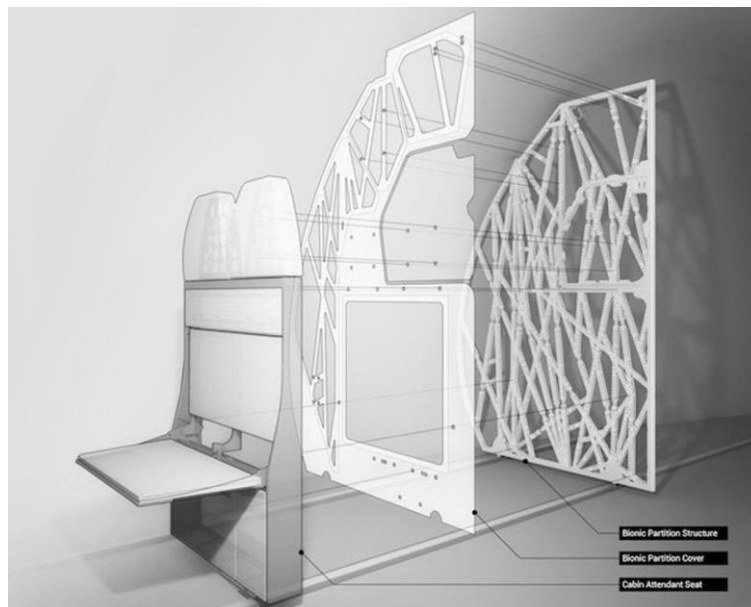


Generative Design

Let's revisit the car wheel design problem once more. By now, you should be well acquainted with the objectives and constraints of this design challenge. Given the significant advancements in technology, are there any approaches that can automate the design process? In other words, can we obtain computer-generated designs by specifying the design objectives and constraints? The answer is yes. There exist methods known as Generative Design (GD) methods that can accomplish precisely that.

GD is a design paradigm in which computer algorithms computationally consider human-defined objectives, parameter ranges, and constraints to generate designs. GD is an exciting and innovative technology that can help designers explore new possibilities and optimize their designs. GD originated in the 1970s when computers were used for complex design challenges such as shipbuilding and architecture. GD was first used commercially in the manufacturing industry as there was a demand for more efficient, cost-effective ways to design products. GD has been applied in various fields, such as aerospace, automotive, and biomedical. It has also been used for creating novel forms and structures that are not possible with traditional methods, such as an airplane partition wall that achieves 45% less weight¹.



¹ <https://www.airbus.com/en/newsroom/news/2016-03-pioneering-bionic-3d-printing>

3.1 GD: Problem Definition

Imagine you are looking for the best car wheel design. You have already defined a parametric schema that describes the number of spokes, the shape of each spoke, etc. You may test a few combinations yourself, but what does it take to have the computer find the best combination(s) for you?

The first step is to tell the computer what “best” means. In generative design, the designer must define the **design objectives and constraints** not just as vague descriptions, but rather as **computable expressions**. For example, if your objective is to maximize the stiffness of the car wheel, then you need to input an equation for stiffness. Common objectives may be built into the GD software already, but you still need to choose the right objective(s), or you may end up with wheels that are very *strong*, but not very *stiff*.

