# Business Intelligence in Trial-Based Problem Solving: A Theory-Building Simulation Study

**ABSTRACT**: In turbulent environments where change is not only rapid but also unpredictable, a trial-based problem solving (TBPS) approach may be a prudent alternative to over-reliance on organizational experience. In this paper we use a simulation methodology to develop a new theory of trial-based problem solving in order to identify key challenges and to locate the leverage points where information technology may be used to enable or support TBPS. The problem solver in our model faces a dual task: to determine *how* to solve a problem as well as *whether* it is profitable to solve it. Based on this conceptualization, we surface two important trade-offs – between exploration and exploitation, and between search scope and search depth – and consider mechanisms for coping with the perishability of data in a turbulent environment. Our propositions are summarized as a new model of the antecedents of adaptive problem solving performance in IT-supported TBPS.

**CONTRIBUTION:** This article contributes not only a new theoretical understanding of the role of information systems in organizational problem-solving in turbulent environments, but also a new *type* of theoretical understanding in the form of a stochastic processes model. Utilizing a methodology of simulation research that is current in the organization theory discipline but not yet frequently seen in the IS literature, we are able to give articulation and detail to theoretical constructs and relationships without resorting to proxy variables and correlations. As a result, we can uncover relationships of nonlinearity and equifinality (e.g. concerning the effects scope and depth of search strategies on performance), and provide insightful explanations for unexpected findings (such as the dominance of exploitation over exploration even in an early idea-screening stage of search).

The goal and the benefit of this approach to theorizing has been to understand the “leverage points” by which information technology capabilities might be able to support trial-based problem solving, to open up the theoretical black box of business intelligence and enable meaningful distinctions to be made amongst its possible affordances. We are able to separate the impacts of improved measurement, from those of data freshness, from those of a rapid feedback loop, and show how each aspect contributes to organizational performance in turbulent environments. This is intended to be a foundation for new research of a more theoretical flavor in the business intelligence area, as each of the new constructs and relationships we surface in this research is a proper subject worthy of study in its own right.

# 1. Introduction: Turbulence and Trial-Based Problem Solving

Successful problem solving in changing organizational environments depends upon dynamic adaptation of one or more factors: the environment itself, the definition of the problem, the structure of the problem solving system or its state (i.e. its knowledge about the problem and environment or its allocation of tasks and resources; Gasser & Ishida, 1991). Models of *adaptive problem solving* are therefore diverse and differentiated by the thing to be adapted; for example, the artificial intelligence literature has developed models for problem solving systems that autonomously adapt their own structures (e.g., Gasser & Ishida, 1991; Gleizes et al, 1999; Kota, 2009; Timmis et al, 2008), and the organizational literature offers theories of organizational learning (e.g., Argyris & Schön, 1978; Levitt & March, 1988) and dynamic capabilities (e.g., Eisenhardt & Martin, 2000; Teece, 2007; Winter, 2003) that explain how knowledge, routines, and resource allocations in organizations are adapted over time.

Relatively little consideration has been given to the type(s) of change being adapted *to*. It turns out that many existing models of adaptive problem solving by organizations break down when faced with *environmental turbulence*, a condition faced by more and more organizations in the digital era. Turbulent environments are those in which unpredictable and increasingly rapid changes emerge endogenously from the complex interconnections between organizations and other actors (Emery & Trist, 1965). The causal texture of turbulence is such that organizational reactions to changing conditions may trigger greater and even more disruptive changes, rather than returning the field to any sort of equilibrium. A variety of different environmental dynamics result from turbulence (Davis et al, 2009) and these amplify a number of tensions for business managers (Carlsson & El Sawy, 2008). Most salient among these issues is that organizations in turbulence face an increasingly unpredictable future, and this unpredictability undermines problem solving methods that leverage knowledge from an organization’s own experience.

Unfortunately, most models of adaptation in organizational problem solving assume that change is more or less predictable and that the *velocity* of change is the most, or only, salient challenge of turbulence. For example, behavioral models of organizational learning focus on the ways that organizational knowledge is built up from retrospective experience (i.e., the development of routines) and overlook the need for organizations to sometimes “drop their tools” when the environment changes in an unexpected way (Weick, 1993). The theory of *dynamic capabilities* assumes that regular processes for updating organizational capabilities are sufficient for adapting to change, missing the potential need for improvisation as unforeseen disruptions occur (Pavlou & El Sawy, 2010). What is missing in these models is a solution to the *unpredictability* of change in turbulent environments. If managers guided by these theories respond to turbulence by using familiar methods only more quickly, they can fuel vicious cycles that end in hyper-competitive hyper-turbulence where no organization wins (Selsky et al, 2007).

A more prudent strategy for turbulence is to adopt an adaptive problem solving approach that is trial-based rather than experience-based. The basic outline of *trial-based problem solving* (hereafter TBPS) is that organizational experience be augmented with a number of small-scale trials of alternative experiences (e.g., pilot tests of products, services, strategies, etc., that differ from the organization’s standard fare), that analysis of the environment be based more on these diverse but recent trials than on narrow but deep experience with the organization’s main line of business, and that the organization maintain a readiness to rapidly update its solutions in response to the signals of change it detects from this analysis. Brown and Eisenhardt (1997) have described a trial-based strategy in the context of product innovation, with experimental products, strategic alliances, futurists, and creative meetings serving as “low-cost probes”. They suggest that “semi-structured” organizations, which couple flexibility of action with a limited set of strategic rules, are the ones best equipped to move quickly when the environment changes (cf. Davis et al, 2009). In a similar vein, Pavlou and El Sawy (2010) propose that researchers identify “improvisational capabilities” that enable rapid but unplanned adaptation to abrupt and unforeseen changes.

As turbulence becomes increasingly a part of economic reality, it is important that we better understand how TBPS can be manifested in context and what implications it has for managers. As digital technologies play key enabling roles in TBPS (making it easier to generate trials, to analyze feedback, and to adapt products and services to the changes detected), it is also important to determine which affordances may have the strongest effects on TBPS performance. This paper articulates a process model of a trial-based problem solving organization, a model which allows us to develop some new theory about both the trade-offs that matter in trial-based problem solving, and the ways that new technologies impact TBPS. This study’s objective is a preliminary theory of TBPS which highlights critical antecedents of adaptive problem solving performance and points out where further research and practical solutions are needed.

# 2. Challenges

Consider a problem that an organization might attempt to solve, for example, a product idea it might try to create or a service it might try to develop. The trial-based problem solver must be able to reach two types of inferences – *how* to solve the problem and *whether* to solve the problem. Thus he is at risk of making two types of mistakes: false negatives, in which opportunities for profit are missed because the right solutions (e.g. the right combinations of product features or service elements) could not be found, and false positives, in which resources are wasted developing ideas that could never have been profitable (Bonabeau et al, 2008). Ideally these inferences would be reached sequentially, with some sort of idea-screening or opportunity-finding process preceding a development or problem-solving process. In turbulent environments, however, there are two complications. First, the *apparent* potential of an opportunity is only as good as the best-performing trial solution, and second, the true best solution continually changes as the business environment (e.g. set of customer preferences) changes over time. Thus some solution-seeking activity must precede the decision to keep or abandon a problem, and the organization must continually iterate between seeking better solutions and using them to evaluate whether a problem is worth solving.

