches im Kapitel 7.3 aufgelistet ist, ist dies die Stufe 3. Da jeder Zeitpunkt einen spezifischen Sonnenstand aufweist und die Einstrahlungsintensität und der Schattenwurf sich dadurch verändern, gibt es Jahreszeiten und Zeitfenster in welchen viel mehr kWh produziert wird als in anderen (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 13). Mittels einem Forschungs-Script wird versucht, die String-Verschaltungsoptimierung auf sechs verschiedene einstrahlungsintensive Zeitfenster zu optimieren. Dabei wird darauf geachtet, dass der Azimut Winkel bei diesen verschiedenen Zeitfenster möglichst unterschiedlich ist. Dieses String-Verschaltungsvorgehen widerspiegelt die Stufe 5 des Level of Detail und birgt die Möglichkeit einer weiteren Ertragssteigerung.

Das Vorgehen des Scripts ist wie folgt. Es werden folgende sechs differenzierte Zeitfenster, 1. Juli 6-7 Uhr, 1. März 8-9 Uhr, 1. August 13-14 Uhr, 1. Februar 15-16 Uhr, 21. Juni 13-14 Uhr und 21. Dezember 13-14 Uhr, mit der Logik der String-Verschaltungsoptimierung vom Script Level 3 berechnet, welche im Kapitel 7.6.3 erläutert ist. Die Optimierung, mit dem Galapagos Algorithmus, ist jedoch lediglich auf den Ertrag ausgelegt und beinhaltet die Veränderung der Ertragsdifferenz und des Minusfaktors der Eingangsspannung eines vordefinierten Wechselrichters (siehe Abb. 49).

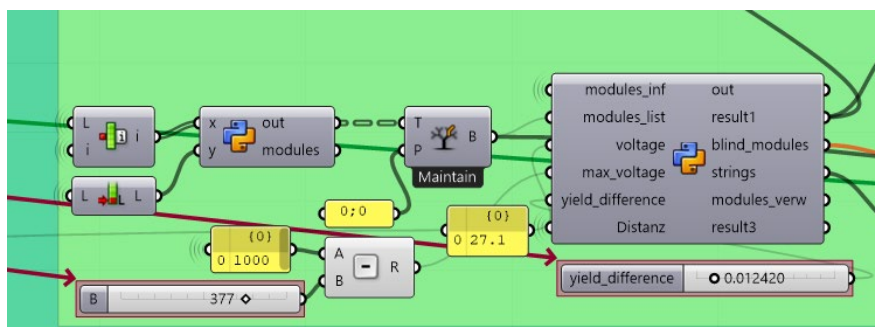


Abb. 49: Veränderbare Parameter Azimut Optimierung

Dies aus dem Grund, da diese sechs verschiedenen Resultate anschliessend miteinander kombiniert werden müssen, damit in einem späten Zeitpunkt das beste Resultat eruiert werden kann. Als Resultat aus diesen sechs verschiedenen Optimierungen werden sechs spezifische String Gruppen Kombinationen ausgegeben, welche in der Abbildung 50 mit den Klammern '['[[...]]]' unterteilt sind. Die einzelnen String Gruppierungen sind mit folgenden Klammern '['..]' unterteilt.

```

([([29, 15, 18, 33, 25, 35, 32, 22, 7, 24, 42, 17, 0, 2, 3, 9, 36, 31, 10, 49, 53], [30, 58, 65, 23, 34, 47, 1, 76, 16, 51, 8, 40, 69], [19, 20, 13, 12, 28, 5, 4, 14, 6, 27, 26, 11, 21], [37, 39, 38, 86], [48, 52, 66, 41, 70, 59, 77], [68, 54, 72, 61, 50, 79, 43], [46, 45, 44, 87], [55, 63, 84, 62, 64, 57, 85, 56], [78, 71, 60, 67], [73, 81, 75, 82, 80, 74, 83]])], [[([5, 4, 7, 6, 1, 0, 3, 2, 14, 9, 8], [19, 20, 27, 11, 28, 10], [13, 12, 21], [29, 15, 23, 33, 35, 32, 22, 34, 24, 44, 16, 36, 31, 30], [18, 26, 25, 17], [37, 45, 43], [46, 39, 38, 87, 86], [58, 48, 52, 54, 62, 47, 51, 41, 61, 40, 59, 49, 53, 60], [55, 42, 50], [63, 64, 57, 56], [65, 71, 70, 79], [76, 66, 78, 77, 69], [73, 81, 68, 72, 80, 67], [82, 75, 74, 83], [84], [85]])], [[([13, 23, 5, 4, 7, 6, 1, 0, 3, 2, 14, 9, 8, 10, 15, 12, 11, 21], [48, 33, 44, 34, 16, 36, 50, 40, 59, 49, 30, 29, 42, 35, 22, 47, 41, 31, 43], [18, 26, 25, 17], [19, 20, 27, 28], [37, 45, 39, 46, 38, 87, 32, 24, 86], [73, 81, 75, 82, 51, 68, 74, 83, 60], [58, 52, 65, 72, 78, 80, 77, 67, 76, 66, 71, 70, 79, 69], [54, 62, 61, 53], [55, 63, 84, 64, 57, 85, 56]])], [[([5, 4, 7, 6, 1, 0, 3, 2, 14, 9, 8], [19, 20, 27, 11, 28, 10], [13, 12, 21], [29, 15, 23, 33, 35, 32, 22, 34, 24, 44, 16, 36, 31, 30], [18, 26, 25, 17], [37, 45, 39, 46, 38, 87, 86], [58, 48, 52, 54, 62, 47, 51, 41, 61, 40, 59, 49, 53, 60], [55, 56, 63, 84, 42, 64, 50, 57, 85, 43], [65, 71, 70, 79], [76, 66, 78, 77, 69], [73, 81, 75, 82, 68, 72, 80, 74, 83, 67]])], [[([13, 23, 5, 4, 7, 6, 1, 0, 3, 2, 14, 9, 8, 10, 15, 12, 11, 21], [48, 33, 44, 34, 16, 36, 50, 40, 59, 49, 30, 29, 42, 35, 22, 47, 41, 31, 43], [18, 26, 25, 17], [19, 20, 27, 28], [37, 45, 39, 46, 38, 87, 32, 24, 86], [73, 81, 75, 82, 51, 68, 74, 83, 60], [58, 52, 65, 72, 78, 80, 77, 67, 76, 66, 71, 70, 79, 69], [54, 62, 61, 53], [55, 63, 84, 64, 57, 85, 56]])], [[([5, 4, 7, 6, 1, 0, 3, 2, 14, 9, 8], [19, 20, 27, 11, 28, 10], [13, 12, 21], [29, 15, 23, 33, 35, 32, 22, 34, 24, 44, 16, 36, 31, 30], [18, 26, 25, 17], [37, 45, 39, 46, 38, 87, 86], [58, 48, 52, 54, 62, 47, 51, 41, 61, 40, 59, 49, 53, 60], [55, 56, 63, 84, 42, 64, 50, 57, 85, 43], [65, 71, 70, 79], [76, 66, 78, 77, 69], [73, 81, 75, 82, 68, 72, 80, 74, 83, 67]])]

```

Abb. 50: Sechs kombinierte Resultate

Um die finale Gruppierung zu erhalten, welche abgeleitet von den sechs verschiedenen Resultaten, die bestmögliche Modulgruppierung beinhaltet, wird ein Algorithmus erstellt. Um die Funktion des Algorithmus vereinfacht zu erläutern, wird die Funktion anhand eines Beispiels erklärt. Das Beispiel besteht aus vier verschiedenen Resultaten, welche mit zwölf Modulen berechnet wurden (siehe Abbildung 51).

```

Result1 = [[0, 2, 1], [6, 3, 7, 10], [4, 5, 8, 9, 11]]
Result2 = [[0, 2, 6], [1, 3, 7, 10, 11], [4, 5, 8, 9]]
Result3 = [[0, 2, 6], [1, 3, 4, 10, 11], [7, 5, 8, 9]]
Result4 = [[0, 2, 6, 11], [1, 3, 7, 10], [4, 5, 8, 9]]

```

Abb. 51: Resultate aus Azimut Optimierung

Der Algorithmus iteriert über jede Modulkombination und hat die Möglichkeit herauszufinden, wie oft die Module in derselben Gruppe vorkommen. Des Weiteren werden die Modulkombinationen nach der Häufigkeit sortiert.

Im nächsten Schritt werden die Modulkombinationen gefiltert und nur diejenigen weiterverwendet, welche eine Anzahl von ≥ 2 enthalten. Das Resultat dieses Abschnittes ist in der Abbildung 52 ersichtlich. Zum Beispiel ist das Modul 0 und 2 viermal ((0,2): 4) in derselben Gruppe.

```
{(0, 2): 4, (3, 10): 4, (5, 8): 4, (5, 9): 4, (8, 9): 4,
(0, 6): 3, (1, 3): 3, (1, 10): 3, (2, 6): 3, (3, 7): 3,
(4, 5): 3, (4, 8): 3, (4, 9): 3, (7, 10): 3, (1, 7): 2,
(1, 11): 2, (3, 11): 2, (4, 11): 2, (10, 11): 2}
```

Abb. 52: Sortierte und gefilterte Modulkombinationen

Anschliessend gruppiert dieser Algorithmus gleich wie der Algorithmus der String-Verschaltung die Module, welche mindestens zwei Verbindungen zwischen einander aufweisen (siehe Kapitel 7.6.3). Dadurch, dass die Modulkombinationen vorgängig gefiltert und sortiert werden, werden die Module, welche eine höhere Anzahl aufweisen, präferiert. In der Abbildung 53 sind die finalen Gruppen ersichtlich, welche aufgrund der beschriebenen Logik eingeteilt werden. (siehe Anhang A26).

```
Final groups: [{0, 2, 6}, {1, 3, 7, 10, 11}, {8, 9, 4, 5}]
```

Abb. 53: Finale String Gruppierung

In Zukunft könnten die Werte, welche für die Azimut Optimierung verwendet werden, auf die Stromknappheit in der Schweiz ausgelegt werden. In der Abbildung 54 ist ersichtlich, dass der Stromverbrauch in den Wintermonaten deutlich erhöht ist. Weiter ist dabei zu beachten, dass an Wochentagen der Energieverbrauch im Gegensatz zum Wochenende höher ist (Füglister, Frischknecht, & Bodenmann, 2022). Damit die Berechnung und Optimierung der String-Verschaltung weiter optimiert werden kann, könnte für die Verwendung von den Wechselrichter- und Modultypen eine Datenbank erstellt werden. Dadurch kann sichergestellt werden, dass die bestmögliche Option eruiert wird und die Rendite weiter verbessert werden kann.

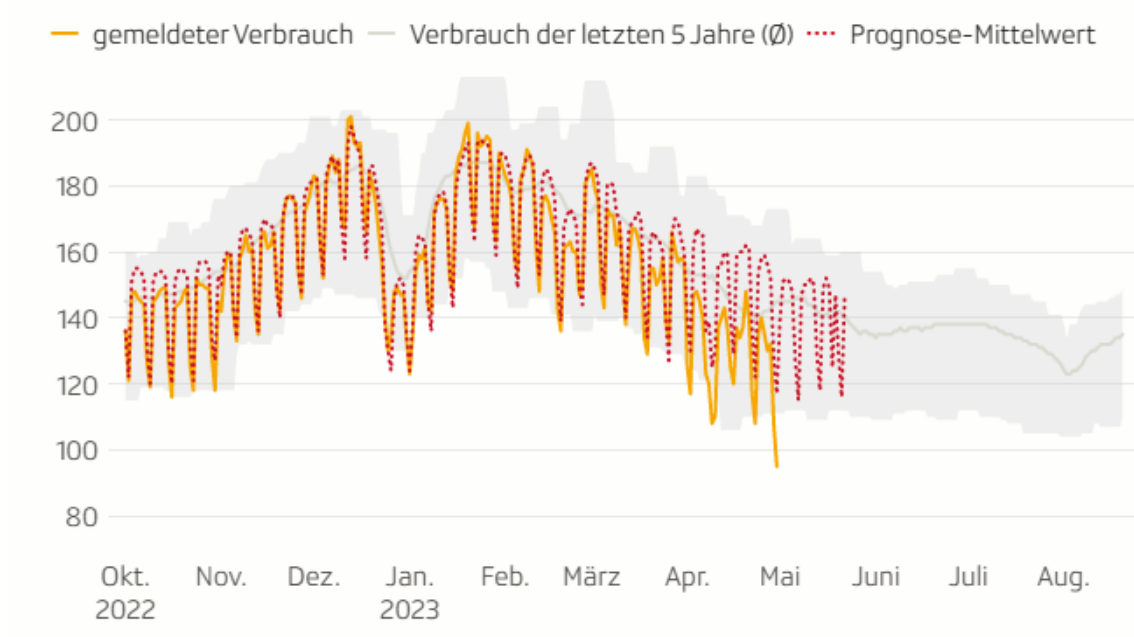


Abb. 54: Verbrauch gemäss Swissgrid (Füglister, Frischknecht, & Bodenmann, 2022)

7.8.2 Automatische Modulplatzierung

Eine mögliche Erweiterung des Skripts Level 4 wäre das manuelle Setzen eines Startpunktes, also der Ausgangsrichtung für die Modulplatzierung (siehe Abb. 55). Dafür müsste die Logik der Grid-Erstellung nochmals neu aufgerollt werde, da bisher mit einer Normal-Plane gearbeitet wurde, deren Startpunkt bisher manuell gesetzt werden musste (siehe Kapitel 7.7.3). Eine Option wäre mittels Python Skript mit den Inputs: Origin, End_point, Surface, length, width ausgehend vom Punkt «Origin» in die Richtung des Punktes «End_point» Zellen mit der Grösse «length/width» zu erstellen. Den Rest vom bisherigen Code in welchem die Öffnungen erkannt und Module verschnitten werden, könnte dabei wiederverwendet werden.

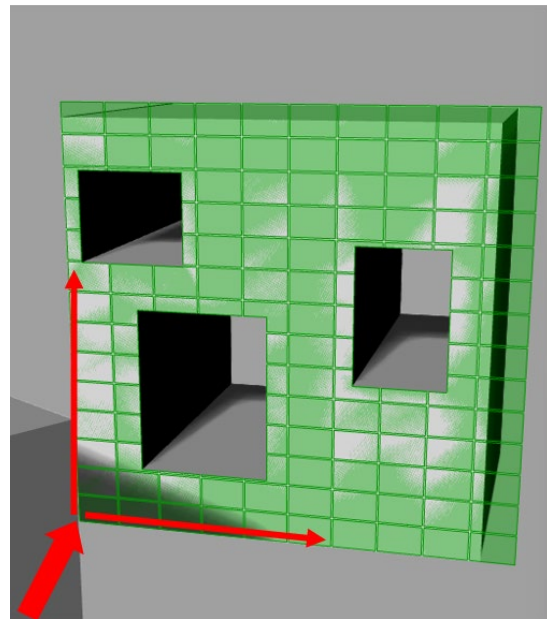


Abb. 55: Startpunkt manuell setzen

Mit dieser Änderung wäre es dem Nutzer möglich, den Ausgangspunkt (Eckpunkte oder Punkte in der Fassade) der Modulgenerierung selbst zu bestimmen. Dadurch würde sich die Position und Anzahl der verschnittenen Module ändern sowie ein anderes Modulbild auf der Fassade entstehen.

Des Weiteren kann mittels eines Galapagos Algorithmus, der Prozess der optimalen Modulplatzierung optimiert werden. Identisch wie im Kapitel 7.6.3 beschrieben, kann mittels veränderbaren Optionen und einem Fitnessindex die optimale Anordnung der Module eruiert werden. Als veränderbare Optionen werden die Modultypen und die Ausrichtung Hoch- und Querformat klassifiziert (siehe Abbildung 56). Als Fitnessindex wird die Fläche der generierten Photovoltaik Modulen angegeben. Durch diesen Vorgang kann sichergestellt werden, dass die Option ausgewählt wird, welche die grösste mögliche nutzbare Fläche aufweist. In Zukunft könnte eine Datenbank mit allen Modultypen verknüpft werden, um alle möglichen Optionen in der Optimierung miteinzubeziehen. Der Nachteil dieser Optimierung liegt darin, dass dieser Prozess rechenintensiv ist und dadurch schwierig ist im Level 4 zu implementieren.

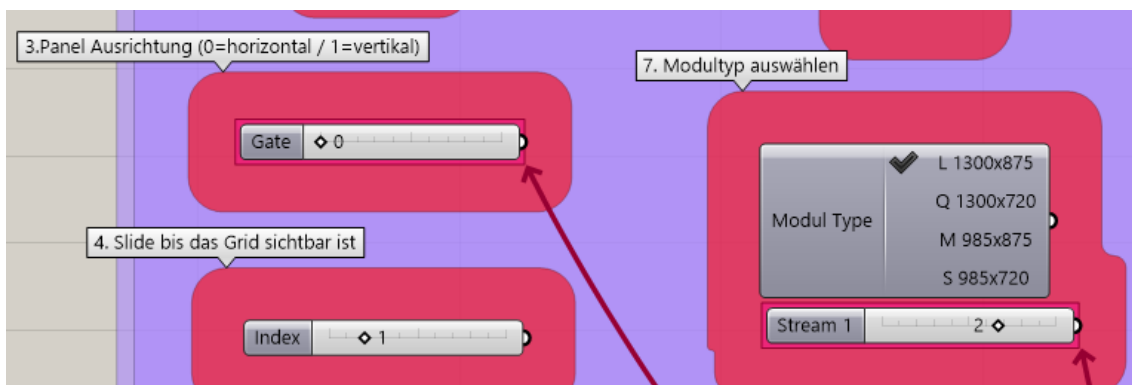
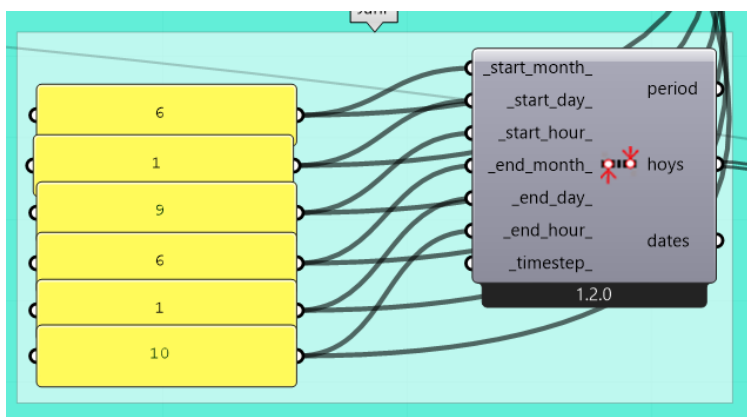


Abb. 56: Veränderbare Optionen Modultyp / Ausrichtung

7.8.3 Bypass Implementierung

Grundsätzlich besitzt jedes aktuelle PV-Modul eine oder mehrere Bypass Dioden (siehe Kapitel 3.5). Diese funktionieren nach folgendem Prinzip: Wird ein Modul in einem String verschattet, wird dieses «übergangen» und das jeweilige Modul hat zu diesem Zeitpunkt keine Leistung. Auf Grundlage der Tabelle in Anhang A27 wurde diese Logik im Forschungs-Script «Bypass_Implementierung» für einen Stundenwert umgesetzt. Als Modell wird zuerst der maximale Einstrahlungswert eines Panels mit derselben Ausrichtung wie die anderen Panels berechnet. Anschliessend werden die rohen Einstrahlungswerte sowie die berechneten Ertragswerte als Input angegeben. Die String-Gruppen werden benötigt, sodass anschliessend der Ertrag pro Gruppe berechnet werden kann. Der Threshold welcher manuell angepasst werden kann, gibt an, wie viel Prozent Abweichung ein Panel zum maximalen Einstrahlungswert des Strings haben darf und immer noch als «nicht-verschattet» angegeben wird.

