Exploring the Interaction Between Open Innovation Methods and System Complexity

Zoe Szajnfarber  
Department of Engineering Management and Systems Engineering and International Affairs  
The George Washington University  
Washington DC, USA  
yszajnf@gwu.edu

Ademir Vrolijk and Jason Crusan  
Department of Engineering Management and Systems Engineering  
The George Washington University  
Washington DC, USA  
avrolijk@gwu.edu; jcrusan@gwu.edu

Abstract—This paper contributes a systems engineering perspective to the discussion of where open innovation methods can (and could) work and can (and should) be used. It maps the existing experience-base of applying open methods to real-world problems, illustrating an apparent gap in applicability. The paper then explores the theoretical basis for expanding the use of open innovation methods. The discussion is structured around efficiency tradeoffs associated with (a) upfront system decomposition on the part of seekers, (b) the amenability of different kinds of problems to be solved by solvers with different levels of expertise, and (c) incentives for solvers to contribute solutions. These three dimensions of tradeoffs are then used to categorize existing innovation methods and frame the challenge of screening future systems problems for “openability.”

Keywords—open innovation; system complexity; decomposition; problem solving

I. INTRODUCTION

The America COMPETES Reauthorization Act of 2010, for the first time, granted all agencies the legal authority to conduct prize competitions. The act encourages the use of open innovation methods to “spur innovation, solve tough problems, and advance their core missions.” ([1], quoting America COMPETES, [2]) This new language is part of a broader trend to use prizes and challenges as part of an organization’s innovation toolkit [1], [3], [4]. Open innovation methods, which broadly include: prize competitions, grand challenges and collaborative communities have been touted as the solution to the stagnation of our national innovation economy. When applied appropriately, open methods can: (1) focus communities, without picking winners; (2) benefit from a wider spectrum of solution approaches, without bearing additional programmatic risk; (3) encourage wider participation, and in so doing garnering public engagement in important problems; and (4) achieve extremely high-leveraged investment, since the aggregate expenditure of contributors tends to exceed the prize purse by an order of magnitude [1], [5].

However, the approach is not appropriate for all problems. For example, Terwiesch and Xu demonstrate a relationship between solving expertise, search uncertainty and the relative returns from prize competitions and internal R&D [6] for one dimensional solving, illustrating how different kinds of problems are more amenable to different kinds of solving strategies. Similarly, Lakhani and his colleagues have begun to explore the relationship between the appropriateness of open methods and the interdisciplinarity of the problem [7], and the task structure of the industry [8]. These studies have done an excellent job identifying “low hanging fruit” in the current marketplace, where open methods can provide immediate returns. Yet, they do not address many of the kinds of more complex problems implied in the President’s Strategy for American Innovation. We argue that by taking the technical systems architecture as a given, and focusing on static returns, prior studies (a) miss a valuable opportunity to expand the applicability of open methods and (b) fail to consider the long term ecosystem implications of reliance on open methods.

To that end, this paper contributes a systems engineering perspective to the discussion of where open can (and could) work and can (and should) be used. It begins by mapping the existing experience-base of applying open methods to real-world problems. By categorizing past challenges in terms of their inherent complexity and level of a priori decomposition, the study illustrates an apparent gap in applicability. In the second half of the paper we explore the theoretical basis for actively decomposing complex systems to make them more amenable to open innovation methods. The discussion is structured around efficiency tradeoffs associated with (a) upfront system decomposition on the part of seekers, (b) the amenability of different kinds of problems to be solved by solvers with different levels of expertise, and (c) incentives for solvers to contribute solutions. These three dimensions of tradeoffs are then used to categorize existing innovation methods and frame the challenge of screening future systems problems for openability.

Consistent with the literature on prizes and open innovation, we use the following short-hand to describe challenge stakeholders. Seekers refers to the agency or firm that has the problem they wish solved. They may or may not do

1 This work was partially supported by grant NNX13AR06G
the actual posing of the challenge, depending on the mechanism being used. Solver refers to the contributing entity, be it an individual or a self-organized team. Platform refers to the third-party administrator of the challenge. Not all competitions involve a platform.

II. PAST USES OF OPEN METHODS IN GOVERNMENT

The use of prize competitions to solicit novel approaches to solve hard problems is not new. One of the first recorded prize competitions took place in the early 1700s, when the British Parliament used the “Longitude Prize” to catalyze the development of an instrument to accurately measure longitude at sea [5], [9]. More recently, increased connectedness has sparked a burst of prize competitions and open communities, including the Ansari X-prize, Netflix Challenge, and the proliferation of Linux. Prizes and challenges have also proven effective in assessing market size and stimulating interest. For example, the LEGO Cuusoo platform is set up to bring production considerations for ideas that receive more than 10,000 votes. In the case of LEGO’s “Back to the Future” set, during the 9 months it took to reach 10,000 votes, LEGO was able to collect valuable market research: the product was viewed over 400,000 times and received 200 comments from potential users [10].

