2023

International Mathematical Modeling Challenge (IM^{2}C)

Summary Sheet

Although it may seem vast, land is still a finite resource, similar to food and water. Therefore, we must continue to acknowledge that it must be managed carefully. However, this task of optimizing land utilization is further complicated by the fact that each community, in conjunction with business owners, has unique interests, needs, and preferences. It is essential to involve local stakeholders, including the community and businesses, in the decision-making processes. This problem is applied to a piece of land in New York, with a mathematical model.

Task 1: Development of Best Land Use Decision Model provides a computational, benefit-to-cost optimization model that uses real-world data to calculate a score on which among eight provided facilities is the most optimal to establish within that land. The weight is based on the monetary benefits and costs that a facility or business incurs within the short term (construction) and the long term (maintenance and operation). Afterward, a greedy, stochastic algorithm designates the best way to divide up the land, should different kinds of facilities be built on the land. It was implemented through a graph theory clustering algorithm, a deterministic way of combining isolated regions, and taking into account a decrease in productivity when the size of a facility continues to increase.

Task 2: Testing the Decision Model applies the model in the previous section for several scenarios, to determine the best facility to construct. There were two tests ran: first a comparison between all of the facilities if they were the only ones built on the land, and a comparison between three pairs of facilities each pertaining to different business interests. The tests conclude that a regenerative farm followed by a cross-country skiing farm provided the best use of the given land. Moreover, for two competitive pieces of land, the model, through the clustering algorithm, was able to provide an optimal allocation that matches the expected distributions of the facilities.

Task 3: Impacts of Population Influx determines the impacts of an influx of population on a rural business, assuming that everyone is suited to the demand. A standard microeconomic supply and demand model was utilized in order to determine the relationship of the revenue R of a facility to the average income I of its residents, which concluded $R \propto I^2$. As such, the model concludes that the scores of building a singular facility increased an average of $\approx 0.1\%$ across several facilities. Moreover, for competing, complementary facilities within the same businesses, the business owners are in their interests to change the layout of the land to maximize revenue.

Task 4: Discussion on Model Generalizability finally determines how generalizable our model is when it comes to other applications. In our analysis, we concluded that our model is indeed easily generalizable to other locations by simply modifying the scoring function based on construction costs, taxes, business policies, and economic supply and demand.

Letter to the Decision-Makers

Dear community leaders and business planners:

Thank you for your trust in our research team in your request for assistance in optimizing the use of land in Syracuse, New York, United States of America.

Our team was able to develop a model that takes into account the economic benefits and costs of installing facilities on the property. In essence, the model can help decide which facility from your options can best maximize the earnings while minimizing the costs, such as through construction and environmental taxes.

If you have plans to accommodate just one facility throughout the parcel of land, we highly suggest that you construct a regenerative farm as it has the highest facility score according to our model. We attribute this to its minimal construction costs while having adequately high revenue. Another strong choice is constructing a cross-country skiing facility, should you opt for a sports-related business.

In the event that you will be considering building more than one facility, our team highly proposes that a combination of the outdoor sports complex and cross-country skiing facilities be built on the said parcel. Through our model, we discovered that such a combination will provide the largest benefit for both the business planners and the community. The combination of facilities will deliver the highest revenue to the business while minimizing incurred costs. In addition, both facilities provide a larger variety of facilities for exercise for the nearby community. Furthermore, attached to this letter is a report containing layout maps generated by our model of the aforementioned facility configurations, which divide the property optimally into these two facilities.

However, if other facilities will be considered to be built on this parcel, our team is proud to say that our model is ready to accommodate these additional facilities you may have in mind. Our model can also be used in nearly any other location throughout the globe. Rest assured, that the well-being of the community and, most importantly, the environment was included in the development of this model.

Attached to this letter is the entire technical report of our model. We hope that you will find our recommendations useful in your decision-making process. If you have any other concerns or inquiries, feel free to contact us, and we will happily provide additional details. We hope to work with you again in the near future.

Sincerely,
IMMC2023055 Modeling Team

Contents

1 Overview

Although it may seem vast, land is still a finite resource, similar to food and water. With the rapid urbanization of today's interconnected world, the problem of optimizing land use becomes increasingly important to solve. This challenge involves balancing several factors, such as owner profitability, environmental impact, and social appropriateness. In this paper, we present a scalable stochastic computational approach for deciding what is the "best" way of using a given land area. Specifically, we explore its application to a rural, 3-square-kilometer area located in New York.

2 Task 1: Development of Best Land Use Decision Model

2.1 Assumptions

Our model made several important assumptions in order to simplify the quantitative metric that describes the benefits of establishing several facilities within the given area. For brevity, we will be referring to the given piece of land located in New York as a **"property"** for the rest of this paper.

1. **The developers of the property are business-oriented.** In a microeconomic approach, we assume that the owners wish to maximize their profits. In a rural setting such as that of the property, this assumption becomes more credible since there are multiple business incentives for constructing new facilities in such areas. For example, rural areas tend to have more loyal customer bases due to their close-knit community ties. Additionally, the proximity of the rural property to an urban population provides another demand source. These two factors give the property more business potential that could be maximized by the developers.
2. **The developers have the incentive to be environmentally sustainable.** In New York, there are plans to implement Local Law 97[14], which aims to increase the tax on companies that produce large amounts of carbon emissions. Therefore, business owners across a large area of land, including the given property, would be incentivized to make their future facility operations environmentally sustainable, such as through the addition of solar panels as energy sources.
3. **Land use costs and benefits vary with the geography in a discretized manner.** We recognize that the property is not uniform. That is, certain regions in the property, due to unique topography (e.g., lack of trees, higher elevation) may have more "benefits" through the construction of certain kinds of facilities over others. While we would account for these geographic variations in our model, we will only be doing so in a discrete manner since acquiring accurate, continuous data for the given property's terrain is difficult due to top-view maps being the only resource available.
4. **Visitors are more inclined to visit a facility if it has cell coverage in the area.** Good network access is a key attractor for visitor-based facilities, such as the agritourist center. The availability of a cell network in the area allows for better communication and social media, which can encourage more people to visit.
5. **Revenue for products/services are independent of time unless population changes.** For facilities that produce some good/service that may be bought by consumers, we will not be

accounting for any fluctuations in the price and quantity sold since there is no real-world data that may be used for it. Instead, we will be using average revenue as an alternative measure since this is more feasible to collect from reliable sources.

6. **We can associate a monetary value to social and ecological welfare.** Profitability is not the only factor that is vital for land use decision-making. Social and environmental welfare also need to be accounted for. To account for this somewhat qualitative element in a quantitative manner, we opted to use a monetary value. This is because existing policies accounting for these factors, such as carbon taxation, already exist.

Based on these assumptions, our decision model should therefore consider both the **benefits, such as owner profitability and environmental impact,** and **costs, such as operating and maintenance expenses**. These are both best measured through **economic activity (i.e., monetary values)**. Furthermore, our decision model should also account for possible variations in the property's terrain and geography since this plays a real-world influence on the installation of new facilities to a land area.

2.2 Development of Decision Model

Following these standards and assumptions, we decided to develop a **stochastic, computational, cost-benefit optimization model** that searches not only the **highest-possible "score" (i.e., a quantifiable metric)** given certain options of facilities but also **approximate locations on where these facilities should be built in the property** in order to achieve the optimal score result. Through this model, we are able to account for both economic activity (through the score info) and geographic variations (through the approximate location info).

2.2.1 Model Overview

Formally, we will be denoting our decision model as $\mathcal{M}(\mathcal{G}, \mathcal{O})$, where $\mathcal{G}$ represents the property's geographic layout and $\mathcal{O}$ represents the set of facilities (i.e., options) that we wish to be installed in the property. This model outputs two results: $\mathcal{W}$, the average score over the entire property, and $\mathcal{G}_{\text{alloc}}$, a colored version of $\mathcal{G}$ wherein different colors represent distinct facilities to be built in certain regions of the property. Our decision model has three main features that we wish to highlight:

1. **Discretised Property.** In our decision model, property $\mathcal{G}$ is subdivided into discrete square units called "regions" with dimensions $u \times u$. Each region will contain certain geographic information (i.e., tree presence, terrain elevation, cell coverage). More details on the implementation of this information extraction can be found in the appendix.
2. **Option Scoring.** For each region, the decision model determines what is the best facility to be built in that region using our scoring function. Formally, this function calculates for each $r \in \mathcal{G}$ the array $A_r := [S(f) | f \in \mathcal{O}]$ and assigns whichever facility corresponds to $\max(A_r)$ as the "best facility" for region r. The details of the scoring function are found in Section 2.3.
3. **Location Propagation.** In the scenario that we wish to calculate the score of installing more than one facility in the property (i.e., $|\mathcal{O}| > 1$), it is possible that the scoring method will return a "best facility" map that is *messy* (see Figure 1 for an example). That is, regions with the same "best facility" are not adjacent to one another. This is a rather unrealistic facility installation layout. To address this, we utilized a "propagation algorithm", $\mathcal{P}$, that will adjust

the layout to section together same-facility regions while still optimizing for the best score. The details of this algorithm are found in Section 2.4

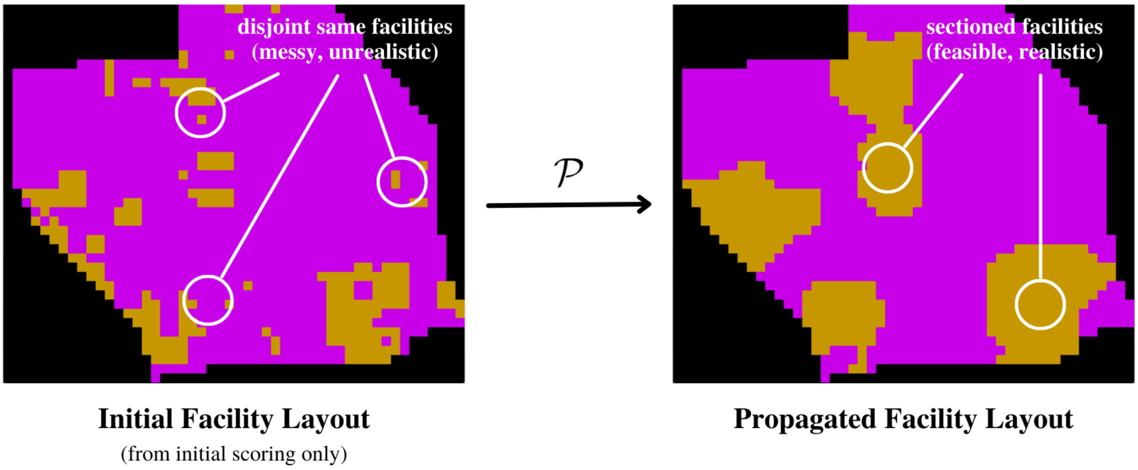

Figure 1: Sample visualization of propagation algorithm for two facilities to be built in the property (colored orange and purple). Disjoint same-facility regions (i.e., the unit squares) from the initial assignment are sectioned together after the algorithm is run.

