Improving Supply Chain Resiliency through Solar Panel Delivery Optimization

by

Regina Ceballos Mondragón

B.S., Applied Mathematics

Instituto Tecnológico Autónomo de México, 2019

Submitted to the MIT Sloan School of Management and Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of

Master of Business Administration

and

Master of Science in Electrical Engineering and Computer Science

in conjunction with the Leaders for Global Operations program

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by: Regina Ceballos Mondragón
MIT Sloan School of Management and
Department of Electrical Engineering and Computer Science
May 10, 2024

Certified by: Daniel Freund
Assistant Professor of Operations Management
Thesis Supervisor

Certified by: Thomas Magnanti
Institute Professor of Artificial Intelligence and Decision Making
Thesis Supervisor

Certified by: Duane Boning
Clarence J. LeBel Professor of Electrical Engineering and Computer Science
Thesis Reader

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Accepted by: Maura Herson
Assistant Dean, MBA Program
MIT Sloan School of Management

Improving Supply Chain Resiliency through Solar Panel Delivery Optimization

by

Regina Ceballos Mondragón

Submitted to the MIT Sloan School of Management and
Department of Electrical Engineering and Computer Science
on May 10, 2024, in partial fulfillment of the
requirements for the degrees of
Master of Business Administration
and
Master of Science in Electrical Engineering and Computer Science

Abstract

Following NextEra Energy Resources' accelerated growth and disruptions in the solar panel supply chain, their solar panel allocation process is becoming more complex. This process results in a schedule that determines when to deliver close to 150 million solar panels to more than fifty project sites under development and construction, while balancing requirements from multiple stakeholders. Due to project and contract interdependencies, modifying the equipment delivery schedule leads to costs that have consequential impacts. This thesis presents and implements a novel mixed integer programming model to determine the optimal schedule for delivering solar panels to project sites. The model abstracts impactful and quantifiable costs and minimizes them to propose a realistic solution. It produces a schedule in significantly less time than the current manual approach by finding a feasible solution in less than 15 minutes. The thesis introduces three scenarios of supply chain disruptions that mimic real-world events, demonstrating the model's flexibility and helping NextEra Energy Resources adapt to future supply chain disruptions.

Thesis Supervisor: Daniel Freund
Title: Assistant Professor of Operations Management

Thesis Supervisor: Thomas Magnanti
Title: Institute Professor of Artificial Intelligence and Decision Making

Thesis Reader: Duane Boning
Title: Clarence J. LeBel Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to sincerely thank my advisors, Dr. Thomas Magnanti and Dr. Daniel Freund. Your enthusiasm, engagement, and guidance throughout this project have been instrumental in enriching my thesis, culminating in a work I am proud of. I thank my reader, Dr. Duane Boning, for your insightful comments and unhesitating support in helping me complete my thesis.

This work would not have been possible without the support of my NextEra Energy team, particularly Louis Boguchwal, Kevin Schell, Adam Meranda, and Matt Barrows. Thank you for hosting an internship full of learning and always advocating for me. Having a project that was so thoughtfully planned and that required a diverse range of skills made my experience exciting and fostered my professional growth. I am immensely grateful for the opportunity to take ownership of the project, experiment with the constraints, and improve the results.

Thank you to my parents, Guadalupe Mondragón and Gerardo Ceballos. Your unconditional love is the foundation of my achievement. To my brother, Pablo Ceballos, for being a constant companion and ensuring I always have a home to return to. To Jesús Yruegas, for supporting me and encouraging me every step of the way.

Finally, thank you to MIT and the LGO program. Thank you to the program office for believing in me, supporting my academic development, and granting me the opportunity to be a part of this amazing community. Thank you to my LGO classmates for embarking on this journey with me, for your care and companionship, and for fostering our unique culture together. I will miss laughing with you. And to my friends, you made Cambridge a home, and I cherish the moments we had together always.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Figures

List of Tables

Acronyms

Acronym	Description	Page
AD/CVD	Antidumping and Countervailing Duty	24
C&I	Commercial and Industrial	21
COD	Commercial Operation Date	26
DOC	Department of Commerce	24
EPC	Engineering, Procurement, and Construction company	27
FPL	Florida Power & Light	20
IRA	Inflation Reduction Act	25
LD	Liquidated Damages	75
MSA	Master Supplier Agreement	27

Acronym	**Description**	**Page**
NEER	NextEra Energy Resources	20
PPA	Power Purchase Agreement	26
PSA	Project Supplier Agreements	27
PSC	Florida Public Service Commission	21
UFLPA	Uyghur Forced Labor Prevention Act	24

Chapter 1

Introduction

This thesis presents and implements a novel optimization model to determine the optimal schedule for delivering solar panels to NextEra Energy Resources' project sites. Due to NextEra Energy Resources' accelerated growth and disruptions in the supply chain, their solar panel allocation process is becoming more complex. The schedule that orchestrates this process determines when to simultaneously deliver close to 150 million solar panels to more than fifty project sites under development and construction. It balances the requirements from multiple stakeholders, including Commercial (Development), Contracting (Supply Chain, Early Stage, and Late Stage), Construction, and Logistics. Modifying the equipment delivery schedule results in costs that have consequential impacts across the portfolio because of project and contract interdependencies.

The model improves NextEra Energy Resources' supply chain resiliency by responding faster to disruptions and adapting to changes with greater flexibility. Manufacturing delays disrupt equipment deliveries, in turn disrupting the projects' development. By quantifying the impact of potential changes to the delivery schedule in a short amount of time, the model can select the schedule with the least disruptive change. An automated approach to scheduling and allocation has the potential to minimize costs and make prompt strategic decisions across a growing portfolio.

In this thesis, Chapter 1 introduces the company, industry, and project motivation. Chapter 2 explains the process of developing and testing the model. Chapter 3 presents the mathematical formulation of the optimization used to automate the solar panel allocation process. Chapter 4 shows the results of running the model with different scenarios. Finally, Chapter 5 presents future work, additional considerations, and a conclusion.

1.1 Background

1.1.1 NextEra Energy, Inc.

The history of NextEra Energy, Inc. [1] dates back to 1925 when Florida Power & Light (FPL) was established to provide utility services in the state. Initially a privately owned enterprise, FPL steadily expanded its operations and has grown to become Florida's most prominent electric utility, catering to more than 12 million customers as of 2021.

With the growth in size also came an expansion in territory and capabilities. The business that is now NextEra Energy Resources (NEER) was launched to develop, construct, and operate renewable energy assets, such as wind, solar, and battery projects, throughout the US and Canada. NEER is NextEra Energy's second-largest business and the world's largest generator of renewable energy from the wind and sun. Today, FPL and NEER are NextEra Energy's principal businesses, owning over 50 GW of generation capacity [2]. Together, FPL and NEER have a net owned generation capacity of more than 6.5 GW of solar energy [3].

A main difference between FPL and NEER is that the Florida Public Service Commission (PSC) regulates the former, while the latter competes in regulated and deregulated markets across the country. The PSC provides oversight to ensure that FPL's investments constitute a prudent investment of ratepayer money, and FPL provides long-term visibility to its investment plans through a Ten Year Site Plan. As such, the solar panel allocation model in this thesis focuses on projects from the NEER portfolio, which tends to be more dynamic. NEER's customers are primarily utilities and Commercial and Industrial (C&I) companies.

1.1.2 Solar energy industry

Solar energy has been steadily penetrating the US energy market. Every year, the share of the total energy generated by solar sources has grown. As Figure 1-1 shows [4], the current solar energy generation is approximately 3% of the net US energy generation at a utility-scale [5].

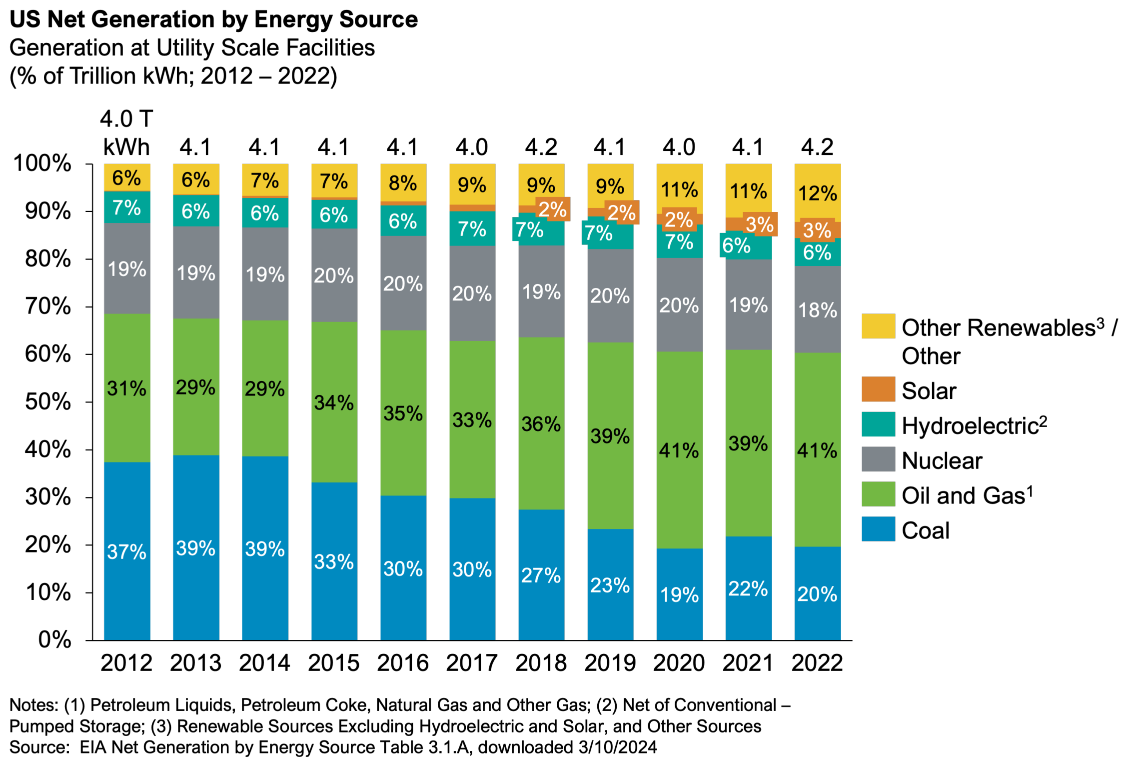

Figure 1-1: US net generation by energy source [4].

In 2022, the US generated approximately 205 billion kilowatt hours of solar energy from utility and small-scale facilities. As Figure 1-2 shows, this represents seven times the electricity that solar sources generated in 2014.

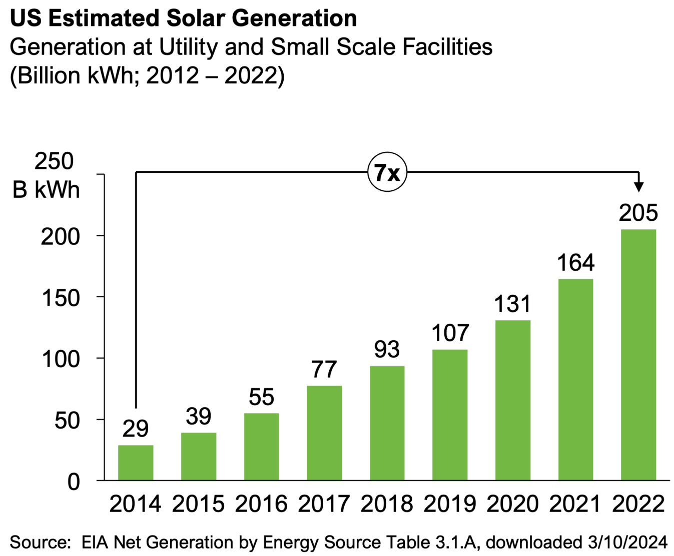

Figure 1-2: US estimated solar generation [4].

The solar energy industry growth is expected to continue. Since 2019, solar energy has had the highest share of new capacity additions in the US. In 2023, solar energy accounted for 53% of the total new capacity generation additions [6]. This massive growth resulted from multiple causes, including government incentives, regulations, and the steady reduction of prices because of technological improvements until 2020 [5, 7]. In particular, the weighted average cost of constructing utility-scale solar projects dropped from nearly 4 to 1.6 thousand dollars per kilowatt from 2013 to 2020 [8].

However, the disruptions to the world supply chain that COVID-19 caused also affected the solar panel industry. The global solar supply chain is heavily concentrated in China. Chinese manufacturers produce every step of the solar panels, from the raw materials like silicon and polysilicon to the actual panel modules. Factory shutdowns and labor shortages heavily delayed panel manufacturing [9, 10]. Additionally, the conventional downward trend in solar panels reversed. Prices skyrocketed due to increasing shipping and component costs. In particular, the cost of polysilicon tripled from 2020 to 2021 [11]. The volatile pricing created uncertainty for solar project developers, including NEER.

In the years following 2021, new trade actions limited the import of solar panels from China into the US, mainly:

- Antidumping and Countervailing Duty (AD/CVD) [12]: The law was established to protect US businesses from unfair competition due to unfair foreign pricing and distorting government subsidies in 2012. In 2022, the US Department of Commerce (DOC) investigated whether several solar module manufacturers circumvented tariffs imposed on China. The investigation sought to determine if suppliers had used parts of the solar panels manufactured in China and assembled the panels in other countries to avoid the tariffs. In 2023, the DOC ruled that some imports from Cambodia, Thailand, Vietnam, and Malaysia circumvented the antidumping and countervailing duties. Thus, some companies were circumventing the AD/CVD law. It also published a list of companies that were found not to be circumventing. Solar panel imports from the investigated countries, which account for around 80% of US imports, were halted during the investigation [13].

- Uyghur Forced Labor Prevention Act (UFLPA) [14]: In December 2021, the UFLPA was signed into law to prevent importing goods mined, produced, or manufactured, wholly or in parts, with forced labor, especially from the Xinjiang Uyghur Autonomous Region. The law was implemented in June 2022, and more than one thousand shipments of solar energy equipment were detained by October 2023 under UFLPA [15]. Increasingly, solar manufacturers have been required to provide documentation affirmatively demonstrating that they did not use forced labor to source the polysilicon or underlying raw materials in their solar panels. Some manufacturers have successfully had their panels released from customs and imported into the US [16].

The global solar industry's heavy reliance on Chinese manufacturers and the limited mechanisms to trace steps in the supply chain resulted in additional delays and costs.

Nevertheless, the Inflation Reduction Act (IRA) was introduced in August 2022 to accelerate the transition to clean energy, including rejuvenating US manufacturing and supply chains for decarbonization [17]. The bill became effective in January 2023, offering monetary incentives amounting to hundreds of billions of dollars for clean energy [18], potentially increasing the cost competitiveness and availability of solar panels manufactured in the US [6]. While the IRA is effective, many of the details of its implementation are still being defined, adding complexity to which projects qualify for the bill's benefits [19]. In the meantime, NEER plans to develop new projects of up to 19 GW of solar energy capacity by 2026, many of which will be eligible for tax credits under the IRA [20].

1.2 Solar project development

At a certain point in time, multiple NEER teams are simultaneously working on different steps to develop a solar energy project. These teams include Commercial (Development), Contracting and Project Management (Supply Chain and Early Stage), Construction, and Logistics. Each team has a critical task within the project development process; they collaborate across activities, and their decisions impact one another. Figure 1-3 presents a simplification of the development process to help understand its intricacies.

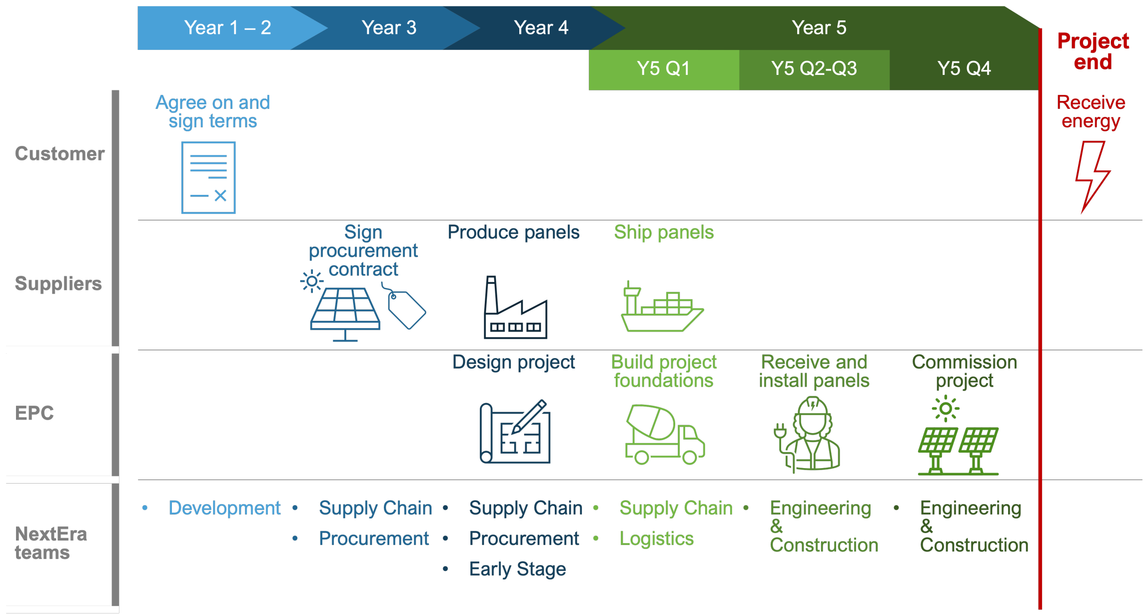

Figure 1-3: Solar project development process.

The development team starts by evaluating potential projects' strategic and financial viability and seeks out potential customers to sell the projects' attributes to (e.g., capacity, energy). They aim to reach commercial agreements for NEER's customers that allow the projects to be developed profitably. The main components of the agreement include the details that make up a project, including the Commercial Operation Date (COD), which refers to the date when the project will go online and start producing energy, the amount of power that the project will produce, and the location and price of the project. If NEER and their customer coordinate to develop a project, a Power Purchase Agreement (PPA) is signed. PPAs are signed several years before their COD. As such, NEER has ample time to plan for project demand.

Next, the Supply Chain team procures equipment to build the projects one to three years in advance for all projects that NEER will develop. The contracts with the suppliers are bulk; that is, NEER buys equipment from the same supplier for multiple projects. This approach helps NEER maintain collaborative relationships with their suppliers and use their scale to have a strong negotiating position. A contract that services various projects is a Master Supplier Agreement (MSA). NEER uses MSAs to define overall governing terms for panel supply and develop a general production schedule that a supplier is expected to meet. As projects receive governance approval—signaling that the project has a risk low enough to support entire capital investment—these MSAs are then broken down into specific Project Supplier Agreements (PSA), which adopt the terms of the MSA. The PSAs detail the weekly schedule, along with the supplier that will be required to deliver solar panels to a given project. Subsection 1.2.1 details how panels are allocated from MSAs and PSAs to the projects.

Before starting construction, NEER contracts an Engineering, Procurement, and Construction company (EPC). NEER outsources the construction of solar projects to the EPC, including receiving and installing solar panels. NextEra Energy's Engineering and Construction team oversees the EPC. Additionally, the Engineering and Construction team manages the project's on-site development and dictates the EPC's construction schedule based on the delivery schedule. Figure 1-4 shows an illustrative view of the activities during the project's construction.

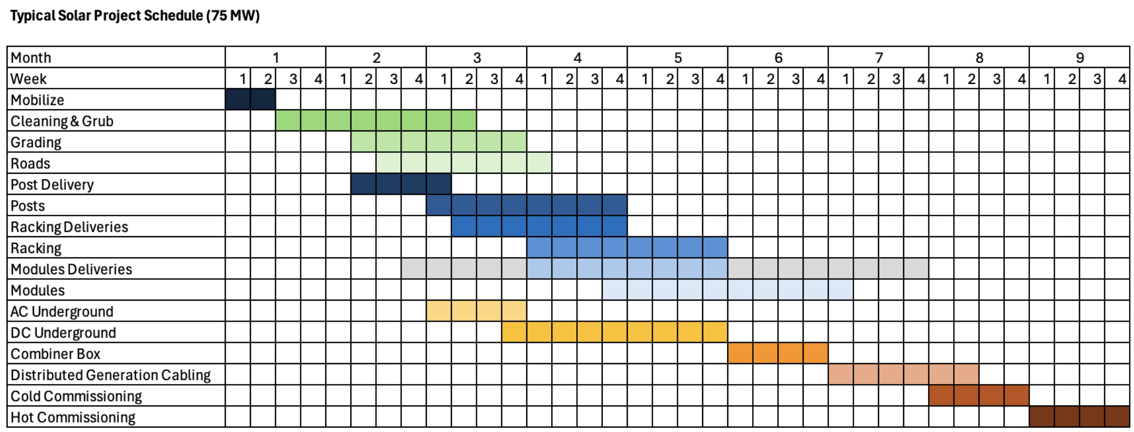

Figure 1-4: Solar project construction schedule.

In parallel, the Logistics team oversees the timely delivery of the equipment, mainly solar panels, to each project. They work hand in hand with the suppliers to monitor the factory production, track the location of different shipments, and troubleshoot any delivery problem. The supplier produces panels and delivers them to each site every week. Across the board, the Pricing & Analytics team supports and communicates strategic decisions regarding project developments, including any changes they make to the solar panel delivery schedule.

1.2.1 Solar panel allocation process

The solar panel allocation process is an iterative approach to assign suppliers to each project and determine when the suppliers will deliver the panels. At the beginning of the process, NEER makes high-level decisions, and as time progresses, the level of detail increases so that exhaustive engineering and project scheduling can be completed. There are four main steps within the solar panel allocation process: demand planning, MSA contracting, PSA allocation, and adaption to disruptions. Figure 1-5 visualizes these processes. All processes are done manually and simultaneously depending on the different project timelines.

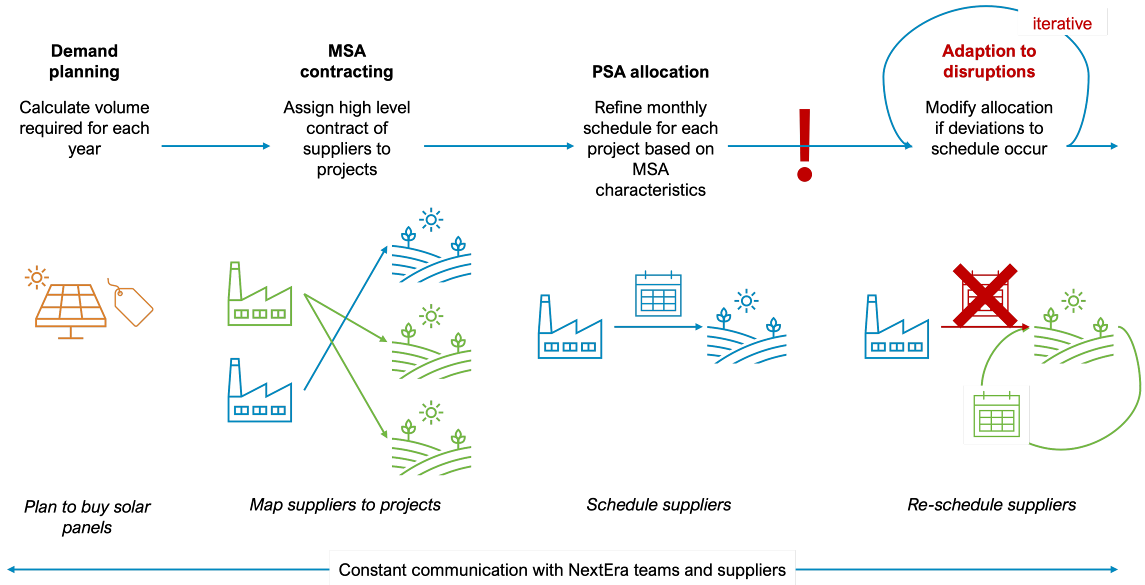

Figure 1-5: Solar panel allocation process.

- Demand planning: This process calculates the MW that projects will need to help the procurement team negotiate contracts. It happens three to five years before the COD. The demand is calculated monthly and assumes that, at this point, NEER has the potential to access an unlimited supply of solar panels. It is a process that is agnostic of suppliers.

- MSA contracting: NEER establishes a high-level contract with the suppliers based on the required volume to complete the projects. While they assign projects to a contract, they are not bound to deliver the panels to the projects in this contract. They can change the schedules and projects they deliver to but cannot change the amount contracted for the MSA without a renegotiation. NEER optimizes these contracts for profitability and deliverability based on technology, geography, and price. At this stage, NEER can create an MSA with any supplier it wants, so it can select suppliers based on strategic supply portfolio objectives (e.g., geographic diversification). However, there are limitations based on the production capacity of each supplier and the panel technology.

- PSA allocation: Once the CODs of projects approach, NEER creates a detailed schedule of weekly deliveries to the projects based on the MSA characteristics. One MSA typically corresponds to multiple PSAs, but a PSA does not correspond to multiple MSAs. In this step, the suppliers' monthly capacities and the projects' receiving capacities limit the potential allocations. This step happens three to eighteen months before the first deliveries of projects start.

- Adaption to disruptions: In this step, NEER modifies the schedule if deviations occur. There are ongoing deliveries, but they cannot happen as per the contract. The remaining volume in each MSA limits the available volume to shuffle, but NEER can add capacity from other MSAs with extra costs. The process of adaption happens by manually updating the schedule using heuristics. They look at the affected projects and decide which project's deliveries should be prioritized based on the volume not affected by the disruption. The prioritization is done based on PPA contractual requirements, experience, strategy, and time to COD. Typically, volume for projects that have later deadlines is shuffled. In extraordinary circumstances, NEER can negotiate deadlines for projects affected by the disruption. Minor disruptions are typical and happen regularly. Major disruptions are less frequent, but their occurrence increased starting with the COVID-19 pandemic.

The model in this thesis focuses on PSA allocation and adaptation to disruption steps. Note that these steps always occur simultaneously throughout NEER's portfolio, as projects have different timelines. For example, some projects will be in the demand planning step while others are in the MSA allocation step.

1.3 Problem statement

Continuing to use a manual process to create the schedule that coordinates equipment delivery for solar projects is becoming unviable. This effort is heavily time-consuming and unlikely to achieve optimal cost solutions. It also limits NEER's capability to respond promptly to the solar industry's changes to achieve its development goals.

NEER's pipeline for solar projects has steadily grown, and they expect it to grow as more businesses pledge to reduce their carbon emissions. However, NEER will need to realize this growth in a volatile solar energy industry. Not only will the scheduling process need to incorporate a more extensive portfolio, but it will also have to be flexible to increasingly frequent disruptions.

Since there is not an infinite supply of panels, and the projects must conform to a set timeline, deciding the schedule to deliver solar panels has different ramifications, including monetary, strategic, and operational impacts. The most evident consequence is NEER's monetary costs if the deliveries do not align with their contracts with the customer, supplier, and EPC.

NEER has the scale and capabilities to handle these complexities. In contrast to competitors with a single solar panel contract with one supplier, NEER sources its solar panels from multiple contractors. As such, if there is a disruption in the supply chain, it can adapt to those disruptions by redirecting unaffected supply to support imminent, high-priority projects. It leads to a cumbersome, albeit functional, manual exercise of allocating contracted supply to satisfy project demands.

This thesis proposes an optimization model that can automate the allocation process, which results in a schedule to deliver solar panels to the required projects. The model seeks to produce a schedule that can minimize the total project cost impacts of a disrupted panel supply in a timely manner.

Chapter 2

Methodology

As previously mentioned, the various teams that collectively develop solar projects must use the schedule to deliver solar panels. We divided the project into two phases since we required the teams' involvement and approval to create the model that automated the process. The initial phase built a "sandbox" model to develop early insights, and the second phase refined the model to generate pragmatic solutions that better suited the company's actual needs and context. In addition to collaboration with key stakeholders, each phase had iterative components of data management, cost calculation, and model development. The result of each phase was a functioning optimization model.

Figure 2-1 shows the project timeline chart, and this chapter provides a detailed explanation of each activity in the chart.

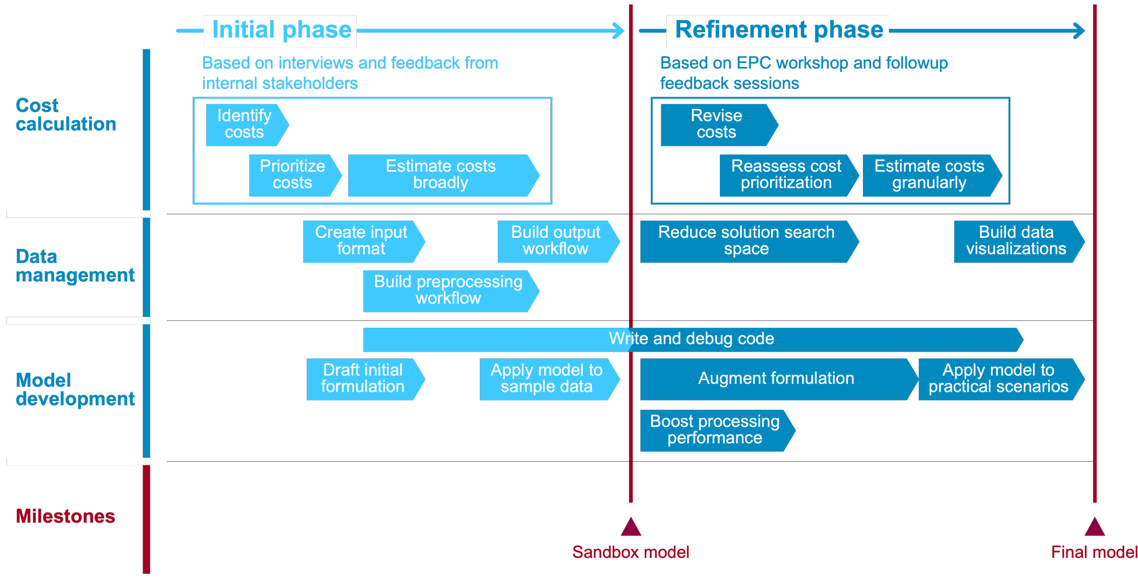

Figure 2-1: Project timeline chart.

The initial phase's "sandbox" model used a smaller dataset and had cost values based on qualitative interviews. While the results only provided a rough approximation to real panel deliveries, they served to gain an understanding of the problem, scope the model, define an initial formulation, and test sample small-sized scenarios. Additionally, by capturing the hypothesis from internal stakeholders, the optimization helped generate interest from high-level executives. Finally, it revealed the gaps remaining to complete the final model:

- Increase the size of the model to process the complete project portfolio.
- Add detail to overly simplified elements of the model's formulation.
- Estimate costs with higher accuracy.

Altogether, these objectives were required to produce actionable estimates of the minimal cost of a schedule given different disruptions to the solar supply chain. Some examples of the disruptions that the final model could process were the results of a new trade action that limited the panels available for deliveries or the acceleration of a COD for a prioritized project.

2.1 Activities within the phases

2.1.1 Cost calculation

In order to obtain cost terms for the optimization model, we identified, prioritized, and estimated the different types of costs in both the initial and the refinement phases. In the initial phase, we based the cost calculation on interviews with stakeholders inside NEER; in the refinement phase, we based it on a workshop we ran with a trusted EPC. We describe these activities next.

2.1.1.1 Initial phase: Collaboration with internal stakeholders

Before any activity, we conducted a series of interviews with internal stakeholders. The main themes of the interviews were understanding which part of the process each team owned, identifying the interdependencies between different teams, and investigating the cost drivers in various parts of the processes for each team interviewed.

After we developed some parts of the model formulation, we reviewed them with internal teams to receive feedback on how they captured the key takeaways from the interviews. We iteratively incorporated the feedback into the model.

Identifying costs During the interviews with the internal stakeholders, we identified elements that impacted the internal teams' expenses. This resulted in an inventory of potential costs that we grouped into recurring themes. This categorized list informed the input for the following prioritization step. Table C in Appendix C shows the list. We based the model's constraints on recurring themes and simplified elements that could cause expenses to represent the model's costs.

Prioritizing costs Given the project timeline, not all costs identified were immediately actionable. We used prioritization of two criteria, ease of quantification and monetary impact, to select the costs to implement in the model. Filtering the high-scoring costs in both criteria narrowed the project's focus while leveraging the most financially impactful measures to prevent costs. Figure 2-2 shows the result of the prioritization. The costs in figure 2-2 were included in the model, and they are explained in detail in chapter 3.

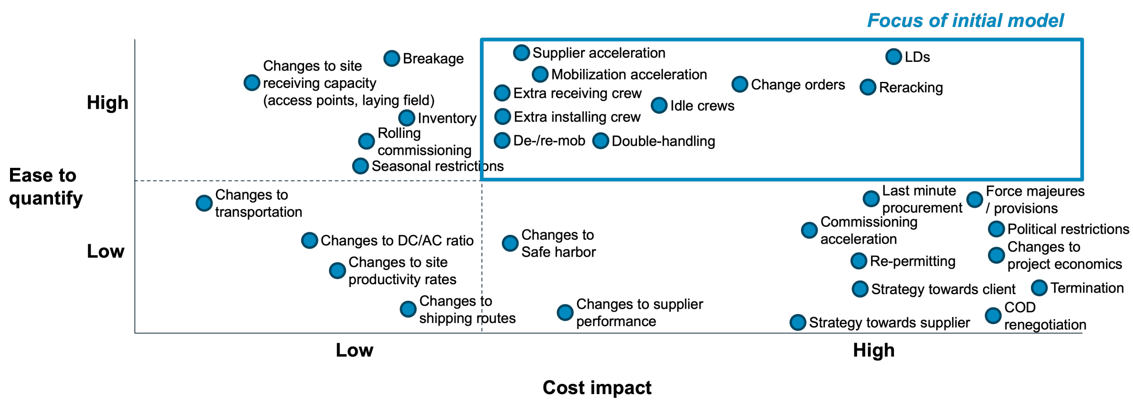

Figure 2-2: Cost prioritization with internal stakeholders.

This prioritization method resulted in some of the most impactful costs being excluded from the base model because they were hard to quantify, perhaps making one question the model's validity. We included cost types like last-minute procurement, commissioning acceleration, and termination in the final model to prevent this. The final model can also quantify the impact of cost types like strategy towards the supplier and the customer, COD renegotiation, and changes to project economics through scenario testing as described in Chapter 4. Finally, not including other cost types like force majeures, provisions, re-permitting, and political restrictions is not a significant concern because they occur with a low frequency.

Estimating costs The first cost estimation exercise used high-level assumptions. There was less concern about the accuracy of the costs, but rather, the aim was to have a working model that could directionally give cost insights. The data required to make accurate cost estimations was scattered across different company teams, and the effort required to collect all of that data was longer than the time allocated for the first phase. This resulted in simplifications; for example, we calculated some costs with averages for which project-specific or contract-specific information was unavailable.

Without going into detail about the meaning of each cost, as we will intricately describe them in Chapter 3, Figure 2-3 presents the costs that we estimated during the initial phase and gives a brief description of each cost.

	Cost description
Mobilization acceleration	• Early receiving and double handling
Suppler acceleration	• Premium over curve price
De-/re-mobilization	• Letting crew go and rehiring (as opposed to keeping crew idle)
Racking changes	• Change order
	• Racking and hardware rework
Extra crew	• Extra receiving crew • Extra installing crew • Double handling
LDs	• Penalties of not finishing before contractual COD

Figure 2-3: Cost estimation with internal stakeholders.

2.1.1.2 Refinement phase: Workshop with EPC and follow-up sessions

During the refinement phase, we used a higher degree of collaboration with an EPC with a strong partnership with NextEra Energy. The EPC selected for collaborative development is one of NextEra Energy's strategic partners for solar execution; NextEra Energy and the selected EPC jointly navigated the trade disruptions that impacted the US solar industry in 2021 and 2022 and gained valuable first-hand insight into the costs that are incurred when material delivery schedules are disrupted. As such, we replicated the cost identifying, prioritizing, and estimating steps with them in a guided manner. The interactions with the EPC included a one-day in-person workshop and multiple remote follow-up sessions.