Ist der Einstrahlungswert des jeweiligen Moduls grösser oder gleich 80 % des maximalen Wertes pro String, wird dieses Modul zu einer neuen Liste hinzugefügt. Anschliessend wird ausgerechnet, wie viel Prozent der Module pro String in der Sonne sind. Danach wird der Gruppenenertrag pro String, sowie der Gesamtertrag aller Strings berechnet. Damit berechnet werden kann, wie viel Prozent Ertragsverlust auf Grundlage der Verschattung sowie der Einstrahlung auftreten, wird über ein If / Else – Statement der Prozentwert aus der Tabelle Anhang A27 zugewiesen. Zum Schluss wird der Gruppenwert mal die zugewiesene Prozentzahl, sowie der Totalertrag aller Gruppen berechnet.



Bei der berechneten Stunde in diesem Beispiel (siehe Abbildung 57) handelt es sich um den 1. Juni von 9:00 – 10:00 Uhr, wobei der Maximal-Einstrahlungswert bei ca. 1000W/m² liegt.

Abb. 57: Berechnete Stunde

Aufgrund der Komplexität wurde bisher nur eine Stunde mit der Bypass-Logik implementiert. Das Skript könnte so erweitert werden, dass der maximale Einstrahlungswert über die stündliche Berechnung für alle 8760 Stundenwerte im Jahr berechnet wird. Ausserdem könnten alle Einstrahlungswerte durch eine separate stündliche Berechnung durchlaufen. So würden alle stündlichen Werte für die maximale Einstrahlung sowie alle Panels berechnet werden. Danach müsste die Skript-Logik so angepasst werden, dass sie durch alle stündlichen Werte iteriert und diese am Schluss kombiniert.

7.9 Rechenleistung

Alle Skripts wurden auf folgendem System mit den Beispielmodellen getestet:

Prozessor: Intel core i7 10700KF

Grafikkarte: NVIDIA GeForce RTX 2070 Super 8 GB

RAM : 32 GB, 3200MHz

Die Berechnung mit stündlichen Werten hat im Level 3 durchschnittlich 1 Minute gedauert. Diese Zeit variiert stark von der Anzahl angegebener Geometrien sowohl der Panels, wie als auch der Kontext Geometrien.

Es muss beachtet werden, dass die stündliche Berechnung mit 730-mal mehr Kennzahlen als die monatliche Berechnung durchgeführt wird. Deshalb kann bei der Verwendung des Galapagos-Algorithmus im stündlichen Modus das Skript abstürzen. Bevor die Berechnung gestartet wird, sollte darauf geachtet werden, dass alle notwendigen Geometrien angegeben sind. Für Änderungen sollte die Berechnung immer mit einem Doppelklick auf den «Run me» - Button gestoppt werden.

8. Erkenntnisse

Die Erkenntnisse in diesem Kapitel beziehen sich ausschliesslich auf die Berechnung einer vordefinierten Fassade des Modells in Abbildung 58 mit dem Skript Level 3 (siehe Kapitel 7.6). Die Wirtschaftlichkeitsberechnung wird in Kapitel 7.6.5 detailliert aufgeführt.

Für die Berechnung wurden folgende wirtschaftlich relevante Parameter eingegeben:

- Kosten pro MLPE-Optimizer: **60**
- Eigenverbrauch pro Jahr in %: **50**
- Investitionskosten PV-Module: **30'000**
- Wechselrichter: **«Von Galapagos automatisch ermittelt»**
- Instandhaltungskosten in %: **1.5**
- Förderbeitrag Kanton in % der Investitionskosten: **20**
- Strompreis Durchschnitt HT/NT: **0.27015**
- Einspeisevergütung Durchschnitt HT/NT: **0.15415**
- Jahre: **30**

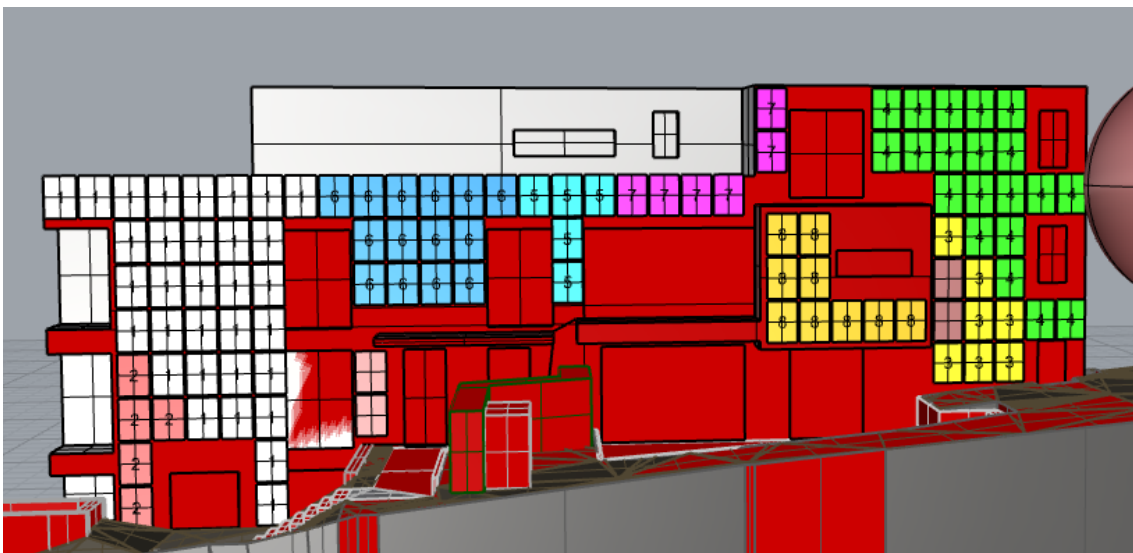


Abb. 58: Ergebnis Berechnung Skript Level 3

8.1 Kosten/ Erträge/ Differenzen

Weil die MLPE im Gegensatz zu einer herkömmlichen String-Verschaltung den spezifischen Ertragswert von jedem Modul in einem String im Endertrag verwenden kann (siehe Kapitel 3.5), ist der Ertrag in kWh/a dieser MLPE-Strings zu fast jedem Zeitpunkt höher. Die optimierte Variante ohne MLPE zieht den Vorteil daraus, keine zusätzlichen Investitionen zu verlangen. Daher sind die Gesamtinvestitionskosten im berechneten Modell mit MLPE um 17 % höher als ohne MLPE (siehe Abb. 59 & 60).

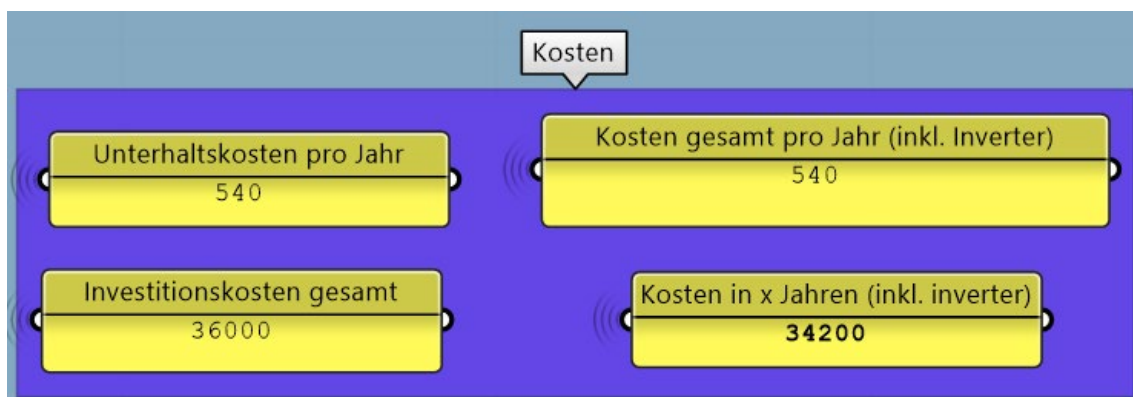


Abb. 59: Kosten ohne MLPE

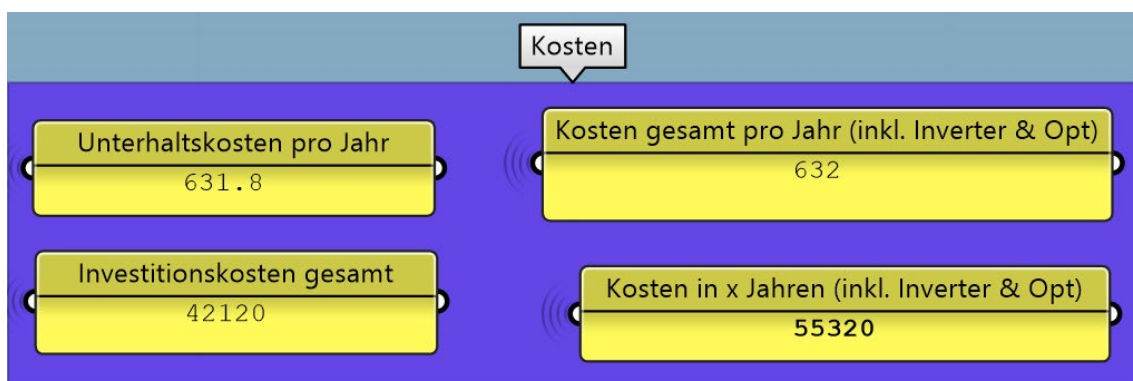


Abb. 60: Kosten mit MLPE

Weil die Unterhaltskosten direkt von den Investitionskosten abhängig sind, sind auch diese mit MLPE um 17 % höher als ohne MLPE-Installation. Grosse Auswirkungen hat die Installation von MLPE's vor allem auf die laufenden Kosten nach der Lebensdauer von Wechselrichter und allen MLPE. Nach Ablauf der angenommenen Lebensdauer von 15 Jahren müssen alle Geräte neu gekauft und installiert werden. Das bedeutet es findet eine zyklische Investition alle 15 Jahre mit dem Umfang aller Wechselrichter und

MLPE statt, wobei ohne MLPE nur die Wechselrichter ersetzt werden müssen. Dies schlägt sich in den «Kosten in X Jahren» nieder, sobald die Berechnung für mehr als 15 Jahre durchgeführt wird. Gemäss der Berechnung für das vorliegende Modell sind die Kosten für ein System mit MLPE nach 30 Jahren um 61 % höher als das optimierte System ohne MLPE, was ungefähr 21'200 CHF sind. Dies bei einem Mehrertrag durch MLPE nach 30 Jahren von 12 %, was knapp 7'000 CHF entspricht (siehe Abb. 61 & 62).

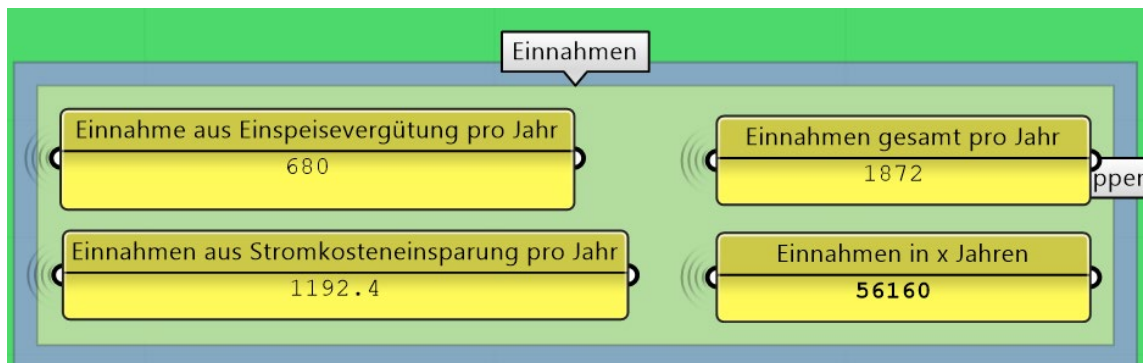


Abb. 61: Einnahmen ohne MLPE

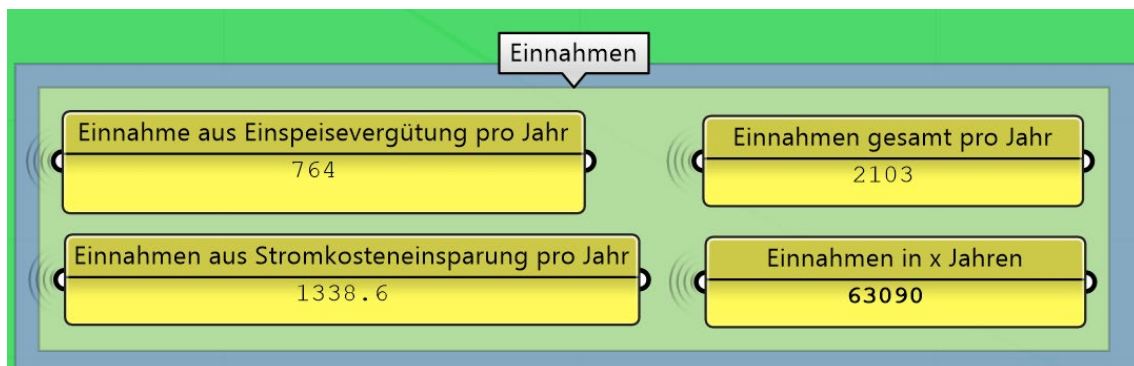


Abb. 62: Einnahmen mit MLPE

Aus dem Wissen, dass die Installation und Wartung der MLPE nicht in die Wirtschaftlichkeitsrechnung mit eingeflossen sind, lässt sich ableiten, dass der Einsatz von MLPE im vorliegenden Objekt keinen wirtschaftlichen Nutzen, gar eine Verschlechterung des Systems zur Folge hätte.

Verwendete Module	
Anz. Module in Verwendung	
Von	102
sind	98
Module in Verwendung	
Blindmodule ohne Ertrag	
0	4
1	13
2	23
3	31

Im berechneten System ohne MLPE werden 4 Module aufgrund ihrer Position oder ihres Einstrahlungswertes als Blindmodule deklariert (siehe Abb. 63). Das bedeutet, die Investitionskosten dieses Systems würden noch um ca. 1'000 CHF gesenkt werden.

Abb. 63: Deklaration von Blindmodulen

Auf den Ertrag bezogen ist, wie vorgängig beschrieben, das MLPE-System besser. In dieser berechneten Situation hat das MLPE System einen um 12 % höheren Ertrag als das optimierte System ohne MLPE, was ungefähr 1'082 kWh/a entspricht. Im Gegensatz zum herkömmlichen, nicht-optimierten String-Verschaltungsverfahren, weist die optimierte Variante 13 % mehr Ertrag auf, was ca. 1'123 kWh/a entspricht (siehe Abb. 64).

Optimierte Strings	Mit MLPE	"Standard" Strings
kWh/a 8827.550054	kWh/a 9909.736508	kWh/a 7704.942309
Differenz zwischen optimierten und MLPE Strings -12.0%		Differenz zwischen optimierten und "Standard" Strings 13.0%

Abb. 64: Erträge und Differenzen

8.2 Jährlich-/ Monatliche-/ Stündliche Berechnung

Wie im Kapitel 6.1 beschrieben, sollte als Forschung überprüft werden, welchen Einfluss die Verwendung von jährlichen/ monatlichen und stündlichen Einstrahlungswerten der Panels auf den Gesamtertrag der Fassade hat. In Abbildung 65 ist ersichtlich, dass sich die Erträge der jährlichen sowie monatlichen Berechnung wie erwartet verhalten haben. Da bei der monatlichen Berechnung für jeden Monat der String Ertrag neu berechnet wird, entstehen mehr lokale Minima, welche den Gesamtertrag minimieren.