2.2.2 General Variables

For easier reference, the commonly-used decision model variables that will be used in later parts of this section are presented in Table 1.

Variable	Definition
r	A region, or square sub-unit, of the property.
f	A facility to be installed in the property.
t	A time period classification which can denote either a short-term value (s) or a long-term value (ℓ)
P_i	Total price of good/service i [in \$]
p_i	Unit price of good/service i [in \$/unit]
d_i	Quantity demanded for a certain component i
$B_t(r, f)$	Total economic benefits of constructing facility f in region r over a time period t [in \$]
$C_t(r, f)$	Total economic costs of constructing facility f in region r over a time period t [in \$]
$S(r, f)$	Score (i.e., a quantifiable metric) for installing a facility f in a certain property region r. If we do not specify which facility and region to calculate for, the general notation for the score is simply S.

Table 1: General variables in the model

2.3 Scoring Function: Defining "Best" Facility

Our decision model differentiates between facilities that could be installed through the scoring function $S(r, f)$, which is a **time-weighted, benefit-to-cost ratio formula**. We chose this approach because our assumptions about the land developers' business-oriented mindset can be easily related to economic activity. This framework provides a **monetary standard** for quantifying how "worth it" it is to build some facility f in some given region r of the property.

We account for the relative influence of both the short-term and long-term periods on the benefits and costs by weighing each term by their corresponding time duration. By doing this, we arrive at the following scoring function definition:

$$S(r,f) = \frac{B_s(r,f)T_s + B_\ell(r,f)T_\ell}{C_s(r,f)T_s + C_\ell(r,f)T_\ell} \quad \text{(Eq. 2.1)}$$

where $B_s(r,f)$ and $C_s(r,f)$ refer to the **yearly short-term (construction phase) benefits and costs**, respectively, $B_\ell(r,f)$ and $C_\ell(r,f)$ refer to the **yearly long-term (facility operation phase) benefits and costs**, respectively, and T_s and T_ℓ refer to the **duration in years of the short-term and long-term periods**, respectively. From this formula, it can be easily observed that when $T_\ell \gg T_s$, the benefit-to-cost ratio of the long-term production governs. Additionally, notice that since all of the terms in the formula will have the same units of $\$ \cdot \text{year}$, the score function $S(r,f)$ **returns a dimensionless value**.

Through our research and analysis, we realized that economic benefits should include possible revenues for the company and the welfare that you provide for the local community (e.g., increased employment and environmental care) since both community leaders and business owners should benefit from the affairs of the facility installed. In contrast, economic costs would include construction costs, production costs, maintenance costs, and possible environmental/welfare costs. That way, if businesses do not operate to be more environmentally friendly, it would equate to a “cost” on the welfare of society.

In line with this, we will further define the economic benefits and the costs over the short term and the long term.

2.3.1 Short-Term Benefits

During the short term, the primary benefit comes in the form of increased employment for the local community, generally in the form of temporary jobs for constructing the facility. Formally, we model this as:

$$B_s(r,f) = p_{\text{wage}}(f) \cdot d_{\text{worker}}(r) \quad \text{(Eq. 2.2)}$$

where $p_{\text{wages}}(f)$ is the yearly wage [in \$] of a worker's construction labor of facility f, and $d_{\text{worker}}(r)$ is the number of construction workers needed for one region r. From our analysis, we realized that this is the most feasible way of quantifying the social benefit of the employment increase.

2.3.2 Short-Term Costs

Since there will be a portion of a new facility installed in any given parcel region, the most reasonable model for the economic costs in the short term is the construction-related expenses. Formally, we modeled this as:

$$C_s(r,f) = P_{\text{acc}}(r) + P_{\text{irr}}(r) + P_{\text{labor}}(r,f) + P_{\text{tree}}(r,f) + P_{\text{cons}}(r,f) \quad \text{(Eq. 2.3)}$$

where $P_{\text{acc}}(r)$ is the cost of accessibility components (e.g., pavements and roads), P_{irr} is the cost of supplying water/irrigation, P_{labor} is the cost of construction labor (equivalent to $B_s(r,f)$), $P_{\text{tree}}(r,f)$ is the cost associated with deforestation and $P_{\text{cons}}(r,f)$ is the facility-proper construction costs. We explain each of those costs below.

- **Accessibility Price ($P_{\mathbf{acc}}(r)$).** The accessibility price of a region r is modeled as a price for the construction of transportation toward that region. Therefore, we decided to value this as:

$$P_{\text{acc}}(r) = p_{\text{acc}} \cdot d_{\text{acc}}(r) \tag{Eq. 2.4}$$

where $p_{\text{acc}}(r)$ is a per-unit price of accessibility [in \$/meter], which includes the cost for constructing roads/pavements, and $d_{\text{acc}}(r)$ is the distance [in meters] to one of the nearest roads bounding the entire property. It should be noted that this is just an approximation to account for such accessibility costs. The actual layout of the roads will not be in the scope of this model.

- **Irrigation Price ($P_{\mathbf{irr}}(r)$).** The irrigation price is defined similarly:

$$P_{\text{irr}}(r) = p_{\text{irr}} \cdot d_{\text{irr}}(r) \tag{Eq. 2.5}$$

where p_{irr} is a per-unit price [in \$/meter] of irrigating a given region, and $d_{\text{irr}}(r)$ is the distance [in meters] from region r to the nearest source of water in the property.

- **Labor Price ($P_{\mathbf{labor}}(r,f)$).** The labor price is defined in the same way as the short-term benefit $B_s(r,f)$, which is the annual cost of labor for constructing a portion of facility f in the region r. This information is easily accessible from the salary data online of people from New York.

- **Deforestation Price ($P_{\mathbf{tree}}(r,f)$).** To account for the environmental welfare of constructing a portion of some facility f in the region r, we define the price associated as:

$$P_{\text{tree}}(r,f) = p_{\text{tree}}(f) \cdot d_{\text{tree}}(r) \tag{Eq. 2.6}$$

where $p_{\text{tree}}(f)$ refer to the deforestation price constant for facility f (which varies since certain facilities would require more trees to be removed compared to others), and $d_{\text{tree}}(r)$ refers to the tree cover percentage in a region r.

- **Construction Price ($P_{\mathbf{cons}}(r,f)$).** This is equivalent to the cost of construction of facility f in the region r, including terraforming expenses due to the elevation of the land (Assumption 5). Therefore, the construction price is defined as follows:

$$P_{\text{cons}}(r,f) = \left(1 - \left|1 - k_f \frac{E_r}{E_{\text{ave}}}\right|\right) \cdot p_{\text{con}} \cdot A \tag{Eq. 2.7}$$

where E_r is the mean elevation of the region r, E_{ave} is the mean elevation of the entire property, p_{con} is the unit price of construction [in \$/m^2], and A is the area of region r [in m^2]. However, some facilities might have no significant changes in cost due to elevation (e.g., farmland), compared to others (e.g., sports complexes). Hence, the multiplier k_f describes the susceptibility of construction costs of facility f to elevation deviations.

2.3.3 Long-Term Benefits

Similar to Section 2.3.1, there are several components that need to be accounted for in long-term benefits. We modeled these long-term economic benefits as:

$$B_\ell(r,f) = R(r,f) + W_{\text{jobs}}(f) + T_{\text{solar}} \tag{Eq. 2.8}$$

where each of the terms is defined as follows:

- **Revenue ($R(r,f)$).** This refers to the average amount of money that a facility f makes throughout an operational year. This can be expressed as follows

$$R(r,f) = (1+c) \cdot R_{raw} \tag{Eq. 2.9}$$

  where R_{raw} is the average revenue of the facility per region [in \$]. However, according to Assumption 4 in Section 2.1, people are more inclined to go to a facility when it is cell service accessible; hence, we set a multiplier $c = 0$ when there is no cell service in that region, and $c \propto r_{\text{cell coverage}}$ otherwise.

- **Wages ($W_{\text{jobs}}(f)$).** Similarly to Section 2.3.1, we categorize the value of the labor that is created in that facility's operation as a benefit. We modeled this as:

$$W_{\text{jobs}}(f) = w_f \cdot N_{\text{workers}}(f) \tag{Eq. 2.10}$$

  where w_f is the mean annual salary [in \$/worker] for that facility, and $N_{f,\text{workers}}$ is the number of workers in a region.

- **Clean Energy Benefits (T_{solar}).** We consider the generation of renewable energy, such as solar power, as an environmental welfare benefit. Formally, we modeled this as the potential reduction in carbon tax experienced by the facility:

$$T_{\text{solar}} = p_{\text{solar}} \left(\frac{2}{3}\sigma_s^{5/2} - \frac{1}{3} \right) \tag{Eq. 2.11}$$

  where p_{solar} is a default price that is assigned to the reduction of solar prices, and $0 \leq \sigma_s \leq 1$ is the fraction of land that is covered in solar panels. The denominator just normalizes the values between $-1/3$ and $1/3$, such that when the fraction of solar panels is higher, the value of the benefit also increases.

2.3.4 Long-Term Costs

Finally, we define the long-term costs as expenses incurred throughout the operation of the business. We modeled this through the following equation:

$$C_\ell(r,f) = P_{\text{operating}}(f) + P_{\text{utility}}(f) + P_{\text{tax}}(r,f) \tag{Eq. 2.12}$$

where each of the terms is defined as follows:

- **Operating Cost ($P_{\text{operating}}(f)$).** The operating cost is the average annual cost of maintenance and supplies that are required to sustain the production of facility f per region. Our value for this variable is based on existing data.

- **Utility Cost ($P_{\text{operating}}(f)$).** Meanwhile, the utility cost refers to the average annual cost of water and electricity used by the facility f per region. Our value for this variable is based on existing data.
- **Carbon Taxation ($P_{\text{tax}}(f)$).** To penalize a lack of environmental sustainability, a cost for carbon emissions will be implemented, which is defined as

$$P_{\text{tax}}(r, f) = A_r p_{\text{tax}} \max\{m_{\text{prod}} - m_{\text{lim}}, 0\} \tag{Eq. 2.13}$$

where A is the area of the region r, p_{tax} is the unit tax [in \$/kg$\cdot m^2$], m_{prod} is the mass of the carbon dioxide produced, and m_{lim} is the mass of the carbon emissions limit in New York.

2.4 Propagation Algorithm

To properly allocate the given parcel when given two or more facilities, a greedy, stochastic algorithm was developed. Given a two-dimensional array r of regions $r_{x,y}$, there are two possible approaches to determining its initial configuration such that each region is assigned a facility. One could either assign each $r_{x,y}$ a facility in a deterministic manner, by choosing the facility with the maximum score in that area, or otherwise. We chose to assign facilities nondeterministic so that the algorithm is not restricted to the initial maxima produced by the map—which could prove not to be the optimal configuration. The full code can be found in Appendix A2.