We can adapt the concepts of exploration and exploitation from the literature on problemistic search (e.g. March, 1991) to understand the relationship between these two inferences. The principle is that search activity can include *exploration*, seeking new possibilities for solutions, or *exploitation*, seeking incremental improvements to existing solutions (Levinthal & March, 1993). The literature treats exploration and exploitation as a forced trade-off, that is, the two types of activities compete for a budget (of attention or of resources) and each organization must find the appropriate balance of the two. The *ambidexterity* proposition is that in turbulent environments, some mix of exploration and exploitation activities will perform better in the long run than either strategy by itself (He & Wong, 2004; Uotila et al, 2009). Trial-based problem solvers face a similar trade-off. They must conduct a range of diverse trials in order to find better solutions and update their solution knowledge as the environment changes, but they must also conduct a number of trials with their current best-known solution in order to estimate the profit potential of an opportunity (as well as to capture revenue), and these exploration and exploitation trials compete for a resource budget.

In addition to the concept of a forced trade-off between activities, the exploration/exploitation framework characterizes how far-ranging an organization is in its search for solutions. (Indeed there is some disagreement in this literature as to which concept is at the core; for Uotila et al, 2009, for example, exploration/exploitation is about the trade-off, while for Katila and Ahuja, 2002, it is about the breadth and depth of search, dimensions which may be orthogonal rather than zero-sum.) This second conceptualization of exploration and exploitation illuminates a second challenge in trial-based problem solving – how to generate the exploratory trials that are used in the search for solutions. Because solutions combine many elements (new products combine features, technologies, and design; services combine people, places, and information; marketing solutions combine message, medium, price, and so on), exploratory trials may differ from each other on many aspects or only one or a few. Thus, the solution search process is more complicated than the simple evaluation of discrete “options”.

Katila and Ahuja (2002) offer a model of combinatorial search defined by levels of *search scope* and *search depth* and, unlike exploration and exploitation, these are orthogonal variables with no forced trade-off. Search scope refers to the rate or extent to which the problem solver seeks out new, previously-untried solution elements. Search depth refers to the length of time or amount of effort that the problem solver spends re-testing and re-combining the elements that it has decided to try. With no need to trade them off, both variables could be low or both could be high. However, there are some good reasons why we might want both scope and depth in moderation. Too little scope may prevent the problem solver from finding solutions beyond a local optimum, but too much scope can drive up the cost and complexity of knowledge integration, creating more confusion than learning. Too little depth may make evaluation of new solution elements unreliable, but too much depth exhausts the potential for learning and wastes resources better used trying out newer elements. A broad diversity of trials enables the problem solver to probe more alternatives, but less breadth and more repetition enables more reliable inference about the solutions being tested.

The reliability of both inferences (about alternative solutions and about potential profitability for a given problem) is further impaired by the *perishability of learning* in turbulent environments. As the environment changes faster and less predictably, the problem solving knowledge gained from trials degrades increasingly quickly. A trial-based problem solver must strike a balance between unreliable judgments based on too few recent trials, or judgments based on more data but potentially biased by out-of-date learning. In addition to being perishable, trial-based feedback is also likely to be “noisy” when trials are conducted in real markets with live customers rather than in a laboratory. Business intelligence technologies are being used in innovative ways to improve the accuracy and reliability of feedback measurement (cf. Clark & El Sawy, 2010), yet we have no theory as to how these technologies might impact a TBPS process.

Based on these challenges, we identify four research questions that are important to understanding how trial-based problem solving leads to adaptive problem solving performance in turbulence:

1. How does the trade-off of exploration and exploitation in TBPS relate to performance?
2. How do search scope and depth in the generation of trials impact TBPS performance, and is there a necessary trade-off between them?
3. How can a TBPS process manage the perishability of trial-based learning in turbulence?
4. What effects should business intelligence affordances for reducing intrinsic randomness (“noise”) have on a TBPS process?

# 3. Methodology

Our research employs a simulation methodology to develop a new theory of trial-based problem solving. This methodology, articulated by Davis et al (2007), is especially useful in the “sweet spot” between theory-creating research such as case studies, and theory-testing research that uses empirical and statistical methods. The three key activities of the method are (1) developing a theoretical and simulation model based on “simple theory” (scattered theoretical or proto-theoretical ideas from reference literature, case studies, and intuition); (2) verifying that the computational representation of the model faithfully reflects the behaviors predicted by its simple theory premises; and (3) conducting virtual experiments (Carley, 2002) to build new theoretical understanding. These three steps work together to create theory with very strong *internal validity*. By forcing the researcher to commit to explicit definitions of constructs and processes, the first (model building) step transforms what may be fuzzy building blocks into logically precise theory. By confirming that the theoretical mechanisms were programmed as intended, the second (verification) step gives the researcher the ability to say with some confidence that any *other* emergent phenomena of the simulation logically follow from these theoretical premises. The third (experimentation) step takes advantage of this groundwork to probe and explore for trade-offs, nonlinearities, and other behaviors of the simulated system, yielding propositions that can be tested in subsequent empirical research.

The findings are therefore to be interpreted as consequences of the simple theories: *if* *we accept* the simple theories, these propositions *do follow* logically from their interactions as we have modeled them. On the contrary, *if we reject* these propositions, then the simple theories or their interactions require updating. In addition to identifying new propositions, a secondary benefit of combining these three activities in one project is that we can refer back to the model’s assumptions and structure to find explanations for unexpected behaviors. The three steps of this research are reported in §4, §5, and §6 respectively. In §7, we review the new propositions and summarize our progress toward a new theory.

## Method

The simulation model developed in this paper is a *stochastic processes* model, a type which has been used in several exemplars of organization theory (e.g.: March, 1991; Davis et al, 2009; Posen & Levinthal, 2012). Stochastic processes simulations are characterized by the use of computer-generated random numbers to simulate stochasticity, and due to this randomness each simulation run may have a different outcome. Therefore, unlike some other types of simulations, stochastic processes models are typically studied with statistics (at least means and confidence intervals) that aggregate numerous simulation runs, as if they constituted a “sample” of the universe of possible simulation runs. Stochastic processes models are often used when the researcher needs to custom-design a new model because there is no suitable model appropriate for adaptation, as was the case in this research.

All simulations in this study were developed in R and run with R version 2.15.1 (R Core Team, 2012). Two manuals we referred to were Adler (2010) and Jones et al (2009). In addition to the language core, we used the packages “plotrix” (Lemon, 2006) and “gplots” (Warnes, 2012) for generating data visualizations. We are happy to make the simulation code available upon request.

# 4. Model Building

This section sets up a model of a TBPS process based on the insights from relevant literature. It is not the only such model that could be conceived; however, it is meant to be as simple a representation as possible that incorporates a model of environmental turbulence, the key concepts of exploration, exploitation, search scope, search depth, the “how” and “whether” problem solving challenges, and some effects of business intelligence. In §5 we will verify the adequacy of this model for our purposes.

## Solution Trials

Let there be a *problem* characterized by an 15-element array of numbers between 1 and 10, representing requirements that potential solutions must match (for example, the needs or preferences of potential customers), e.g.: {8,9,8,2,6,9,6,5,1,5,9,10,9,3,9}. Let each *solution idea* be defined by a similar array of random values, representing potential *solution elements* that could be combined to execute the idea (for example, features and technologies that may be used in a product, activities and processes that may be incorporated into a service), e.g.: {3,2,5,10,8,1,3,7,5,4,1,6,9,2,4}. Let us define a trial *solution* as a selection of five solution elements.[[1]](#footnote-1) It is easily apparent that some solution elements match the corresponding requirements of the problem very closely and others do not. Let the environment’s response to a trial be a function of the distance between the values of the solution’s elements and the corresponding problem requirements, with numeric values wrapping around so that 1 is adjacent to 10. Figure 1 illustrates the scoring of an example trial. Actual response (the “performance” of a trial) is this score plus or minus some random variance reflecting intrinsic unpredictability or noise in the environment.

|  |
| --- |
|  |
| **Figure 1. Scoring a trial solution** |

The problem solving process is a two-fold search. The problem solver searches for solution ideas that can meet the requirements of the problem well enough to be profitable, and searches for the right solutions (trial executions of those ideas) to realize that potential. The problem is complicated by the fact that the problem’s requirements are unknown to the problem solver, the requirements change over time, and the environmental feedback to trials is noisy, confounded by random variance.