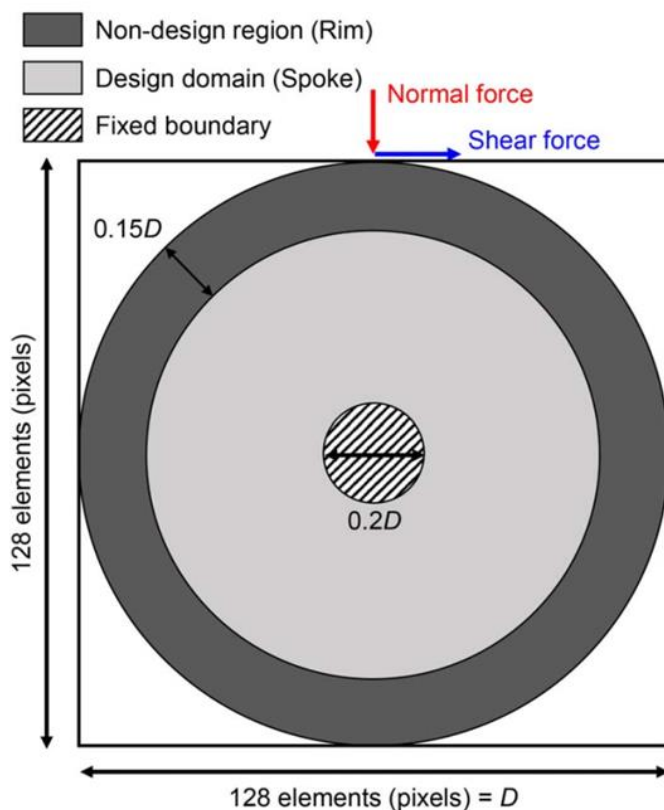


Figure: Example of the Problem Definition for car wheel design in GD. The figure sketches the design domain and boundary conditions, which serve as the design constraints. The objective can be set to maximize the stiffness of the car wheel.

Practice example: *Generative solar farm design*

A design problem often has more than one objective. Initially, our objective of designing a solar farm was to **maximize the total energy output** while **minimizing the cost**. In the real world, there are often other objectives that designers need to consider. For example:

- **Profit:** Solar farms are also a type of investment. The investors hope to **sell the energy** produced to **cover the cost** and eventually make a profit.
- **Average energy output:** The investors have many projects to invest in. The solar farm design needs to generate **as much output per solar panel** as possible, or the investors may look elsewhere.

For example, how can we calculate the **yearly average output** of a solar farm design? We can express it as a function of several variables:

$$\text{Yearly average output} = \text{yearly total output} / \text{number of solar panels}$$

Aladdin is also equipped with a GD algorithm that can generate solar farm designs automatically. But first, you - the human designer - must define the objective for it. Remember, the algorithm doesn't understand words like "profit", but it sure understands mathematical formulae or expressions.

1. Define **yearly profit** using these known quantities:

- Total yearly energy output (kWh)
- Electricity selling price (\$/kWh)
- Number of solar panels
- Yearly cost of solar panels (\$/panel)

Hints:

- You may want to calculate the yearly revenue (money earned) and yearly cost (money spent) first.
- Assume there is no upfront cost for solar panels. The only cost comes from yearly operations.

Define yearly profit:	Yearly profit = yearly revenue - yearly cost =
------------------------------	---

3.2 GD: Exploration

Given the explicitly defined constraints and objectives as input, the GD method can now automatically generate a large number of design options. The designer could review the generated solutions and select the most promising options for further exploration. They could use visualization and analysis tools to understand the performance and behavior of each solution. For example, in the car wheel design problem, there are three designs highlighted in the figure below that the designer finds appealing. Human preferences are essential for assessing and choosing designs in situations where non-engineering factors (e.g., the design's aesthetics) are critical. These designs can be chosen for further evaluation, including an assessment of their engineering performance (i.e., the stiffness of the car wheel), which has been calculated during the generative design process.

