Towards an Optimal Design: What Can We Recommend to Elon Musk?

1st Martine Ceberio
2nd Olga Kosheleva
3rd Vladik Kreinovich
University of Texas at El Paso
El Paso, Texas, USA
mceberio@utep.edu

4th Hung T. Nguyen
Department of Mathematical Science
New Mexico State University
Las Cruces, New Mexico, USA, and
Faculty of Economics, Chiang Mai University
Chiang Mai, Thailand, hunguyen@nmsu.edu

***Abstract*—Elon Musk's successful "move fast and break things" strategy is based on the fact that in many cases, we do not need to satisfy all usual constraints to be successful. By sequentially trying smaller number of constraints, he finds the smallest number of constraints that are still needed to succeed – and using this smaller number of constrains leads to a much cheaper (and thus, more practical) design. In this strategy, Musk relies on his intuition – which, as all intuitions, sometimes works and sometimes doesn't. To replace this intuition, we propose an algorithm that minimizes the worst-case cost of finding the smallest number of constraints.**

***Index Terms*—optimal design, worst-case cost, move fast and break things, Elon Musk**

I. Formulation of the problem

General strategy behind the successes of Elon Musk. Elon Musk has had many successes in different application areas: from a successful electronic financial system Paypal to practical electric cars to effective reusable rockets for space exploration.

According to a recent semi-authorized biography of Elon Musk [3] – semi-authorized in the sense that the author was allowed to follow Musk for several years – Musk's general "move fast and break things" strategy (that has led to his many successes) is motivated by his experience that not all the constraints that are usually required for a design are actually necessary. Instead of following all the constraints — which would make the design very expensive – he tries to find the smallest number of constraints that are still necessary for the success. This minimal necessary number of constraints is usually much smaller than what is currently required. As a result, his final design – that does not have to follow all the original constraints – is drastically cheaper.

Musk's usual strategy for finding the minimal number of constraints uses the fact that usually, all the constraints can be naturally sorted in the descending order of their importance – so that the most important ones are listed first. What Musk does is he guesses the minimally necessary number of constraints, and makes a design based on these many constraints.

- If the resulting design works – e.g., if the rocket successful reaches the orbit – then he tries to see if some other constraints are not necessary.
- If the cut was too harsh – and the resulting rocket design is not successful, the rocket explodes – then he tries a design in which more original constraints are satisfied.

Comment. Of course, Elon Musk was not the first person to use intuition in decision making, use of intuition is a well known practice; see, e.g., [1], [2], [4]. What was novel in his approach is that he applied intuition to decide how many constraints to take into account, and, to the best of our knowledge, this paper is the first to formalize his idea.

Can we replace intuition in this strategy? The above-described strategy is based on intuition, on guessing the minimally necessary number of constraints. Sometimes this intuition works, sometimes it does not: for example, for a reusable rocket, the first guess, that only 10 constraints are necessary, led to an explosion. A natural question is: can we replace intuition with justified recommendations?

How can we replace intuition: an idea. The main objective is to minimize the cost of this search for this minimal number of constraints. Usually, Musk deals with a completely new area, in which there is practically no past experience, so we cannot come up with meaningful probabilities of different situations. In such cases, a natural idea is to minimize the worst-case cost of a strategy.

In finding the strategy that minimizes the cost, we need to take into account that the cost of each experiment drastically depends on whether this experiment led or a success or to a failure: for example, if a rocket blows up, the cost is much larger that when it successfully returns to Earth. So, we need to distinguish between the cost of success c and the cost of failure C – which is much larger than c.

We cannot completely avoid failures – if we do not experience a failure, how can we be sure that the number of constraints is indeed the smallest possible? But what we need

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), HRD-1834620 and HRD-2034030 (CAHSI Includes), EAR-2225395 (Center for Collective Impact in Earthquake Science C-CIES), and by the AT&T Fellowship in Information Technology. It was also supported by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

to do is to minimize the overall cost of this search, including the cost of possible failures.

What is known. To the best of our knowledge, this optimization problem has not been considered earlier.

What we do in this paper. In this paper, we present a solution to the corresponding optimization problem.

Remaining open prooblem. At this moment, what we are proposing is a theoretically justified algorithm, an algorithm based on the assumptions that the known order of constraint importance and the known cost estimates are correct. In view of these two assumptions, it makes sense to consider our algorithm as the first approximation to the optimal strategy.

It is desirable to analyze – both theoretically and practically – how to get more accurate strategies, by taking into account that the real-life situation may be somewhat different from these two assumptions: some ordering of constraint importance may be wrong, and the actual cost estimates may differ from the expert estimates.