2.4.1 Facility Clustering

Each r_{x_i,y_i} is connected bidirectionally to r_{x_j,y_j} if and only if the two are adjacent to each other vertically or horizontally. We define a group being a set g wherein all regions within it could be traversed from one to another and are of the same facility. Two regions are disjoint if there is no shared $r_{x,y}$.

Figure 2: Sample map r wherein two groups are disjoint.

The propagation algorithm is the driving force for the clustering of previously disjoint regions. Given a node $r_{x,y}$ and its vertically, horizontally, or diagonally adjacent neighbors, there are two ways that any $r_{x,y}$ could be transformed into a different facility such that two previously disjoint regions are connected:

- if a region of facility encloses the majority of an $r_{x,y}$ of a different facility, it transforms into the said facility;
- if the score of any $r_{x,y}$, taking into consideration its region, is the minima among its surrounding regions, it transforms into the facility of the region with the local maxima.

2.4.2 Isolated Regions and Directional Spread

The algorithm above is insufficient wherein given a gap of three or more nodes between two disjoint regions of the same facility, there is no possibility of those regions being connected. Regions that are in such a state are referred to as isolated; to encourage the search for what could be a better configuration of facilities, a way of closing larger gaps was considered. This algorithm, which is integrated into the propagation system, is a directional spread.

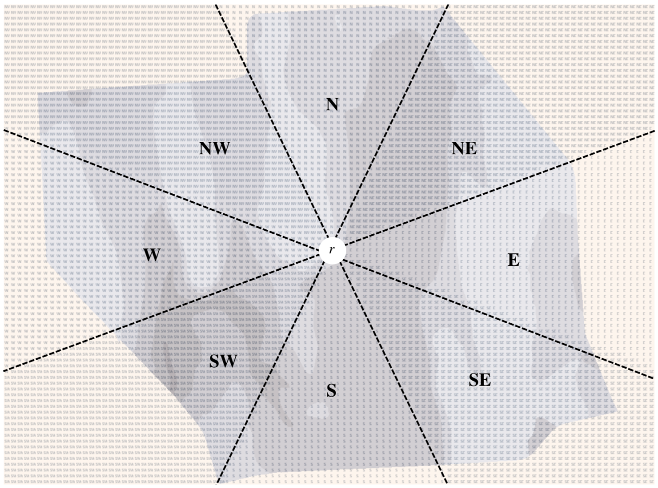

Figure 3: The map is separated into 8 distinction regions (N, S, E, W, NS, NE, SW, SE) with the center point marked as r.

The map was divided into eight separate regions as shown above, For each facility, tally where the majority of its regions are with respect to the center of r. The aforementioned portion will be translated with respect to the immediate vertical, horizontal, or diagonal neighbors of any $r_{x,y}$—let this be its preferred direction of propagation v. With the initial propagation algorithm in mind, the "attacking" region could only transform an $r_{x,y}$ if and only if x, y is in its v.

2.4.3 Law of Diminishing Marginal Returns

The Law of Diminishing Marginal Returns (LDMR) states that with increasing capacity, in this case, the size of a facility, the additional output becomes reduced after a certain point. To account for this in our model, a multiplier was included in the calculation of the score of each facility cluster, which is defined in the following equation:

$$c_{\text{LDMR}} = \begin{cases} \exp\left(-\frac{7}{A_{\min}}A + \ln(M_{\text{adv}} - 1) + \frac{7}{A_{\min}}\right) & \text{, if } A < A_{\min} \\ 1 & \text{, if } A_{\min} \leq A \leq A_{\max} \\ M_{\text{pen}} + \frac{1}{A - A_{\max} + 5} & \text{, if } A > A_{\max} \end{cases} \quad \text{(Eq. 2.14)}$$

where A is the area of the facility cluster, $A_{\min}$ is the minimum area when LDMR begins, $A_{\min}$ is the maximum area when negative LDMR beings, M_{adv} is the maximum advantage multiplier, and M_{pen} is the maximum penalty multiplier.

For example, if a facility cluster has a score lower than the A_{min}, it will receive a multiplier greater than one but less than M_{adv}. Meanwhile, if a facility cluster has a score greater than A_{min}, it will receive a multiplier lower than 1 but greater than M_{pen}. Such allows smaller facility clusters to propagate while applying a microeconomics concept, the LDMR.

3 Task 2: Testing the Decision Model

3.1 Collection of Data

For each of the variables in the previous section, extensive research to arrive at a suitable value assignment is necessary. The full list of data that we have found and used is tabulated in Appendix A1. However, as expected, sufficient data for some variables are not available online, especially for variables that are unique to the model. As such, we introduce the following new assumptions to aid the data collection process:

1. **Operating labor is minimum wage.** The value for the long-term wage w_f that was used in calculating this model is \$15/hr. We assumed that employees work for 8 hours each day and 300 days a year, arriving at an assumed annual salary of \$36000. This will also be the same value used for the cross-country skiing facility since, despite the seasonal operation, employees are paid more due to the colder work conditions.
2. **Tree cover density is constant.** For areas in the property with tree coverage, we assume the tree density is consistent as accounting for these variations, while realistic, is difficult to achieve with the top-view satellite imagery we are provided with.
3. **Worker density is constant in the property.** We assumed that there is a constant 1 worker per region in order to simplify model terms regarding the number of workers in a facility. With the region resolution we will be using (resulting in $\approx$ 1000 regions), the total number of workers ($\approx$ 1000 workers) would be reasonable for the property's 3 sq. km size. Moreover, for simplification of the model, this value will not change with respect to the demand of the facility as it requires a more complicated model.

The complete list of data we have collected for testing our decision model can be found in Appendix A2, which utilizes the real-world data that is modeled in terms of area. However, in the implementation of the algorithm into a program, these values were translated into a currency per pixel value.

3.2 Overview of Sensitivity Analysis

Our model sensitivity analysis will comprise two kinds of tests: "single-option", wherein we calculate the average score if we were to build only one facility on the entire parcel, and "double-option" wherein we match certain facility options and calculate their average scores.

Facility Options

We used the listed facilities in the problem statement as the options to choose from when testing. For reference, the assigned visualization colors of each facility are presented in Figure 4.

Figure 4: List of facilities to be tested, along with their color assignments.

Single-Option Testing

Formally, for each facility f in the list of options presented in Section 3.2, we will be performing $\mathcal{M}(\mathcal{G}, f)$, or the average of $S(r, f)$ for all $r \in \mathcal{G}$. By performing a comparison of the scores produced, we will determine which facility has the highest benefit-to-cost ratio, making the facility worth it to build.

Double-Option Testing

The second section of the sensitivity analysis would involve determining how the interaction between a pair of facilities would affect their score according to our model. Formally, for two facilities f_a and f_b in the list of options presented in Section 3.2, we will be performing $\mathcal{M}(\mathcal{G}, \{f_a, f_b\})$, or the average of $S(r, f_r)$ for all $r \in \mathcal{G}$ where f_r is the facility allocated for region r.

We chose the following three pairs of facilities as the most suitable to be grouped together in the same property: **(1)** agrivoltaic farm and agritourist facility, **(2)** crop farm and grazing farm, and **(3)** outdoor sports complex and skiing facility. These three pairs are highly beneficial in determining the optimal allocation of land depending on the different business interests of the developers. For instance, the business owners and the community leaders may aim for a more progressive form of farming (for Pair 1), for more traditional farmers (for Pair 2), and/or for more urban development (for Pair 3).

3.3 Results and Discussion

Single-Option Testing

The first run of the decision model was performed for single-run testing, where each of the facility data was run through the algorithm. This section will discuss the comparative values of S_{ave}, the average score of installing one facility in the property.

Score Results. The complete results of the average scores that the decision model outputted can be seen in Table 2. This has several implications. First, considering the costs and the benefits of the creation of the facilities, the **cross-country skiing facility has the highest benefit-to-cost ratio (i.e., score)**, which implies high margins of profit, and possibly the lowest aversions to losses in the company. Therefore, if the developers were profit-oriented people, the cross-country skiing facility would be the best.

Option	Weight
Outdoor Sports Complex	1.1875
Cross-Country Skiing Facility	2.2518
Crop Farm	2.1898
Grazing Farm/Ranch	2.0263
Regenerative Farm	2.6496
Solar Array	1.7732
Agrivoltaic Farm	1.8968
Agritourist Center	1.7657

Table 2: Our decision model's average scores, S_{ave}, if each facility were to be installed in the entire property.

However, the community leaders might want to keep their interests and say that the rural area deserves a location that keeps the "ruralness" of the area. Therefore, they may come to a compromise and determine that **a regenerative farm is optimal to preserve the rural atmosphere**. Therefore, it would just be a matter of compromise between the businesses and the communities to determine which facility would be built in this region, should only one facility would be chosen.

Land Cover. Each of the facility trials resulted in a property land cover of 100%, which means that the facility is beneficial to be installed on the entire land. This has certain implications. First, since the score S is a weighted version of the benefit-to-cost ratio B/C for each facility, we know that $B > C$, which means that the benefits always outweigh the costs for all options. This just goes to show that none of the businesses that we have presented in this section will go bankrupt (assuming good management by the owners). Second, this means that installing businesses is much more beneficial than keeping the undisturbed state of the land (which has $S = 1$ for obvious reasons).

Double-Option Testing

In this section, we analyze the scores returned by our decision model for three different scenarios listed in Section 3.2. This section assumes that business owners and community leaders are willing to build two kinds of facilities. As such, we cover three different business ventures. For easier reference in our analysis, the different features of the property are presented visually in Table 5.

Pair 1: Agrivoltaic Farm and Agritourist Facility

As mentioned earlier, this business venture is much more likely to involve a progressive form of farming. Agrivoltaic farms involve the use of solar panels along with farming crops, while an agritourist facility would be much more likely to demonstrate the people's interest in seeing the agritourism side of the land. As can be seen in Table 2, the values $S(\text{Agrivoltaic Farm}) = 1.8968$ and $S(\text{Agritourist Center}) = 1.7657$ are extremely close to each other. Therefore, we shall determine what distribution will provide the lowest cost-benefit ratio among the two options. The expected result would be an approximately equal distribution. The complete diagram of the distribution can be seen in Figure 6, where the dark purple regions represent the locations of the agritourist center, and the light purple regions represent the locations of the agrivoltaic farm. The average score for this is $S(\text{Combined}) = 3.5311$.