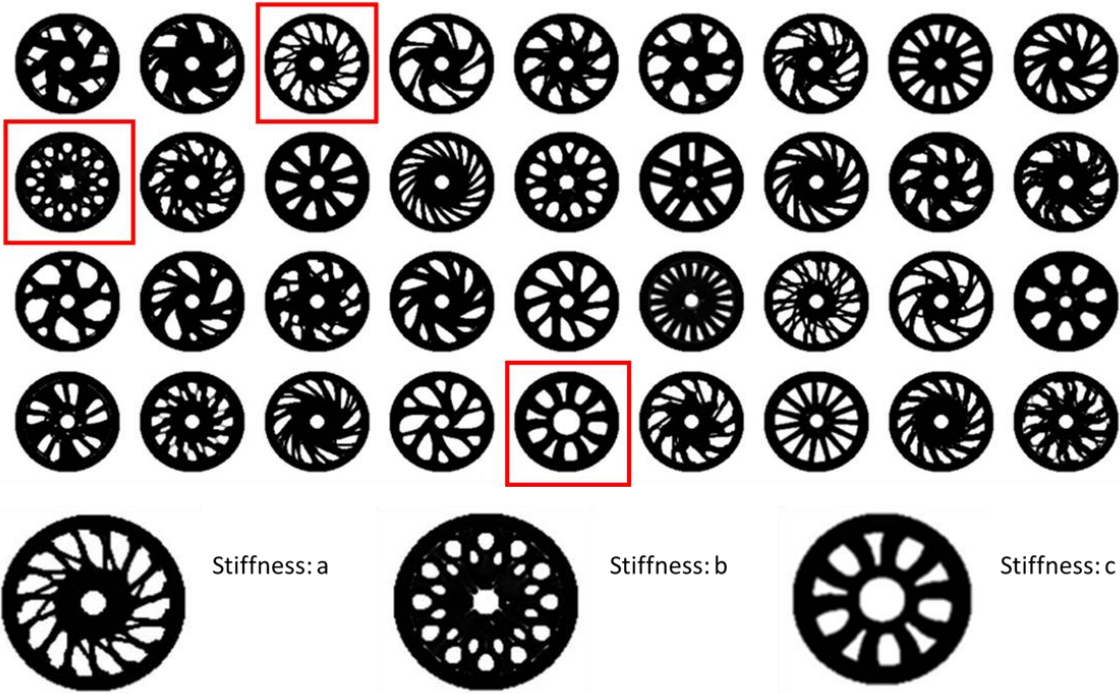


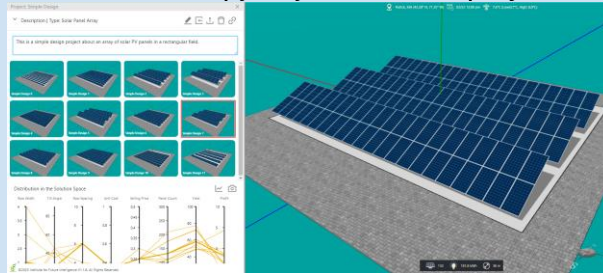
Figure: Example of Design Exploration of car wheels in GD. Numerous design options can be automatically generated along with the engineering performance. Comparison of designs can be made in terms of the shapes and stiffness.

Practice example: Generative solar farm design

Organize Your Designs with Projects

You can create, view, and compare multiple designs within a single *project*:

- **Open** this [Aladdin GD project template](#).
- **Follow slides 12-15** to make a copy of your own GD project.



Think back to the PD activity, where you entered the 3 design variables (tilt angle, row width, inter-row spacing), and the layout wizard generated one solar farm design based on those variables. Now in GD, you can simply set an **objective** and enter the **search ranges** for each design variable, and the GD algorithm will iterate through dozens of designs within those search ranges.

Steps:

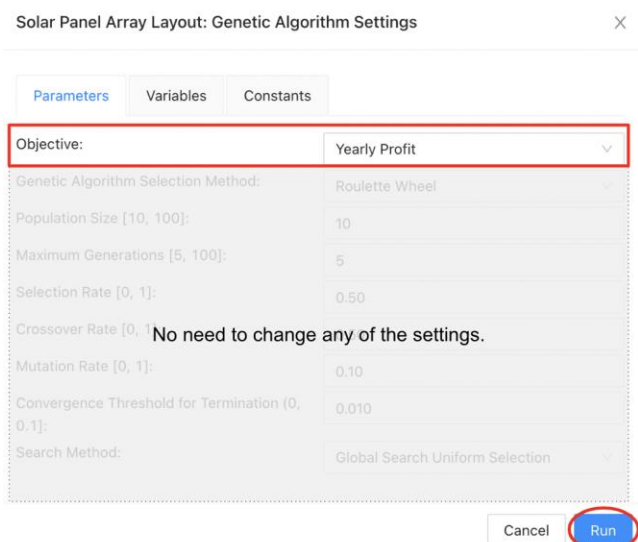
A. **Open** your own copy of the GD project and **double click** on the first design.