## Environmental Turbulence

We model environmental turbulence by implementing incremental change in the problem’s requirements array each turn. Each dimension has an independent probability of changing by +1 or -1 in each time period.[[2]](#footnote-2) Figure 2 shows how the best possible performance of a solution idea (attainable with a solution constituted of the five closest-matching solution elements, and ignoring noise) changes over 500 turns of turbulence in a single simulation run.

|  |
| --- |
|  |
| **Figure 2. Solution idea potential changes over time** |

We see the best possible performance follows a sort of random walk. Although this particular solution idea is a relatively poor one (worse than the expected median[[3]](#footnote-3)) for most of the time period simulated, its potential performance increases and decreases dramatically a few times during this brief history. This illustrates two important issues in trial-based problem solving in turbulence:

* There is a race against time to capture the profit potential of an idea, as its potential may only last for a limited window of time. Thus a trial-based problem solver must move quickly, and there is a trade-off between thoroughness/accuracy and the need for rapid reaction.
* Solution ideas that do not perform well today may have the potential for profit in the future, so it may be advisable to re-test formerly-rejected solution ideas as the environment changes. Learning is perishable. The value of trial-based data deteriorates with age.

As with solution ideas, solutions themselves (the selections of five elements) will experience changes in performance over time. As the environment changes, ongoing search activities must continue to seek the best possible combination of solution elements, a moving target.

## Problem Solving Search

Three theoretical concept pairs are important in modeling TBPS in turbulence:

* Exploration and exploitation: Problem-solution knowledge is built up through two complementary activities: exploration for new solutions and exploitation of already-known solutions. Exploration is risky but necessary for finding better solutions, while exploitation is necessary to capture the value of existing knowledge. Resources and attention are limited, so the two activities are in competition and it is necessary to determine the optimal balance or trade-off between them (March, 1991; Kim & Atuahene-Gima, 2010).
* Search scope and search depth: Learning from solution trials that test multiple elements in combination is affected by the novelty of the elements and the extent of their repetition across multiple trials. If the problem solver hesitates to test previously-unknown elements, he may miss out on potential breakthroughs, but if he prefers variety too greatly to repetition, the reliability of his inferences will suffer. In the generation of trial solutions, there are synergies from introducing new elements and from testing and re-testing the ones already introduced (Katila & Ahuja, 2002).
* *Whether* and *How* inferences: In a stable environment a problem solver determines first *whether* a problem can be profitably solved, and if so, secondly *how* to solve the problem (Bonabeau et al, 2008). In a changing and uncertain turbulent environment, however, it is necessary to reach both inferences in parallel. The problem solver’s ability to evaluate profit potential depends upon finding effective trial solutions, and both determinations must be updated as the problem environment changes.

### Exploration and exploitation

Two types of solution trials are involved in the trial-based search process. Exploration trials are geared toward inferring which combination of solution elements constitutes the *best* solution to a given problem. The budget for exploration is spent on generating and testing a variety of differentiated trials so that the problem solver can determine which elements contribute the most to performance. Exploitation trials, on the other hand, are geared toward inferring *whether* a solution idea has the potential to be profitable. Exploitation trials take the best known solution (determined by analyzing the feedback to recent exploration trials) and test it repeatedly. Because of the intrinsic unpredictability (noise) in feedback to trials, these repetitions are necessary for accurate estimation of performance of the best known solution, and consequently of the solution idea’s potential.

|  |
| --- |
|  |
| **Figure 3. Exploration and exploitation trials.** |

### Search scope and search depth

Within the exploration trials, search scope and search depth speak to the way solution elements are combined into exploratory solutions and, as a consequence, the tuning of the exploration activity’s inferential power. In order to simulate a range of “tunings”, we focus on an “experimental set” of elements that the problem solver is exploring, including the five believed best elements as well as some others. Search *scope* is a parameter that determines the rate at which new elements are added to the experimental set. Solution elements are retained in the experimental set, and are re-combined to generate exploration trials, for a number of turns determined by the search *depth* parameter before being rejected (with the exception that the five best-performing solution elements are always retained).

|  |
| --- |
|  |
| **Figure 4. Search scope and search depth determine the set of elements under consideration.** |

### Whether and How inferences

In a turbulent problem environment, solutions must be continually adapted, so exploration and exploitation activities must both be ongoing. Exploration trials enable the firm to try out different solution elements and seek to determine which five elements comprise the best solution (*how* to solve the problem), while exploitation trials test that believed-best solution and estimate the (potential) performance of the solution idea (*whether* to solve the problem). Each activity relates to adaptive problem solving performance (i.e. profit) in more than one way, however. Exploitation trials are first of all profitable in themselves (or as close to profitable as is possible for a given solution idea). Secondarily, they enable the problem solver to *estimate* potential profitability and therefore to make accurate decisions about whether to develop a solution idea or abandon it. In both of these functions exploitation is dependent on good information coming from exploration trials. Exploration trials not only increase the profitability of exploitation trials, but by finding the best possible solution elements they increase the accuracy of the problem solver’s decision to keep or abandon an idea. Thus there is a dependency of exploitation on exploration (Garcia et al, 2003).

One implication of this dependency is that there may be good reason for the problem solver to divide the TBPS process into two stages. The first stage would emphasize exploration trials to rapidly find the best combination of solution elements for a given solution idea; while unprofitable (because it involves few exploitation trials) this phase would be time-limited and enable an early and accurate decision about the profit potential of an idea. For those ideas deemed profitable, a second stage would emphasize exploitation activity in order to capture as much of that potential as possible, with minimal exploration to keep the solution knowledge updated. This two-stage model would serve to reduce false positive errors and false negative errors by getting quickly to an accurate decision about idea potential (Bonabeau et al, 2008). I developed two versions of the simulation, a one-stage and a two-stage model, which will be compared in the next section. The two-stage TBPS model’s logic is illustrated in Figure 5.

|  |
| --- |
|  |
| **Figure 5. Process and inferences in a two-stage TBPS model.** |

## Business Intelligence in Feedback Processes

A digital infrastructure serves the critical function of transforming feedback (environmental responses) into inferences that drive the search forward. In for-profit business cases, a necessary metric is the sales response to each trial. Without any sort of business intelligence, feedback in the form of sales could not be easily attributed to one trial or another but would rather come in the form of a lump sum for a day, week, or month (Andersen & Simester, 2011). In that case comparing the performance of different trials (e.g., of product or service ideas) would only be possible by running the tests in separate geographic areas, or by separating them in time. In emerging practice, however, business intelligence technologies are enabling analysts to trace sales back precisely to the individual transaction, time, place, and even the advertisements that triggered them (Clark & El Sawy, 2010). With this kind of digitally-enabled feedback loop, which requires significant integration of back-end systems, a problem-solver’s ability to use TBPS is greatly enhanced. Feedback from the environment can be accurately attributed to each of multiple trials being conducted at the same time in the same markets. [[4]](#footnote-4)

|  |
| --- |
|  |
| **Figure 6. Business intelligence capabilities are the linchpin of the search process’s feedback loop.** |

# 5. Verification of Computational Representation

Davis et al (2007) recommend two steps for verifying a new computational model to be used in simulation research for theory building: first, to verify that the model replicates the behavior predicted by the propositions of its theoretical premises; and second, to confirm that the simulation study’s findings are insensitive to arbitrary choices made in computer programming, such as the starting values of certain parameters. For purposes of brevity, selected highlights of this verification process are reproduced here.