During the workshop, we presented the EPC with five delivery scenarios, which Figures 2-4 to 2-8 show. Each scenario included a planned schedule with desired deliveries, which are defined as deliveries where the volume remains the same each month, there are no gaps in months delivered, and deliveries are finished with enough time to be commissioned before the planned COD date.[1] Each scenario approximates a real disruption that NEER has observed. The scenarios included a schedule resulting from one of the following disruptions, which we carefully constructed to be as orthogonal as possible to test for the impact of different cost coefficients efficiently:

1. Early deliveries: The solar panels arrived three months before planned, with some arriving before the planned mobilization (Figure 2-4).[2]

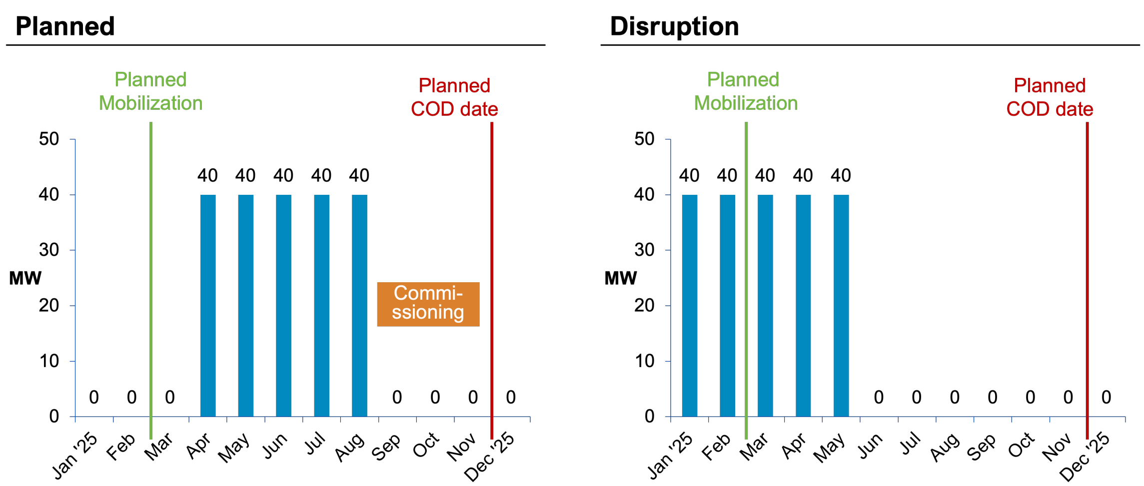

Figure 2-4: Early deliveries scenario.

2. Late deliveries: The solar panels arrive two months later than planned, leaving less time to commission before the planned COD (Figure 2-5).

[1]Commissioning is an intensive process of testing all of the equipment at the site to confirm that it can deliver power according to the requirements of the system it will be dispatching into. Projects must reserve a significant amount of time for commissioning.

[2]The mobilization date refers to the date when the EPC starts working on a project site. Section 3.8 presents more information about the mobilization concept.

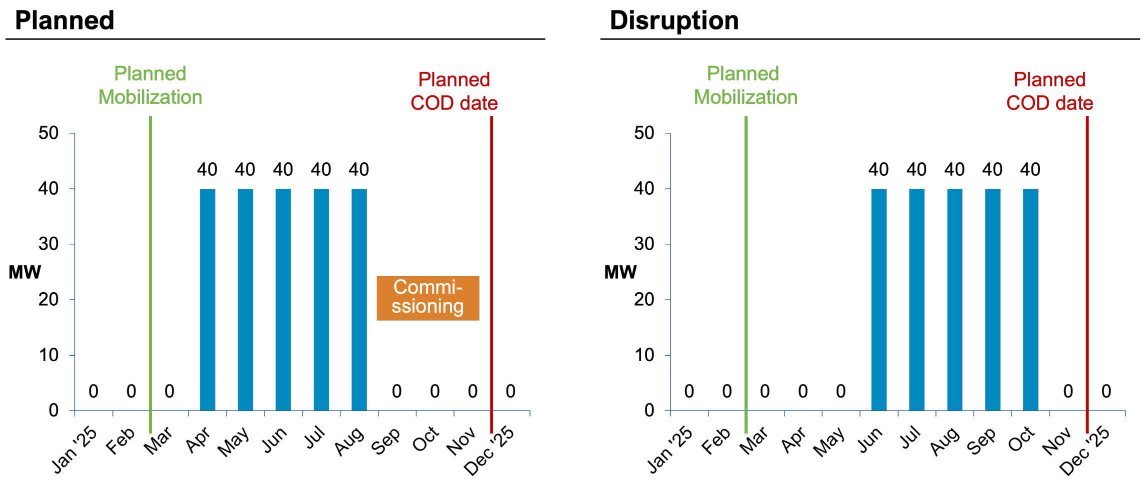

Figure 2-5: Late deliveries scenario.

3. Compressed deliveries: No solar panels arrive in the first three months expected, and all arrive in the remaining two months, with the majority arriving in the last expected month. While the deliveries finish in the same month as expected, this schedule causes logistic challenges (Figure 2-6).

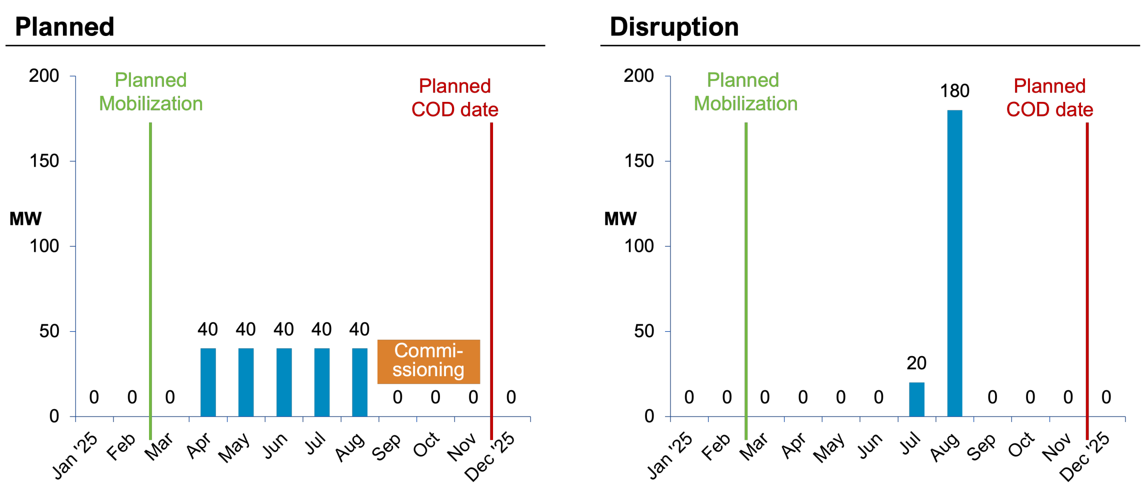

Figure 2-6: Compressed deliveries scenario.

4. Minor changes to technology: The amount of solar panels that arrive each month is the same, but 40% of the panels have technology changes that are considered minor. Examples of these changes are minor revisions to electrical specifications, which do not result in changing the construction layout of the project (Figure 2-7).

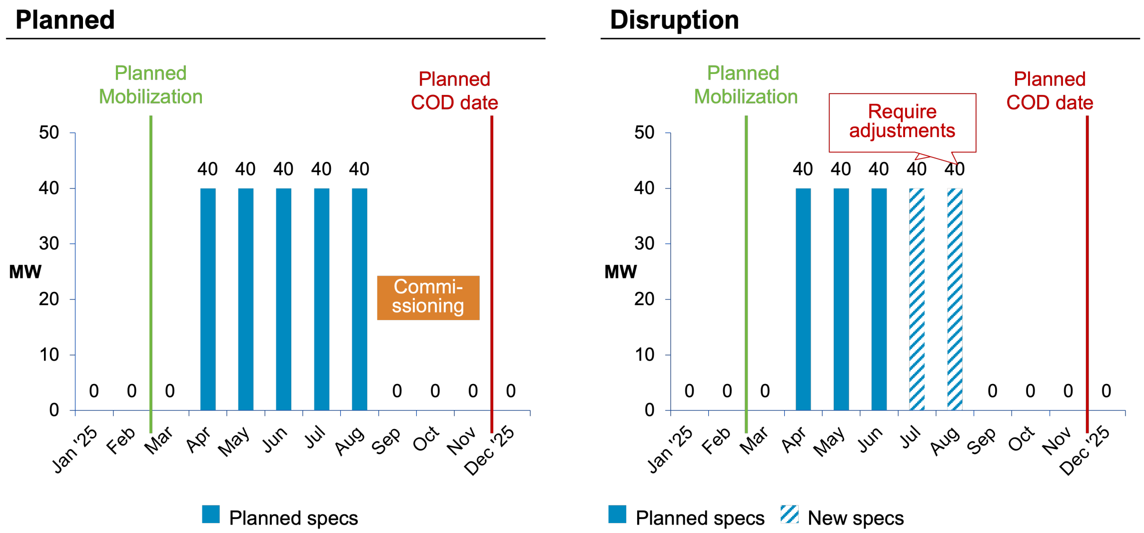

Figure 2-7: Minor technology changes scenario.

5. Major changes to technology: Similarly, the amount of solar panels that arrive each month is the same, but 40% of the panels have technology changes that are considered significant. Examples of these changes are major form factor revisions, which require a different racking solution (Figure 2-8).[3]

[3]The reader can find more information about racking in A.10.3.2.

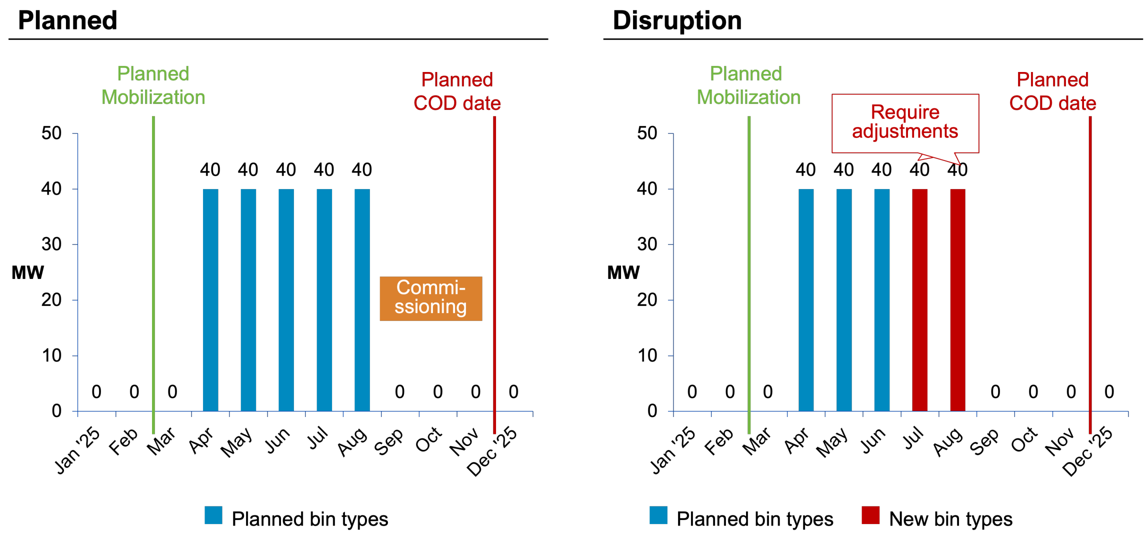

Figure 2-8: Major technology changes scenario.

Based on these figures, the EPC shared potential results from each disruption that NEER team members recorded.

Identifying costs To identify costs, the EPC team brainstormed on the potential costs that a project would face, given different disruptions. Individual team members first identified these costs. Afterward, we gathered and combined them based on similarities. Figure 2-9 presents an example of the individual costs resulting from the exercise.[4]

[4]This figure's objective is not to focus on the post-it details but rather to showcase the nature of the ideation exercise.

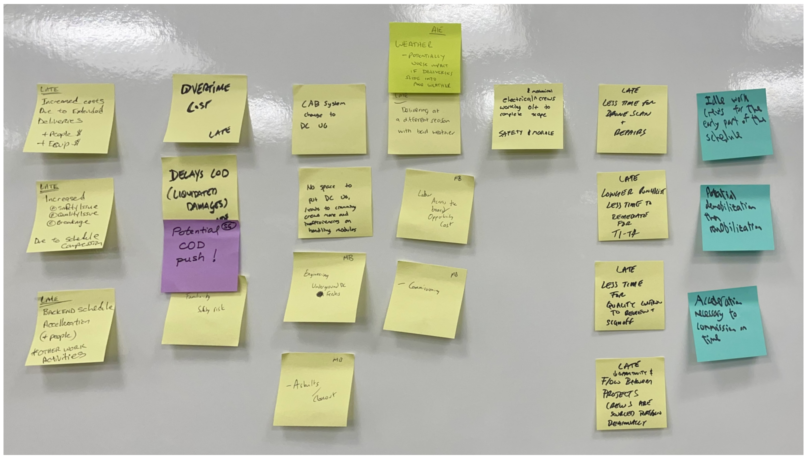

Figure 2-9: Cost ideation with EPC.

We collected 180 cost types from individual team members and combined them into 26 cost types across the team.

Prioritizing costs We prioritized the 26 cost types that resulted in the identifying exercise using the same criteria as in the initial phase. Recall that these criteria were "ease to quantify" and "cost impact." The EPC team identified eight costs as the most important to incorporate into the final model. We incorporated these costs as main costs or part of a cost that we had previously identified. Figure 2-10 shows the results from this exercise.[5] The criteria are labeled on each axis.

[5]This figure's objective is not to focus on the post-it details but rather to showcase the nature of the prioritization exercise.

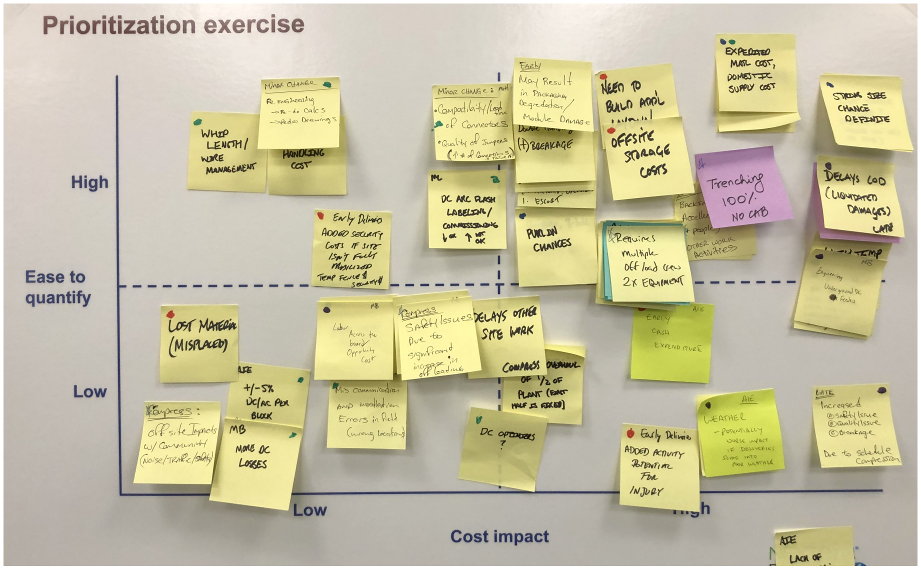

Figure 2-10: Cost prioritization with EPC.

This diagram was later simplified and transcribed into a digital format, shown in Figure 2-11. In this matrix, the colors denote the disruption exercise from which the cost resulted.

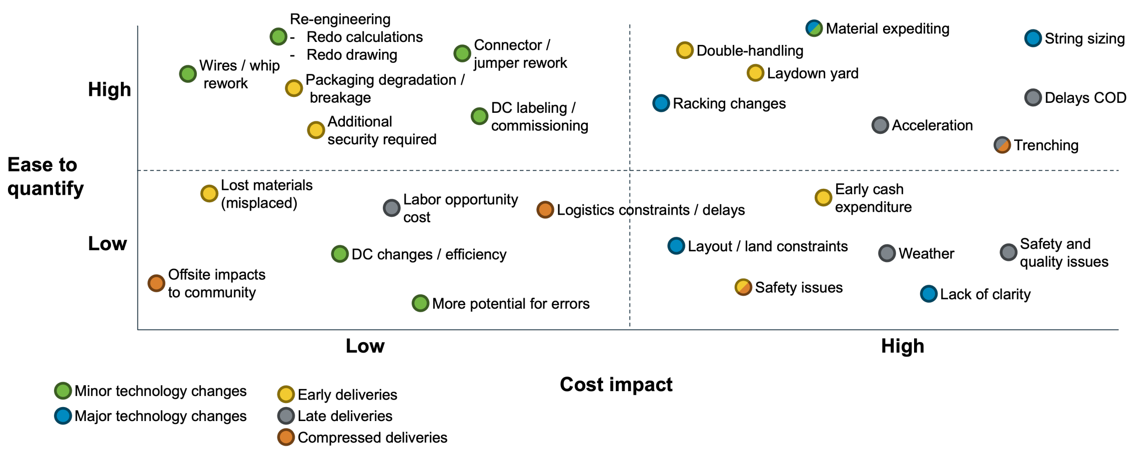

Figure 2-11: Transcribed cost prioritization with EPC.

Estimating costs Several insights on how the disruptions impacted costs resulted from the workshop. One of them was that any changes to the date that panels are delivered would affect the cost. The higher the time difference, the higher the cost will be. Figure 2-12 shows a simplification of this idea.

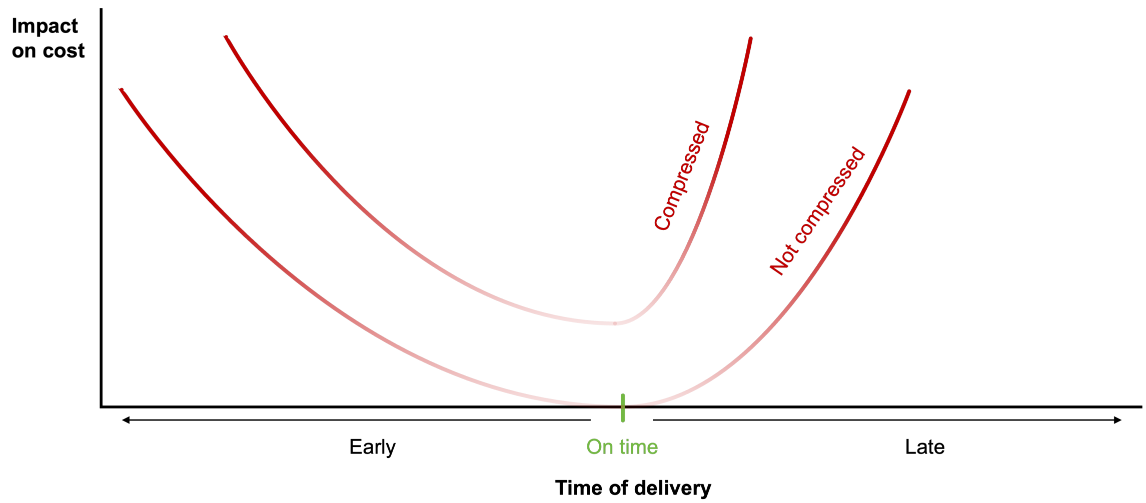

Figure 2-12: Cost impact of the delivery time.

In Figure 2-12, the delivery time is on the x-axis, and the impact on the cost is on the y-axis. The midpoint on the x-axis represents deliveries that are on time. Everything left to the mid-point is delivered before expected and is an early delivery, and everything on the right to the mid-point is a late delivery. Cost increases for late deliveries are generally more impactful than cost increases for early deliveries. If there are compressed deliveries, changes in the time delivered will cause higher costs.

Another insight from the workshop was that the impact will differ depending on when the decision to change the deliveries is made. That is, not only is the degree of the change critical, but the time when the NEER teams and the EPC are notified of the change can also heavily impact the costs. The earlier the change is decided, the more flexibility the teams have to adapt. This concept is simplified in Figure 2-13.

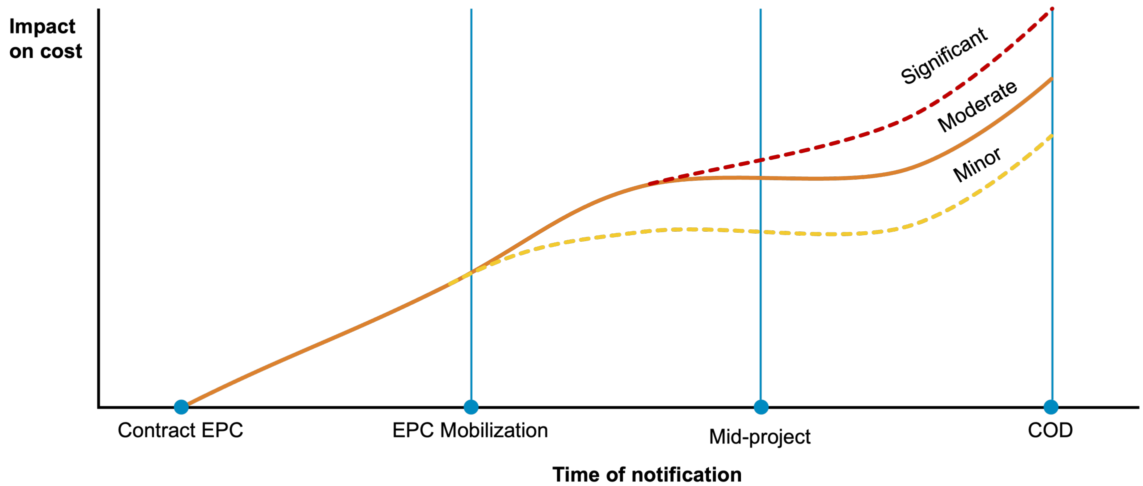

Figure 2-13: Cost impact of the notification time.

As seen in Figure 2-13, the x-axis is the timing of a decision to change the schedule, and the y-axis is the cost impact. In this case, if the decision to change is made before the EPC is contracted, then there are no costs from the external stakeholders. The cost gradually increases until the EPC is mobilized. Until this stage, the magnitude of the change does not impact the cost. However, once the EPC crew has been mobilized, a change of higher magnitude results in a higher cost. In Chapter 3, we will explore how the model approximates the non-linearity shown in Figures 2-12 and 2-13 using piecewise linear functions.

The workshop concluded with the agreement that the EPC would help exhaustively estimate the six essential costs that characterize the majority of the impact incurred when panel disruptions occur. Figure 2-14 shows these costs.

Cost	Cost Coefficient Segmentation
Acceleration	• After / before commissioning start date
Double-handling	• Fixed / variable cost • After / before having access to site
String sizing	• Broken up / uninterrupted site • Cold / hot weather • Per panel type swap combination
Trenching	• Rocky / sandy soil
Material expediting	• Requires air / regular freight
Racking changes	• High / low wind loading • Per panel type swap combination

Figure 2-14: Cost estimation with external stakeholders.

Collaboratively, we defined each cost using a formula, a coefficient based on segmentation, and a scaling unit that the EPC provided. We used the cost coefficient segmentation when different categories of sites required different coefficients. For example, there are some instances when changing a project's panel bin type results in the EPC having to create trenches to place the cables that connect the solar panels underground. In this case, the cost would be much higher if the project site had rocky soil than if it had sandy soil because it would require higher power equipment to create the trenches. This is one of the segmentations we used to make the cost coefficients more precise.

The cost scaling unit is the variable the coefficient multiplies in the model. Taking trenching as an example, if the cost is to be calculated in an actual project, it will be calculated by multiplying the coefficient by the meters of cable that need to be placed underground. Since this calculation depends so much on the characteristics of each project, the team decided to simplify the calculation by using an abstraction based on

MW_{dc}.[6] For each MW_{dc} requires a change, the EPC trenches certain meters of cable. We can approximate this calculation for all projects using the average cable meters trenched per MW_{dc}. Therefore, the cost can then be scaled with MW_{dc} as a unit.

We took a similar approach to calculate the rest of the cost coefficients. In the follow-up sessions with the EPC, an explicit formula for each cost was produced, presented in Chapter 3 and Appendix A.

2.1.2 Model development

The main activities during the model development were the creation of the model formulation and its translation into code. The details of this formulation are not presented here. We present them in Chapter 3 and Appendix A. However, since the model will be used in a real-life application that requires quickly getting to the optimal solution, we had to make significant technical efforts to ensure that the model could run at scale within the required time frame of less than two hours. We describe these efforts next.

2.1.2.1 Boost processing performance

Solvers We considered two options when choosing the computational solver: CBC (COIN-OR branch-and-cut) [23] and Gurobi [24]. The former is an open-source solver based on the branch-and-cut algorithm. We chose it for its flexibility and easy access[25]. CBC solved the initial problems to optimality at a fast rate. However,

[6] An electric current can be direct or alternating [21]. The alternating and direct currents are abbreviated as AC and DC, respectively. Since solar panels generate electricity in the form of direct current [22], NEER plans their projects using MW_{dc} units. As the electric grid uses MW_{ac}, NEER converts DC to AC using inverters to meet customer needs. They account for the loss in the conversion by having more MW_{dc} than the MW_{ac} the customers are expecting. The remainder of this thesis will use MW and MW_{dc} interchangeably to refer to MW_{dc}.

once the model was required to run at full scale (around 90 projects), the solving time using CBC was more prolonged than needed (more than 2 hours), and we sought an alternative. We chose Gurobi, a leading commercial solver. NEER was required to complete legal and procurement processes to use Gurobi. Once they were finished, the solver reached optimal solutions in the time magnitude of minutes despite the problem's scale.

Decomposition sequential approach This approach sought to break the main problem into more basic problems and then iteratively add variables and constraints to build the entire problem progressively. The model first found a solution to a more straightforward problem quickly. Then, when more variables and constraints were added to the problem, this "hot start" solution was used as a starting point to solve the next step in the process. We constructed the sequence to add variables and constraints into the model so that the hot starts would always be a feasible solution for the next step, albeit perhaps not optimal. Appendix B explains the details of the compositional sequential approach.

2.1.3 Data management

2.1.3.1 Input format

The internal data comprised the current and historical delivery schedule, the contracted volume per supplier, the production schedule per supplier, and the cost-estimating assumptions. We gathered these data through the interviews mentioned before with the Pricing & Analytics project sponsor team and other internal stakeholders. The external data were the cost-estimated assumptions resulting from the previously described collaboration with the EPC.

We compiled both data sources into a series of Excel spreadsheets, which served as the input format for all parameters in the model. We used Excel for easy access and availability to users with less coding experience. In future iterations of the project, the input data will be uploaded into a server that is automatically updated to avoid the need for manual computations in Excel.

The input data for the initial phase included nine projects and two types of panel bins. We expanded the input data for the final phase into 86 projects and 31 types of panel bins.

2.1.3.2 Preprocessing workflow

The preprocessing workflow uses R to transform the Excel input file into a format that can be used to define the model with the given parameters. For context, the model formulation first defines an abstract model, and only when it is populated with parameters does it become a concrete model [26]. For example, the abstract model could be defined as having projects in general, but the concrete model may be determined to have projects A, B, and C specifically. This gives the model the flexibility to change with new parameters based on an input source without having to make changes to the code. However, the data must have a specific format, so we developed a script during the project to adjust the input to this format.

The workflow first reads the data from Excel as a CSV into R. It then does a series of cleaning and processing steps, including removing projects with incomplete values and calculating implicit parameters used in the model. Figure 2-15 shows the broad steps in the preprocessing workflow.

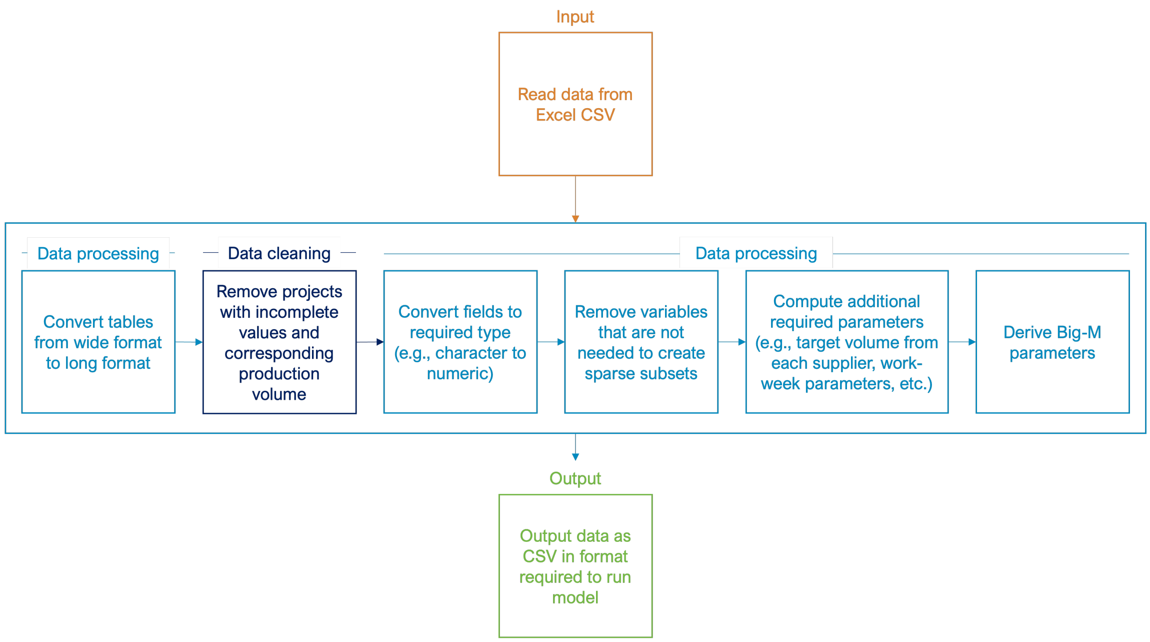

Figure 2-15: Preprocessing workflow diagram.

2.1.3.3 Output workflow

We produced a series of outputs to aid the interpretation of the model results. These consisted of tables and visualizations with the optimized schedule and the breakdown of the costs incurred by each project. We post-processed the output in Python and created the visualizations in Tableau. Figure 2-16 presents the output process.

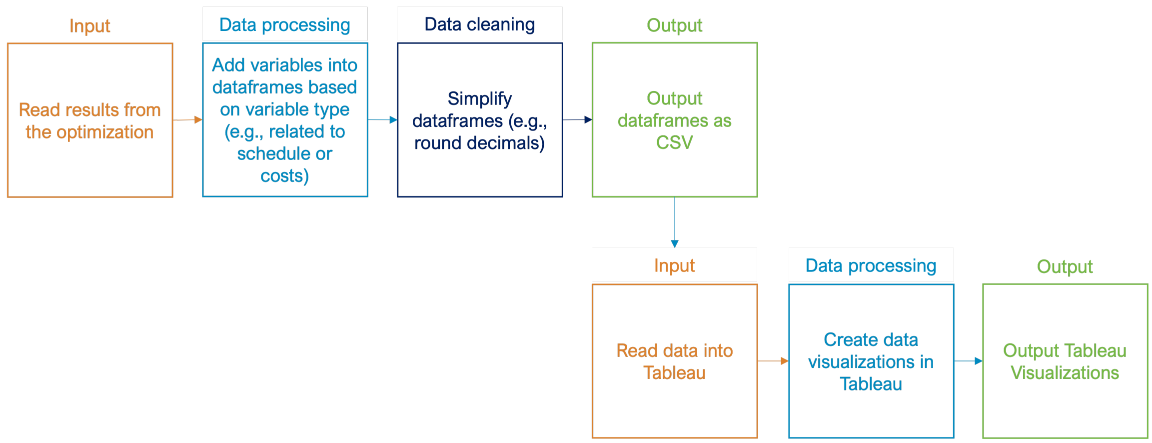

Figure 2-16: Output workflow diagram.

2.1.3.4 Reduce solution space

When increasing the size of the model in the refinement phase, we made an effort to reduce the size of the solution search space to avoid high runtimes. Two approaches were implemented and are presented next.

Calculate minimal big-$\mathcal{M}$ The model uses many variables whose value is dependent on that of another variable. These variables can be calculated using logical statements of the form if-else. However, the if-else statements are non-linear and must be modified for a linear program. This model uses the big-$\mathcal{M}$ method, which is widespread in mixed-integer linear programming, to solve for these variables. It introduces a parameter $\mathcal{M}$ large enough to make some constraints redundant [27]. Appendix A.1 presents a detailed explanation of how this method works.

While the method is called the big-$\mathcal{M}$ method, having a larger $\mathcal{M}$ than needed causes problems when searching for a solution [28]. The larger the $\mathcal{M}$ is, the larger the linear programming solution space will be. As such, for every $\mathcal{M}$ that we used in the model, we calculated the minimal possible $\mathcal{M}$ so that the problem remained feasible. For example, the model uses a big-$\mathcal{M}$ to determine if deliveries have been finished for each project. In this case, the big-$\mathcal{M}$ parameter takes the value of the maximum deliveries that a project can have plus one.

Create sparse subsets Some sets were made more sparse to avoid using unnecessary variables. Only the variables and constraints that could happen were declared, thereby reducing the model size. An example of how the sparse subset of a set was created was the delivery of certain bin types. Suppose that bin type A panels have never been produced before the fifth period. Then, panels of this bin type cannot be delivered to projects in the first four periods. In this case, we can remove all variables that denote that the bin type A panels are delivered to a project in the first four periods. Analogously, all other bin types do not need a variable pertaining to the delivery of panels that have not been produced before. Similar efforts were made to decrease the size of the variable sets.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Solar panel allocation model

The solar panel allocation model seeks to minimize the costs arising from creating a new schedule after a disruption to the original schedule. This chapter progressively introduces the components required to calculate the costs in the objective function, including sets, variables, parameters, and constraints.

3.1 Objective function

The objective function is defined as the sum of the costs that NEER could incur due to a disruption:

$$
\begin{aligned}
\min \quad \sum_{p,t,b} f(x_{p,t,b}) = & \sum_{p,t,b} c\_new\_buy_{p,t,b} + \\
& \sum_{t,b} c\_expedite\_supply_{t,b} + \\
& \sum_{p} c\_additional\_crew_{p} + \\
& \sum_{p,t} c\_compressed_{p,t} + \\
& \sum_{p} c\_late\_deliveries_{p} + \\
& \sum_{p,t} c\_early\_deliveries_{p,t} + \\
& \sum_{p} c\_remob_{p} + \\
& \sum_{p} c\_change\_type_{p}
\end{aligned} \tag{3.1}
$$

where the subscripts p, t, and b represent the project, period, and bin type that define the solar panel deliveries, respectively.[1]

Based on the structure of the objective function, this chapter is divided into nine sections, as shown in the following list, along with a short description of the components of each section.

3.2 Delivery basics: Presents the fundamental delivery variables in the model and primary constraints tied to them. The model derives all other variables and constraints from those in this section. $c\_new\_buy_{p,t,b}$ is included in this section.

[1]These subscripts are defined in detail in the next section.

3.3 Minimum MW per bin type and form type: Introduces the components that ensure that projects receive at least the minimum amount of MW per bin type and form type depending on the characteristics of the project.

3.4 Supplier capacities: Defines the components that handle the potential decisions arising from a supplier with limited or surplus production capacity. $c\_expedite\_supply_{t,b}$ is defined in this section.

3.5 Changes to the weeks that the EPC crew works: Incorporates the concept of work weeks and shows how changes in the deliveries can affect the number of planned work weeks ($c\_additional\_crew_p$).

3.6 Compressed deliveries: Calculates the cost of delivering too many MW to a project in a limited time ($c\_compressed_{p,t}$).

3.7 Late deliveries: Implements the logic required to address the consequences of deliveries arriving later than expected due to a disruption ($c\_late\_deliveries_p$).

3.8 Early deliveries: Implements the logic required to address the impact of deliveries arriving on a date earlier than expected due to a disruption ($c\_early\_deliveries_{p,t}$).

3.9 Remobilization: Introduces the constraints to compute the cost of not having deliveries for multiple periods, resulting in having to demobilize the workforce first to remobilize it again later (c_remob_p).

3.10 Bin type changes: Explains how the model builds the capabilities so that projects can exchange panels between projects, discard panels, and buy new ones ($c\_change\_type_p$).

This chapter describes the costs that are part of the objective function in its remainder. Appendix A gives a more detailed explanation of the concepts presented in this chapter. Figure 3-1 can help the reader visualize the key milestones a project goes through and how the milestones segment the costs in each section.