B. **Right click** on the shaded area and **select** “Layout > Solar Panel Array Layout: Generative Design > Genetic Algorithm”.

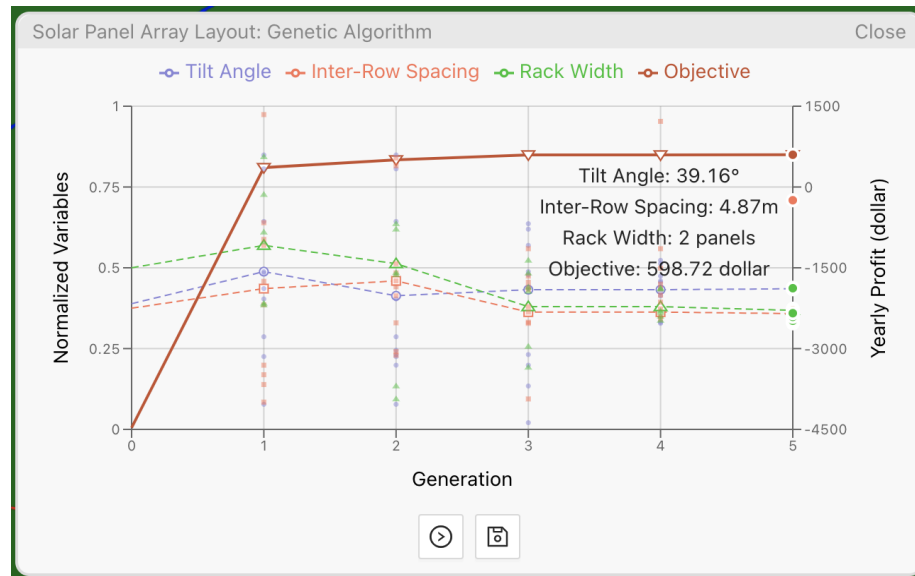



C. On the “Parameters” tab, **select** the desired objective. Keep everything else at their default values.

D. **Select “Run”**. (Use the default settings - no need to change anything.). Watch as AI tries different new designs.



- E. When AI completes its work, *Aladdin* will report the result in a graph. **Hover over** the last generation to see the best design that AI has generated in this run.

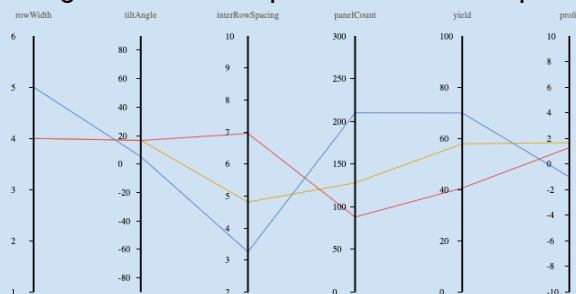


- F. **Update** the selected design  to save your first GD design, which will also overwrite the first design template. ([Aladdin tip: Update a design in a project](#))
- G. **Double click** the next design in the project and repeat steps B-F.

Organize Your Designs with *Projects*

2. **Generate** three different designs for each of the following objectives:
- Yearly profit
 - Yearly mean yield

Replace the placeholder image below with a parallel coordinate plot of your six GD designs.



3.3 GD: Evaluation

As mentioned earlier, the engineering performance, which serves as the objective in the generative design process, is typically provided alongside the generated designs. In the case of the car wheel design problem, stiffness is a crucial performance metric, but other factors like aerodynamic performance are also significant. Designers can choose their preferred designs and carry out additional performance evaluations accordingly.

