1 From Sensitivity to Robustness

In the last reading, we explored what-if analysis to evaluate how changes in parameters affect our model. By examining the technological coefficients of binding constraints, we identified when our optimal solution could become infeasible. This highlighted an important concept: a solution may not be considered *robust* if small changes in the model lead to failure. A *robust* solution is one that remains feasible under a wide range of conditions [2]. Up to this point, we've solved LPs using fixed parameter estimates and then assessed sensitivity after the fact. But what if we can't confidently estimate a parameter? In such cases, we can turn to *robust optimization*, where the goal is to find a solution that remains feasible and **near-optimal**, across all plausible parameter combinations [1]. This type of solution is referred to as a *robust solution* [1].

A key point here is that the robust solution may not be the optimal solution for all situations. Rather, it is optimal for the worst-case scenario we are guarding against. It remains *feasible* across all plausible parameter values, even if that comes at the cost of reduced performance in more favorable conditions. When we say that a robust solution is "near-optimal", this should not be misunderstood as "almost as good" in an absolute sense; how close the robust solution is to the original optimal depends on the level of uncertainty we're accounting for. A wider range of uncertainty may lead to a more conservative, and potentially lower-performing, solution. The tradeoff we're making is between maximizing performance under ideal assumptions and ensuring reliability when conditions vary.

2 Robust Optimization

2.1 Worst-Case Thinking in an LP

Robust optimization focuses on finding a solution that remains feasible under the worst-case values of uncertain parameters. For example, if we were unsure of an accurate estimate for the number of hours it takes to produce a window at Plant 2, but we did know that it could take anywhere from one to three hours, robust optimization solves the problem under the assumption that we are in the worst-case scenario. For this specific context, windows requiring three hours to produce is the worst-case, so we would solve our problem with the technological coefficient $a_{22} = 3$. This assumption is a context based assumption and does not affect the model-based LP assumptions that we defined in a previous reading.

The worst-case can apply to any parameter in our model. We might analyze a worst-case objective function, a worst-case set of constraints, or even a combination of both. Because the additivity assumption in linear programming treats parameters independently, we are free to apply worst-case values selectively. That means we can assign the worst-case to a parameter with high uncertainty while keeping best estimates for parameters we trust.

How do we identify what a worst-case scenario is? We rely on context to help us determine what is meant by a worst case. In a maximization objective function, for example, the worst-case of a range of values would be the minimum value. So, in the Wyndor Glass Co. example, if we were unsure of the profit margin on the sale of one door, but we generally knew it was between \$200 and \$400 the worst-case would be a profit margin of \$200.

For example, in the Wyndor Glass Co. problem, perhaps there have been some equipment malfunctions at Plant 3. The equipment malfunctions have the potential to reduce the available number of total production hours at Plant 3 to 10 hours a week; so the total number of available production hours could be anywhere from 10 to 18 hours a week. The equipment malfunctions also can increase the time it takes to manufacture each door and window. We are unsure the time it will take, but we do know that it could take 2.5 to 4 hours to manufacture one door and 1.5 to 3 hours to manufacture one window. We should be able to think through this contextually: more hours available means we can manufacture more things, so the worst-case for available hours is 10 hours per week. However, more hours with respect to production time per each product means that we can produce fewer windows and doors, making the worst-case for the technological coefficients the higher times, so 4 hours for doors and 3 hours for windows.

This means the constraint for Plant 3 that we would use to solve for the robust solution is:

$$4x_1 + 3x_2 \leq 10$$

2.2 Exploring Robustness Graphically

If we continue to use the example from above, we should recognize from our what-if analysis lessons that the constraint created to solve for the robust solution is going to change our optimal solution. But how does it change, and how is it now robust?

First, let's solve for the robust solution graphically as shown in Figure 1. We can see that this constraint significantly changed the feasible region. Our new optimal solution is to produce 0 doors and 3.34 windows every week.

Our definition for a robust solution was that it was feasible and near-optimal for all possible values of the parameters. So, what does this mean? Let's look at what happens as we select parameters within the range of uncertainty, working back up to the best-case scenario for our problem.