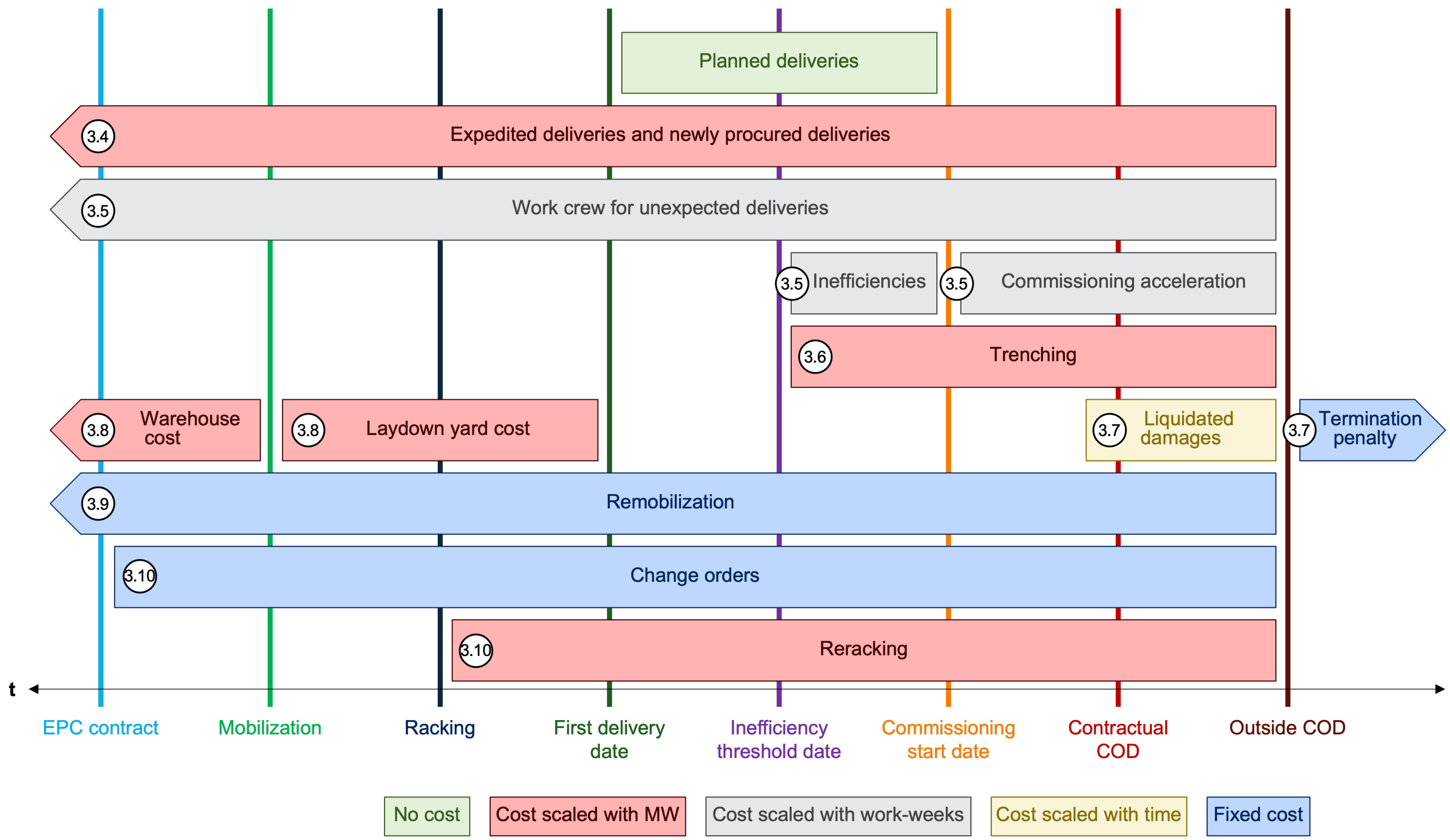

Figure 3-1: Timeline of a typical project schedule with key milestones.

3.2 Delivery basics

The model's fundamental variables determine when and how many panels from each supplier should be delivered to each project to produce the output delivery schedule. We call these variables x, and all other variables depend on them. The subsequent sets index the x variables:

- Set $\mathscr{P}$, a list of the names of projects p to which the suppliers deliver panels.
- Set $\mathscr{T}$, which is a list of the periods $t \in \mathbb{Z}$ during which the decisions in the model take place.[2]
- Set $\mathscr{B}$, which represents the bin type b for solar panels. A bin type classifies the solar panels to be delivered based on the supplier's name, watts per module, and production year.

Based on these index sets, $x_{p,t,b} \in \mathbb{R}^{\geq 0}$ represents the MW allocated to project p in time t using panels with bin type b.[3]

3.2.1 Pre-negotiated contracts and new procurement

Using the x variables as a base, we add more constraints to the model. We determine if the panels sent to a project come from a pre-negotiated contract or directly from a supplier without a pre-existing contract. In other words, a project has two potential sources to receive MW: the negotiated volume or the new volume the project buys. The model introduces these sources using the variables $x\_from\_contract_{p,t,b}$ and

[2] This thesis uses months as the periods. However, other periods, including weeks and days, could be used with the same model formulation.

[3] More details about bin types and how they impact the design of a project site are presented in Appendix A.10.2.

$x\_new\_buy_{p,t,b}$. Explicitly, $x\_from\_contract_{p,t,b} \in \mathbb{R}^{\geq 0}$ is the amount of MW sourced from a contract and allocated to project p in time t using panels with bin type b. $x\_new\_buy_{p,t,b} \in \mathbb{R}^{\geq 0}$ is the amount of MW bought outside of a contract and allocated to project p in time t using panels with bin type b. They constrain the x using:

$$x_{p,t,b} = x\_from\_contract_{p,t,b} + x\_new\_buy_{p,t,b} \qquad \forall p, t, b \qquad (3.2)$$

The model does not account for the solar panel cost since NEER already negotiated and paid for the $x\_from\_contract_{p,t,b}$ volume.[4] Across every period, we introduce the ensuing constraint to ensure that projects cannot source more MW from a contract than those that are actually in the contract:

$$\sum_{p,t} x\_from\_contract_{p,t,b} \leq \sum_{p,t} contracted_{p,t,b} \quad \forall b \qquad (3.3)$$

Without constraint (3.3), a project could source unlimited free volume.

On the other hand, NEER still needs to pay a cost associated with $x\_new\_buy_{p,t,b}$. The model estimates it using the variable $c\_new\_buy_{p,t,b} \in \mathbb{R}^{\geq 0}$, defined as the cost of buying new panels with bin type b outside of a contract to deliver them to project p in time t.

We calculate the cost of buying additional MW outside of a contract by multiplying the amount of MW that projects bought times a cost coefficient. The model defines this cost coefficient as $coeff\_new\_buy_{t,b} \in \mathbb{R}^{\geq 0}$, or the cost per MW of buying solar panels outside of a contract from a supplier with the production capacity to deliver

[4] While the cost of these panels may not truly be sunk, the assumption is that the financial and reputational cost to terminate these panels and re-procure is such that NEER will always prefer to utilize existing contracts before adding incremental contracted volume.

panels with bin type b in time t. That is, the cost of buying new deliveries has a linear relationship with the volume bought. With this in mind, the model calculates $c\_new\_buy_{p,t,b}$ with the constraint:

$$c\_new\_buy_{p,t,b} = coeff\_new\_buy_{t,b} \cdot x\_new\_buy_{p,t,b}, \qquad \forall p, t, b \tag{3.4}$$

3.2.2 Minimum delivery

The $min\_delivery\_amount \in \mathbb{R}^{+}$ parameter states the minimum amount of MW of each bin type b that can be delivered to any project at any time t. Since the unit for x is MW and not a number of panels, we want to avoid situations where the model allocates unrealistically low MW to a project.

We constrain the model to allocate at least this minimum amount of MW if it allocates any panels to any project in a certain period. In other words, deliveries will be either 0 or a value greater than or equal to the minimum amount, avoiding deliveries of an unrealistically small MW amount. The reader can find details behind the use of MW as a unit and the constraints that enforce the minimum delivery logic in Appendix A.2.1.

3.2.3 Past decisions

Another aspect that the model takes into account is that some decisions have been made in the past that cannot be changed after the model runs. For example, the model cannot change past deliveries to a project. As such, it introduces the parameter $t\_current \in \mathbb{Z}^{\geq 0}$, which is the period at which the algorithm runs. Periods smaller than $t\_current$ are in the past, and periods larger than $t\_current$ are in the future. Generally, the model can only make decisions for periods in the future.

To enforce the logic of past deliveries, the model uses the *delivered* variables. The model defines $delivered_{p,t,b} \in \mathbb{R}^{\geq 0}$ as the amount of MW a supplier delivered to project p using panels with bin type b in time t. If a period is in the past, then the MW that a project allocates in the model should be exactly equal to what it delivered during that period.

$$x_{p,i,b} = delivered_{p,i,b} \quad \text{for } i = 1, ..., t\_current, \quad \forall p, b \tag{3.5}$$

Notice that the $delivered_{p,t,b}$ variables help reduce the decision space because they constrain x to a single value. Similarly, a project cannot buy new MW of a bin type if a period has passed:

$$x\_new\_buy_{p,i,b} = 0 \quad \text{for } i = 1, ..., t\_current, \quad \forall p, b \tag{3.6}$$

3.2.4 Completing projects

We introduce the concept of whether NEER can and should complete all projects. If a project receives all the MW required, we say we complete it. In other words, each project has a required total number of MW to receive to complete, or NEER must terminate the project. We say we terminate the project if it gets less MW than needed. The parameter $MW_p \in \mathbb{R}^{\geq 0}$ defines the number of MW worth of panels to complete project p, also known as project demand.

To track if the model terminates projects or not, it defines the $terminate_p \in \{0, 1\}$ binary variable, which takes the value of 1 if the model recommends terminating project p and 0 if it recommends completing it. The model introduces the concept with the subsequent constraint:

$$\sum_{t,b} x_{p,t,b} \geq MW_p \cdot (1 - terminate_p) \qquad \forall p \qquad (3.7)$$

On the left-hand side of the constraint, we have the MW that project p will receive from all of the suppliers in all periods. If the sum equals or exceeds the MW required to complete the project, the variable $terminate_p$ can take values one or zero. It is unconstrained. However, if p receives less MW than required to be completed, then $terminate_p$ will necessarily equal one. The termination of a project results in a high cost. We will dive into its details in a future section.

3.3 Minimum MW per bin type and form type

In most cases, a project contracts only panels with one bin type. Having one bin type simplifies many aspects of the project development process. For example, the delivery routes are the same for all panels, the entire project has the same hardware and electric requirements, and the installing crew does not need to learn installation nuances specific to different bin types.

However, there are instances in which having multiple bin types in one solar site can be beneficial or even necessary. An example would be if NEER can no longer import panels with a particular bin type for a project that has already had deliveries. In this case, NEER would need to import new panels with a different bin type to complete the project. As we allow some flexibility for projects to have panels with varying bin types, it is crucial to avoid a situation where we have an excess of bin types with very little MW. This section introduces the constraints that handle situations like the one just mentioned.

Additionally, we introduce a new concept: the panel form type. A form type classifies the panels to be delivered based on the watts per module and the panels' paradigm or shape. The set $\mathscr{F}$ encodes the list of form types used in the model. A form type is a broader category that can classify multiple bin types under it. In other words, the model can classify several bin types under the same form type, but it cannot classify several form types under the same bin type.

3.3.1 Minimum MW per bin type

First, we introduce the parameter $minper\_bin \in \mathbb{R}^+$, which is the threshold that establishes how many MW per bin type is each project's minimum. In other words, if a project receives any MW from a bin type, it has to receive at least $minper\_bin$. The following example can help us visualize the logic behind the constraints in this section. Suppose we complete projects A, B, and C with 140MW, parameter $minper\_bin = 40$, and the projects can receive panels with bin type 1, 2, or 3. Table 3.3.1 and Figure 3-2 illustrate how the projects allocate the volume by bin type.

p	$\sum_t x_{p,t,1}$	$\sum_t x_{p,t,2}$	$\sum_t x_{p,t,3}$
A	140	0	0
B	52	48	40
C	120	0	20

Table 3.3.1: Example of a schedule of three projects with deliveries of different bin types.

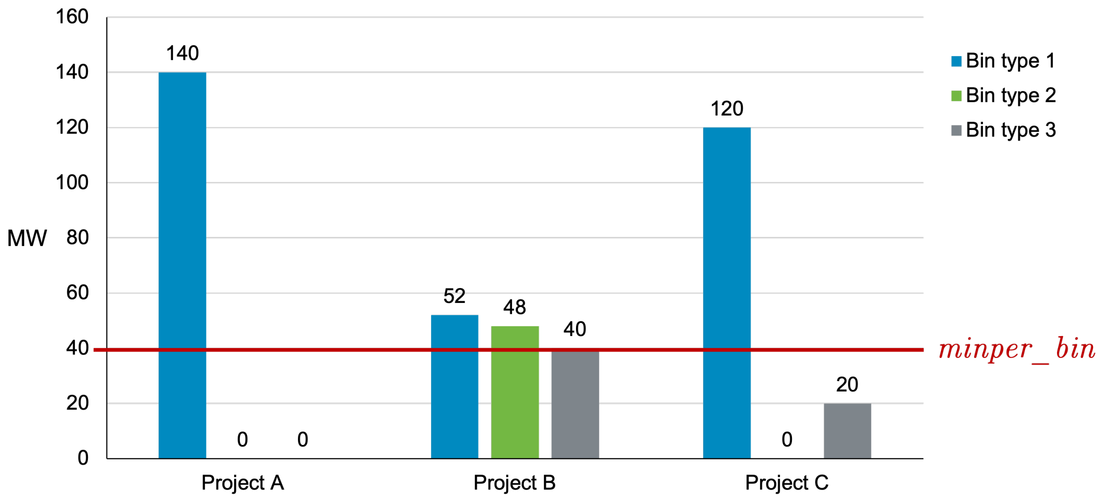

Figure 3-2: Example of a schedule of three projects with deliveries of different bin types.

In this example, projects A and B have a valid allocation. Project A uses only bin type: bin type 1. It is not a problem since the bin type it uses receives more MW than the minimum ($140 > minper\_bin = 40$). Project B uses all bin types: 1, 2, and 3. Again, this is not a problem since it receives more than the minimum MW with all bin types. However, project C does not have a valid allocation since it is trying to receive 20MW, which is less than the minimum. In Figure 3-2, the stacked bar from project C and bin type 3 is below the red $minper\_bin$ line. We can generalize this visualization into a rule where every positive bin type amount must touch the red $minper\_bin$ line for a project's allocation to be valid. If an amount is below the line, then it must be zero.

Although this concept is simple, there are some exceptions to consider, which result in the following rule: for every bin type a project uses, it must receive at least the minimum established quantity unless it has less than the minimum quantity overall, or it must have a contract that states otherwise. The reader can find the constraints that implement this rule in Appendix A.3.1.

3.3.2 Minimum MW per form type

Similar to the previous subsection, projects prefer to have panels with one form type, but there is flexibility in having multiple form types. All constraints for the minimum MW per form type follow the same structure as the minimum MW bin type constraints with some modifications. Additionally, a project must receive at least the minimum established quantity per form type for every form type it uses unless it has less than the minimum quantity overall, or it must have a contract that states otherwise. We present the constraints to codify the minimum MW per form logic in the model in Appendix A.3.2 to avoid repetition.

3.4 Supplier capacities

In a given period, a supplier might have the capacity to produce fewer solar panels than it had initially planned. Its production might be limited by external causes like weather or trade controls or by internal causes like the efficiency in a factory. Limitations can result in the supplier's failure to deliver part of its contract. In parallel ways, its production may be increased. Therefore, NEER's deliveries can be limited by a supplier's restricted production, but they can also be expanded using the supplier's overcapacity.

The main cost in this section is $c\_expedite\_supply_{t,b} \in \mathbb{R}^{\geq 0}$, which is defined as the cost per MW of expediting the delivery of panels with bin type b so that the supplier can deliver them in time t. The cost of expediting deliveries scales linearly in the expedited volume. Expediting deliveries is different from procuring new supplies because the expedited deliveries use volume that is part of the pre-negotiated contract. While there is a cost of expediting panel deliveries since the supplier will have to rush to ship them before they are established in the contract, it tends to be less than the cost of procuring new panels.

The rest of this section introduces two examples to capture the volatility of the suppliers' production capacities and the costs that arise from it.

3.4.1 Limited supply

First, suppose that a project that is complete with 160MW and a supplier that sends 40MW in periods 1 to 4 (illustrated in Figure 3-3).

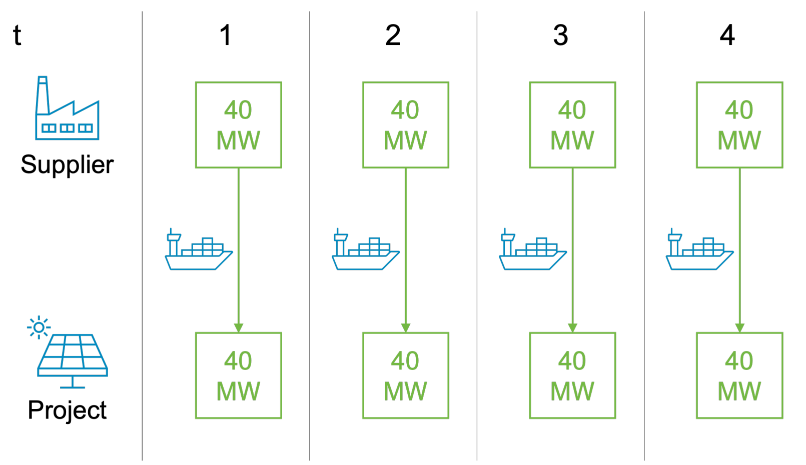

Figure 3-3: Example of a simple schedule without a disruption.

Suppose that in period 4, one of the machines in the supplier's factory broke down, and they could not send the scheduled 40MW, so they sent them in period 6 (illustrated in Figure 3-4).

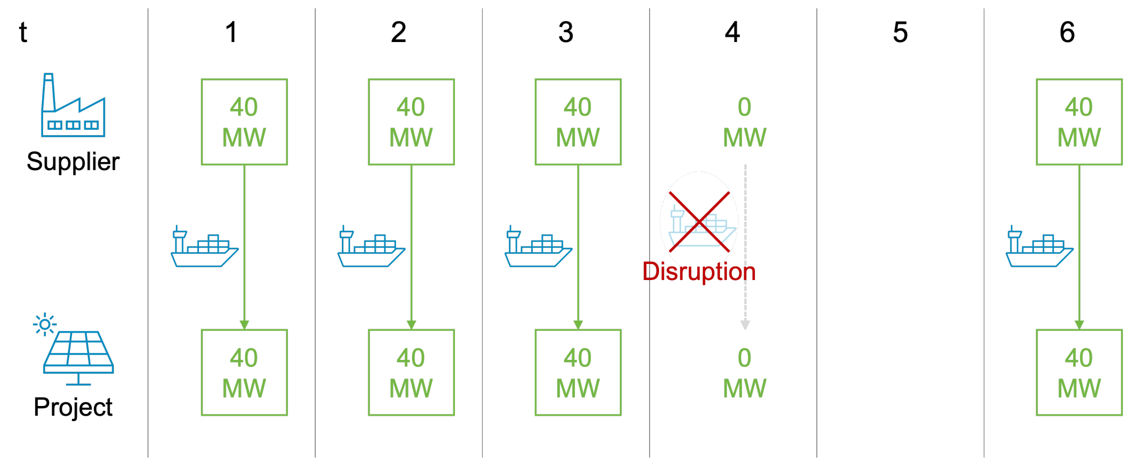

Figure 3-4: Example of a simple schedule with a disruption.

Then, in this example, the project could not receive the scheduled MW because of the supplier's production capacity limitations. We introduce the parameters $expected\_production_{t,b}$ and $contracted_{p,t,b}$ to model instances similar to the example. Specifically, $expected\_production_{t,b} \in \mathbb{R}^{\geq 0}$ is the amount of MW the suppliers expect to produce and have ready for delivery in time t to any project using panels with bin type b. Note that $expected\_production_{t,b}$ is not related to any specific project, but it is the overall supply of that bin type in time t. $contracted_{p,t,b} \in \mathbb{R}^{\geq 0}$ is the amount of MW project p contracted for delivery in time t using panels with bin type b. We also introduce the constraint coming up to ensure that the model does not allocate more MW across all projects in any period than the overall expected production for each bin type:

$$\sum_{p} x_{p,t,b} \leq expected\_production_{t,b} \qquad \forall t, b \qquad (3.8)$$

In the example illustrated by Figure 3-4, we would have the values for the $contracted_{p,t,b}$ and $expected\_production_{t,b}$ and the variable x shown in Table 3.4.1.

t	1	2	3	4	5	6
$expected\_production_{t,b}$	40	40	40	0	0	40
$contracted_{p,t,b}$	40	40	40	40	0	0
$x_{p,t,b}$	40	40	40	0	0	40

Table 3.4.1: Values of an example of a simple schedule with a disruption.

3.4.2 Expedited contract supply

In the previous scenario, the supplier owed the project 40MW in period 6, so it did not charge an additional cost for that delivery. Let us explore a scenario where the supplier can meet its delivery obligations for this and future periods. Again, the project is complete with 160MW, and the supplier schedule is to send 40MW in periods 1 to 4. However, in this example, the model decides that delivering all panels in period 1 is better, and the supplier can do so. Figure 3-5 illustrates this example, and Table 3.4.2 presents its values.

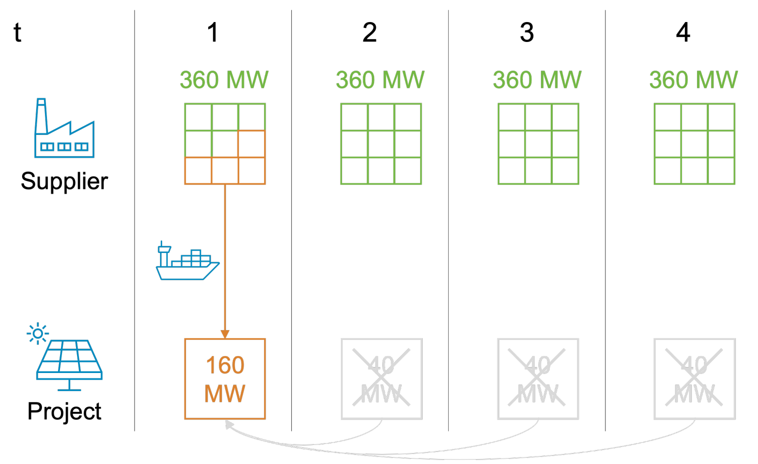

Figure 3-5: Example of simple schedule with expedited deliveries.

t	1	2	3	4
$expected\_production_{t,b}$	360	360	360	360
$contracted_{p,t,b}$	40	40	40	40
$x\_from\_contract_{p,t,b}$	160	0	0	0

Table 3.4.2: Values of an example of a simple schedule with expedited deliveries.

For additional explanations of how the $c\_expedite\_supply_{t,b}$ variable is calculated, refer to Appendix A.4.1.

3.5 Changes to the weeks that the EPC crew works

When an EPC is contracted, NEER shares the original delivery schedule with them so that they can plan the size of the work crew they require in each period to receive and install the solar panels and commission the project. If there is a disruption, the labor necessary for product development will likely change. However, the EPC tends to have the flexibility to move crews around to different projects, resulting in fewer additional costs for NEER. On a high level, NEER incurs a fee related to the work crew if it makes changes that result in more crews than they had initially contracted.

The model uses the "work weeks" unit to standardize the number of crews needed to receive and install the panels in all the costs related to additional labor. A work week captures the labor required to receive and install a certain number of solar panels in one week. Typically, a standard crew can receive and install 10MW in one week. As such, contracting one work week is equivalent to hiring a crew to receive and install 10MW. Two work weeks would be required to receive and install 20MW. Depending on the project's needs, the same amount of work can be done in more or less periods. That is, installing 20MW can be done in one week if two crews work simultaneously or in two weeks if one crew is working each week, but both cases have used two work weeks.

The model calculates the cost of additional labor required for each project based on additional work weeks, inefficiency weeks, and weeks where commissioning needs to be accelerated. Figure 3-6 shows a timeline of when each labor cost is activated.

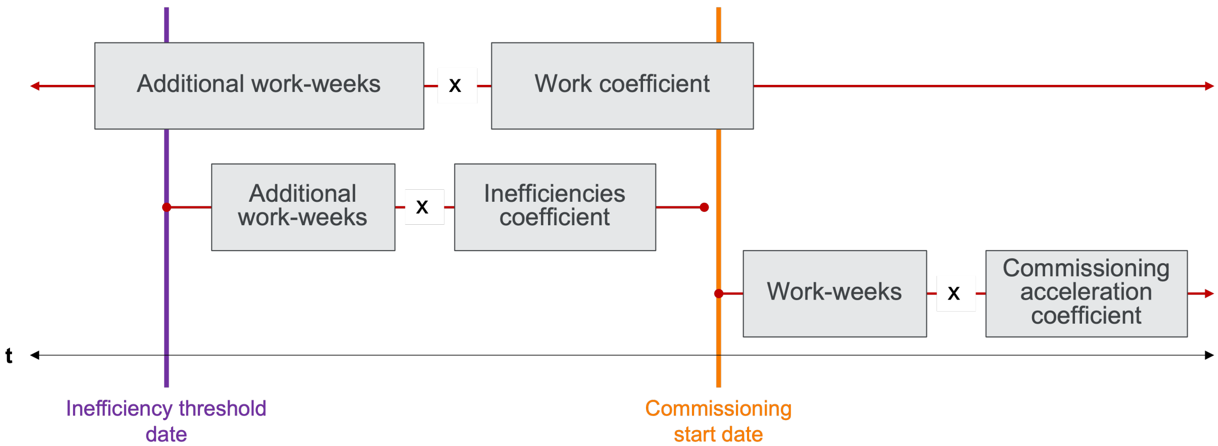

Figure 3-6: Work weeks costs timeline.

To summarize these concepts, the model uses the $c\_additional\_crew_p \in \mathbb{R}^{\geq 0}$

variable and determines it through the following constraint:

$$
\begin{aligned}
& c\_additional\_crew_p = \\
& \quad coeff\_additional\_work\_week \cdot additional\_work\_weeks_p + \\
& \quad \left(\sum_{t} coeff\_inefficiencies \cdot inefficiency\_weeks_{p,t} \right) + \\
& \quad \left(\sum_{\substack{t > t\_current \\ t \geq t\_comm_p}} coeff\_commissioning\_acceleration \cdot required\_work\_weeks_{p,t} \right) \\
& \hspace{20em} \forall p \qquad (3.9)
\end{aligned}
$$

The cost coefficients that partake in the previous constraint are:

- $coeff\_additional\_work\_week \in \mathbb{R}^{\geq 0}$ is the cost of hiring an additional crew for one week's work.

- $coeff\_inefficiencies \in \mathbb{R}^{\geq 0}$ is the cost of having one week's worth of work with inefficiencies. A period has inefficiencies if it is after or during the inefficiency period ($t\_inefficiency$) and requires more work weeks than contracted initially.

- $coeff\_commissioning\_acceleration \in \mathbb{R}^{\geq 0}$ is the cost of hiring a crew for one week's work to accelerate the commissioning. A project must accelerate the commissioning if it has deliveries after the commissioning start period (t_comm).

Appendix A.5 explains how the components of the $c\_additional\_crew_p$ variable are calculated, and it presents an example that captures the compilation of the concepts in this section.

3.6 Compressed deliveries

Having more deliveries than expected affects how the EPC installs panels. If the project crew is tasked with receiving more panels than they usually manage, they will need to compress the deliveries. Compressing deliveries in this context means implementing measures allowing a crew to receive more MW than anticipated. Some examples of these measures are digging ground trenches to place the cables that connect the panels underground (these typically go above ground) and working overtime.

There is a maximum delivery size in MW beyond which the model defines deliveries as compressed. It is defined in the model using the *compressed_thresh* parameter. In Figure 3-7, we show an example of compressed deliveries where the area shaded with red represents the zone with compressed deliveries. A period can receive a maximum of *compressed_thresh* MW in one period without incurring a cost for a high volume of deliveries.

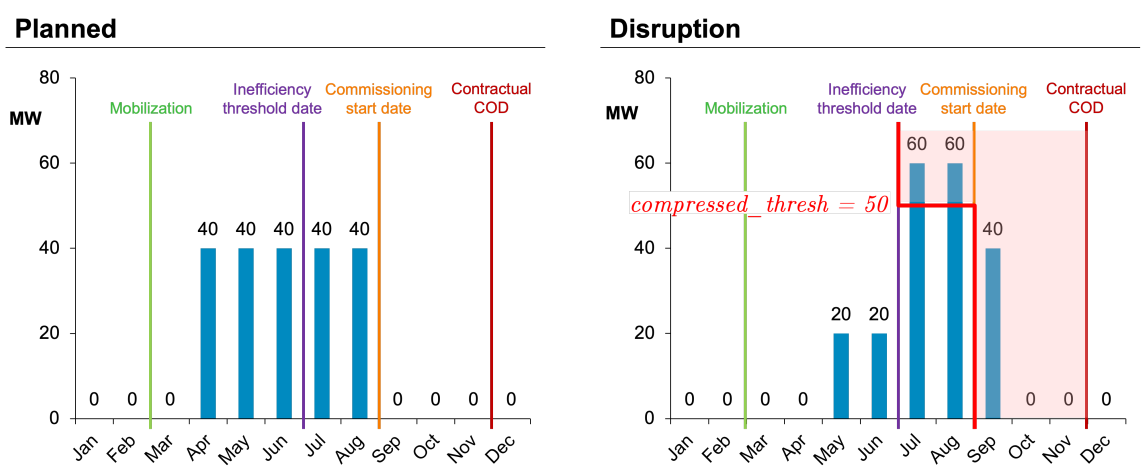

Figure 3-7: Example of compressed deliveries in a project schedule after a disruption.

It is worth noting that while a project may want to receive a large amount of MW, there is a limit to the MW it can receive per period. In other words, a project can receive up to a set amount per period, even if the project is willing to pay an additional cost. The threshold of the maximum amount of MW a project can receive in one period is $maxreceive \in \mathbb{R}^+$.

That way, $compressed\_thresh$ acts as a soft maximum for the number of MW a project can receive in one period, whereas $maxreceive$ is a hard maximum. In other words, NEER can pay for delivery sizes that exceed $compressed\_thresh$ but cannot exceed $maxreceive$ under any circumstance. By definition, $maxreceive \geq compressed\_thresh$. Figure 3-7, along with the rest of the constraints used to calculate the cost of compressed deliveries, $c\_compressed_{p,t}$, are explained in more detail in Appendix A.6.

3.7 Late deliveries

The model includes two costs resulting from having delays in panel deliveries: Liquidated Damages (LD) and termination. The LDs refer to the penalty that NEER has to pay to a customer if they do not finish a project by the contractual COD. Not finishing a project results in a breach of contract, so NEER monetarily compensates the customer by paying LDs every period NEER is late. The termination penalty is the compensation that NEER has to pay the customer if they do not finish the project, that is, if they terminate the project. We define a project as terminated if it does not finish deliveries by the outside COD. Figure 3-8 shows the timeline of when NEER incurs each cost.

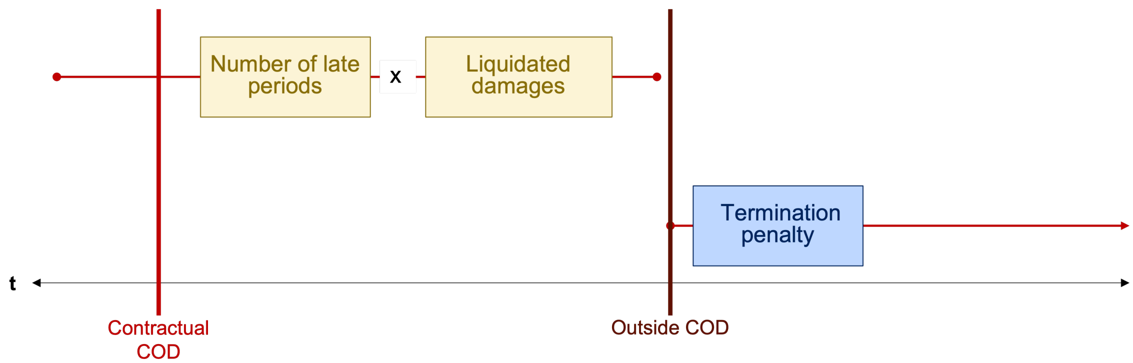

Figure 3-8: Late deliveries cost timeline.

Appendix A.7 analyzes the costs of LDs and termination penalties in detail.

3.8 Early deliveries

Suppose a project makes deliveries before the EPC mobilization date, which we define as the date the EPC arrives on-site, as paid by NEER. In that case, the supplier must deliver the panels to a warehouse because the EPC does not yet have access to the solar site. Since the MW are stored in a warehouse, NEER will have to pay for cumulative deliveries each month. The cumulative nature of the cost results from the MW needing to be stored in the warehouse from their delivery until the EPC has access to the project site. So, if a project stores 10 MW starting period 1 and the EPC mobilization date is in period 5, it will have to pay to store 10 MW for five periods.

Suppose a project receives deliveries before its first contracted delivery period. In that case, the EPC will need to build a laydown yard since the assigned location for the panels will not be ready. A laydown yard is an empty plot of land inside the project where the EPC can unload the solar panel packages from the containers where they arrived. It tends to be a large yard towards the far edges of a project. The cost of a laydown yard is not cumulative because once the EPC builds it, it can store MW for as many periods as needed.

Visually, the costs of having early deliveries can be seen in the timeline in Figure 3-9.

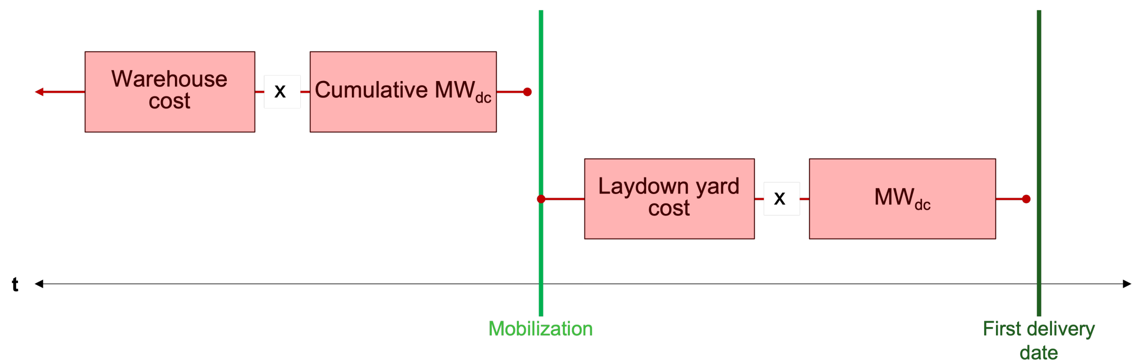

Figure 3-9: Early deliveries cost timeline.

Appendix A.8 details the parameters, variables, and constraints that calculate the costs of early deliveries.

3.9 Remobilization

There are instances when a project may stop receiving solar panels for several periods. If this happens, the EPC may need to dismiss or demobilize part of its workforce. When deliveries start again, the EPC will have to rehire this workforce to remobilize it. Remobilizing the workforce may require the EPC to pay a premium on top of the standard salaries. Typically, if the workforce demobilization lasts for two months or more, the work crew will have sought supplementary jobs elsewhere. The EPC pays the premium to rehire the crew back into the project.

The formal definition of a remobilization in the model is an instance where all of the following three conditions occur:

- A project has a delivery in this period t.
- The project had already had deliveries in previous periods.
- The project did not have a delivery for a certain number of consecutive periods, 2 periods in this thesis, in the immediate last periods.

$has\_remob\_gap_{p,t} \in \{0, 1\}$ is the binary variable that indicates if all three previous conditions happen to project p in time t.

The overall remobilization costs scale with the number of remobilizations a project has. In other words, every time a remobilization in a project happens, the project incurs a cost. The $c\_remob_p \in \mathbb{R}^{\geq 0}$ variable calculates the total cost incurred due to project p remobilizations. The constraint that enforces the logic for c_remob_p is:

$$c\_remob_p = coeff\_remob_p \cdot \sum_i has\_remob\_gap_{p,i} \quad \text{for } i > t\_current, \qquad \forall p \tag{3.10}$$

where $coeff\_remob_p \in \mathbb{R}^{\geq 0}$ is the cost of remobilizing project p after stopping deliveries for δ_remob_p periods. The details of how the variables in this section are calculated are in Appendix A.9.

3.10 Bin type changes

There are some instances where NEER may want to send a different number of solar panels of a particular bin type to a project than the one they contracted for that project. For example, if a project urgently requires deliveries but its supplier has delays, NEER can send panels from a different supplier (and, as such, a different bin type) to the project.

NEER has well-established relationships with its suppliers and has significant flexibility in the delivery location for its contracts. Therefore, NEER can reroute panels to different projects without incurring a cost from the adjustments that the suppliers make to deliver the panels. However, the EPCs have less flexibility in adjusting to bin type changes. The costs that NEER will incur from changing a bin type result from the adjustments that the EPCs have to make. These include change orders and reracking costs, as shown in Figure 3-10.

