



Ubiquity Symposium

Evolutionary Computation and the Processes of Life

The Emperor is Naked: Evolutionary algorithms for real-world applications

by Zbigniew Michalewicz

Editor's Introduction

During the past 35 years the evolutionary computation research community has been studying properties of evolutionary algorithms. Many claims have been made—these varied from a promise of developing an automatic programming methodology to solving virtually any optimization problem (as some evolutionary algorithms are problem independent). However, the most important claim was related to applicability of evolutionary algorithms to solving very complex business problems, i.e. problems, where other techniques failed.

So it might be worthwhile to revisit this claim and to search for evolutionary algorithm-based software applications, which were accepted by businesses and industries. In this article Zbigniew Michalewicz attempts to identify reasons for the mismatch between the efforts of hundreds of researchers who make substantial contribution to the field of evolutionary computation and the number of real-world applications, which are based on concepts of evolutionary algorithms.

Editor

Ubiquity Symposium

Evolutionary Computation and the Processes of Life

The Emperor is Naked: Evolutionary algorithms for real-world applications

by Zbigniew Michalewicz

The evolutionary computation (EC) community has been making claims that their methods are ideal for hard problems—problems, where other methods usually fail. Hans-Paul Schwefel illustrated this point by saying—in many presentations—evolutionary algorithms (EA) are indeed the last-resort methods. His favorite example was that one must be crazy to use EAs for minimizing easy functions, say, x^2 , as other (derivative-based) methods would be more appropriate. On the other hand, for a highly nonlinear and/or discontinuous function, EAs are often the methods of choice—the harder the problem, the better. Clearly, most real-world problems are very hard and complex, with nonlinearities and discontinuities; hence EAs should provide a great benefit to the real-world community.

However, after 35 years of research, tens of thousands of papers written on evolutionary algorithms, dedicated conferences (e.g. GECCO, IEEE CEC, PPSN), dedicated journals (e.g. *Evolutionary Computation Journal*, *IEEE Transactions on Evolutionary Computation*), special sessions and special tracks at most AI-related conferences, special sessions on real-world applications, etc., it is still not that easy to find EA applications in the real world.

Over the last several years I stayed in touch with (and visited in person) a few academic organizations that run “special units” for real-world applications and I discovered that none of these “real-world applications” are really real-world applications. Such application-oriented units often publish their statements on their [websites](#), stating, for example, that they “...research into models, heuristics and algorithms for automatically producing high-quality solutions to a variety of real-world applications and optimisation problems, including scheduling, timetabling, manufacturing, logistics, space allocation, stock cutting, anomaly

detection, bioinformatics and co-operative decision support.” Some other claim that “...we have the solution to almost any complex business issue” (!). The language varies; the areas of research vary (they include also target marketing, fraud detection, customer retention, price optimization, product development, medical applications, to name a few), but the focus is the same: real-world applications. However, usually after many years in operation, none of their applications are in real use in a real-world environment. Even if it had been used for a while, it was usually a tool for a single user, was not integrated with other software systems and applications (e.g. SAP, JD Edwards), and usually disappeared some months later— as they would not constitute the core system for an organization. Each such tool was unique (i.e. custom built), so that the same application was never used in two or more organizations—this limited further their usefulness. At the very best, these applications served as tools (toys) for single individuals within organizations and become obsolete soon after delivery (often due to changes in business rules of the organization).

Further, from 2009 to 2010, together with Raymond Chiong and Thomas Weisse, we edited a book *Variants of evolutionary algorithms for Real-World Applications* (Springer, 2011). Our call-for-chapters announcement generated over 40 chapter proposals, which have been grouped into four categories:

- I. An application is described which is used in some business/industry on a daily (regular) basis.
- II. An application is described which is tested on real data (e.g. taken from a hospital, city council, a particular business unit).
- III. An application is described which is tested on some known model (e.g. TSP, VRP, JSSP) of a real-world problem.
- IV. Other applications (e.g. numerical optimization, constraint-handling, multi-objective optimization).

However, the first category was empty...

Last year, with Frank Neumann and Andreas Ernst, we placed a call for papers for the special issue on “Heuristic Search Methods for Large Scale Optimisation Problems in Industry” for the *Evolutionary Computation Journal*. This time, in the call for papers we inserted the following phrase: “This special issue solicits novel high-quality contributions on heuristic methods for large applied optimisation problems **that have been used in practice**.” Anyone can guess, how many papers we received—the idea of such special issue was finally abandoned.