As shown in Figure 2, as we create "better" scenarios by decreasing the manufacturing time and increasing the available time at the plant, the feasible region increases

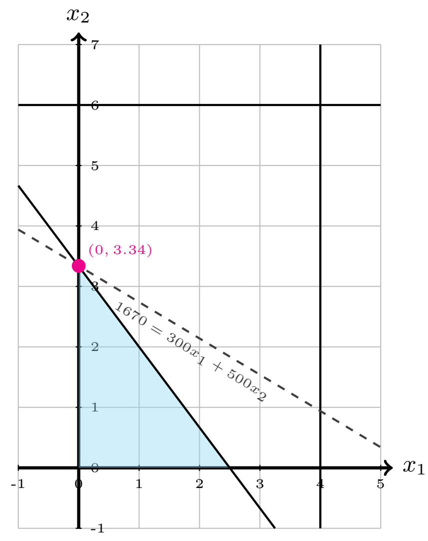

Figure 1: Feasible region for the robust constraint $4x_1 + 3x_2 \leq 10$, showing the resulting optimal solution $(0, 3.34)$.

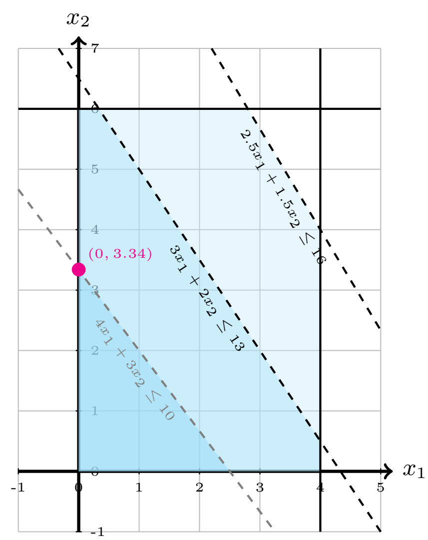

Figure 2: Feasible regions created as different combination of parameters are selected for constraint $a_{31}x_1 + a_{32}x_2 \leq b_3$, with the current robust solution $(0, 3.34)$ plotted in pink.

in size. However, notice that the robust solution remains a feasible solution regardless of how we change the parameters. The robust solution will be usable even if the conditions change. It is not the optimal solution for every scenario, but it will remain feasible. This is beneficial, especially when there is uncertainty or volatility within parameters.

2.3 Interpreting and Evaluating Robust Solutions

Robust solutions represent a tradeoff between optimality and reliability. In contexts where uncertainty is high, such as unpredictable equipment availability, choosing a robust solution means prioritizing feasibility across all plausible scenarios rather than maximizing performance under ideal conditions.

In the Wyndor Glass Co. example, the robust solution of producing 0 doors and 3.34 windows per week is not optimal in the best-case scenario. However, it ensures that the production plan remains feasible even if Plant 3 experiences equipment malfunctions that reduce available time or increase production hours per unit. If such failures occur, a non-robust plan could fail entirely, leaving Wyndor unable to meet production goals.

Evaluating a robust solution involves both interpretation and judgment. Key questions to consider include:

- What scenarios or risks does this solution protect against?
- How does it compare to the non-robust optimal solution?
- What are the consequences if the non-robust solution becomes infeasible?
- Are our assumptions about uncertainty reasonable and supported by data or expertise?
- Are we clearly communicating the tradeoffs involved to decision-makers?

These questions are both practical and ethical. Overly conservative models may lead to underutilized resources or lost opportunities, while overly optimistic ones may fail under pressure. Just as we assess the validity of our assumptions and data, we must ensure that our solutions reflect responsible planning. Robust optimization is not about pessimism; it's about preparedness in the face of uncertainty.

2.4 Ethical Considerations and Future Responsibility

While you may not be the one making final decisions on risk and uncertainty today, in the future you likely will be. More often than not, you won't be the analyst creating the model; you'll be the decision-maker interpreting its results and assuming its risk. This makes it essential to revisit our ethical checklist.

Start with data validity: Where did the uncertainty ranges come from? Are they supported by historical data, expert input, or just gut instinct? If the underlying data is flawed or incomplete, even the most carefully constructed robust solution can give a false sense of security. Next, consider model validity and stakeholder impact:

Who are the stakeholders affected by this solution? What assumptions were made about uncertainty, and who bears the cost if those assumptions are wrong? What happens if you're too conservative or too optimistic? Finally, think about communication: Have the tradeoffs been clearly explained, not just in terms of dollars or units, but in terms of impact?

Even if you don't yet have all the answers, you are responsible for asking the right questions. That mindset is what turns a mathematically sound model into an ethically sound decision.

References

[1] Frederick Hillier et al. *MA103 Mathematical Modeling: Introduction to Management, Science, & Business Analytics with Connect*. McGraw-Hill, 2024.

[2] Merriam-Webster. *Robust*. https://www.merriam-webster.com/dictionary/robust. Accessed: 2025-06-06. 2025.