In the technology space, NASA, through its Center of Excellence for Collaborative Innovation, has been a leader in exploring the value of open innovation methods for complex problems. Since 2010, in collaboration with multiple government agencies including U.S. Patent and Trademark Office, Health and Human Services – Centers for Medicare and Medicaid Services, Office of Personnel Management, U.S. Agency for International Development, and the Environmental Protection Agency, NASA and the federal agencies have launched more than 50 challenges [2]. The challenges have ranged from narrowly defined procurement-style challenges (e.g., develop new outreach videos to spur involvement and participation in a specific activity) to much more complicated multi-stage competitions in which contributors collectively define the sequence of future challenges, submit solutions and evaluate responses (e.g., Center for Medicaid services doctor enrollment application which included over 130 contests to build a complete enterprise class application using challenge methodologies). Table 1 summarizes the responses to a small selection of challenges.

By any measure, the competitions that have been run to date have been successful – they have elicited a broader response than expected, improved performance drastically, and identified unexpected complementarity across disciplines. However, there remains skepticism within the agency at large as to whether open methods can be applied beyond so-called corner cases, to problems at the core of the agency’s mission.

<table>
<thead>
<tr>
<th>Challenge Title</th>
<th>Series Sub-Titles</th>
<th># of Registered Participants</th>
<th>Were teams allowed?</th>
<th># of Submissions (quantity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asteroids Challenge</td>
<td>[ASTEROIDS-1] - Bug Race</td>
<td>n/a</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>[BUGR-10254] - Create a Glossary of Terms for Space/Asteroid Science</td>
<td>n/a</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>Sample Return Robot Challenge</td>
<td>Sample Return Robot Challenge - 2011 - 2012</td>
<td>11</td>
<td>Yes</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Sample Return Robot Challenge - 2012 - 2013</td>
<td>14</td>
<td>Yes</td>
<td>11</td>
</tr>
<tr>
<td>Night Rover Challenge</td>
<td>Night Rover Challenge 2013-2014</td>
<td>0</td>
<td>Yes</td>
<td>0</td>
</tr>
<tr>
<td>NITRD</td>
<td>FAST!! 48 hours! Restyle presentation for Big Data Challenge</td>
<td>28</td>
<td>No</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>NASA Climate Modeling - Search for New Sources of Observational Data</td>
<td>63</td>
<td>No</td>
<td>46</td>
</tr>
<tr>
<td>Robonaut Vision Challenge</td>
<td>NASA Robonaut2 Taskboard Controller Assembly</td>
<td>38</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Main Robonaut Challenge</td>
<td>1190</td>
<td>No</td>
<td>304</td>
</tr>
<tr>
<td>ISS Longeron Challenge</td>
<td>Longeron Shadowing API</td>
<td>9</td>
<td>No</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>ISS Longeron Shadowing Architecture</td>
<td>10</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>NASA ISS challenge T-Shirt and Logo Design Contest!</td>
<td>35</td>
<td>No</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>International Space Station Ray Tracing Software and Marathon Challenge</td>
<td>17</td>
<td>No</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>FAST!! NTL Marathon Match Infographic Design Contest</td>
<td>26</td>
<td>No</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>ISS Longeron Challenge</td>
<td>459</td>
<td>No</td>
<td>2185</td>
</tr>
<tr>
<td>DTN Infrastructure</td>
<td>DTN Dictionary Contest</td>
<td>n/a</td>
<td>No</td>
<td>2</td>
</tr>
<tr>
<td>DTN DTPC Interoperability Test Challenge Series</td>
<td>DTN DTPC Challenge Test Scenarios</td>
<td>17</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>DTN DTPC Challenge Test Cases</td>
<td></td>
<td>10</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>Collective Minds and Machine Learning Exploration Challenge</td>
<td>NTL - Crowd-Assisted Automatic Image Classification</td>
<td>22</td>
<td>No</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Marathon Match</td>
<td>39</td>
<td>No</td>
<td>394</td>
</tr>
<tr>
<td>NASA ECG Phase 1</td>
<td>Module Assembly - NASA ECG Viewer for Android Prototype Animation Assembly</td>
<td>17</td>
<td>No</td>
<td>3</td>
</tr>
</tbody>
</table>
The validity of this concern lies in the complex integrated nature of most of the technology NASA considers core. While the utility of open innovation methods have been demonstrated in practice across a wide range of domains, from wind surfing to scientific instruments [11] to spacecraft design (X-prize), and hint at broad applicability of the approach, the reality is that our empirical understanding resides in a few specific regions of the potential application space.