Recently, I have exchanged a few emails with J. David Schaffer—as some of you may remember, he got his Ph.D. in genetic algorithms in the 1980s, he was the first person to address the issue of multi-objective optimization by genetic algorithms (he proposed the VEGA system), and he made several substantial contributions to the field of evolutionary computation. Last year he retired from Philips Labs in NY where he worked for 25 years or so. In our correspondence, he wrote: “Larry [Eshelman] and I had only one really successful application [of genetic algorithms] at Philips in 25 years...” I have been also exchanging emails with Darrell Whitley (former editor-in-chief of *Evolutionary Computation Journal*), who wrote: “Your comments are one reason that I started looking for real-world applications of EA that **are** used on a daily basis... .” My understanding is that his search resulted only in a handful of examples.

All this is really amazing, as at most EA conferences there are always special sessions on “real-world applications” (not to mention special issues of journals and edited volumes). It seems that the main problem is in terminology, and all these “real-world applications” are like “Guinea pigs”—not really from Guinea, and not really pigs. It seems researchers are interpreting the phrase “real-world application” in a very arbitrary way. So it is important to distinguish between “solvers” or “tools” and software applications. Solvers and tools are quite useful (and there are many EA-based solvers available), but they have very limited usefulness and limited audience. Some individuals in some companies use solvers that are based on modern heuristic methods (and evolutionary algorithms in particular) for solving some small-scale optimization problems; however, due to the limited usefulness of such tools I heard quite often negative statements from researchers employed in major companies, e.g. “I tried a genetic algorithm on my problem, but I did not get any results...they do not work.” There are of course papers describing some interesting applications (e.g. applications of covariance matrix adaptation evolution strategies, CMA-ES; see <http://www.lri.fr/~hansen/cmaapplications.pdf>). However, many of these are not real-world applications. They are usually restricted to continuous variables only, and they hardly address complex business problems. What the evolutionary community needs (I think) are a few real-world success stories; for example, the emergence of an EA-based enterprise¹ software application, which dominates some niche market,² would be a great step toward gaining real recognition in business and industry.³ Of course, universities

¹ Enterprise software addresses the needs of organization processes and data flow, often in a large distributed environment (e.g. supply-chain management software).

² A niche market is the subset of the market on which a specific product is focusing, so the market niche defines the specific product features, which aimed at satisfying specific market needs.

³ For an example description of such application, see Ibrahimov M., Mohais, A., Schellenberg, S., and Michalewicz Z., *Advanced planning in vertically integrated supply chains*, chapter in *Intelligent Decision Systems in Large-Scale Distributed Environments*, P. Bouvry, H. González-Vèlez, and J. Kołodziej (Editors), Springer, 2011.

are not really in a position to develop and maintain large-scale real-world applications (for similar reasons that are applicable to small companies, which is discussed below), but they should be able to assist in the process. There are some ways the academic evolutionary computation community can make their approach more visible.

After more than 35 years of extensive research in the area of evolutionary algorithms such EA-based enterprise software applications have not yet emerged. So it might be worthwhile to ask the question: Why is it so?

Why Is It So?

It seems there are several reasons for this significant mismatch between the efforts of hundreds of researchers who have been making substantial contribution to the field of evolutionary computation over many years and the number of real-world applications, which are based on concepts of evolutionary algorithms. These reasons include:

1. Experiments focus on silo problems
2. Experiments focus on global optima
3. Theory does not support practice
4. General dislike of business issues in research community
5. Limitations of EA-based companies
6. Dominance of operation research (OR) methods in industry

We discuss these points briefly in turn.

1. Experiments focus on silo problems. It seems most experimental research efforts have been directed toward small-scale silo problems. There are hundreds (if not thousands) of research papers addressing traveling salesman problems, job shop and other scheduling problems; transportation problems; inventory problems; stock cutting problems; packing problems; and various logistic problems, to name a few. While most of these problems are NP-hard and clearly deserve research efforts, it is not exactly what the real-world community needs. Most companies run complex operations and they need solutions for problems of much higher complexity (e.g. multi-silo problems). Note, for example, that the issue of scheduling production lines (e.g. maximizing the efficiency or minimizing the cost) has direct relationships with inventory costs, stock-safety levels, replenishments strategies, transportation costs, delivery-in-full-on-time (DIFOT) to customers, etc., so it should not be considered in isolation.