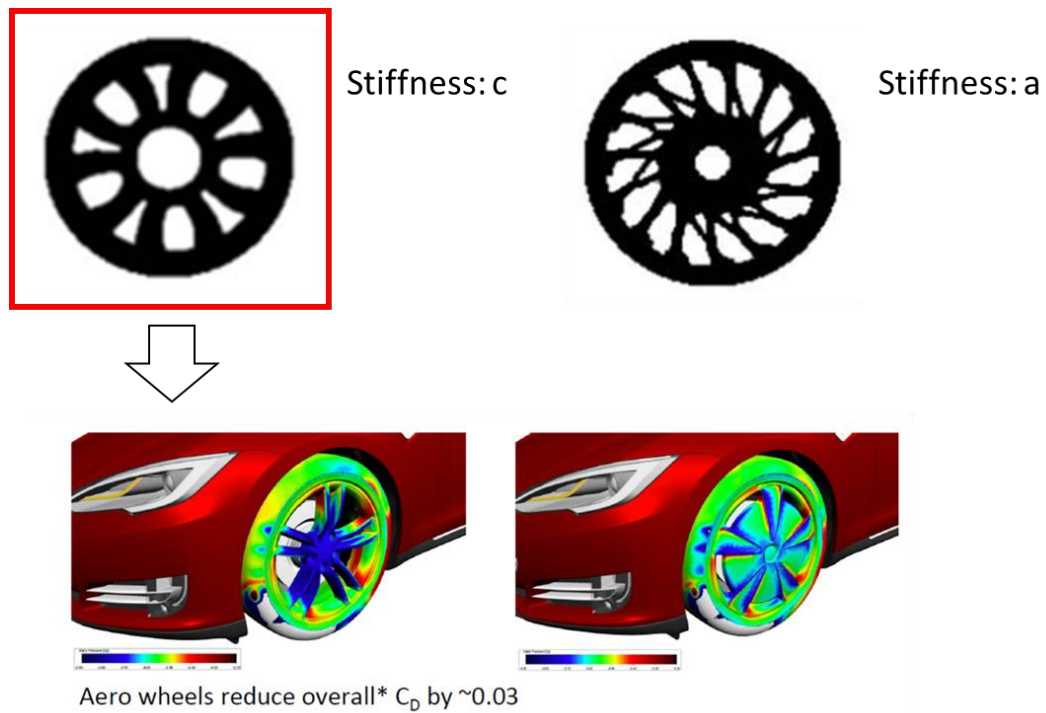
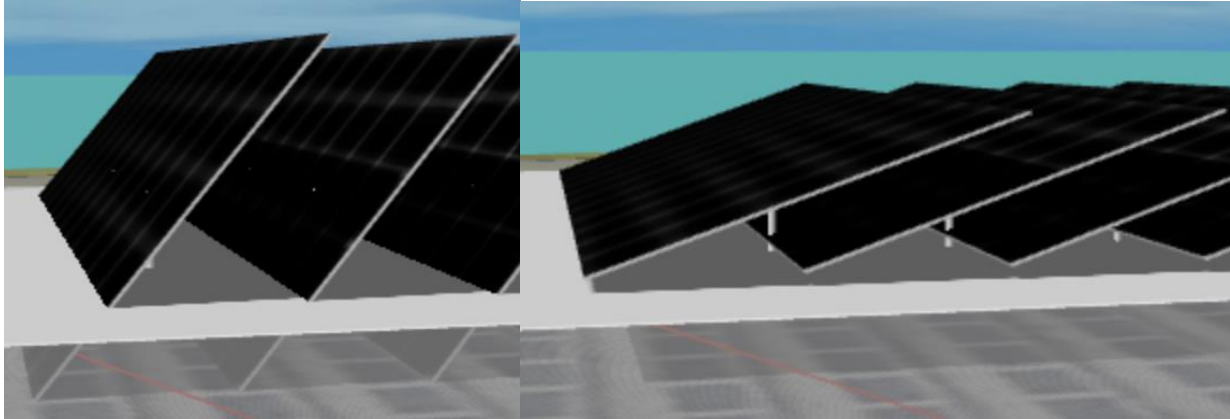


Figure: Example of Design Evaluation of car wheels in GD. Besides the stiffness available along with the designs, further engineering performance evaluation (e.g., aerodynamic performance evaluation) can be performed.

