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

1st Francisco Zapata
Department of Industrial, Manufacturing, and Systems Engineering
University of Texas at El Paso
El Paso, Texas 79968, USA
fcozpt@outlook.com

2nd Eric Smith
Department of Industrial, Manufacturing, and Systems Engineering
University of Texas at El Paso
El Paso, Texas 79968, USA
esmith2@utep.edu

3rd Olga Kosheleva
Department of Teacher Education
University of Texas at El Paso
El Paso, Texas 79968, USA
olgak@utep.edu

4th Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
El Paso, Texas 79968, USA
vladik@utep.edu

***Abstract*—In the late 1990s, researchers analyzed what distinguishes great companies from simply good ones. They found several features that are typical for great companies. Interestingly, most of these features seem counter-intuitive. In this paper, we show that from the algorithmic viewpoint, many of these features make perfect sense. Some of the resulting explanations are simple and straightforward, other explanations rely on complex not-well-publicized results from theoretical computer science.**

***Index Terms*—Good to great companies, algorithmics, seemingly counter-intuitive ideas**

I. Formulation of the Problem

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 – what features these companies have in common that other less successful companies don't have. The results of this analysis appeared in a book form [1]. 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 most probably be helpful to everyone who wants to become more effective, be it companies, universities, or even individuals.

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), and 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, and by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

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, a natural questions is:

- Why these seemingly counter-intuitive features help the companies succeed?
- How can we explain this empirical phenomenon?

What we do in this paper. In this paper, 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 "chronological" order, namely, 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.

Comment. 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.

II. First Who Then What

Description of the feature. 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.

Why this feature seems counter-intuitive. 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?

One of the common criticisms of management is that often, managers move from one area to another one – e.g., from soft drinks to car manufacturing – and 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.

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; see, e.g., [14], [20]. 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, i.e., in precise terms, for which 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.
- 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: 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 difficult). 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 – although no proof is known yet – that no feasible algorithm is possible that would solve *all* the instances of an NP-complete problem. This is known as P≠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 (in particular, for every other NP-complete problem), we automatically get a feasible algorithm for solving some of its instances.

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,
- and to solve more and more instances of an NP-complete problem, beyond what people are already doing, we need to come up with new algorithms.

III. Be the Best in the World

Description of the feature. 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.

Why this feature seems counter-intuitive. 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, we need a large number of people who perform mundane work and come up with interesting experimental results for Einsteins to have enough data to generate their genius ideas.

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 performing this task:

- Einsteins – the best in the world 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.

IV. Confront the Brutal Facts

Description of the feature. 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 – but, surprisingly, this is one of the important features that makes good companies great – and maybe one of the reasons why few companies follow this route.

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.

V. Difference in Opinions

Description of the feature. 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.

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.

It is well known (see, e.g., [21]) that:

- if we have several independent estimates $x_1, \ldots, x_n$ of the same quantity,
- then the arithmetic average
$$\overline{x} = \frac{x_1 + \ldots + x_n}{n}$$
of these estimates is more accurate than any of them.

For example:

- all estimates are unbiased, and the mean square error of each estimate is σ,
- then the mean square error of the arithmetic average is equal to
$$\frac{\sigma}{\sqrt{n}}.$$

VI. Ignore Competition

Description of the feature. 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. In a competitive world, 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?

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, and – 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.

Good optimization algorithms in the beginning, perform worse than simple strategies such as a straightforward gradient descent, but in the long run, they perform much better; see, e.g., [19].

VII. Core Values

Description of the feature. 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. If our goal is to optimize some objective function – e.g., if we want to optimize profit – 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 – especially since the competition is not necessarily bound by these additional constraints?

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 (see, e.g., [2]–[4], [6], [8], [14]–[17]), and
- no general algorithm is possible even for cases when the problem has two or more solutions; see, e.g., [5]–[10], [13], [14], [18].

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. The simplest example if that a quadratic equation has, in general two solutions.

For example, 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 ith 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, vice versa, we require that the image consists only of a relatively small number of practically point-wise objects.

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

Acknowledgments

The authors are greatly thankful to Janusz Kacprzyk and Aaron Velasco for their valuable discussions.