## Verification of Ambidexterity Proposition

March (1991) argued that both exploration and exploitation activity are necessary and the organization’s challenge is to find an effective balance between the two. The so-called ambidexterity hypothesis, that superior performance is achieved with a balance of the two types of activity, has been supported empirically (e.g., He & Wong, 2004; Uotila et al, 2009). Consistent with the literature, our simulations exhibit the following behaviors: (1) exploitation is correlated with mean per-turn performance; (2) exploration is correlated with the variance of performance, and with long-run viability of a solution idea; (3) and total profit per solution idea is maximized with a mix of both activities.

A virtual experiment to confirm the ambidexterity proposition was conducted by holding all parameters constant while varying the activity budget from 1 to 9 exploration trials out of 10 total trials per turn. Figure 7 illustrates the results of this experiment and clearly shows that a mix of exploration and exploitation activity outperformed “extreme” budgets on either end of the spectrum.[[5]](#footnote-5)

|  |
| --- |
|  |
| **Figure 7. Profit per solution idea at different levels of exploration/exploitation.** |

## Verification of Search Scope/Depth Propositions

Katila and Ahuja (2002) present a theory in which search scope and depth are constructs very much like exploration and exploitation but orthogonal, rather than traded-off. Theirs is typical of a number of responses to March’s (1991) theory that do away with the necessity of a competition between the two activities for resources or attention, instead focusing on their synergies (cf. Uotila et al, 2009, who critique this approach). In the simulation, we use the concept of scope and depth to define the strategy by which solution trials are composed as (re)combinations of solution elements. Katila and Ahuja predict inverted-U relationships between both variables (scope and depth) and successful search, and a positive effect of the interaction term[[6]](#footnote-6). If search scope is too high, trials learn too little about too many different potential solution elements, and if search depth is too high, the potential learning about each element is exhausted, so moderate levels of both variables are expected to lead to the best performance. They are expected to be complementary, and mutually reinforcing. We conducted a virtual experiment to look for evidence of the predicted patterns in the simulation model’s behavior.[[7]](#footnote-7)

|  |  |
| --- | --- |
|  | |
| **Figure 8. Performance of exploitation trials at different levels of search scope and depth.** Darker shades indicate better performance. Line graph overlays indicate the best-performing levels of search depth (left) and search scope (right). |

Figure 8 illustrates the findings of this experiment as a single heatmap plot overlaid with two line graphs, the first highlighting the best-performing level of search depth for each level of search scope, the second indicating the best-performing level of scope for each level of depth. Darker shades indicate greater performance of exploitation trials, reflecting effective solution search. With the aid of these line graphs, we do indeed see evidence of the inverted-U shaped functions. On the right, it is clear that the optimal level of search scope is usually somewhere in the middle range of the plotted values, performance generally worse at higher and lower levels of scope. On the left, it appears that performance increases as depth increases up to about 45 turns. To visual inspection alone, however, it is unclear whether there is an upper limit to search depth beyond which sales decline. We also cannot easily see an interaction between the two variables. To probe deeper, we conducted a set of linear regressions on the output of these simulations to test for the predicted relationships. These regressions, reported in Table 2, were conducted in R using the least squares method.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 2. Regression analysis of Figure 8 data.** DV is performance of exploitation trials. | | | | |
| Coefficient: | Model 1 | Model 2 | Model 3 | Model 4 |
| Intercept | 16.09\*\*\* | 15.90\*\*\* | 14.41\*\*\* | 14.22\*\*\* |
| Scope | 4.22\*\*\* | 5.68\*\*\* | 24.46\*\*\* | 25.91\*\*\* |
| Scope2 |  |  | -77.08\*\*\* | -77.08\*\*\* |
| Depth | 0.0297\*\*\* | 0.0358\*\*\* | .0989\*\*\* | .1050\*\*\* |
| Depth2 |  |  | -.0011\*\*\* | -.0011\*\*\* |
| Scope×Depth |  | -0.046 |  | -.0462\* |
| R2 | 0.418 | 0.422 | 0.667 | 0.670 |
| Key to p-values: \* p<0.05, \*\* p<0.01, \*\*\* p<0.001 | | | | |

With 400 data points (each representing the mean performance of exploitation ads in turn 200 of 100 simulations), we find clear and significant nonlinear effects of scope and depth on performance of the exploitation ads. Both coefficients are positive for the linear term, negative for the squared term, the classic inverted-U function. Adding the squared terms increases the R2 dramatically over the linear model with performance as just a function of breadth and depth. In neither the linear nor the nonlinear models did the addition of an interaction term improve the explanatory power of the model. Marginal significance was found for the interaction term in the full model, but its coefficient was negative, not in the predicted direction. In conclusion, we find that the simulation behaves in accordance with two of the three propositions regarding search scope and depth from Katila and Ahuja (2002).

## Verifying Expectations Concerning a Two-Stage Process

We identified the importance of two inferences that the trial-based problem solver makes: *whether* a solution idea has profit potential, and *which* elements constitute the most effective solution. In our simulation model, exploitation activity is used to assess idea potential while exploration activity is used to learn about solution elements. In turbulent environments, unlike stable ones, both types of inferences need to be made in parallel and continuously because the problem never ceases to change. Following from this premise, we suggested that the problem solver may benefit from considering two distinct stages in the TBPS process, a first (idea-screening stage) emphasizing exploration trials to quickly find a good solution, enabling a relatively accurate and early determination of the solution idea’s potential, and a second (profit-sustaining) stage emphasizing exploitation trials to capture as much value as possible from those ideas that are determined to have profit potential. We proposed that a two-stage process would outperform a one-stage process by reducing both false positive and false negative errors. To test this proposition, we developed both one-stage and two-stage versions of the simulation.

The key proposition to be tested was whether a two-stage TBPS process would outperform a one-stage TBPS. To set up the comparison, we needed to optimize each model so that we could make a fair comparison of the best possible performance of the one-stage model to the best possible performance of the two-stage model. This optimization was achieved with virtual experiments that compared numerous combinations of the parameters for exploration/exploitation, search scope and search depth. The optimized parameters are given in Table 3. The performance of each optimized model was simulated (50 simulations of 1000 turns each) and profit-per-turn was compared by way of independent samples t-test, with Welch’s solution for the unequal variances of the two populations (Welch, 1947). The two-stage model outperformed the one-stage model by a wide margin (with a p-value vanishingly different from zero), confirming our expectation.

|  |
| --- |
| **Table 3. Optimized parameters for one-stage and two-stage TBPS simulations.** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  |  | **Two-stage Model** | | |  | **One-stage Model** |  | Idea-screening stage | Profit-sustaining stage | | Exploration | 3/10 |  | 4/10 | 4/10 | | Search Scope | 0.1 |  | 1.0 | 0.05 | | Search Depth | 40 |  | 32 | 28 | |

What was far more interesting was, our expectation that the emphasis would shift from exploration to exploitation across the two stages was *not* confirmed, but the expected transition was seen with *search scope* instead. The optimal idea-screening stage was not distinguished by a higher level of exploration activity, but rather by a high level of search scope. The two-stage model worked best when the subsequent profit-sustaining stage was then rather low in scope, while the one-stage model performed best with an in-between level of scope. The reasons for this are explored in the next section.

# 5. Experimentation to Build New Theory

In this section, we employ additional virtual experiments to explore the trial-based problem solving model more richly and uncover its emergent relationships. Guided by the project’s research questions, we examine the trade-offs of exploration/exploitation and scope/depth in trial-based problem solving, test solutions for managing the perishability of learning in TBPS, and explore the impact of business intelligence technologies that reduce intrinsic noise in the feedback loop.