To illustrate this point, Figure 1 plots each of the technology challenges in NASA’s database in terms of the complexity of the system on the y-axis and extent of problem decomposition on the x-axis. In this context, complexity of the system (the y-axis) is an aggregate measure of two things: the interdisciplinarity of the problem and the depth of disciplinary knowledge in each area. The measure was operationalized as an exponential ladder, where each rung represents an added depth of knowledge for each relevant discipline (e.g., 1 = no specific knowledge required and 3\*4 = deep (PhD-level) and broad (multiple disciplines) required).

In developing this measure we also considered more traditional measures of complexity, e.g., functions of number of components, maturity and connections among them [12]. However, implementing such measures is extremely labor intensive and there is no timesaving associated with re-applying the measure to additional systems. Given our need to apply the measure to more than 50 different systems, it seemed infeasible. As a result, we sought inter-coder reliability as a means to validate the more subjective measure of “interdisciplinarity at depth” measure specified above.

Extent of problem decomposition (the x-axis) is a measure percent-complexity reduction of the problem as posed. It is operationalized as a ratio of the complexity of the problem, as posed to the complexity of the system, of which it is part. Both are evaluated in terms of the ladder defined above. Defined this way, extent of problem decomposition, can take on values from zero to one, where one represents a challenge that is as complicated as posing the full problem, and <<1 is a fully decomposed problem.

Each point in the plot represents a single challenge that was run by the Center of Excellence for Collaborative Innovation (CoECI). To date, the CoECI has not run any trivial challenges (simple problems, fully decomposed) or any extremely complex challenges. There are three areas in the space where multiple challenges have been run (not easily visually apparent since the points overlap one another). The first is in the top-left, where complex problems are run in a fully decomposed manner. In such instances the seeker invests substantially in posing the problem in such that it is context agnostic, and solvable by a single solver with relevant expertise. For example, the goal of the International Space Station (ISS) Longeron challenge was to develop an algorithm that would allow for the position the solar collectors on the ISS to generate as much power as possible during the most difficult orbital positions. The screening to determine the top performer was solely based on overall predicted power performance as an output of a provided simulator. There was a high-perceived level of complexity of this problem, yet, more than 2000 submissions were received. Although the seeker perceived this challenge as difficult and complex, from the perspective of the solver, it was seen as an important math problem. To support the challenge, NASA did significant upfront decomposition, posing a well-contained problem. A simplified model of the ISS was provided which included a set of defined mobility joints for the solar arrays. In addition, a virtual simulation environment was made available to run the complex orbit calculations for power generation. This simplification resulted in a problem that required only a single discipline or even trial and error to solve.

The other well-explored region in Fig 1 is the top right corner, where complex problems are challenged as fully integral systems. These are typically run over multiple years as Centennial Challenges, and attract participation from self-organizing small teams who enjoy the challenge and likely have a strong multidisciplinary background across the team. The last dense region is mid-low complexity across a wide range of decompositions. This is the region where traditional coding communities operate. As software based systems, the costs of contributions are low and within-community prestige is high, making solvers willing to invest substantial time working on meaty problems.

Should there be other regions where open innovation methods are widely applied in practice? In the above discussion, we introduced three dimensions that underlie the answer to this question: (1) extent of upfront decomposition on the part of the seeker; (2) interaction of solving ability and problem formulation; and (3) willingness to contribute on the part of the solver. The next three sections explores each of these dimensions in detail and then returns to broader question of where open methods can/do apply and can/should be extended.

III. TRADEOFFS ASSOCIATED WITH UPFRONT DECOMPOSITION

All systems are decomposable to some extent [13] and broadly speaking, higher levels of decomposition make problems more amenable to open solving. However, decomposition is not free. In a given system, the cost of
decomposition it’s related to the natural and potential decomposability, which can vary widely from one system to another. In this section we explore the extant theory related costs and benefits of upfront decomposition.

A. Natural vs. Potential Decomposability

Natural decomposability refers to the level of decoupling that comes essentially for free: separating parts of the system that were only ever weakly linked in the first place. Baldwin calls these inherent structural cleavages thin crossing points [14]. As a practical example, consider one system concept for NASA’s ongoing Asteroid Grand Challenge. The challenge, to “find all asteroid threats to human populations and know what to do about them” has at least two high-level functional components: detection and deflection/redirection. The detection pieces will likely require a space-based observatory, ground stations and significant post processing. In Baldwin’s there is a thin crossing point between the physical hardware and the post-processing component. The only link is pass forward raw data to be processed. As a result, it can be decoupled from the other system elements without limiting the explorable solution space, constraining future technical trajectories or enforcing an additional coordination burden. Not surprisingly, it is among the first elements for which NASA intends to use open innovation methods to improve performance.

Naturally decomposable elements are fairly straightforward to identify, are easy to assess for appropriateness and can yield high returns almost immediately. They represent most current applications of (e.g., crowd sourcing) open methods. However they are, by definition, peripheral to the core system [15], and only embody a small portion of the potential decomposability.

