AARHUS UNIVERSITY

Coversheet

This is the accepted manuscript (post-print version) of the article.

Contentwise, the post-print version is identical to the final published version, but there may be differences in typography and layout.

How to cite this publication

Please cite the final published version:

Vuculescu, O. (2017). Searching far away from the lamp-post: An agent-based model. *Strategic Organization*, 15(2), 242-263. DOI: 10.1177/1476127016669869

Publication metadata

Title:	Searching far away from the lamp-post: An agent-based model
Author(s):	Vuculescu, O
Journal:	<i>Strategic Organization</i>
DOI/Link:	10.1177/1476127016669869
Document version:	Accepted manuscript (post-print)

General Rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Searching far away from the lamp-post: An agent-based model¹

Oana Vuculescu

Aarhus University, Denmark

Abstract

This paper presents insights from a laboratory experiment on human problem solving in a combinatorial task. I rely on a hierarchical rugged landscape to explore how human problem solvers are able to detect and exploit patterns in their search for an optimal solution. Empirical findings suggest that solvers do not engage only in local and random distant search, but as they accumulate information about the problem structure, solvers make “model-based” moves, a type of cognitive search. I then calibrate an agent based model of search to analyse and interpret the findings from the experimental setup and discuss implications for organizational search. Simulation results show that, for non-trivial problems, performance can be increased by a low level of persistence, i.e. an increased likelihood to quickly abandon unsuccessful paths.

1. Introduction

A central idea in the problem solving literature is that the process of discovering a solution can be conceptualized as an adaptive search through a space of alternatives (Simon 1962). Research so far has emphasized the clear distinction between local and distant search, focusing primarily on “the where” (local/distant) to the detriment of “the how” (search heuristics) (Lopez-Vega et al.

¹ For the final version of this manuscript, please see Strategic Organization, Special Issue "Organizing Crowds and Innovation", First Published 18 Sep 2016, DOI: 10.1177/1476127016669869

Forthcoming). Less literature deals with search strategies that are non-local and cognition driven (Gavetti and Levinthal 2000). There is also a lack of focus on further identifying individual search behaviours, which are the basic elements of organizational search, and this limits our understanding about how these processes aggregate into macro-level behaviour (e.g. a collective problem solving approach such as crowdsourcing).

Herbert Simon's seminal (1956) work on bounded rationality and the metaphor of problem solving as a search have inspired a large number of organizational scholars and works from the influential March (1991), to organizational learning (Levitt and March 1988) and finally evolutionary economics (Nelson and Winter 2002). One of the most prominent streams of research derived from this tradition relies on the metaphor of search on a rugged landscape i.e. the NK model (Kauffman 1993; Levinthal 1997) as a way to conceptualize how problems of varying complexity pose qualitatively different challenges to a solver.

One difficulty with work built on the NK model is that it can seldom overcome the initial assumptions derived from the canonical model. With notable exceptions (Winter et al. 2007), in most NK variants, experiential search- local search and long jumps (randomly selecting a completely different solution) - are the only types of search that agents can engage in. Research on individual (as opposed to organizational) problem-solving suggests that human problem-solvers have far more sophisticated search strategies (Doll et al. 2012) and rely on a combination of cognitive (reflective) and experiential (reflexive) moves (Gavetti and Levinthal 2000).

However, if employed, the problem representation in the NK models is relatively distant from the empirical phenomenon (Winter et al. 2007). For example, in Gavetti (2005) agents are simply given a set of possible problem representations from which they choose -“a cognitive memory”-, while in Winter et al. (2007) the cognitive influence is “independent of experience” (Winter et al. 2007: p.404).

In this paper, I put forward a different model of adaptive search that allows solvers to make moves relying on problem representations that they themselves have shaped using their gradually accumulated expertise: “model-based” search. This approach is consistent with the idea that problem formulation is not a top-down one-shot process, but rather an adaptive, iterative process where different “models” of the problem are formulated as more information is gathered about the problem structure (Newell and Simon 1972).