Moreover, optimizing one silo of the operation may have negative impact on upstream and/or downstream silos.

For example, at PPSN'10 I gave a keynote talk on a powerful EA-based enterprise⁴ software application that was developed recently⁵ to address many of wine production challenges present in different parts of the wine supply chain. This enterprise software application for wine industries consists of a suite of software modules (these include predictive modeling for grape maturity; using weather forecasts and readings on Baume, PH, and titratable acidity⁶; vintage planning; crush scheduling; tank farm optimization, bottling-line sequencing; and demand planning) that can optimize the end-to-end wine supply chain. When deployed together, these applications can optimize all planning and scheduling activities across a winery's entire supply chain.

Each of these modules represents a significant large-scale optimization/prediction problem in itself and includes several interesting research aspects (e.g. variable constraints in the bottling module). However, finding the optimal solution in one silo of the operation may have negative impact downstream and/or upstream of the whole operation. Consider, for example, the following example. A bottling plant has to process 700,000 liters of Jacob's Creek⁷ to satisfy demand. However, the closest (in terms of volume) tank on the tank farm contains 800,000 liters of Jacob's Creek—and what is the optimal decision here? Clearly, from the bottling perspective they should get only 700,000 liters of that wine and process it, but it would be a bad decision from the perspective of another silo: the tank farm. They will be left with 100,000 liters leftover with all risks (the quality of these 100,000 liters quickly deteriorate due to oxygen levels in the tank) and inefficiencies in operations (the tank with the leftover 100,000 liters can't be used in their operation). From tank farm perspective, the "optimal" decision would be to send 800,000 liters of Jacob's Creek for bottling (no leftovers with clear benefits), but the bottling operation would not be happy with such a solution. (Note that once a wine is bottled and labeled, the choice for its destination is quite limited as different customers have different requirements for packaging.) This is why the global optimum here should consider implications of various decisions across all silos. Further, only by considering the entire supply chain we can

⁴ Enterprise software addresses the needs of organization processes and data flow, often in a large distributed environment (e.g. supply-chain management software).

⁵ The application was developed at SolveIT Software (www.solveitsoftware.com) together with similar applications in other verticals (e.g. for grain handling, pit-to-port mining logistics, and mine-planning activities).

⁶ Viti-Notes, 2005. <http://www.crcv.com.au/viticare/vitinotes/Viti-Notes/winegrape%20assessment/Grape%20Maturity%201.%20Total%20soluble%20solids,%20pH%20and%20titratable%20acidity.pdf>

⁷ Jacob's Creek is one of the most known, trusted and enjoyed Australian wines around the world.

attempt to answer some key (global) questions, as “What is the impact of increasing demand by 10 percent of Jacob’s Creek across all operations”? And the answers for such questions are sought today by businesses and industries.

Thus businesses usually need “global solutions” for their operations, not silo solutions. The OR community recognized this more than 30 years ago; in 1979 R. Ackoff wrote: “Problems require holistic treatment. They cannot be treated effectively by decomposing them analytically into separate problems to which optimal solutions are sought.”⁸ However, there are very few EA-based research efforts that aim in that direction, mainly due to the lack of benchmarks or test cases available. It is also much harder to work with a company on such global level as a delivery of successful software solution usually involves many other (apart from optimization) skills, from understanding the company’s internal processes to complex software engineering issues.

2. Experiments focus on global optima. A large portion of the research on evolutionary algorithms is experimental—hence the researchers use a variety of benchmarks and test functions. Almost in all cases a new method (whether a new representation, a new set of operators, a new selection method, a novel way to incorporate problem-specific knowledge into the algorithm, a novel way to adapt parameters in the algorithm, to name a few) is tested against such accepted benchmarks—and the most popular quality measure of a new method is the closeness of the generated solution to the known, global optima in a number of function evaluations. Of course, usually some silo problems are used for testing, as many of these the global optima are known (or it is possible to estimate their values). However, for many real-world applications the issue of getting to the global optima is secondary. First, the concept of global optima in business is different to that in academia. A global optimum for a silo is referred to as a local optimum solution, as it does not take into account other interacting silos of the business. And the global optimum solution in business refers to whole multi-silo operation of the organization. Second, for large-scale (e.g. multi-silo) problems, it would take days (if not more) to generate a global optimum solution, while decision-makers have minutes to react. Third, because of delays, unexpected orders, failures, etc. the multi-silo environment is highly variable (dynamic). Robust, quality solutions are of higher importance, as the current solution would be modified, anyway, due to changes in the environment. Fourth, due to many—possibly conflicting—objectives, business rules, and soft constraints, the meaning of the term “global optimum” is not that clear. Even experienced decision makers often have difficulties in pointing to a better solution out of two available solutions. Finally, the name of the game in industry is not to find an elusive global optimum, but rather to match (and hopefully improve)