In addition to the first level of natural decomposition, all systems can further be actively decomposed into progressively smaller sub-problems, by imposing design rules [16]. Design rules are global explicit rules about how a system will operate. The idea is to define key design parameters and guarantee that they will not change; this allows modules to depend on design rules and not on each other [16]. Design rules can take the form of standards (e.g., 802.11b) or interface control documents (e.g., common in systems engineering). When chosen well, they can streamline design and upgrade processes substantially. For example, Baldwin and Clark documented how design rules enabled to computer industry to evolve from integral mainframes to desktops with modular interchangeable subsystem. This was enabled by a standard bus interface structure and defined form factors. The act of imposing design rules on the system drastically increased the rate of overall innovation in the industry [16]. The competitive structure of the industry also fragmented, allowing smaller firms to find niches in the newly created subsystem markets. As expected, future innovation was confined to with-in module changes, locking the product into the boundaries imposed by the defined technical and industry architecture [17], [18].

In the case of the computer industry, the benefits of decomposition and associated co-specialization more than outweighed the costs. However, this tradeoff is system dependent, and for a new design depends on two dimensions: (1) the structure of the inherent technical interdependencies; and (2) the nature of the uncertainties present in the intended operating environment. The former is a static design question, while the latter is dynamic and can only be measured over a systems lifecycle. For design evolutions, legacy industry structure also factors in.

1) Static Technical Structure

Volumes have been written about the value of modularity [16], [19], [20]. Standard modularizing operators have been defined, and their benefits discussed qualitatively. However, the costs have only been discussed in a superficial way (e.g., in terms of structural overhead associated with splitting [16]). An important, yet unexplored, question lies in how decomposition limits the extent of a design space that is explored. Specifically, systems engineering scholars have long advocated for the importance of exploring a fully enumerated tradespace of potential systems architectures [21]–[23]. Proposed methodologies typically consider the full factorial set of design alternatives in terms of cost and multi-attribute utility [23]. A stylized tradespace is shown in the left of Figure X. Framed this way, the set of best designs (worthy of more detailed comparison), are those on, or close to, the Pareto frontier (drawn in red in the figure).

However, the act of imposing a design rule, by definition, reduces the set of possible architectures that can be explored. In fact, design rules typically eliminate “whole families” of architectures. For example, in a fractionated spacecraft design, eliminating a family might remove all two-module designs that share a power subsystem. Some reductions in the scope of search are beneficial. Others are less so. In the ideal case, an imposed design rule might limit the scope of search to the set of architectures that lie on, or close to, the Pareto front (see green oval, in figure 2). In such cases, efficiency of search is drastically improved because much less search is required, while still exploring all good architectures. More realistically, a set of design rules would confine a search to a particular family of architectures. This might take the form of the column of alternatives highlighted by the blue oval. In such cases, search efficiency remains essentially constant: the scope is smaller, but remains proportional to the number of “good” architectures in the original space. Weak design rules can constrain search to exclusively non-Pareto architectures,
denoted by the purple oval. In such cases, efficiency drastically reduced.

In general, decomposition speeds up the design process by parallelizing tasks and reducing total search. It also focuses attention on core elements. As will be discussed later, it also opens up opportunities for open contributions. However, decomposing comes at the cost of across module exploration and therefore precludes some solutions. For most systems, higher levels of decomposition are likely to yield diminishing efficiency gains after a point. That point is dependent on characteristics of the specific system and the associated design rules.

2) Uncertainty in the operating environment

In addition to limiting the scope of the solution space explored, the act of decomposing limits the future trajectory of design evolution for the system. As noted previously, innovation is much more likely to occur within modules, than across [14]. This so-called lock in is a function of both specialization and mirroring in the supporting industry. As a result, design choices made early on (e.g., to run a challenge on a fully decoupled module) are likely to be preserved in the system architecture until some disruption (e.g., [24]) in the sector causes a new cycle of variation retention and selection [25]. This is particularly important in complex systems domains (e.g., the space and defense sectors) where the monopsony market structure limits the potential for creative destruction [26].

Evaluating the costs of lock-in is akin to the dual problem of valuing the “-ilities” [27]–[29]. Depending on the structure of uncertainties in the operating environment, being able to change, adapt, and evolve a system can be critical to maintaining system value over its lifecycle. The systems engineering community has developed numerous metrics and frameworks for valuing flexibility [28], changeability [30], survivability [31], adaptability [29], among others. One of the core contributions in this area is due to Ross and his colleagues. They formulate the lifecycle of a system in terms of a sequence of discretisable Epochs (during which use context remains constant) and Eras, when context changes. Framed this way, a Robust design is one that stays on, or close to, the Pareto frontier under multiple conceivable Epochs. However, robust designs are rarely optimal (i.e., you are more likely to have a pretty good architecture stay pretty good, than have a great architecture be great under multiple scenarios) and only become a good choice under extreme uncertainty. The concepts of Adaptability/Changeability/Evolvability are important because they embody an option value. A design that is Pareto optimal under expected conditions, and can be easily changed/improved if the context does, can be a much better choice. Ross measures these concepts in terms of out degrees – paths through which a particular architecture can be evolved [29]. As with the efficiency notion introduced above, there is a relationship between “illities” and upfront decomposition.