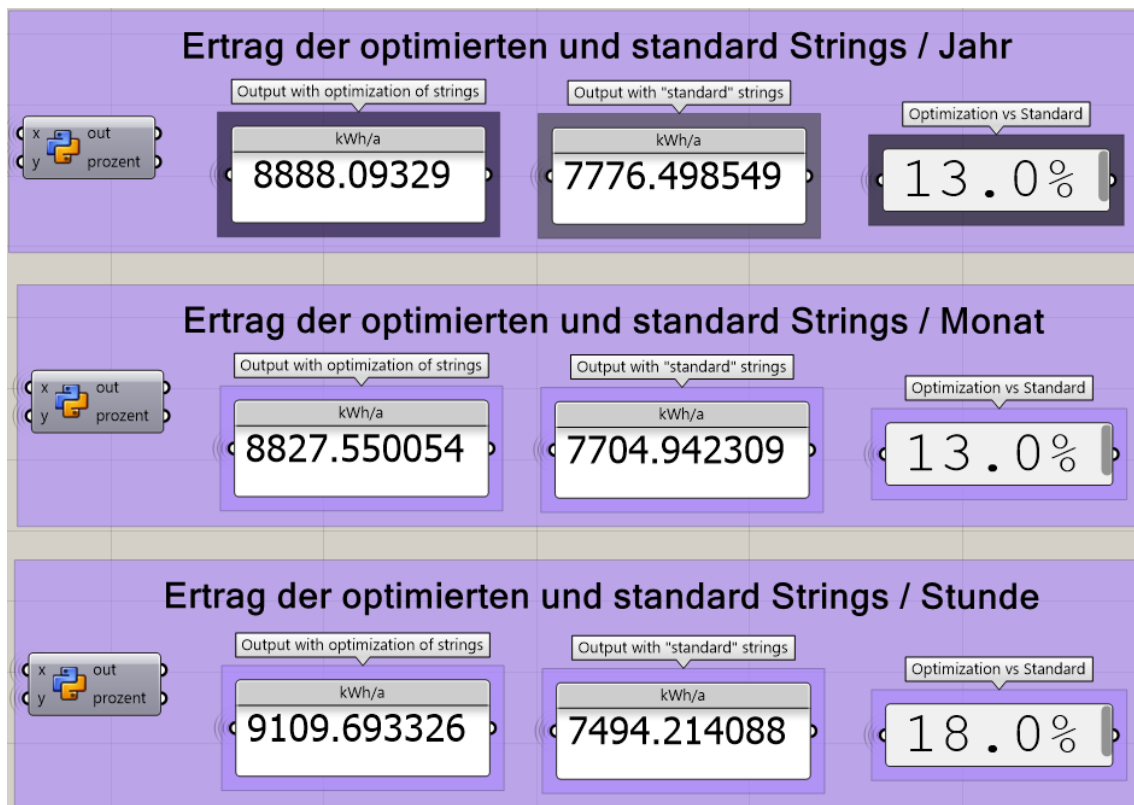


Abb. 65: Vergleich Jahres-/ Monats-/ Stundenwerte

Die Berechnung mit stündlichen Werten fällt bei den «Standard» Strings wie erwartet niedriger als in den monatlichen und jährlichen Berechnungen aus. Da es sich bei der Berechnung mit stündlichen Werten jedoch nicht um ein Ladybug-, sondern Honeybee Tool handelt, sind die Ertrags-Wertemarginal unterschiedlich. Deshalb kann kein aussagekräftiger Vergleich zwischen den Berechnungen gezogen und der grössere Wert der optimierten Strings auch nicht präzise nachvollzogen werden. Dies lässt sich in einer einfachen Summe aller berechneten Werte pro Jahr, Monat und Stunde visualisieren (siehe Abbildung 66).

Jahr	Monat	Stunde
{0}	{0;0}	{0}
0 61935.853175	0 61928.613835	0 64454.01614

Abb. 66: Summe aller berechneten Werte pro Jahr, Monat, Stunde

Da es sich bei der jährlichen sowie der monatlichen Berechnung um dasselbe Sonnenanalysetool «Ladybug» handelt, kann ein Vergleich gezogen und die Vermutung, dass der Gesamtertrag sich bei der Berechnung mit mehr Daten verringert (Kapitel 6.1) widerlegt werden. Es kann trotz der nicht möglichen Vergleichbarkeit der Werte mit der stündlichen Berechnung angenommen werden, dass dieser Wert noch niedriger als die monatliche Berechnung ausfallen würde. Eine weitere Erkenntnis ist, dass je genauer die Berechnung ist, desto grösser wird die Ertrags-Differenz zwischen den optimierten und der Standard-Strings. Dies unterstreicht die Bedeutung einer Optimierung der Strings.

8.3 String-Verschaltung Optimierung mit Azimut Werten

Die Erkenntnisse in diesem Kapitel beziehen sich ausschliesslich auf die Berechnung einer vordefinierten Fassade des Modells in Abbildung 67 mit dem Forschungsskript (siehe Kapitel 7.8.1). Für die Berechnung wurde der Wechselrichter SolarEdge SE100KuS, welcher eine maximale Eingangsspannung von 1000 Volt aufweist, verwendet.

Mittels den Azimut Werten, welche im Kapitel 7.8.1 aufgelistet wurden, konnte gegenüber der jährlichen Optimierung keine Verbesserung des Ertrags festgestellt werden (siehe Abb. 68). Dies kann, aber damit zusammenhängen, dass bei dem Algorithmus, welcher die «Finalen Gruppen» erstellt, nicht die Modulkombinationen präferiert werden, welche die grösste Anzahl Verbindungen aufweisen. Im Beispiel welches erläutert wird funktioniert dies. Aufgrund dessen, dass die Berechnung nicht wie im Beispiel in PyCharm vollzo-

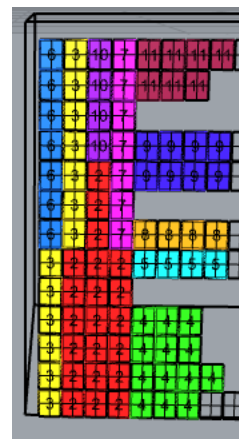


Abb. 67: Ergebnis Azimut Skript

gen wird, sondern im Python Node in Grasshopper, kann es sein, dass Formatierungsprobleme die Optimierung beeinträchtigen.

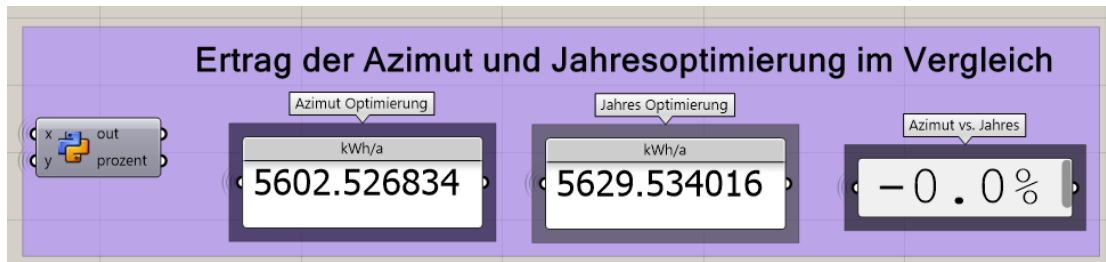


Abb. 68: Azimut Optimierung im Vergleich mit der Jahresoptimierung

Weiter wurde eine Berechnung mit demselben Script nur mit Morgenstunden vom 6. und 7. Juli vollzogen. Durch diese Optimierung kann im Vergleich zur Optimierung mit den Jahreswerten eine Differenz von 5 % erzielt werden (siehe Abb. 69). Daraus lässt sich schließen, dass der Algorithmus besser funktioniert, wenn die Azimut Werte sich in ähnlichen Zeitfenstern befinden.

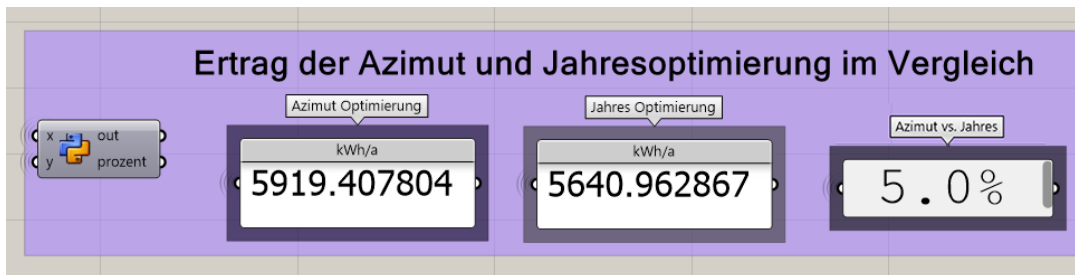


Abb. 69: Azimut Optimierung (6. und 7. Juli, Morgenstunden) im Vergleich mit der Jahresoptimierung

Dieser Vergleichswert ist aber mit Vorsicht zu betrachten. Denn mit der String-Verschaltung Optimierung aus dem Level 3, welche mit Jahreswerten optimiert wird, gibt es zudem die Möglichkeit, den Ertrag aufgrund von den veränderbaren Wechselrichtertypen zu optimieren. Des Weiteren bietet das Script Level 3 die Möglichkeit, die String-Verschaltung, mittels der Rendite zu optimieren, dies ist aufgrund der differenzierten Logik des Algorithmus im Forschungs-Script nicht möglich (siehe Kapitel 7.6.3).

8.4 Bypass Erträge für eine Stunde

Die Erkenntnisse in diesem Kapitel basieren ausschliesslich auf der Berechnung des Modells in Abbildung 70. Für die Berechnung wurde der Wechselrichter SolarEdge SE100KuS verwendet, welcher eine maximale Eingangsspannung von 1000 Volt aufweist. Die Bypass Implementierung wurde bereits im Kapitel 7.8.3 erfasst. Im Vergleich zu den vorherigen Berechnungen ohne Bypass-Implementierung, wird in derselben Stunde kein grosser Unterschied zur Berechnung mit der Bypass-Implementierung festgestellt (siehe Abb. 71).

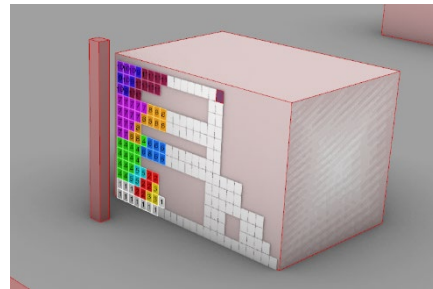


Abb. 71 Berechnetes Modell

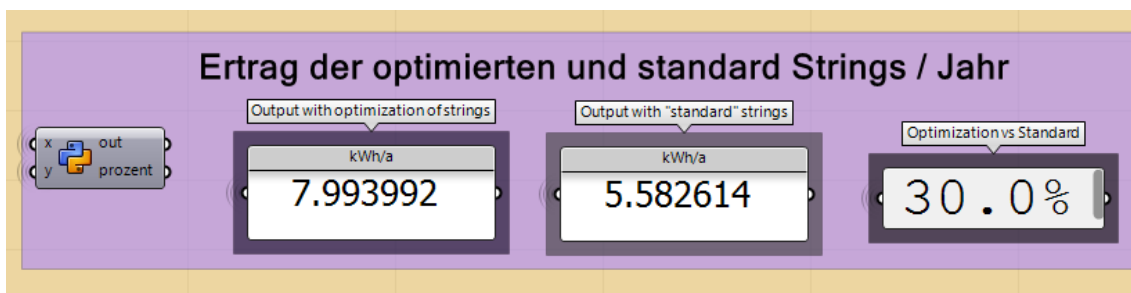


Abb. 70: Berechnung ohne Bypass-Implementierung

Die Verhältnisse zwischen optimiert, Standard und MLPE-Strings bleiben ausserdem bestehen. Es gibt einen 8-prozentigen Unterschied der Erträge zwischen optimiert und Standard, mit und ohne Bypass-Dioden (siehe Abb. 72).

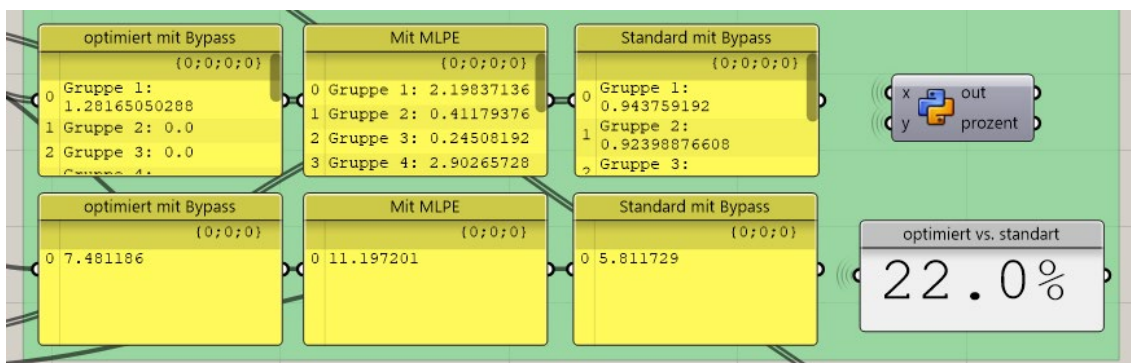


Abb. 72: Berechnung mit Bypass-Implementierung

Eine grundsätzliche Erkenntnis liegt darin, dass die Bypass-Dioden vor allem einen Vorteil bei nicht-optimierten Strings haben. Optimierte Strings, wie sie in dieser Arbeit vorgestellt wurden, erleiden eine Ertragseinbusse. In diesem Beispiel mit der angegebenen Stunde handelt es sich um eine Einbusse bei den optimierten Strings von ca. 10 %, wobei die Standard-Strings eine Ertragssteigerung von 4 % erfahren.

9. Diskussion und Interpretation

Forschungsfrage:

«Es stellt sich die zentrale Forschungsfrage, wie der Ertrag und der Planungsprozess einer Photovoltaikanlage, ohne MLPE verbessert werden kann.»

Hypothese:

Die forschungsleitende Vermutung lautet: «Durch die Optimierung der String-Verschaltung kann der Ertrag einer Photovoltaikanlage an einer Fassade um 10 % erhöht werden. Der Arbeitsablauf der Photovoltaikplanung an Fassaden kann ebenfalls vollständig automatisiert werden.»

F.1: Kann durch die Optimierung der String-Verschaltung eine Erhöhung des Ertrags erzielt werden?

Wie im Kapitel 8.1 beschrieben, konnte der Ertrag der berechneten Photovoltaik Fassade gegenüber einer standardmässig geplanten Fassade um ca. 18 % gesteigert werden. Dies beantwortet zum einen die Forschungsfrage, widerlegt aber auch die Hypothese H1, dass der Ertrag einer Fassadenanlage durch Optimierung um 10 % erhöht werden kann.

F.2: Kann der gesamte Photovoltaikplanungsteil von Simulation über Analyse und Planung bis hin zur Ertragsrechnung automatisiert werden?»

Damit der gesamte Photovoltaikplanungsteil automatisiert werden kann, müssen folgende Teilfunktionen im Skript inkludiert sein:

1. Generierung von PV-Modulen und Platzierung auf einer vordefinierten Fassade
2. Ertragsanalyse der generierten Panels auf der Fassade erstellen

3. Berechnung der effizientesten String-Verschaltung
4. Berechnung der Wirtschaftlichkeit mit Kostenkennzahlen zu Installation, Investition, Unterhalt, Einsparung, Energieeinspeisung, Energieeinkauf
5. Anpassung der String-Verschaltung auf Grundlage der Optimierung von ROI und Energieertrag
6. Ausgeben der Daten in einfachem User Interface

Die Teilfunktionen 1 - 6 wurden bis auf die Kostenkennzahlen alle implementiert. Die Kostenkennzahlen zur Installation von Wechselrichtern- oder Modulen, konnten aus Zeitgründen nicht implementiert werden. Trotzdem kann daraus abgeleitet werden, dass es möglich ist, mit vorgängigem planerischen Aufwand eine Software zu erstellen, welche den gesamten Planungsprozess einer PV-Anlage automatisieren kann. Der Vorteil einer automatisierten Planung ist die einfache Variantenerstellung und Erweiterbarkeit der Funktionalitäten durch Ergänzen des Skripts. Mögliche Ergänzungen sind im Kapitel 7.8 aufgelistet.

10. Reflexion

Photovoltaik-Anlagen sind massgebend für den Umstieg auf nachhaltige Energiesysteme in der Zukunft. Die vorhandene Technologie so zu nutzen, dass das Maximum an Ertrag aus ihr gezogen werden kann, sollte ebenfalls ein Ziel der nachhaltigen Entwicklung in der Branche sein. In Form dieser Projektarbeit konnte aufgezeigt werden, dass durch eine logische Verschaltung von PV-Modulen der Ertrag erhöht, und die Kosten im Vergleich zur herkömmlichen Planungsweise mit MLPE verringert werden können. Die verschiedenen Levels der Skripts beziehen sich auf verschiedene Phasen der Planung und sprechen dadurch auch alle bei der Planung tätigen Personen an. So kann das Level 1 eine schnelle Aussage darüber geben, ob sich eine Fassade für die Montage einer Anlage lohnt, wobei in der vollautomatisierten Variante «Level 4» detaillierte Angaben zu Modultypen, Platzierung, Wechselrichtermodellen und Wirtschaftlichkeitsdaten angegeben werden können. Mithilfe des erarbeiteten Skripts kann ebenfalls Kennzahlen basiert aufgezeigt werden, zu welchen Zeitpunkten die Photovoltaik den besten Ertrag erzielt. Der Nutzer erhält dabei die Möglichkeit, den zu berechnenden Zeitpunkt selbst auszuwählen und dadurch seine spezifische Photovoltaiklösung zu erarbeiten. Einer der grössten Vorteile bei der Arbeit mit automatisierten Berechnungsmethoden stellt der schnelle Vergleich verschiedener Varianten dar. Dies ist vor allem im Bereich der Photovoltaik ein grosses Thema, da bei deren Planung technische, sowie architektonische Faktoren eine Rolle spielen. Eine vom Architekten geplante Variante kann so zeitnah auf ihre energetische Relevanz überprüft werden.

Im Verlauf der Arbeit konnte auch festgestellt werden, dass die Genauigkeit der erhaltenen Ergebnisse stark von der Anzahl Einstrahlungsdaten abhängt. Dies vor allem in Situationen mit stark verschatteten PV-Modulen.

Da in der Arbeit einige Themen aus zeitlichen Gründen nicht direkt in das Skript implementiert werden konnten, wurden verschiedene zusätzliche Forschungsanalysen angestellt. Diese sollen zum einen die Komplexität des Themas ausdrücken, sowie aber auch die Ausbaubarkeit des Projektes darstellen. Es werden dabei zukünftige Anwendungsbeispiele hervorgehoben, wie zum Beispiel der Abgleich von Spitzenwerten eines öffentlichen Stromerzeugers mit der Optimierung der String-Verschaltung. Ebenso wer-

den Erweiterungen des Skripts via Datenbanken zu Wechselrichter- oder Photovoltaiktypen oder String Optimierungen mittels Azimut Werten hervorgehoben.

Das Ziel, eine automatisierte Planungs- und Berechnungslösung für Photovoltaik Fassaden für den Industriepartner Basler & Hofmann zu schaffen, wurde erreicht.