The first major observation to notice is that the double-option score S(Combined) is larger than the value of $S(\text{Agrivoltaic Farm})$ and $S(\text{Agritourist Center})$. This can most likely be attributed to

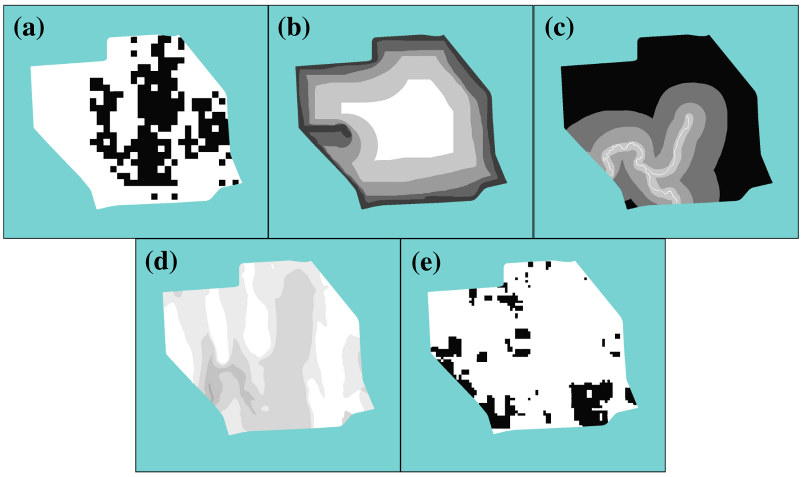

Figure 5: Textures of different features of the property: (a) Location of cell coverage (white is covered) (b) Distance from the road (darker is closer) (c) Distance from nearest water or irrigation source (darker is further) (d) Topography and elevation map (darker is higher), and (e) Tree cover (white is covered)
.

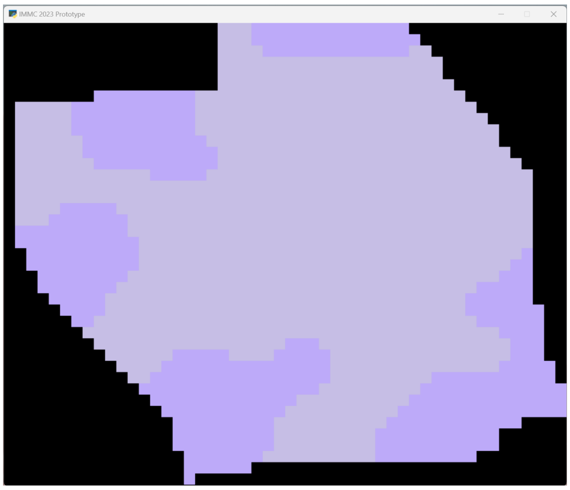

Figure 6: Distribution of the agritourist center and agrivoltaic farm across the land, with score $S(\text{Combined}) = 3.5311$. Generated by the team.

the fact that certain regions score well for one of the two facilities but have bad scores for the other, which is the reason why there are parcels of land that could be traded off between the two lands. All these are evidence that the land is **highly optimized** for the use of those two centers.

The second major observation that could be seen is in the terms of the layout. The primary locations of the agritourist center could be seen to the west side of the map, continuing towards a large area across the east and eventually ending up south. As could be seen from Figure 5, this is where cell coverage is present. Therefore, the revenue of tourist centers is highest when there is cell coverage present, which could likely be attributed to social media presence. A final observation is that the agritourist center, which likely produces the most plants, is directly connected to the source of water for the region. However, this does not mean that the agritourist centers are tourist-only; it means that plants can grow along with them. Therefore, the **allocation is highly optimal in this regard**. We verify that this allocation with the algorithm in this regard is indeed optimal.

Pair 2: Crop Farm and Grazing Farm

As mentioned earlier, this business venture is more likely to involve traditional values of farming. However, as can be observed from Table 2, these also have extremely similar values, where $S(\text{Crop Farm}) = 2.1898$ and $S(\text{Grazing Farm}) = 2.0263$. Therefore, we shall determine what distribution will provide the lowest cost-benefit ratio among the two options. The expected result would be an approximately equal distribution. The complete diagram of the distribution can be seen in Figure 7, where the red regions represent the locations of the crop farm, and the yellow regions represent the locations of the grazing farm. The average score for this is just $S(\text{Combined}) = 3.5033$.

Figure 7: Distribution of the crop farm and grazing farm across the land, with score $S(\text{Combined}) = 3.5033$. Generated by the team.

Similar to the analysis of the results earlier, we could easily observe that $S(\text{Combined}) > S(\text{Crop Farm})$ and $S(\text{Combined}) > S(\text{Grazing Farm})$. Hence, the land is highly optimized for use according to our decision model.

However, similarly to the use, the crop farm and grazing farm do not require cell coverage or colossal changes in the elevation in order to function properly. However, we can observe that the grazing farms are located in sections where there are small changes in the elevation, which is optimal for creating structures. Moreover, the crop farm is located in the regions where it is closest to the natural water source, which reduces costs for irrigation. Hence, both quantitatively and qualitatively, the distribution of land in this simulation is **highly optimal**.

Pair 3: Sports Complex and Skiing Facility

As mentioned earlier, this business venture is more likely to involve traditional values of farming. However, as can be observed from Table 2, these also have extremely different values, where $S(\text{Sports Complex}) = 1.1875$ and $S(\text{Skiing Facility}) = 2.2518$. Therefore, we shall determine what distribution will provide the lowest cost-benefit ratio among the two options. The expected result would be an inclination towards the skiing facility. The complete diagram of the distribution can be seen in Figure 8, where the purple-pink regions represent the locations of the cross-country skiing facility, and the cyan regions represent the locations of the sports complex. The average score for this is $S(\text{Combined}) = 4.1784$.

The trend of the outdoor sports complex is quite similar, with the regions of the outdoor sports complex being nearest to the road. These regions could also be observed to be extremely close to the regions with good cell coverage. However, the major trend of the cross-country skiing facility is that

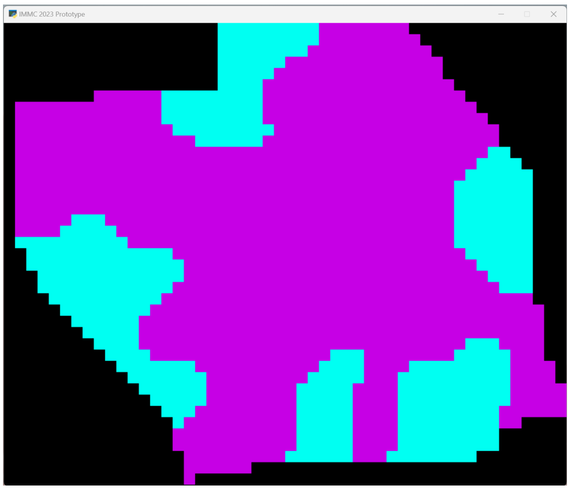

Figure 8: Distribution of the outdoor sports complex and cross-country skiing facility across the land, with score $S(\text{Combined}) = 4.1784$. Generated by the team.

it is much more present in the regions that have higher changes in elevation, meaning the region in the southwest of the map (as seen in Figure 5. This implies that the costs are minimized when the changes in elevation are quite high since the skiing facilities definitely do not mind when there are high changes in elevation when they are creating skiing facilities. Hence, both quantitatively and qualitatively, the distribution of land in this area is **highly optimal**.

In summary, the results are as follows in Table 3.

Facility 1	S(Facility 1)	Facility 2	S(Facility 2)	S(Combined)
Agrivoltaic Farm	1.8968	Agritourist Center	1.7657	3.5311
Crop Farm	2.1898	Grazing Farm	2.0263	3.5033
Sports Complex	1.1875	Skiing Facility	2.2518	4.1784

Table 3: Values of $S(f)$ for each facility pairs

3.4 Strengths and Weaknesses

Strengths. First, our computational decision model is **highly customizable**. The scoring function we created can be easily adjusted to account for other factors involving economic benefit and economic cost, which will be beneficial for business sectors that have much more complicated avenues for revenues, taxes, and such. Second, our model is **generalized**. It is designed to accept any kind of land parcel and any set of facility options. This makes it possible to apply to a variety of land use scenarios, as we will support in Section 5. Moreover, it would also apply to different scales of businesses, such as large-scale businesses. One instance where this might apply is to the carbon taxation model, which can be seen in Eq. 2.13. While $m_{\text{prod}} \ll m_{\text{lim}}$ for small businesses, this tax would be much more beneficial for large-scale companies where $m_{\text{prod}} \gg m_{\text{lim}}$. Third, our computational algorithm is **relatively efficient**. It runs in roughly $\mathcal{O}((mn/u^2)^2)$, where m and n denote the dimensions of the bounding rectangle of the given property and u denote the dimensions of the discrete regions the property is subdivided into.

Weaknesses. Ultimately, our model is **dependent on real-world data** for it to be accurate. In fact, data gathering was a major challenge that we encountered during the testing of the model. However, this weakness applies to other kinds of land use models since the goal is to provide metrics for a real-world scenario, which requires real data to support it. However, there are many methods that could counteract these weaknesses. For instance, the revenue models do not come from the same region, as such the prices may be different. There are also economic models that change the revenue of a company based on its place of operation, but this is beyond the scope of our paper.

4 Task 3: Impacts of Population Influx

The construction of the nearby facility of Micron Technologies will affect the dynamics of the community within the vicinity of the property. Specifically, there will be an **influx of relatively high-income individuals which may increase revenue** for certain kinds of facilities, most notably tourism centers (e.g., agritourist centers).

4.1 Model of Change in Revenue and Population

The model of the change in revenue is dependent on a hypothetical supply and demand curve among the facilities that are of interest in the property. That is to say that given a supply curve $\mathcal{S}(P)$ in terms of the price P and a demand curve $\mathcal{D}(P)$ in terms of the price P, there would be a price P where $\mathcal{S}(P) = \mathcal{D}(P)$, which is the equilibrium price (i.e., assumed selling price). In this revenue-change model, we assume the following.

1. The function $\mathcal{S}(P)$ increases linearly with P and $\mathcal{S}(0) = 0$. In essence, $\mathcal{S}(P) = \alpha P$ where $\alpha > 0$.
2. If the intercepts of the demand curve $\mathcal{D}(P)$ are P_0 and Q_0 as shown in the supply and demand curve figure in Figure 9, then the intercepts of the demand curve after a shift in demand are βP_0 and βQ_0 where $\beta > 0$.
3. The amount of people interested in buying a good or availing a service inside a facility is proportional to their annual income.

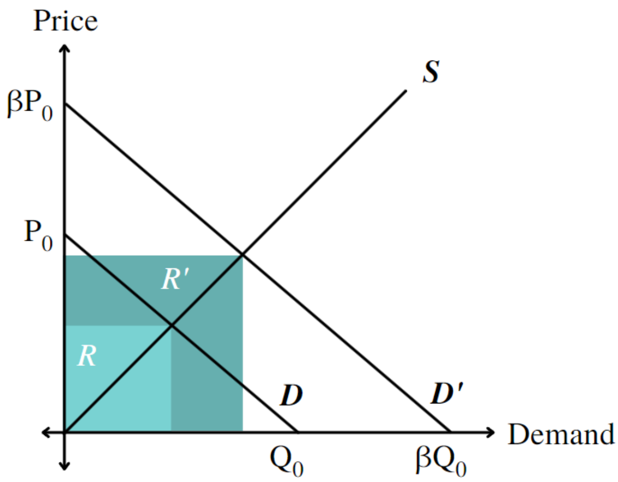

Figure 9: Supply and demand curve for the model illustration. The value $R = P_{eq}Q_{eq}$ represents revenue, P_0 and Q_0 denote intercept prices and quantities.