References

[1] J. Collins, *Good to Great: Why Some Companies Make the Leap ... And Others Don't*, HarperCollin, New York, 2001.

[2] U. Kohlenbach. *Theorie der majorisierbaren und stetigen Funktionale und ihre Anwendung bei der Extraktion von Schranken aus inkonstruktiven Beweisen: Effektive Eindeutigkeitsmodule bei besten Approximationen aus ineffektiven Eindeutigkeitsbeweisen*, Ph.D. Dissertation, Frankfurt am Main, 1990.

[3] U. Kohlenbach, Effective moduli from ineffective uniqueness proofs. An unwinding of de La Vallée Poussin's proof for Chebycheff approximation, *Annals for Pure and Applied Logic*, 1993, Vol. 64, No. 1, pp. 27–94.

[4] U. Kohlenbach, *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*, Springer Verlag, Berlin-Heidelberg, 2008.

[5] V. Kreinovich, *Complexity measures: computability and applications*, Master Thesis, Leningrad University, Department of Mathematics, Division of Mathematical Logic and Constructive Mathematics, 1974 (in Russian).

[6] V. Kreinovich, Uniqueness implies algorithmic computability, *Proceedings of the 4th Student Mathematical Conference*, Leningrad University, Leningrad, 1975, pp. 19–21 (in Russian).

[7] V. Kreinovich, Reviewer's remarks in a review of D. S. Bridges, *Constrictive functional analysis*, Pitman, London, 1979; *Zentralblatt für Mathematik*, 1979, Vol. 401, pp. 22–24.

[8] V. Kreinovich, *Categories of space-time models*, Ph.D. dissertation, Novosibirsk, Soviet Academy of Sciences, Siberian Branch, Institute of Mathematics, 1979 (in Russian).

[9] V. Kreinovich, "Unsolvability of several algorithmically solvable analytical problems", *Abstracts Amer. Math. Soc.*, 1980, Vol. 1, No. 1, p. 174.

[10] V. Kreinovich, *Philosophy of Optimism: Notes on the possibility of using algorithm theory when describing historical processes*, Leningrad Center for New Information Technology "Informatika", Technical Report, Leningrad, 1989 (in Russian).

[11] V. Kreinovich, "Physics-motivated ideas for extracting efficient bounds (and algorithms) from classical proofs: beyond local compactness, beyond uniqueness", *Abstracts of the Conference on the Methods of Proof Theory in Mathematics*, Max-Planck Institut für Mathematik, Bonn, Germany, June 3–10, 2007, p. 8.

[12] V. Kreinovich, G. P. Dimuro, and A. C. da Rocha Costa, *From Intervals to? Towards a General Description of Validated Uncertainty*, Catholic University of Pelotas, Brazil, Technical Report, January 2004.

[13] V. Kreinovich and R. B. Kearfott, "Computational complexity of optimization and nonlinear equations with interval data", *Abstracts of the Sixteenth Symposium on Mathematical Programming with Data Perturbation*, The George Washington University, Washington, D.C., May 26–27, 1994.

[14] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1998.

[15] V. Kreinovich and K. Villaverde, "Extracting computable bounds (and algorithms) from classical existence proofs: girard domains enable us to go beyond local compactness", *International Journal of Intelligent Technologies and Applied Statistics (IJITAS)*, 2019, Vol. 12, No. 2, pp. 99–134.

[16] D. Lacombe, "Les ensembles récursivement ouvert ou fermés, et leurs applications à l'analyse récurslve", *Compt Rend.*, 1957, Vol. 245, No. 13, pp. 1040–1043.

[17] V. A. Lifschitz, "Investigation of constructive functions by the method of fillings", *J. Soviet Math.*, 1973, Vol. 1, pp. 41–47.

[18] L. Longpré, V. Kreinovich, W. Gasarch, and G. W. Walster, "m solutions good, $m-1$ solutions better", *Applied Math. Sciences*, 2008, Vol. 2, No. 5, pp. 223–239.

[19] G. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 2006.

[20] C. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, Massachusetts, 1994.

[21] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.