## The Trade-off of Exploration and Exploitation

We confirmed the ambidexterity proposition, that some balanced level of exploration and exploitation activity would typically outperform the extreme budgets that focus almost exclusively on one of the two activities and computed that the best-performing budget for a one-stage model was 3/10 exploration, 7/10 exploitation (see Figure 7 and Table 3). Given that in this model the problem solver needs to simultaneously explore for the best solution and exploit to capture most of its profit, it was not surprising that the balance somewhat favors exploitation.

What is potentially much more interesting is that in the two-stage variation of the model used in Chapter 5, the best-performing budget in *both* the idea-screening and profit-sustaining stages was 4/10 exploration, 6/10 exploitation. This is contrary to what we expected. Our expectation was that the problem-solver should not seek a profit during its idea-screening stage, but rather focus its resources entirely on the learning process – a focus on exploration in order to make the most accurate possible assessment of idea potential – then shift to an exploitation focus in the profit-sustaining stage. The comparison between Figures 6-2 and 6-3, reflecting two alternative models of the idea-screening stage, illustrates why this expectation didn’t pan out:

|  |
| --- |
|  |
| **Figure 9. Profit and cost of idea-screening stages of length 1 to 50 turns with exploration budget 6/10.** Means based on 1000 simulations. Best expected profit achieved with a decision at turn 14. |

|  |
| --- |
|  |
| **Figure 10. Profit and cost of idea-screening stages of length 1 to 50 turns with exploration budget 4/10.** Means based on 1000 simulations. Best expected profit achieved with a decision at turn 33. |

The key difference between the exploration-focused case (Figure 9), and the exploitation-focused case we deemed optimal (Figure 10) is that in the exploration-focused case the cost of screening increases in almost a straight line over time, cutting deeply into the profit that might have otherwise been earned by exploiting a good solution idea. This confirms the practical problem identified by Bonabeau et al (2008) – that problem solving search is *expensive* – but does not uphold their argument that compartmentalizing it in a search-focused stage necessarily reduces the cost.

Instead, what we see the best-performing firms doing is using exploitation trials from the very beginning of the idea-screening process in order to recoup its costs. Instead of a sharp transition between “research” and “going live”, the TBPS organization creates trials and from the very beginning uses its immediate findings to try to capture as much value as it can. It is quick to carry over its findings into the next round of trials, and after every turn in the simulation its exploitation activities get better. As we see in Figure 6-3, by the time the problem solver transitions to the profit-sustaining stage, the *average* idea is already almost profitable, even before the unprofitable idea candidates are filtered out.

Bonabeau et al (2008) claim that a two-stage model of new product development, with the first stage focused exclusively on problem solving learning, is justified for industries where the cost of product screening is very low compared to the downside risk of investing in bad product ideas. We find here that a slightly different justification for a two-stage problem solving model may hold for those industries where the screening cost is relatively *high* compared to the potential profit of new product or service ideas. In these cases, a process of trial-based problem solving allows the organization to pay for the cost of early trials by rapidly turning tentative insights into better and better trials. This is possible because information technology in the business feedback loop allows rapid learning and iterative rollout of new trials. These IT affordances may make TBPS increasingly viable in what are otherwise marginal industries or niches (high cost of problem solving, low profit potential).

This investigation leads to two inductive propositions for future research:

* **Proposition 1.** In industries where the cost of problem solving search is high compared to the potential profit, trial-based problem solving is more likely to succeed if the problem solver conducts exploration and exploitation trials side-by-side from the earliest stages, using exploration to continually update its profit-capturing activities.
* **Proposition 2.** Information technologies that help the organization to (a) rapidly learn from feedback to trials, and (b) quickly roll out new iterations of product or service solutions in response to this learning, will improve the potential for profit in such marginal industries, or make profit possible where it was impossible otherwise.

## The Trade-Off of Search Scope and Search Depth

We confirmed that Katila and Ahuja’s (2002) first two propositions about search scope and depth are upheld in trial-based, recombinant problem solving search (Table 2): there are inverted-U relationships linking both search scope and search depth to the outcome of successful search (getting to the ideal solution). Both parameters have a positive impact on search up to a point, and then a negative impact. However, in that section we did not investigate the impacts of these parameters on the ultimate outcome of profitable trial-based problem solving.

In order to visualize the nonlinear impacts of search scope, search depth, and exploration budget on total performance in a one-stage model, we conducted a virtual experiment with 900 treatment levels (9 levels of exploration, 10 levels of scope, and 10 levels of depth). Figure 11 shows the impacts of these treatments on total performance. As we can see, the most favorable outcomes occurred where scope was low and depth was high, or the opposite, a visible curve indicating a trade-off with some equifinality (multiple configurations that reach a similar result). This pattern can be explained by two issues that arise when exploration and exploitation activities are conducted side-by-side.

|  |
| --- |
|  |
| **Figure 11. Total performance in turn 200 as function of exploration budget, search scope, and search depth.** 100 simulations at each of 900 treatment levels. Darker shades indicate higher values. |

First, as we just discovered, the strength of a TBPS model may stem from the way it offsets the cost of learning by continually incorporating what it learns into the next round of exploitation trials. Additionally, the exploration trials themselves may vary in how costly they are. At a low degree of search scope, trials mainly re-test the elements of the solution already believed to be optimal, with perhaps one or two novel elements. These exploration trials may perform almost as well as the exploitation activities, even if the novel elements are very poor matches to the problem requirements. On the other hand, at a high degree of search scope, trials may include many new and sub-optimal solution elements, and are more likely to be costly than profitable. Therefore we see that low levels of search scope are generally better, as they allow the problem solver to continually update his knowledge without wasting resources on radical tests of unlikely solutions. “Radicalness” in trials is punished, while “conservatism” is rewarded.

The second phenomenon, the apparent trade-off of search depth and search scope, results from the structural property that the proportion of novel elements in exploration trials depends on the *product* of scope and depth. A low rate of introducing new elements (low scope) is offset by keeping the elements under consideration for a longer time (high depth). If depth is very low, a higher level of scope is needed to achieve the optimal proportion.

When we consider the product-screening stage of a two-stage model, on the other hand, low levels of scope are punished and search depth is almost irrelevant. The best-performing level of search scope in this stage is 1.0, which means that a new solution element is introduced to the experimental set every turn. Because there are only 15 possible solution elements in the simulation, and 5 are in the experimental set when the simulation is initiated, this means that from turn 10 until the “depth” parameter, the problem solver conducts trials with any and all possible solution elements.[[8]](#footnote-8) Trying all possible solution elements does not seem to have a negative effect in the idea-screening stage (refer back to Figure 10), and in fact, we see errors start to increase immediately after the problem solver starts dropping elements from the testing pool after turn 32 (the parameter for search depth).

The biggest difference between the idea-screening and profit-sustaining stages is the change from search scope 1.0 to search scope 0.05. In the profit-sustaining stage, one new solution element is introduced only every 20th time period. In a time-limited first stage, such a low scope of search like would make it impossible to evaluate all the options. Thus, the idea-screening stage should prioritize a high degree of search scope because its priority is to evaluate all options, it is time-limited, and it need only be *almost* profitable. Scope alone is relevant in the idea-screening stage, but the profit-sustaining stage (or one-stage model) must manage the product of scope and depth, because this determines the cost of exploratory search *over time*.