To address this research agenda, first, I conducted a series of lab experiments where solvers were asked to play successively two games with payoffs described by the NK and a hierarchical function (Watson and Pollack 1999). Similar to Billinger et al. (2013), the task was to solve a puzzle by generating combinations of two different symbols. Each participant had a limited number of attempts and received immediate and accurate feedback (i.e. their score) after each attempt. The experiment provided a realistic benchmark for the agent-based model parametrization as well as insight into real (as opposed to simulated) problem solving behaviour. Additionally, the experimental setup was constructed to be a one-to-one match with the agent-based model, facilitating the comparison between the two data-sets. Using a family of the same hierarchical functions as a problem space, I use simulations to explore the efficiency of problem solving processes in different settings.

The main contribution of this paper is twofold: first, I identify a distinct type of search behaviour that solvers rely on as they attempt to navigate the problem space. I find evidence that model-based distant search is an important strategy that changes the adaptive search process. The agent-based model derived from the empirical study reproduces well-known results in the innovation management literature and informs the second contribution. Main results from the simulation suggest that, in structured environments, “fail often, fail early” is a reliable problem solving strategy, contingent on problem complexity (Nickerson and Zenger 2004). The remainder of this

paper is organized into four sections: following an overview of previous work in Section 2, in Section 3.1 I describe the experiment and the results from the lab, while Section 3.2. is centred around the model and a number of virtual experiments. Finally, in Section 4, I discuss these results focusing on theoretical and managerial implications.

2. Theory

A relatively large body of theoretical work on problem solving in strategic organization research e.g. Levinthal (1997), Siggelkow and Levinthal (2005), was built around the powerful search metaphor put forward by Newell and Simon (1972), in particular around the NK model (Kauffman 1993) and March's exploration-exploitation model (March 1991). Extensions and variants of these models have been used subsequently to study the dynamics of problem solving.

Thus there appears to be two potential avenues for further development in what concerns the applicability of the model for problem solving: 1) the landscape (i.e. the problem space) and 2) the way the search is conducted (Rivkin and Siggelkow 2007).

One of the limitations of the NK model is that it only allows for single-level interdependence, while most innovation problems can be thought of as hierarchical problems (Yu et al. 2009) with interdependencies *between* levels, as well. A problem space with interdependencies between levels is still rugged, but has discernible patterns (i.e. the ruggedness is not random). The random ruggedness of the NK landscape (Mckelvey et al. 2013) restricts the use of the model further in what concerns implementing more cognitively plausible assumptions regarding search. As mentioned earlier, the problem-solving literature largely focuses on experiential search (cf. Gavetti 2005; Billinger et al. 2013). For example, in Levinthal's seminal paper (1997), search is "non-

greedy local” (imitate the first nearby alternative that is superior) or “non-greedy random long-jump” (imitate the first randomly selected distant alternative that is superior).

Still, trial and error learning is but one of two fundamental cognitive mechanisms that humans engage in. Individuals rely exclusively on trial and error only in very uncertain environments (Doll et al. 2012) and, as they explore, they actually do form “world models” and their subsequent actions are driven by model-based search (Graybiel 2008). Model-based search is thus a mode of search in which agents employ previously acquired information to form so called cognitive maps of the search space and rely on these maps to determine and evaluate future actions. Consequently, model-based search is both *influenced by* and *influences* the way agents perceive their environment.

Imagine the fictitious and stylized task of designing a smart phone. A very simple problem decomposition would identify two modules: screen and battery. These two modules are naturally interdependent: i.e. screen size affects battery life, hence the bigger the screen, the larger the electric output (amps) from the battery should be. This would ostensibly come with a cost in terms of battery size. For a non-trivial problem, an engineer cannot really optimize both components in isolation and *then* match them up. The best possible battery is, we assume, also the smallest possible one and this is not enough to power the best possible screen which should be of high-fidelity and of at least medium size. The NK model allows for this intuition to be formed. Furthermore, it tells us that with more components (N) and more interdependent components (K) it becomes harder for an engineer to figure out the right combination. However, the NK does not tell us the whole story. Going one level lower, batteries are also made up of several interconnected components and there are in fact multiple ways of designing a battery, all optimal, by some criterion. Similarly, there are multiple ways of designing a screen, all perfectly valid and (as the wide variety of smartphones on the market clearly illustrates) all appealing in some way to a different consumer need. Again, following the NK intuition, one engineer would start randomly