Tabellenverzeichnis

Tabelle 1 Einstrahlungswerte B&H (Excel A10).....	16
Tabelle 2: Grundlage Einstrahlungswerte (Excel A10).....	17
Tabelle 3: Berechnete Einstrahlungswerte (Variante 1) (Excel A10).....	17
Tabelle 4: Grundlage Einstrahlungswerte (Excel A10).....	17
Tabelle 5: Berechnete Einstrahlungswerte (Variante 2) (Excel A10).....	18
Tabelle 6: Gesamtertragswerte (kWh/a) Variante 1 und Variante2 (Excel A10).....	18

Abbildungsverzeichnis

Abb. 1: "Powerswitcher" Axpo Szenario aktuelles Ausbautempo bis 2050 (Axpo, 2023)	6
Abb. 2: Szenario 2050 Ausbau PV x5.5 (Axpo, 2023)	7
Abb. 3: jährliche Einstrahlung auf unterschiedliche Dachflächenorientierungen im Vergleich zur Horizontalen (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 12)	9
Abb. 4: Minimierung der Verschattung in Sommermonaten (ebd.)	10
Abb. 5: Montagetechniken rahmenloser PV-Module (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 52)	12
Abb. 6: Punktförmige Halterungen (Weller, Hemmerle, Jakubetz, & Unnewehr, 2009, S. 54)	13
Abb. 7: Sonneneinstrahlung B&H (PV Syst)	16
Abb. 8: Zoneneinteilung B&H (PV Syst)	16
Abb. 9: Herkömmliche Berechnung ohne String-Verschaltung	19
Abb. 10: Genaue Berechnung String-Verschaltung	20
Abb. 11: Ertragsberechnung pro Jahr und pro Monat	20
Abb. 12: Übersicht Preise elektrische Optimizer (Energie-Experten, 2022)	24
Abb. 13: Modell für Skript Level 1	25
Abb. 14: Modell für Skript Level 2	25
Abb. 15: Modell für Skript Level 3	26
Abb. 16: 22 Modell für Skript Level 4	26
Abb. 17: Vorlage Basler & Hofmann (Anhang A12)	28
Abb. 18: "Standard" Auslegung der Strings	28
Abb. 19: Basis Inputs Level 1	30
Abb. 20: Wirtschaftlichkeit Inputs	31
Abb. 21: Output Skript Level 1	33
Abb. 22: User Interface Skript Level 2 und	34
Abb. 23: Visualisierungstool	35
Abb. 24: Output Skript Level 2	37
Abb. 26: Galapagos Nodes (Optimierung jährlich / vordefinierte Stunden)	38
Abb. 25: Nutzerauswahl (Optimierung jährlich / vordefinierte Stunden)	38
Abb. 27: Bsp. Logik String-Verschaltung	40
Abb. 28: Inputs Algorithmus	41
Abb. 29: Filterung mögliche Modulkombinationen	42
Abb. 30: Gruppierung Module / Finale String Gruppen	43
Abb. 31: Slider Ertragsdifferenz & Minus Faktor	44
Abb. 32: Eingangsspannungswerte der Wechselrichter	44
Abb. 33: Slider Wechselrichter	44
Abb. 34: Stündliche Ertragsberechnung	45

Abb. 35: Output Skript Level 3, Vergleiche.....	48
Abb. 36: Output Skript Level 3, ohne MLPE.....	48
Abb. 37: Output Skript Level 3, mit MLPE	49
Abb. 38: Raster auf Fassade erstellen.....	51
Abb. 39: Neue Module mit Abstand erstellen	52
Abb. 40: Erkennung von Öffnungen	52
Abb. 41: Module innerhalb Öffnungen entfernt.....	53
Abb. 42: Verschnittene Module entfernt	53
Abb. 43: Verschnittene Module = Blindmodule	53
Abb. 44: Offsetkurve Aussenkante & Normals direction	54
Abb. 45: "Backen" auf spezifischen Layer.....	54
Abb. 46: Output Skript Level 4, ohne MLPE.....	55
Abb. 47: Output Skript Level 4, mit MLPE	56
Abb. 48: Output Skript Level 4, Vergleiche.....	56
Abb. 49: Veränderbare Parameter Azimut Optimierung.....	57
Abb. 50: Sechs kombinierte Resultate.....	58
Abb. 51: Resultate aus Azimut Optimierung.....	58
Abb. 52: Sortierte und gefilterte Modulkombinationen	59
Abb. 53: Finale String Gruppierung.....	59
Abb. 54: Verbrauch gemäss Swissgrid (Füglister, Frischknecht, & Bodenmann, 2022).....	60
Abb. 55: Startpunkt manuell setzen.....	60
Abb. 56: Veränderbare Optionen Modultyp / Ausrichtung.....	61
Abb. 57: Berechnete Stunde	62
Abb. 58: Ergebnis Berechnung Skript Level 3.....	65
Abb. 59: Kosten ohne MLPE	66
Abb. 60: Kosten mit MLPE.....	66
Abb. 61: Einnahmen ohne MLPE	67
Abb. 62: Einnahmen mit MLPE	67
Abb. 63: Deklaration von Blindmodulen	68
Abb. 64: Erträge und Differenzen	68
Abb. 65: Vergleich Jahres-/ Monats-/ Stundenwerte.....	69
Abb. 66: Summe aller berechneten Werte pro Jahr, Monat, Stunde	70
Abb. 67: Ergebnis Azimut Skript	70
Abb. 68: Azimut Optimierung im Vergleich mit der Jahresoptimierung.....	71
Abb. 69: Azimut Optimierung (6. und 7. Juli, Morgenstunden) im Vergleich mit der Jahresoptimierung	71
Abb. 71: Berechnung ohne Bypass-Implementierung.....	72
Abb. 70 Berechnetes Modell	72
Abb. 72: Berechnung mit Bypass-Implementierung.....	72

Literaturverzeichnis

- andheum. (13. 11. 2019). *Metahopper*. Abgerufen am 08. 05. 2023 von Food4Rhino:
<https://www.food4rhino.com/en/app/metahopper>
- Axpo. (2023). *Power Switcher*. Abgerufen am 28. 04. 2023 von
<https://powerswitcher.axpo.com/>
- Baumgärtner, P. D. (27. 04. 2021). *Optimizer: Nur ein Hype oder die Zukunft?*
Abgerufen am 30. 04. 2023 von [https://www.bulletin.ch/de/news-
detail/optimizer-nur-ein-hype-oder-die-zukunft.html](https://www.bulletin.ch/de/news-detail/optimizer-nur-ein-hype-oder-die-zukunft.html)
- Chris. (01. 04. 2022). LB cumulative sky matrix. [Blog]. Abgerufen am 28. 04. 2023
von <https://discourse.ladybug.tools/t/lb-cumulative-sky-matrix/18101>
- Drzimalla, P. (08. 02. 2023). *Fassaden-Photovoltaik: Wann lässt die Schweiz die Sonne
ran?* Schweiz. Abgerufen am 24. 04. 2023 von [https://www.energie-
experten.ch/de/wohnen/detail/fassaden-photovoltaik-wann-laesst-die-schweiz-
die-sonne-ran.html](https://www.energie-
experten.ch/de/wohnen/detail/fassaden-photovoltaik-wann-laesst-die-schweiz-
die-sonne-ran.html)
- Energie-Experten. (10. 11. 2021). *Wartung der Solaranlage: Maßnahmen und Kosten
im Überblick*. Abgerufen am 03. 05. 2023 von [https://www.energie-
experten.org/erneuerbare-energien/photovoltaik/betrieb/wartung](https://www.energie-
experten.org/erneuerbare-energien/photovoltaik/betrieb/wartung)
- Energie-Experten. (08. 05. 2022). *Leistungsoptimierer: SolarEdge und TIGO im
Vergleich*. Abgerufen am 26. 04. 2023 von [https://www.energie-
experten.org/erneuerbare-energien/photovoltaik/montage/leistungsoptimierer](https://www.energie-
experten.org/erneuerbare-energien/photovoltaik/montage/leistungsoptimierer)
- Füglister, L., Frischknecht, L., & Bodenmann, N. (14. 12. 2022). *So viel Strom
verbraucht die Schweiz – und weitere Grafiken*. Abgerufen am 22. 05. 2023 von
[https://www.srf.ch/news/schweiz/moegliche-energiekrise-so-viel-strom-
verbraucht-die-schweiz-und-weitere-grafiken](https://www.srf.ch/news/schweiz/moegliche-energiekrise-so-viel-strom-
verbraucht-die-schweiz-und-weitere-grafiken)
- Knopf, H. (30. 03. 2022). *5 Mythen über DC-Leistungsoptimierer*. Abgerufen am 30.
04. 2023 von [https://www.sma-sunny.com/5-mythen-ueber-dc-
leistungsoptimierer/](https://www.sma-sunny.com/5-mythen-ueber-dc-
leistungsoptimierer/)
- Krymsky, Y. (31. 01. 2014). *yazdanistudioresearch*. Abgerufen am 13. 05. 2023 von
Reaktionsfähige Skins:
[https://yazdanistudioresearch.wordpress.com/2014/01/31/optimization-with-
galapagos/](https://yazdanistudioresearch.wordpress.com/2014/01/31/optimization-with-
galapagos/)
- Kutschera, P. (10. 2014). *Wirtschaftlichkeitsberechnung von PV-Anlagen*. Abgerufen
am 17. 05. 2023 von Swissolar:
[https://www.swissolar.ch/fileadmin/user_upload/Tagungen/Wirtschaftlichkeit_P
V_V3.pdf](https://www.swissolar.ch/fileadmin/user_upload/Tagungen/Wirtschaftlichkeit_P
V_V3.pdf)

- Ladybug. (01. 01. 2023). *Analyze Data*. Abgerufen am 28. 04. 2023 von https://docs.ladybug.tools/ladybug-primer/components/1_analyzedata
- Ladybug Tools. (17. 01. 2023). *Ladybug Tools*. Abgerufen am 08. 05. 2023 von Food4Rhino: <https://www.food4rhino.com/en/app/ladybug-tools>
- Miller, N. (24. 04. 2023). *Lunchbox*. Abgerufen am 08. 05. 2023 von Food4Rhino: <https://www.food4rhino.com/en/app/lunchbox>
- Orduz, R., Solorzano, J., Egido, M., & Roman, E. (20. 11. 2011). Analytical study and evaluation results of power. Madrid, Spanien.
- Roberts, S., & Guariento, N. (2009). Gebäudeintegrierte Photovoltaik / Ein Handbuch. Basel: Birkhäuser Verlag AG. Abgerufen am 28. 04. 2023 von <https://doi.org/10.1007/978-3-0346-0489-5>
- Stadtrat Stadt Luzern. (2021). *Klima- und Energiestrategie Stadt Luzern*. Bericht und Antrag, Stadtrat Luzern, Luzern. Abgerufen am 24. 04. 2023 von https://www.stadtluzern.ch/_docn/3237703/BA_22_vom_30._Juni_2021.pdf
- Swissolar. (2019). Jahresbericht 2019. Abgerufen am 24. 04. 2024 von https://www.swissolar.ch/fileadmin/user_upload/Shop/SWS-Jahresbericht_2019.pdf
- Swissolar. (14. 07. 2022). Statistik Sonnenenergie Referenzjahr 2021. Bern. Abgerufen am 24. 04. 2023 von https://www.swissolar.ch/fileadmin/user_upload/Solarenergie/Fakten-und-Zahlen/220712_Statistik_Sonnenenergie_Bericht.pdf
- Theele, P. (07. 12. 2016). *Reihen- oder Parallelschaltung von Solarmodulen*. Abgerufen am 27. 04. 2023 von <https://www.photovoltaike4all.de/blog/reihen--oder-parallelschaltung-von-solarmodulen#:~:text=Die%20Verschaltung%20eines%20Modulfeldes%20hat,Modulen%20eine%20Stringspeisung%20von%20300V>.
- Weller, B., Hemmerle, C., Jakubetz, S., & Unnewehr, S. (2009). *Photovoltaik: Technik, Gestaltung, Konstruktion* (1. Ausg.). (DETAIL, Hrsg.) München. Abgerufen am 28. 04. 2023 von <https://doi.org/10.11129/detail.9783955530259>

Hochschule Luzern
Technik & Architektur

Bachelorarbeit
Im Studiengang
«Digital Construction in Architecture»

Anhang

Verfasser:

Marco Kohler

Matrikelnummer: 20-274-486

marco.kohler@stud.hslu.ch

Samuel Henseler

Matrikelnummer: 20-273-025

samuel.henseler@stud.hslu.ch

Inhalt

Anhang	3
A1 Fragebogen PV Fassaden Planung Basler & Hofmann	3
A2 Components Level 1	6
A3 Components Level 2	7
A4 Components Level 3	8
A5 Components Level 4	10
A6 Tutorial-Video Script Level 1	12
A7 Tutorial-Video Script Level 2	12
A8 Tutorial-Video Script Level 3	13
A9 Tutorial-Video Script Level 4	13
A10 Excel Berechnung Basler & Hofmann.....	14
A11 Datenblatt PV-Module	16
A12 Plan Ansichten Haus B, Basler & Hofmann.....	18
A13 Algorithmus String-Verschaltung.....	19
A14 Fitnessfaktor Kombination ROI und Ertrag.....	21
A15 Bake to Layer C# Skript.....	22
A16 Wechselrichter SMA Sunny Boy 3.0.....	28
A17 Wechselrichter Fronius Symo 10.....	29
A18 Wechselrichter SolarEdge Commercial Three Phase Inverter for the 277/480V Grid	31
A19 Wechselrichter IQ 7, IQ 7+, IQ 7A, IQ 7X	32
A20 Wirtschaftlichkeitsberechnung ohne MLPE (Level 3 und 4)	33
A21 Wirtschaftlichkeitsberechnung mit MLPE (Level 3 und 4).....	34
A22 Wirtschaftlichkeitsberechnung (Level 1 und 2)	35
A23 Ertragsberechnung jährlich.....	36
A24 Ertragsberechnung monatlich / stündlich	37
A25 Vorlage monatliche Berechnung, Elesawy Amr, B&H.....	39
A26 Algorithmus Optimierung mit Azimut Werten	40

Anhang

A1 Fragebogen PV Fassaden Planung Basler & Hofmann

Basler & Hofmann

Fragebogen PV Fassaden Planung

Personalien

Name: **Eric Langenskiöld**

Datum: **25.4.2023**

Offene Fragen

Frage 1: Wie ist Ihrer Meinung nach die Rolle der PV-Fassade in Zukunft einzuschätzen? In welchen Fällen eignen sich PV-Module an der Fassade? Was ist Ihr geschätztes Potenzial von PV-Fassadenanlagen?

Wie ist Ihrer Meinung nach die Rolle der PV-Fassade in Zukunft einzuschätzen?
> PV-Fassaden werden mit steigender Nachfrage nach erneuerbarem Strom an Bedeutung gewinnen. Die Stärke der PV-Fassade im Vergleich zum PV-Dach ist die Stromproduktion am Vor- und Nachmittag und im Winter

In welchen Fällen eignen sich PV-Module an der Fassade?
> Bei Neubauten, da so die Kosten einer anderen Fassade eingespart werden können. Südfassade bringen doppelt so viel wie Nordfassaden. Doch oft wird die ganze Fassade mit PV Belegt, um ein homogenes Erscheinungsbild zu erhalten.

Was ist Ihr geschätztes Potenzial von PV-Fassadenanlagen?
> Das Marktpotential für PV Fassaden würde ich ganz grob auf 30% der Neubauten schätzen.

Frage 2: Welchen Einfluss haben verschattete Module auf die einzelnen Strings und den Gesamtertrag?

> Dazu verweise ich auf die Tabelle, die ich euch gestern zugestellt habe und dessen Werte ich noch liefern werde

Umsatz:

Ertrag einbasse wegen Verschattung einzelner Stringmodule

maximale Einstrahlung auf String	% Module im String, die verschattet sind					
	100%	80%	60%	40%	20%	0%
1000 kWh/m ²	50%	0%
800 "	50%	0%
600 "	30%	0%
400 "	10%	0%
200 "	5%	0%

EIA 28.6.2022

Frage 3: Wie hat Basler & Hofmann bis anhin die PV-Anlagen geplant bezogen auf die String-Verschaltung und die Wirtschaftlichkeitsberechnung? (Strings ohne Optimizer)

Wir haben das den Ertrag des Modules mit den schlechtesten Jahreswert bezüglich Einstrahlung auf den gesamten String hochgerechnet.

Frage 4: Welche Daten und Planungstools werden in Ihrer Firma in der PV-Planung verwendet?

MeteoNorm
PV Syst

Frage 5: Was waren die grössten Probleme bei der PV-Planung seitens Basler & Hofmann?

Da war ein Projekt, 20 m hoch, bei dem ein Modul nach dem Rückbau des Gerüstes heruntergefallen ist. Der Kleber zwischen PV-Glas und PV-Alu-Rückhalteschiene hatte sich gelöst. Das Gerüst wurde unmittelbar wieder installiert zum Schutz der Gebäudebenutzer. Rechtsanwälte und Schadensexperten klärten Jahre, bis klar wurde, wer den Austausch aller Module an der Fassade zahlen muss. Gerüstkosten 20'000 CHF pro Monat + Ertragsausfall + Expertenkosten.....

Fazit: Es ist wichtig, als Planer die Qualitätssicherung der Klebeverbindung zu "planen", da dort bei Fehlern Personenschäden und hohe Sachschäden entstehen können.

Frage 6: Was sind aus Ihrer Sicht die Vor- und Nachteile von dem Einbau von elektrischen Optimizern?

Vorteil

1. Der Vorteil von Optimizern liegt in der einfachen Planung.
2. Es wird auch ein paar % mehr Strom generiert.
3. Das Monitoring der einzelnen Module ist möglich.
4. Gewisse Optimizer-Typen können im Brandfall den Strom abschalten.

Nachteil

1. PV Module sind solide, einfache Bauteile, welche bereits seit 1956 unter härtesten Bedingungen im Weltall auf Satelliten eingesetzt werden. Die Lebenserwartung schätzen wir auf über 40 Jahre. Der Nachteil der Optimizer liegt darin, das Optimizer Elektronikkomponenten sind. Von den Wechselrichtern, welche auch Elektronikkomponenten sind, haben wir die Erfahrung, dass in den ersten 15 Lebensjahren 50% davon ausfallen. Nun hat man pro PV-Modul ein Optimizer, und dieser ist in der Fassade und hinter dem Modul. Das Auswechseln bedingt 2 Spezialisten und ggf. eine Hebebühne.
Fazit: Die Unterhaltskosten nach dem Jahr 15 werden mit grosser Wahrscheinlichkeit stark ansteigen
Ein Beispiel: 2000 Module und 2000 Optimizer in einer Fassade. Jährliche Ausfallrate in einer Gaus-Verteilung mit dem Zenit bei 30 Jahren. Das sind dann 50 – 100 Optimierer, die jährlich irgendwo in der Fassade ersetzt werden müssen. Zu hoffen ist, dass der Hersteller nach 20 und 35 Jahren noch existiert im vergleichsweise dynamischen PV-Herstellermarkt
2. Weiter stellen die Elektronikkomponenten in der hinterlüfteten Fassade für die Feuerpolizei eine Brandlast und ein Entzündungsrisiko dar.
3. Die NIS-Strahlung (elektromagnetische Strahlung) von Elektronikgeräten in der Fassade könnte zu Problemen bei empfindlichen Personen oder bei strengen Richtwerten im Gebäude führen.