These assumptions provide crucial ease of creating the model since the major factor that would change is the revenue of the model R after a change in demand (or a change in population). From Assumption 3, the model for the revenue of a facility can then be expressed as:

$$R \propto P^2 \propto I^2 \qquad \text{(Eq. 4.1)}$$

where P is the population and I is the annual income. The quadratic relationship between R and P is easy. Since the supply is directly proportional to the price, a factor k along the equilibrium price would result in a factor k shift along the equilibrium quantity as well.

We consider online available data on the location of Cayuga County, where the land is located. We see that the population of the County is $P_1 = \$75880$ and the annual income is $Q_1 = \$59602/\text{yr}$, assuming all of the population of the nearby facilitates meets the demand of the revenue. Considering other available data that was provided to the problem, the additional population that will be provided to the county is $P_2 = \$9000$ and their annual income is $Q_2 = \$100000/\text{yr}$. In the initial scenario, the mean annual income is $I_1 = \$59602/\text{yr}$ while the mean annual income after the influx of the population would be then equivalent to $I_2 = \$63885/\text{yr}$. Hence, we shift the revenue values used in Task 2: Testing the Decision Model by a factor of:

$$\left(\frac{I_2}{I_1}\right)^2 = \left(\frac{\$63885/\text{yr}}{\$59602/\text{yr}}\right)^2 \approx 1.14888.$$

4.2 Overview of Model Changes

We run a similar test again on each of the components, except that we re-calculate the weight $S(f)$ for each of the businesses. Another scenario that would be re-ran is the agrivoltaic farm and the agritourist facility double-option testing, in order to have a benchmark on how a population influx affects business decisions when it comes to comparing their options for two different businesses.

4.3 Results and Discussion

Single-Option Testing

The effect of the change for single-facility average scores can be found in Table 4.

Option	$S(f)$ before influx	$S(f)$ after influx	Percent change
Outdoor Sports Complex	1.1875	1.1876	0.0112%
Cross-Country Skiing Facility	2.2518	2.2555	0.1652%
Crop Farm	2.1898	2.1898	0.0006%
Grazing Farm/Ranch	2.0263	2.0263	0.0002%
Regenerative Farm	2.6496	2.6496	0.0001%
Solar Array	1.7732	1.7739	0.0394%
Agrivoltaic Farm	1.8968	1.8969	0.0035%
Agritourist Center	1.7657	1.7658	0.0060%

Table 4: Values of $S(f)$ for each facility before and after population influx due to Micron Technologies facility construction

There are several insights that we could derive from the first single-change values. First, **there is an increase in the decision model score corresponding to the Micron Technologies facility construction**, which is highly expected due to the influx of high-income people. Therefore, our model is valid in this regard. We can even see that **the highest percent change in score corresponds to the cross-country skiing facility**, which has a 0.1652% increase in weight, which is expected since high-income people come to cross-country skiing facilities like these. However, one major downfall of this model is the effect that it has on the weight since the increases are small. One thing to observe is that revenue is just one small part out of all factors we are accounting for in the scoring function, which explains the small variations.

Double-Option Testing

Considering the influx of the population, we also consider the differences in the distribution of the agritourist center and the agrivoltaic farm when there is an influx in business, which can be seen in Figures 10 and 11.

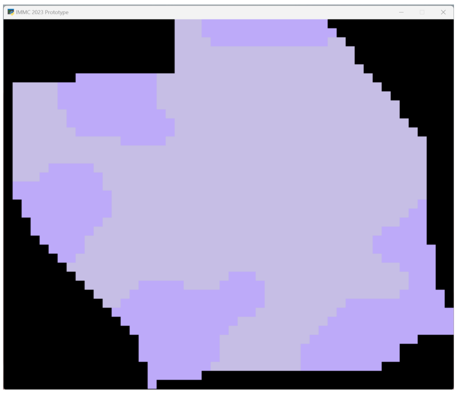

Figure 10: Distribution of the agritourist center (dark purple) and the agrivoltaic farm (light purple) before population influx ($S = 3.5311$)

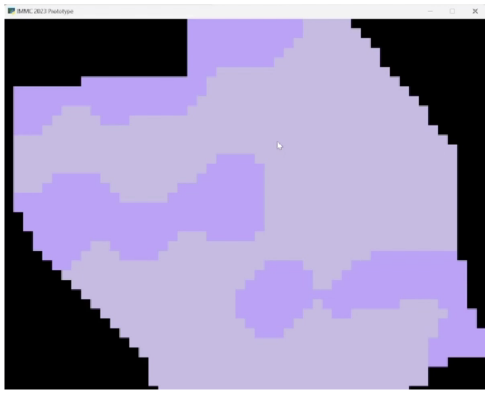

Figure 11: Distribution of the agritourist center (dark purple) and the agrivoltaic farm (light purple) after population influx ($S = 4.1560$)

There is a significant increase in score from $S(\text{Combined}) = 3.5311$ before the influx to $S(\text{Combined}) = 4.1560$ after the influx of population, which implies that either the benefits of bringing in an influx of population increase when the population increases or that the costs decrease. Either way, introducing an influx of high-income residents near a location would be **beneficial** to the visitor-centric business.

4.4 Strengths and Weaknesses

Strengths. The primary strength of this revenue-change model is its **simplicity**. Procuring data from Eq. 4.1 yields a result that is simplistic since this is an easy question to ask statistically. Additionally, **income is often a good predictor of consumer behavior**, as people with higher incomes tend to spend more money on goods and services. Moreover, determining the receptivity of a company to a certain annual income may be beneficial in identifying target markets in order to open their businesses to new markets, preferably high-income markets.

Weaknesses. One major weakness of this simple economic model is failing to account for the diversity of a consumer base that would be interested in the various goods and services that would be facilitated in the land since we assumed that the demand, or the people who want to buy a good or service, is monotonous for each of the facilities, which is not completely realistic. Moreover, the model doesn't account for other factors that could impact revenue, such as changes in the market, shifts in consumer preferences, or competitive pressures. However, one major change in the model that could be addressed is by combining it with other data such as demographic or psychological data in order to fully demonstrate consumer behavior and purchasing patterns.

5 Task 4: Discussion on Model Generalizability

So far in this paper, we have explored the potential land use of a three-square-kilometer New York property. However, our model is not restricted to be applied to this property only. As we have shown in Section 2, we constructed our model such that it can calculate the score and determine the approximate facility layout **for any given land area**. That is, the general framework of our computational model is already designed to be **flexible** toward a variety of input data.

However, there are certain aspects of the model that may be customized to adapt to changes in the business model, land location, and land-use policy.

5.1 Shifts in Business Models

Our model operates on a cost-to-benefit ratio, which is primarily profit-oriented and measured through the economic activity of the facility to be built on the given property. As such, all sources of revenue, taxes, operating costs, and others could be accounted for. That is, any flow of cash in or out of the facility could be classified into a benefit B or cost C.

However, there are other business models that do not operate to maximize profits, such as charity businesses or non-profit organizations. For this kind of model application, translating social welfare into monetary values becomes even more important. For example, we could redefine the long-term benefit in our scoring function to include terms such as the amount of money that the facility donates to charity, possibly with a multiplier to account for the "hidden", welfare influence that this would have on society. Meanwhile, the long-term cost could still be maintained as operating costs to keep the organization/facility running.

5.2 Changes in Land Locations

In our model, we accounted for five prominent land features (see Figure 5), namely the location of cell coverage, distance from the road, distance from the nearest water or irrigation source, topography/elevation, and tree cover. Additionally, a stable supply of water and electricity was already accounted for in the assumptions of the problem statement. However, not all lands have these features as their most prominent features. For instance, a different property could have unstable soil, which is unfavorable for certain facility options (e.g., high-rise buildings). For these other land features, we could customize the scoring function definition similar to Eq. 2.7 that would "penalize" (i.e., increased costs) of installing certain facilities in unfavorable region conditions.

5.3 Changes in Land-Use Policies

Since the given property is located in New York, our model accounted for existing local policies on land usage, such as Local Law 97 for carbon taxation. However, the model may be customized to factor in different land-use policies, especially if the property is located in a different state or country. For example, if there are government policies penalizing hazardous waste production, then the long-term cost in the model's scoring function may be modified to include a term on the price associated with hazardous waste.

6 Conclusion

In conclusion, our paper presents a computational, benefit-to-cost optimization model that allows for the division of a land property in New York that serves the interests of both community and business leaders.

In Task 1: Development of Best Land Use Decision Model, we described how we developed our model that calculates a metric S taking into account the economic costs and economic benefits, wherein a higher value of this metric increases the benefit of establishing a facility in this piece of land. This would be highest when the benefits are maximized and the costs are minimized. It also takes into account a frequent scenario where disjoint pieces of property (as shown in Figure 2) are conjoined into one piece of land through a graph theory clustering algorithm, a deterministic way of combining isolated regions, and taking into account a decrease in productivity when the size of the facility cluster continues to increase.

In Task 2: Testing the Decision Model, we were able to justifiably use real-world data to determine which facilities are best allocated towards the land. We ran two tests: (1) a test where we just established one facility on the property, and (2) a test where we allocated a pair of facilities based on varying business interests. Through the single facility test in 1, the highest score was for a regenerative farm with $S = 2.6496$, followed by a cross-country skiing facility with $S = 2.2518$, and the lowest was an outdoor sports complex with $S = 1.1875$. Notably, $S > 1$ for all of the tests, meaning that all options are likely to recuperate costs in the long term. The test for a pair of facilities is able to optimize the allocation of facilities based on varying business interests. However, out of three tests, the highest combined score was for a sports complex and skiing facility with $S(\text{Combined}) = 4.1784$, followed narrowly by the agrivoltaic farm and agritourist center ($S(\text{Combined}) = 3.5311$) and the crop farm and grazing farm ($S(\text{Combined}) = 3.5033$).

Task 3: Impacts of Population Influx is able to utilize a model relating the revenue of a company R and the income I of the residents in order to determine the change in score S after an influx or outflux of people with varying income. After a test wherein the high-income population increased, all of the values of the scores increase for single-facility tests. However, for a combination of business interests as shown in Figures 10 and 11, there is a massive increase in the score.