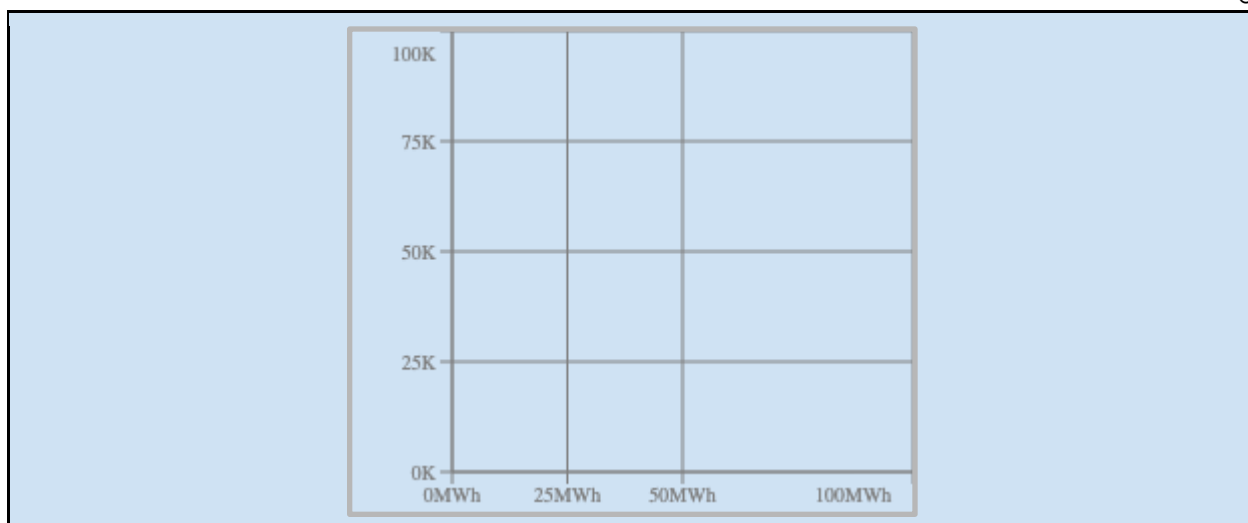
Practice example: *Generative solar farm design*

With generative design, all generated solar farms have already been evaluated by default, but there is also one drawback: The GD algorithm does not validate whether each generated design is physically feasible. In some cases, you may see generated solar farms with solar panels that overlap with one another or intersect with the ground. Therefore, only your human intervention can ensure that your final designs are also feasible in the real world.



Organize Your Designs with *Projects*

1. Open each of your GD designs and inspect if they meet the following constraints. For any design that does not meet either constraint, uncheck the checkbox on the bottom right corner of its thumbnail image in the project to remove it from the parallel coordinate plot.
 - a. Solar panels cannot overlap or intersect with one another.
 - b. Solar panels cannot intersect with the ground.
2. Generate a scatter plot with "Total Profit" as the X-axis and "Total Mean Yield" as the Y-axis.
3. Replace the placeholder plot below with the actual scatter plot.



1. Which design has the highest yearly total profit?
2. Which design has the highest yearly mean yield?
3. Which designs are non-dominated?
4. Which design achieves the best balance between profit and mean yield? Can you briefly explain why in 1-2 sentences? (There is no "one right answer".)

3.4 GD: Iteration

Based on the evaluation, the designer could iterate on the solution by adjusting the inputs and algorithms. The GD method could then generate a new set of solutions based on the updated inputs, and the design process could continue until a satisfactory solution is reached. Through a series of iterations, designers iterate between the design exploration and evaluation stages (sometimes even go back to the problem definition stage), continually refining and enhancing the concepts until they reach the optimal design. The optimal design is the one that best addresses the problem, satisfies user needs, and meets the desired objectives while considering constraints such as cost, manufacturability, and aesthetics. This iterative process allows for continuous improvement, ensuring that the final design is well-considered, effective, and meets the desired goals.

