REMSOFT®

# A recipe for understanding linear programming

Linear programming (LP) is a mathematical model for optimal solution of resource allocation problems. It does so by finding the maximum (or minimum) of linear functions in many variables subject to constraints. So what does that all mean to non-mathematicians?

To use LP effectively, the problem you are trying to solve must involve decisions that are linearly independent of each other. That means your alternative uses of a resource must scale in a linear fashion (e.g. if 1 serving = 1 chicken breast then 4 servings must equal 4 chicken breasts, not 3 and not 5). It also means that the decision to make chicken cacciatore is not somehow influenced by decision to make chicken Alfredo instead. Finally, the decisions must be divisible, meaning that you can choose to make any fractional quantity of each that you like (including none at all).

So what would a linear programming version of the menu problem look like? Suppose one serving of cacciatore requires $^{4}/_{5}$ of a chicken breast but one serving of Alfredo requires an entire chicken breast. If we want to serve as many people as possible, what is our optimal menu if we have 4 chicken breasts on hand? Well, if CC = # servings of cacciatore and CA = # servings of Alfredo, our objective is to maximize CC + CA, but we are limited (constrained) by the number of chicken breasts required, which is 0.8 CC + 1 CA = 4. If we make all Alfredo (CA=4), we can feed 0 + 4 = 4 people but if we make all cacciatore, we can actually serve 5 people because 0.8(5) +1(0) = 4 breasts and 5 + 0 = 5 (maximum). This isn't a very difficult or interesting problem and can be solved by inspection.

But now suppose the other ingredients in our recipe are also limited. For example, cacciatore requires ½ can of tomatoes and ½ oz grated parmesan cheese per serving; Alfredo requires 1 oz of grated parmesan cheese per serving. Our pantry only has one can of tomatoes and we have 2 oz of grated parmesan on hand. What is our optimum menu now?

Max CC + CA ; objective function subject to
0.8 CC + 1 CA $\leq$ 4 ; chicken breasts
0.5 CC + 0 CA $\leq$ 1 ; can of tomatoes, Alfredo uses no tomatoes
0.5 CA + 1 CA $\leq$ 2 ; oz of parmesan.
CC, CA $\geq$ 0 ; cannot prepare negative quantities of food

Clearly, we can't make 5 servings of cacciatore because that would require 2.5 cans of tomatoes. Similarly, we can't make more than 2 servings of Alfredo because that would require more than 2 oz of parmesan cheese. However, if we make 2 servings of cacciatore (using up the tomatoes and 1 oz of parmesan), we can still make one serving of Alfredo to serve a total of 3 people.

There are a couple of lessons here. First, even though a resource is constrained, the optimal solution may not be limited by that resource (there is surplus). Such constraints are called *non-binding*, and in our example, chicken breasts are no longer a binding constraint because we have one left over. There is no way to use that chicken breast in our menu without requiring more parmesan cheese and/or tomatoes.

Second, when you have additional limitations on resources (constraints), your objective will either stay the same or go down; you cannot add a constraint to yield a higher objective function value. With constraints on chicken only we could serve at most 5 guests; once other resources became limited, we could only serve at most 3 people even though we had leftover chicken. If we had invited 4 guests to dinner before checking the ingredients on hand, we would have an additional constraint on the problem: CC + CA $\geq$ 4. Since we have already determined that we can serve at most 3 people with our available ingredients, the addition of the constraint on guests makes the problem *infeasible* (no solution).

Suppose we realize in our hurried preparations that we had cooked pasta for 2 servings of chicken Alfredo, and rather than waste the pasta we decide to use it by making 2 servings of Alfredo. In effect, we have added another constraint (CA = 2) and as a result, we can no longer prepare any cacciatore and therefore by forcing this decision we have reduced the objective function value from 3 guests to 2. Among operations research practitioners, this is known as a *reduced cost*.

Now suppose we look deep in the back of the refrigerator and find a 1 oz scrap of leftover parmaggiano reggiano that can be grated by hand. What is that extra resource worth to us? We have changed the resource limit on parmesan by increasing the *right-hand-side* of the equation from 2 to 3 oz of parmesan (0.5 CA + 1 CA $\leq$ 3).What is the effect of that? Well, without more tomatoes, we still can't make any more cacciatore, but we can make an additional serving of Alfredo! This concept is called shadow price, and indicates the value of an additional unit of a resource relative to our objective. Since our objective is measured in guests served, the shadow price of an ounce of parmesan cheese is 1 guest.

In a nutshell, that's all there is to LP. Obviously real-world problems can have hundreds or thousands of resources, and millions of decisions, but the fundamentals are the same. What is great about LP is that solvers will tell report to you not only the objective function value and the optimum decisions, but also the shadow price and reduced costs, so you can really understand why you got the optimum solution you did.