with some combination of components and then tweak them one by one, until no more improvements can be made. For large Ns and Ks the task becomes exponentially long. Fortunately, there is an alternative way of solving such problems. Like a chess-master who is no longer concerned with individual positions on the chess board, but rather with configurations of pieces, after a number of failed attempts, our engineer *learns* how viable batteries and screens can be designed. He now has a number of configurations he can choose from trying to simply match the different viable configurations for screens and hardware. There is no need to exhaustively evaluate all possible combinations of screws and bolts that make up a phone, but rather focus his attention on what he has filtered through prior experience as ‘good possible combinations’. This search heuristic bears resemblance to descriptions of new product development teams in Takeuchi and Nonaka (1986). It is an adaptive advantage for solvers to assume some underlying structure about the task environment and attribute a set of interdependencies between the building blocks. As the problem becomes more complex (there is a higher degree of interdependence), these heuristics often fall short and re-decomposition is required (Luo and Knoblich 2007).

Searching far away from the lamp-post

Despite Levitt and March (1988: p.516) initial formulation of organizational learning as “encoding inferences from history into routines that guide behaviour”, a great deal of the early literature focuses on local uninformed search, as highlighted in a recent review by McKelvey et al. (2013). Relying on the NK model, previous research often assimilated exploration with random distant search. Distant search was therefore seen as exclusively random (e.g. “long random jumps” in Levinthal (1997)). It held the somewhat uncertain promise of introducing enough variance into the genetic mark-up of the organization, such that it may lead the search away from a suboptimal peak and towards the global maximum.

However, later developments, in particular with respect to the cognitive micro foundations of organizational change have brought back the spot-light on model-based distant search (Gavetti and Levinthal 2000; Tripsas and Gavetti 2000; Gavetti 2005). This work is informed by cognitive science research which distinguishes between habit and goal directed learning (Graybiel 2008). Psychology and related disciplines have a long tradition of postulating two qualitatively different ways of processing information and decision making: e.g. Kahneman's system 1 and system 2 (2011). The theories and concepts do not necessarily overlap, but there is consensus over a substantial difference to be noticed between reflective and reflexive decision making (Dolan and Dayan 2013). The former takes into account possible prospective courses of actions and consequent actions (i.e. model-based search) and the latter describes a type of search where the assessment of future actions is based exclusively on immediate experience (Doll et al. 2012; Dolan and Dayan 2013). Notably, a number of authors have looked into strategies that could mitigate the pitfalls of relying exclusively on local search and introduced more sophisticated behavioural assumptions. For instance, in an influential paper, Gavetti and Levinthal (2000) endow their agents with partial representations of the landscape that allow a different interpretation of decision-making. As such, while still relying on local search (only looking to nearby neighbours), under his own problem representation, the agent is capable of cognitive search as he searches the landscape guided by imperfect cognitive maps. Furthermore and more significantly for the present argument, the agent is capable of changing its problem representation, but only when his performance is decreasing and only by learning from peers or random change (Gavetti and Levinthal 2000).

These results are complemented by Winter et al. (2007) and Baumann (2010) who suggest that there is a desirable middle level of persistence in the face of failure. Using different assumptions and implementations, these models have in common an attempt to describe the search behaviour of agents who are not dominated in their actions by immediate feedback. This allows their models to

distance themselves from the “ant-like” (cf. Winter et al. 2007) model of search. However, while search behaviour is sometimes informed by cognitive problem representations, these representations are exogenous to the problem solving process.

Problem representation

Model-based distant search assumes that agents are informed in their moves by a representation of the landscape they are navigating through. The agent’s problem representation is thought to determine what and how information is perceived and what further structures are identified (Zhang 1997). However, the relationship between problem representation and the knowledge an agent acquires is not unidirectional. Perfect decomposability of a problem can only occur insofar as the solver has a coherent representation of the underlying structure (Greeno 1977), which cannot happen in early stages of solving so called ill-structured problems. This knowledge is formulated by the solver by gaining familiarity with a problem (Hayes and Simon 1975) and thus problem solving becomes a mix of general search heuristics and task-specific information (problem structure refinement) acquired through trial and error. Problem representations do not just come *ex nihilo*, but rather by gradually accruing experience and increased ability to organize information into larger chunks (Ericsson et al. 2006).