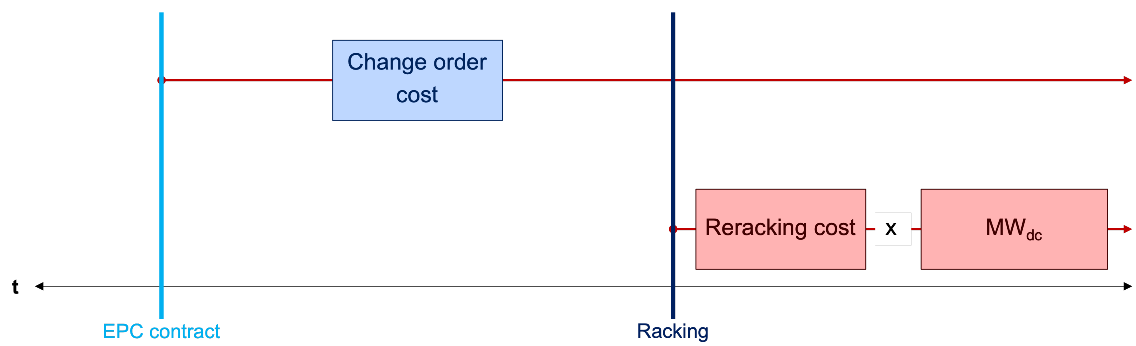

Figure 3-10: Type changes costs timeline.

Change orders When NEER signs the project construction contract with an EPC, they establish the solar panels the project will use in the contract. The EPC makes plans for the project based on the panels in the contract, which include calculations around the project's electrical requirements and drawing the project's map. So, if NEER needs to make a bin type change, the EPC will have to redo some of their work. The process of including the modifications of the bin type changes in a document is called a change order. There is a cost associated with every change order. If NEER makes a bin type change after contracting the EPC, they will incur a change order cost. A change order has a fixed cost, regardless of the MW worth of panels that change. If they change the bin type before contracting the EPC for the project, NEER can avoid the change order cost.

Reracking costs Before the EPC installs a project's solar panels, they need to install the racking equipment for the project. The racking equipment is the foundations that hold the solar panels in place [29]. Each panel bin type uses different rackings as each has a different shape and size. Therefore, if a project changes a bin type after the EPC installs the racking, the EPC will have to adjust the racking equipment. They will likely have to make other adjustments, but we include the cost of the different adjustments in the racking costs to simplify the model. The cost of reracking will vary depending on the level of change that the EPC needs to make to the racking equipment. If NEER decides to make the bin type changes before placing the racking, they do not pay reracking costs.

Timing of the changes As mentioned, a project may incur change orders and reracking costs depending on when the model runs. There is no cost if a bin type change happens before the EPC hiring. If the change occurs after the EPC hiring but the racking has yet to start, then there is only a change order cost. If the change occurs after the racking starts, there is a change order and a reracking cost. This timing concept alludes to cost non-linearity visualized in Figure 2-13.

To gather the change orders and the reracking costs, we introduce the variable $c\_type\_change_p$ as the total cost project p incurs because of changes it had to receive panels with different bin types than initially anticipated. We calculate it with:

$$c\_type\_change_p = coeff\_changeorder_p \cdot needs\_change\_order_p + \sum_b c\_reracking_{p,b} \quad \forall p \tag{3.11}$$

Appendix A.10 presents the elements required to enforce constraint (3.11).

In summary, this chapter presented the objective function of the model along with a description of the costs that comprise it. The next chapter will apply these concepts to different scenarios.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Implementation and results

The main criterion for evaluating the model's performance was the runtime required to reach an optimal solution. As a baseline, we assumed that creating a manual schedule takes ten to twelve working hours. We established that a suitable runtime would be less than two hours long. A supplemental evaluation criterion was the cost that the optimal model calculated. The costs in the model are directional, and the reader should not interpret them as estimations of the actual costs. However, evaluating if the model's proposed schedule costs align with NEER's expectations serves as an audit for the schedule's validity.

We ran three different scenarios to evaluate the model:

1. We derived an optimal schedule based on actual production expectations and the planned NEER solar project portfolio, and we evaluated it against a schedule that the NEER team manually created.

2. We removed the entire production of one supplier to simulate a manufacturing delay disruption.

3. We modified the CODs of some projects to simulate a reprioritization of contracts by accelerating the COD of one project and delaying the COD of three others.

In all three scenarios, the model took less than 15 minutes to find the minimal cost schedules.[1] These results signify success since the model produced a potential schedule with minimal, albeit directional, costs in a fraction of the time it would take to make a manual schedule.

This chapter describes the dataset we used to define the model's parameters and presents the model implementation's results in detail for each of the scenarios.

4.1 Implementation

Description of data We collected the data required to run this exercise from the NEER team. This thesis modifies the names of the projects and suppliers and does not include many parameters we used in the model to preserve confidentiality. Table 4.1.1 shows the size and elements of each set in the model.

[1]The NEER team continued to test the performance of the model beyond the scenarios presented in this thesis. In the scenarios that they tested, most runtimes were around 30 minutes, with a minimum of 5 minutes and a maximum of 60 minutes.

Set	Description	Elements	Size
$\mathscr{P}$	Projects	AM, AS, CW, CV, CP, CO, DE, DS, DB, EX, FF, FT, FV, FC, FGA, FGI, FSB, TU, GA, GRI, $GRII$, GR, GL, GRE, HI, KC, KM, MA, MOI, $MOII$, NM, PE, PB, PC, RU, SEI, $SEII$, SI, SK, SLI, $SLII$, ST, TR, TB, WA, WTI, $WTII$, WC, WG, WM, WW, WII, YA, $YPII$, $YPIII$, $YPIV$	56
$\mathscr{T}$	Periods	1, 2, ..., 36	36
$\mathscr{B}$	Bin types	$A1$, $A2$, $A3$, $B1$, $B2$, $B3$, $C1$, $C2$, $D1$, $D2$, $E1$, $G1$, $G2$	13
$\mathscr{F}$	Form types	$F1$, $F2$, $F3$, $F4$	4
$\mathscr{Z}$	Unallocated project representation	By definition: $Unallocated$	1
$\mathscr{G}$	None bin type representation	By definition: $None$	1

Table 4.1.1: Size and elements of the model sets.

We assumed that no periods were in the past. That is, we assumed that $t\_current = 0$. As such, we did not restrict the model to making decisions in any period. As previously mentioned, we use months for the periods in this thesis. The Gantt chart in Figure 4-1 shows the values for the suppliers' expected production parameters.

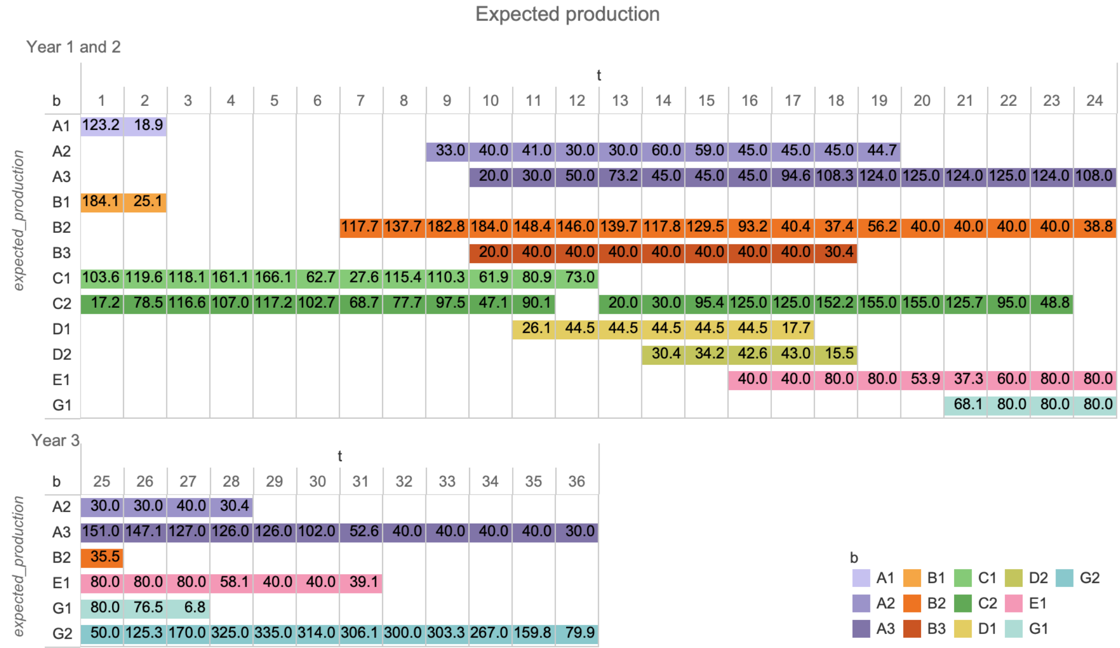

Figure 4-1: Values of the $expected\_production_{t,b}$ parameter.

The Gantt chart in Figure 4-2 presents the parameter values for the volume that each project contracted. All but one of the projects source their panels from one bin type, as it is simpler to manually allocate one bin type per project.

Contracted volume

p	b	t=1	2	3	4	5	6	7	8	9	10	11	12	13	14
WA	A1	123.2	18.9												
WC	B1	87.8	4.2												
YPII	C2	17.2	15.0	26.8											
FGI	C2		30.0	42.0	36.3										
WM	C1	44.7	47.6	29.5	39.2	48.6									
DB	B1	96.3	20.9												
	C1	11.0	21.3	40.9	68.1	65.3	52.7								
DE	C1	47.9	50.7	47.8	53.9	52.2	10.1								
FGA	C2		18.7	25.0	26.2	27.1	11.3								
TU	C2				8.1	44.4	44.7	11.1							
FC	C2		12.1	15.7	16.3	16.9	16.9	13.9	16.3						
AM	C1							27.6	62.3	43.8					
FSB	C2							24.9	36.0	40.6	7.0				
GR	C2		2.7	7.1	20.1	28.8	29.8	18.8	25.3	56.9	40.2	34.1			
PB	C1								53.1	66.5	61.9	80.9	73.0		
YPIII	A3										20.0	30.0	14.4		
ST	B2							62.1	57.9	65.4	64.2	35.5	32.2		
YPIV	A3												15.6	28.2	
HI	B2							55.7	59.4	37.9	44.0	40.5	43.8	45.3	29.1

b: A1, A2, A3, B1, B2, B3, C1, C2, D1, D2, E1, G1, G2

p	b	t=8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
AS	A2		33.0	40.0	41.0	30.0	30.0	30.0	14.0											
DS	B2	0.5	34.5	30.8	27.4	25.1	49.3	48.7	51.7	29.5										
SEI	B2	20.0	45.0	45.0	45.0	45.0	45.0	40.0	40.0	23.8										
YA	D1				26.1	44.5	44.5	44.5	44.5	44.5	17.7									
MOI	E1									40.0	21.3									
SK	B3			20.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0	30.4								
FF	A3					20.0	45.0	45.0	45.0	45.0	45.0	21.3								
CW	C2				56.0		20.0	30.0	30.0	30.0	30.0	7.2								
NM	D2							30.4	34.2	42.6	43.0	15.5								
FV	A2							30.0	45.0	45.0	45.0	45.0	44.7							
CO	B2								37.8	40.0	40.4	37.4	26.2							
MOII	E1										18.7	40.0	40.0	23.9						
WII	E1											40.0	40.0	30.0	18.1					
TR	C2											30.0	30.0	30.0	18.6					
WG	C2											20.0	30.0	30.0	12.1					
GA	C2								35.4	45.0	45.0	45.0	45.0	45.0	45.0	45.0				
EX	C2								30.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	48.8			
KC	B2												30.0	40.0	40.0	40.0	40.0	38.8	35.5	
CP	G1														40.0	40.0	40.0	40.0	40.0	36.5

p	b	t=17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
FT	A3	29.2	35.0	35.0	35.0	35.0	35.0	35.0	35.0	35.0	21.1										
SEII	E1					19.2	40.0	40.0	40.0	40.0	40.0										
GRE	G1					28.1	40.0	40.0	40.0	40.0	40.0	6.8									
WTI	A3	20.4	30.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0	20.0									
WW	A2									30.0	30.0	40.0	30.4								
MA	E1						20.0	40.0	40.0	40.0	40.0	40.0	18.1								
CV	G2												40.0	40.0	19.0						
SLI	E1											40.0	40.0	40.0	6.5						
PC	A3		22.0	49.0	50.0	49.0	50.0	49.0	33.0	46.0	46.0	47.0	46.0	46.0	22.0						
GL	A3									30.0	40.0	40.0	40.0	40.0	40.0	12.6					
KM	G2										25.3	40.0	40.0	40.0	40.0	25.6					
SLII	E1														33.5	39.1					
SI	G2											30.0	40.0	40.0	40.0	40.0	40.0	38.3			
GRI	G2															20.5	40.0	43.8			
PE	G2												35.0	35.0	35.0	35.0	35.0	35.0	32.0		
RU	G2												30.0	40.0	40.0	45.0	45.0	45.0	40.0	9.8	
GRII	G2																	1.2	55.0	60.0	39.9
TB	G2												40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0
WTII	A3											20.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0	40.0	30.0

Figure 4-2: The $contracted_{p,t,b}$ parameter values.

Note that as time progresses, the projects start to phase out some bin types (e.g., $B1$). As solar panel technology improves, projects have fewer incentives to keep using older bin types.

To analyze the volume in the system, we can plot the $expected\_production_{t,b}$ and $contracted_{p,t,b}$ parameters in the same graph and compare them, as is done in Figure 4-3. In Figure 4-3, the colored bars represent the MW of $expected\_production_{t,b}$, and the horizontal black dash represents the MW of $contracted_{p,t,b}$.

For every period in Years 1 and 2, $expected\_production_{t,b} = contracted_{p,t,b}$. In those years, the projects contracted all suppliers' available production. Year 3 has 1000 MW of additional bin type $F2$ production volume. So, in Year 3, the system will have 1000 MW for projects to buy (i.e., the *Unallocated* abstraction will have a value of 1000 MW for the $target_{p,b}$ parameter[2] for more information on the *Unallocated* abstraction and the $target_{p,b}$ parameter). The last year has more volume because the bin type $F2$ supplier has enough time to ramp up its production if needed.

[2]See Appendix A.10.1

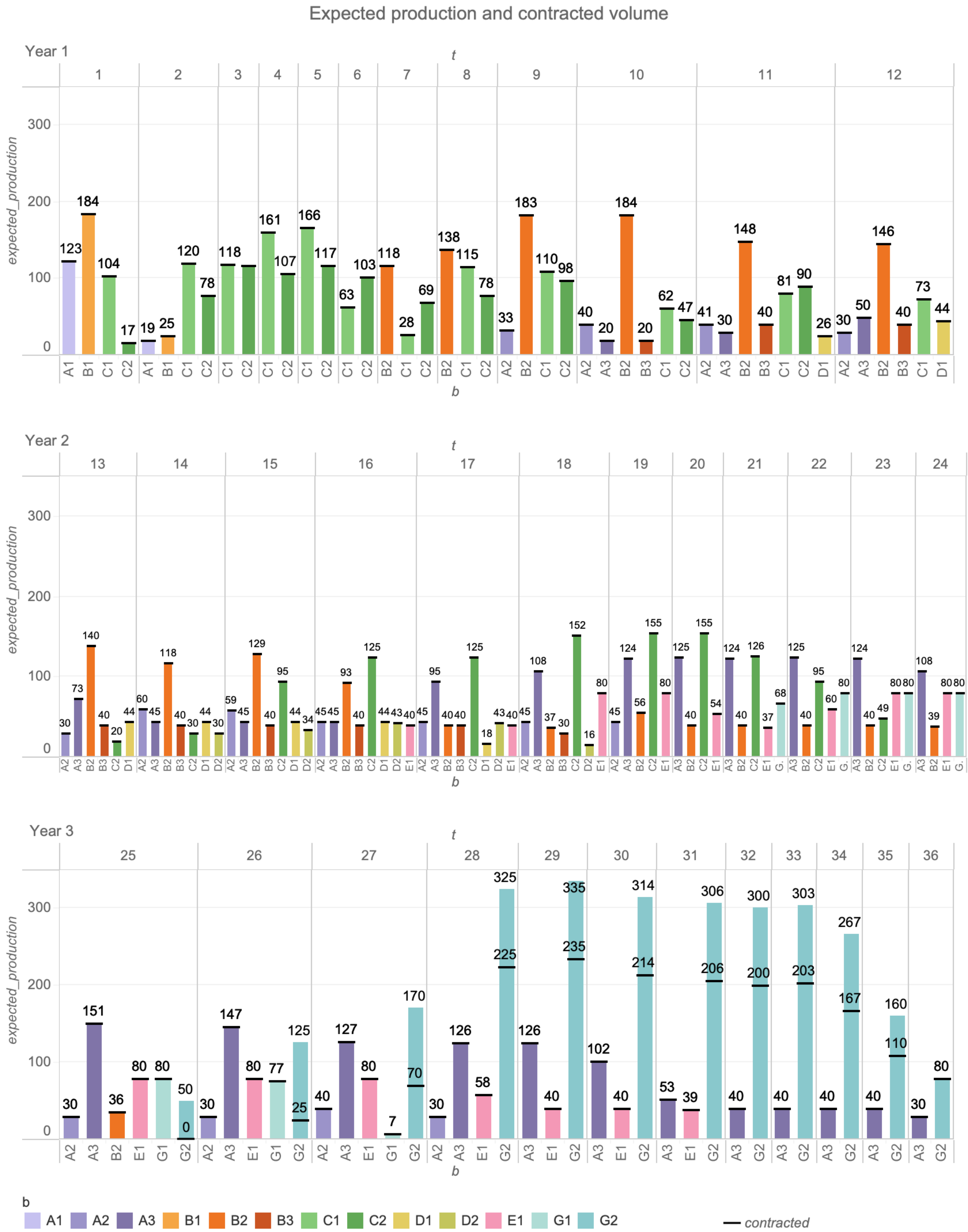

Figure 4-3: Comparison between the $expected\_production_{t,b}$ and $contracted_{p,t,b}$ parameters.

4.2 Results

4.2.1 Scenario: Actual production expectations

In the first scenario, we compared the optimal schedule that the model produced to one that the NEER team had manually crafted based on actual supplier productions. We deliberately abstained from introducing any major disruptions to the supplier production to evaluate the model in a quotidian environment.

The runtime to produce the optimal schedule was 473 seconds. Even without a major supply chain disruption, the algorithm finds a schedule with lower costs than the manual schedule, as Figure 4-10 presents.

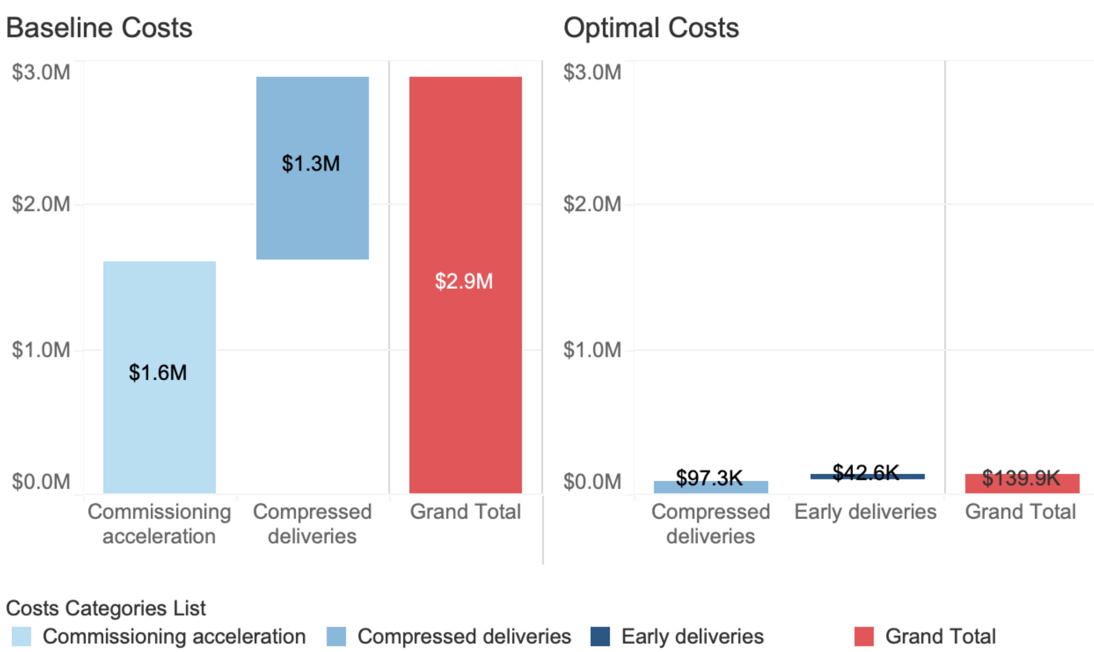

Figure 4-4: Comparison of the costs from the manual and optimal schedules by cost category.

The manual schedule we used as a baseline is equivalent to the $contracted_{p,t,b}$ parameter values. It is the schedule that the team was using to coordinate the deliveries at the moment that we developed the model. While it might be counterintuitive, this schedule has costs because it is not optimal according to the model. The suppliers face changes to their production regularly, and NEER has to adapt its schedule to these. As such, the schedule that determined the $contracted_{p,t,b}$ parameter results from adapting to changes in a sub-optimal way. Figure 4-5 shows the costs of the baseline schedule.

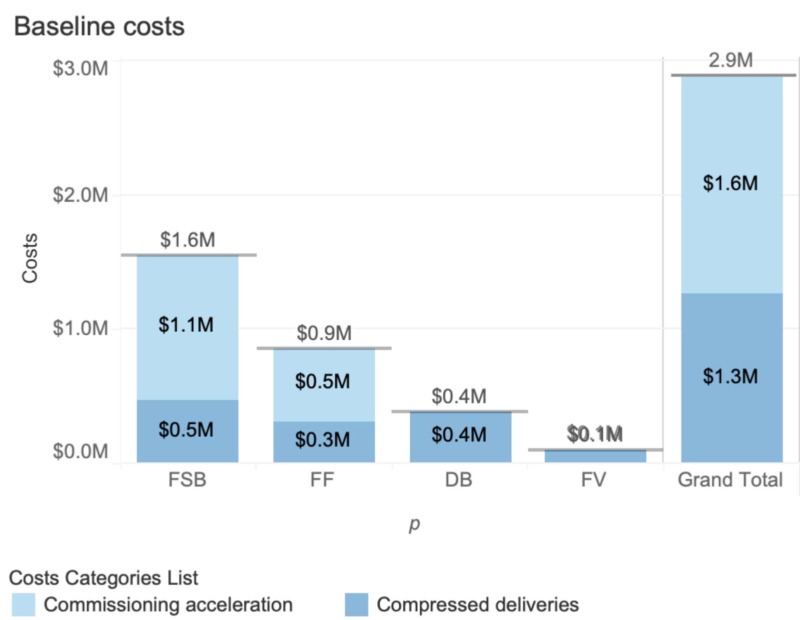

Figure 4-5: Costs of the baseline schedule by project and cost category.

Figures 4-6 and 4-7 present the optimal schedule from the optimization.

Allocated volume

p	b	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
WA	A1	123.2	18.9																
WC	B1	66.9	25.1																
YPII	C2	17.2	38.5	3.3															
DB	B1	117.2																	
	C1	63.6	39.6	78.1	70.0	8.0													
DE	C1	40.0	80.0	40.0	91.1	11.3													
FGI	C2				70.0	38.3													
WM	C1					146.8	62.7												
FGA	C2			90.0		12.5	5.8												
TU	C2					40.0	68.0	0.3											
FC	C2		40.0	23.3	24.5		0.3	0.3	20.0										
FSB	C2						28.4	40.0	40.0										
AM	C1							27.6	26.1	80.0									
GR	C2				12.5	26.5	0.3	28.2	17.7	97.5	47.1	34.1							
PB	C1								89.3	30.3	61.9	80.9	73.0						
YPIII	A3										20.0	30.0	14.4						
HI	B2							117.7	17.7	36.1	40.0	90.0	50.0	4.2					
YPIV	A3													33.5	10.3				
SEI	B2								40.0	106.7	104.0	17.8	30.3	40.0	10.0				
AS	A2									33.0	40.0	41.0	30.0	30.0	35.0	9.0			
DS	B2								80.0	40.0	40.0	26.7	0.3	40.0	40.0	30.4			
FF	B2												65.3	15.8	27.8	13.0	0.3		
	D1												4.7	44.2	44.5	0.3	0.3	0.3	
	E1																30.3	19.8	
CW	C2											56.0		20.0	30.0	52.4	34.8	10.0	
ST	A3												35.3		34.4	36.8		30.5	14.6
	B3												0.6	39.8	5.6			39.5	30.1
	D2																34.7		15.3

p	b	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
YA	B2		13.9	0.3	39.8	40.0			0.3	0.3									
	D1		26.1	39.8	0.3		33.3	13.1	9.3										
	E1							1.1	20.3	28.7									
SK	A3			0.3	39.8	0.3	0.3	45.0	0.3	29.8									
	B3	20.0	40.0	39.4	0.3	34.2	14.1	0.8	0.3	0.3									
	D2					0.3	25.6	0.3	39.5										
NM	A3						7.9		42.1										
	B3					0.3	25.9	39.3	0.3										
	D2					30.1	8.5	7.6	3.5	0.3									
CO	B2						40.0	92.9	40.2	8.7									
FV	A2					25.0	50.0	45.0	45.0	45.0	44.7								
MOII	A3								21.8	30.8	60.0	10.0							
MOI	A3												61.3						
GA	C2							90.0	83.1	112.2	39.6	25.3	0.3						
WII	E1										50.0	28.1							
	G1												40.0	10.0					
TR	C2										40.0	40.0	8.6	20.0					
WG	C2										40.0	50.0	0.3	1.9					
SEII	C2												97.8	72.9	48.6				
EX	B2						46.1				4.3								
	C2						42.9	0.3	31.9	40.0	35.4	39.8	18.9	0.3	0.3				
	D1						11.0	31.1	8.1										
	E1							8.7			0.3	0.3	0.3	0.3	39.5	9.8			
	G1												9.7	39.5	10.3	0.3			
KC	B2										40.0	40.0	30.5	40.0	39.8	38.8	35.3		
CP	E1												18.4	39.8	40.0				
	G1												18.1	0.3		40.0	40.0	40.0	
FT	A3									32.9	25.6	25.3	40.0	95.3	37.9	23.5	40.0	10.0	
GRE	E1															0.3	79.8	79.5	3.8
	G1														33.1	31.5	0.3	0.5	6.3

b: A1 A2 A3 B1 B2 B3 C1 C2 D1 D2 E1 G1 G2

Figure 4-6: $x_{p,t,b}$ variable values (Part 1 of 2).

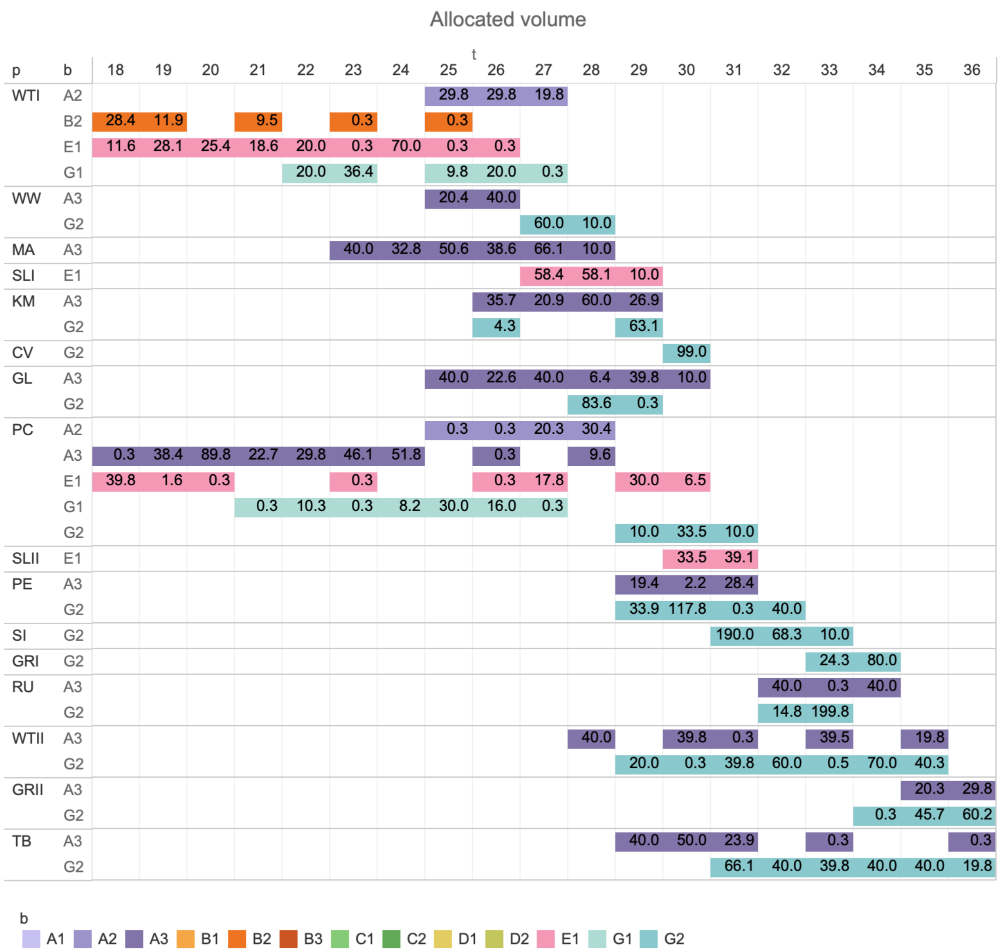

Figure 4-7: $x_{p,t,b}$ variable values (Part 2 of 2).

A difference when comparing the optimal solution to the manual solution is that the projects use up to five different bin types in the optimal schedule, for example, in projects EX and PC. Additionally, the optimal schedule tends to have more projects with delivery gaps of one month, such as projects FGA and FC. No project has more than one month's gap, avoiding a remobilization cost. Figure 4-8 shows the manual and optimal schedule for project FGA based on the portfolio optimization.

In it, we can visualize how the deliveries of project FGA result in a gap in deliveries.

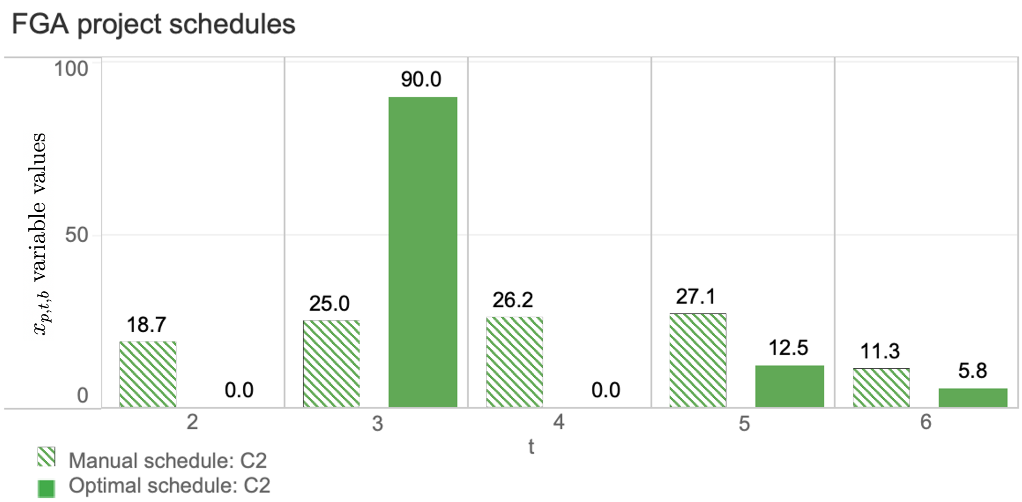

Figure 4-8: Manual and optimal schedules for FGA project.

The model is pushing for earlier and larger deliveries. Take the example of project SEI. In the manual schedule, this project had stable deliveries of 45 MW most months for seven months. In the optimal schedule, project SEI has more than 100 MW deliveries in periods 9 and 10. It is not the only project where the model compresses the deliveries towards earlier months. Since the EPC can compress the deliveries before the inefficiency threshold date without an added cost, the project does not incur a cost despite receiving a high volume of panels. Figure 4-9 compares the manual and optimal delivery schedule for project SEI.

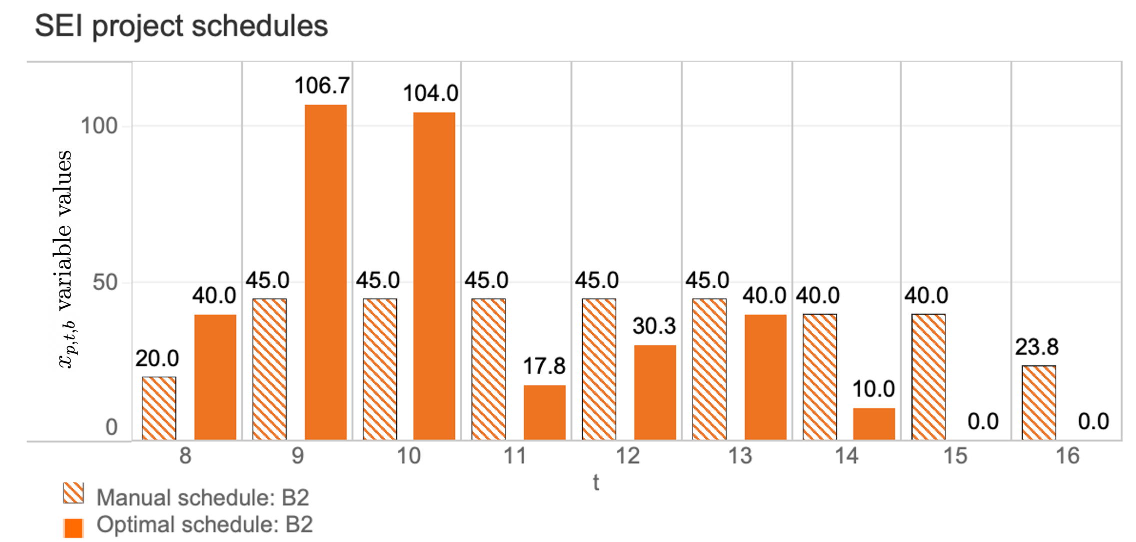

Figure 4-9: Manual and optimal schedules for SEI project.

The observation that the model is pushing for earlier and larger deliveries aligns with the categories of the costs for the optimal schedule: early deliveries and compressed deliveries. Figure 4-10 visualizes the incurred costs based on the model optimization.

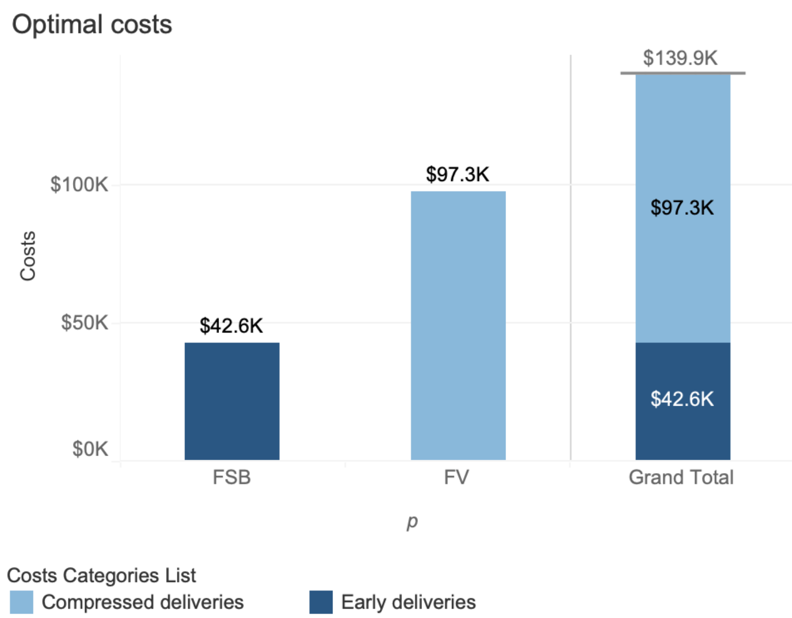

Figure 4-10: Costs of the optimal schedule by project and cost category.