Task 4: Discussion on Model Generalizability was then able to explain why our model is generalizable for any given land area, where our general model is designed to be flexible towards a variety of input data. We showed that by modifying the benefit or cost functions B and C, we are able to determine how to change the model scoring functions in order to determine the best allocation of a facility or group of facilities in any given group of land.

Our land is a precious and limited gift that we must preserve for future generations; it is not just a resource to be exploited. We find our nourishment, beauty, and connection to the Earth there. Every part of the Earth is a work of art created by nature that has taken thousands of years to develop and is necessary for the survival of countless species, including our own. In order to maintain its beauty while juggling the demands of society and the environment, we must maximize its use especially guided by mathematical models. We owe it to everyone to protect our land and make the best use of it.

References

[1] United States Department of Agriculture. *Changes in the size and location of U.S. dairy farms.* 2004. URL: https://www.ers.usda.gov/webdocs/publications/45868/17034_err47b_1_.pdf.

[2] United States Department of Agriculture. *Farm production expenditures 2020 summary.* July 2018.

[3] United States Department of Agriculture. *Farms and land in farms 2021 summary.* Feb. 2022. URL: https://www.nass.usda.gov/Publications/Todays_Reports/reports/fnlo0222.pdf.

[4] United States Department of Agriculture. *United states department of agriculture ERS - farm business income.* URL: https://www.ers.usda.gov/topics/farm-economy/farm-sector-income-finances/farm-business-income/.

[5] *Agritourism salaries.* URL: https://www.simplyhired.com/salaries-k-agritourism-jobs.html.

[6] Usman Aslam. *How much does it cost to start a farm?* Apr. 2022. URL: https://farmingbase.com/cost-to-start-a-farm/.

[7] Wikipedia contributors. "Alpine valley ski area". In: *Wikipedia* (Mar. 2023). URL: https://en.wikipedia.org/wiki/Alpine_Valley_Ski_Area.

[8] Our World in Data. *Land use per kilogram of food product.* URL: https://ourworldindata.org/grapher/land-use-per-kg-poore.

[9] Virginia Gewin. *A new study on regenerative grazing complicates climate optimism.* Dec. 2022. URL: https://civileats.com/2021/01/06/a-new-study-on-regenerative-grazing-complicates-climate-optimism/.

[10] SE Group. *Quarry road recreation area opinion of probably project costs.* URL: https://static1.squarespace.com/static/558b558de4b0adbb054fc841/t/5bbb6135e79c70e043735db1/1539006775743/Quarry+Rd+Cost+Summary.pdf.

[11] *Irrigation - frequently asked questions.* URL: https://www.ag.ndsu.edu/irrigation/faqs.

[12] Virginia A. Ishler. *Dairy sense: Keeping the dairy right sized.* URL: https://extension.psu.edu/dairy-sense-keeping-the-dairy-right-sized.

[13] Jenna Jonaitis. "How much does it cost to build a road on my property? " In: *Angi* (Mar. 2022). URL: https://www.angi.com/articles/how-much-cost-build-road-property.htm.

[14] Allison Katz. "Planning for new york city's carbon tax: Local law 97 explained". In: *Withum* (Oct. 2022). URL: https://www.withum.com/resources/planning-for-new-york-citys-carbon-tax-local-law-97-explained/.

[15] Georgette Kilgore. "Solar farm income per acre calculator: See profit margin, costs, money made". In: *8 Billion Trees: Carbon Offset Projects Ecological Footprint Calculators* (Apr. 2023). URL: https://8billiontrees.com/solar-panels/solar-farm-income-per-acre/#:~:text=How%5C%20Much%5C%20Solar%5C%20Farm%5C%20Income,vary%5C%20depending%5C%20on%5C%20individual%5C%20projects.

[16] Stephanie Koncewicz. "How much does tree removal cost? (2023 guide)". In: *This Old House* (Apr. 2023). URL: https://www.thisoldhouse.com/gardening/reviews/average-cost-of-tree-removal.

[17] *Largest cross-country skiing areasin north america - biggest cross-country skiing areas*. URL: https://www.snow-online.com/largest_cross-country-skiing-areas/north-america.

[18] *Local law 97 - sustainable buildings*. URL: https://www.nyc.gov/site/sustainablebuildings/ll97/local-law-97.page.

[19] Gianfranco Morocutti. *Maintenance and management costs of open sport facilities*. URL: https://www.fig.net/resources/proceedings/fig_proceedings/korea/full-papers/pdf/session29/morocutti.pdf.

[20] *Mountain trails cross country ski center - 4301000 revenue*. URL: https://www.konaequity.com/company/mountain-trails-cross-country-ski-center-4864032943/.

[21] *Nordic united trail grooming*. URL: https://www.nordicunited.org/cross-country-skiing.html.

[22] Nathan Pelletier, Rich Pirog, and Rebecca Rasmussen. "Comparative life cycle environmental impacts of three beef production strategies in the upper midwestern united states". In: *Agricultural Systems* 103.6 (July 2010), pp. 380–389. DOI: 10.1016/j.agsy.2010.03.009.

[23] Lidia Piccerillo, Francesco Misiti, and Simone Digennaro. "Assessing the environmental impact of a university sport event: The case of the 75th italian national university championships". In: *Sustainability* 15.3 (Jan. 2023), p. 2267. DOI: 10.3390/su15032267. URL: https://www.mdpi.com/2071-1050/15/3/2267/pdf?version=1674731627.

[24] Kim Porter. "How much does a barn house cost?" In: (Feb. 2023). URL: https://www.forbes.com/advisor/mortgages/barn-house-cost/#:~:text=Generally%5C%2C%5C%20you%5C%20can%5C%20expect%5C%20to,as%5C%20electric%5C%20and%5C%20plumbing%5C%20installation..

[25] Salary.com. *Construction worker salary in new york, new york*. URL: https://www.salary.com/research/salary/listing/construction-worker-salary/new-york-ny.

[26] Salary.com. *Cross country coach salary in new york, new york*. URL: https://www.salary.com/research/salary/recruiting/cross-country-coach-salary/new-york-ny.

[27] Salary.com. *Farmer salary*. URL: https://www.salary.com/research/salary/recruiting/farmer-salary.

[28] Salary.com. *Solar energy salary*. URL: https://www.salary.com/research/salary/posting/solar-energy-salary.

[29] Salary.com. *Sports performance coach salary in new york, new york*. URL: https://www.salary.com/research/salary/posting/sports-performance-coach-salary/new-york-ny.

[30] Crossroads Consulting Services. *Economic analysis for a proposed new outdoor sports field complex in worcester county, maryland*. Aug. 2017. URL: https://mdstad.com/sites/default/files/2017-08/Worcester%5C%20County%5C%20Arena%5C%20&%5C%20Outdoor%5C%20Sports%5C%20Field%5C%20Complex%5C%20Economic%5C%20Analysis%5C%20Final%5C%20Report%5C%20August%5C%202017.pdf.

[31] Coldwell Solar. *How much investment do you need for a solar farm?* Jan. 2023. URL: https://coldwellsolar.com/commercial-solar-blog/how-much-investment-do-you-need-for-a-solar-farm/#:~:text=The%5C%20typical%5C%20cost%5C%20of%5C%20building,SEIA's%5C%20average%5C%20national%5C%20cost%5C%20numbers..

[32] SolarNRG Marketing Team. *Solar farm installation: A closer look - SolarNRG*. Jan. 2022. URL: https://solarnrg.ph/blog/solar-farm-installation/#:~:text=The%5C%20land%5C%20size%5C%20required%5C%20for,for%5C%20every%5C%201%5C%2Dmegawatt%5C%20installation..

[33] Daily Times. "How much would worcester county athletic complex cost? Study raises stakes with estimate". In: (Dec. 2022). URL: https://www.delmarvanow.com/story/news/local/maryland/2022/12/01/controversial-worcester-athletic-complex-study-predicts-cost-of-153m/69689482007/#:~:text=The%5C%20plan%5C%20for%5C%20the%5C%2095.521,restrooms%5C%2C%5C%20parking%5C%20and%5C%20concession%5C%20stands..

[34] Tristram O. West and Gregg Marland. "A synthesis of carbon sequestration, carbon emissions, and net carbon flux in agriculture: Comparing tillage practices in the united states". In: *Agriculture, Ecosystems Environment* 91.1-3 (Sept. 2002), pp. 217–232. DOI: 10.1016/s0167-8809(01)00233-x.

[35] *What would be the annual maintenance cost for a solar PV system?* URL: https://www.itsmysun.com/faqs/what-would-be-the-annual-maintenance-cost-for-a-solar-pv-system/#:~:text=Typically%5C%2C%5C%20the%5C%20maintenance%5C%20costs%5C%20for,1%5C%25%5C%20of%5C%20the%5C%20initial%5C%20cost..

[36] Sam Wigness. "What is the carbon footprint of solar panels?" In: *Solar.com* (Feb. 2023). URL: https://www.solar.com/learn/what-is-the-carbon-footprint-of-solar-panels/#:~:text=There%5C%20have%5C%20been%5C%20many%5C%20studies,kilowatt%5C%20hour%5C%20of%5C%20electricity%5C%20produced..

[37] Jessi Wyatt and Maggie Kristian. *The true land footprint of solar energy*. Sept. 2021. URL: https://betterenergy.org/blog/the-true-land-footprint-of-solar-energy/.

A1 Complete Data for Task 2 and 3 Model Testing

Option	W_{jobs} [in \$/yr]
Outdoor Sports Complex	36000
Cross-Country Skiing Facility	36000
Crop Farm	36000
Grazing Farm/Ranch	36000
Regenerative Farm	36000
Solar Array	36000
Agrivoltaic Farm	36000
Agritourist Center	36000

Table 5: Short-Term Economic Benefits Variables

Option	p_{acc} [\$/m]	p_{irr} [\$/m]	k	p_{cons} [\$/m^2]	p_{trees} [\$/m^2]	p_{labor} [\$/m^2]
Outdoor Sports Complex	1562.50	10	0.1	256.4	600	36000
Cross-Country Skiing Facility	1562.50	10	0.05	16.7	600	36000
Crop Farm	1562.50	10	0	0.19	600	36000
Grazing Farm/Ranch	1562.50	10	0.01	217.3	600	36000
Regenerative Farm	1562.50	10	0	1.9	600	36000
Solar Array	1562.50	10	0	99.5	600	36000
Agrivoltaic Farm	1562.50	10	0	99.69	600	36000
Agritourist Center	1562.50	10	0.02	74.3	600	36000

Table 6: Short-Term Economic Costs Variables

Option	R [\$/m^2]	w	N_{workers}	P_{solar} [\$]	σ_s
Outdoor Sports Complex	1.1900	36000	3	10.50	in program
Cross-Country Skiing Facility	17.8168	36000	3	10.50	in program
Crop Farm	0.0627	36000	3	10.50	in program
Grazing Farm/Ranch	0.0192	36000	3	10.50	in program
Regenerative Farm	0.0077	36000	3	10.50	in program
Solar Array	4.2500	36000	3	10.50	in program
Agrivoltaic Farm	0.3656	36000	3	10.50	in program
Agritourist Center	0.6360	36000	3	10.50	in program

