# Seemingly Counter-Intuitive Features of Good-to-Great Companies Actually Make Perfect Sense: Possible Algorithmics-Based Explanations

Francisco Zapata[1], Eric Smith[1],
Olga Kosheleva[2] and Vladik Kreinovich[3]
[1]Department of Industrial, Manufacturing,
and Systems Engineering
[1]Department of Teacher Education
[2]Department of Computer Science
University of Texas at El Paso
El Paso, Texas 79968, USA
fcozpt@outlook.com, esmith2@utep.edu
olgak@utep.edu, vladik@utep.edu

## 1. How can we all be more effective?

- To become more effective, a natural idea is to learn from those who are already more effective than everyone else.
- In line with this natural idea, a group of researchers analyzed several companies that succeeded in drastically improving their performance.
- They analyzed what features these companies have in common that other less successful companies don't have.
- The results of this statistical analysis appeared in a book form.
- In this book, companies that succeeded in drastically improving their performance are called good-to-great ones.
- The book emphasized that:
  - while the analysis was limited to companies,
  - these features will probably be helpful to everyone who wants to become more effective: universities, individuals, etc.

## 2. Many of the revealed features are somewhat counter-intuitive

- The book's analysis showed that many specific features of good-to-great companies are counter-intuitive.
- This makes some sense – if these features were more intuitive, most companies would follow them and achieve similar successes.
- So, natural questions are:
  - Why these seemingly counter-intuitive features help the companies succeed?
  - How can we explain this empirical phenomenon?

## 3. What we do in this talk

- In this talk, we provide such an explanation for many of these features, features that we perceived as the most important ones.
- We list these features in the order in which these features are important on different stages of company development.
- Each of these features is analyzed in a separate "section":
  - we start each such section with describing the feature;
  - then we explain why this feature seems counter-intuitive; and
  - finally, we provide a possible explanation of why this seemingly counter-intuitive feature has led to successes.
- Of course, the book contains many more features than we cover here.
- We tried our best to select the main ones, but other features may be important too.
- So we encourage others to seek explanations for other features as well.

## 4. First Who Then What

- In all the analyzed successful companies, their motion from good to great started:
  - not with selecting the application area,
  - but by selecting a team: first who then what.
- Selecting right people is more important than selecting a domain or selecting a proper technology.
- Technology can enhance or inhibit the success but it is secondary in predicting which companies will become great.
- This sounds counter-intuitive because common sense tells us that:
  - to succeed in some task,
  - we need to have people who are specialists in this task.
- How can we decide who to invite to the team if we have not yet decided what the task will be?

## 5. First Who Then What (cont-d)

- One of the common criticisms of management is that often, managers move, e.g., from soft drinks to car manufacturing.
- The fact that they do not know much about the new area makes them much less effective.
- On the university level, you do *not*:
  - first hire people with high IQ and
  - then decide whether to make them Department of Physics or Department of Computer Science.

## 6. Why this feature is actually important for success: our explanation

- One of the great achievements in theoretical computer science is the not-well-publicized result that:
  - all well-formulated complex problems
  - are, in effect, equivalent to each other.
- Let us explain it in detail.
- A well-formulated problem is a problem in which it is easy (or at least feasible) to check whether we have solved this problem or not.
- In precise terms, there exists a feasible algorithm that:
  - given a candidate for a solution,
  - checks whether this candidate is indeed a solution.
- In business terms, if the goal is to increase profit by 10%, then it is easy to check whether this result was achieved.

## 7. Why this feature is actually important for success: our explanation (cont-d)

- In mathematical terms, if the task is to solve an equation or a system of equations, then:
  - if someone gives us a candidate for a solution,
  - we can plug in this candidate into the equation and check whether the equation is satisfied.
- The class of all such problems is usually denoted by NP.
- Some problems are *not* in the class NP.

## 8. Why this feature is actually important for success: our explanation (cont-d)

- E.g., if the goal is optimization – such as finding the optimal trajectory for a robot,
  - and someone gives us a candidate solution,
  - then, in many cases, the only way to check that this candidate is indeed optimal is to compare it with all other possible solutions,
  - and there is usually an astronomical number of possible solutions, which makes such checking unfeasible.
- Some problems from the class NP are more difficult than others.
- In particular:
  - if every instance of the general problem $A$ can be feasibly reduced to an appropriate instance of a general problem $B$,
  - this means that $B$ is more difficult than $A$ (or at least of the same difficulty).