Therefore, it would seem that at least when it comes to ill-defined problems formulating a representation does not *preclude* problem solving, but rather it is part of an iterative, interactive process (Hayes and Simon 1974). The circular relationship between the two is emphasized by Klein (1998) and more recently, in the collective problem solving context by Sieg et al. (2010) who note that the initial problem representation can in fact be refined or even profoundly changed. Returning to the landscape metaphor, by resorting to these mappings, landscapes are reduced, making the search more effective. As illustrated by Kaplan and Simon (1990) nobody searches completely at random for their car keys but resorts to a heuristic to constrain the search space, for example: to

think of the last time you used them (Kaplan and Simon 1990). This heuristic can later on be refined, using information gathered in the search process. This is in line with recent evidence on how human problem solving actually occurs – that is, a mix between experiential and model-based trials (Dolan and Dayan 2013). This perspective explains how acquiring extensive expertise can be both beneficial (Ward 1994) and detrimental (Wiley 1998), since the same expertise can promote fixation and trap the solver in a local optimum.

By focusing on a third type of search behaviour: model-based distant search, I set out in this paper to understand how changing problem representations restructure the search space and how this in turn affects the problem solving process.

3. Research methods

Recent work dealing with search on rugged landscapes cautions against using modelling and simulation in isolation from empirical evidence (Mason and Watts 2012). To mitigate the risk of embedding unsupported assumptions into the agent based model, a laboratory experiment was conducted prior to conducting the simulations.

3.1. Experimental data

3.1.1. Experimental design

The lab experiment is similar to the set-up used in Billinger et al. (2013). The data-collection was designed to be a one-to-one match with the simulation model so that results are easily interpreted in the modelling framework. 200 people were recruited from a pool of participants and asked to come to the lab for one-hour sessions (approx. 40 min of actual playing time).

Participants played two combinatorial games in succession (a hierarchical and an NK² problem, corresponding to the two different functions). Both games were presented as 8-slot puzzles, where the task is to find the combination of two symbols that generates the highest score (Figure 1). A participant could “switch on or off” any of the eight slots to construct a given “solution” (i.e. for any of the eight positions they could set its “value” to be either triangle or square). Once a participant decided on a particular combination, they would “submit” the solution and receive a message informing them of the score associated with that solution. They could then construct a new combination of symbols, based on the feedback they received. At the outset of the game, participants were told that they had 25 attempts (thus 25 submissions) to identify the combination of symbols that would yield the best score. There was no upper or lower limit to how many changes could be made from one submission to the next. In order to avoid additional variation in the search behaviours due to different expectation levels, participants were informed before starting the task of the highest possible score. To insure that the order of the games did not have any significant effects on the search strategies, the order of the two games was randomized. Finally, the participants were informed that their reward would depend on their performance³.

[Insert Figure 1 around here]

The solvers filled in a short survey (testing their understanding of the game instructions) and were afterwards redirected to the browser-games. Players had access to their own previous attempts. Note that both puzzles had each 256 possible combinations with one (NK function), respectively two (hierarchical function) optimal solutions.

3.1.2. Landscape

² I decided to use the NK problem as an additional validity check for the experimental design

³ Ranging from 6\$-24\$

Most problems are nearly decomposable (Simon 1962). In particular, innovation problems are closer to a hierarchical structure (Marengo 2014). In the smartphone example, the optimization of any given module needs to take into account all other modules, but this also “ripples down” the hierarchy. A choice for a particular type of screen is influenced by the choice of battery and vice versa, but it also influences several other interconnected components which make up the screen (e.g. digitizer and LCD panel) and so on. Accordingly, I use in this paper a different implementation of the rugged landscape metaphor, relying on a class of problems known as hierarchical problems.

