www.mbym.it

SINGLE OBJECTIVE FUNCTION REDUCTION IN DECISION MODELING WHITE PAPER

February 2011

By Filippo Vasta , President MbyM s.r.l. Corciano (PG)

Abstract

This paper is intended to provide recommendations on how to reduce an optimization problem to a single objective function optimization regardless of the type of algorithm used

Problem Definition

A decision imply the choice of a set of values for a number of discretionary variables influencing the level of a number of ouput variable that we intend to control.

Optimizing a decision is about finding the levels for the discretionary variables that allow us to get the desired output as close as possible to a target (min or max)

The "Objective Function"

In general we can refer to the "Objective Function" as to the algebraic / numeric relations between those output variables and the discretionary variables they depend on.

For example we might want to maximize profit by selling a higher proportion of high contribution margin product , yet we might incur in management costs related to logistics or shelf space and the like having a different impact on various product items. In this case our objective functions would be the set of calculation leading to total profit depending on the number of product sold, their price and unit costs and all the formulas detailing all related management costs.

"Constraints"

Optimization is about scarce resources. There is no need to optimize the use of an unlimited and priceless resources. Therefore every time we deal with an optimization problem we are likely to face scarcity of a number of resources. In the case of a production plan, scarce resources might be workers available hours , maximum machines production rates , available space for storing finished goods and the like.

An optimization algorithm on top of maximizing or minimizing the objective function has to take into account the usage limitations imposed by the problem specific scarce resources.

Our suggestion is to manage constraints by including them within the objective functions in terms of penalties and rewards. This will be further analyzed in a subsequent section.

The problem with multiple objective functions and the differential cost concept

In real life is often the case that we want to maximize or minimize more than one objective function. For example we might want to maximize profit but also keep total sales above a certain level , or we might want to define a production plan that minimizes set up costs and in the same time minimize stock-out and maximize personnel utilization. This generate "competition" between objectives, and this competition adds instability to the convergence of whatever optimization algorithm.

In many cases there will be several possible solutions or equally applicable. To overcome this problem we also suggest to follow the same recommendation we gave for managing the constraints , that is including all objective functions in one function via penalty / reward sub-functions.

The case for just one objective functions

Optimization can be visualized as the search of the highest peak in a mountain chain. We can hike one and get to the top but we will not know if this peak is the highest so we should go down and up somewhere else finding may be a higher one. Our walking correspond to different values of the discretionary variables. At times we may find on our way up a fence (constraint) that forces us to go back (although that peak we were hiking might be the highest) and search for a new path or just stop there if we believe that where we are is actually the highest reachable point.

Imagine now you have a twin brother searching for the peak on a different chain of mountain and you both win when you both find the highest peak. Also you both have a GPS and the rule is that he moves exactly the way you move and vice versa. When he sees a path that might lead him to a high peak he'll call you and suggest he drives and you follow, you might accept or in the case you believe the path you are on is leading you to a higher peak you might negotiate with him that he follows and you drive. After a while probably you will both be exhausted and frustrated. This because any time you thought might be the right peak for you he called you and suggested you change path as he was going on a very steep and promising trail and may be in the end his trail was not so promising and you because of having to follow exactly his moves found yourself in a bottom of a canyon. This is what can happen when you try to optimize multiple objective function. Now imagine you change the rules and say that you win when the sum of the peak height you are in is highest. In that case will be easier to collaborate as you could both choose less risky path that may be are not so steep but showing a definite visible trend up so you both can be pretty sure that moving in that direction will lead to positive increase for both so for the sum of the heights . What we have done here we have combined two objective function in a single one making life easier for our optimizing algorithm.

Aligning objective functions

We observed in the previous section how "frustrating" was the inability to achieve an optimal situation with multiple objectives. We solved that by adding together the different objective function so moving a step forward concerning stability. Clearly the game will be easier to win if the two objective functions would be aligned so that when one goes up also the other one goes up so does the sum. We need to plan for that by carefully engineering the overall function.

Let's make an example and imagine you want to define an objective function representing the overall quality of a production schedule . The two main objective here are serving the customer according to his priority and in the same time minimize set-up. Clearly reducing set-up release more production resources for faster customer service so we might begin by considering the total plan horizon as an initial proxy for the objective function. If we only considered the set-ups and not the customer service we could find the optimal sequence as the one that reduces at a minimum total set-up time and so total planning horizon time (set-up time + processing time).