IV. INTERACTION BETWEEN PROBLEMS AND SOLVING

Having discussed the static and dynamic tradeoffs associated with decomposing a system above, this section considers the relationship between decomposition and the ability for external entities to contribute useable solutions. A core reason why seekers typically pose highly decomposed challenges is because, in general, simpler problems are solvable by a larger pool of people. However, complexity and level of decomposition are not perfectly correlated to one another, and particularly when looking for opportunities to pose challenges on more complex systems there is an important interaction between the way that problems are solved and what kinds of problems are best solved by different kinds of solvers. This section focuses on the interaction between solving-strategy and expertise, for a given framing, and discusses how this influences the way open can and should be used. The next section deals with the issue of incentives (i.e., given that a solver can solve the problem, why do they?).

A. Problem Framing

In the context of open innovation prizes and challenges, problem framing is shared, to varying degrees, by both the seeker and the solver. Except in cases where the problem-asposed is identical to the solution-as-it-will-be-used, seekers do some amount of upfront problem framing – often narrowing the scope of the problem as seen by the solver. Historically, solvers have enjoyed success posing naturally decomposable problems, but they have also seen varying degrees of success actively decomposing partially decouple-able problems. For example, in posing the ISS Longeron challenge, NASA invested substantially in giving solvers a realistic (simulated) environment in which to explore subsystem solutions. Conversely, Netflix has failed to adopt the improved recommendations algorithm that won their challenge, because the infusion costs would outweigh the benefits.

Once the seeker poses the challenge to the solvers, each engages in his, or her, own framing as they begin to solve the problem. Depending on the level at which the problem is posed, some of the solvers initial activities might mirror those of the seeker. The internalizing and solving processes are typically conceptualized as two separate stages. First, the solver makes sense of the problem, producing the solver's personal “problem space” out of the “task environment” given by the seeker [35]. Then the solver begins to search for a solution within the problem space. The subsections that follow consider the general approaches that have been observed and theorized with regards to how the search is conducted.

The literature on framing and solving has largely focused on cognitive processes at the individual level. For more complex problems, and solving by teams, it is understood that multiple solving mechanisms can be used in parallel and in sequence over the course of the solving process. Generally speaking, more complex problems are solved with an initial superficial path, followed by progressive depths in certain parts of the system (e.g., spiral design, SE vee). In the
sections that follow, we focus on the mechanisms individually, before considering how their interaction with different kinds of problems.

B. Solving Processes

Human problem-solvers search solution spaces in a way that is analogous to optimization algorithms. However, unlike algorithms, they rarely search the whole space. They focus on particular parts, or use heuristics to attain a certain level of “goodness” and then generally stop without confirming that there is no better option. This lack of optimization has been termed “satisficing” by Simon. The methods described below explore the different ways that humans “search” within Simon’s mechanic of satisficing.

Broadly, there are three categories of human problem solving strategies: (1) Searches that seek to explore the full set of possible solutions; (2) Searches that spiral out from a selected starting point, and stop when a good enough solution is found, and (3) Searches that explore a reduced region that is believed to hold an at least local optima. The methods are arrayed roughly in terms of increasing expertise required.

1) Random Experiments over Full Problem Space

a) Pure Chance Method

The pure chance method is rarely used in practice, and is considered as a conceptual extreme. Here, a solver, possessing no skill or knowledge related to the problem at hand, randomly explores the space. During the search, no learning is assumed to occur in the mind of the solver, nor does the problem space change over time (positively or negatively). If design points, or elements, on the Pareto-front are arrived at at all, it is on the basis of chance. And, the path traversed to get there, which constitutes the specific operations performed to reach an element, are not recorded.

b) Trial-and-Error Method

Trial-and-error methods add a distinct nuance to the above method. If the solver does not possess skills or knowledge required for this task, they can gain information from overt exploratory activity [36]. But this exploratory activity should not be interpreted as blind random activity: the solver is now capable of exploiting any additional information made available during the exploratory activity [37]. This effectively narrows the solver's problem-space, sometimes considerably. However, the operations conducted to actually solve the problem at hand are still performed at random.

These two methods of search explore the entire problem space. As a result, the time to solution will be roughly proportional to the total size of the problem space. More sophisticated methods, which incorporate a solver's skills, knowledge, learning ability, and search strategy, can effectively cut down parts of the space that must be explored in order to find an appropriate solution [37].

Trial-and-error search can, in theory, be applied at any level of a system, however true trial-and-error is most relevant when the cost of multiple is low, limiting the value of leveraging analytical methods to reduce number or trials.

