



Moshe Y. Vardi

DOI:10.1145/3388890

Efficiency vs. Resilience: What COVID-19 Teaches Computing

AS I AM writing these lines, in mid-March 2020, COVID-19, the disease caused by the coronavirus virus, is spreading around the world. From a local epidemic that broke out in China in late 2019, the disease has turned into a raging pandemic the likes of which the world has not seen since the 1918 Spanish Flu Pandemic. Thousands have already died, and the ultimate death toll may be in the millions. Attempting to mitigate the pandemic, individuals are curtailing travel, entertainment, and more, as well as exercising “social distancing,” thus causing an economic slowdown. Businesses hoard cash and cut spending in order to survive a slowdown of uncertain duration. These rational actions by individuals and businesses are pushing the global economy into a recession.

Observing the economic consequences of this unexpected crisis, William A. Galston asks in a recent *Wall Street Journal*^a column: “What if the relentless pursuit of efficiency, which has dominated American business thinking for decades, has made the global economic system more vulnerable to shocks?” He goes on to argue that there is a trade-off between efficiency and resilience. “Efficiency comes through optimal adaptation to an existing environment,” he argues, “while resilience requires the capacity to adapt to disruptive changes in the environment.”

A similar point, in a different setting, was made by Adi Livnat and Christos Papadimitriou in their 2016 *Communications*^b article, “Sex as an Algorithm.” Computational experience has shown that Simulated Annealing, which is a lo-

cal search—via a sequence of small mutations—for an optimal solution, is, in general, superior computationally to genetic algorithms, which mimic sexual reproduction and natural selection. Why then has nature chosen sexual reproduction as the only reproduction mechanism in animals? Livnat and Papadimitriou’s answer is that sex as an algorithm offers advantages other than good performance in terms of approximating the optimum solution. In particular, sexual reproduction favors genes that work well with a greater diversity of other genes, and this makes the species more adaptable to disruptive environmental changes, that is to say, more resilient.

And yet, we have educated generations of computer scientists on the paradigm that analysis of algorithm only means analyzing their computational efficiency. As Wikipedia states: “In computer science, the analysis of algorithms is the process of finding the computational complexity of algorithms—the amount of time, storage, or other resources needed to execute them.” In other words, efficiency is the *sole* concern in the design of algorithms. (Of course, the algorithm has to meet its intended functionality). What about resilience? Quoting Galton again: “Creating resilient systems means thinking hard in advance about what could go wrong and incorporating effective countermeasures into designs.” How can we make our algorithms more resilient?

Of course, fault tolerance has been part of the canon of computing-system building for decades. Jim Gray’s 1998 Turing Award citation refers to his invention of transactions as a mechanism to provide crash resilience to databases. Leslie Lamport’s 2013 Turing Award citation refers to his work on fault tolerance in distributed systems. Neverthe-

less, I believe that computer science has yet to internalize the idea that resilience, which to me include fault tolerance, security, and more, must be pushed down to the algorithmic level. Case in point is search-result ranking. Google’s original ranking algorithm was PageRank, which works by counting the number and quality of links to a page to determine how important the website is. But PageRank is not resilient to manipulation, hence “search-engine optimization.” Today’s result-ranking algorithms are well-kept trade secrets. Indeed, adversarial machine learning, which looks at the impact of maliciously manipulated data on machine learning, is a highly active research area.

As I pointed out in an earlier column, this quest for efficiency has been single minded: “Our discipline is dedicated to reducing friction. Latency must be eliminated, bandwidth must increase, and ubiquity should be universal. Our goal is to reduce the friction of computing and communication as much as possible.” Facebook’s CEO Mark Zuckerberg speaks of “frictionless sharing” as a goal. This reduction of friction has enabled the amazing world of the Internet and the Web we have created over the past 50 years. It also provides us today with tools that enables us to work and socialize under social distancing. But we now know the imagined utopia of frictionless sharing on social media leads to filter bubbles, fake news, and extreme content.

Computing today is the “operating system” of human civilization. As computing professionals we have the awesome responsibility as the developers and maintainers of this operating system. We must recognize the trade-off between efficiency and resilience. It is time to develop the discipline of resilient algorithms. □

a <https://on.wsj.com/3b82zcl>

b <http://bit.ly/2J0JJaQ/>