Frage 7: Was sind die grössten Probleme bei der Konfigurierung der Strings? Welcher Einfluss hat der Einstrahlungswinkel und der Faktor Zeit auf die einzelnen Strings?

Es dürfen nur Module mit gleicher Ausrichtung in einen String verkabelt werden. Dies führt zu einer grossen Anzahl erforderlicher Wechselrichter-MPP-Eingängen. Um die Anzahl MPPs zu erhalten, muss man mehrere kleine Wechselrichter mit möglichst vielen MPP-Eingängen installieren.

A2 Components Level 1

Total Components - 106

LunchBox:

...Workflow:

.....Rhino Command - 2

MetaHopper:

...Attributes:

.....Control Wire Display - 1

Params:

...Geometry:

.....Geometry - 1

.....Surface - 2

...Util:

.....Scribble - 7

...Primitive:

.....Integer - 1

.....File Path - 1

...Input:

.....Boolean Toggle - 3

.....Value List - 2

.....Panel - 59

.....Button - 3

Ladybug:

...2 :: Visualize Data:

.....LB Cumulative Sky Matrix - 1

...1 :: Analyze Data:

.....LB Analysis Period - 1

...0 :: Import:

.....LB Import EPW - 1

...3 :: Analyze Geometry:

.....LB Incident Radiation - 1

Curve:

...Primitive:

.....Rectangle - 4

Maths:

...Operators:

.....Multiplication - 2

.....Mass Addition - 1

.....Gate Not - 1

.....Division - 2

...Script:

.....GhPython Script - 1

.....VB Script - 1

OTools:

...3 | Debugging:

.....OTools_GetComponents2 - 2

Surface:

...Analysis:

.....Area - 1

Sets:

...List:

.....List Length - 2

...Tree:

.....Stream Filter - 3

A3 Components Level 2

Total Components - 153Number Slider - 6Subtraction - 2
LunchBox:	Ladybug:	...Script:
...Workflow:	...2 :: Visualize Data:OGetUsedComps_SDKMode - 1
.....Rhino Command - 2LB Cumulative Sky Matrix - 1GhPython Script - 3
MetaHopper:	...1 :: Analyze Data:VB Script - 1
...Attributes:LB Analysis Period - 1	Surface:
.....Control Wire Display - 1	...0 :: Import:	...Analysis:
Params:LB Import EPW - 1Evaluate Surface - 2
...Geometry:	...3 :: Analyze Geometry:Brep Edges - 1
.....Brep - 1LB Incident Radiation - 1	Sets:
.....Geometry - 5	Display:	...List:
...Util:	...Preview:List Length - 2
.....Scribble - 7Custom Preview - 2Split List - 1
.....Param Viewer - 1	...Dimensions:List Item - 2
.....Cluster - 1Text Tag - 2Item Index - 1
...Primitive:	Maths:	...Text:
.....Data - 2	...Util:Text Split - 4
.....File Path - 1Round - 1	...Tree:
...Input:	...Operators:Simplify Tree - 1
.....Value List - 2Smaller Than - 2Stream Filter - 6
.....Panel - 71Gate Not - 1Stream Gate - 1
.....Button - 3Division - 1Tree Branch - 2
.....Colour Swatch - 4		
.....Boolean Toggle - 3		

A4 Components Level 3

Total Components - 508File Path - 3	...Dimensions:
LunchBox:	...Input:Text Tag - 2
...Workflow:Value List - 7	Maths:
.....Rhino Command - 2Panel - 214	...Util:
MetaHopper:Button - 3Round - 1
...Attributes:Colour Swatch - 4Average - 13
.....Control Wire Display - 1Boolean Toggle - 4	...Operators:
Honeybee:Number Slider - 13Division - 5
...0 :: Create:	Ladybug:Mass Addition - 2
.....HB Shade - 1	...2 :: Visualize Data:Gate Not - 1
.....HB Model - 1LB Cumulative Sky Matrix - 15Subtraction - 4
.....HB Face - 1	...3 :: Analyze Geometry:Multiplication - 1
.....HB Room - 1LB Real Time Incident Radiation - 12Smaller Than - 2
Params:LB Incident Radiation - 3	...Script:
...Geometry:	...1 :: Analyze Data:OGetUsedComps_SDKMode - 1
.....Brep - 7LB Analysis Period - 2GhPython Script - 30
.....Geometry - 7LB Deconstruct Data - 2VB Script - 1
.....Mesh - 1	...0 :: Import:	HB-Radiance:
.....BRep - 1LB Import EPW - 3	...2 :: Light Sources:
...Util:	...4 :: Extra:HB Wea From EPW - 1
.....Sketch - 1LB Spatial Heatmap - 12	...4 :: Results:
.....Param Viewer - 1LB Generate Point Grid - 1HB Annual Results to Data - 1
.....Cluster - 2	Display:	...0 :: Basic Properties:
.....Scribble - 40	...Preview:HB Sensor Grid - 1
.....Round To Increments - 2Custom Preview - 2HB Assign Grids and Views - 1
.....Galapagos - 2		...3 :: Recipes:
...Primitive:		
.....Data - 9		

.....HB Annual Irradi-
ance - 1

Surface:

...Analysis:

.....Evaluate Surface - 2

.....Area - 1

.....Brep Edges - 1

Sets:

...List:

.....List Length - 7

.....Split List - 1

.....List Item - 2

.....Item Index - 3

...Text:

.....Text Split - 4

...Tree:

.....Merge - 1

.....Tree Branch - 4

.....Simplify Tree - 2

.....Flip Matrix - 1

.....Stream Filter - 29

.....Flatten Tree - 1

.....Stream Gate - 1

A5 Components Level 4

Total Components - 631Surface - 5Deconstruct - 2
Sasquatch:Geometry - 9	Mesh:
...Brep:	...Util:	...Analysis:
.....Surface Border - 1Sketch - 1Face Normals - 1
LunchBox:Param Viewer - 1	Ladybug:
...Workflow:Cluster - 3	...2 :: Visualize Data:
.....Layer Reference - 1Scribble - 42LB Cumulative Sky Matrix - 15
.....Rhino Command - 2Round To Increments - 2	...3 :: Analyze Geometry:
Human:Galapagos - 2LB Real Time Incident Radiation - 12
...Document Modifications:	...Primitive:LB Incident Radiation - 3
.....Create/Modify Layers - 1Data - 9	...1 :: Analyze Data:
...Misc:File Path - 3LB Analysis Period - 2
.....DeleteOnLayer - 1	...Input:LB Deconstruct Data - 2
MetaHopper:Value List - 7	...0 :: Import:
...Attributes:Panel - 238LB Import EPW - 3
.....Control Wire Display - 1Button - 4	...4 :: Extra:
Honeybee:Colour Swatch - 4LB Spatial Heatmap - 12
...0 :: Create:Boolean Toggle - 5LB Generate Point Grid - 1
.....HB Shade - 1Number Slider - 16	Display:
.....HB Model - 1	Vector:	...Preview:
.....HB Face - 1	...Plane:Custom Preview - 2
.....HB Room - 1Plane Normal - 1	...Dimensions:
Params:Align Plane - 1Text Tag - 2
...Geometry:	...Grid:	Curve:
.....BRep - 1Rectangular - 1	...Util:
.....Mesh - 1	...Vector:	
.....Curve - 2Unit Z - 1	
.....Brep - 7Vector XYZ - 1	
	...Point:	

.....Offset Curve - 3Division - 9Flip - 1
.....Join Curves - 1Mass Addition - 3	...Freeform:
...Primitive:Addition - 4Extrude - 1
.....Line - 1Negative - 2	...Analysis:
.....Rectangle - 1Gate Not - 1Evaluate Surface - 3
...Analysis:Subtraction - 5Deconstruct Brep - 1
.....Length - 2Larger Than - 1Point In Brep - 1
.....Point in Curves - 1Multiplication - 1Dimensions - 1
.....Point In Curve - 1Smaller Than - 3Brep Edges - 2
Intersect:	Transform:Area - 7
...Region:	...Euclidean:	Sets:
.....Trim with Region - 1Rotate 3D - 1	...List:
Maths:Move - 1Sort List - 3
...Script:	HB-Radiance:Split List - 1
.....OGetUsedComps_SDKMode - 1	...2 :: Light Sources:List Length - 8
.....C# Script - 1HB Wea From EPW - 1Item Index - 3
.....GhPython Script - 32	...4 :: Results:List Item - 6
.....VB Script - 1HB Annual Results to Data - 1	...Text:
...Domain:	...0 :: Basic Properties:Text Split - 4
.....Bounds - 1HB Sensor Grid - 1	...Sequence:
.....Deconstruct Domain - 1HB Assign Grids and Views - 1Cull Pattern - 3
...Trig:	...3 :: Recipes:	...Tree:
.....Radians - 1HB Annual Irradiance - 1Merge - 1
...Util:	Surface:Tree Branch - 4
.....Round - 3	...Primitive:Simplify Tree - 2
.....Average - 13Bounding Box - 2Flip Matrix - 2
...Operators:	...Util:Stream Filter - 33
	Flatten Tree - 5
	Stream Gate - 1

A6 Tutorial-Video Script Level 1



https://youtu.be/8bjdC_AzMX0

A7 Tutorial-Video Script Level 2



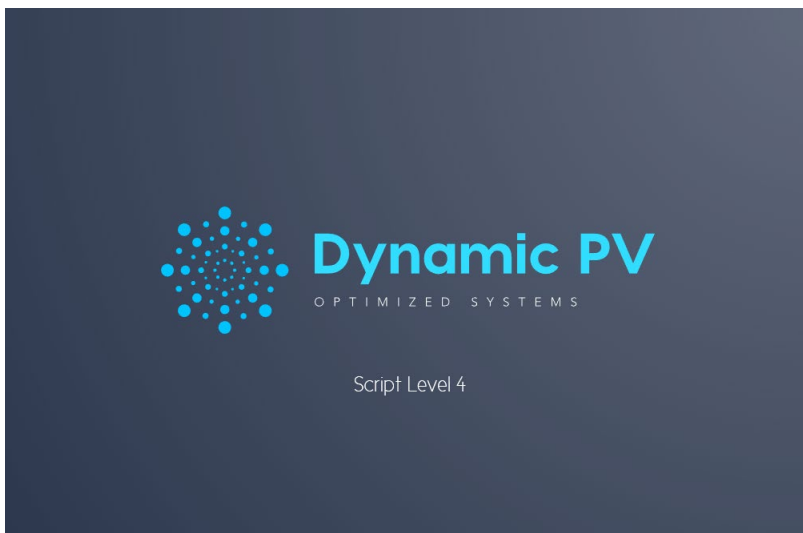
https://youtu.be/w_OQteuwQgc

A8 Tutorial-Video Script Level 3



<https://youtu.be/WizWroXRi04>

A9 Tutorial-Video Script Level 4



<https://youtu.be/P9TjfsmlETE>


Fassade				
		Haus A	Haus B	Total
Anzahl Module	(Stk)	311	311	622
Leistung	(kWp)	47	47	93
Ertrag*	(kWh/a)	14'000	11'000	25'000

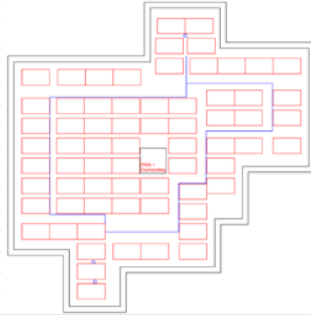
Dach				
		Haus A	Haus B	Total
Anzahl Module	(Stk)	68	68	136
Leistung	(kWp)	24	24	48
Ertrag	(kWh/a)	23'000	23'000	46'000
Dachrandbleche	kWp	4.2	4.2	8.4

Total				
		Areal	Pronovo	
Anzahl Module	(Stk)	758	Fassade	
Leistung	(kWp)	141	Dach	62
Ertrag	(kWh/a)	71'000	Dachrand	11
				185 Bild 1 Fassadensimulation

*) Fassadenmodule ohne Optimizer.
 Mit Optimizer wäre der Ertrag um 1'000 kWh/a höher
 Zusatzkosten Optimizer ca. 70 CHF/ Modul total 40'000 CHF + 300 CHF / Jahr für Unterhalt
 dem stehen 6'000 CHF Mehreinnahmen gegenüber (bei 30 Jahr zu 20 Rp/kWh)

Max Stromstärke und Abwärme				
Dach und Südfassade	79 kWp	100%	79 kW DC	
Andere Fassaden	70 kWp	13.5%	9 kW DC	
			89 kW DC	
			85% DC/AC	
		max AC Leistung	75 kVA	
		max Strom	109 A	
		max Abwärme	1.5 kW	





A11 Datenblatt PV-Module



MegaSlate® II Fassade

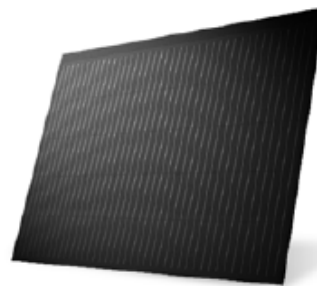
Das MegaSlate Glas-Glas-Modul für die Fassade

Entwickelt & produziert in Thun

Massgeschneidert für jede Fassade

Höchster Stromertrag

Ästhetisch & elegant



Optimal für Fassadeneindeckung

Das beliebte MegaSlate Fassadenmodul bringt dank der Glas-Glas-Ausführung zusätzliche Vorteile in den Bereichen Brandschutz und Resttragfähigkeit mit. Das Schweizer Qualitätsprodukt wird in Thun mit CO₂-freiem Strom produziert und besticht durch seine Eleganz und Ästhetik. Die vier kombinierbaren Modulgrößen des MegaSlate-Systems ermöglichen eine maximale Nutzung der Fassadenfläche. Der Systemaufbau gewährleistet durch eine optimale Hinterlüftung an jedem einzelnen Modul maximale Energieerträge.



MegaSlate®-System

Das MegaSlate-System gewinnt effizient Energie aus Sonnenlicht und verwandelt Ihr individuelles Gebäude in ein Plusenergiebau der Extraklasse. Das System bietet für Dach und Fassade Flexibilität und Integration auf höchstem technologischen und ästhetischen Niveau.

Mit Hilfe des MegaSlate-Systems machen Sie einen Schritt in Richtung Zukunft und leisten Ihren persönlichen Beitrag zur Energiewende.

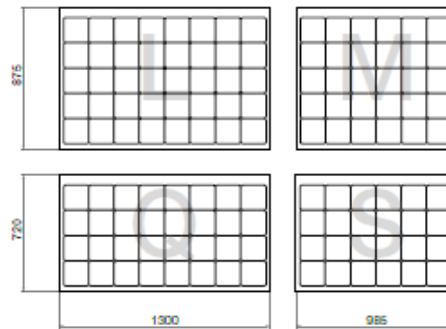


Elektrische Spezifikationen

	L	Q	M	S
Nennleistung	200 Wp	180 Wp	150 Wp	120 Wp
Spannung U_{mp}	22,2 V	17,7 V	16,6 V	13,3 V
Strom I_{mp}	9,0 A			
Leerlaufspannung U_{oc}	27,1 V	21,8 V	20,3 V	18,3 V
Kurzschlussstrom I_{sc}	9,5 A			
Wirkungsgrad	18,6%	18,4%	18,5%	18,2%
Maximale Systemspannung	1000 V			
Rückstrombelastbarkeit	18 A			
Toleranz Nennleistung	±3%			±5%

Toleranz Strom I_{sc} und Kurzschlussstrom I_{sc} beträgt ±3%
 Elektrische Leistungsdaten bei STC (1000 W/m², 25 °C, AM 1.5).
 Messprotokolle auf Anfrage erhältlich.

Abmessungen Module



Mechanische Spezifikationen

	L	Q	M	S
Anzahl Zellen	8 x 5 = 40	8 x 4 = 32	6 x 5 = 30	6 x 4 = 24
Dimensionen	1300 x 875 x 9 mm	1300 x 720 x 9 mm	985 x 875 x 9 mm	985 x 720 x 9 mm
Sichtbare Fläche	1300 x 825 mm	1300 x 670 mm	985 x 825 mm	985 x 670 mm
Gewicht (ca.)	22.8 kg	19 kg	17.1 kg	14.2 kg
Glasstärke	4 mm TVG Solarglas + 4 mm TVG Solarglas			
Zellentyp	156.75 x 156.75 mm monokristallin SBB PERC			
Anschlussdose	OC Junction Box (Dezentral)			
Anschlusskabel	PV Cable 4mm ² , Länge je 1 m			
Stackverbinder	MC4 original (PV-KS(E)T4/BI-UR, Stäubli International AG)			
Maximale Belastung*	2400 N/m ² (Druck)			
	1600 N/m ² (Sog bei 3 Haken)			
	1900 N/m ² (Sog bei 5 Haken)			

*Geplante Druck- und Soglast für möglichste Sicherheitsgaranz. Festlegungsgültig nachgewiesen nach DIN 18008.