2) Heuristics that guide the path of search

Heuristic methods essential choose the path from an initial element in the problem space to one near(er) to the aspiration level. In this case, the problem space is assumed to be characterized both by the solver's interpretation of the task environment, as well as the strategies used to select operators which generate subsequent (neighboring) points.

There are several operator selection strategies that come into play using this method, and the solver's selection will depend on their history of experience with specific methods and operators, which that can be used to adapt subsequent performance [38]:

a) Local Search Strategy

The term “local search” is a catch-all for denoting strategies that pertain to the solver selecting operators that they are most familiar with, have the most access to, and/or have seen success when employed on given problem [39]-[41]. Intuitively, if the current problem is similar in structure to previous ones, an experienced “expert” can increase search efficiency substantially by picking a starting point close to good enough. For example, a master chess player will search for, and immediately adopt, a strategy that guarantees a checkmate; they do not look for all possible checkmating strategies and adopt the best [42].

Novices can also use local searches, but there is a higher risk of stopping at a local optimum that is globally suboptimal, because they lack the experience to choose a good starting point. Once the starting point is chosen, the search path strategies are similar for novices and experts. A specific implementation of a local search is “hill-climbing” or a directional search strategy [38],[43],[44]; a heuristic that leads solvers to make step-wise operator choices that take the current state of a problem closer to the aspiration level. This is closely related to the means-ends strategy [37], [38], which involves recursively setting subgoals to reduce large differences between the current state and the aspiration level first, then proceeding to select operators in a similar fashion to hill-climbing. As elaborated by de Groot [45], a local search also incorporates progressive deepening: after a particular path has been explored to a position that could be evaluated, the solver can return to a base position to pick up a new path for exploration.

Local search can nominally be conducted at any level of the system, however it is only done at one particular level at a given time. For example, aerospace engineers typically perform a local search when building a next spacecraft, only considering incremental or component improvements to a stable design (system level). Similarly, device physicists will typically begin with known materials when they are searching for better material properties to support a particular application (component level).

b) Recombination Strategy
Recombination search looks for existing modules that can be recombined to achieve some new end. Though discussed by many authors, we can find traces of this strategy's origin in Schumpeter's work [46], who stated that “innovation combines factors [or modules] in a new way, or that it consists in carrying out New Combinations.” This idea was expanded upon by Ethiraj and Levinthal [43]; the recombination strategy describes the solver's actions to identify and select modules (knowledge/subsystems) from (potentially) disparate sources, and to modify and combine these to produce an element on the Pareto-surface. They contend that the recombination strategy differs from the local search strategy above in that they are applied to different levels of a system; local search at a higher level of decomposition than the recombination strategy. Fleming [41], on the other hand, does not consider the levels of the system when explaining that recombination usually happens with modules that are within the solver's grasp: “local” vs. “distant”.

In theory, recombination can occur at any level of the system except the lowest level, which by definition has no sub-elements to combine. However, in recombination search there is an interaction between level of the system and the effectiveness of novices and experts. In the midlevel of decomposition, one would expect experts to arrive at good solutions much more quickly since they are familiar the variants of existing modules. Conversely, at levels where decoupling among modules is clean, the novices lack of familiarity can be an advantage. For example, von Hippel made famous the story of a medical resident who found himself biking to work in hot weather. To avoid dehydration he placed an IV bag in his backpack. We now call his recombination a Camelback [32]. In this example, it is unlikely that a bicycle water bottle manufacturer would have thought to attach the bottle to the rider instead of the bike, were this an obvious change for the rider.

3) Experience-based reduction of the problem space
a) Cognitive Search Strategy

Cognitive search describes a means to reduce the scope of search upfront. When experience is relevant to the problem at hand, a concerted effort can be made to distinguish the design parameters and/or the focus areas which will drive an element closest to the problem space's aspiration level, before the trial-and-error/operation selection phase. This is referred to as a cognitive search [44]: trials are selected based on a cognitive map or implicit theory of how knowledge sets and specific design choices relevant to the problem interact to maximize final performance.

b) Intuition

Intuition is a passive variant of the same kind of search. It recognizes that problem-solving also occurs in the unconscious mind, where multiple strands of information are integrated simultaneously [47], referred to by Chase and Simon [48] as “perceptual processing”. In their commentary on chess, they elaborated on how the skill level of chess masters allow them to quickly reduce their problem space to only the “good moves”, that are then further processed. These “good moves” are thus “seen”, and not consciously deduced [48]. This process occurs without (conscious) effort, and the solution(s) arrived at instantly. Using Chase and Simon's definition, the solver is required to have extensive knowledge and skill in the requisite field(s) in order to be able to distinguish perceptual processing (where a favorable element is the result) from the Pure Chance method described earlier.

c) Recognition

Finally, at the other conceptual extreme lies the recognition method: occasionally a solver will solve a problem in this way relying on their, at times extensive, prior experience [37], [49]. Newell and Simon contend that, in the case of complex problems on the solver’s side, this method proceeds by reduction: a hard problem is solved by replacing it with ostensibly easier problems. However, we must note the difference between this method and the means-ends search; here, the decomposition performed by the solver produces subproblems that are recognizable, as opposed to the means-ends strategy which produces problems which are manageable.