We intend now to add a corrective factor to take into account customer priority. We can do that by defining a priority scale with let's say three priorities (9 = highest, 3 = medium, 1=low) if we multiply the time to release the production order for customer “k” by its priority and add them up we get an aligned objective function. In fact in order to lower this overall function the algorithm need to lower each single term. The highest terms will be the ones with the highest priority and the longest planned delivery time. In order to reduce such terms the algorithm will try to anticipate such orders so reducing the delivery time part of the factor. As the delivery time is the result of processing time plus set-up time then reduction of delivery time will imply reduction of total set-up time. We see that the new objective functions is sufficient for overall optimization, in fact reducing the sum of all factors will lead to the best sequence (lowest delivery time for each single order) and the best service level (highest priority customers served before lowest priority customers).

Integrating constraints into the overall objective function

Managing constraints is always tricky , there is evidence that tight constraints reduce the search space and often reduce the ability to find optimal solutions especially when using non linear methods. When we say “think out of the box” we mean getting rid of “mental constraints” so our creative energy can flow “unconstrained” and get us quickly to the solution. Getting out of the box means to be aware of constraints while giving them a relative value. For example someone might say that alcohol must be prohibited because is a social plague creating discomfort , security problems and potentially have a negative impact on overall economy. If we want to define an objective function to maximize overall social wealth , from this perspective we should include in the model “alcohol” (or any other potentially harmfull substance) prohibition as an “hard constraint” in our decision model.

If our model is representing adequately the relations between different elements of society we might end up with a suboptimal solution for growth with factors such as : security costs for excess of organized crime for illegal distribution market, additional police costs for fighting the growing organized crime, higher health costs for the treatment of a progressively growing number of addicted as a consequence of a natural attitude of human being to go against prohibition just for the sake of it.

Transforming the hard constraints in to a soft constraints might mean for example: allowing the sale of alcohol subjective to a penalty function (sales tax on spirit) in the overall economy objective function. To begin with the sales tax will add a positive term to wealth . This additional wealth might offset some additional health care costs and awareness campaign costs though probably we would see a reduction on health treatment costs as a consequence of reduce percentage of addiction related to a lower desire to practice excess as a pure reaction to prohibition and so on.

Same concepts can be applied from political economy modeling to production planning. A typical production planning model might include constraints such as: capacity utilization that cannot exceed 100% , shelf life that cannot exceed a certain number of weeks and or warehouse maximum capacity that cannot be overcome. Transforming this hard constraints into soft penalty functions for inclusion into the overall objective function is a sensible approach. For example we might allow the system going above maximum utilization by including a differential utilization cost term with an personnel hourly rate as a function of utilization going unrealistically high above 100% utilization. In this case the optimization algorithms will naturally avoid situations leading to overcapacity because the overutilization term will be so high to imply a sharp increase in the overall objective function.

Same logic can be followed for the warehouse constraint where the overcapacity situation might be considered as the need for a very expensive additional rented space. This opportunity need not to be real as the algorithm, because of the penalty function, will rarely propose solutions implying the use of additional space.

We see that the idea of transforming hard constraints into soft constraints is to envision theoretical situations allowing for solutions to exceed the limits imposed by the constraint at a cost sufficiently high to discourage the algorithm from choosing such solutions.

Potential applications

So far applications explored by MbyM include:

1. Production planning and scheduling
    a. How to serve customers at best while minimizing differential production costs
2. Production Lay-out optimization
    a. How to choose a layout that optimize available space while reducing overall internal material transfer costs
3. Custom production parametric based cost estimate
    a. How to realistically allocate total production hours to elements of a bill of material
4. Strategic planning optimization for private and public sector
    a. How to select the most effective portfolio of research projects to ensure maximum medium term profit
5. Strategic sourcing optimization
    a. In a multiple production site, multiple absorbing markets , multiple products assign items to factory in order to minimize overall production and logistic costs.
6. Payable optimization
    a. In a situation of limited cash-flow and many outstanding payables find a delayed payment schemes allowing for stability of cash-flow and relative satisfaction of suppliers.