4.2.2 Scenario: Remove one supplier's production

As mentioned in Chapter 1, a DOC investigation of the AD/CVD circumvention halted more than 80% of the solar panel imports in 2022. The disruption affected all solar project developers. To understand the flexibility that the optimization model would give NEER in a similar situation, we analyzed the results of removing all of the volume of the $E1$ supplier from the system. Figure 4-11 compares the contracted and expected production volumes of bin type $E1$ by plotting the contracted parameter and making the $expected\_production_{t,b}$ equal to zero.

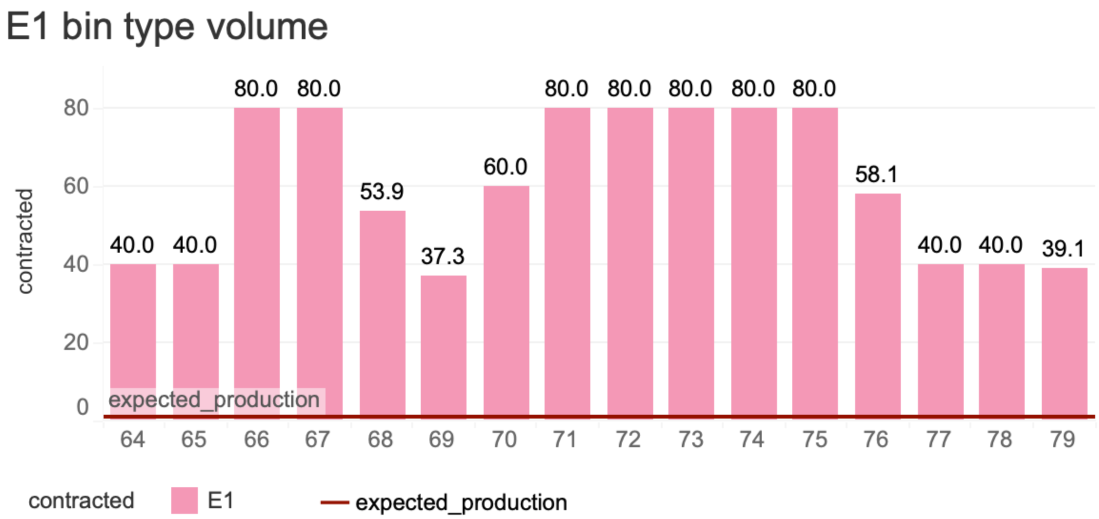

Figure 4-11: Comparison of the contracted and expected production volume of bin type $E1$.

The model found the optimal solution in 752 seconds. It suggested terminating no projects, and the costs for the optimization model were lower than those of terminating seven projects.

In this case, we do not use the costs from a manual schedule as a baseline. We opted to compare the model's cost to the cost of terminating the seven projects that used panels with bin type $E1$ as the baseline. While it would be unreal for NEER to terminate that many projects, we can interpret the cost resulting from the multiple terminations as the worst case that NEER would face in this scenario. As an audit, the optimal schedule costs should always be less than the worst case cost. In this instance, the costs of terminating projects *MOI, MOII, WII, SEII, MA, SLI,* and *SLII* amount to \$72.6M.

Figure 4-12 shows the costs of the optimal schedule. Most of the costs result from change orders and buying new panels, as the projects at risk of being terminated need to change their bin type to avoid termination. The remaining costs are minor, categorized as compressed and early deliveries. This result means that with significant disruption, NEER can adapt its schedule to avoid most of the substantial costs we had prioritized. This is relevant because such speedy results give NEER more flexibility when making strategic decisions. Not only does it help NEER avoid costs, but it also helps them complete projects for their customers. Figures 4-13 and 4-14 display this scenario's schedule in the Gantt chart.

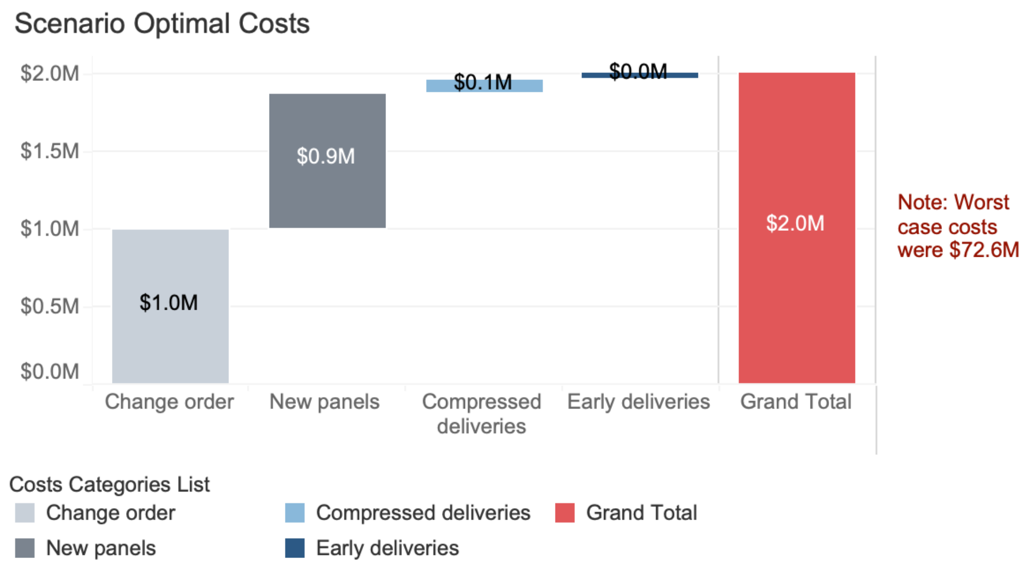

Figure 4-12: Waterfall with costs from scenario removing $E1$ panels.

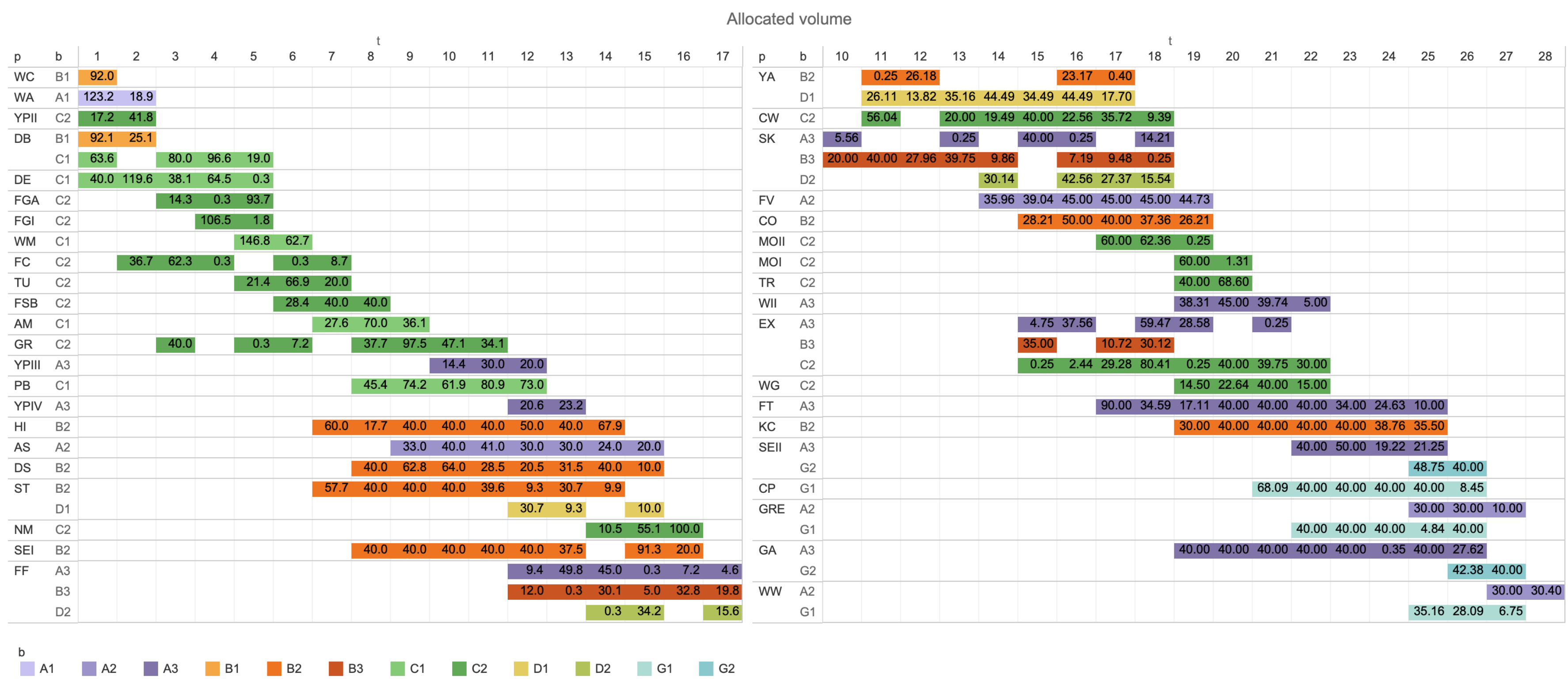

Figure 4-13: Values of the $x_{p,t,b}$ variable for the second disruption scenario (Part 1 of 2).

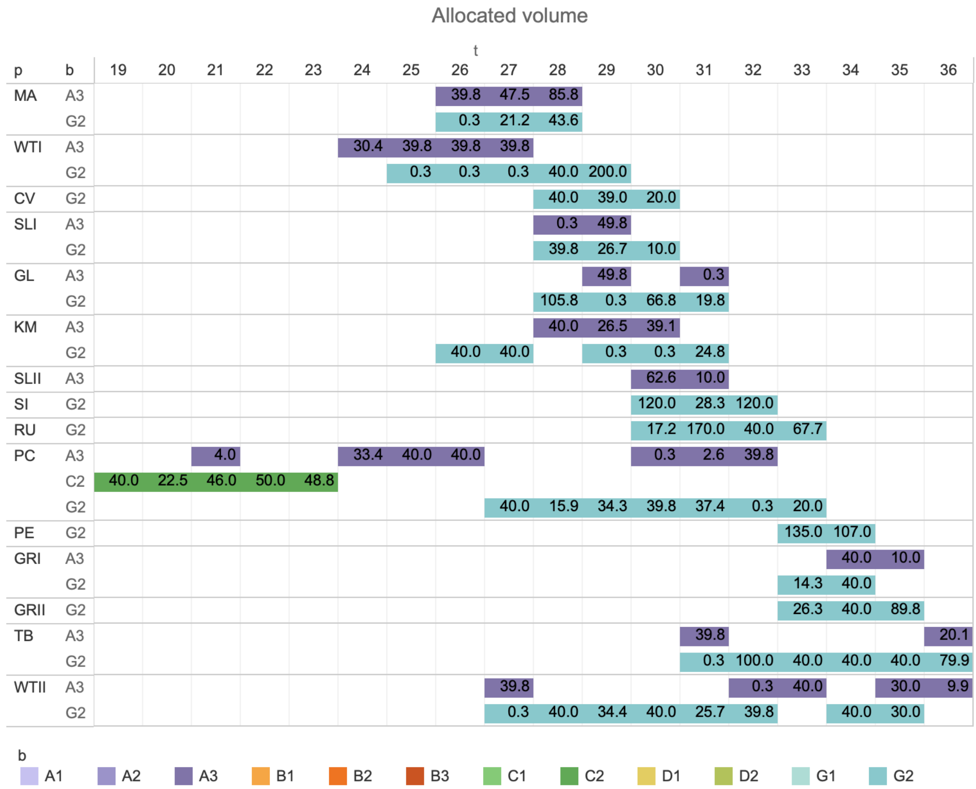

Figure 4-14: Values of the $x_{p,t,b}$ variable for the second disruption scenario (Part 2 of 2).

For most projects, the model looks to reduce their delivery windows. On average, deliveries last 4.1 periods, whereas before, the average delivery window was 5.0. There are only three projects out of the 57 for which the delivery window increased. The model also seeks to start deliveries later while not changing the ending delivery date. The average time deliveries started was 15.5 compared to 14.6 on the original schedule. The scenario's optimal average for the ending date was 19.7, similar to the original schedule's 19.6. For example, project WTI had contracted to start and end deliveries in periods 17 and 27. The model finds that it can help adapt to the disruption by delivering in periods 24 to 29, reducing its delivery window time by half. By starting deliveries later and compressing the delivery time, the model can deliver more volume later, helping the projects access the additional volume of bin type $G2$ in the third year. Additionally, some projects have the flexibility to complete their deliveries later than planned. Project GA, for example, ends its deliveries five periods later than planned without incurring late delivery costs. The summarizing statistics for the starting and ending dates, along with the length of the delivery windows for each project for the original schedule and the optimal disruption schedule, are in Table 4.2.1.

Schedule	Original schedule			Remove the EI volume scenario		
Metric	Started deliveries	Finished deliveries	Delivery window length	Started deliveries	Finished deliveries	Delivery window length
Average	14.6	19.6	5.0	15.5	19.7	4.1
Minimum	0	2	1	0	1	1
Maximum	33	36	12	33	36	14

Table 4.2.1: Summarizing statistics of delivery dates for the original and disrupted schedules.

4.2.3 Scenario: Modify project CODs

Internal and external factors often influence project CODs, including changes in legislation, customer requirements, and company strategies. We created this scenario to evaluate the capacity of the model to adapt to changes in the COD of multiple projects. In particular, Table 4.2.2 displays the changes to the CODs.

	Original periods		Periods after disruption	
Project p	$t\_cont\_COD_p$	$t\_out\_COD_p$	$t\_cont\_COD_p$	$t\_out\_COD_p$
YA	24	27	30	33
KC	30	33	24	27
GRE	33	36	24	27
WW	33	36	24	27

Table 4.2.2: COD values in the original schedule and after a disruption.

The three earlier CODs will constrain the deliveries more. This scenario's worst case cost was terminating the projects with the earlier deadline, KC, GRE, and WW, which amounted to \$47.2M.

The runtime for this scenario was 510 seconds. Again, the model passed the audit of finding a solution with a much smaller cost than the worst case cost, which was \$139.9K. Figure 4-15 shows the costs resulting from the model optimization. The costs from this disruption are mainly from compressing deliveries of project FV.

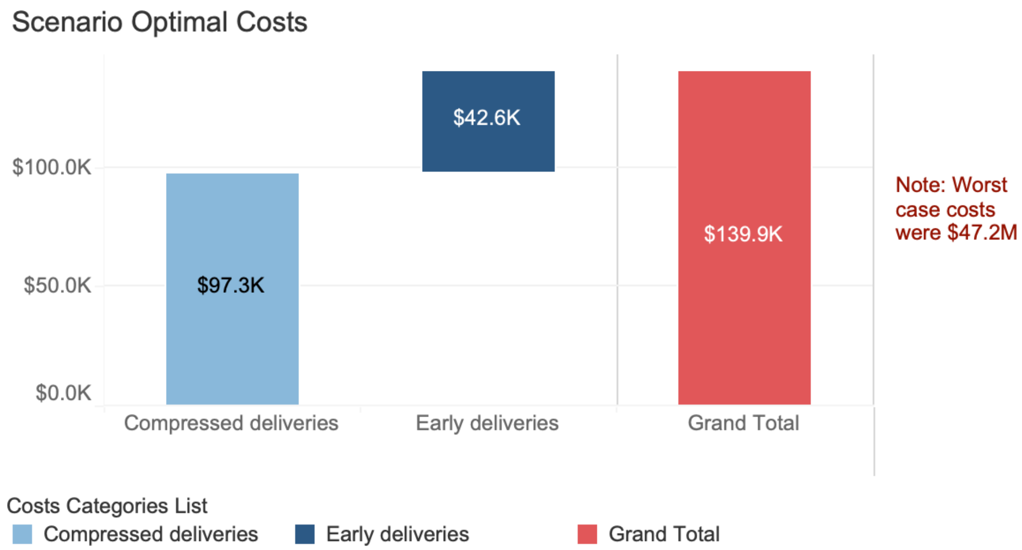

Figure 4-15: Waterfall with costs from changing CODs.

Note that these costs are the same as those incurred in the first scenario, even if the parameters and solutions for the scenarios are different. This means that there are no additional costs resulting from changing the projects' CODs to those from the first scenario. In other words, the model did not incur costs from changing the projects' prioritization. It only incurred costs that were unavoidable because they happened even without a disruption. This shows that the model has a high flexibility when adapting to disruptions.

Additionally, this deadline shift only affected some projects. Most projects had deliveries in the same window as they had initially anticipated. Figures 4-16 and 4-17 show the full allocation schedule.

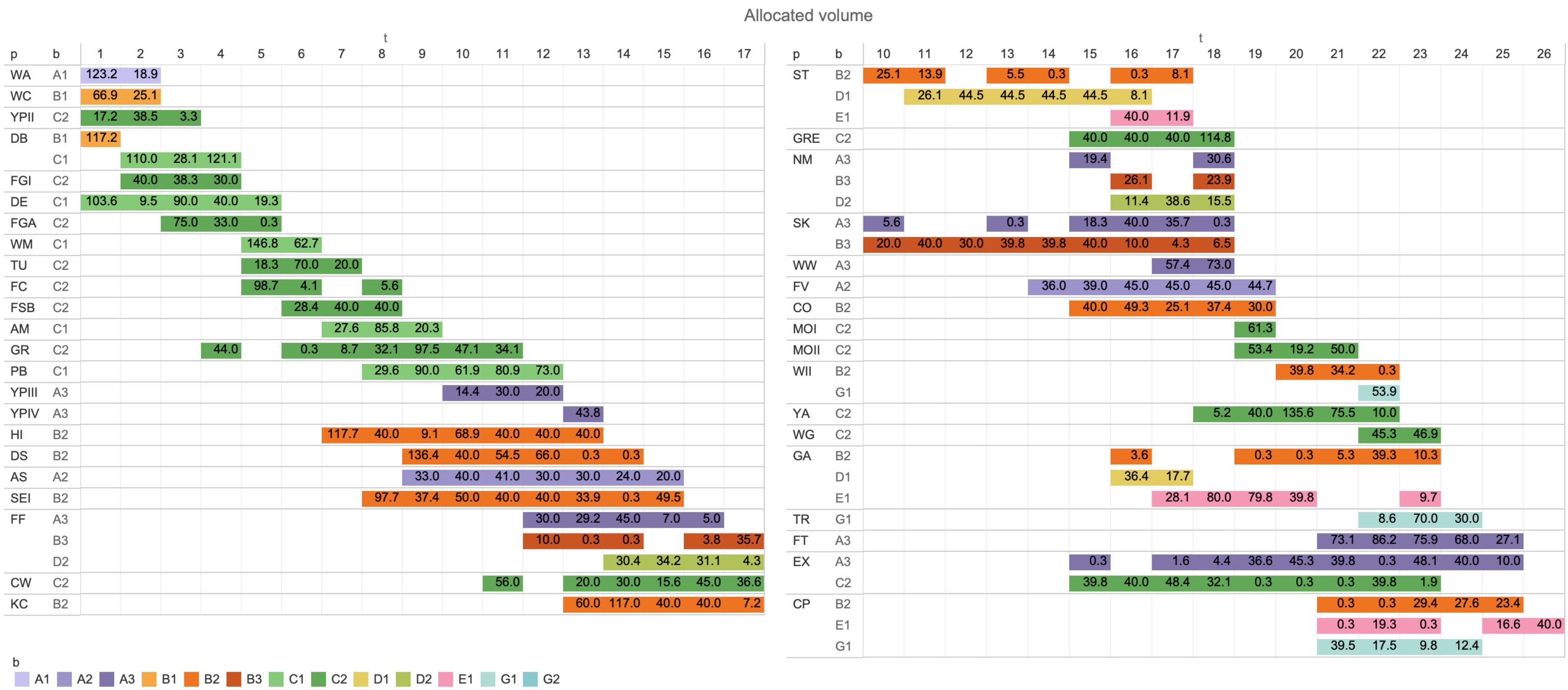

Figure 4-16: Values of the $x_{p,t,b}$ variable for the third disruption scenario (Part 1 of 2).

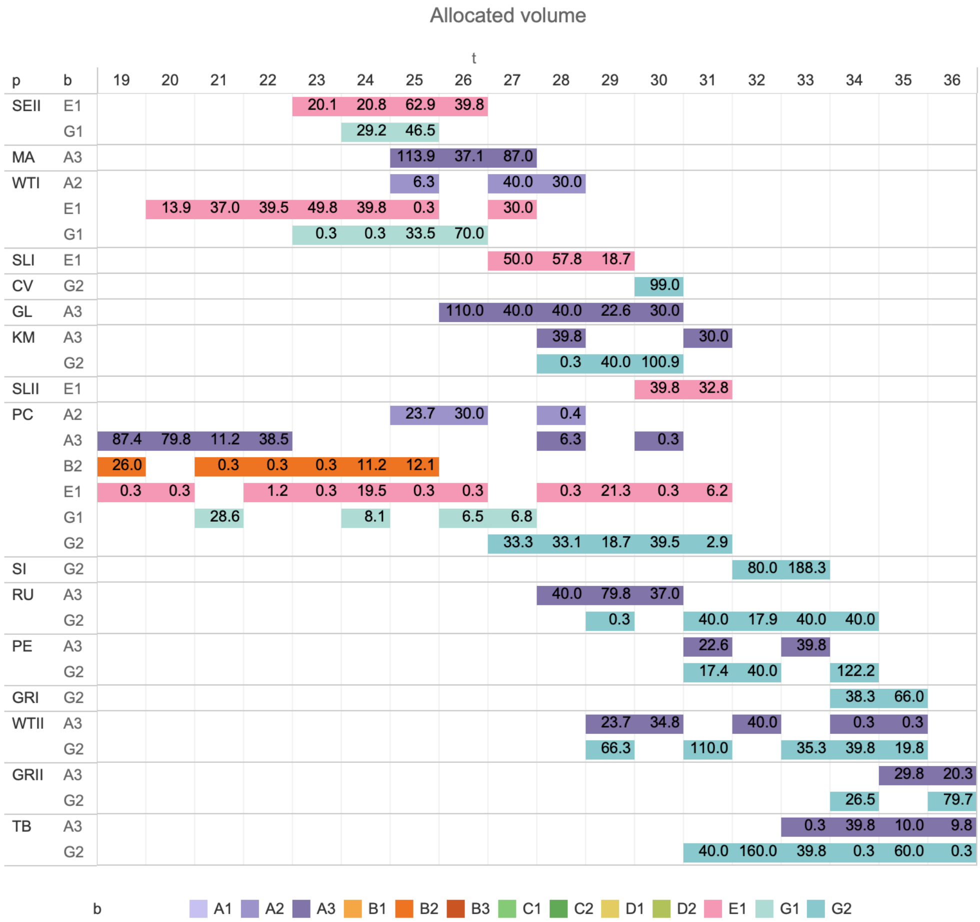

Figure 4-17: Values of the $x_{p,t,b}$ variable for the third disruption scenario (Part 2 of 2).

Let us zoom into the original and disrupted schedule for the projects that had the COD change using Figure 4-12. We observe the shift in deliveries to earlier and later dates. With the new CODs for each project in mind, we also view that all projects are delivered before the corresponding COD, as expected.

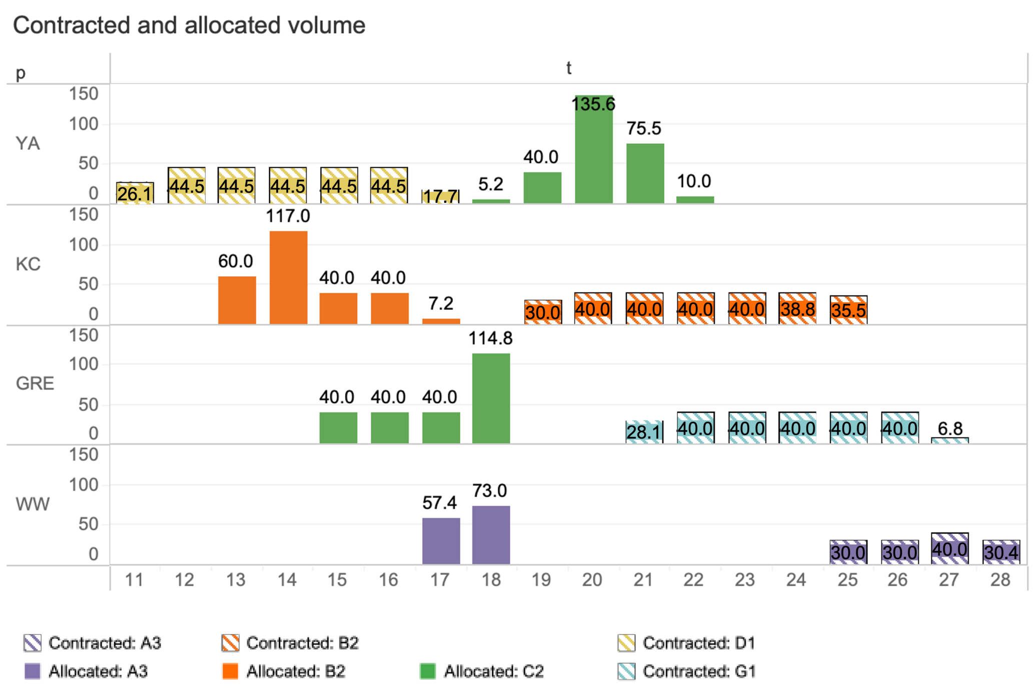

Figure 4-18: Contracted and allocated volume for the projects with COD changes.

Based on these results, the solar panel allocation model is a powerful tool for producing and evaluating delivery schedules for routine operations and if a major disruption happens. The tool can identify potential schedules that adapt to disruptions while minimizing costs in minutes. It also uses hard-to-ideate allocations and does not follow the team's traditional heuristics. These schedules might require manual review, but they help expand the possibilities of the allocations.

Chapter 5

Future work and conclusion

5.1 Future work

As this thesis has mentioned in previous chapters, the costs from the model implementation are directional, not precise, estimations. The model also does not capture the non-linear nature of the costs. Upon future iterations of the model, more details could be introduced to improve the precision of cost estimation. However, this project prioritized using less complex techniques to promote a better understanding and an easier adoption of the model within the organization.

Some additional areas of opportunity to expand on include adding the delivery of other equipment, such as inverters, racking, and storage, into the schedule. As NEER standardizes and increases its data collection, it will have more information regarding the costs, which can also be introduced into the model.

It is worth noting that NEER currently optimizes profitability for each project, not across the portfolio. They seek for every project to be profitable on its own because of strategic implications. This model optimizes the costs across the entire portfolio, so some of the manual allocations might be more aligned with the company strategy, even if they are sub-optimal in terms of the model in this thesis. The model could be altered to produce results that better align with the company's strategy.

5.2 Conclusion

This thesis introduces a mixed integer programming model that creates a viable schedule for solar panel deliveries. The model abstracts impactful and quantifiable costs and minimizes them to propose a realistic solution. By producing a schedule in significantly less time than the current manual approach, the model can adapt to disruptions in the solar panel supply chain faster. This thesis introduces three scenarios that mimic real-world events and disruptions to prove the model's flexibility. In all cases, the model found a feasible solution in less than 15 minutes, which could help NEER prepare for future disruptions.

Appendix A

Model formulation explanations and examples

A.1 Big-$\mathcal{M}$ Method

As mentioned in Chapter 2, the model uses the big-$\mathcal{M}$ method to transform non-linear logical statements of the form if-else into linear constraints.

An example of a logical statement of this type would be the following: for positive integer variable x and binary variable y, if variable $x = 0$, then the variable $y = 0$, else variable $y = 1$. Using the big-$\mathcal{M}$ method, this logical statement can be modeled as:

$$x \leq \mathcal{M}y, \tag{A.1}$$

$$x \geq y, \tag{A.2}$$

$$x \in \mathbb{Z}^{\geq 0}, \tag{A.3}$$

$$y \in \{0, 1\}. \tag{A.4}$$

Substitute different values of x in the constraints to visualize how the big-$\mathcal{M}$ constraints work:

- If $x > 0$, in (A.1), y needs to be equal to one to validate the constraint. $\mathcal{M}$ needs to be multiplied by one so that the right side of the constraint can be greater than or equal to x. If $\mathcal{M}$ is multiplied by zero, then the constraint will be invalid. (A.2) is redundant because $x \geq 1$, which is the maximum value of y.

- If $x = 0$, (A.1) is redundant because $x \leq 0 \leq \mathcal{M}y$. In (A.2), y needs to be equal to zero to validate the constraint. If y equals one, then $x = 0 \geq y = 1$ is a contradiction.

Other logical statements are modeled in this thesis using the big-$\mathcal{M}$ method and similar constraints.

A.2 Delivery basics

A.2.1 Minimum Delivery

The model uses MW and not a number of panels as the unit for x to avoid constraining the solution to an integer solution. NEER has the flexibility to allocate MW in a way that can approximate a non-integer solution.

This approximation can happen because panels within a bin type have variations in their wattage resulting from a manufacturing process. Typically, suppliers classify panels into bin types by 5W or 10W increments [30, 31]. For example, 300W, 310W, and 320W are bin types in the 10W increments classification. Manufacturers round W down to the nearest ten units, so the bin type of 300W would have panels with W in the interval $[300, 310)$.

The caveat of this approximation is that it does not work with quantities close to zero. We cannot split a solar panel into a portion. Allocating 5W (or 5×10^{-6}MW) of a particular bin type is unrealistic because there are no panels with such low MW. More broadly, suppliers send solar panels overseas using containers. Sending a half-filled container would be costly, so it is more realistic that suppliers only send full containers.

Recall that Subsection 3.2.2 introduced the concept of a minimum delivery to capture these issues using the *min_delivery_amount* parameter. The constraints that accompany this parameter and enforce the minimum delivery logic are:

$$x\_from\_contract_{p,t,b} \geq min\_delivery\_amount \cdot min\_delivery\_from\_contract\_binary_{p,t,b} \quad \forall p, t, b \qquad \text{(A.5)}$$

$$x\_new\_buy_{p,t,b} \geq min\_delivery\_amount \cdot min\_delivery\_new\_buy\_binary_{p,t,b} \quad \forall p, t, b \qquad \text{(A.6)}$$

where $min\_delivery\_from\_contract\_binary_{p,t,b} \in \{0, 1\}$ is the binary indicator variable that determines if $x\_from\_contract_{p,t,b} > 0$ and $min\_delivery\_new\_buy\_binary_{p,t,b} \in \{0, 1\}$ is the binary indicator variable that determines if $x\_new\_buy_{p,t,b} > 0$.

If project p sources deliveries from a contract in time t using panels with bin type b, $min\_delivery\_from\_contract\_binary_{p,t,b}$ activates the constraints that make $x\_from\_contract_{p,t,b}$ larger than the minimum delivery. If project p buys deliveries outside a contract in time t using panels with bin type b, $min\_delivery\_new\_buy\_binary_{p,t,b}$ activates the constraints that make $x\_new\_buy_{p,t,b}$ larger than a minimum delivery threshold.

Since $x_{p,t,b}$ is the sum of $x\_from\_contract_{p,t,b}$ and $x\_new\_buy_{p,t,b}$, $x_{p,t,b}$ is also constrained to receive at least $min\_delivery\_amount$ or zero by constraints (A.5) and (A.6).

A.2.2 Completing Projects

In addition to the constraints that differentiate between projects that NEER completes or terminates that we introduce in Subsection 3.2.4, we can introduce a constraint to ensure that no project receives more than the MW it needs to complete. Since the model minimizes costs and delivering more MW would create costs, this constraint is not required. However, reaching optimality takes longer than reaching feasibility when solving a model. As we may be time-constrained, we want to arrive at a feasible solution quickly, even if it is not optimal. In these instances, we would like to have a logical solution, and constraints like the following would help the solution be not only feasible but realistic:

$$\sum_{t,b} x_{p,t,b} \leq MW_p \qquad \forall p \tag{A.7}$$

Since $x_{p,t,b}$ is the sum of $x\_from\_contract_{p,t,b}$ and $x\_new\_buy_{p,t,b}$, constraint (A.7) implies that $\sum_{t,b} x\_from\_contract_{p,t,b}$ and $\sum_{t,b} x\_new\_buy_{p,t,b}$ are also constrained to receive at the most MW_p.

A.3 Minimum MW per bin type and form type

A.3.1 Minimum MW per bin type

To calculate if a project has the minimum MW required per bin type as introduced in Subsection 3.3.1, the model starts by tracking when a project uses a specific bin type to limit the number of bin types a project uses. It introduces $uses\_bintype_{p,b} \in \{0, 1\}$, the binary auxiliary variable that indicates if project p receives any MW using panels with bin type b. It calculates $uses\_bintype_{p,b}$ with:

$$\sum_t x_{p,t,b} \leq \mathcal{M}\_uses\_bin\_type \cdot uses\_bintype_{p,b} \qquad \forall p, b \qquad (A.8)$$

$$\sum_t x_{p,t,b} \geq \epsilon\_uses\_bin\_type \cdot uses\_bintype_{p,b} \qquad \forall p, b \qquad (A.9)$$

where $\mathcal{M}\_uses\_bin\_type \in \mathbb{R}^+$ and $\epsilon\_uses\_bin\_type \in (0, 1)$.

$\mathcal{M}\_uses\_bin\_type$ is a "big-$\mathcal{M}$" parameter (see Appendix A.1) that ensures the model assigns the $uses\_bintype_{p,b}$ variables to the expected binary values based on their related constraints. $\epsilon\_uses\_bin\_type$ is a "small-ϵ" parameter that transforms strict inequalities into non-strict inequalities for the constraints regarding the $uses\_bintype_{p,b}$ variables. $\epsilon\_uses\_bin\_type$ is an infinitesimal quantity that makes the left-hand side of the (A.9) constraint be something strictly greater than zero without forcing it to take a value more significant than presumed.

To understand how the (A.8) and (A.9) constraints behave, we review the examples:

- If $\sum_t x_{p,t,b} = 10$ (or any value > 0), then the only way that (A.8) can be valid is if $uses\_bintype_{p,b}$ equals 1. Recall from 2.1.3.4 that $\mathcal{M}\_uses\_bin\_type$ is a number large enough that it does not constrain how large $\sum_t x_{p,t,b}$ can be. Since the left-hand side of (A.9) is larger than the infinitesimal value of $\epsilon\_uses\_bin\_type$ and zero, (A.9) is always valid regardless of the value of $uses\_bintype_{p,b}$. As such, $uses\_bintype_{p,b} = 1$.

- If $\sum_t x_{p,t,b} = 0$, then the left-hand side of (A.8) is always valid regardless of the value of $uses\_bintype_{p,b}$. However, the only way that (A.9) can be valid is if $uses\_bintype_{p,b}$ equals 0. If $uses\_bintype_{p,b}$ were to equal 1, then we would have $0 \geq \epsilon\_uses\_bin\_type$, which is a contradiction since we defined $\epsilon\_uses\_bin\_type$ as strictly greater than 0. Therefore, $uses\_bintype_{p,b} = 0$.

As such, if a project p receives panels with bin type b in any period, then $uses\_bintype_{p,b} = 1$. Otherwise, $uses\_bintype_{p,b} = 0$. With the $uses\_bintype_{p,b}$ variables, the model can not only track which bin types each project uses but also track and limit how many bin types each uses.

Some exceptions to consider in this set of constraints are the following. First, if a project requires less MW than or equal to the $minper\_bin$ parameter, then even if it receives all of its volume with one bin type, the amount of MW for that bin type will be less than the minimum. To account for this, the model establishes different constraints for projects with less MW than or equal to the minimum and those with more than the minimum. To differentiate the projects, the model uses the $mw\_geq\_min\_per\_bin_p \in \{0, 1\}$ binary auxiliary parameter, which indicates if the MW project p requires are greater than or equal to the $minper\_bin$ threshold amount.

If a project has less MW than or equal to the $minper\_bin$, then the model limits the project to using panels with only one bin type. However, it accounts for another exception. There might be an instance where a project had contracted panels with more than one bin type, even when it had overall less MW than or equal to the

$minper\_bin$. The model takes the maximum between one bin type and the number of bin types a project had contracted before. It introduces $contracted\_bin\_type_{p,b} \in \{0, 1\}$ as the binary parameter determining if project p had contracted to have any MW from bin class p in the original schedule. The mathematical constraint looks like this:

If $mw\_geq\_min\_per\_bin_p = 0$,

$$\sum_b uses\_bintype_{p,b} \leq \max\left\{1, \sum_b contracted\_bin\_type_{p,b}\right\} \qquad \forall p \quad \text{(A.10)}$$

On the other hand, if a project has more MW than $minper\_bin$, it can have panels with multiple bin types. However, each bin type's minimum quantity is $minper\_bin$. There is again an exception where a project might have previously contracted less MW per bin type. In parallel to the previous constraint, the model takes the minimum between the $minper\_bin$ and the amount of MW for all the bin types a project had contracted before. The constraint that defines this logic is:

If $mw\_geq\_min\_per\_bin_p = 1$,

$$\sum_t x_{p,t,b} \geq \min\left\{\{target\_bin_{p,b} | target\_bin_{p,b} > 0\} \cup \{minper\_bin\}\right\} \cdot uses\_bintype_{p,b} \qquad \forall p, b \quad \text{(A.11)}$$

where the $target\_bin_{p,b} \in \mathbb{R}^{\geq 0}$ parameter represents the contracted (or targeted) amount of MW that project p has with bin-type b, as per the original schedule. Notice that by definition $\sum_t contracted_{p,t,b} = target\_bin_{p,b} \quad \forall p, b$. We use $target\_bin_{p,b}$ when possible for simplicity.