C. Solving and Expertise in an Open Competitions

Clearly, some solving strategies are more efficient than others. A chess master with an intuitive sense of the right few moves to assess will almost always beat a novice who needs to consider a much larger set of possibilities to find a good next-move. However, as evidenced by Deep Blue, IBM's chess supercomputer, when (human) processing power is not at issue, and the full space can be explored effectively instantly, the expert advantage fades. One way that crowd-sourcing is being used to great effect is essentially by turning crowds into efficient computers. In some cases, prizes for one million laymen are much more economical than retaining a single well-paid expert.

However, the extent to which this statement is true is dependent on (1) the underlying problem structure, (2) the level of decomposition of the problem, and (3) the willingness of external contributors to participate. First, so-called rugged landscapes limit the ability for experts to find global optima with local searches [6]. By the nature of the problem space, or the limited priori knowledge that is available, experts still need to do a wide search to be confident in having found a good solution. These types of problems are even more amenable to open methods of the crowd-sourcing variety than the chess example. Repetitive and tedious tasks also fall into this category (e.g., PTO image tagging challenge).

Second, mid-levels of decomposition are most likely to advantage domain experts because it is least likely in this region that the problems can be fully decomposed and made domain agnostic. As a result, trial-and-error searches are unlikely to be effective, and, since the solution will need to work with other pre-defined modules, the novice advantage at seeing radically different approaches is more likely to make proposals uninfuseable (e.g., Netflix). Improving our
understanding of how open works in this middle range is an important area of future work.

Third, the issue of willingness to contribute is the topic of the next section.

The meta-takeaway from this section is that novices and experts solve problems differently and their strategies work better for different kinds of problems. Thus, when determining how to decompose a problem to make it “openable” the seeker must consider who the target solvers are. In complex problems, if the seeker chooses to play the role of systems integrator, multiple simultaneous subproblems can be posed, with post hoc effort to integrate the resultant elements.

V. WILLINGNESS TO CONTRIBUTE

The last theoretical piece that must be considered is the question of what motivates potential solvers to submit solutions. The actual number of contributors participating in a competition is a proportion of the potential solving population that actually choose to put in the effort required to propose a solution. Where the above discussion focused on who can contribute, this section explores why they would. In addition to sheer numbers, motivation factors into the level of effort associated with the contribution, and must be considered as well.

Economic tournament theory places its focus on the particular conditions that occur during a contest. Participants, who are risk neutral or risk averse under traditional reward structures, accept riskier incentive structures than they normally would [50], leading to compensation schemes based solely on the contestants’ ordinal rank, and not the size of their output (or effort exerted). In fact, research shows that if the prizes are very sensitive to the participants’ output, it could lead to destructive competition [51]. Nevertheless, the motivation for participation remains a probability of payoff [50], which is dependent on the number of participants and their efforts/investments [52], which is in turn dependent on the size of the prize [53]. Although conventional economic theory suggests that the optimal number of competitors is two, Terwiesch and Xu [6] have shown mathematically that for some classes of problems, the benefit of solver diversity outweighs any losses of motivation due to heated competition.

The existence of collaborative communities of volunteers suggests that motivations are not limited to the prospect of financial rewards. For example, Lakhani and his colleagues have explored the impact of scoreboards and status as a non-monetary source of motivation [7]. The practitioner community generally agrees that motivations stem from some combination of Gold (e.g., a $10M prize purse as is the case with the X-prize foundation), Guts (e.g., a desire to solve an unsolvable problem – NASA receives 20% more contributors than an average challenge), Good (e.g., the basis for many philanthropic challenges) and Glory (e.g., the recognition associated with winning) [33].

When the cost of contribution is small and the probability of success is relatively high, any of “4 Gs” can serve as sufficient motivation. However, as the cost increases, or the expected return decreases, these dynamics change [7]. Substantially more empirical work is needed to truly understand willingness to contribute in the context of more complex problems.

VI. RELATING CHALLENGE TYPES TO WHERE THEY WORK

The above sections have focused on integrating the diverse extant theories related to three dimensions of challenge success: tradeoffs in decomposition, interaction of problem framing and solving, and why people contribute. In this section, we use that language to revisit our assessment of what has worked and how the scope of applicability of open methods can be extended. To date, the range of open methods that have been tried fit into five categories described below:

1. Market stimulating challenges solicit fully implemented solutions to extremely difficult, typically complex problems (e.g., Ansari X-prize). They require no upfront decomposition on the part of the seeker and solvers often spend much more than they hope to gain from the prize (in monetary terms). The lead solver often assembles (sometimes formally hiring) teams of experts to work over years. These times often look very much like small businesses and use traditional systems design methodologies. They participate for the prestige and potential for future returns.