Table 7: Long-Term Economic Costs Variables

Option	$p_{\text{operating}}$ [\$/m^2]	p_{utility} [\$/m^2]	p_{tax} [\$/kg]	m_{limit} [kg]	A_{tax} [kg/m^2]
Outdoor Sports Complex	1.0619	0.4013	2.68×10^{-7}	114.5	0.9097
Cross-Country Skiing Facility	0.0031	0.0005	2.68×10^{-7}	114.5	0.0168
Crop Farm	0.0993	0.0219	2.68×10^{-7}	114.5	0.0168
Grazing Farm/Ranch	0.0071	0.0034	2.68×10^{-7}	114.5	0.0589
Regenerative Farm	0.0177	0.0021	2.68×10^{-7}	114.5	0.0155
Solar Array	1.9990	0.8869	2.68×10^{-7}	114.5	0.0010
Agrivoltaic Farm	2.0983	0.0789	2.68×10^{-7}	114.5	0.0178
Agritourist Center	0.5841	0.1361	2.68×10^{-7}	114.5	0.4927

Table 8: Long-Term Economic Benefits Variables

A2 Source Code

A2.1 `__init__.py`

```
from pyglet.app import run
from pyglet.graphics import Batch
from pyglet.window import Window
from secrets import choice

from constants import (
    WIDTH,
    HEIGHT,
    CAPTION,
    VALUES_PATH,
    VALUES_SHEETNAMES,
    FACILITIES,
    IMAGE_DATA,
    RESOLUTION,
    ATTACK_DIRECTIONS,
    DIRECTIONS_PATH,
    DIRECTIONS_SHEET,
)
from library import PROPAGATION_STYLE_CHOICES
from utils import (
    center,
    initialize,
    parse_xlsx,
    get_directions,
)

import sys

sys.setrecursionlimit(int(1e9))

window = Window(width=WIDTH, height=HEIGHT,)
window.set_caption(caption=CAPTION)
center(window)

paused = True
batch = Batch()
grid, named_maps = initialize(
    facility_variables=parse_xlsx(VALUES_PATH, *VALUES_SHEETNAMES),
    facilities=FACILITIES,
    image_data=IMAGE_DATA,
    resolution=RESOLUTION,
    height=HEIGHT,
    batch=batch,
)

directions = get_directions(DIRECTIONS_PATH, DIRECTIONS_SHEET, len(grid.values[0]),
    len(grid.values))

@window.event
def on_key_press(symbol, modifier):
    global paused
    paused = not paused

@window.event
def on_draw():
    window.clear()
```

```
    if not paused:
        grid.update(
            named_maps=named_maps,
            attack_directions=ATTACK_DIRECTIONS,
            facilities=FACILITIES,
            directions=directions,
            style=choice(PROPAGATION_STYLE_CHOICES)
        )
    batch.draw()

if __name__ == '__main__':
    run()
```

A2.2 constants.py

```
'''
For formula values, kindly refer to library.py
'''
from library import Facility

WIDTH = 1000
HEIGHT = 822

RESOLUTION = 20 # Must be in multiples of 10

CAPTION = 'IMMC 2023 Prototype'

INPUT_DIRECTORY = 'input'
IMAGE_DIRECTORY = 'img'

DIRECTIONS_PATH = f'{INPUT_DIRECTORY}/directions.xlsx'
DIRECTIONS_SHEET = 'main'

IMAGE_NAMES = (
    'topography',
    'cell_coverage',
    'tree_cover',
    'distance_from_road',
    'distance_from_water',
)

IMAGE_DATA = tuple(
    (
        f'{IMAGE_DIRECTORY}/{name}.png',
        0,
        name
    ) for name in IMAGE_NAMES
)

VALUES_PATH = f'{INPUT_DIRECTORY}/values_.xlsx'
VALUES_SHEETNAMES = (
    'B_s',
    'B_l',
    'C_s',
    'C_l',
)

FACILITIES = (
    Facility(
        name='Outdoor Sports Complex',
        color=(0, 255, 242),
```

```
    ),
    Facility(
        name='Cross-country Skiing Facility',
        color=(198, 0, 229),
    ),
    Facility(
        name='Crop Farm',
        color=(216, 0, 57),
    ),
    Facility(
        name='Grazing Ranch',
        color=(198, 150, 29),
    ),
    Facility(
        name='Regenerative Farm',
        color=(8, 180, 22),
    ),
    Facility(
        name='Solar Array',
        color=(18, 110, 29),
    ),
    Facility(
        name='Agrivoltaic Farm',
        color=(198, 190, 229),
    ),
    Facility(
        name='Agritourist Center',
        color=(189, 170, 249),
    ),
)

ATTACK_DIRECTIONS = {
    'N': (0, 1),
    'NE': (1, 1),
    'E': (1, 0),
    'SE': (1, -1),
    'S': (0, -1),
    'SW': (-1, -1),
    'W': (-1, 0),
    'NW': (-1, 1),
}
```