The following examples evaluate the constraints A.10 and A.11 for different projects that have more or less MW than $minper\_bin$. Suppose a project completes with 30MW and $minper\_bin = 40$. If it had only contracted one bin type before, then substituting in A.10:

$$\begin{aligned}
&mw_geq_min_per_bin_p = 0,\\
&\sum_b contracted_bin_type_{p,b} = 1,\\
&\max\left\{1, \sum_b contracted_bin_type_{p,b}\right\} = \max\{1, 1\} = 1,\\
&\Rightarrow \sum_b uses_bintype_{p,b} \leq 1.
\end{aligned}$$

So, this project could only have one bin type. Now, suppose the same project had previously contracted panels with three different bin types. Then:

$$\begin{aligned}
&mw_geq_min_per_bin_p = 0,\\
&\sum_b contracted_bin_type_{p,b} = 3,\\
&\max\left\{1, \sum_b contracted_bin_type_{p,b}\right\} = \max\{1, 3\} = 3,\\
&\Rightarrow \sum_b uses_bintype_{p,b} \leq 3.
\end{aligned}$$

Therefore, this project could continue to have three bin types.

Let us now suppose that there is a project that is complete with 100MW, and it has contracted all of its MW using only one bin type. Assume $minper\_bin = 40$.

Substituting on A.11:

$$\begin{aligned}
&mw_geq_min_per_bin_p = 1, \\
&\min\{\{target_bin_{p,b} | target_bin_{p,b} > 0\} \cup \{minper_bin\}\} = \{\{100\} \cup \{40\}\} \\
&\qquad = \{100, 40\} = 40 \\
&\Rightarrow \sum_t x_{p,t,b} \geq 40 \cdot uses_bintype_{p,b} \quad \forall b
\end{aligned}$$

This means that if the project uses a bin type, it must receive at least 40MW from that bin type.

Now, suppose the project had previously contracted 80MW of bin type 1 and 20MW of bin type 2. Again, if we substitute on A.11:

$$\begin{aligned}
&mw_geq_min_per_bin_p = 1, \\
&\min\{\{target_bin_{p,b} | target_bin_{p,b} > 0\} \cup \{minper_bin\}\} = \{\{80, 20\} \cup \{40\}\} \\
&\qquad = \{80, 20, 40\} = 20 \\
&\Rightarrow \sum_t x_{p,t,b} \geq 20 \cdot uses_bintype_{p,b} \quad \forall b
\end{aligned}$$

In this case, the project can have bin types that only use 20MW because it already has a contract below the minimum quantity of $minper\_bin$.

A.3.2 Minimum MW per form type

The elements related to the minimum MW per form type subsection are analogous to the previous subsection. First, the model introduces the $bin\_form\_conversion_{b,f} \in \{0, 1\}$, a family of binary parameters to indicate whether panels classified under bin type b also fall under form type f. If a supplier classifies the panel under bin type b

and form type f, then $bin\_form\_conversion_{b,f}$ will be 1. It will be 0 otherwise.

Similar to the constraints on bin types, the model also introduces the $uses\_formtype_{p,f} \in \{0,1\}$ binary auxiliary variable, which indicates if project p receives some MW using panels with form type f. The model calculates the $uses\_formtype_{p,f}$ variable with:

$$\sum_{t,b} (bin\_form\_conversion_{b,f} \cdot x_{p,t,b}) \leq \mathcal{M}\_uses\_bin\_type \cdot uses\_formtype_{p,f} \qquad \forall p, f \tag{A.12}$$

$$\sum_{t,b} (bin\_form\_conversion_{b,f} \cdot x_{p,t,b}) \geq \epsilon\_uses\_bin\_type \cdot uses\_formtype_{p,f} \qquad \forall p, f \tag{A.13}$$

Notice that $\mathcal{M}\_uses\_bin\_type$ and $\epsilon\_uses\_bin\_type$ are the same parameters introduced in the previous subsection. Since the maximum MW of a specific bin type that a project can receive is the same as the maximum MW of a particular form, then there is no need to introduce new "big-$\mathcal{M}$" and "small-ϵ" parameters.

The model also introduces the parameter $minper\_form \in \mathbb{R}^+$, which is the threshold parameter that establishes how many MW per form type is the minimum that each project can have. In this case, if a project receives any MW from a form type, it has to receive at least $minper\_form$. Typically, $minper\_form$ is larger than $minper\_bin$.

The model distinguishes between the projects with less MW than or equal to the minimum threshold per form type and those with at more than the minimum with the $mw\_geq\_min\_per\_form_p \in \{0,1\}$ binary auxiliary parameter. It indicates if the MW project p requires are greater than or equal to the $minper\_form$ threshold amount.

Next, the model introduces $contracted\_form\_type_{p,f} \in \{0,1\}$ as the binary parameter determining if project p had contracted to have any MW from the form class f in the original schedule. If a project has more MW than $minper\_form$, the model restricts the project to using panels with only one form type unless the project has previously contracted panels with more than one form type. It uses the following constraint to do so:

$$\text{If } mw\_geq\_min\_per\_form_p = 0, \quad \sum_f uses\_formtype_{p,f} \leq \max\left\{1, \sum_f contracted\_form\_type_{p,f}\right\} \qquad \forall p \tag{A.14}$$

A project with more MW than $minper\_form$ can have panels with multiple bin types. However, the minimum quantity that each form type can have is $minper\_form$ unless the project has previously contracted less MW per form type. In a constraint, it looks like:

$$\text{If } mw\_geq\_min\_per\_form_p = 1, \quad \sum_{t,b} (bin\_form\_conversion_{b,f} \cdot x_{p,t,b}) \geq \min\{\{target\_form_{p,f} | target\_form_{p,f} > 0\} \cup \{minper\_form\}\} \cdot uses\_formtype_{p,f} \qquad \forall p, f \tag{A.15}$$

A.4 Supplier capacities

A.4.1 Expedited contract supply

Recall that the variable $c\_expedite\_supply_{t,b}$ is the cost per MW of expediting the deliveries as introduced in Subsection 3.4.2. The model calculates it by multiplying the amount of MW projects needed to expedite times the cost coefficient parameter $coeff\_expedite\_supply_{t,b}$. The $coeff\_expedite\_supply_{t,b} \in \mathbb{R}^{\geq 0}$ parameter is defined as the cost per MW of having a supplier expedite its production of panels with bin type b so they can deliver in time t, which is before the date established in the contract. The model introduces it using the constraint:

$$c\_expedite\_supply_{t,b} = coeff\_expedite\_supply_{t,b} \cdot extra_{t,b} \qquad \forall t, b \quad \text{(A.16)}$$

where $extra_{t,b} \in \mathbb{R}^{\geq 0}$ is the variable that represents the amount of MW additional to those contracted and not yet delivered, allocated to projects in time t using panels with bin type b. A positive amount of $extra_{t,b}$ means NEER expedites some of the panels associated with the contract for bin type b.

The model constrains $extra_{t,b}$ with:

$$extra_{t,b} \geq \left(\sum_{p} \sum_{i=1}^{t} x\_from\_contract_{p,i,b} \right) - \left(\sum_{p} \sum_{i=1}^{t} contracted_{p,i,b} \right) - \left(\sum_{i=1}^{t-1} extra_{i,b} \right)$$
$$\text{for } p \in \mathscr{P}, \quad \forall t, b \quad \text{(A.17)}$$

We examine the $extra_{t,b}$ variable more closely in the remainder of this subsection. It has three components:

- $\sum_{p} \sum_{i=1}^{t} x\_from\_contract_{p,i,b}$ is the volume of bin type b that the model has decided to deliver to all projects up until period t.

- $\sum_{p} \sum_{i=1}^{t} contracted_{p,i,b}$ is the overall volume that NEER has contracted of panels with bin type b until period t for all projects. It is the volume the supplier that produces bin type b is required to send to NEER by period t.

- $\sum_{i=1}^{t-1} extra_{i,b}$ is the cumulative sum of the $extra$ variables from previous periods to t.

The $extra_{t,b}$ variable assumes the difference between the volume the model allocates to projects and the volume the supplier should have delivered to NEER. It assumes the panels that a supplier may owe NEER panels and that NEER may wish to expedite deliveries. Each constraint's $extra_{t,b}$ variables are calculated using cumulative periods. Therefore, the $extra_{t,b}$ variables corresponding to previous periods are subtracted to avoid calculating duplicate costs. Intuitively, the model subtracts NEER's payments for expedited volume in prior periods to prevent double counting costs.

In other words, given a contract that serves a set of projects using panels with bin type b, the projects can receive, without a cost, the MW they contracted for this period plus the MW they had contracted in previous periods but did not receive. The $extra_{t,b}$ variable will be positive if the projects allocate more than their entitled volume. Since $extra_{t,b} \in \mathbb{R}^{\geq 0}$, it cannot be negative. Nevertheless, if the calculation of $\left(\sum_p \sum_{i=1}^{t} x\_from\_contract_{p,i,b}\right) - \left(\sum_p \sum_{i=1}^{t} contracted_{p,i,b}\right) - \left(\sum_{i=1}^{t-1} extra_{i,b}\right)$ were negative, it would mean that the projects are allocated less than their entitled volume. There is no cost for allocating less than the entitled volume, so the model does not track negative value.

To better visualize how the model calculates $extra_{t,b}$, suppose that a project is complete with 160MW, and the supplier schedule is to send 40MW in periods 1 to 4. In the first period, the model decides it is best to deliver 60MW. Since the supplier does not owe volume to NEER, the supplier has to expedite 20MW, which will have a cost. In the second period, the model decides to deliver 40MW. In the third period, a disruption prevents the supplier from delivering volume. The supplier will provide the remaining 60MW for the project in period 4 when the disruption subsides. Figure

A-1 illustrates the example, and Table A.4.1 presents its values.

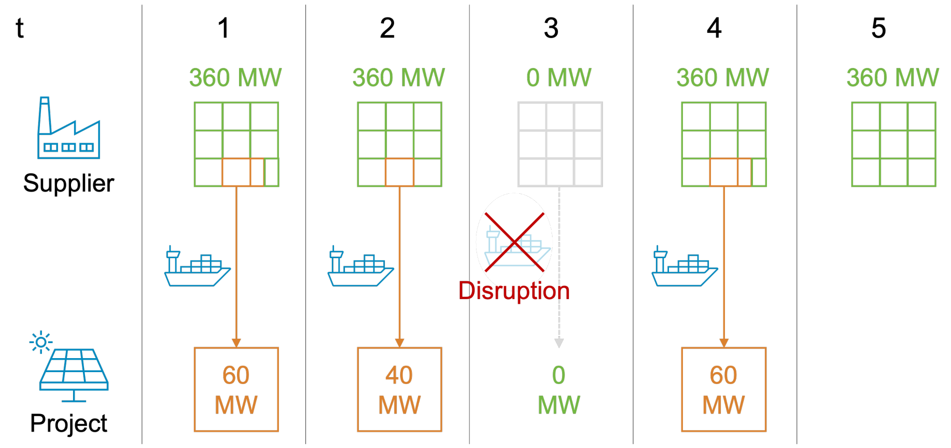

Figure A-1: Example of a simple schedule with limited and expedited deliveries.

t	1	2	3	4	5
$expected\_production_{t,b}$	360	360	0	360	360
$contracted_{p,t,b}$	40	40	40	40	0
$x\_from\_contract_{p,t,b}$	60	40	0	60	0

Table A.4.1: Values of an example of a simple schedule with limited and expedited deliveries.

In period 1, we would expect $extra_{t,b}$ to take the value of 20 because the model allocated 60MW, but NEER had only contracted 40MW until that period. Replacing the values from Table A.4.1 in the constraint (A.17):

$$extra_{1,b} \geq \left(\sum_{p}\sum_{i=1}^{1} x\_from\_contract_{p,i,b}\right) - \left(\sum_{p}\sum_{i=1}^{1} contracted_{p,i,b}\right) - \left(\sum_{i=1}^{0} extra_{i,b}\right)$$
$$\text{for } p \in \mathscr{P}, \quad \forall t, b$$

$$\Rightarrow extra_{1,b} \geq (x\_from\_contract_{p,1,b}) - (contracted_{p,1,b})$$

$$\Rightarrow extra_{1,b} \geq (60) - (40)$$

$$\Rightarrow extra_{1,b} \geq 20$$

Since the model seeks to minimize costs and higher values of $extra_{t,b}$ result in higher costs, $extra_{1,b}$ will equal 20.

In period 2, we would expect $extra_{t,b}$ to take the value of 0 because the model allocated 40MW, and NEER contracted 40MW. Again, replacing in the constraint (A.17):

$$extra_{2,b} \geq \left(\sum_{p}\sum_{i=1}^{2} x\_from\_contract_{p,i,b}\right) - \left(\sum_{p}\sum_{i=1}^{2} contracted_{p,i,b}\right) - \left(\sum_{i=1}^{1} extra_{i,b}\right)$$
$$\text{for } p \in \mathscr{P}, \quad \forall t, b$$

$$\Rightarrow extra_{2,b} \geq (x\_from\_contract_{p,1,b} + x\_from\_contract_{p,2,b})$$
$$- (contracted_{p,1,b} + contracted_{p,2,b}) - (extra_{1,b})$$

$$\Rightarrow extra_{2,b} \geq (60 + 40) - (40 + 40) - (20)$$

$$\Rightarrow extra_{2,b} \geq 0$$

Since the model seeks to minimize costs, $extra_{2,b} = 0$. Because the model calculates the $extra_{t,b}$ variables using cumulative periods, it needs to remove previous $extra_{t,b}$ variables from the current $extra_{t,b}$ variable to avoid calculating duplicate costs. We would have had $extra_{2,b} \geq 20$ if we had not subtracted the sum of the previous $extra_{t,b}$ variables or $\left(\sum_{i=1}^{1} extra_{i,b}\right)$. We have indeed allocated more MW than we are entitled to cumulatively. However, we paid for those 20MW in period 1. So, not removing them in period 2 would result in a duplicate payment.

We can repeat this process for the following periods and arrive at the results of $extra_{t,b} \geq 0$ for $t \geq 3$. Therefore, the $extra_{t,b}$ variable correctly calculates the amount of MW that NEER is expediting in each period.

A.5 Changes to the weeks that the EPC crew works

The most relevant variable in this section is $c\_additional\_crew_p$, which calculates the cost of paying for additional crews to account for the additional work weeks required to receive the MW allocated to project p, including the weeks with inefficiencies and when commissioning needs to be accelerated. This section expands on the concepts related to additional labor that we briefly defined in Section 3.5, including work weeks, inefficiencies, and commissioning.

A.5.1 Work weeks

Recall from Section 3.5 that a work week is the unit that measures the number of crews needed to receive and install a certain number of panels in one week. The costs related to work weeks arise if one of the following situations happens:

- There is a positive change in the number of work weeks required to handle the deliveries. For example, this cost would result from changing the schedule in Table A.5.1 to that in Table A.5.2.
- More work weeks are required after the inefficiency threshold has begun.

- Work weeks are required after the EPC plans to start commissioning a project.

This section builds the components required to calculate these costs and presents examples to facilitate learning.

Starting work weeks The model uses the $work\_weeks_{p,t} \in \mathbb{Z}^{\geq 0}$ parameter to integrate the weeks' worth of work that project p has contracted in time t. Additionally, the model uses the parameter $MW\_per\_work\_week \in \mathbb{R}^{+}$ as the number of MW that one crew can receive and install at the most in one week. Since the model can use different periods (days, weeks, months, and others), it introduces the $weeks\_per\_period \in \mathbb{Z}^{+}$ parameter to standardize the workload into an equivalent unit. $weeks\_per\_period$ assumes how many weeks are equivalent to one period. In the case of this thesis, the periods are months, so we take $weeks\_per\_period$ to be 4. As such, all periods (or months) are equal in length.

We can review an example schedule that gathers the $work\_weeks_{p,t}$, $MW\_per\_work\_week$, and $weeks\_per\_period$ parameters. Examine a project p that receives 40MW in periods 1 to 5. The project can receive 10MW per work week ($MW\_per\_work\_week = 10$) and $weeks\_per\_period = 4$. Table A.5.1 calculates the work weeks required in each period and overall:

t	1	2	3	4	5	Total
$\sum_b contracted_{p,t,b}$	40	40	40	40	40	200
$work\_weeks_{p,t}$	4	4	4	4	4	20

Table A.5.1: Example of work week based on simple project schedule.

In this example, all the work crews are very efficient since they receive the most MW they can each week. However, that is only sometimes the case. Let us review a variation where the project gets 35MW in period 1 and 45MW in period 2. Everything else remains the same. Table A.5.2 calculates the work weeks required in each period and overall:

t	1	2	3	4	5	Total
$\sum_b contracted_{p,t,b}$	35	45	40	40	40	200
$work\_weeks_{p,t}$	4	5	4	4	4	21

Table A.5.2: Example of work weeks based on a non-efficient project schedule.

In this case, the project still receives 200MW overall. However, in period 1, the crews received 35MW, 5MW less than before, but they still needed four work weeks to receive them. They need three work weeks to receive 30MW and one work week to receive the remaining 5MW. The EPC must hire a crew for the remaining 5MW in period 1. The same happens in period 2. The preprocessing script calculates the work weeks for which the EPC hires the crew before running the algorithm.

Additional work weeks The model differentiates between the actual deliveries for the new schedule and the contracted deliveries using the variable $required\_work\_weeks_{p,t} \in \mathbb{Z}^{\geq 0}$. This variable estimates the number of weeks' worth of work required to receive the MW allocated to project p in time t. The number of work weeks project p requires in period t depends on the MW delivered to project p in time t and the MW one crew can receive in one week. When compiled into a constraint, this becomes:

$$\sum_b x_{p,t,b} \leq MW\_per\_work\_week \cdot required\_work\_weeks_{p,t}, \qquad \forall p, t \quad \text{(A.18)}$$

There is an additional consideration that impacts the number of required work weeks. Sometimes, projects need to keep a crew on for an entire period. Keeping a crew means that a project must have a crew for every week in the period. For example, if a crew is required in period 1, which is assumed to have four weeks, then period 1 must have four work weeks.

If a project starts receiving deliveries, it must keep a crew on for the entire period until it finishes deliveries. In other words, project p needs to keep a crew on in period t if it meets the following three conditions:

- Project p is receiving a delivery in period t, or it has received a delivery before period t.
- It still needs to finish deliveries at period t. That is, the total MW delivered are less than the total MW the project requires for completion.
- The model decided not to terminate project p.

If project p finished deliveries in period t, it could leave a partial crew on during that last period. The variable $requires\_keeping\_crew_{p,t} \in \{0, 1\}$ is the binary indicator determining if project p needs to keep a crew in time t.

For example, Table A.5.3 shows a scenario where project p started deliveries in period 1, paused deliveries in period 2, and finished deliveries in period 5. Assume that $MW\_per\_work\_week = 10$ and $weeks\_per\_period = 4$.

t	1	2	3	4	5	Total
$\sum_b x_{p,t,b}$	15	0	35	40	20	110
$required\_work\_weeks_{p,t}$	4	4	4	4	2	16

Table A.5.3: Example of the required work weeks based on a variable project schedule.

Periods 1 and 2 need to keep a crew, so the $required\_work\_weeks_{p,t}$ variable equals 4, even if they are not using the crews for the entire period. In other words, if a project needs to keep a crew for a certain period, it must account for enough work weeks. Since period 5 is the last period, the project has $required\_work\_weeks_{p,t} = 2$ because they do not need to keep on a crew for the final period.

To introduce the constraints that translate this logic into the optimization model, we need to present some auxiliary variables:

- $made\_delivery_{p,t} \in \{0, 1\}$ is the binary variable that indicates if project p has deliveries in time t.
- $has\_delivered_{p,t} \in \{0, 1\}$ is the binary variable that indicates if project p has or will have had deliveries before time t.
- $finished\_delivery_{p,t} \in \{0, 1\}$ is the binary variable that indicates if project p finished deliveries before or in time t.

For $made\_delivery_{p,t}$, we want the variable to take the value of 1 if project p makes deliveries in period t of any bin type and the value of 0 otherwise. Using logical statements, this would be:

$$\begin{aligned} \text{If } \sum_b x_{p,t,b} > 0, \quad & made\_delivery_{p,t} = 1 \\ \text{else,} \quad & made\_delivery_{p,t} = 0 \qquad\qquad \forall p, t \end{aligned}$$

However, these logical statements are non-linear since $\sum_b x_{p,t,b} > 0$ is the sum of variables. Therefore, we use the big-$\mathcal{M}$ method to turn the logical statement into linear equations:

$$\sum_{b} x_{p,t,b} - \epsilon\_made\_delivery \geq -\mathcal{M}\_made\_delivery \cdot (1 - made\_delivery_{p,t}) \qquad \forall p, t \tag{A.19}$$

$$\sum_{b} x_{p,t,b} \leq \mathcal{M}\_made\_delivery \cdot made\_delivery_{p,t} \qquad \forall p, t \tag{A.20}$$

where $\mathcal{M}\_made\_delivery \in \mathbb{R}^+$ is a "big-$\mathcal{M}$". It ensures that the model assigns the $made\_delivery_{p,t}$ variables to the expected binary values based on the previous constraints. Also, $\epsilon\_made\_delivery \in (0, 1)$ is a "small-ϵ" parameter that transforms strict inequalities into non-strict inequalities for the constraints regarding the $made\_delivery_{p,t}$ variables. We do not dive into how the "big-$\mathcal{M}$" constraints work to avoid repetition (see 2.1.3.4, A.8, and A.9).

Next, a project is defined as having made a delivery if it has had positive deliveries in previous periods. $has\_delivered_{p,t}$ will take the value of 1 if project p made deliveries before period t of any bin type and the value of 0 otherwise. In a logical statement:

$$\text{If } \sum_{i=1}^{t-1} made\_delivery_{p,i} \geq 1, \quad has\_delivered_{p,t} = 1$$
$$\text{else,} \quad has\_delivered_{p,t} = 0 \qquad \forall p, t$$

Again, the big-$\mathcal{M}$ method turns this logical statement into linear equations, where $\mathcal{M}\_has\_delivered \in \mathbb{R}^+$ and $\epsilon\_has\_delivered \in (0, 1)$ are "big-$\mathcal{M}$" and "small-ϵ"

parameters for the constraint related to the $has\_delivered_{p,t}$ variable, respectively:

$$\sum_{i=1}^{t-1} made\_delivery_{p,i} \geq 1 - \mathcal{M}\_has\_delivered \cdot (1 - has\_delivered_{p,t}) \qquad \forall p, t \tag{A.21}$$

$$\sum_{i=1}^{t-1} made\_delivery_{p,i} + \epsilon\_has\_delivered \leq 1 + \mathcal{M}\_has\_delivered \cdot has\_delivered_{p,t} \qquad \forall p, t \tag{A.22}$$

Thirdly, to have finished deliveries for a project, the cumulative MW delivered each period must be more than the required MW. That is,

$$\text{If } \sum_{b} \sum_{i=1}^{t} x_{p,i,b} \geq MW_p, \quad finished\_delivery_{p,t} = 1$$
$$\text{else,} \quad finished\_delivery_{p,t} = 0 \qquad \forall p, t$$

Using the big-$\mathcal{M}$ method:

$$\sum_{b} \sum_{i=1}^{t} x_{p,i,b} \geq MW_p - \mathcal{M}\_finished\_delivery \cdot (1 - finished\_delivery_{p,t}) \qquad \forall p, t \tag{A.23}$$

$$\sum_{b} \sum_{i=1}^{t} x_{p,i,b} + \epsilon\_finished\_delivery \leq MW_p + \mathcal{M}\_finished\_delivery \cdot finished\_delivery_{p,t} \qquad \forall p, t \tag{A.24}$$

where $\mathcal{M}\_finished\_delivery \in \mathbb{R}^{+}$ and $\epsilon\_finished\_delivery \in (0,1)$ are the "big-$\mathcal{M}$" and "small-ϵ" parameters for the $finished\_delivery_{p,t}$ variable, respectively.

Consolidating the many components into constraints to calculate $requires\_keeping\_crew_{p,t}$ amounts to:

$$\begin{aligned}2 + &requires\_keeping\_crew_{p,t} \\ &\geq has\_delivered_{p,t} + (1 - finished\_delivery_{p,t}) + (1 - terminate_{p}) \qquad \forall p, t\end{aligned} \tag{A.25}$$

$$\begin{aligned}2 + &requires\_keeping\_crew_{p,t} \\ &\geq made\_delivery_{p,t} + (1 - finished\_delivery_{p,t}) + (1 - terminate_{p}) \qquad \forall p, t\end{aligned} \tag{A.26}$$

$$required\_work\_weeks_{p,t} \geq weeks\_per\_period \cdot requires\_keeping\_crew_{p,t} \qquad \forall p, t \tag{A.27}$$

Now, we can calculate the difference between the required work weeks and the ones we had initially contracted. As we have seen in Chapter 3, if a project requires more work weeks than contracted, there is a cost for the additional work weeks. In the model, $additional\_work\_weeks_{p} \in \mathbb{Z}^{\geq 0}$ is the additional weeks' worth of work required to receive the MW allocated to project p for all periods. They are in addition to the ones already contracted for project p. In the form of a constraint, this is:

$$\sum_{t} required\_work\_weeks_{p,t} - \sum_{t} work\_weeks_{p,t} \leq additional\_work\_weeks_{p} \qquad \forall p \tag{A.28}$$

For example, suppose project p contracted 40MW of deliveries in periods 1 to 5. Assume that $MW\_per\_work\_week = 10$, $weeks\_per\_period = 4$, and $terminate_p = 0$. Periods will 1 to 5 have $work\_weeks_{p,t} = 4$, and any other period will have $work\_weeks_{p,t} = 4$. There is a disruption, so now the project receives 20MW in period 1, no MW in period 2, 50MW in periods 3, 4, and 5, and 30MW in period 6. Table A.5.4 calculates the $required\_work\_weeks_{p,t}$ and $additional\_work\_weeks_p$ variables.

t	1	2	3	4	5	6	Total
$\sum_b contracted_{p,t,b}$	40	40	40	40	40	0	200
$work\_weeks_{p,t}$	4	4	4	4	4	0	20
$\sum_b x_{p,t,b}$	20	0	50	50	50	30	200
$made\_delivery_{p,t}$	1	0	1	1	1	1	
$has\_delivered_{p,t}$	0	1	1	1	1	1	
$finished\_delivery_{p,t}$	0	0	0	0	0	1	
$requires\_keeping\_crew_{p,t}$	1	1	1	1	1	0	
$required\_work\_weeks_{p,t}$	4	4	5	5	5	3	26
$additional\_work\_weeks_p$							6

Table A.5.4: Example of the required work weeks based on a disrupted project schedule.

Periods 1 to 5 are required to keep a crew because they started delivering but have not finished. Therefore, they must have at least four work crews, even if they make no deliveries (like period 2). Periods 3 to 5 require more than four work weeks because they need five work weeks to receive 50MW. Period 6 does not keep a crew on because it finishes delivering in that period, requiring only three work weeks. So overall, the project requires 26 work weeks. It amounts to 6 additional work weeks than the project's initially contracted.

A.5.2 Inefficiencies in labor

While there is some flexibility to how the EPC can allocate the working crew, it has less flexibility to do so the further along the project gets. The closer deliveries get to the commissioning start date, the less flexibility a project has to shuffle work crews. Additionally, having more deliveries in later periods may lead to the crews having to work overtime. The model captures this issue by introducing a concept of inefficiency. If a project requires more work weeks than contracted in periods after the inefficiency threshold date and before the commissioning begins, it will need to pay for additional crews with inefficiencies.

In particular, the model parametrizes the inefficiency threshold as $t\_inefficiency_p \in \mathbb{Z}^{\geq 0}$, representing the period after which the EPC will work with inefficiencies. It parametrizes the date when EPC plans to start commissioning the project p as $t\_comm_p \in \mathbb{Z}^{+}$. A period has inefficiencies if $t\_inefficiency \leq t < t\_comm_p$ and requires more work weeks than were initially contracted. In the model, the variables $inefficiency\_weeks_{p,t} \in \mathbb{Z}^{\geq 0}$ calculate the number of weeks' worth of work with inefficiencies for project p in month t using the ensuing constraint:

$$inefficiency\_weeks_{p,t} \geq required\_work\_weeks_{p,t} - work\_weeks_{p,t}$$
$$\text{for } t > t\_current \text{ and } t\_inefficiency_p \leq t < t\_comm_p, \quad \forall p \quad \text{(A.29)}$$

Notice that if the inefficiency period has passed ($t > t\_current$), the model does not charge any cost; instead, it assumes that NEER has already paid the past costs.

Figure A-2 presents an example that can help better visualize the concept of inefficiency. A project planned to receive 40MW from April to August. Assume that $MW\_per\_work\_week = 10$, $weeks\_per\_period = 4$, and $terminate_p = 0$. Then, after a disruption, the project has to receive deliveries after the inefficiency threshold date. The window of deliveries does not change; only the delivered MW changes.

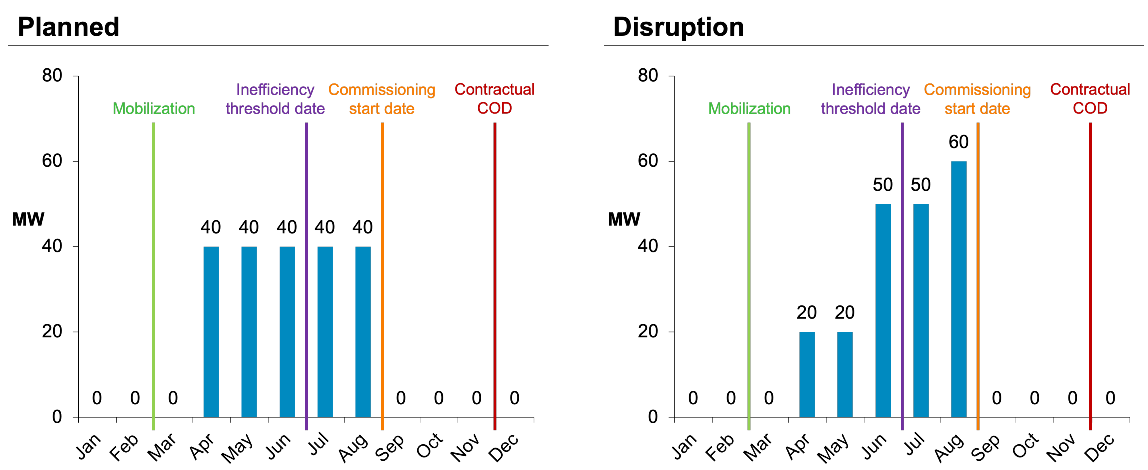

Figure A-2: Example of inefficiencies in a project schedule after a disruption.

Table A.5.5 calculates the $required\_work\_weeks_{p,t}$, $additional\_work\_weeks_p$, and $inefficiency\_weeks_{p,t}$ variables. Overall, the project requires four additional work weeks. However, after the inefficiency threshold date, the end of June, it also requires three more work weeks than anticipated. So, there will be one cost for the total additional work weeks and another for the additional work week in inefficiency periods.

t	Apr	May	Jun	Jul	Aug	Total
$\sum_b contracted_{p,t,b}$	40	40	40	40	40	200
$work\_weeks_{p,t}$	4	4	4	4	4	20
$\sum_b x_{p,t,b}$	20	20	50	50	60	200
$made\_delivery_{p,t}$	1	1	1	1	1	
$has\_delivered_{p,t}$	0	1	1	1	1	
$finished\_delivery_{p,t}$	0	0	0	0	1	
$requires\_keeping\_crew_{p,t}$	1	1	1	1	0	
$required\_work\_weeks_{p,t}$	4	4	5	5	6	24
$additional\_work\_weeks_p$						4
$inefficiency\_weeks_{p,t}$				1	2	3

Table A.5.5: Example of the required work weeks based on a project with inefficiencies.

A.5.3 Commissioning acceleration

If a project's deliveries continue after its commissioning is supposed to start, it will affect the commissioning processes. First, the EPC will commission the different parts of the project on a rolling basis determined by the sections where they have installed solar panels. It will likely cause the crew to work overtime and to bring additional crews in.

To account for this, the model introduces a cost for the work crews it needs to hire after the commissioning start date. Figure A-3 presents the example where a project plans to receive 40MW from April to August. This example can help the reader visualize how the model calculates the crews it needs to accelerate commissioning. We assume that $MW\_per\_work\_week = 10$, $weeks\_per\_period =$

4, and $terminate_p = 0$. In this case, after a disruption, the project has to receive deliveries after the commissioning start date. It does not have weeks with inefficiencies.

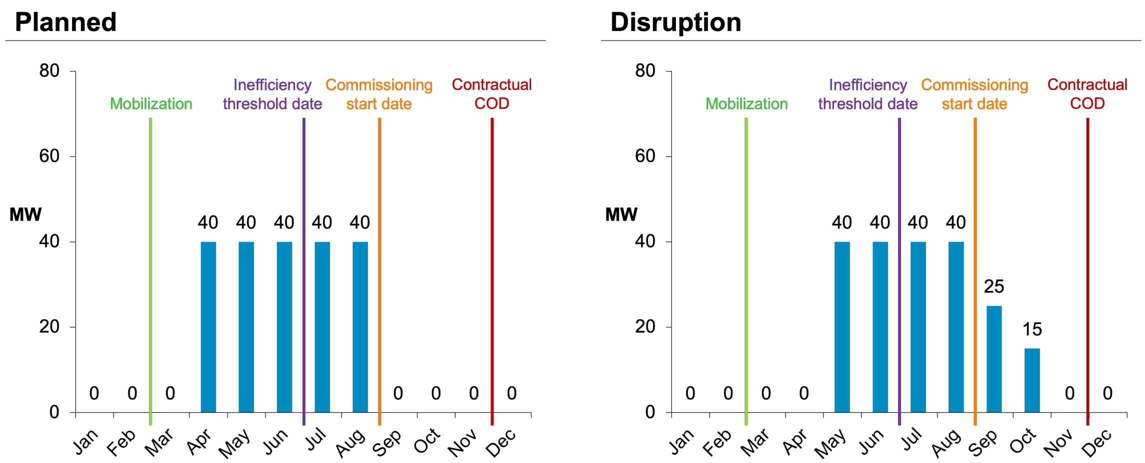

Figure A-3: Example of accelerating commissioning in a project schedule after a disruption.

Table A.5.6 calculates the variables related to the work weeks. Overall, the project requires one additional work week. After the commissioning start date, five work weeks are needed for the end of August (highlighted in red). There is an overall cost of 1 additional work week and another cost for five work weeks when commissioning has to be accelerated. In this case, the delivery window changes.

t	Apr	May	Jun	Jul	Aug	Sep	Oct	Total
$\sum_b contracted_{p,t,b}$	40	40	40	40	40	0	0	200
$work\_weeks_{p,t}$	4	4	4	4	4	0	0	20
$\sum_b x_{p,t,b}$	0	40	40	40	40	25	15	200
$made\_delivery_{p,t}$	0	1	1	1	1	1	1	
$has\_delivered_{p,t}$	0	0	1	1	1	1	1	
$finished\_delivery_{p,t}$	0	0	0	0	0	0	1	
$requires\_keeping\_crew_{p,t}$	0	1	1	1	1	1	0	
$required\_work\_weeks_{p,t}$	0	4	4	4	4	**3**	**2**	21
$additional\_work\_weeks_p$								1
$inefficiency\_weeks_{p,t}$				0	0	0	0	0

Table A.5.6: Example of the required work weeks based on a project that needs to accelerate its commissioning.

To solidify these concepts, we present the example in Figure A-4. The project plans to receive 40MW from April to August, but a disruption causes both inefficiencies and the commissioning to accelerate. We assume that $MW\_per\_work\_week = 10$, $weeks\_per\_period = 4$, and $terminate_p = 0$.