Garantien und Normen

Angewandte Normen	IEC 61730:2016, IEC 61215:2016
Produktgarantie	10 Jahre
Leistungsgarantie	1 Jahre auf 97% der Mindestleistung
	25 Jahre auf 80% der Mindestleistung
Witterungsbeständigkeitsgarantie	40 Jahre
SENS of Recycling	Inkl. vorgezogene Recyclinggebühr

Temperaturkoeffizienten

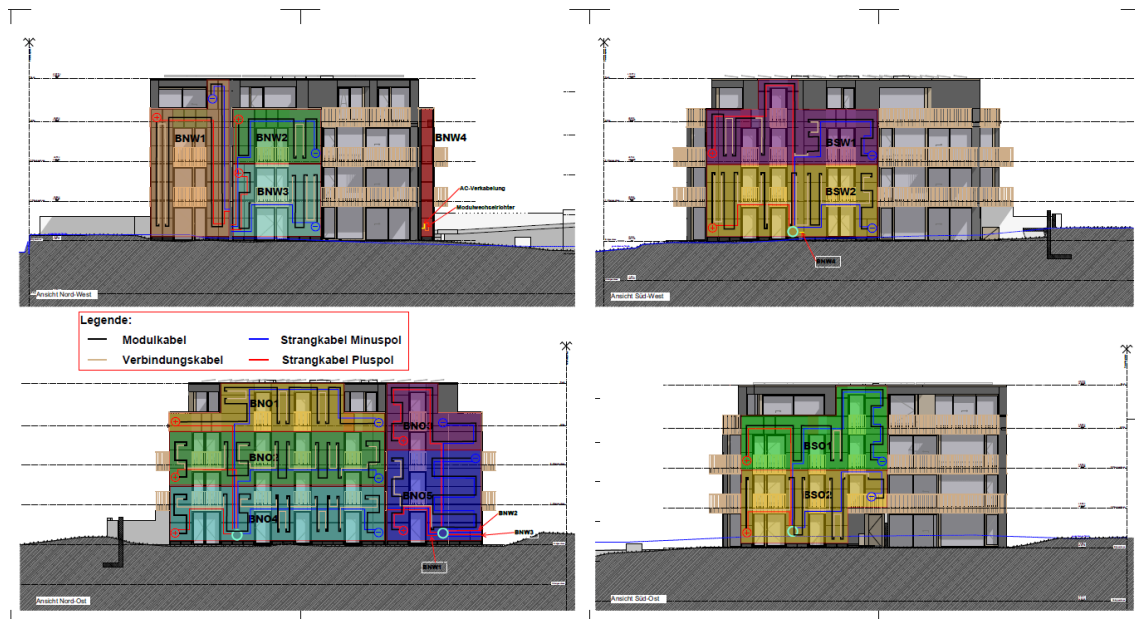
$\alpha (I_{sc})$	+0.045 %/K
$\beta (U_{mp})$	-0.276 %/K
$\gamma (P_{mp})$	-0.371 %/K



Ihr Installationspartner:

3S Swiss Solar Solutions AG
 Schorenstrasse 39
 CH-3645 Gwatt (Thun)
 +41 33 224 25 00
 www.3s-solar.swiss
 info@3s-solar.swiss

A12 Plan Ansichten Haus B, Basler & Hofmann



A13 Algorithmus String-Verschaltung

```

import rhinoscriptsyntax as rs
import ghpythonlib.treehelpers as th
import math
import ast

modules_inf = modules_inf
modules = eval(modules_list[0])

voltage_modules = float(voltage)
voltage_max = float(max_voltage)
max_string = int(math.floor(voltage_max / voltage_modules))

group_ls = []
for i in range(len(modules_inf)):
    if modules_inf[i][2] <= float(Distanz) and modules_inf[i][3] <= yield_difference:
        group_ls.append([modules_inf[i][0], modules_inf[i][1]])

group_old = [tuple(sublist) for sublist in group_ls]

connections = [(str(x), str(y)) for x, y in group_old]

groups = []

# Liste der bereits verwendeten Module
assigned_modules = set()

# Iterieren über jede Kombination von Modulen
for i, module_i in enumerate(modules):
    for j, module_j in enumerate(modules[i + 1:], i + 1):
        # Überprüfen, ob mindestens eine Verbindung besteht
        if (module_i, module_j) in connections or (module_j, module_i) in connections:
            # Gruppenzuweisung
            group_found = False
            for group in groups:
                if module_i in group or module_j in group:
                    if len(group) < max_string:
                        # Überprüfen, ob das Modul bereits zugewiesen wurde
                        if module_i not in assigned_modules:
                            group.add(module_i)
                            assigned_modules.add(module_i)
                        if module_j not in assigned_modules:
                            group.add(module_j)
                            assigned_modules.add(module_j)
                        group_found = True
                        break
            if not group_found:
                groups.append({module_i, module_j})
                assigned_modules.add(module_i)
                assigned_modules.add(module_j)

# Entfernen der Module aus den Gruppen, die bereits in anderen Gruppen enthalten sind
for i, group in enumerate(groups):
    # Erstellen eines Sets aller Module in anderen Gruppen
    other_modules = set()
    for j, other_group in enumerate(groups):
        if j != i:
            other_modules.update(other_group)
    # Entfernen der gemeinsamen Module
    group.difference_update(other_modules)

```

```
# Entfernen der Module aus den zugewiesenen Modulen, die nicht in Gruppen enthalten
sind
assigned_modules.intersection_update(set.union(*groups))

#Blindmodule#####

blind_modules = []
for module in modules:
    if module not in assigned_modules:
        blind_modules.append(module)
    if any({module} == group or len(group) == 2 and module in group for group in
groups):
        blind_modules.append(module)

# Entfernen der Gruppen, die Blindmodule enthalten
for i, group in enumerate(groups):
    if any(module in blind_modules for module in group):
        groups[i] = set()

# Entfernen der leeren Gruppen
groups = [group for group in groups if group]

# Entfernen der Module aus den zugewiesenen Modulen, die nicht in Gruppen enthalten
sind
assigned_modules.intersection_update(set.union(*groups))

# Ausgabe der Gruppen ohne diejenigen, die Blindmodule enthalten sind
result = ""
for i, gm in enumerate(groups):
    extra = ('Gruppe {}: {}'.format(i, [int(x) for x in gm]))
    result += extra
    print(extra)

# Ausgabe der Anzahl von Strings
strings = str(len(groups))
print("Anzahl Strings: ", strings)

# Ausgabe der Liste von Blindmodulen ohne Gruppen
blind_modules = [module for module in blind_modules if not any(module in group for
group in groups)]
print("Blindmodule: ", blind_modules)

#Ausgabe von verwendeten Modulen
modules_verw = ("Von ", len(modules), "sind ", len(assigned_modules), "Module in Ver-
wendung")
print(modules_verw)
```

A14 Fitnessfaktor Kombination ROI und Ertrag

```
calc = 100 - (100 / x * y)
ertrag = str(round(calc))
```

```
fitness_faktor = ((rendite-1)/10+1)*((ertrag-1)/20+1)
```

A15 Bake to Layer C# Skript

Abgerufen am 14.05.2023 von:

<https://www.giuliopiacentino.com/grasshopper-tools/>

```
using System;
using System.Collections;
using System.Collections.Generic;

using Rhino;
using Rhino.Geometry;

using Grasshopper;
using Grasshopper.Kernel;
using Grasshopper.Kernel.Data;
using Grasshopper.Kernel.Types;

using System.IO;
using System.Linq;
using System.Data;
using System.Drawing;
using System.Reflection;
using System.Windows.Forms;
using System.Xml;
using System.Xml.Linq;
using System.Runtime.InteropServices;

using Rhino.DocObjects;
using Rhino.Collections;
using GH_IO;
using GH_IO.Serialization;

/// <summary>
/// This class will be instantiated on demand by the Script component.
/// </summary>
public class Script_Instance : GH_ScriptInstance
{
    #region Utility functions
    /// <summary>Print a String to the [Out] Parameter of the Script component.</summary>
    /// <param name="text">String to print.</param>
    private void Print(string text) { /* Implementation hidden. */ }
    /// <summary>Print a formatted String to the [Out] Parameter of the Script component.</summary>
    /// <param name="format">String format.</param>
    /// <param name="args">Formatting parameters.</param>
    private void Print(string format, params object[] args) { /* Implementation hidden. */ }
    /// <summary>Print useful information about an object instance to the [Out] Parameter of the Script component. </summary>
    /// <param name="obj">Object instance to parse.</param>
    private void Reflect(object obj) { /* Implementation hidden. */ }
    /// <summary>Print the signatures of all the overloads of a specific method to the [Out] Parameter of the Script component. </summary>
    /// <param name="obj">Object instance to parse.</param>
    private void Reflect(object obj, string method_name) { /* Implementation hidden. */ }
}

#region Members
/// <summary>Gets the current Rhino document.</summary>
private readonly RhinoDoc RhinoDocument;
/// <summary>Gets the Grasshopper document that owns this script.</summary>
private readonly GH_Document GrasshopperDocument;
/// <summary>Gets the Grasshopper script component that owns this script.</summary>
private readonly IGH_Component Component;
```

```

    /// <summary>
    /// Gets the current iteration count. The first call to RunScript() is associated
    with Iteration==0.
    /// </summary>
    private readonly int Iteration;
#endregion

    /// <summary>
    /// This procedure contains the user code. Input parameters are provided as regular
    arguments,
    /// Output parameters as ref arguments. You don't have to assign output parameters,
    /// they will have a default value.
    /// </summary>
    private void RunScript(List<System.Object> objs, List<string> names, List<string>
    layers, List<System.Drawing.Color> colors, List<double> pWidths, List<int> wiresEach,
    List<System.Object> materials, bool activate, bool groupListTgthr)
    {

        //Inserts geometry into the Rhino document, with custom attributes
        //Written by Giulio Piacentino
        //Version written 2011 10 18 - for Grasshopper 0.8.52

        if (activate)
        {
            int groupIndex = -1;
            if(groupListTgthr)
                groupIndex = doc.Groups.Add();

            for(int i = 0; i < objs.Count; i++)
            {
                object obj = objs[i];
                if(obj == null)
                {
                    Print("No object to bake");
                    return;
                }

                //Make new attribute to set name
                Rhino.DocObjects.ObjectAttributes att = new Rhi-
                no.DocObjects.ObjectAttributes();

                string name = names.Count > 0 ? names[i % names.Count] : null;
                //Set object name
                if(!string.IsNullOrEmpty(name))
                {
                    att.Name = name;
                }

                Color color = colors.Count > 0 ? colors[i % colors.Count] : new Color();
                //Set color
                if(!color.IsEmpty)
                {
                    att.ColorSource = Rhino.DocObjects.ObjectColorSource.ColorFromObject;
                    //Make the color type "by object"
                    att.ObjectColor = color;

                    att.PlotColorSource = Rhi-
                    no.DocObjects.ObjectPlotColorSource.PlotColorFromObject; //Make the plot color type
                    "by object"
                    att.PlotColor = color;
                }

                string layer = layers.Count > 0 ? layers[i % layers.Count] : null;
                //Set layer
                if(!string.IsNullOrEmpty(layer) && Rhino.DocObjects.Layer.IsValidName(layer))
                {
                    //Get the current layer index

```

```

Rhino.DocObjects.Tables.LayerTable layerTable = doc.Layers;
int layerIndex = layerTable.Find(layer, true);

if(layerIndex < 0) //This layer does not exist, we add it
{
    Rhino.DocObjects.Layer onlayer = new Rhino.DocObjects.Layer(); //Make a
new layer
    onlayer.Name = layer;

    layerIndex = layerTable.Add(onlayer); //Add the layer to the layer table
    if (layerIndex > -1) //We managed to add layer!
    {
        att.LayerIndex = layerIndex;
        Print("Added new layer to the document at position " + layerIndex + "
named " + layer + ". ");
    } else
        Print("Layer did not add. Try cleaning up your layers."); //This never
happened to me.
    }
    else
        att.LayerIndex = layerIndex; //We simply add to the existing layer
}

double pWidth = pWidths.Count > 0 ? pWidths[i % pWidths.Count] : 0;
//Set plotweight
if(pWidth > 0)
{
    att.PlotWeightSource = Rhi-
no.DocObjects.ObjectPlotWeightSource.PlotWeightFromObject;
    att.PlotWeight = pWidth;
}

object material = materials.Count > 0 ? materials[i % materials.Count] :
null;
//Set material
bool materialByName = !string.IsNullOrEmpty(material as string);
Rhino.Display.DisplayMaterial inMaterial = material as Rhi-
no.Display.DisplayMaterial;
if(material is Color)
{
    inMaterial = new Rhino.Display.DisplayMaterial((Color) material);
}
if(material != null && inMaterial == null && !materialByName)
{
    if(!(material is string))
    {
        try //We also resort to try with IConvertible
        {
            inMaterial = (Rhino.Display.DisplayMaterial) Con-
vert.ChangeType(material, typeof(Rhino.Display.DisplayMaterial));
        }
        catch(InvalidCastException)
        {
        }
    }
}
if(inMaterial != null || materialByName)
{
    string matName;

    if(!materialByName)
    {
        matName = string.Format("A:{0}-D:{1}-E:{2}-S:{3},{4}-T:{5}",
            Format(inMaterial.Ambient),
            Format(inMaterial.Diffuse),

```

```

        Format(inMaterial.Emission),
        Format(inMaterial.Specular),
        inMaterial.Shine.ToString(),
        inMaterial.Transparency.ToString()
    );
}
else
{
    matName = (string) material;
}

int materialIndex = doc.Materials.Find(matName, true);
if(materialIndex < 0 && !materialByName) //Material does not exist and we
have its specs
{
    materialIndex = doc.Materials.Add(); //Let's add it
    if(materialIndex > -1)
    {
        Print("Added new material at position " + materialIndex + " named \"" +
matName + "\". ");
        Rhino.DocObjects.Material m = doc.Materials[materialIndex];
        m.Name = matName;
        m.AmbientColor = inMaterial.Ambient;
        m.DiffuseColor = inMaterial.Diffuse;
        m.EmissionColor = inMaterial.Emission;
        //m.ReflectionColor = no equivalent
        m.SpecularColor = inMaterial.Specular;
        m.Shine = inMaterial.Shine;
        m.Transparency = inMaterial.Transparency;
        //m.TransparentColor = no equivalent
        m.CommitChanges();

        att.MaterialSource = Rhi-
no.DocObjects.ObjectMaterialSource.MaterialFromObject;
        att.MaterialIndex = materialIndex;
    }
    else
        Print("Material did not add. Try cleaning up your materials."); //This
never happened to me.
}
else if(materialIndex < 0 && materialByName) //Material does not exist and
we do not have its specs. We do nothing
{
    Print("Warning: material name not found. I cannot set the source to this
material name. Add a material with name: " + matName);
}
else
{
    //If this material exists, we do not replace it!
    att.MaterialSource = Rhi-
no.DocObjects.ObjectMaterialSource.MaterialFromObject;
    att.MaterialIndex = materialIndex;
}
}

int wires = wiresEach.Count > 0 ? wiresEach[i % wiresEach.Count] : 0;
//Set wire density
if(wires == -1 || wires > 0)
{
    att.WireDensity = wires;
}

Bake(obj, att, groupIndex);
}
}
else

```

```

    {
        Print("Inactive");
    }
}

// <Custom additional code>

void Bake(object obj, Rhino.DocObjects.ObjectAttributes att, int groupIndex)
{
    if(obj == null)
        return;

    Guid id;

    //Bake to the right type of object
    if(obj is GeometryBase)
    {
        GeometryBase geomObj = obj as GeometryBase;

        switch(geomObj.ObjectType)
        {
            case Rhino.DocObjects.ObjectType.Brep:
                id = doc.Objects.AddBrep(obj as Brep, att);
                break;
            case Rhino.DocObjects.ObjectType.Curve:
                id = doc.Objects.AddCurve(obj as Curve, att);
                break;
            case Rhino.DocObjects.ObjectType.Point:
                id = doc.Objects.AddPoint((obj as Rhino.Geometry.Point).Location, att);
                break;
            case Rhino.DocObjects.ObjectType.Surface:
                id = doc.Objects.AddSurface(obj as Surface, att);
                break;
            case Rhino.DocObjects.ObjectType.Mesh:
                id = doc.Objects.AddMesh(obj as Mesh, att);
                break;
            case (Rhino.DocObjects.ObjectType)
1073741824://Rhino.DocObjects.ObjectType.Extrusion:
                id = (Guid)
typeof(Rhino.DocObjects.Tables.ObjectTable).InvokeMember("AddExtrusion", Bind-
ingFlags.Instance | BindingFlags.InvokeMethod, null, doc.Objects, new object[] {obj,
att});
                break;
            case Rhino.DocObjects.ObjectType.PointSet:
                id = doc.Objects.AddPointCloud(obj as Rhino.Geometry.PointCloud, att);
//This is a speculative entry
                break;
            default:
                Print("The script does not know how to handle this type of geometry: " +
obj.GetType().FullName);
                return;
        }
    }
    else
    {
        Type objectType = obj.GetType();

        if(objectType == typeof(Arc))
        {
            id = doc.Objects.AddArc((Arc) obj, att);
        }
        else if(objectType == typeof(Box))
        {
            id = doc.Objects.AddBrep(((Box) obj).ToBrep(), att);
        }
        else if(objectType == typeof(Circle))

```



```
{
    id = doc.Objects.AddCircle((Circle) obj, att);
}
else if(objectType == typeof(Ellipse))
{
    id = doc.Objects.AddEllipse((Ellipse) obj, att);
}
else if(objectType == typeof(Polyline))
{
    id = doc.Objects.AddPolyline((Polyline) obj, att);
}
else if(objectType == typeof(Sphere))
{
    id = doc.Objects.AddSphere((Sphere) obj, att);
}
else if(objectType == typeof(Point3d))
{
    id = doc.Objects.AddPoint((Point3d) obj, att);
}
else if(objectType == typeof(Line))
{
    id = doc.Objects.AddLine((Line) obj, att);
}
else if(objectType == typeof(Vector3d))
{
    Print("Impossible to bake vectors");
    return;
}
else if(obj is IEnumerable)
{
    int newGroupIndex;
    if(groupIndex == -1)
        newGroupIndex = doc.Groups.Add();
    else
        newGroupIndex = groupIndex;
    foreach(object o in obj as IEnumerable)
    {
        Bake(o, att, newGroupIndex);
    }
    return;
}
else
{
    Print("Unhandled type: " + objectType.FullName);
    return;
}
}

if(groupIndex != -1)
{
    doc.Groups.AddToGroup(groupIndex, id);
    Print("Added " + obj.GetType().Name + " to group number " + groupIndex);
}
else
    Print("Added " + obj.GetType().Name);
}

static string Format(Color c)
{
    return (new System.Text.StringBuilder("A")).Append(c.A.ToString()).Append("R").Append(c.R.ToString())
        .Append("G")
        .Append(c.G.ToString()).Append("B").Append(c.B.ToString()).ToString();
}

// </Custom additional code>
}
```

A16 Wechselrichter SMA Sunny Boy 3.0

11.04.23, 22:55

Sunny Boy 3.0 / 3.6 / 4.0

Sunny Boy 3.0 / 3.6 / 4.0

DC-Eingang

	SB3.0-1AV-41	SB3.6-1AV-41	SB4.0-1AV-41
Maximale Leistung der PV-Module	5500 Wp	5500 Wp	7500 Wp
Maximale Eingangsspannung	600 V	600 V	600 V
MPP-Spannungsbereich	125 V bis 500 V	125 V bis 500 V	125 V bis 500 V
Bemessungseingangsspannung	365 V	365 V	365 V
Minimale Eingangsspannung	100 V	100 V	100 V
Start-Eingangsspannung	125 V	125 V	125 V
Maximaler Eingangsstrom, Eingang A	15 A	15 A	15 A
Maximaler Eingangsstrom, Eingang B	15 A	15 A	15 A
Maximaler Eingangsstrom pro String, Eingang A	15 A	15 A	15 A
Maximaler Eingangsstrom pro String, Eingang B	15 A	15 A	15 A
Maximaler Kurzschluss-Strom, Eingang A (> Sunny Boy 3.0 / 3.6 / 4.0) (10055043467.html)	20 A	20 A	20 A
Maximaler Kurzschluss-Strom, Eingang B (> Sunny Boy 3.0 / 3.6 / 4.0) (10055043467.html)	20 A	20 A	20 A
Maximaler Rückstrom in die PV-Module	0 A	0 A	0 A
Anzahl der unabhängigen MPP-Eingänge	2	2	2
Strings pro MPP-Eingang	2	2	2

A17 Wechselrichter Fronius Symo 10

11.04.23, 22:55

Fronius Symo 10.0-3-M

FRONIUS SYMO


MAXIMALE FLEXIBILITÄT FÜR DIE ANWENDUNGEN VON MORGEN.