A2.3 library.py

```
from collections import defaultdict
from decimal import Decimal
from math import e, log
from pyglet.graphics import Batch
from pyglet.shapes import Rectangle as PygletRectangle
from secrets import randbelow
from typing import Any, Iterable, Optional

SHORT_TERM_DURATION = Decimal(3)
LONG_TERM_DURATION = Decimal(15)

RANDOMIZED_INITIAL_GRID = True

HAS_INFLUX = False

MINIMUM_AREA = Decimal(800)
MAXIMUM_AREA = Decimal(1000)
```

```

BLACKLIST = {
    'Outdoor Sports Complex',
    'Cross-country Skiing Facility',
    'Crop Farm',
    'Grazing Ranch',
    'Regenerative Farm',
    'Solar Array',
    # 'Agrivoltaic Farm',
    # 'Agritourist Center',
}

PROPAGATION_STYLE_CHOICES = (0, 1, 2, 3)

PENALTY = Decimal(0.5)
ADVANTAGE = Decimal(3)

MAX_COLOR_VALUE = Decimal(255)
INFLUX_EFFECT = Decimal(1.14888)
_0 = Decimal(0)
_1 = Decimal(1)
_2 = Decimal(2)
_3 = Decimal(3)
_5 = Decimal(5)
_7 = Decimal(7)
_e = Decimal(e)
_100 = Decimal(100)
_neg_1 = Decimal(-1)

def ln(value: Decimal):
    return Decimal(log(value))

def color_code_to_value(value: Decimal):
    return value * _100 / MAX_COLOR_VALUE

class Facility:
    name: str
    color: tuple[int, int, int]

    def __init__(self, **kwargs):
        self.name = kwargs['name']
        self.color = kwargs['color']

    def average(
            self,
            map: list[list[Any]],
            connected_cells: Iterable[tuple[int, int]]
    ):
        return sum(
            map[y][x] * _100 / MAX_COLOR_VALUE
            for x, y in connected_cells
        ) / len(connected_cells)

    def short_term_benefits(self):
        return self.short_term_wages * self.short_term_workers

    def revenue(
        self,
        named_maps: dict[str, list[list[int]]],
        x: int,
        y: int,
```

```
        has_influx: bool,
    ):
        return (
            (
                _1 + color_code_to_value(named_maps['cell_coverage'][y][x])
            ) * self.average_revenue
            * (INFLUX_EFFECT if has_influx else 1)
        )

    def clean_energy_benefits(self):
        return (
            self.solar_reduction
            * (
                _2 / _3 * pow(
                    base=self.percent_solar,
                    exp=(_5 / _2)
                ) - _1 / _3
            )
        )

    def long_term_benefits(
        self,
        named_maps: dict[str, list[list[int]]],
        x: int,
        y: int,
        has_influx: bool,
    ):
        return self.revenue(
            named_maps=named_maps,
            x=x,
            y=y,
            has_influx=has_influx,
        ) + (
            self.long_term_wages * self.long_term_workers
        ) + self.clean_energy_benefits()

    def accessibility_price(
        self,
        named_maps: dict[str, list[list[int]]],
        x: int,
        y: int,
    ):
        return self.accessibility_factor *
            color_code_to_value(named_maps['distance_from_road'][y][x])

    def irrigation_price(
        self,
        named_maps: dict[str, list[list[int]]],
        x: int,
        y: int,
    ):
        return self.irrigation_factor *
            color_code_to_value(named_maps['distance_from_water'][y][x])

    def deforestation_price(
        self,
        named_maps: dict[str, list[list[int]]],
        x: int,
        y: int,
    ):
        return self.deforestation_factor * color_code_to_value(named_maps['tree_cover'][y][x])
```

```
    def construction_price(
        self,
        named_maps: dict[str, list[list[int]]],
        x: int,
        y: int,
        connected_cells: Iterable[tuple[int, int]] | None,
    ):
        area = Decimal(len(connected_cells)) if connected_cells is not None else _1
        average_topography = self.average_topography if connected_cells is not None else _1
        return area * self.construction_factor * (
            _1 - abs(
                _1 - self.constant * color_code_to_value(named_maps['topography'][y][x]) /
                    average_topography
            )
        )

    def labor_price(self):
        return self.short_term_wages * self.short_term_workers

    def short_term_costs(
        self,
        connected_cells: Iterable[tuple[int, int]] | None,
        named_maps: dict[str, list[list[int]]],
        x: int,
        y: int,
    ):
        return (
            self.accessibility_price(
                named_maps=named_maps,
                x=x,
                y=y,
            ) + self.irrigation_price(
                named_maps=named_maps,
                x=x,
                y=y,
            ) + self.deforestation_price(
                named_maps=named_maps,
                x=x,
                y=y,
            ) + self.construction_price(
                named_maps=named_maps,
                x=x,
                y=y,
                connected_cells=connected_cells,
            ) + self.labor_price()
        )

    def carbon_taxation(
        self,
        connected_cells: Iterable[tuple[int, int]] | None,
    ):
        area = Decimal(len(connected_cells)) if connected_cells is not None else _1
        return area * self.taxation_factor * max(_0, self.carbon_produced -
            self.upper_carbon_limit)

    def long_term_costs(
        self,
        connected_cells: Iterable[tuple[int, int]] | None,
    ):
        return self.operating_costs + self.utility_costs + self.carbon_taxation(
```

```python
            connected_cells=connected_cells
        )

    def ldmr_multiplier(
        self,
        connected_cells: Iterable[tuple[int, int]] | None,
    ):
        if connected_cells is None:
            return _1
        area = Decimal(len(connected_cells))
        if area < MINIMUM_AREA:
            return pow(
                base=_e,
                exp=(
                    _neg_1 * _7 / MINIMUM_AREA * area
                    + ln(ADVANTAGE - _1)
                    + _7 / MINIMUM_AREA
                ),
            )
        if area > MAXIMUM_AREA:
            return PENALTY + _1 / (
                area - MAXIMUM_AREA + _5
            )
        return _1

    def score(
        self,
        named_maps: dict[str, list[list[int]]],
        x: int,
        y: int,
        connected_cells: Optional[Iterable[tuple[int, int]]] = None,
    ):
        return (
            (
                self.short_term_benefits() * SHORT_TERM_DURATION
                + self.long_term_benefits(
                    named_maps=named_maps,
                    x=x,
                    y=y,
                    has_influx=HAS_INFLUX,
                ) * LONG_TERM_DURATION
            ) / (
                self.short_term_costs(
                    named_maps=named_maps,
                    x=x,
                    y=y,
                    connected_cells=connected_cells,
                ) * SHORT_TERM_DURATION
                + self.long_term_costs(
                    connected_cells=connected_cells
                ) * LONG_TERM_DURATION
            ) * self.ldmr_multiplier(
                connected_cells=connected_cells
            )
        )

class Rectangle(PygletRectangle):
    facility: Facility
    connected_cells: Iterable[tuple[int, int]]

    def __init__(self, **kwargs):
```

```
        self.facility = kwargs.pop('facility')
        super().__init__(**kwargs)

def valid(x: int, y: int, grid: list):
    return all([
        0 <= x < len(grid[0]),
        0 <= y < len(grid),
    ]) and grid[y][x]

def neighbors(x: int, y: int, grid: list[list[Any]]):
    for dx in (-1, 0, 1):
        for dy in (-1, 0, 1):
            if valid(x + dx, y + dy, grid):
                yield (x + dx, y + dy)

class Grid:
    values: list[list[Rectangle | None]]

    def __init__(
        self,
        named_maps: dict[str, list[list[int]]],
        merged_map: list[list[dict[str, int]]],
        facilities: Iterable[Facility],
        resolution: int,
        batch: Batch,
        height: int,
    ):
        self.values = [[None for _ in row] for row in merged_map]
        for y in range(len(self.values)):
            for x in range(len(self.values[y])):
                if any(merged_map[y][x].values()):
                    best_facility = None
                    best_score = -1
                    for facility in facilities:
                        if facility.name in BLACKLIST:
                            continue
                        score = randbelow(100) if RANDOMIZED_INITIAL_GRID else facility.score(
                            named_maps=named_maps,
                            x=x,
                            y=y,
                        )
                        if score > best_score:
                            best_facility = facility
                            best_score = score
                    self.values[y][x] = Rectangle(
                        x=x * resolution,
                        y=height - y * resolution,
                        width=resolution,
                        height=resolution,
                        color=best_facility.color,
                        batch=batch,
                        facility=best_facility,
                    )

    def clear_connected_cell_data(self):
        for y in range(len(self.values)):
            for x in range(len(self.values[y])):
                if self.values[y][x]:
                    self.values[y][x].connected_cells = None

    def set_connected_cell_data(
```

```
        self,
        named_maps: dict[str, list[list[int]]],
    ):
        def bfs_(
            connected_component: set,
            x: int,
            y: int,
        ):
            connected_component.add((x, y))
            for dx, dy in (
                (1, 0),
                (0, 1),
                (-1, 0),
                (0, -1)
            ):
                u = x + dx
                v = y + dy
                if valid(u, v, self.values):
                    if all([
                        (u, v) not in connected_component,
                        self.values[v][u].facility == self.values[y][x].facility,
                    ]):
                        bfs_(connected_component, u, v)

        for y in range(len(self.values)):
            for x in range(len(self.values[y])):
                if self.values[y][x]:
                    if getattr(self.values[y][x], 'connected_cells', None) is None:
                        connected_component = set()
                        bfs_(connected_component, x, y)
                        connected_component = tuple(connected_component)
                        average_topography = self.values[y][x].facility.average(
                            map=named_maps['topography'],
                            connected_cells=connected_component
                        )
                        for u, v in connected_component:
                            setattr(self.values[v][u], 'connected_cells', connected_component)
                            setattr(self.values[v][u].facility, 'average_topography',
                                average_topography)

    def get_preferred_direction(
        self,
        facilities: Iterable[Facility],
        directions: dict[str, set],
    ):
        seen = set()
        record = dict((facility, defaultdict(int)) for facility in facilities)
        for y in range(len(self.values)):
            for x in range(len(self.values[y])):
                if self.values[y][x] and (x, y) not in seen:
                    seen.update(self.values[y][x].connected_cells)
                    for u, v in self.values[y][x].connected_cells:
                        for direction, group in directions.items():
                            if (u, v) in group:
                                record[self.values[v][u].facility][direction] += 1
                                break
        for y in range(len(self.values)):
            for x in range(len(self.values[y])):
                if self.values[y][x]:
                    self.values[y][x].preferred_direction = max(
                        record[self.values[y][x].facility],
```

```
                        key=lambda direction: record[self.values[y][x].facility][direction]
                    )

    def update(
        self,
        named_maps: dict[str, list[list[int]]],
        attack_directions: dict[str, tuple[int, int]],
        facilities: Iterable[Facility],
        directions: dict[str, set],
        style: int,
    ):
        self.clear_connected_cell_data()
        self.set_connected_cell_data(named_maps)
        self.get_preferred_direction(
            facilities=facilities,
            directions=directions,
        )

        total_score = _0
        region_count = _0
        for y in range(len(self.values)):
            for x in range(len(self.values[y])):
                if self.values[y][x]:
                    region_count += _1
                    total_score += self.values[y][x].facility.score(
                        named_maps=named_maps,
                        x=x,
                        y=y,
                        connected_cells=self.values[y][x].connected_cells,
                    )
        print(total_score / region_count)

        for y in range(len(self.values)):
            for x in range(len(self.values[y])):
                if self.values[y][x]:
                    surrounded_by_tally = defaultdict(Decimal)
                    surrounded_by_scores = defaultdict(Decimal)
                    can_attack = set()
                    for nx, ny in neighbors(x=x, y=y, grid=self.values):
                        if all([
                            self.values[ny][nx].facility != self.values[y][x].facility,
                            (nx - x, ny - y) ==
                                attack_directions[self.values[ny][nx].preferred_direction]
                        ]):
                            can_attack.add(self.values[ny][nx].facility)
                        surrounded_by_tally[self.values[ny][nx].facility] += 1
                        surrounded_by_scores[self.values[ny][nx].facility] +=
                            self.values[ny][nx].facility.score(
                            named_maps=named_maps,
                            x=nx,
                            y=ny,
                            connected_cells=self.values[ny][nx].connected_cells,
                        )
                    for facility, tally in surrounded_by_tally.items():
                        surrounded_by_scores[facility] /= tally

                    new_value = None
                    can_rank_by_score = surrounded_by_scores and self.values[y][x].facility == min(
                        surrounded_by_scores,
                        key=lambda facility: surrounded_by_scores[facility]
                    )
```

```
                can_rank_by_tally = surrounded_by_tally and self.values[y][x].facility == min(
                    surrounded_by_tally,
                    key=lambda facility: surrounded_by_tally[facility]
                )
                match style:
                    case 0:
                        if can_rank_by_score:
                            new_value = sorted(
                                surrounded_by_scores,
                                key=lambda facility: surrounded_by_scores[facility]
                            )[-1]
                    case 1:
                        if can_rank_by_score:
                            new_value = sorted(
                                can_attack.union((None,)),
                                key=lambda facility: surrounded_by_scores[facility]
                            )[-1]
                    case 2:
                        if can_rank_by_tally:
                            new_value = sorted(
                                surrounded_by_tally,
                                key=lambda facility: surrounded_by_tally[facility]
                            )[-1]
                    case 3:
                        if can_rank_by_tally:
                            new_value = sorted(
                                can_attack.union((None,)),
                                key=lambda facility: surrounded_by_tally[facility]
                            )[-1]
                if new_value:
                    self.values[y][x].facility = new_value
                    self.values[y][x].color = new_value.color
```

A2.4 utils.py

```
from collections import defaultdict
from decimal import Decimal
from openpyxl import load_workbook
from PIL.Image import open as open_image
from pyglet.canvas import get_display
from pyglet.graphics import Batch
from pyglet.window import Window
from typing import Any, Iterable

from library import (
    Facility,
    Grid
)

def center(window: Window):
    screen = get_display().get_screens()[0]
    x = screen.width // 2 - window.width // 2
    window.set_location(x, 50)

def interpret(value):
    if value is None:
        return None
    if any([
        isinstance(value, float),
        isinstance(value, int),
        isinstance(value, str) and (value.isdigit() or value.isdecimal())
```

```
    ]):
        return Decimal(value)
    return str(value)

def parse_xlsx(path: str, *sheetnames):
    workbook = load_workbook(path)
    worksheets: list[list[list]] = []
    for name in sheetnames:
        worksheet = workbook[name]
        data: list[list] = []
        for row in worksheet.rows:
            data.append(list(interpret(cell.value) for cell in row))
        worksheets.append(data)
    workbook.close()
    return worksheets

def initialize(
    facility_variables: list[list[list[Any]]],
    facilities: Iterable[Facility],
    image_data: Iterable[tuple[str, int, str]],
    resolution: int,
    batch: Batch,
    height: int,
):
    mapped_facilities = dict((facility.name, facility) for facility in facilities)
    for sheet in facility_variables:
        headers = sheet[0]
        content = sheet[1:]
        for x in range(1, len(headers)):
            for y in range(len(content)):
                if content[y][0] is None:
                    break
                setattr(mapped_facilities[content[y][0]], headers[x], content[y][x])

    named_maps: dict[str, list[list[int]]] = {}
    for image_path, band_to_note, image_name in image_data:
        image = open_image(image_path, 'r')
        pixels = list(image.getdata(band=band_to_note))
        named_maps[image_name] = tuple(
            tuple(pixels[image.width * i:image.width * (i + 1)])
            for i in range(len(pixels) // image.width)
        )

    merged_map: list[list[dict[str, int]]] = [
        [defaultdict(int) for _ in row[::resolution]]
        for row in list(named_maps.values())[0][::resolution]
    ]
    for name, map in named_maps.items():
        for x in range(0, len(map[0]), resolution):
            for y in range(0, len(map), resolution):
                merged_map[y // resolution][x // resolution][name] = map[y][x]
        named_maps[name] = tuple(
            tuple(map[y][::resolution])
            for y in range(0, len(map), resolution)
        )

    grid = Grid(
        named_maps=named_maps,
        merged_map=merged_map,
        facilities=facilities,
        resolution=resolution,
```

```
        height=height,
        batch=batch,
    )

    return grid, named_maps

def get_directions(path, sheet, width, height):
    data = parse_xlsx(path, sheet)[0]
    directions = defaultdict(set)
    for y in range(height):
        for x in range(width):
            directions[data[y][x]].add((x, y))
    return directions
```