II. Analysis of the problem and the resulting optimal algorithm

Problem: reminder. In general, we have a situation in which we have a list of N_0 constrains, and we know that following all these constraints leads to a success. Let us denote by $c(N)$ the smallest worst-case cost of a strategy that finds the minimal necessary number of constraints in a situation when only take into account the first N out of N_0 original constraints.

Proposed algorithm. Suppose that we know the values c, C, and N_0. Then, we take $c(1) = 0$, and for $N = 2, 3, \ldots, N_0$, we use the formula to sequentially compute the values

$$c(2), c(3), \ldots, c(N_0).$$

In this process, for each value $N \le N_0$, we get the value $k(N)$ that minimizes the expression

$$c(N) = \min_{0<k<N} \max(c + c(k), C + c(N-k)).$$

Then, we select $k_0 = k(N_0)$ constraints to try.

- If the experiment of satisfying only k_0 constraints was a success, this means that we need to find the smallest possible number of the selected k_0 constraints. In line with our analysis, we select $k(k_0)$ constraints to try.
- If the experiment of satisfying only $k_0 < N$ constraints was a failure, this means that need to find the smallest possible number of the remaining $N - k_0$ constraints. In line with our analysis, we select $k(N - k_0)$ additional constraints to try – in addition to the k_0 constraints that turned out to be absolutely necessary.

Numerical example. Suppose that $c = 1$, $C = 2$, and $N_0 = 4$. Here, $c(1) = 0$. Then, first, we compute

$$c(2) = \max(1 + 0, 2 + 0) = 2.$$

In this case, $k(2) = 2$. Then, we compute

$$c(3) = \min_{0<k<3} \max(c + c(k), C + c(3-k)) = 3.$$

In this case, the minimum is attained when $k = 2$, so $k(3) = 2$. After that, we compute

$$c(4) = \min_{0<k<4} \max(c + c(k), C + c(3-k)) = 4.$$

In this case, the minimum is attained when $k = 2$ or when $k = 3$, so we can select both $k(4) = 2$ or $k(4) = 3$.

If we select $k(4) = 3$, then first, we try satisfying only 3 constraints. Then:

- if this results in a success, i.e., if we know that 3 constraints are sufficient, we will then – since $k(3) = 2$ – need to check if two constraints are sufficient;
- if this results in a failure, then we know that 3 constraints are not sufficient, so the original 4 constrains is the smallest number that need to be satisfied.

Comments.

- The number of computation steps of this algorithm is equal to
$$\frac{(N_0 - 1) \cdot N_0}{2}.$$
This is quite feasible, since N_0 is usually in the dozens.
- We can compute $k(N)$ even faster if we take into account that for every c and C, there exists a constant C_0 such that for all N, we have
$$|c(N) - a \cdot \log_2(N)| \le C_0,$$
where a is the solution to the equation
$$2^{-c/a} + 2^{-C/a} = 1.$$
- The optimality and above-mentioned asymptotic property of our algorithm can be proven similarly to [5]–[8].

Acknowledgments

The authors are greatly thankful to the anonymous reviewers for valuable suggestions.

References

[1] T. Betsch and A. Glöckner, "Intuition in judgment and decision making: Extensive thinking without effort", *Psychological Inquiry*, 2010, Vol. 4, pp. 279–294.

[2] D. G. Elms and C. B. Brown, "Intuitive decisions and heuristics–an alternative rationality, *Civil Engineering and Environmental Systems*, 2013, Vol. 30, No. 3–4, pp. 274–284.

[3] W. Isaacson, *Elon Musk*, Simon & Schuster, New York, 2023.

[4] O. P. Johnny, *A Computational Model of "Artificial Intuition" in Decision Making*, Doctoral Dissertation, Edge Hill University, Ormskirk, UK, 2020.

[5] E. N. Kamoroff, *How to Extract Knowledge From an Expert so That His Effort is Minimal*, Master Project, Computer Science Department, University of Texas at El Paso, 1993.

[6] H. T. Nguyen, V. Kreinovich, and E. Kamoroff, "Asymmetric information measures: how to extract knowledge from an expert so that the expert's effort is minimal", *International Journal of Automation and Control (IJAAC)*, 2008, Vol. 2, No. 2/3, pp. 153–177.

[7] R. A. Trejo, J. Galloway, C. Sachar, V. Kreinovich, C. Baral, and L. C. Tuan, "From planning to searching for the shortest plan: an optimal transition", *Proceedings of the International Conference on Intelligent Technologies*, Bangkok, Thailand, December 13–15, 2000, pp. 17–23.

[8] R. A. Trejo, J. Galloway, C. Sachar, V. Kreinovich, C. Baral, and L. C. Tuan, "From planning to searching for the shortest plan: an optimal transition", *International Journal of Uncertainty, Fuzziness, Knowledge-Based Systems (IJUFKS)*, 2001, Vol. 9, No. 6, pp. 827–838.