## 9. Why this feature is actually important for success: our explanation (cont-d)

- Indeed, in this case:
  - if we have a feasible algorithm for solving the problem $B$,
  - then this reduction automatically gives us a feasible algorithm for solving the problem $A$.
- The breakthrough result that we mentioned is that:
  - in the class NP,
  - there are problems to which every other problem from the class NP can be reduced.
- Such problem are known as *NP-complete.*
- These are the problems which are the most complex.
- It is believed that no feasible algorithm is possible that would solve *all* the instances of an NP-complete problem.

## 10. Why this feature is actually important for success: our explanation (cont-d)

- No proof of this result is known.
- This is known as P$\neq$NP hypothesis – one of the main open problems in computing.
- Thus, all a feasible algorithm can do is solve *some* instances of this problem.
- A consequence of this result is that all such complex problems can be reduced to each other.
- From this viewpoint, it does not matter which of these problems we are solving:
  - if we have a feasible algorithm for solving some instances of one NP-complete problem,
  - then for every other problem from the class NP, we automatically get a feasible algorithm for solving some of its instances.

## 11. Why this feature is actually important for success: our explanation (cont-d)

- In effect, what every person on a team does is solves problems.
- From the above viewpoint, the efficiency of a person does not depend on which problem he/she has been solving.
- What is important is how deep are the resulting algorithms, how many instance they cover.
- This is exactly what the "first who then what" principle suggests.
- We select people based on their abilities *before* we decide what exactly problems these people will be solving.
- This also explains why the selection of technology is secondary:
  - technology can speed up solutions,
  - but available technology is based on already known algorithms.
- To solve new instances of an NP-complete problem, we need to come up with new algorithms.

## 12. Be the Best in the World

- Once the team is assembled, it is important to select the area.
- Another common feature of good-to-great companies is that:
  - they select the area in which they can have the potential be the first (or the second best) in the world,
  - ignoring all other possible areas – selling the corresponding divisions, dismissing them, making explicit "not to do" lists.
- This also sounds counter-intuitive.
- For example, in science, it is good to aspire to be an Einstein, but someone need to do the mundane work.
- Actually:
  - for Einsteins to have enough data to generate their genius ideas,
  - we need a large number of people who perform mundane work and come up with interesting experimental results.

## 13. Why this feature is actually important for success: our explanation

- The appearance of this feature is relatively easy to explain.
- It does not require any complex results – as the above "first who then what" feature.
- Indeed, we want the humanity as a whole to be the most productive.
- This means that each task should be assigned to the person who is the best in the world – i.e., the most efficient – in this task:
  - Einsteins – the best in fundamental theoretical physics – should be given a task of explaining the experimental results.
  - The task of setting up these experiments should be given to folks who are the best in the world in experimental physics.
  - And the task of designing their shoes should be given to those who are the best in the world in designing shoes.

## 14. Confront the Brutal Facts

- In all good-to-great companies, all problems, all limitations are known.
- In particular, they are well known to people in charge.
- Why this feature seems counter-intuitive?
- Usually, we try to concentrate on the positive.
- There are sometimes team members that dig up lots of bad things, do we like them?
- Do we really want these negative people in our teams? Not really.
- However, surprisingly, this is one of the important features that makes good companies great.
- It may be one of the reasons why few companies follow this route.

## 15. Why this feature is actually important for success: our explanation

- From the viewpoint of general decision making, this feature is also easy to explain:
  - to decide how to improve the company's performance, how to move the company (or any other system) from its current state to a better future state,
  - we need to have a good understanding of where we are right now.
- This includes knowing:
  - not only all the good things about the current state,
  - but also all the bad things, all the things that need improvement.

## 16. Difference in Opinions

- In all the good-to-great companies, the governing board show great difference in opinions.
- Every decision is made only after a heated discussion.
- Why this feature seems counter-intuitive?
- At first glance, these discussions slow down the progress.
- We have all witnessed situations in which lengthy discussions unnecessarily postpone decisions that need to be urgently made, be it:
  - on the government level or
  - on the level of a department.

## 17. Why this feature is actually important for success: our explanation

- Difference in opinions usually comes from difference in viewpoint.
- As we have mentioned in the previous section:
  - to make a right decision,
  - we need to have a good understanding of the current state.
- In particular, this means that:
  - for all difficult-to-measure-directly quantities characterizing the system,
  - we need to have estimates which are as accurate as possible.