⁸ R. Ackoff, The Future of OR is Past, *JORS*, 1979.

the results of the human team of experts,⁹ who have been involved in particular decision-making activity for a significant time. However, many researchers reject such comparisons (we return to this point later). Further, it would be extremely hard to document such comparisons in a systematic way without revealing sensitive data of an organization.

3. Theory does not support practice. At the Workshop on evolutionary algorithms, organized by the Institute for Mathematics and Its Applications, University of Minnesota, Minneapolis, Minnesota, October 21 – 25, 1996, one of the invited speakers, Dave Davis made an interesting claim. As the most recognized practitioner of evolutionary algorithms at that time he said all theoretical results in the area of evolutionary algorithms were of no use to him—actually, his claim was a bit stronger. He said there was one use only of such results: If a theoretical result indicated that, say, the best value of some parameter was such-and-such, he would never use the recommended value in any real-world implementation of evolutionary algorithm! Clearly, there was—in his opinion—a significant gap between theory and practice of evolutionary algorithms.

It seems that 15 years later we are in the same spot; the theory of evolutionary algorithms focused on convergence properties, diversity, exploration, exploitation, ruggedness of the landscape, deceptiveness, epistasis, pleiotropy, among others—and all these are researched on classic cases of artificial landscapes. Indeed, there was a progress on topics essential for real-world applications, like constraint-handling methods, handling noise and robustness, or multi-objective optimization; however, these were usually tested on simple silo problems or standard sets of numerical functions. These theoretical results are of little help (if any) to any practitioner who works on EA-based enterprise software applications. The issue of a growing gap between theory and practice of evolutionary algorithms is a separate topic for discussion.¹⁰

4. General dislike of business issues in computer science community. It seems that the term “business” is regarded as pejorative in computer science community in general, and in evolutionary computation community in particular. This personal observation is based on many conversations I had over the last 15 years and on many “experiences.” Let’s illustrate this by providing a few examples.

⁹ There is a nice, one sentence summary of how evolution works: “You don’t have to outrun the bear, but you just have to outrun the other hunter.” However, it seems that the EC community does not apply this evolutionary principle for judging the significance of some practical results/implementations.

¹⁰ Michalewicz, Z. [Quo Vadis, Evolutionary Computation? On a growing gap between theory and practice](#). Springer LNCS State-of-the-Art Survey, J. Liu, C. Alippi, B. Bouchon-Meunier, G. Greenwood, H. Abbass (Editors), 2012.

Several IT and Business Schools used the book *Adaptive Business Intelligence*¹¹ as a text in courses on modern heuristic methods, as the book covers many modern heuristic methods used for prediction and optimization. However, many of my friends from computer science schools told me that it is impossible to teach (in their schools) a topic that includes “business” in the title. I found it quite remarkable.

Further, most journals do not appreciate real-world (business) application papers; except, of course, when a paper describes a simple application, for some classic silo problem, tested on some benchmark cases available on internet or in other publications. Darrell Whitley (as I mentioned already, a former editor-in-chief of *Evolutionary Computation Journal*) wrote, “...when we have written pure application papers we have regularly had them rejected. A typical review: **This is just an application, there is no new theory.** This discourages the long and detailed work that must be done on real-world problems.”¹² It seems to me there is no symmetry between theory and practice in the EC community in the sense that it is much easier to publish a paper on, say, convergence properties of some version of EA on the unconstrained sphere problem (which will not help any practitioner in any way implementing EA-based solution to a complex business problem), than a paper on real-world application that was accepted and used by some businesses as their standard solution (but does not contribute directly to the theory of EAs).