Based on these findings, we identify the following propositions:

* **Proposition 3.** Search scope and search depth taken together determine how radical individual trials will be. At high levels of scope and depth, trials combine a number of new elements in highly variable but individually unlikely-to-succeed solutions. At low levels of both, trials lean toward recombinations of familiar elements with moderate innovations that are unlikely to deviate far from best known performance.
* **Proposition 4.** In a one-stage model of trial-based problem solving, or the profit-sustaining stage of a two-stage model, lower levels of both scope and depth (resulting in more conservative trials) are more likely to sustain performance over time.
* **Proposition 5.** When seeking to maximize learning about solution elements in a short period, as in the idea-screening stage of a two-stage TBPS process, higher levels of scope and depth (resulting in more radical trials) are more likely to achieve the learning goal.

## Coping With Knowledge Perishability

One of the challenges faced by problem solvers as they incorporate more digital technologies into TBPS, and become more reliant on data, is that they must be careful in how they deal with the fact that the feedback to solution trials is perishable. Our model of trial-based problem solving supposes that they may deal with the perishability of this data in two ways: first, a discounting factor is applied such that results of trials from the current turn are weighted more than those from several turns ago (as in an exponentially weighted moving average). Second, the search depth parameter sets the outer limit of time from which results will be used in analysis. If search depth is 40, then results from 41 turns ago will not be used. Furthermore, after a solution element is rejected from the testing pool (after the 40 turns), it may be rediscovered and reintroduced to the testing pool. As Davenport (2009) recommended, the problem solver can re-evaluate solution elements that were formerly rejected, as the changing environment may have made them valuable once again.

Two experiments test the value of these policies. First, we computed a “typical history” of adaptation in the basic one-stage simulation model with the optimized parameters. Figures 12 and 13 illustrate this history as a mean and confidence interval of the problem solver’s proximity to the ideal solution in each turn (100 data points per turn, 350 turns). Lower values are better, as they indicate the problem solver getting closer to achieving the target of his search. The typical pattern is that the process takes up to 100 turns to build up to a characteristic level of performance which it maintains indefinitely thereafter.

In the first experiment, we computed an alternate typical history but removed the discounting factor. Without the discounting, all data from the current turn up to t-minus-depth (40 turns ago) are weighted equally in determining the underlying value of a solution element. As a result, the estimates are therefore somewhat less accurate. Nevertheless, because the organization can “forget” and rediscover solution elements every 40th turn or so, the inaccuracy is constant and does not get worse over time. The thin line in Figure 12 shows the typical history without the discount factor. By comparison, it deviates more from the true-best solution (a higher value is worse performance) but not very far outside the confidence interval for the base model.

|  |
| --- |
|  |
| **Figure 12. 350-turn typical history without the discounting factor (thin red line).**  Compared to the base model (thick line) over 350 turns. |

|  |
| --- |
|  |
| **Figure 13. 350-turn typical history without the ability to revisit formerly-tested solution elements (thin red line).** Compared to the base model (thick line) over 350 turns. |

We see a different pattern in the second experiment, where we have prevented the problem solver from revisiting previously-tested solution elements in composing trials. When the process cannot revisit its earlier trials, it may initially perform *better* than the base model, as we see it does in Figure 13 around turn 100-150, because it does not waste resources re-testing elements *recently* rejected. However, by around turn 200, the problem solver will have tested every possible solution element in the environment, and his ability to continue learning through new trials has essentially ceased. As the demands of the environment change, he finds he can no longer solve the problem effectively.

Whether these experiments reflect real-world phenomena depends on whether we deem it realistic that the problem solver could ever run out of *new* solution elements. If we are modeling a case where a product is gradually superseded by better technology, it may indeed make sense that certain features or designs cease to satisfy the requirements of customers as time goes on. The real-world result will be a product lifecycle like the one seen in Figure 13, where beyond a certain point of time the product idea’s potential must inevitably decline and the problem solver must move on to other ideas.[[9]](#footnote-9) However, in cases where solution elements may rise and fall in profitability indefinitely (for example, in some cases of service design), the decision to never revisit formerly-tested solution elements is clearly a mistake that artificially limits the problem solver’s potential for gain.

Two propositions sum up our findings:

* **Proposition 6.** In a changing environment, even as a problem solver learns from repeated trials, his learning process will be more effective if he weights recent trials more heavily than ones from earlier time periods.
* **Proposition 7.** If the available solution elements may increase in effectiveness over time, the problem solver will sustain adaptive performance over time if he occasionally revisits formerly-tested elements. This may not apply where formerly-tested elements, such as obsolete technologies, cannot become more effective over time.

## Outcome Unpredictability and Business Intelligence

Since trial-based problem solving depends on rapidly and iteratively drawing inferences from feedback to trials in real markets with real customers, the intrinsic unpredictability (or noise) of market response to trials clearly poses a threat to its effectiveness. The greater the irreducible noise in feedback, the less accurately the problem solver can estimate the quality of a solution and the quality of solution elements. Good trials may perform poorly in the market, and poor trials may perform well. To compensate, the problem solver may wish to conduct more repetitions before drawing conclusions about solution elements (greater search depth) or conduct more exploitation trials (a lower exploration budget) to get a more reliable estimate of a solution idea’s potential.

Based on the evidence of case studies, we suggested in §2 that the advent of new business intelligence infrastructures to integrate and analyze data may result in far more accurate (less noisy) feedback to trials. In the case of a direct-response marketing company, for example (Clark & El Sawy, 2010), a data warehouse replaced a simple measure of sales response to an offer (“watching the phones ring”) with sophisticated new key performance indicators (KPIs) linking each trial offer also to returns, re-orders, and other revenues and costs for a far more accurate measure of its bottom-line performance. This amounts to a reduction of noise that can be modeled by altering simulation parameters. We would expect problem solvers with similar business intelligence advantages to conduct fewer repetitions (less depth) and fewer exploitation activities as a result.

To test this intuition, we conducted a virtual experiment based on the ambidexterity proposition, repeating it this time with three different levels of unpredictability. Unexpectedly, we found the opposite interaction of the exploration variable and environmental unpredictability (see Figure 14). We found that in more unpredictable (noisier) environments the problem solver achieves better performance with a *higher* exploration budget, at the expense of exploitation activity. As the figure illustrates, the best-performing level of exploration is about 3/10 in the low- and medium-unpredictability cases and increases to 5/10 in the high-unpredictability case.[[10]](#footnote-10)

|  |
| --- |
|  |
| **Figure 14. Unpredictability/noise impacts on performance at different levels of exploration.** Dashed (blue) lines represent low noise (=2), dotted (purple) lines represent high noise (=6). |

We find an explanation for this result in the path-dependency by which exploitation follows and is informed by upstream exploration (Garcia et al, 2003). In our model, the problem solver’s ability to accurately estimate an idea’s potential with exploitation trials is dependent upon accurate learning through prior exploration trials. Figure 5 anticipated this dependency. In difficult environments, neither exploration nor exploitation will be quite as effective. Due to the direction of dependency, however, increased investment in exploration appears to be the best approach to “damage control”. As technology increases the reliability of measurement, we may expect trial-based problem solvers to use fewer exploration trials and more exploitation trials.

* **Proposition 8:** In environments where feedback is more intrinsically unpredictable (noisy), a problem solver should budget for more exploration trials (trying out new solution elements) as the success of later exploitation depends on effective exploration.

# 7. Toward a Theory of Business Intelligence in TBPS

Through virtual experimentation we have identified eight new propositions that follow from the theory-based model developed in §4 and provide us with a richer understanding of how TBPS works in turbulence. Taken together, these propositions explain that the adaptive problem solving performance (sustained profit over time) of a trial-based problem solving process in a turbulent environment is a function of two constructs which we may call exploration effectiveness and exploitation effectiveness.