## 18. Why this feature is actually important for success: our explanation (cont-d)

- It is well known that:
  - if we have several independent estimates $x_1, \ldots, x_n$ of the same quantity,
  - then the arithmetic average $\overline{x} = \dfrac{x_1 + \ldots + x_n}{n}$ of these estimates is more accurate than any of them.
- For example:
  - if all estimates are unbiased, and the mean square error of each estimate is $\sigma$,
  - then the mean square error of the arithmetic average is equal to

$$\frac{\sigma}{\sqrt{n}}.$$

## 19. Ignore Competition

- In many organizations, competition is important in day-by-day decisions:
  - companies aspire to overcome their competition,
  - universities try their best to get a higher ranking, i.e., to overcome those who are currently ranked higher, etc.
- Interestingly, in good-to-great companies, beating competition never plays an important role in decision making.
- Why this feature seems counter-intuitive, since usually:
  - if a competitor comes up with a move that brings it an advantage,
  - it seems like a natural idea is to immediately react – otherwise, we may lose.
- If Airbus designs a new plane, Boeing immediately start thinking on how to reply to this challenge.
- How can we ignore the competition?

## 20. Why this feature is actually important for success: our explanation

- No matter what objective function we select, we want to optimize this objective function.
- Optimization is difficult.
- There are many feasible optimization algorithms, but in many cases, they lead to a local optimum.
- Once we are in a local optimum, we need to get out of it and try to get better results.
- At this stage, we are doing worse that we ourselves did before.
- Also, unless the competitor does the same – we are doing worse that the competitor.
- However, ultimately we will prevail if we follow the right optimization strategy.

## 21. Why this feature is actually important for success: our explanation (cont-d)

- Good optimization algorithms in the beginning, perform worse than simple strategies such as a straightforward gradient descent.
- However, in the long run, they are known to perform much better.

## 22. Core Values

- An important feature of all good-to-great companies is that they all have a set of core values.
- These core values serve as severe constraints on all possible decisions that these companies make.
- These constraints are different for different companies, since the book's list of good-to-great companies include both:
  - companies aiming at improving people's health and
  - companies that specialize in tobacco products.
- However, each good-to-great company has its own list of such constraints.
- Why this feature seems counter-intuitive?
- Our goal is to optimize some objective function – e.g., we want to optimize profit.

## 23. Core Values (cont-d)

- So why would we want to limit our options by imposing additional constraints and thus, possibly missing great solutions?
- There are already many legal constraints making sure that our products do not harm people, do not harm environment, etc.
- Why do we want to impose additional constraints of this type on ourselves?
- This is especially strange since the competition is not necessarily bound by these additional constraints.

## 24. Why this feature is actually important for success: our explanation

- This feature is not so easy to explain, but it can be explained by another not-well-known result.
- Namely, it is known that in many situations:
  - while there is no algorithm for solving all possible instance of a problem (e.g., of an optimization problem),
  - there *are* algorithms that solve all the instances in which the problem has a unique solution, and
  - no general algorithm is possible even for cases when the problem has two or more solutions.
- So, to make a problem algorithmically easier to solve, we need to decrease the number of possible solutions – ideally to one.
- A natural way to decrease the number of possible solution is to impose additional constraints.

## 25. Why this feature is actually important for success: our explanation (cont-d)

- The simplest example is that the equation $x^2 = a$ has two solutions $x = -\sqrt{a}$ and $x = \sqrt{a}$.
- If we restrict ourselves to non-negative solutions, we get only one solution.
- This is also the idea behind regularization techniques.
- For example, if we want to de-noise a noisy image, there are usually many images which are consistent with the noisy observations.
- To select a unique image, we impose natural restrictions; e.g.:
  - we require that the image be sufficiently smooth
  - or, for an image of a starry sky, we require that the image consists only of a small number of point-wise objects.

## 26. Why this feature is actually important for success: our explanation (cont-d)

- In all these cases, adding constraints decreases the number of possible solutions:
  - if we impose exactly as many constraints as to make the remaining solution unique
  - we thus make it algorithmically easier to find a solution.

## 27. Acknowledgments


- 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), and
  - HRD-1834620 and HRD-2034030 (CAHSI Includes).
- It was also supported by the AT&T Fellowship in Information Technology.
- It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478.