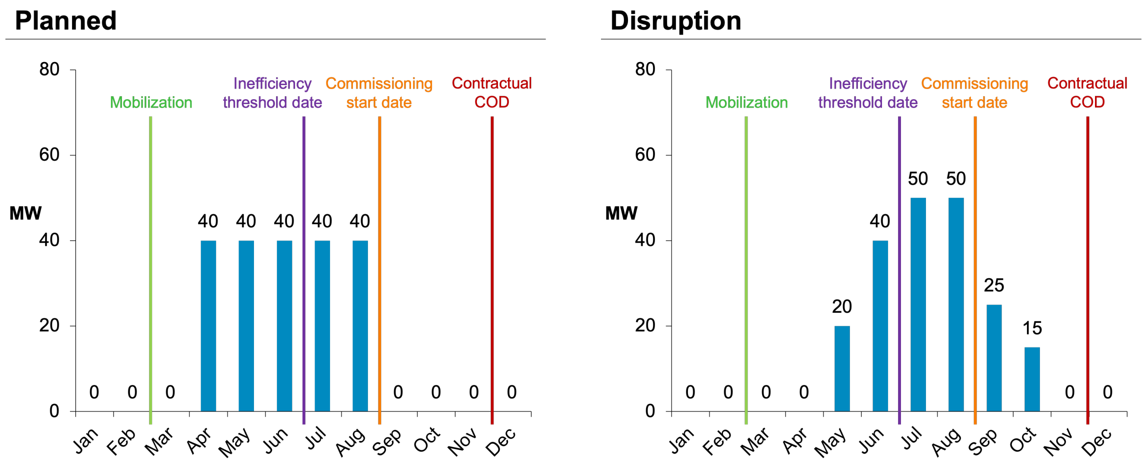

Figure A-4: Example of inefficiencies and accelerating commissioning in a project schedule after a disruption.

Table A.5.7 calculates the variables related to the work weeks. The project requires three additional work weeks, two inefficiency weeks, and 5 work weeks where commissioning is accelerated.

t	Apr	May	Jun	Jul	Aug	Sep	Oct	Total
$\sum_b contracted_{p,t,b}$	40	40	40	40	40	0	0	200
$work\_weeks_{p,t}$	4	4	4	4	4	0	0	20
$\sum_b x_{p,t,b}$	0	20	40	50	50	25	15	200
$made\_delivery_{p,t}$	0	1	1	1	1	1	1	
$has\_delivered_{p,t}$	0	0	1	1	1	1	1	
$finished\_delivery_{p,t}$	0	0	0	0	0	0	1	
$requires\_keeping\_crew_{p,t}$	0	1	1	1	1	1	0	
$required\_work\_weeks_{p,t}$	0	4	4	5	5	**3**	**2**	23
$additional\_work\_weeks_p$								3
$inefficiency\_weeks_{p,t}$				1	1	0	0	2

Table A.5.7: Example of the required work weeks based on a project with inefficiencies that needs to accelerate its commissioning.

If we replace these values in the (3.9) constraint, we get:

$$\begin{aligned} c\_additional\_crew_p = & 3 \cdot coeff\_additional\_work\_week \\ & + 2 \cdot coeff\_inefficiencies \\ & + 5 \cdot coeff\_commissioning\_acceleration \end{aligned}$$

Generally, we define $coeff\_additional\_work\_week < coeff\_inefficiencies < coeff\_commissioning\_acceleration$. So, for this example, the most significant component in $c\_additional\_crew_p$ will be $5 \cdot coeff\_commissioning\_acceleration$.

A.6 Compressed deliveries

Recall that projects have a maximum delivery beyond which their deliveries are compressed. The $compressed\_thresh \in \mathbb{R}^{+}$ parameter is the threshold of maximum MW that a project can receive in one period without incurring a cost for a high volume of deliveries. The model keeps track of deliveries over this threshold because compressed deliveries result in costs. It tracks it using the $amount\_compressed_{p,t} \in \mathbb{R}^{\geq 0}$ variable, which is the amount of MW above the $compressed\_thresh$ delivered to project p in time t. In a mathematical constraint, it looks like this:

$$amount\_compressed_{p,t} \geq \sum_{b} x_{p,t,b} - compressed\_thresh \qquad \forall p, t \qquad \text{(A.30)}$$

A non-zero value of $amount\_compressed_{p,t}$ means that project p receives more volume than $compressed\_thresh$ in time t.

As mentioned, projects have less flexibility to adapt to disruptions the closer deliveries are to the COD. The logic of how the model calculates the cost of compressed deliveries is very similar to how it calculates the costs of changes in the weeks that the EPC works. In particular, if a project makes more deliveries than $compressed\_thresh$ after the inefficiency threshold period, it must compress some of its deliveries.[1] In that case, the project will incur a cost for the amount of MW above the compressing threshold. If a project makes deliveries after or during the commissioning start date, it must compress all of its deliveries and incur a cost for all MW delivered. The model calculates the cost of project p receiving more MW in time t than the $compressed\_thresh$ amount using the $c\_compressed_{p,t} \in \mathbb{R}^{\geq 0}$ variable. The model

[1]As a reminder, we defined the inefficiency threshold period in Subsection A.5.2.

defines the following constraints to incorporate the cost of compressed deliveries logic:

$$
\begin{aligned}
&\text{If } t > t\_current_p, \\
&\qquad c\_compressed_{p,t} = coeff\_trenching \cdot amount\_compressed_{p,t} \\
&\qquad\qquad\qquad\qquad \text{for } t\_inefficiency_p \leq t < t\_comm_p, \qquad \forall p \\
&\qquad c\_compressed_{p,t} = coeff\_trenching \cdot \sum_b x_{p,t,b} \\
&\qquad\qquad\qquad\qquad \text{for } t \geq t\_comm_p, \qquad \forall p \\
&\text{else,} \quad c\_compressed_{p,t} = 0, \qquad \forall p
\end{aligned} \tag{A.31}
$$

$coeff\_trenching_p \in \mathbb{R}^{\geq 0}$ is the cost per MW of trenching a certain number of MW in the project p. Notice that if a period has passed before the model is run, the model assumes that NEER paid the compressed delivery costs before running the model.

Recall that $maxreceive$ is the threshold of the maximum amount of MW a project can receive in one period.[2] The model uses this parameter in the following constraint:

$$\sum_b x_{p,t,b} \leq maxreceive \qquad \forall p, t \tag{A.32}$$

To visualize the compressed deliveries concept, we examine the example in Figure A-5, where $compressed\_thresh = 50$.[3]

[2]In future model enhancements, one could index the $compressed\_thresh$ and $maxreceive$ parameters by project to calculate more precise costs.

[3]Section 3.6 showed this previously as Figure 3-7.

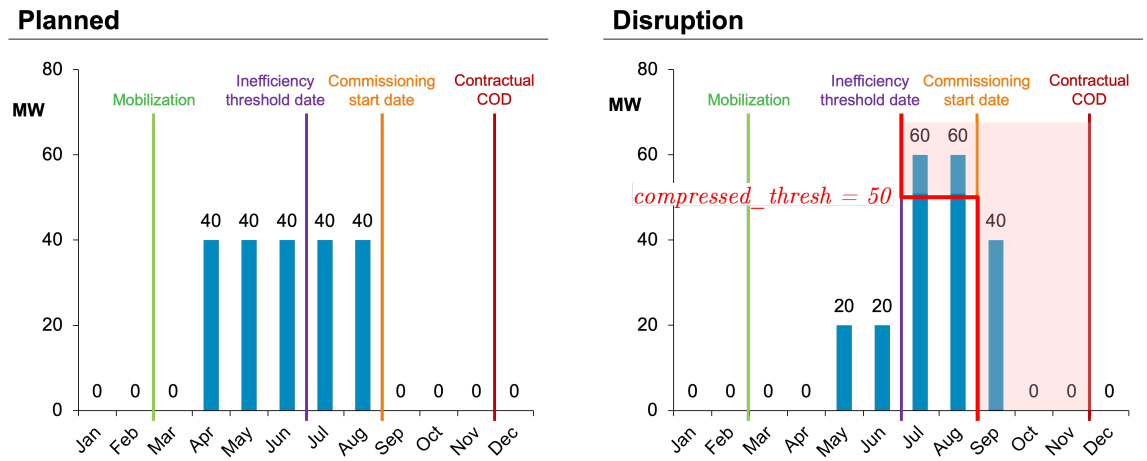

Figure A-5: Example of compressed deliveries in a project schedule after a disruption.

July and August are the months after the inefficiency threshold and before the commissioning start date. The project delivers 60MW in these months, so it must compress 10MW. In September, the project delivers 40MW after the commissioning start date. It will have to compress all of the 40MW. So, the project will need to compress 60MW: 10MW in July, 10MW in August, and 40MW in September.

A.7 Late deliveries

Since a project may incur either LDs or termination penalties, the model calculates the costs incurred due to late deliveries in the variable $c\_late\_deliveries_p$. That is, $c\_late\_deliveries_p \in \mathbb{R}^{\geq 0}$ is the variable that calculates the cost of project p not receiving all of the MW required by the contractual COD or the outside COD using the ensuing constraint:

$$c\_late\_deliveries_p = c\_LD\_with\_term_p + c\_termination_p \qquad \forall p \qquad \text{(A.33)}$$

A.7.1 Liquidated damages

We introduce the following elements in this section:

- The variable $c\_LD_p \in \mathbb{R}^{\geq 0}$ is the cost of project p incurring in LDs.
- The parameter $coeff\_LD_p \in \mathbb{R}^{\geq 0}$ is the cost per period of incurring liquidated damages for project p.
- The parameter $t\_cont\_COD_p \in \mathbb{Z}^{+}$ is the established contractual COD of project p.[4]

We establish that a project incurs LDs if the required MW of project p are not delivered at least a certain number of periods before the contractual COD. We used a fixed two periods before the contractual COD in this model, but this number could be parametrized if required.[5] For each period later, the project will incur LDs. That is because after finishing deliveries, the EPC still needs to do activities to complete a project.

Additionally, if the contractual COD of a project has passed before the model is run, then the model assumes NEER to have already paid the LDs. As such, projects where $t\_cont\_COD_p \leq t\_current$ have a zero value in c_LD_p. We introduce this

[4] The date of the contractual COD is mapped to a period in the set $\mathscr{T}$ using the $t\_cont\_COD_p$ parameter.

[5] In this case, we assume that deliveries need to finish at least two periods before the contractual COD, but the number of periods can change depending on the context of the model.

logic using the mathematical constraints:[6]

$$\text{If } t\_cont\_COD_p \leq t\_current, \quad c\_LD_p = 0 \quad \forall p$$

else,

$$c\_LD_p \geq coeff\_LD_p \cdot \left(t\_out\_COD_p - \left(\sum_t finished\_delivery_{p,t} - 1 \right) - (t\_cont\_COD_p - 2) \right) \quad \forall p \quad (A.34)$$

A.7.2 Terminations

The logic behind the termination concept is as follows: First, we restrict deliveries of all projects so that they do not happen after the outside COD. If we allow deliveries after the outside COD, then projects could be completed after the outside COD. It is illogical to have deliveries after the COD because, by definition, if not enough MW have been delivered to a project by the outside COD, that project must be terminated. In other words, the outside COD is the deadline for a project to receive deliveries and avoid termination. In the model, the parameter $t\_out\_COD_p \in \mathbb{Z}^+$ determines the established outside COD of project p.[7]

We also restrict terminated projects from having future deliveries. It is an optional constraint, as minimizing the objective function will likely result in no deliveries for terminated projects. However, similar to constraint (A.7), we add it to improve the performance of exercises we do not solve to optimality. The model introduces this

[6]These logical statements are linear equations and do not need the big-$\mathcal{M}$ method since $t\_cont\_COD_p$ and $t\_current$ are both parameters.

[7]The date of the outside COD is mapped to a period in the set $\mathscr{T}$ using the $t\_out\_COD_p$ parameter.

logic in the ensuing constraint.

$$x_{p,t,b} = 0 \qquad \text{for } t \geq t\_out\_COD_p \quad \forall p,t,b \tag{A.35}$$

$$x_{p,t,b} \leq MW_p \cdot (1 - terminate_p) \qquad \text{for } t > t\_current \quad \forall p,t,b \tag{A.36}$$

In conjunction with constraint (3.7), constraint (A.35) enforces the logic that the model terminates a project if it cannot finish deliveries by the outside COD. A terminated project could have positive MW if it had deliveries in past periods, or $delivered_{p,t,b}$ is positive for $t < t\_current$ for the terminated project.

To calculate the cost of terminating a project p, we introduce the parameter $coeff\_termination_p \in \mathbb{R}^{\geq 0}$. A project only incurs termination costs if the model decides to terminate the project. The variable $c\_termination_p \in \mathbb{R}^{\geq 0}$ captures the cost of termination if the model terminates project p, or lack thereof if it does not terminate it. Additionally, if the outside COD of a project has passed, then the model assumes that NEER has already paid the termination cost. As such, projects where $t\_out\_COD_p \leq t\_current$ have a zero value in c_LD_p. The constraints that the model uses are the following:[8]

$$\begin{aligned} &\text{If } t\_cont\_COD_p \leq t\_current, \quad c\_termination_p = 0 && \forall p \\ &\text{else,} \quad c\_termination_p = coeff\_termination_p \cdot terminate_p && \forall p \end{aligned} \tag{A.37}$$

To avoid double counting costs, we establish that a project does not incur LDs if the model decides to terminate it. It stems from the logic that if NEER knows that they will have to terminate a project, they will likely look to avoid other costs. The

[8]The logical statements are linear equations, and the model does not need to use the big-$\mathcal{M}$ method since $t\_out\_COD_p$ and $t\_current$ are both parameters.

model introduces the variable $c\_LD\_with\_term_p \in \mathbb{R}^{\geq 0}$ to calculate the product between the c_LD_p and $terminate_p$ variables. Using this variable helps keep track of the actual late delivery costs since a project only incurs LDs if the model does not terminate it at the time the algorithm runs. In the form of constraints, it looks like:

$$c\_LD\_with\_term_p \geq c\_LD_p - \mathcal{M}\_termination\_LD \cdot terminate_p \qquad \forall p \quad \text{(A.38)}$$

$$c\_LD\_with\_term_p \leq \mathcal{M}\_termination\_LD \cdot (1 - terminate_p) \qquad \forall p \quad \text{(A.39)}$$

where $\mathcal{M}\_termination\_LD \in \mathbb{R}^{+}$ is the "big-$\mathcal{M}$" parameter that ensures that the constraints related to the $terminate_p$ and $c\_LD\_with\_term_p$ variables are linear.

A.8 Early deliveries

The parameters that keep track of the early delivery milestones are t_mob_p and $t\_first\_delivery_p$. Explicitly, $t\_mob_p \in \mathbb{Z}^{+}$ is the established period to mobilize the EPC for project p. The parameter $t\_first\_delivery_p \in \mathbb{Z}^{+}$ is the first-period project p contracted deliveries. It is the earliest time t when $contracted_{p,t,b} > 0$ for any b for project p, and it is computed based on $contracted_{p,t,b}$. These parameters map actual dates to periods in the set $\mathscr{T}$.

The warehouse and the laydown yard are scaled with MW_{dc} as a unit. The coefficients used to calculate the costs are $coeff\_warehouse_p$ and $coeff\_laydown\_yard_p$. The parameter $coeff\_warehouse_p \in \mathbb{R}^{\geq 0}$ is the cost per MW of storing panels in a warehouse for project p if they need to be delivered before the EPC can access the project site. The parameter $coeff\_laydown\_yard_p \in \mathbb{R}^{\geq 0}$ is the cost per MW of creating a laydown yard in the project p to store panels the suppliers deliver before the project mobilization starts. Together, they calculate if project p

receives deliveries in a time t before its planned first delivery date in the variable $c\_early\_deliveries_{p,t} \in \mathbb{R}^{\geq 0}$ using the constraint:

$$\begin{aligned}
&\text{If } t_current_p < t < t_mob_p, \\
&\qquad c_early_deliveries_{p,t} = coeff_warehouse_p \cdot \sum_b \sum_{i=0}^{t} x_{p,i,b} \qquad \forall p \\
&\text{else if } t_current_p < t < t_first_delivery_p, \\
&\qquad c_early_deliveries_{p,t} = coeff_laydown_yard_p \cdot \sum_b x_{p,t,b} \qquad \forall p \\
&\text{else, } c_early_deliveries_{p,t} = 0 \qquad \forall p
\end{aligned} \tag{A.40}$$

Similar to previous constraints, if a period has passed, the model assumes that NEER has paid the early delivery costs before implementing the algorithm.

A.9 Remobilization

Recall from Section 3.9 that the remobilization costs linearly scaled with the number of remobilization gaps a project has.[9] This part of the Appendix defines the variables that comprise the $has\_remob\_gap_{p,t}$ variable. In the model's constraints, $has\_remob\_gap_{p,t}$ is calculated using:

$$made\_delivery_{p,t} + has\_delivered_{p,t} + has\_gap_{p,t} \leq 2 + has\_remob\_gap_{p,t} \qquad \forall p, t \tag{A.41}$$

[9] If a period has passed, the model assumes the remobilization costs as paid before the implementation of the algorithm, and the model does not account for the period's cost in the overall remobilization cost.

From this constraint, we can see that the model introduces the $has\_gap_{p,t} \in \{0, 1\}$ variable, which is the binary variable to indicate if project p made no deliveries in the past δ_remob_p periods or more before time t. The following logic defines this variable:

$$\text{If} \sum_{i=1}^{\delta\_remob_p} made\_delivery_{p,t-i} = 0, \quad has\_gap_{p,t} = 1$$
$$\text{else}, \quad has\_gap_{p,t} = 0 \qquad \forall p, t$$

Alternatively, the following constraints translate this logic into linear equations:

$$\sum_{i=1}^{\delta\_remob_p} made\_delivery_{p,t-i} \leq \mathcal{M}\_remob \cdot (1 - has\_gap_{p,t}) \qquad \forall p, t \qquad \text{(A.42)}$$
$$\sum_{i=1}^{\delta\_remob_p} made\_delivery_{p,t-i} \geq 1 - has\_gap_{p,t} \qquad \forall p, t \qquad \text{(A.43)}$$

$\mathcal{M}\_deliveries \in \mathbb{R}^+$ is a "big-$\mathcal{M}$" that ensures that the model assigns the $has\_gap_{p,t}$ variables to the expected binary values based on the previous constraint.

In the model, $\delta\_remob_p \in \mathbb{Z}^+$ is the established number of periods after which project p incurs a remobilization cost if project p has received a delivery and has a gap of this number of periods between deliveries. δ_remob_p typically equals 2 months. $coeff\_remob_p \in \mathbb{R}^{\geq 0}$ is the cost of remobilizing project p after stopping deliveries for δ_remob_p periods.

The $has\_gap_{p,t}$ variables calculate if a project did not have deliveries in the last δ_remob_p periods, but they do not calculate if there is a remobilization gap. For example, suppose that there is a project that has yet to make deliveries until period

3 and $\delta\_remob_p = 2$. Then, substituting in A.42 and A.43 for $t = 3$, for a given p we have:

$$\sum_{i=1}^{2} made\_delivery_{p,3-i} \leq \mathcal{M}\_remob \cdot (1 - has\_gap_{p,3})$$

$$\Rightarrow made\_delivery_{p,2} + made\_delivery_{p,1} \leq \mathcal{M}\_remob \cdot (1 - has\_gap_{p,3})$$

$$\Rightarrow 0 + 0 \leq \mathcal{M}\_remob \cdot (1 - has\_gap_{p,3})$$

$$\Rightarrow 0 \leq \mathcal{M}\_remob \cdot (1 - has\_gap_{p,3})$$

$$\Rightarrow has\_gap_{p,3} = 1 \text{ or } has\_gap_{p,3} = 0$$

and

$$\sum_{i=1}^{2} made\_delivery_{p,3-i} \geq 1 - has\_gap_{p,3}$$

$$\Rightarrow made\_delivery_{p,2} + made\_delivery_{p,1} \geq 1 - has\_gap_{p,3}$$

$$\Rightarrow 0 + 0 \geq 1 - has\_gap_{p,3}$$

$$\Rightarrow 0 \geq 1 - has\_gap_{p,3}$$

$$\Rightarrow has\_gap_{p,3} \geq 1$$

$$\Rightarrow has\_gap_{p,3} = 1$$

$$\Rightarrow has\_gap_{p,3} = 1$$

This example shows that all periods without deliveries in δ_remob_p periods will have $has\_gap_{p,t} = 1$, regardless of whether the project has started deliveries. As such, we need all the variables $made\_delivery_{p,t}$, $has\_delivered_{p,t}$, and $has\_gap_{p,t}$ to equal 1, to have a remobilization gap.[10]

Recall that a project can have multiple remobilizations. For example, assume

[10] See Section 3.9 for more detail.

a project has the delivery schedule in Table A.9.1 when $\delta\_remob_p = 2$. The table shows the values for the $made\_delivery_{p,t}$, $has\_delivered_{p,t}$, $has\_gap_{p,t}$, and $has\_remob\_gap_{p,t}$ variables.

t	1	2	3	4	5	6	7	8	9	10
$\sum_b x_{p,t,b}$	0	0	50	0	0	50	0	0	50	50
$made\_delivery_{p,t}$	0	0	1	0	0	1	0	0	1	1
$has\_delivered_{p,t}$	0	0	0	1	1	1	1	1	1	1
$has\_gap_{p,t}$	0	0	1	0	0	1	0	0	1	0
$has\_remob\_gap_{p,t}$	0	0	0	0	0	1	0	0	1	0

Table A.9.1: Example of a schedule with multiple remobilization gaps.

This project has two remobilizations: one in period 6 and another in period 9. In these columns, $made\_delivery_{p,t}$, $has\_delivered_{p,t}$, and $has\_gap_{p,t}$ all equal one, so $has\_remob\_gap_{p,t}$ also equals one.

A.10 Bin type changes

This section handles the logic of NEER sending a different amount of solar panels of a particular bin type to a project than the one they had contracted for that project. These constraints include the logic of buying new panels and discarding panels because of a terminated project.

A.10.1 Potential exchanges

Before diving into the details of the constraints, we introduce the concept of projects exchanging solar panels. We define that a project makes an exchange if it receives more panels of a bin type it did not contract or if it gets fewer panels of a bin type it contacted at the end of the solar panel allocation. The model tracks exchanges using the $exchange_{p,b_{old},b_{new}} \in \mathbb{R}^{\geq 0}$ variable, which is the amount of MW that project p exchanges of panels with bin type b_{old} in return for panels with bin type b_{new}.

We also explore the four different types of exchanges a project can make:

- A project exchanges panels of one supplier with those of another supplier.
- A project loses panels because of a disruption.
- A project discards panels because the model terminates it.
- A project procures new panels outside of its initial contract.

We describe each type of exchange below.

Exchange of panels between suppliers Let us review a straightforward example where we have two projects, 1 and 2, and two suppliers, A and B. Project 1 contracted 360 MW worth of panels with bin type A, and Project 2 contracted 360 MW worth of panels with bin type B. Figure A-6 visualizes the example.

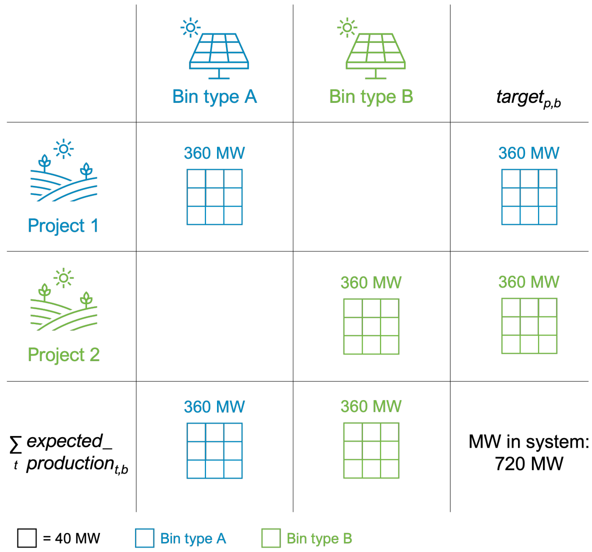

Figure A-6: Example of two projects that contracted different bin types.

As we advance, we will introduce different scenarios with disruptions. For all of them, we use the allocation from Figure A-6 as the official contracted schedule. Then, for the examples from now on, we have the following:

$$\sum_t contracted_{1,t,A} = target\_bin_{1,A} = 360$$

$$\sum_t contracted_{1,t,B} = target\_bin_{1,B} = 0$$

$$\sum_t contracted_{2,t,A} = target\_bin_{2,A} = 0$$

$$\sum_t contracted_{2,t,B} = target\_bin_{2,B} = 360$$

$$\sum_t expected\_production_{t,A} = 360$$

$$\sum_t expected\_production_{t,B} = 360$$

Assume that after a disruption, the schedule from Figure A-6 changes, and now Project 1 receives 240 MW of bin type A and 120 MW of bin type B, and Project 2 receives 120 MW of bin type A and 240 MW of bin type B. Figure A-7 visualizes this new schedule.

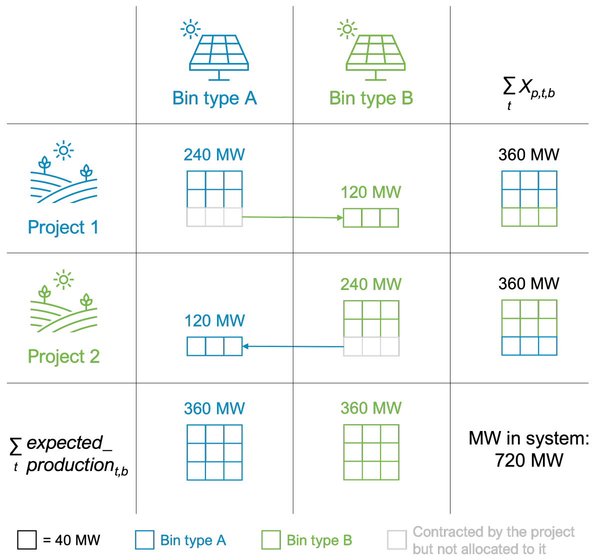

Figure A-7: Example of projects that make an exchange between suppliers.

Additionally, the values for the $exchange_{p,b_{old},b_{new}}$ variables are:

$$exchange_{1,A,B} = 120$$

$$exchange_{1,B,A} = 0$$

$$exchange_{2,A,B} = 0$$

$$exchange_{2,B,A} = 120$$

In practice, what is happening is that Project 1 is not using 120 MW of bin type A so that Project 2 can use them, and Project 2 is not using 120 MW of bin type B so that Project 1 can use them. While we can infer which project transfers MW to another, we do not keep track of the interactions between projects; we just keep track of the absolute change the projects had in panel types. Calculating the interactions would require us to introduce a variable indexed on p_1, p_2, b_{old}, b_{new}, adding much complexity to the model. The complexity outweighs the usefulness of a variable like this.

Also, notice that the total values of each supplier and each project did not change by looking at the total of columns and rows. Projects 1 and 2 still need to receive the same amount of MW to complete, and suppliers A and B still produce the same amount of MW. As such, the MW in the system remain the same. Whether there is an exchange or not, the MW in the system will always remain the same.

Panel losses Using the scenario in Figure A-6 as a starting point, let us assume there is a disruption where Supplier A no longer has production. Regarding the model's parameter, $\sum_t expected\_production_{t,A} = 0$. It means that there will be 360 MW less in the system. Also, assume that Project 1 has a higher priority than Project 2. Since Project 1 is a priority, the model would allocate all of the MW from Supplier B to Project 1 and terminate Project 2. Figure A-8 presents the results from the allocation.

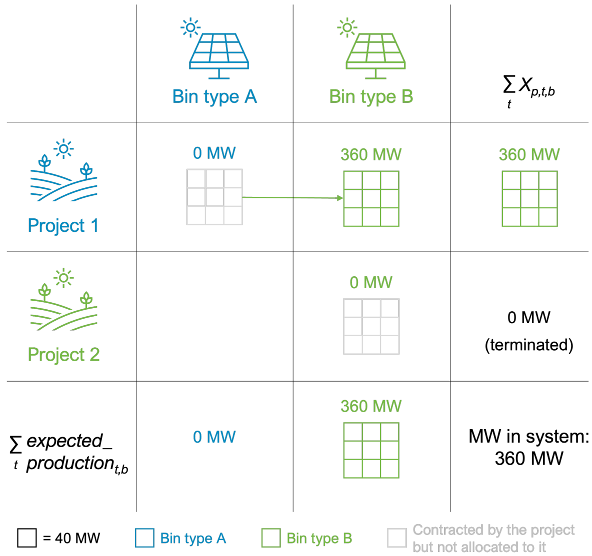

Figure A-8: Example of exchanges when a project loses MW.

Calculating the $exchange_{p,b_{old},b_{new}}$ variables for Project 1 follows the logic we used in the previous example. However, to represent the decrease in MW and termination that Project 2 had, we need to introduce a new concept. Project 2 initially had contracted 360 MW worth of panels with bin type B, but the model now allocated 0 MW of all bin types. Intuitively, Project 2 exchanges 360 MW worth of panels with bin type B for nothing in return. We introduce the abstract representation of $None$-type panels to make this exchange through the one item set $\mathscr{G} = \{None\}$. $None$ can be thought of as a fictional bin type. With this in mind, we can now calculate the values for $exchange_{p,b_{old},b_{new}}$:

$$exchange_{1,A,B} = 360$$
$$exchange_{1,B,A} = 0$$
$$exchange_{1,A,None} = 0$$
$$exchange_{1,B,None} = 0$$
$$exchange_{2,A,B} = 0$$
$$exchange_{2,B,A} = 0$$
$$exchange_{2,A,None} = 0$$
$$exchange_{2,B,None} = 360$$

No exchanges go from $None$ to another panel type because all the projects start with an allocated MW volume equal to the required volume. In other words, projects cannot have more MW allocated than they need to be completed (recall constraint (A.7)). In other words, a project can not gain MW overall; it can just gain MW of a specific bin type if it loses panels of another bin type. Again, this goes back to the logic that the MW in the system always remain the same. Therefore, for $exchange_{p,b_{old},b_{new}}$, $p \in \mathscr{P}$, $b_{old} \in \mathscr{B}$, and $b_{new} \in \mathscr{B} \cup \mathscr{G}$.

A project ending up with 0 MW is only one of many where exchanges with disparate amounts can happen. We use the scenario in Figure A-6 as a starting point and a disruption where Supplier A delivered 240 MW to Project 1 and Supplier B delivered 240 MW to Project 2. Then, a disruption happens that results in Supplier A being unable to provide the remaining 120 MW to Project 1. That is, $\sum_t expected\_production_{t,A} = 240$, and there are 120 fewer MW in the system. If Project 1 is still a priority, Supplier B would send the remaining 120 MW to Project

1 instead of Project 2. The model would terminate Project 2. The values for the $exchange_{p,b_{old},b_{new}}$ variables are below, and the allocation visualization is in Figure A-9.

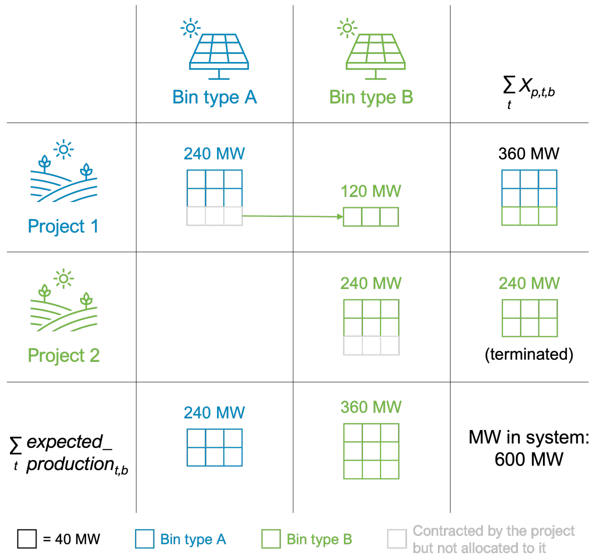

Figure A-9: Another example of exchanges when a project loses MW.

$$exchange_{1,A,B} = 120$$

$$exchange_{1,B,A} = 0$$

$$exchange_{1,A,None} = 0$$

$$exchange_{1,B,None} = 0$$

$$exchange_{2,A,B} = 0$$

$$exchange_{2,B,A} = 0$$

$$exchange_{2,A,None} = 0$$

$$exchange_{2,B,None} = 120$$

Discarding panels The model may terminate a project even if the system has enough MW to complete every project because the projects need to receive the MW by a specific deadline (see constraints (3.7) and (A.35)). In these instances, a project will exchange the panels it has yet to receive for the $None$-type representation. Since no project receives these unclaimed MW, the model discarded them. Discarding the MW of a particular bin type is equivalent to the model not allocating these MW to a project when they have already been contracted.

We introduce the new concept of the $Unallocated$ abstraction to handle instances of discarded MW while maintaining the logic that the overall volume in the system remains the same when there are changes to the allocation. It is an abstract representation of a location with the panels that the suppliers can produce but that the model decides not to deliver to any project. $Unallocated$ can be thought of as a fictional project. We also introduce the item set $\mathscr{X} = \{Unallocated\}$ to facilitate managing indexes in the model when using the $Unallocated$ project abstraction.

To better understand the concept of the $Unallocated$ project abstraction, suppose a disruption delays Supplier B's MW by a year. Figure A-6 is the starting point, and the model terminates Project 2. Then, the model will discard the MW Project 2 contracted to $Unallocated$. Figure A-10 visualizes this allocation, and the $exchange_{p,b_{old},b_{new}}$ are presented below.

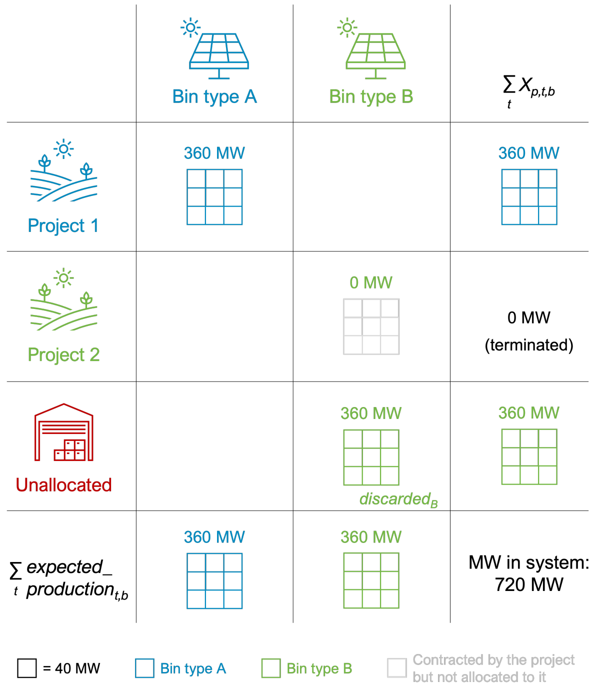

Figure A-10: Example of exchanges when a project discards MW.

$$exchange_{1,A,B} = 0$$

$$exchange_{1,B,A} = 0$$

$$exchange_{1,A,None} = 0$$

$$exchange_{1,B,None} = 0$$

$$exchange_{2,A,B} = 0$$

$$exchange_{2,B,A} = 0$$

$$exchange_{2,A,None} = 0$$

$$exchange_{2,B,None} = 360$$

Notice that there are no $exchange_{p,b_{old},b_{new}}$ for *Unallocated*. Unlike real projects, *Unallocated* can receive MW even when it did not have MW allocated at the start of an exercise through the MW that projects discard. We introduce the $discarded_b \in \mathbb{R}^{\geq 0}$ variable to track the number of MW contracted using panels with bin type b, which the supplier will no longer deliver to any project. In the example from Figure A-10, the values of the $discarded_b$ variable are:

$$discarded_A = 0$$
$$discarded_B = 360$$

Buying new panels Another type of exchange happens when a project buys panels from a supplier without a pre-existing contract using the $x\_new\_buy_{p,t,b}$ variables defined in Subsection 3.2.1. Recall that a project could buy MW from a supplier without a pre-existing contract if the supplier could produce it. It is capacity in the system but not allocated to any project. As such, if a supplier has an additional capacity to produce more MW than it is obligated to based on its contract, these additional MW will be initially assigned to the *Unallocated* abstraction. The model can deliver these panels to projects at an extra cost. We amend the $target_{p,b}$ variables first presented in constraint (A.11) to be indexed over $p \in \mathscr{P} \cup \mathscr{Z}$ and $b \in \mathscr{B}$.