In my recent private correspondence with several senior members of the EC community—past and current editors of the major EC journals who commented on the earlier draft of this paper—it was interesting to see that they expressed very clear (and opposite!) perspectives on application papers. Their comments varied from “There should be more rewards for doing real applications. Such work should be valued and published” and “It’s clear that we need a new set of review criteria for application papers” to “I would never recommend publishing such kind of material in a scientific journal [...] you usually will not learn much from such papers.” I believe the main reason for negative perception was a belief that business problems are easy to solve: “In business applications, the goal function is often quite simple, a linear one. Most of the modeling is put into the inequality constraints, which very often can be linearized and integer conditions can be relaxed. As a result, one often ends up with an LP [linear programming]. And if not, then the modeling is changed in such a manner that the resulting problem is again an LP.” With such understanding of business problems by senior people in the field, the case for application papers for some EC journals is lost.

¹¹ Z. Michalewicz, M. Schmidt, M. Michalewicz, C. Chiriach, *Adaptive Business Intelligence*, Springer, 2007.

¹² Private correspondence, 2011.

It seems (in general) that technical journals are not appropriate for publicizing business successes. Journal editors and reviewers are well versed in the standard criteria for acceptance: (a) clearly revealed algorithms, so the reader can at least attempt to replicate your approach, (b) well characterized problems, so the reader can tell if his problem is the right type, (c) rigorous comparison with known results, so the reader can have confidence that your results are significant. All this is needed for verifiability—the soul of science. So the EC community is no exception. However, it seems that our community, by adopting this approach, is making a huge disservice to themselves. Papers that describe real-world EA-based applications, which are in daily use at major organizations, and EA-based applications that support decision-making activities, which are essential to the core business of the organizations, should get very strong preference. Unfortunately, this is not the case.

5. Limitations of EA-based companies. Several companies emerged during the last 20 years, where the founders included researchers from EC community. There was one spectacular success (the company founded by Gil Syswerda was acquired by i2 Technologies for more than \$50M in 1997), but this acquisition did not result in any EA-based enterprise software application. There were also several spectacular failures; for a variety of reasons (the analysis of these reasons is outside the scope of this article) these companies faltered. Many companies with an EA profile which exist today operate in “survival mode”—their growth rate is minimal (if any), the number of full-time employees is usually expressed as a single-digit number, and in most cases they offer just consulting services (e.g. reports) as opposed to software delivery. They do not have any real plans for any exit strategy and they represent life-style business at the best (often it is just part-time activity to supplement full-time position somewhere else). Most of them emphasize the research component of their services, praising the technology (e.g. “...to bring the cutting edge of high technology to bear on real-world problems in industry, medicine, and defense...”) and people (e.g. “...our pioneering personnel have solved some of the most challenging problems...”). Often the other components necessary for running a successful technology company (e.g. specific knowledge on some verticals, graphical user interfaces, knowledge of business processes and software technologies, integration skills, which would provide a seamless fit with company’s existing databases, MRP/ERP systems, and other enterprise software) are missing. And for those companies, which do offer software, the offer is often limited to a variety of EA-based solvers—a far cry from EA-based enterprise software applications that would make a difference in the community.

6. Dominance of operation research methods in industry. Operations research is a well-known and mature discipline—almost every engineer learns linear programming (at university or on the last project)—and this is what the engineers do when faced with real problems. Thus the standard OR methods dominated optimization aspects of business operations for more than 30

years. Further, the OR community has a few standard and powerful tools (e.g., CPLEX) which are widely used in many organizations. On the other hand, this is not the case for the EC community—there are no “plug-in” software tools appropriate to deal with thousands of variables and hundreds of constraints, there are no tools available of comparable power to integer programming. Further, many OR methods are exact; they guarantee the optimum solution, which is not the case with heuristic methods in general and evolutionary algorithms in particular.

However, there is one catch here, as every time we solve a problem we must realize that we are in reality only finding the solution to a model of the problem. So whether we use exact or heuristic methods, it's difficult to obtain a precise solution to a problem because we either have to approximate a model or approximate the solution. In other words, neither exact methods nor heuristic methods return optimum solution to *the* problem, as the former methods simplify the problem (by building simplified, usually linear, model of the problem). So the optimum solution to the simplified model does not correspond to the optimum solution of the problem, and the heuristic methods return near-optimum solutions (but to the more precise models). A large volume of experimental evidence shows the more complexity in the problem (e.g., nonlinearity, conflicting objectives, noise, and constraints), the more appropriate it is to use heuristic methods. However, there is an amazing amount of inertia in the business that has nothing to do with which technology really works—and it is much easier to plug-in CPLEX than to develop an EA for a problem at hand.