2. Lead user tinkerers and entrepreneurs innovate by modifying existing systems to suit their advanced specialized needs [8], [9]; (e.g., adventure sports, scientists developing specialized instruments). Tinkerers are generally motivated by a need to meet their own needs (in advance of the rest of the market). They tackle problems from extremely simple to quite complex. They typically capture the benefits internally; though recent studies have unpacked situations were lead users become user entrepreneurs [11]. In the open innovation space, the maker community epitomizes tinkerers.

3. Software development communities are characterized by low costs of individual contributions, complete decomposability of problems, and a large number of qualified contributors [34] (e.g., Linux, TopCoder, hackathons). They are the tinkerers for software problems. Solvers participate because they enjoy the challenging problems and the respect they receive from their peers.

4. Broadcast searches seek to pose problems in a domain agnostic way, hoping to find answers where the seeker wasn’t previously looking. The challenges are often structured like prize-based procurements (below), but the aim is to reach solvers who can answer challenges with pre-known [38], pre-packaged, occasionally complex answers (e.g., innocentive, proctor and gamble) that are low cost to the solver, but high value to the seeker [54]. This typically involves substantial upfront decomposition.

5. Prize based procurements pose self-contained (i.e., fully decomposed), minimal effort, challenges that are later aggregated by the seeker (explored to some extent in [6]);
(e.g., mechanical turk). The prizes are usually small, but so is the associated effort.

Figure 3 illustrates the regions where these open mechanisms are most appropriate in the complexity vs. decomposition space. There is a band of mid-level complexity where multiple open mechanisms have been shown to be effective. Similarly, the full range of complexities is in bounds when the problem is decomposed (e.g., TopCoder receives hundreds to even thousands of contributions to their algorithm challenges). Lastly, high complexity, integrated problems are challengeable through X-prize like platforms (e.g., competitors invested a collective $100M+ to win the $10M X-prize). However, none of the established approaches have been shown to be appropriate in the mid-to-high joint complexity and integrity region – where most core agency problems lie. Figure 3 is roughly consistent with the data points shown in Figure 1, recognizing that not all challenge mechanisms are currently in use at NASA.

![Figure 3 – Where different open mechanisms work](image)

Historically, the white space between (1) and (2-5) has been the domain of subcontractors. It is unlikely that open mechanisms will ever fully overtake the role for firms to play in inventing the future, however to date, history has only given us evidence of the value of the obvious opportunities for open; there may be opportunities for much more.

VII. AN AGENDA FOR FUTURE RESEARCH

There are two avenues of future research that can leverage a systems engineering perspective to extend the boundaries of open applicability: empirical experiments at the boundaries of applicability and strategies for upfront problem decomposition to push challenges into feasible regions.

With respect to the first, additional empirical/experiment work is required to full characterize the boundaries of where open methods can be applied to complex systems. At this point, there is strong evidence of a fundamental limit. To date, in NASA’s experience, there is only one example of a failed challenge. This challenge resulted in no submissions even after altering the response timeframe and incentives for the solvers. The challenge was perceived by the seeker to be a simple system architecture challenge to develop an incremental improvement to the existing architecture. The incremental improvements included new hardware (sensors and control systems), software, and the firmware. The problem statement included the release of the original detailed design that led to the existing system architecture. The results of the challenge showed a more integrated challenge than what was perceived with 45% of the surveyed respondents stating the multiple discipline nature of the problem was the reason for the lack of response. More challenges are planned in order to explore the nature of this perceived integrity on the part of solvers. They will be described in a future paper.

With respect to the second, this paper has contributed a preliminary theoretical framework that integrates multiple streams of literature. It emphasizes the importance of clear communication between the solvers and the seekers to properly leverage potential contributions on the part of the solver. It highlights how problem-solving strategies can change as a function of subject matter expertise. It advocates a view of shared framing as an avenue to bring value in more complex systems. In theory, any system, no matter how inherently complex can be decomposed into at least some challengeable problems. However, there are of course costs, and like any other tool, open should not be applied everywhere. Future work will aim to further bridge the gap between the different stages of the open process; complexity and decomposability of the problem, participants’ motivation, problem solving strategies, and quality of results. Furthermore, screening tools will be developed to identify and prepare high value opportunities in any particular system.

Open innovation methods hold enormous potential, particularly in budgetary constrained environment. However, as they stand the methods cannot and should not be applied to all kinds of problems. If a better understanding of the interaction between problem complexity and the use of open innovation methods is not achieved, there is a risk of locking future innovation pathways into suboptimal trajectories that constrain rather than enable the game changers that the government seeks.

REFERENCES