We modify the example in Figure A-6 so Supplier A has an additional 360 MW of unallocated capacity. Then, the parameters for this example are:

$$\sum_t expected\_production_{t,A} = 720$$
$$\sum_t expected\_production_{t,B} = 360$$
$$target\_bin_{1,A} = 360$$
$$target\_bin_{1,B} = 0$$
$$target\_bin_{2,A} = 0$$
$$target\_bin_{2,B} = 360$$
$$target_{Unallocated,A} = 360$$
$$target_{Unallocated,B} = 0$$

Suppose the production of Supplier B is delayed, as in the example in Figure A-11. In that case, the model will use the unallocated 360 MW from Supplier A to complete Project 2, as it is very costly to terminate a project. Figure A-11 visualizes the example, and the values of the $exchange_{p,b_{old},b_{new}}$ and $discarded_b$ variables are below.

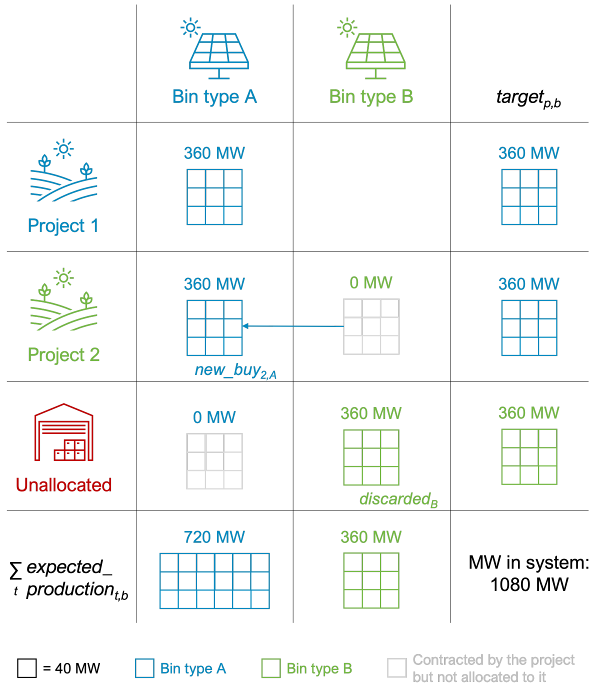

Figure A-11: Example of exchanges when a project buys new MW.

$$exchange_{1,A,B} = 0$$
$$exchange_{1,B,A} = 0$$
$$exchange_{1,A,None} = 0$$
$$exchange_{1,B,None} = 0$$
$$exchange_{2,A,B} = 0$$
$$exchange_{2,B,A} = 0$$
$$exchange_{2,A,None} = 0$$
$$exchange_{2,B,None} = 360$$
$$discarded_A = 0$$
$$discarded_B = 360$$

A.10.2 Bin type changes

A.10.2.1 Type change variables definition

We mentioned that the volume in the model remains the same after the exchanges. To model this behavior, we first define a new variable that captures the absolute difference between the $target_{p,b}$ and the allocated values for each project. The differences in examples from the previous sections were easy to calculate because we only had two suppliers, but it is more complex when we increase the number of projects. For instance, we can have that for a project p, $target_{p,A} = 180$, $exchange_{p,A,B} = 100$, $exchange_{p,A,C} = 50$, $exchange_{p,A,B} = 30$, and the rest of the $target_{p,b}$ and $exchange_{p,b_{old},b_{new}}$ variables equal 0. We want a variable tracking that project p received 180 less MW than it contracted, regardless of the MW from other bin types that it used to substitute them.

The model introduces the variable $type\_change_{p,b} \in \mathbb{R}$ to represent the increase or decrease in MW from panels with bin type b when comparing the project p's resulting schedule to its initial schedule. Positive values of $type\_change_{p,b}$ mean project p has more panels with bin type b than initially anticipated, and negative values mean it has less. The model assumes that a project makes a panel bin type change if there is a difference between the target panels of the project and the panels that the model determines to deliver in the exercise, so it is constrained by:

$$type\_change_{p,b} = \sum_{t} x_{p,t,b} - target\_bin_{p,b} \quad \forall p \in \mathscr{P}, \quad \forall b \tag{A.44}$$

We also want to keep track of how the MW of the abstract location *Unallocated* changed. The model does not use $x_{p,t,b}$ variables for *Unallocated*, so *type_change* is defined differently for *Unallocated* than for projects $p \in \mathscr{P}$. Instead, the definition for $type\_change_{Unallocated,b}$ is based on the intuition that *Unallocated* has more MW with a bin type if projects discard some of their MW and fewer MW with a bin type if projects procure new panels. Therefore,

$$type\_change_{z,b} = discarded_b - \sum_{p \in \mathscr{P}} \sum_{t} x\_new\_buy_{p,t,b} \quad \text{for } z \in \mathscr{Z}, \quad \forall b \tag{A.45}$$

From now on, we will use $z = Unallocated \in \mathscr{Z}$ to simplify the notation of the *Unallocated* abstraction.

Additionally, we want to differentiate the cases when $type\_change_{p,b}$ is positive from the ones when it is negative. We need this distinction because if $type\_change_{p,b}$ is positive, project p must have received MW of bin type b that other projects contracted or the *Unallocated* representation had. If it is negative, other projects (including the *Unallocated* representation) received MW that p contracted. Therefore, we introduce $type\_change^+_{p,b} \in \mathbb{R}^{\geq 0}$ and $type\_change^-_{p,b} \in \mathbb{R}^{\geq 0}$ as the positive and negative parts of the $type\_change_{p,b}$ variable. A non-zero value of $type\_change^+_{p,b}$ means that project p receives more panels with bin type b than initially anticipated. A non-zero value of $type\_change^-_{p,b}$ means that project p has fewer panels with bin type b than initially anticipated. The model introduces the variables with the following constraint:

$$type\_change_{p,b} = type\_change^+_{p,b} - type\_change^-_{p,b} \qquad \forall p \in \mathscr{P} \cup \mathscr{Z}, \quad \forall b \quad \text{(A.46)}$$

To correctly isolate the positive and negative parts of $type\_change_{p,b}$, we must ensure that at most one of $type\_change^+_{p,b}$ and $type\_change^-_{p,b}$ is positive while the other is zero. For example, we want to avoid a solution where

$$\begin{aligned} type\_change_{p,b} &= 10 \\ type\_change^+_{p,b} &= 15 \\ type\_change^-_{p,b} &= 5, \end{aligned}$$

because $type\_change^+_{p,b}$ is estimating that project p received 5 more MW with bin type b than initially anticipated. Instead, we would want to have

$$\begin{aligned} type\_change_{p,b} &= 10 \\ type\_change^+_{p,b} &= 10 \\ type\_change^-_{p,b} &= 0. \end{aligned}$$

Therefore, we introduce the following constraints:

$$type\_change^{+}_{p,b} = \mathcal{M}\_type\_change\_composition\_activation \cdot binary\_type\_change_{p,b} \quad \forall p \in \mathscr{P} \cup \mathscr{Z}, \quad \forall b \tag{A.47}$$

$$type\_change^{-}_{p,b} = \mathcal{M}\_type\_change\_composition\_activation \cdot (1 - binary\_type\_change_{p,b}) \quad \forall p \in \mathscr{P} \cup \mathscr{Z}, \quad \forall b \tag{A.48}$$

where $\mathcal{M}\_type\_change\_composition\_activation \in \mathbb{R}^{+}$ is the "big-$\mathcal{M}$" parameter and $binary\_type\_change_{p,b} \in \{0,1\}$ is the binary auxiliary variable that together ensure that at most one of $type\_change^{+}$ and $type\_change^{-}$ take a positive value.

We can now tie the $type\_change_{p,b}$ variables to the $exchange_{p,b_{old},b_{new}}$ variables. Notice that for a project to receive MW of a particular bin type (or $type\_change^{+}_{p,b} > 0$), it must have exchanged to receive these MW. Similarly, for a project to have less MW than intended of a particular bin type, it must have made exchanges to stop receiving these MW. In the model constraints, this logic is established by:

$$type\_change^{+}_{p,b_{new}} = \sum_{b_{old} \in \mathscr{B}} exchange_{p,b_{old},b_{new}} \quad \forall p \in \mathscr{P}, \quad \forall b_{new} \in \mathscr{B} \tag{A.49}$$

$$type\_change^{-}_{p,b_{old}} = \sum_{b_{new} \in \mathscr{B}} exchange_{p,b_{old},b_{new}} \quad \forall p \in \mathscr{P}, \quad \forall b_{old} \in \mathscr{B} \tag{A.50}$$

We also want to tie the $type\_change_{p,b}$ variables to the $discarded_{b}$ variables. To do so, we assume that for the *Unallocated* abstraction to receive MW, other projects must discard these MW, captured in:

$$discarded_b = type\_change^+_{z,b} \quad \text{for } z \in \mathscr{Z}, \quad \forall b \in \mathscr{B} \tag{A.51}$$

Notice that constraint (A.51) also constrains $\sum_{p \in \mathscr{P}} \sum_t x\_new\_buy_{p,t,b}$ because

$$type\_change^+_{z,b} - type\_change^-_{z,b} = discarded_b - \sum_{p \in \mathscr{P}} \sum_t x\_new\_buy_{p,t,b}$$
$$\text{for } z \in \mathscr{Z}, \quad \forall b$$
$$\text{(from } A.45 \text{ and } A.46\text{)}$$
$$\Rightarrow discarded_b - type\_change^-_{z,b} = discarded_b - \sum_{p \in \mathscr{P}} \sum_t x\_new\_buy_{p,t,b}$$
$$\text{for } z \in \mathscr{Z}, \quad \forall b$$
$$\text{(from } A.51\text{)}$$
$$\Rightarrow type\_change^-_{z,b} = \sum_{p \in \mathscr{P}} \sum_t x\_new\_buy_{p,t,b} \qquad \text{for } z \in \mathscr{Z}, \quad \forall b$$

Table A.10.1 shows the values of the $type\_change_{p,b}$ variables for the example in Figure A-11.

p	b	$type\_change_{p,b}$	$type\_change^{+}_{p,b}$	$type\_change^{-}_{p,b}$
1	A	0	0	0
1	B	0	0	0
2	A	360	360	0
2	B	-360	0	360
Unallocated	A	-360	0	360
Unallocated	B	360	360	0

Table A.10.1: Values of the type change variables for the example in Figure A-11.

The requirement we introduced in Section 3.2.2 that deliveries cannot be too small is the base for the last constraints that define the t$type_change_{p,b}$ variables. Similarly, we require panel bin type changes to be made to a specific size. In particular, panel bin type changes must be at least $min\_delivery\_amount$ or equal to zero. The model constrains $type\_change_{p,b}$ to enforce this logic:

$$type\_change_{p,b} \leq MW_p \cdot type\_change\_min\_binary_{p,b} \qquad \forall p \in \mathscr{P}, \quad \forall b \quad (A.52)$$

$$type\_change_{p,b} \geq min\_delivery\_amount \cdot type\_change\_min\_binary_{p,b} \qquad \forall p \in \mathscr{P}, \quad \forall b \quad (A.53)$$

where $type\_change\_min\_binary_{p,b} \in \{0, 1\}$ is the binary auxiliary variable that ensures that $type\_change_{p,b}$ is either 0 or a positive number above the $min\_delivery\_amount$ threshold.

Since $type\_change_{p,b} = type\_change^{+}_{p,b}$ when $type\_change_{p,b} \geq 0$ and $type\_change_{p,b} = type\_change^{-}_{p,b}$ when $type\_change_{p,b} \leq 0$, the constraints A.52 and A.53 also constrain $type\_change^{+}_{p,b}$ and $type\_change^{-}_{p,b}$ to be at least $min\_delivery\_amount$ or equal to zero.

The $type\_change_{p,b}$ variables result from the $exchange_{p,b_{old},b_{new}}$ and $discarded_b$ variables, so one could argue that we could have avoided using the $type\_change_{p,b}$ altogether. However, having variables that $type\_change_{p,b}$ simplifies the notation and logic of many constraints we introduce next.

A.10.2.2 Fixed volume in the system

We now introduce the constraints that make the system's MW remain unchanged even when the model changes bin types. First, all the bin type changes that projects make must cancel each other out. If a project stops receiving MW of a specific bin type, the other projects (including the *Unallocated* abstraction) must have more MW of that bin type in the same amount. Conversely, suppose a project receives more MW of a specific bin type than the target. In that case, the other projects must receive less MW of that bin type in the same amount (in the case of the *Unallocated* abstraction, we can think of it receiving less as equivalent to a project procuring new panels). It means that if we sum all of the $type\_change_{p,b}$ variables for a particular b, we will get zero:

$$\sum_{p \in \mathscr{P} \cup \mathscr{Z}} type\_change_{p,b} = 0, \quad \forall b \in \mathscr{B} \tag{A.54}$$

Constraint (A.54) also enforces $\sum_{p \in \mathscr{P} \cup \mathscr{Z}} type\_change^{+}_{p,b} = \sum_{p \in \mathscr{P} \cup \mathscr{Z}} type\_change^{-}_{p,b} \quad \forall b \in \mathscr{B}$, which is logical and desired. Intuitively, it ensures that the model does not allocate more MW to projects than those available in the system and allocates all the system's MW to projects despite the changes in bin types.

Similar to how we constrained the $type\_change_{p,b}$ variables based on the projects having changes with a particular bin type, we have also constrained them based on the bin type changes of a specific project. We do not add additional constraints to the bin type changes that the *Unallocated* representation can make; it is only for the actual projects. First, we start by constraining a terminated project's bin type changes. Recall that in constraint (A.36), we established that a terminated project would not receive any more deliveries. To align the $type\_change_{p,b}$ variables to this logic, we enforce a constraint so that terminated projects cannot receive any MW in exchanges:

$$type\_change^{+}_{p,b} \leq \mathcal{M}\_type\_change\_terminate \cdot (1 - terminate_p) \quad \forall p, b \quad \text{(A.55)}$$

$\mathcal{M}\_type\_change\_terminate \in \mathbb{R}^{+}$ is the "big-$\mathcal{M}$" parameter that ensures the constraint (A.55) works as expected. Based on constraint (A.55), terminated projects will not use part or all of the MW they contracted but will not compensate this unused volume with MW of other bin types.

For a completed project, we have that if it stops receiving panels from one bin type, it needs to compensate them with panels of other bin types. We want the $\sum_b type\_change_{p,b}$ to be zero for a complete project. To do so, we introduce the ensuing constraints:

$$\sum_{b \in \mathscr{B}} type\_change_{p,b} \leq 0 \qquad \forall p \in \mathscr{P} \qquad \text{(A.56)}$$

$$\sum_{b \in \mathscr{B}} type\_change_{p,b} \geq -\mathcal{M}\_type\_change\_terminate \cdot terminate_p \qquad \forall p \in \mathscr{P} \qquad \text{(A.57)}$$

To see how the constraints A.56 and A.57 work, suppose that $terminate_p = 0$. Then,

$$\sum_{b \in \mathscr{B}} type\_change_{p,b} \leq 0, \qquad \forall p \in \mathscr{P}$$

$$\text{and } \sum_{b \in \mathscr{B}} type\_change_{p,b} \geq 0 \qquad \forall p \in \mathscr{P}$$

$$\Rightarrow \sum_{b \in \mathscr{B}} type\_change_{p,b} = 0 \qquad \forall p \in \mathscr{P}$$

Now, suppose that $terminate_p = 1$. Then,

$$\sum_{b \in \mathscr{B}} type\_change_{p,b} \leq 0, \qquad \forall p \in \mathscr{P}$$

$$\Rightarrow \sum_{b \in \mathscr{B}} -type\_change^{-}_{p,b} \leq 0, \qquad \forall p \in \mathscr{P}$$

$$\Rightarrow \sum_{b \in \mathscr{B}} type\_change^{-}_{p,b} \geq 0, \qquad \forall p \in \mathscr{P}$$

It is always true because $type\_change^{-}_{p,b} \geq 0 \quad \forall b \in \mathscr{B}$.

$$\sum_{b \in \mathscr{B}} type\_change_{p,b} \geq -\mathcal{M}\_type\_change\_terminate \cdot terminate_p \qquad \forall p \in \mathscr{P}$$

$$\Rightarrow \sum_{b \in \mathscr{B}} type\_change_{p,b} \geq -\mathcal{M}\_type\_change\_terminate \qquad \forall p \in \mathscr{P}$$

$$\Rightarrow \sum_{b \in \mathscr{B}} type\_change^{-}_{p,b} \geq -\mathcal{M}\_type\_change\_terminate \qquad \forall p \in \mathscr{P}$$

It is always true because $type\_change^{-}_{p,b} \geq 0 \quad \forall b \in \mathscr{B}$

and $-\mathcal{M}\_type\_change\_terminate < 0$.

As such, constraints A.56 and A.57 are irrelevant to a terminated project.

A.10.3 Costs of changing bin types

Now that we have defined the $type\_change_{p,b}$ variables and their behavior, we can use them to calculate the costs of changing the panel bin types.

A.10.3.1 Change orders

We introduce $decide\_change_p \in \{0,1\}$ as the binary auxiliary variable that indicates if project p has changed the number of panels for any bin type. When a specific project p changes a bin type, $type\_change^{+}_{p,b}$ will be positive for at least one b. Or, in logical statements,

$$\text{If } \sum_{b\in\mathscr{B}} type\_change^{+}_{p,b} > 0, \quad decide\_change_p = 1$$
$$\text{else,} \quad decide\_change_p = 0 \qquad \forall p \in \mathscr{P}$$

The big-$\mathcal{M}$ method turns this logical statement into linear equations:

$$\sum_{b\in\mathscr{B}} type\_change^{+}_{p,b} - \epsilon\_type\_change\_decide \geq - \mathcal{M}\_type\_change\_decide \cdot (1 - decide\_change_p) \qquad \forall p \in \mathscr{P} \tag{A.58}$$

$$\sum_{b\in\mathscr{B}} type\_change^{+}_{p,b} \leq \mathcal{M}\_type\_change\_decide \qquad \forall p \in \mathscr{P} \tag{A.59}$$

where $\mathcal{M}\_type\_change\_decide \in \mathbb{R}^{+}$ and $\epsilon\_type\_change\_decide \in (0,1)$ are the "big-$\mathcal{M}$" and the "small-ϵ" parameters, respectively, that ensure the expected functioning of constraints (A.58) and (A.59).

We assume that NEER decides to change the bin types of a project right after the algorithm runs. Therefore, we introduce the $post\_epc\_binary\_param_p \in \{0,1\}$ binary auxiliary parameter to indicate if the model runs after contracting an EPC for project p. Since the decision of when to make the bin type changes informs if there are costs, the model tracks if a project has decided to make changes and if that decision happened after NEER hired the project's EPC. Using $post\_epc\_binary\_param_p$ and $decide\_change_p$, we calculate the new variable $needs\_change\_order_p \in \{0,1\}$, which is the binary auxiliary variable that indicates if the algorithm is running before having started racking for project p and if project p had a bin type change. The model calculates it using the following constraint:

$$decide\_change_p + post\_epc\_binary\_param_p \leq 1 + needs\_change\_order_p \quad \forall p \tag{A.60}$$

Projects where $needs\_change\_order_p = 1$ will incur a change order cost. In particular, $coeff\_changeorder_p \in \mathbb{R}^{\geq 0}$ is the cost of making a change order resulting from delivering panels with different bin types to those contracted for project p after hiring the EPC.

A.10.3.2 Reracking costs

The variable $is\_gaining_{p,b} \in \{0,1\}$ is the binary auxiliary variable that indicates if project p has more volume of panels with bin type b than initially planned. It indicates if $type\_change^{+}_{p,b}$ is greater than zero. We track this variable because if a project has more MW of a particular bin type, it might need to modify some of its racking to adjust to the additional volume. We do not track when a project loses a particular bin type volume because it is irrelevant to the reracking cost. For example, if a project is terminated and loses MW, it will not pay for a reracking cost. The logic behind the $is\_gaining_{p,b}$ variable is the following:

$$\begin{aligned} \text{If } type\_change^{+}_{p,b} > 0, &\quad is\_gaining_{p,b} = 1 \\ \text{else}, &\quad is\_gaining_{p,b} = 0 \qquad\qquad \forall p, b \end{aligned}$$

Alternatively, using the big-$\mathcal{M}$ method to turn the logical statement into linear equations:

$$type\_change^{+}_{p,b} - \epsilon\_reracking \geq -\mathcal{M}\_reracking \cdot (1 - is\_gaining_{p,b}) \qquad \forall p, b \tag{A.61}$$

$$type\_change^{+}_{p,b} \leq \mathcal{M}\_reracking \cdot is\_gaining_{p,b} \qquad \forall p, b \tag{A.62}$$

where $\mathcal{M}\_reracking \in \mathbb{R}^{+}$ and $\epsilon\_reracking \in (0, 1)$ are the "big-$\mathcal{M}$" and "small-ϵ" parameters that ensure the model assigns $is\_gaining_{p,b}$ to the expected binary value and transform strict inequalities into non-strict inequalities for the constraints regarding the $is\_gaining_{p,b}$ variable.

As we did in the change order constraints, we introduce the $post\_rack\_binary\_param_{p} \in \{0, 1\}$ binary auxiliary parameter to indicate if the model runs after having started racking for project p. Using $post\_rack\_binary\_param_{p}$ and $is\_gaining_{p,b}$, we calculate the variable $needs\_reracking_{p,b} \in \{0, 1\}$, which is the binary auxiliary variable that indicates if project p needs reracking work due to the additional panels with bin type b it will receive. We use the following constraint to calculate it:

$$is\_gaining_{p,b} + post\_rack\_binary\_param_p \leq 1 + needs\_reracking_{p,b} \quad \forall p, b \tag{A.63}$$

Projects where $needs\_reracking_{p,b} = 1$ will incur a reracking cost.

The EPC will only make racking changes for the MW that require them. Therefore, the reracking cost is variable and scales with the exchanged MW. The reracking cost is also different depending on the exchanged bin types. We introduce the parameter $coeff\_reracking_{b_1,b_2} \in \mathbb{R}^{\geq 0}$ to represent the cost per MW of making a racking or hardware rework to allow for changing panels with bin type b_1 to those with bin type b_2. Also, we introduce the variable $activate\_reracking_{p,b_{old},b_{new}} \in \mathbb{R}^{\geq 0}$ to log the amount of MW project p needs to rerack because it changed panels with b_{old} bin type to panels with b_{new} bin type. Logically, $activate\_reracking_{p,b_{old},b_{new}}$ is the product between $needs\_reracking_{p,b_{new}}$ and $exchange_{p,b_{old},b_{new}}$. The big-$\mathcal{M}$ method turns this logical statement into linear equations:

$$activate\_reracking_{p,b_{old},b_{new}} \leq \mathcal{M}\_reracking \cdot needs\_reracking_{p,b_{new}} \quad \forall p \in \mathscr{P}, \quad \forall b_{old}, b_{new} \in \mathscr{B} \tag{A.64}$$

$$activate\_reracking_{p,b_{old},b_{new}} \leq exchange_{p,b_{old},b_{new}} \quad \forall p \in \mathscr{P}, \quad \forall b_{old}, b_{new} \in \mathscr{B} \tag{A.65}$$

$$activate\_reracking_{p,b_{old},b_{new}} \geq exchange_{p,b_{old},b_{new}} - \mathcal{M}\_reracking \cdot (1 - needs\_reracking_{p,b_{new}}) \quad \forall p \in \mathscr{P}, \quad \forall b_{old}, b_{new} \in \mathscr{B} \tag{A.66}$$

We sum the products of the $activate\_reracking_{p,b_{old},b_{new}}$ and $coeff\_reracking_{b_{old},b_{new}}$

variables over all of the b_{new} to calculate the total cost of reracking. $c\_reracking_{p,b}$ is the cost of the reracking work that project p needs to do to receive panels with bin type b in the model. Explicitly, we calculate it as:

$$c\_reracking_{p,b_{new}} = \sum_{b_{old}} (coeff\_reracking_{b_{old},b_{new}} \cdot activate\_reracking_{p,b_{old},b_{new}})$$

$$\text{for } b_{old} \in \mathscr{B}, \quad \forall p \in \mathscr{P}, \quad \forall b_{new} \in \mathscr{B} \quad \text{(A.67)}$$

Appendix B

Decomposition sequential approach

As introduced in Chapter 2, the decomposition sequential approach decreases the algorithm's runtime. We started by introducing a subset of the constraints and solving only for this subset. Once we had a solution, we added more constraints and initiated the algorithm using the solution obtained before as a hot start. We repeated this process until we added all of the constraints. If the parameters are correctly defined, the model should always be able to find a feasible solution because the $x_{p,t,b}$ variable is the base of all other variables. The $x_{p,t,b}$ variable can always take the value of zero for future periods, which is the primary time range of the decision space (past periods have a fixed volume in terms of $x_{p,t,b}$ and of the parameters). No constraints prevent $x_{p,t,b}$ from equaling zero for future periods, so $x_{p,t,b} = 0 \quad \text{for } t > t\_current \quad \forall p, b$ and $x_{p,t,b} = delivered_{p,t,b} \quad \text{for } t \leq t\_current \quad \forall p, b$ should always be a feasible solution. Therefore, we should always be able to use the solution from a previous step to start the next step. We present the pseudocode to introduce the constraints iteratively.

Algorithm Decomposition Sequential Approach Pseudocode

Input: model, input_data, subsets $c_1, c_2, ..., c_k$ of set of constraints C

```
i ← 1
while i ≤ k do
    h_i ← solve(model, input_data, ⋃_{j=1}^{i} c_j)
    i ← i + 1
return h_k
```

We based the subset of constraints in Chapter 2 and Appendix A. Based on trial and error, we found an order that reduced the runtime and produced a feasible solution in each step. The order in which we introduced the constraints is the following:

1. Delivery basics, Minimum MW per bin type and form type, Supplier capacities, and Compressed deliveries (Sections 3.2, 3.3, 3.4, and 3.6).
2. Late deliveries (Section 3.7).
3. Bin type changes (Section 3.10).
4. Remobilization (Section 3.9).
5. Changes to the weeks that the EPC crew works (Section 3.5).

Implementing the decomposition sequential approach helped reduce the model's runtime solve for the base scenario allocation from more than 2 hours to less than 8 minutes.

Appendix C

Extended tables

Table C.0.1: Costs mapped with internal stakeholders.

Team	Theme	Elements that can cause expenses

Development	Time of delivery	• Size of project
		• COD
		  ○ Renegotiation of COD
		  ○ Damages of not meeting COD
		• Execution date
		• Reprioritization based on
		  ○ Importance of customer
		  ○ Flexibility of provisions
		• Termination rights
		• Force majeures
		• Permits
		  ○ Reemission

Continued on next page

Table C.0.1: Costs mapped with internal stakeholders. (Continued)

Early Stage	Panel characteristics	• Alignment to geography ◦ Albedo ◦ Wind ◦ Slope • Panel technology • DC/AC ratio • Net capacity factor • Economic performance of the site • Safe harbor allocation • Optimal size of land
Procurement	Panel characteristics	• Panel types (N-type or P-type) • Bin classes

Continued on next page

Table C.0.1: Costs mapped with internal stakeholders. (Continued)

	Panel production	• Number of panels produced per supplier per week per bin
		• Extra capacity from
		    ○ Unassigned supply in MSA
		    ○ Safe harbor
		    ○ Stocking earlier at higher prices
		• Lower capacity from
		    ○ Factory delays
		    ○ Government intervention
		• Supplier price
		• Time of purchase
		• Quantity of purchase

Logistics	Panel production	• Extra capacity from
		    ○ Ramping production up
		• Lower capacity from
		    ○ Port delays
		    ○ Truck delays
		    ○ Lack of labor

	Time of delivery	• Delivery routes ○ Rerouting

	Panel reception	• Storing
		    ○ Extra days in ports
		    ○ In warehouses (short- or long-term)

Continued on next page

Table C.0.1: Costs mapped with internal stakeholders. (Continued)

Engineering & Construction	Panel characteristics	• Re-engineering costs with the EPC • Change in racking needs • Change in cables • Change in acreage required
	Time of delivery	• Commissioning

	Panel reception	• Offloading space
		• Access to site
		  ◦ Type
		  ◦ Date
		• Storing
		  ◦ In laydown yard
		• N1 delivery date
		• Labor redistribution / addition
		• Weather disruptions
		• De / re-mobilization
		• Extra delivery options due to
		  ◦ Double handling
		  ◦ Centralized offloading
		  ◦ New access points
		• Installation capacity depending on
		  ◦ State
		  ◦ Season

Bibliography

[1] NextEra Energy, Inc. *NEE 2022 Annual Report.* URL: https://www.investor.nexteraenergy.com/~/media/Files/N/NEE-IR/reports-and-fillings/annual-reports/NEE%202022%20Annual%20Report.pdf (visited on 03/11/2024).

[2] *NextEra tops leaderboard of renewable projects planned in energy communities.* URL: https://www.spglobal.com/marketintelligence/en/news-insights/research/nextera-tops-leaderboard-of-renewable-projects-planned-in-energy-communities (visited on 03/11/2024).

[3] *2022 Environmental, Social and Governance Report.* URL: https://www.nexteraenergy.com/content/dam/nee/us/en/pdf/2022_NEE_ESG_Report_Final.pdf (visited on 04/04/2024).

[4] *Net Generation by Energy Source: Total (All Sectors), 2012 - 2022.* U.S. Energy Information Administration (EIA). URL: https://www.eia.gov/electricity/annual/html/epa_03_01_a.html (visited on 03/11/2024).

[5] Elesia Fasching. *Electricity explained - Electricity generation, capacity, and sales in the United States.* U.S. Energy Information Administration (EIA). June 30, 2023. URL: https://www.eia.gov/energyexplained/electricity/electricity-in-the-us-generation-capacity-and-sales.php (visited on 03/11/2024).

[6] Michelle Davis and Shawn Rumery. *Wood Mackenzie/SEIA US Solar Market Insight®.* URL: https://www.woodmac.com/industry/power-and-renewables/us-solar-market-insight/ (visited on 03/11/2024).

[7] *Wind, solar, and batteries increasingly account for more new U.S. power capacity additions.* U.S. Energy Information Administration (EIA). Mar. 6, 2023. URL: https://www.eia.gov/todayinenergy/detail.php?id=55719 (visited on 03/11/2024).

[8] Alex Mey. *Today In Energy - Average U.S. construction costs drop for solar, rise for wind and natural gas generators*. U.S. Energy Information Administration (EIA). Nov. 3, 2022. URL: https://www.eia.gov/todayinenergy/detail.php?id=54519 (visited on 04/03/2024).

[9] Catherine Lane and Gianna Cappuccio. *Coronavirus threatens health of the global solar industry*. Solar Reviews. Oct. 22, 2020. URL: https://www.solarreviews.com/content/blog/coronavirus-impacts-solar (visited on 03/11/2024).

[10] *How COVID-19 Disrupted the Renewable Energy Transition - JHU*. MA in Sustainable Energy. Dec. 12, 2022. URL: https://energy.sais.jhu.edu/articles/how-covid-19-disrupted-renewable-energy-transition/ (visited on 03/11/2024).

[11] Nicolás Rivero. *Here's how supply chain issues are affecting renewable energy projects*. World Economic Forum. Nov. 4, 2021. URL: https://www.weforum.org/agenda/2021/11/supply-chain-problems-solar-power-renewable-energy/ (visited on 03/11/2024).

[12] *Antidumping and Countervailing Duty Orders on Crystalline Silicon Photovoltaic Cells, Whether or Not Assembled Into Modules, From the People's Republic of China: Final Scope Determination and Final Affirmative Determinations of Circumvention With Respect to Cambodia, Malaysia, Thailand, and Vietnam*. Federal Register - International Trade Administration. Aug. 23, 2023. URL: https://www.federalregister.gov/documents/2023/08/23/2023-18161/antidumping-and-countervailing-duty-orders-on-crystalline-silicon-photovoltaic-cells-whether-or-not (visited on 03/04/2024).

[13] Sylvia Leyva Martinez. *US solar: Biden's Executive Order brings (some) relief*. June 28, 2022. URL: https://www.woodmac.com/news/opinion/us-solar-bidens-executive-order-brings-some-relief/ (visited on 03/13/2024).

[14] *Uyghur Forced Labor Prevention Act*. UFLPA | Homeland Security. URL: https://www.dhs.gov/uflpa (visited on 03/11/2024).

[15] Nichola Groom and Richard Valdmanis. "Exclusive: U.S. solar panel imports from China grow, alleviating gridlock, officials say". In: *Reuters* (Mar. 7, 2023). URL: https://www.reuters.com/business/energy/us-solar-panel-imports-china-grow-alleviating-gridlock-officials-say-2023-03-06/ (visited on 03/11/2024).

[16] Michelle Davis. *Supply chain relief slowly arrives for US solar | Wood Mackenzie*. Wood Mackenzie. June 8, 2023. URL: https://www.woodmac.com/news/opinion/supply-chain-relief-us-solar/ (visited on 03/11/2024).

[17] Michelle Davis. *The Inflation Reduction Act and its impact so far*. Wood Mackenzie. Oct. 2, 2023. URL: https://www.woodmac.com/news/opinion/the-inflation-reduction-act-and-its-impact-so-far/ (visited on 03/11/2024).

[18] *Green Power Markets - Summary of Inflation Reduction Act provisions related to renewable energy*. United States Environmental Protection Agency (EPA). Oct. 25, 2023. URL: https://www.epa.gov/green-power-markets/summary-inflation-reduction-act-provisions-related-renewable-energy (visited on 03/11/2024).

[19] *Wood Mackenzie | The Inflation Reduction Act*. 2023. (Visited on 04/03/2024).

[20] *December 2023 Investor Presentation*. Nov. 28, 2023. URL: https://www.investor.nexteraenergy.com/~/media/Files/N/NEE-IR/news-and-events/events-and-presentations/2023/11-28-23/December%202023%20Presentation%20vF.pdf (visited on 04/03/2024).

[21] Matsusada Precision Inc. *Difference DC power and AC power| Tech*. Matsusada Precision. Aug. 12, 2021. URL: https://www.matsusada.com/column/dc_and_ac.html (visited on 04/03/2024).

[22] Dave. *How Is the Size of a Solar Farm Defined?* SolarLandLease. Dec. 13, 2017. URL: https://www.solarlandlease.com/size-of-a-solar-farm (visited on 04/03/2024).

[23] John Forrest et al. *coin-or/Cbc: Release releases/2.10.11*. Version releases/2.10.11. Oct. 25, 2023. DOI: 10.5281/zenodo.10041724. URL: https://zenodo.org/records/10041724 (visited on 02/11/2024).

[24] *Gurobi Optimizer*. Gurobi Optimization. URL: https://www.gurobi.com/solutions/gurobi-optimizer/ (visited on 02/11/2024).

[25] Louis Boguchwal. *Create and Deploy your own Optimization API*. Medium. Mar. 9, 2021. URL: https://louis-boguchwal.medium.com/create-and-deploy-your-own-optimization-api-7b4275f159c9 (visited on 04/05/2024).

[26] *Abstract Versus Concrete Models — Pyomo 6.7.0 documentation*. Pyomo | Sandia National Laboratories. URL: https://pyomo.readthedocs.io/en/stable/pyomo_overview/abstract_concrete.html (visited on 02/11/2024).

[27] Richard J. Forrester and Lucas A. Waddell. “Modeling Using Logical Constraints”. In: *Encyclopedia of Optimization*. Ed. by Panos M. Pardalos and Oleg A. Prokopyev. Cham: Springer International Publishing, 2020, pp. 1–5. ISBN: 978-3-030-54621-2. DOI: 10.1007/978-3-030-54621-2_834-1. URL: https://doi.org/10.1007/978-3-030-54621-2_834-1 (visited on 02/11/2024).

[28] Jeffrey D. Camm, Amitabh S. Raturi, and Shigeru Tsubakitani. “Cutting Big M down to Size”. In: *Interfaces* 20.5 (1990). Publisher: INFORMS, pp. 61–66. ISSN: 0092-2102. URL: https://www.jstor.org/stable/25061401 (visited on 02/11/2024).

[29] Phil Taylor-Parker. *Solar Racking: Best Solar Panel Mounts in 2021*. Unbound Solar. Running Time: 1325. Jan. 1, 2021. URL: https://unboundsolar.com/blog/best-solar-panel-mounts (visited on 03/01/2024).

[30] Aurora Staff. *Understanding PV System Losses, Part 1: Nameplate, Mismatch, and LID Losses*. Aurora Solar. June 16, 2021. URL: https://aurorasolar.com/blog/understanding-pv-system-losses-part-1/ (visited on 02/17/2024).

[31] Adam Baker. *What Solar Bin Class Means for Site Monitoring*. Affinity Energy. Jan. 9, 2018. URL: https://affinityenergy.com/solar-bin-class-means-site-monitoring/ (visited on 02/17/2024).