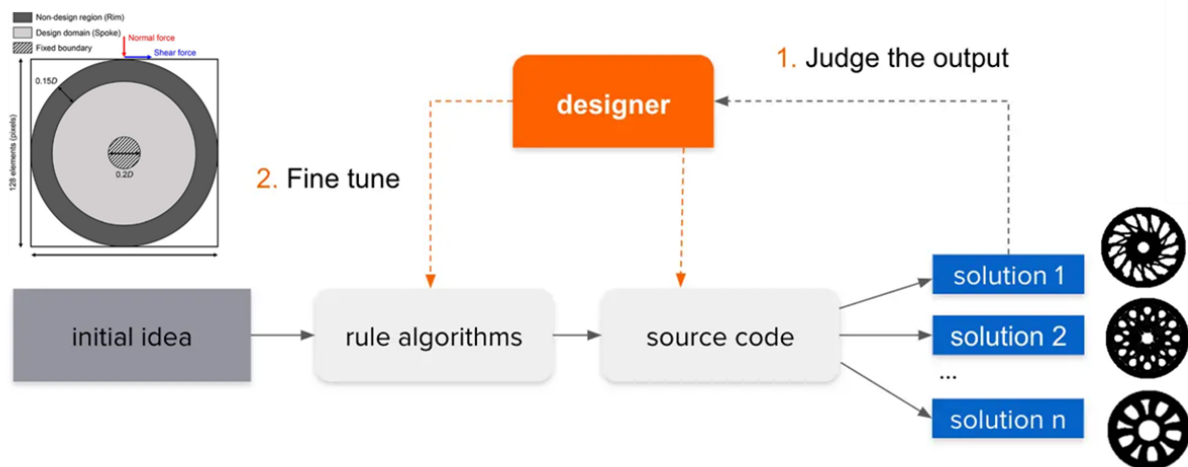


Figure: Example of Design Iteration of car wheels in GD. The key role of the designer is to incorporate feedback and insights from testing and evaluation and finetune the objectives and constraints for further design exploration. By repeating the process and making adjustments to address any issues, a satisfactory solution can be reached.


Practice example: *Generative* solar farm design

Maybe some of the generated solar farm designs already look pretty good to you, but what if there are *even better* designs to be discovered? As the human designer, you have a few options:

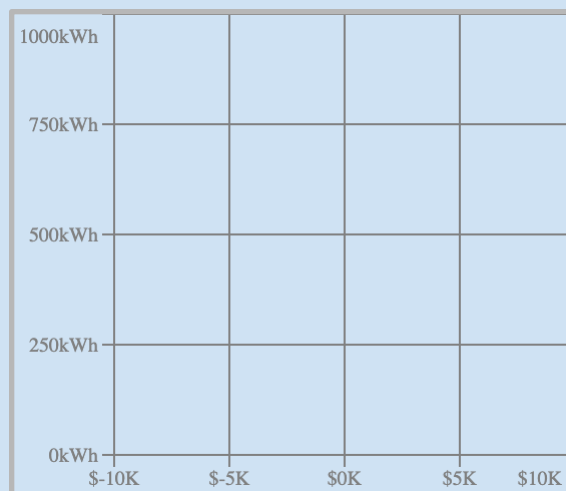
- Rerun GD with the exact same settings. The genetic algorithm it uses involves a certain degree of randomness, so each run may generate different results;
- Change the design objective. In addition to **yearly profit** and **yearly average output**, you can also set the objective to **yearly total output**.
- Change the search ranges of one or more design variables. For example, you can be more creative by searching in new areas in the design space (e.g., designs with wider rows), or be more efficient by avoiding certain areas (e.g., negative tilt angles).

1. What do you want to do next? Briefly explain your reasons in 1-2 sentences.

Organize Your Designs with *Projects*

2. **Generate** at least six more designs using any of the three yearly objectives (**total output**, **profit**, **average output**) and **curate** them  to your GD project. ([Aladdin tip: Add a new design to a project](#))

Once you finish all your designs, replace the placeholder plot below with a scatter plot of Profit (X-axis) vs. Mean Yield (Y-axis).



1. Compare your six (or more) new designs with your six (or more) old designs. Did you discover any new non-dominated designs? Are there any old non-dominated designs that have now become dominated?
2. Out of all 12 (or more) designs, which design achieves the best balance between profit and average output? Can you briefly explain why in 1-2 sentences? (There is no "one right answer".)
3. If you had more time, what else would you try to generate better designs?
4. What are some things you learned from your GD experience?