Mit seinen Leistungsklassen von 3,0 bis 20,0 kW ist der trafolose Fronius Symo der dreiphasige Wechselrichter für jede Anlagengröße. Dank dem SuperFlex Design ist der Fronius Symo optimal für verwinkelte oder unterschiedlich orientierte Dächer. Die serienmäßige Anbindung an das Internet per WLAN oder Ethernet sowie die einfache Integration von Drittanbieter-Komponenten machen den Fronius Symo zu einem der kommunikativsten Wechselrichter am Markt. Zudem ermöglicht die Zählerschnittstelle ein dynamisches Einspeisemanagement und eine übersichtliche Verbrauchsvisualisierung.

Um mehr zu erfahren, wählen Sie bitte ein Produkt aus.



PRODUKTDDETAILS

TECHNISCHE DATEN	
 SOLAR ENERGY <small>INSTALLATEURE & PARTNER</small>	
EINGANGSDATEN 🔍 ☰	
Anzahl MPP-Tracker	2
Max. Eingangsstrom ($I_{dc\ max}$)	27,0 / 16,5 A
Max. Kurzschlussstrom Modulfeld*	56,0 / 34,0 A
DC Eingangsspannungsbereich ($U_{dc\ min} - U_{dc\ max}$)	200 - 1000 V
Einspeisung Startspannung ($U_{dc\ start}$)	200 V
Nominale Eingangsspannung (U_{dc})	600 V
MPP-Spannungsbereich ($U_{mpp\ min} - U_{mpp\ max}$)	270 - 800 V
Nutzbarer MPP-Spannungsbereich	200 - 800 V
Anzahl DC-Anschlüsse	3 + 3
Max. PV-Generatorleistung ($P_{dc\ max}$)	15 kWpeak

11.04.23, 22:55

Fronius Symo 10.0-3-M

AUSGANGSDATEN

AC-Nennleistung ($P_{ac,n}$)	10 kW
Max. Ausgangsleistung ($P_{ac,max}$)	10 kVA
AC-Ausgangsstrom ($I_{ac,nom}$)	14,4 A
Netzanschluss ($U_{ac,n}$)	3~ NPE 400/230, 3~ NPE 380/220 V
AC Spannungsbereich ($U_{min} - U_{max}$)	150 - 280 V
Frequenz (f_n)	50 / 60 Hz
Frequenzbereich ($f_{min} - f_{max}$)	45 - 65 Hz
Klirrfaktor	<3% @Pnom (230/400VAC 50Hz)
Leistungsfaktor ($\cos \varphi_{ac,n}$)	0 - 1 Ind./cap,

ALLGEMEINE DATEN

Abmessung / Breite	512 mm
Abmessung / Höhe	751 mm
Abmessung / Tiefe	227,5 mm
Gewicht	35,8 kg
Gewicht	30,77 kg
Gewicht inkl. Verpackung	39,97 kg
Schutzart	IP 66
Schutzklasse	1
Überspannungskategorie (DC / AC) ¹⁾	2 / 3
Nachtverbrauch	<1 W
Wechselrichterkonzept	Trafoles
Kühlung	Geregelte Luftkühlung
Montage	Innen- und Außenmontage
Umgebungstemperatur-Bereich	-40°C - +60°C
Zulässige Luftfeuchtigkeit	0 - 100 %
Max. Höhe über dem Meeresspiegel ²⁾	2,000 m / 3,400 m
Anschlusstechnologie DC	6x DC+ und 6x DC- Schraubklemmen 2,5 - 16 mm ²
Anschlusstechnologie AC	5 polige AC Schraubklemmen 2,5 - 16 mm ²
Zertifikate und Normerfüllung	ÖVE / ÖNORM E 8001-4-712, DIN V VDE 0126-1-1/A1, VDE AR N 4105, IEC 62109-1/-2, IEC 62116, IEC 61727, AS 3100, AS 4777-2, AS 4777-3, CER 06-190, G83/2, UNE 206007-1, SI 4777, CEI 0-16, CEI 0-21, NRS 097
Störemissionsklasse	B

WIRKUNGSGRAD

Max. Wirkungsgrad (PV - Stromnetz)	98 %
Europäischer Wirkungsgrad (η_{EU})	97,4 %
η bei 5 % $P_{ac,r}$ ³⁾	87,9 / 92,5 / 89,2 %
η bei 10 % $P_{ac,r}$ ³⁾	91,2 / 94,9 / 92,8 %
η bei 20 % $P_{ac,r}$ ³⁾	94,6 / 97,1 / 96,1 %
η bei 25 % $P_{ac,r}$ ³⁾	95,4 / 97,3 / 96,6 %

A18 Wechselrichter SolarEdge Commercial Three Phase Inverter for the 277/480V Grid

solar**edge** SolarEdge Commercial Three Phase Inverters for the 277/480V Grid for North America SE66.6K-SE100K

	SE66.6K	SE100K	
OUTPUT			
Rated AC Power Output	66600	100000	VA
Maximum AC Power Output	66600	100000	VA
AC Output Line Connections	4 wire WYE (L1-L2-L3-N) plus PE		
AC Output Voltage Minimum-Nominal-Maximum ⁽¹⁾ (L-N)	244 - 277 - 305		Vac
AC Output Voltage Minimum-Nominal-Maximum ⁽¹⁾ (L-L)	422.5 - 480 - 529		Vac
AC Frequency Min-Nom-Max ⁽¹⁾	59.3 - 60 - 60.5		Hz
Maximum Continuous Output Current (per Phase) @277V	80	120	A
GFDI Threshold	1		
Utility Monitoring, Islanding Protection, Configurable Power Factor, Country Configurable Thresholds	Yes		
INPUT			
Maximum DC Power (Module STC)	90000 / 45000	135000 / 45000	W
Transformer-less, Ungrounded	Yes		
Maximum Input Voltage DC to Gnd	500		Vdc
Maximum Input Voltage DC+ to DC-	1000		Vdc
Nominal Input Voltage DC to Gnd	425		Vdc
Nominal Input Voltage DC+ to DC-	850		Vdc
Maximum Input Current	80	120	Adc
Maximum Input Short Circuit Current	120		Adc
Reverse-Polarity Protection	Yes		
Ground-Fault Isolation Detection	350kΩ Sensitivity per Unit		
CEC Weighted Efficiency	98.5		%
Nighttime Power Consumption	< 12		W
ADDITIONAL FEATURES			
Supported Communication Interfaces	RS485, Ethernet, Cellular GSM (optional)		
Rapid Shutdown	NEC2014 and NEC2017 compliant/certified, upon AC Grid Disconnect		
RS485 Surge Protection	Built-in		
DC SAFETY SWITCH			
DC Disconnect	1000V / 2 x 40A	1000V / 3 x 40A	
DC Surge Protection	Optional, Type II, field replaceable		
DC Fuses on Plus & Minus	Optional, 30A		
STANDARD COMPLIANCE⁽²⁾			
Safety	UL1741, UL1741 SA, UL1699B, UL1998, CSA 2.22		
Grid Connection Standards	IEEE 1547, Rule 21, Rule 14 (H)		
Emissions	FCC part15 class A		
INSTALLATION SPECIFICATIONS			
Number of units	2	3	
AC Output Conduit Size / Max AWG / Max PE AWG	1.5" / 2/0 / 6	2" / 4/0 / 4	
DC Output Conduit Size / Terminal Block AWG Range / Number of Strings ⁽³⁾	2 x 1.25" / 6-14 / 6 strings	2 x 1.25" / 6-14 / 9 strings	
Dimensions (H x W x D)	Primary Unit: 37 x 12.5 x 10.5 / 940 x 315 x 260; Secondary Unit: 21 x 12.5 x 10.5 / 540 x 315 x 260		in / mm
Weight	Primary Unit: 105.8 / 48; Secondary Unit: 99.2 / 45		lb / kg
Operating Temperature Range	-13 to +140 / -25 to +60 ⁽⁴⁾ (-40°F / -40°C option)		°F / °C
Cooling	Fan (user replaceable)		
Noise	< 60		dB(A)
Protection Rating	NEMA 3R		
Bracket Mounted (Brackets Provided)			

⁽¹⁾ For other regional settings please contact SolarEdge support

⁽²⁾ Pending

⁽³⁾ Single input option per unit (up to 3 AWG) available

⁽⁴⁾ De-rating from 50°C



© SolarEdge Technologies, Inc. All rights reserved. SOLAREEDGE, the SolarEdge logo, OPTIMIZED BY SOLAREEDGE are trademarks or registered trademarks of SolarEdge Technologies, Inc. All other trademarks mentioned herein are trademarks of their respective owners. Date: 03/2018/V01/ENG INM. Subject to change without notice.

A19 Wechselrichter IQ 7, IQ 7+, IQ 7A, IQ 7X

Enphase Mikrowechselrichter IQ 7, IQ 7+, IQ 7A und IQ 7X

EINGANG (DC)	IQ7-60-2-INT	IQ7PLUS-72-2-INT	IQ7A-72-2-INT	IQ7X-96-2-INT
Empfohlene Modulleistung (STC) ¹	235 W – 350 W +	235 W – 440 W +	295 W – 460 W +	320 W – 460 W +
Modul-Kompatibilität	60 Zellen/ 120 Halb-Zellen	60 Zellen/ 120 Halb-Zellen und 72 Zellen/ 144 Halb-Zellen	60 Zellen/ 120 Halb-Zellen und 72 Zellen/ 144 Halb-Zellen	96 Zellen
Max. DC-Eingangsspannung	48 V	60 V	58 V	79,5 V
MPP-Spannungsbereich	27 V – 37 V	27 V – 45 V	38 V – 43 V	53 – 64 V
Betriebsspannungsbereich	16 V – 48 V	16 V – 60 V	18 V – 58 V	25 V – 79,5 V
Min./Max. Startspannung	22 V / 48 V	22 V / 60 V	33 V / 58 V	33 V / 79,5 V
Max. DC-Kurzschlussstrom (Modul I _{sc})	15 A	15 A	15 A	10 A
Überspannungsklasse DC-Anschluss	II	II	II	II
DC-Anschluss-Rückspelsestrom	0 A	0 A	0 A	0 A
PV-Feld-Konfiguration	Der AC-seitige Schutz verlangt maximal 20A pro Zweigstromkreis			
Ausgang (AC)	Mikro-WR IQ 7	Mikro-WR IQ 7+	Mikro-WR IQ 7A	Mikro-WR IQ 7X
Ausgangsspitzenleistung	250 VA	295 VA	366 VA	320 VA
Ausgangsnennleistung	240 VA	290 VA	349 VA	315 VA
Spannung/Nennspannungsbereich ²	230 V / 184 – 276 V	230 V / 184 – 276 V	230 V / 219 – 264 V	230 V / 184 – 276 V
Maximaler Dauerausgangsstrom	1,04 A	1,26 A	1,52 A	1,37 A
Nennfrequenz	50 Hz	50 Hz	50 Hz	50 Hz
Erweiterter Frequenzbereich	45 – 55 Hz	45 – 55 Hz	45 – 55 Hz	45 – 55 Hz
Max. Anzahl Geräte pro 20A-Zweigstromkreis ³	15 (230 VAC)	12 (230 VAC)	10 (230 VAC)	11 (230 VAC)
Überspannungsschutzklasse	III	III	III	III
Fester Leistungsfaktor	1,0	1,0	1,0	1,0
Leistungsfaktor (anpassbar)	0,8 kapazitiv ... 0,8 induktiv	0,8 kapazitiv ... 0,8 induktiv	0,8 kapazitiv ... 0,8 induktiv	0,8 kapazitiv ... 0,8 induktiv
WIRKUNGSGRAD	@230 V	@230 V	@230 V	@230 V
Wirkungsgrad nach EN 50530 (EU)	96,5 %	96,5 %	96,5 %	96,5 %
MECHANISCHE DATEN				
Betriebstemperaturbereich (Umgebung)	-40 °C bis +65 °C			
Relative Luftfeuchtigkeit	4% bis 100% (kondensierend)			
DC-Steckverbindung	MC4-Steckverbinder (im Lieferumfang enthalten)			
Abmessungen (BxHxT)	212 mm x 175 mm x 30,2 mm (ohne Montagehalterung)			
Gewicht	1,08 kg			
Kühlung	Freie Konvektion – kein Lüfter			
Zugelassen für feuchte Umgebung	Ja			
Verschmutzungsgrad	3			
Gehäuse	Klasse II doppelt isoliertes, korrosionsbeständiges Polymergehäuse			
IP-Schutzart	IP67 / Außenbereich			
FUNKTIONALITÄTEN				
Kommunikation mit dem Envoy-S	Über die Stromleitung / Power Line Communication (PLC)			
Überwachungsoptionen	Enlighten Manager und MyEnlighten, kompatibel mit dem Enphase Envoy-S			
Konformität	IEC/EN 61000-6-3, IEC/EN 62109-1, IEC/EN 62109-2, EN 50549*, VDE-AR-N-4105*, TOR Erzeuger Typ A* (geprüft nach OVE Richtlinie R25) *Q-RELAY-1P-INT oder Q-RELAY-3P-INT erforderlich)			
Garantie ⁴	25 Jahre			

1. Kein vorgegebenes DC/AC-Verhältnis. Siehe Online-Kompatibilitätsrechner unter <https://enphase.com/de-de/support/modulkompatibilitat>.

2. Der Nennspannungsbereich kann – falls aufgrund der Anforderungen des Energieversorgers erforderlich – über die Nennwerte hinaus erweitert werden.

3. Grenzen können variieren. Beachten Sie die lokalen Anforderungen, um die Anzahl der Mikrowechselrichter pro Stromkreis in Ihrer Region zu bestimmen. Bei Dauerbelastung darf das Enphase Q-Cable und Q-RAW-Cable mit 2,5 mm² Leitungsquerschnitt (je nach Verlegeart und Betriebstemperatur) maximal 16 A Strom pro Phase führen (mit 20 A Absicherung). Ein 3%-iger Spannungsanstieg im System ist erlaubt. Eine möglichst steifmögliche Auslegung der Q-Cable Zweigstromkreise auf den Generatorflächen minimiert den jeweiligen Spannungsanstieg in den Zweigstromkreisen. Richtwerte für die maximale Anzahl an Mikro-Wechselrichtern in Q-Cable Zweigstromkreisen, um den Spannungsanstieg auf etwa 1 % zu begrenzen sind 15 Stück IQ 7 / 12 Stück IQ 7+ / 11 Stück IQ 7X und 8 Stück IQ 7A Mikro-Wechselrichter.

4. Die Garantie auf den Mikro-Wechselrichter erfordert die Installation eines Enphase Kommunikations-Gateways. Sie wird von 25 auf 2 Jahre eingeschränkt, wenn kein Enphase mit dem System verbunden ist und keine Daten an die Cloud gesendet werden. Zusätzliche Informationen finden Sie unter diesem [LINK](#).

Weitere Informationen auf enphase.com/de

© 2020 Enphase Energy. Alle Rechte vorbehalten.
Alle verwendeten Warenzeichen oder Marken sind Eigentum von Enphase Energy, Inc. 05.12.2020



A20 Wirtschaftlichkeitsberechnung ohne MLPE (Level 3 und 4)

```
import math

module = sum(anz_module)

# Grundlagen
Eigenverbrauch = float(Produktion) * (float(EV_pr) / 100)
Eingespiesener_Strom = float(Produktion) - float(Eigenverbrauch)
Förderbeitrag = float(Investitionskosten) * (float(FB_pr) / 100)
anz_wechselrichter = math.ceil(float(anz_strings) / float(aufn_strings))

# Einnahmen
Einspeisevergütung = round(float(Einspeiseverg) * Eingespiesener_Strom)
Stromkosteneinsparung = round(Eigenverbrauch * float(Strompreis), 1)
Einnahmen_Jahr = round(Einspeisevergütung + Stromkosteneinsparung)
Einnahmen_X_Jahre = round(Einnahmen_Jahr * float(years), 1)

# Kosten
Kosten_wechselrichter = float(kosten_wechselrichter) * int(anz_wechselrichter)
Instandhaltungskosten = (float(Investitionskosten) + float(kosten_wechselrichter)) *
(float(IK_pr) / 100)
Investitionskosten_g = float(Kosten_wechselrichter) + float(Investitionskosten)
Kosten_gesamt_Jahr = round(Instandhaltungskosten)
Kosten_gesamt_X_Jahre = round(Kosten_gesamt_Jahr * float(years) + Kosten_
    wechselrichter + (Kosten_wechselrichter * math.floor(float(years)/15)))

# Rendite
Rendite = round((Einnahmen_Jahr * 100) / (float(Investitionskosten_g) - Förderbei-
    trag), 1)

# Amortisation
Amortisationsdauer = round((float(Investitionskosten) - Förderbeitrag + Kos-
    ten_wechselrichter) / (Einnahmen_Jahr), 1)
```

A21 Wirtschaftlichkeitsberechnung mit MLPE (Level 3 und 4)

```
import math

module = sum(anz_module)

# Grundlagen
Eigenverbrauch = float(Produktion) * (float(EV_pr) / 100)
Eingespiesener_Strom = float(Produktion) - float(Eigenverbrauch)
Förderbeitrag = float(Investitionskosten) * (float(FB_pr) / 100)
anz_wechselrichter = math.ceil(float(anz_strings) / float(aufn_strings))

# Einnahmen
Einspeisevergütung = round(float(Einspeiseverg) * Eingespiesener_Strom)
Stromkosteneinsparung = round(Eigenverbrauch * float(Strompreis), 1)
Einnahmen_Jahr = round(Einspeisevergütung + Stromkosteneinsparung)
Einnahmen_X_Jahre = round(Einnahmen_Jahr * float(years), 1)

# Kosten
kosten_opt = float(module * MLPE)
Kosten_wechselrichter = float(kosten_wechselrichter) * int(anz_wechselrichter)
Instandhaltungskosten = (float(Investitionskosten) + float(kosten_opt) +
float(Kosten_wechselrichter)) * (float(IK_pr) / 100)
Investitionskosten_g = float(Kosten_wechselrichter) + float(Investitionskosten) +
float(kosten_opt)
Kosten_gesamt_Jahr = round(Instandhaltungskosten)
Kosten_gesamt_X_Jahre = round(Kosten_gesamt_Jahr * float(years) + Kos-
ten_wechselrichter + (Kosten_wechselrichter * math.floor(float(years)/15)) + kos-
ten_opt + (kosten_opt * math.floor(float(years)/15)))

# Rendite
Rendite = round((Einnahmen_Jahr * 100) / (float(Investitionskosten_g) - Förderbei-
trag), 1)

# Amortisation
Amortisationsdauer = round((float(Investitionskosten) - Förderbeitrag + Kos-
ten_wechselrichter + float(kosten_opt)) / (Einnahmen_Jahr), 1)
```