Conclusions

A recent submission of a survey article on evolutionary algorithms (written with Marc Schoenauer for *Wiley Encyclopaedia of Operations Research and Management Science*, 2010 edition) generated editor's comment/request: “Can the authors provide objective guidance on the types of problems for which evolutionary methods are more appropriate than standard methods? I know a lot of ‘hearsay’ related to this issue but I am wondering if there is more objective evidence. Is there any solid evidence of EA superiority in a class of problems?” More and more people question the usefulness and applicability of EC methods and it is essential that our community would get ready to answer such questions.

I think that the right answers for above questions are not of the type: “EA techniques are superior for, say, symmetrical TSP” or “EA techniques are superior for, say, such-and-such types of scheduling problems,” as the main issue is in complexity of the problems. For example, multi-silo operational problems with many conflicting objectives, tens of thousands of variables, complex constraints and business rules, variability issues and noise, and interdependencies between operational silos—problems, for which standard operations

research methods are not appropriate.¹³ This is a big chance for evolutionary computation community, but the community should change the way they look at business problems. And the most important direction to achieve the goal of recognition in business and industry—from my perspective—is to focus on real-world applications. And I mean *real* real-world applications (as opposed to just paying the lip-service to the importance of real-world applications and beating the dead horse; solving TSP, JSSP, VRP, etc.). Clearly, there are real difficulties in reconciling the needs of the scientific research community (with strict demands of reproducibility and validation) with real-world optimization, where various constraints and business rules prevent classic comparative studies. We still need a change in the EA culture whereby researchers place greater value and have greater respect for performance improvements over existing business operations.

This may require, for example, the development of artificial problems that better reflect real-world difficulties, which the research community can use to experience (and appreciate) for themselves what it really means to tackle a real-world problem. It may require some education of research community, which is often disconnected from real-world problems. But above all, it would be worthwhile to revisit policies of some EC journals and introduce a different set of criteria to evaluate application papers. Garry Greenwood, editor-in-chief of the *IEEE Transactions on Evolutionary Computation*, summarized it nicely: “The notion that a paper has to have some theoretical component or it doesn't rise to the level of a journal publication is absurd.”¹⁴

Acknowledgements

The author is grateful to Brad Alexander, Hans-Georg Beyer, Jason Brownlee, Adam Ghandar, Garry Greenwood, Krzysztof Krawiec, Xiaodong Li, J. David Schaffer, Marc Schoenauer, James Whitacre, and Darrell Whitley for their comments—many of them were incorporated in the final version of this essay.

About the Author

Zbigniew Michalewicz is professor of computer science at the University of Adelaide in Australia. He completed his M.Sc. degree at Technical University of Warsaw in 1974 and he received Ph.D. degree from the Institute of Computer Science, Polish Academy of Sciences, in

¹³ A recent report (private correspondence) generated by several Operations Research experts on optimization of particular supply chain with 37 nodes and 69 arcs, 862,000 variables and 1.6 million constraints, would require 18 hours per objective by mixed integer programming.

¹⁴ Private correspondence, 2012



1981. He also holds a Doctor of Science degree in computer science from the Polish Academy of Science. Zbigniew Michalewicz also holds professor position at the Institute of Computer Science, Polish Academy of Sciences, the Polish-Japanese Institute of Information Technology, and the State Key Laboratory of Software Engineering of Wuhan University, China. He is also associated with the Structural Complexity Laboratory at Seoul National University, South Korea. He is the Chief Scientific Officer at SolveIT Software (www.solveitsoftware.com), a leading provider of enterprise software for integrated planning and optimization, which was recently acquired by Schneider Electric. Michalewicz has published more than 200 articles and 15 books on subjects related to heuristic methods. These include *Adaptive Business Intelligence* and *How to Solve It: Modern Heuristics* (both published by Springer). Other books include a monograph *Genetic Algorithms + Data Structures = Evolution Programs* (three editions, a few translations), *Winning Credibility: A guide for building a business from rags to riches*, where he described his business experiences over the last years, and the most recent: *Puzzle-based Learning: An Introduction to Critical Thinking, Mathematics, and Problem Solving*.

DOI: 10.1145/2390009.2399411