Exploration effectiveness could be defined as “the organization’s ability to update its solutions at pace with environmental change at minimal cost through exploration trials”. This definition reflects the need to use exploration trials for ongoing search, but also implies that there is a maximum amount of learning capability needed (determined by the pace of change and the cost), so it is clear that the business challenge is more than simply maximizing adaptive learning. Our findings indicate that the learning impact and cost of exploration trials result from the product of search scope and search depth (Proposition 3), a construct that may be referred to as the radicalness or conservatism of exploration trials. Exploration effectiveness is optimized over time with relatively conservative exploration that achieves constant adaptation but without great financial risk or cost (Proposition 4). However, there are likely advantages to an early, time-limited period of more radical exploration to quickly establish a base of problem-solving knowledge when a new problem or opportunity is first taken on (Proposition 5). Exploration effectiveness depends also upon mechanisms for coping with the perishability of trial-based insights. Two factors that may improve the effectiveness of exploration trials are analytics that weight recent trial results more heavily than older data, either by applying a discounting factor or by setting a moderate level of search depth to give data an expiration date (Proposition 6), and a willingness to re-test solution elements that were formerly rejected as ineffective (Proposition 7).

The second explanatory construct, exploitation effectiveness, may be defined as “the organization’s ability to capture profit over time through exploitation trials” and is strongly impacted by the rapidity with which the organization is able to update its solutions as a result of exploratory learning and to generate new rounds of trials. A faster cycle between action and feedback, supported by new technologies for rapid development of solutions, enables the application of fresher insights to capture emerging opportunities (Proposition 2). Additionally, exploitation effectiveness depends on the organization’s ability to make accurate inferences about idea potential, and therefore make fewer errors in retaining and rejecting solution ideas. By improving measurement of short and long-term impacts of trials, business intelligence improves these inferences. Exploitation effectiveness furthermore depends upon exploration effectiveness (Proposition 1), without which the organization does not have acceptable solutions to exploit. As a result of this dependency we find that in environments where reliable measurement of feedback is difficult, the best way to improve exploitation effectiveness is to conduct more *exploration* activity (Proposition 8).

Figure 15 charts the proposed constructs and relationships in a preliminary theory of business intelligence in trial-based problem solving. The propositions taken together draw our attention to the fact that the search for better solutions and the exploitation of those solutions for profit are two parts of TBPS impacted by distinctly different challenges. A focus on exploration effectiveness shows us that its central challenges are the perishability of learning and the need to make exploration activity cost-effective. A focus on exploitation effectiveness brings into view the need for rapid iteration and for more reliable measurement of feedback. Looking at TBPS this way suggests that the critical implications for research and practice are the needs to discover solutions to each of these challenges.

|  |  |
| --- | --- |
|  | |
| **Figure 15. Preliminary theoretical model of trial-based problem solving** |

# 8. Discussion

As more and more organizations find themselves dealing with environmental change that is not only rapid but also unpredictable and disruptive, interest in using market trials or “smart business experiments” coupled with business intelligence and big data technologies is growing (cf. Andersen & Simester, 2011; Davenport, 2009; Day, 2011; Thomke, 2001). This research project began with the intuition that trial-based problem solving is an organizational mechanism for adaptive problem solving particularly well-suited to the problem of environmental turbulence, and used a simulation methodology to synthesize a theoretical model of TBPS. The resulting theoretical propositions and model are an important contribution to our understanding in two ways.

First, these propositions allow us to differentiate the key challenges within TBPS. This allows us both to inform practicing managers of the areas that need their attention, and to identify key areas for further research. An important task for future research may be to develop reliable measures for exploration effectiveness and exploitation effectiveness in TBPS, and to use these measures to empirically study the antecedents of each. For management practice, the model highlights some under-appreciated design concerns: the need to pace exploration to the rate of environmental change, the primacy of data “freshness” over data “bigness”, and the need to integrate business intelligence with operations to maximize the rapidity with which the business iterates between analysis and action.

Secondly, this research helps us to identify leverage points where information technology can enable and support TBPS processes. In light of this model, the key uses of business intelligence and analytics may be improved measurement capabilities that reduce the apparent unpredictability (noise) in the feedback loop, and better analytical capabilities to make inferences about solution elements and about solution idea potential from small and very recent (fresh) samples of trial-feedback data. This vision of business intelligence clashes with the typical emphasis on data warehousing, mining of past experience, and the more-is-better concept of “big data”.

Future research may build on these beginnings to establish a more mature theory in a number of ways: by developing hypotheses from our propositions and testing them empirically; by enriching our process model with additional theoretical building blocks and discovering new emergent propositions; or by theorizing further about the downstream implications of the propositions developed here. Consider this work a first “trial” toward a theory of TBPS with some “solution elements” that may be valuable in other combinations as well!

# 9. References

Adler, J., *R in a Nutshell*. Sebastopol, CA: O'Reilly, 2010.

Andersen, E. T., & D. Simester, “A Step-by-step Guide to Smart Business Experiments,” *Harvard Business Review,* 2011, 89:3, pp. 98-105.

Argyris, C., & D. A. Schön, *Organizational learning: a theory of action perspective*. Reading, Mass.: Addison-Wesley Pub. Co, 1978.

Bonabeau, E., N. Bodick, & R. W. Armstrong, “A More Rational Approach to New-Product Development,” *Harvard Business Review,* 2008, 86:3, pp. 96-102.

Brown, S. L., & K.M. Eisenhardt, “The Art of Continuous Change: Linking Complexity Theory and Time-Paced Evolution in Relentlessly Shifting Organizations,” *Administrative Science Quarterly,* 1997, 42:1, pp. 1-34.

Carley, K. M., “Computational Approaches to Sociological Theorizing” in J. H. Turner (Ed.), *Handbook of Sociological Theory*, New York: Kluwer Academic/Plenum Publishers, 2002, pp. 69-83.

Carlsson, S. A., & O. A. El Sawy, “Managing the five tensions of IT-enabled decision support in turbulent and high-velocity environments,” *Information Systems and e-Business Management,* 2008, 6:3, pp. 225-237.

Clark, J. W., & O. A. El Sawy, “Business Intelligence at Guthy-Renker: The Promise and Challenges of Sensing the Pulse,” *Proceedings of the SIGDSS/TUN BI Congress II*, St. Louis, 2010.

Davenport, T. H., “How to Design Smart Business Experiments,” *Harvard Business Review,* 2009, 87:2, pp. 68-76.

Davis, J. P., K.M. Eisenhardt, & C. B. Bingham, “Developing Theory Through Simulation Methods,” *The Academy of Management Review,* 2007, 32:2, pp. 480-499.

Davis, J. P., K.M. Eisenhardt, & C. B. Bingham, “Optimal Structure, Market Dynamism, and the Strategy of Simple Rules,” *Administrative Science Quarterly,* 2009, 54:3, pp. 413-452.

Day, G. S., “Closing the Marketing Capabilities Gap,” *Journal of Marketing,* 2011, 75:4, pp. 183-195.

Eisenhardt, K. M., & J. A. Martin, “Dynamic capabilities: what are they?” *Strategic Management Journal,* 2000, 21, pp. 1105-1121.

Emery, F. E., & E. L. Trist, “The Causal Texture of Organizational Environments,” *Human Relations,* 1965, 18:1, pp. 21-32.

Garcia, R., R. Cantalone, & R. Levine, “The Role of Knowledge in Resource Allocation to Exploration versus Exploitation in Technologically Oriented Organizations,” *Decision Sciences,* 2003, 34:2, pp. 323-349.

Gasser, L., & T. Ishida, “A Dynamic Organizational Architecture for Adaptive Problem Solving,” Paper presented at the Ninth National Conference on Artificial Intelligence (AAAI-91), Anaheim, California, July 14-19, 1991.