A22 Wirtschaftlichkeitsberechnung (Level 1 und 2)

```
import rhinoscriptsyntax as rs

#Grundlagen
Eigenverbrauch = float(Produktion) * (float(EV_pr) / 100 )
Eingespiesener_Strom = float(Produktion) - float(Eigenverbrauch)
Instandhaltungskosten = float(Investitionskosten) * (float(IK_pr) / 100)
Förderbeitrag = float(Investitionskosten) * (float(FB_pr) / 100)

#Einnahmen
Einspeisevergütung = float(Einspeiseverg) * Eingespiesener_Strom
Stromkosteneinsparung = round(Eigenverbrauch * float(Strompreis), 1)
Einnahmen_Jahr = round(Einspeisevergütung + Stromkosteneinsparung)
Einnahmen_X_Jahre = round(Einnahmen_Jahr * float(years), 1)

#Kosten
Kosten_gesamt_Jahr = Instandhaltungskosten
Kosten_gesamt_X_Jahre = Kosten_gesamt_Jahr * float(years)

#Rendite
Rendite = round((Einnahmen_Jahr * 100) / (float(Investitionskosten) - Förderbeitrag),
1)

#Amortisation
Amortisationsdauer = round((float(Investitionskosten) - Förderbeitrag) / (Einnah-
men_Jahr), 1)
```

A23 Ertragsberechnung jährlich

Ertragsberechnung pro Modul

```
import rhinoscriptsyntax as rs
def calculate_module_output(module_capacity, radiation_year, radiation_hour,
PR_factor):
    module = []
    for i in range(len(radiation_year)):
        module_output_year = radiation_year[i] * (module_capacity * 1) * PR_factor
        module_output_hour = radiation_hour[i] * (module_capacity * 1) * PR_factor
        module.append((module_output_year, module_output_hour))
    return module
```

```
module = calculate_module_output(module_capacity, radiation_year, radiation_hour,
PR_factor)
year_output = [x[0] for x in module]
```

```
hourly_output = [x[1] for x in module]
```

Ertragsberechnung pro String

```
import rhinoscriptsyntax as rs
```

```
groups = eval(groups)
```

```
for i, group in enumerate(groups):
    for j, module in enumerate(group):
        group[j] = (module, yields[module])
```

```
print(groups)
```

```
group_yields = []
for i, group in enumerate(groups):
    min_yield = min(yield_ for _, yield_ in group)
    group_value = min_yield * len(group)
    group_yields.append("Gruppe {}: {}".format(i+1, group_value))
print(group_yields)
```

```
total_yield = sum(min(yield_ for _, yield_ in group) * len(group) for group in
groups)
print("Gesamtgruppenwert: {}".format(total_yield))
```

A24 Ertragsberechnung monatlich / stündlich

Ertragsberechnung pro Modul

```
import rhinoscriptsyntax as rs
def calculate_module_output(module_capacity, radiation_hour, PR_factor):
    module = []
    for i in range(len(radiation_hour)):
        module_output_hour = radiation_hour[i] * (module_capacity * 1) * PR_factor
        module.append((module_output_hour))
    return module
```

```
module = calculate_module_output(module_capacity, radiation_hour, PR_factor)
hourly_output = [x for x in module]
```

Formatierung

```
import rhinoscriptsyntax as rs

def number_and_group_list(input_list, group_size):
    numbered_list = list(enumerate(input_list))
    grouped_list = [numbered_list[i:i+group_size] for i in range(0,
len(numbered_list), group_size)]
    return [[(x[0] % group_size, x[1]) for x in group] for group in grouped_list]
```

```
input_list = x
group_size = size
output_list = str(number_and_group_list(input_list, group_size))
print(output_list)
```

Ertragsberechnung pro String pro Stunde/Monat

```
import rhinoscriptsyntax as rs
import ast
```

```
string = eval(groups)
modules_Monat_str = yields
```

```
modules_Monat = ast.literal_eval(modules_Monat_str)
```

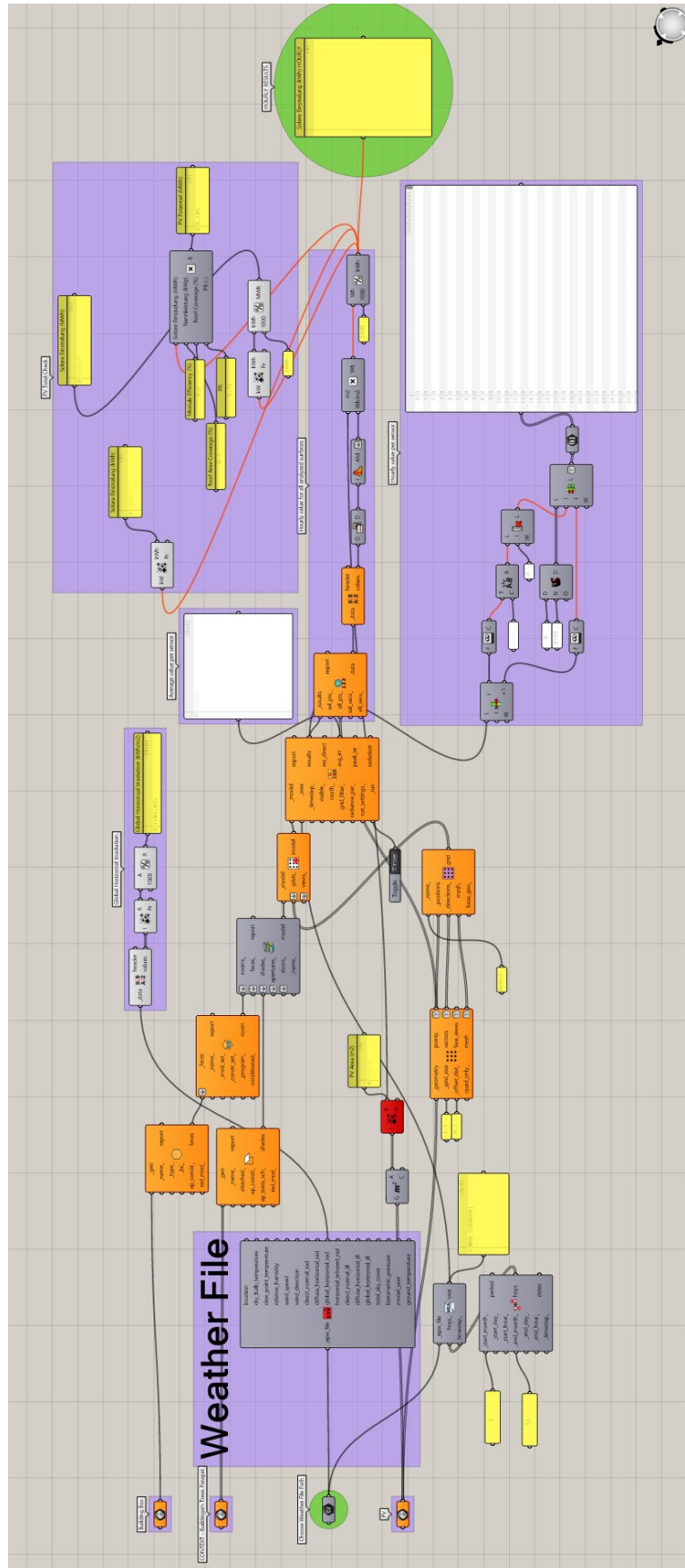
```
result = []
gruppen = []
total_yield = 0
group_yields = []
```

```
for i, module_group in enumerate(modules_Monat):
    yield_result = []
    group_result = []
    for group in string:
        modules = [module for module in module_group if module[0] in group]
        if modules:
            min_value = min(module[1] for module in modules)
            group_value = len(group) * min_value
```

```
        else:
            group_value = 0
            yield_result.append(group_value)
            group_result.append(modules)
            result.append(yield_result)
            gruppen.append(group_result)
            total_yield += sum(yield_result)

string_yield = [sum([result[i][j] for i in range(len(modules_Monat))] for j in
range(len(string))]
gruppen_yield = []
for i, yield_value in enumerate(string_yield):
    gruppen_yield.append("Gruppe {}: {}".format(i+1, yield_value))
    print("Yield of string {}: {}".format(i+1, yield_value))
```

A25 Vorlage monatliche Berechnung, Elesawy Amr, B&H



A26 Algorithmus Optimierung mit Azimut Werten

```

import rhinoscriptsyntax as rs
import ast

def count_connections(results):
    connections = {}
    modules = []

    # Schleife Gruppen
    for module in results:
        modules.append([num for row in module for num in row])

    # Schleife Modulpaare
    for i in range(len(modules[0])):
        for j in range(i+1, len(modules[0])):
            count = 0
            for group in results:
                for row in group:
                    if i in row and j in row:
                        count += 1
            connections[(i, j)] = count

    # Filterung Häufigkeit
    threshold = 4
    connections_filtered = {k: v for k, v in connections.items() if v >= threshold}

    return connections_filtered

connections = count_connections(results)
# Sortiert nach Anzahl Verbindungen
connections_sorted = sorted(connections.items(), key=lambda x: x[1], reverse=True)

#print(connections_sorted)

groups = []
assigned_modules = set()

connections_list = []

connections_list = [format(a[0]) for a in list(connections_sorted)]

connections_list = [ast.literal_eval(c) for c in connections_list]
#print(connections_list)

for i, module_i in enumerate(list_int):
    for j, module_j in enumerate(list_int[i + 1:], i + 1):
        # Überprüfung der Verbindung
        if (module_i, module_j) in connections_list or (module_j, module_i) in connections_list:
            # Gruppenzuweisung
            group_found = False
            for group in sorted(groups, key=lambda g: max(connections.get((module_i, m), 0) + connections.get((module_j, m), 0) for m in g), reverse=True):
                if module_i in group or module_j in group:
                    if module_i not in assigned_modules:
                        group.add(module_i)
                        assigned_modules.add(module_i)
                    if module_j not in assigned_modules:
                        group.add(module_j)
                        assigned_modules.add(module_j)
                    group_found = True
                    break
            if not group_found:
                groups.append({module_i, module_j})
                assigned_modules.add(module_i)

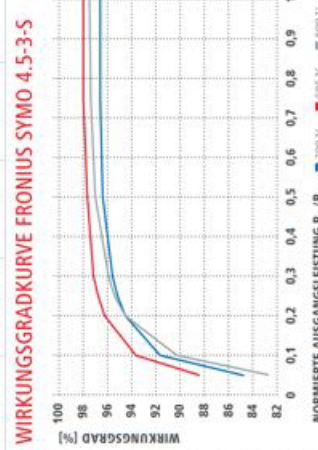
```

```
        assigned_modules.add(module_j)

final_group = [list(x) for x in groups]
print(final_group)
```

A27 Bypass Dioden Berechnung Herleitung (B&H)

Modul		Elektrische Spezifikationen		Abmessungen Module	
Modulfläche Zellen pro Modul Nennleistung pro Modul Spannung pro Modul U _{mpp} U _{oc} = 24V Module/String U _{max}	1,0 m ² 36 (6x6)	L 200 Wp	Q 180 Wp	M 150 Wp	B 120 Wp
	120 Wp (33% Abminderung wegen Farbe)	22,2 V	17,7 V	16,6 V	13,3 V
Einstrahlung W/m ²	24 V	Strom I _{mp} 9,0 A		Kurzschlussstrom I _{sc} 9,5 A	
	40 Module	27,1 V	21,8 V	20,3 V	16,3 V
Module in der Sonne (nicht im Schatten)	960 V	Wirkungsgrad 18,6%		18,4%	
	32	Maximal Systemspannung 1000 V		1000 V	
Module pro String w/	24	Rückstrombelastbarkeit 18 A		18 A	
	16	Toleranz Nennleistung ±5%		±5%	
Eingestrahltung W/m ²	40	<small>Standard Strom I_{sc} und Kurzschlussstrom I_{sc} bei 25 °C, AM 1,5; Elektrische Leistungsprobe bei STC (1000 W/m², 25 °C, AM 1,5); Messprotokolle auf Anfrage erhältlich.</small>			
	8	Anzahl Zellen		L 8 x 5 = 40	Q 8 x 4 = 32
DC Leistung pro String w/	100%	M 6 x 5 = 30		S 6 x 4 = 24	
	80%	8		0	
Spannung pro String	20%	Wechselrichter DC-Eingangsspannungsbereich (U _{dc min} - U _{dc max}) 150 - 1000 V 200 V 150 - 800 V Nutzbare MPP-Spannungsbereich			
	40%	min. Spannung unterschritten			
P / P _{max}	60%	786 628,8 471,6 314,4 157,2 0			
	48%	20% 32% 16% 12% 8% 4%			
Wirkungsgrad	48%	97,6% 97,2% 97,0% 96,0% 94,0%			
	32%	96,5% 96,0% 94,0% 92,0% 88,0%			
Module pro String in der Sonne (nicht im Schatten)	24%	92,0% 90,5% 87,0% 82,0%			
	16%	82,0% 78,1% 77,8% 77,8% 77,8%			
Module pro String in der Sonne (nicht im Schatten)	8%	77,8% 77,8% 77,8% 77,8% 77,8%			
	4%	77,8% 77,8% 77,8% 77,8% 77,8%			
Einstrahlung W/m ²	100%	98,0% 98,0% 98,0% 98,0% 98,0%			
	80%	98,0% 98,0% 98,0% 98,0% 98,0%			
Eingestrahltung W/m ²	60%	98,0% 98,0% 98,0% 98,0% 98,0%			
	40%	98,0% 98,0% 98,0% 98,0% 98,0%			
Eingestrahltung W/m ²	20%	98,0% 98,0% 98,0% 98,0% 98,0%			
	0%	98,0% 98,0% 98,0% 98,0% 98,0%			



A28 Ertragsberechnung Bypass Dioden

```

import rhinoscriptsyntax as rs

groups = eval(groups_string)

result = []
for wert in einstrahlung:
    sub_list = []
    for zahl in wert:
        if zahl[1] >= (max - ((float(threshold) * 0.01) * max)):
            sub_list.append(zahl)
    result.append(sub_list)

liste_prozent = []
for gruppe, ergebnis in zip(groups, result):
    prozent = 100 / len(gruppe) * len(ergebnis)
    liste_prozent.append(prozent)

group_yields = []
for i, group in enumerate(yields):
    group_yield_sum = sum(yield_ for _, yield_ in group)
    group_value = group_yield_sum
    group_yields.append("Gruppe {}: {}".format(i+1, group_value))

total_yield_old = sum(float(group_yield.split(": ")[1]) for group_yield in
group_yields)

Leistungsliste = []
Leistung = 0.0
max_new = max * 1000
#print(max_new)

for proz in liste_prozent:
    if 900 <= max_new <= 1500 and 90 <= proz <= 101:
        Leistung = round(98.0, 1)
    elif 900 <= max_new <= 1500 and 70 <= proz < 90:
        Leistung = round(78.1, 1)
    elif 900 <= max_new <= 1500 and 50 <= proz < 70:
        Leistung = round(58.3, 1)
    elif 900 <= max_new <= 1500 and 30 <= proz < 50:
        Leistung = round(38.6, 1)
    elif 900 <= max_new <= 1500 and proz < 30:
        Leistung = round(0.0, 1)

    elif 700 <= max_new < 900 and 90 <= proz <= 100:
        Leistung = round(77.8, 1)
    elif 700 <= max_new < 900 and 70 <= proz < 90:
        Leistung = round(62.2, 1)
    elif 700 <= max_new < 900 and 50 <= proz < 70:
        Leistung = round(46.6, 1)
    elif 700 <= max_new < 900 and 30 <= proz < 50:
        Leistung = round(30.7, 1)
    elif 700 <= max_new < 900 and proz < 30:
        Leistung = round(0.0, 1)

    elif 500 <= max_new < 700 and 90 <= proz <= 100:
        Leistung = round(58.6, 1)
    elif 500 <= max_new < 700 and 70 <= proz < 90:

```

```
    Leistung = round(46.6, 1)
elif 500 <= max_new < 700 and 50 <= proz < 70:
    Leistung = round(34.7, 1)
elif 500 <= max_new < 700 and 30 <= proz < 50:
    Leistung = round(22.6, 1)
elif 500 <= max_new < 700 and proz < 30:
    Leistung = round(0.0, 1)

elif 300 <= max_new < 500 and 90 <= proz <= 100:
    Leistung = round(38.6, 1)
elif 300 <= max_new < 500 and 70 <= proz < 90:
    Leistung = round(30.7, 1)
elif 300 <= max_new < 500 and 50 <= proz < 70:
    Leistung = round(22.6, 1)
elif 300 <= max_new < 500 and 30 <= proz < 50:
    Leistung = round(14.7, 1)
elif 300 <= max_new < 500 and proz < 30:
    Leistung = round(0.0, 1)

elif 100 <= max_new < 300 and 90 <= proz <= 100:
    Leistung = round(18.8, 1)
elif 100 <= max_new < 300 and 70 <= proz < 90:
    Leistung = round(14.7, 1)
elif 100 <= max_new < 300 and 50 <= proz < 70:
    Leistung = round(10.9, 1)
elif 100 <= max_new < 300 and 30 <= proz < 50:
    Leistung = round(7.0, 1)
elif 100 <= max_new < 300 and proz < 30:
    Leistung = round(0.0, 1)

Leistungsliste.append(Leistung)

print(Leistungsliste)

ergebnisse = []
total_yield_new = 0

for gruppe, prozent in zip(group_yields, Leistungsliste):
    gruppenname, gruppenwert = gruppe.split(':', 1)
    gruppenwert = float(gruppenwert)

    ergebnis = gruppenname + ": " + str(gruppenwert * prozent / 100)

    ergebnisse.append(ergebnis)

total_yield_new += gruppenwert * prozent / 100
```