Gleizes, M.-P., V. Camps, & P. Glize, “A theory of emergent computation based on cooperative self-organization for adaptive artificial systems,” Paper presented at the Fourth European Congress of Systems Science, 1999.

He, Z.-L., & P-K. Wong, (2004). “Exploration vs. Exploitation: An Empirical Test of the Ambidexterity Hypothesis,” *Organization Science,* 2004, 15:4, pp. 481-494.

Jones, O., R. Maillardet, & A. Robinson, *Introduction to Scientific Programming and Simulation Using R*, Boca Raton, FL: Chapman & Hall/CRC, 2009.

Katila, R., & G. Ahuja, (2002). “Something Old, Something New: A Longitudinal Study of Search Behavior and New Product Introduction,” *The Academy of Management Journal,* 2002, 45:6, pp. 1183-1194.

Kim, N., & K. Atuahene-Gima, “Using Exploration and Exploitation Market Learning for New Product Development,” *Journal of Product Innovation Management,* 2010, 27:4, pp. 519-536.

Kota, R, *Self-Adapting Agent Organizations.* Ph.D. Thesis, University of Southampton, 2009.

Lemon, J, “Plotrix: a package in the red light district of R,” *R-News*, 2006, 6:4, pp. 8-12.

Levinthal, D. A., & J. G. March, “The Myopia of Learning,” *Strategic Management Journal,* 1993, 14, pp. 95-112.

Levitt, B., & J. G. March, “Organizational Learning,” *Annual Review of Sociology,* 1988, 14, pp. 319-340.

March, J. G., “Exploration and Exploitation in Organizational Learning,” *Organization Science,* 1991,2:1, pp. 71-87.

Pavlou, P. A., & O. A. El Sawy, “The ‘Third Hand’: IT-Enabled Competitive Advantage in Turbulence Through Improvisational Capabilities,” *Information Systems Research,* 2010, 21:3, pp. 443-471.

R Core Team, R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, URL http://www.R-project.org/, 2012.

Selsky, J. W., J. Goes, & O. N. Baburoglu, “Contrasting Perspectives of Strategy Making: Applications in 'Hyper' Environments,” *Organization Studies,* 2007, 28:1, pp. 71-94.

Teece, D, “Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance,” *Strategic Management Journal,* 2007, 28:13.

Thomke, S. H., “Enlightened Experimentation: The New Imperative for Innovation,” *Harvard Business Review,* 2001, 79:2, pp. 67-75.

Timmis, J., A. Hone, T. Stibor, & E. Clark, “Theoretical advances in artificial immune systems,” *Theoretical Computer Science,* 2008, 403, pp. 11-32.

Uotila, J., M. Maula, T. Keil, & S. A. Zahra, “Exploration, Exploitation, and Financial Performance: Analysis of S&P 500 Corporations,” *Strategic Management Journal,* 2009, 30:2, pp. 221-231.

Warnes, G. R., with R source code and/or documentation contributed by: B. Bolker, L. Bonebakker, R. Gentleman, W. Huber, A. Liaw, T. Lumley, M. Maechler, A. Magnusson, S. Moeller, M. Schwartz, & B. Venables, *gplots: Various R programming tools for plotting data*, R package version 2.11.0, URL http://CRAN.R-project.org/package=gplots, 2012

Weick, K. E, “The Collapse of Sensemaking in Organizations: The Mann Gulch Disaster,” *Administrative Science Quarterly,* 1993, 38:4, pp. 628-652.

Welch, B. L., “The Generalization of ‘Student’s’ Problem When Several Different Population Variances are Involved,” *Biometrika,* 1947, 34:1-2, pp. 28-35.

Winter, S., “Understanding dynamic capabilities,” *Strategic Management Journal,* 2003, 24:10, pp. 991-995.

1. The parameters used here (15 dimensions of problem and solution idea, 10 values, 5 elements of a trial solution) were arbitrary choices not implied by any theory. Other values might have been used, impacting the level of detail and computational demands of the simulation. In the verification stage, we confirmed that the qualitative findings of this research are robust to different levels of these and other arbitrary parameters. [↑](#footnote-ref-1)
2. By default set to 0.04. We also tried simulating turbulence with the addition of “abrupt” change in which a dimension of the problem would probabilistically change to a completely random value. We reached the same conclusions with either model of turbulence. [↑](#footnote-ref-2)
3. Median best possible performance of 4 based on 100,000 simulations with default parameters. [↑](#footnote-ref-3)
4. In these simulations, the problem solver’s analysis of feedback data is fairly simple. Estimates of solution idea potential take an arithmetic mean of the sales response to exploitation trials. For each trial, performance is normally distributed around its true score (match to the environment) with standard deviation set by a parameter (4 in these simulations, except where noted). For exploration, a history is kept for each of the 15 elements, noting when they were used in trials and what the response to each trial was. Those solution elements involved in better-performing trials (on average) tend to be the elements with better true scores, and a simple sort identifies the 5 believed-best elements. As a way of coping with the perishability of learning due to continuous environmental change, results of older trials are weighted less using a discount rate of .08 per turn, and no history is kept beyond the time limit given by the search depth parameter (i.e., a kind of exponentially-weighted moving average).

   Ad hoc testing showed that this algorithm works very well: even with all 15 dimensions in the pool under consideration, perfect ordering of the elements by true score was achieved within about 15 turns with a budget of 10 exploration trials per turn. More sophisticated analyses than these, such as regressions, might have been simulated but are not necessary to theory development in this study. [↑](#footnote-ref-4)
5. All parameters except exploration/exploration were held constant at their default levels, with search scope 0.5 (new solution elements per turn), search depth 5 (turns before rejection). Figure 7 shows the mean result and 95% confidence interval based on 100 simulations for each experimental treatment. Profit was simulated by setting a break-even level (140 by default) and a rejection rule such that the simulated problem-solver continues searching until a 20-turn moving average of performance is below the break-even level, after a 40-turn grace period. We found this to be an effective rule during pilot testing, and small differences (such as a longer or shorter moving average) had no observable effect on the model’s behavior. The justification for the 40-turn grace period was to allow the search process to “ramp up” to the level of its stable over-time performance. Each simulation ended when the solution idea was rejected, or after 1000 turns, whichever was earlier. [↑](#footnote-ref-5)
6. In Katila and Ahuja’s study, the DV is “number of new product introductions” which does not apply here, but what they generalize to is successful search. Sales response to exploitation ads is a DV that captures the same result in our model; measuring effective *search* only, not overall profit. Two of their hypotheses were supported in their empirical study, however, search scope was found to have only a linear and increasing relationship with the DV. [↑](#footnote-ref-6)
7. This large experiment had 400 treatment levels (20 levels of scope, 20 levels of depth) with exploration fixed at 3/10, the best-performing level of that parameter in the previous experiment. For each treatment level, 100 simulations were conducted. The outcome of interest was the mean performance of exploitation trials in turn 200. This we deemed to be a sufficient number of turns to avoid any potential initialization bias. [↑](#footnote-ref-7)
8. These numbers are obviously dependent upon arbitrary specifications of the computer program, but the qualitative phenomena are not sensitive to these parameters. [↑](#footnote-ref-8)
9. Indeed, this finding may even suggest a new explanation for product lifecycles that is endogenous to adaptive problem solving in turbulence. It is an idea worthy of consideration for future research. [↑](#footnote-ref-9)
10. Outcomes are mean total performance in turn 200 from 100 simulations at each experimental treatment, with 95% confidence intervals indicated for the default level of unpredictability. [↑](#footnote-ref-10)