In the traditional NK landscape, the fitness score is determined by averaging over a number of random draws from a uniform distribution. Unlike the NK function, the HXOR function⁴ is defined by a transformation function (1) and a fitness function (2). The transformation function insures the “validity” of the strings of symbols, while the fitness function assigns the corresponding score. The transformation function is applied recursively to transform the chosen string into one single symbol hereby reducing the effective dimensionality of the problem. The function is given by:

$$h(a, b) = \begin{cases} 1, & \text{if } h(a) = h(\bar{b}) \text{ and } a = 1 \\ 0, & \text{if } h(a) = h(\bar{b}) \text{ and } a = 0 \\ \text{null}, & \text{otherwise} \end{cases} \quad (1)$$

where $\bar{X}$ is the bitwise negation of a string X (e.g. $\overline{01} = 10$).

Based on (1) the fitness function is defined as:

$$f(a) = \begin{cases} 1, & \text{if } a = 1, \text{ or } a = 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Every *correctly* identified pair of attributes is transformed and passed on to the next level and its fitness is added to the overall fitness. Thus a pair 01 contributes to the level of fitness in the lower level with $1+1=2$ and is also transformed into a “higher order” 0 (according to the transformation

⁴ The same function that was used in the laboratory experiment and the simulations

function h) and passed into the next level where it has a contribution of 2 (the sum of its components).

For example, the string 00010010 will yield a fitness of 12, according to the following calculations:

Transformation	Fitness score
0 0 0 1 0 0 1 0	= 8+
- 0 - 1	+2+2+
- -	+Null +
-	+Null = 12

With the basic formula, HXOR is a bimodal landscape, with the two optimal solutions either 0110 1001 or 1001 0110. Forming chunks allows for large distances to be covered quickly – e.g. going from 0110 0110 (fitness 24) to 0110 1001 (fitness 32) would be at the lowest hierarchical level (e.g. without chunking) very hard, but working with chunks of size 4 is quicker – hold one fixed and go through the possible combinations of the other.

The more general formula (3 and 4) for a *biased HXOR* allows the model to introduce landscapes of varying complexity:

$$h(a, b) = \begin{cases} 1, & \text{if } h(a) = h(\bar{b}) \text{ and } a = 1 \\ 0, & \text{if } h(a) = h(\bar{b}) \text{ and } a = 0 \\ \text{null}, & \text{otherwise} \end{cases} \quad (3)$$

Based on (3) the fitness function for a variable a is defined as:

$$f(a) = \begin{cases} 1, & \text{if } a = 1, \\ \text{bias}, & \text{if } a = 0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

As such, for $0 < \text{bias} < 1$, the landscape becomes unimodal and for $\text{bias} = 1$, the landscape is the one already detailed in the earlier paragraphs and the one played in the experimental sessions. Note that *bias* is not equivalent to the *K* parameter in the NK landscape. Experimental results (e.g. Iclazan and Dumitrescu 2007) show that for bias values between 0 and 1, the complexity of the problem is increased since payoffs for 1s are larger than payoffs for 0s, but the optimal solution requires a combination of both. However, a problem with $\text{bias} = 0.2$ is not necessarily “more complex” than a problem with $\text{bias} = 0.3$.

3.1.3. Experimental results

It is a daunting task to make inferences about what happens “between the ears” of the participants and evidently the data collected does not allow for a one to one mapping between recorded behaviour and the variables of interest (i.e. whether the players were engaging in “local search” or “distant search” had to be inferred from the sequence of submitted solutions). To ensure additional validity for the coding of the search behaviours, I also collected anecdotal evidence consisting of 8 semi-structured interviews. These were collected after players finished their sessions and focused specifically on the strategies the players used in their attempt to find the optimal solution (i.e. one of the questions they answered was “what kind of strategies did you engage in during the game”), but also on their reference point (i.e. “how did you construct a new solution”).

I initially constructed a simple operationalization of local search as moves of exactly 1 bit from the reference point, and distant search as everything else. E.g. a succession from $\{0, 0, 0, 0, 0, 0, 0, 0\}$ to $\{0, 0, 0, 0, 0, 0, 0, 1\}$ or to any configuration involving exactly one attribute set to “1” and all other attributes remaining “0” is considered a “local search” move, while all others are considered

“distant moves”. The reference point was the individual’s best performance so far (Billinger et al. 2013) due to the human inherent predisposition to compare future states with the status quo (Kahneman and Tversky 1979). The interviews revealed that, indeed, solvers were paying attention to the solution which gave them the highest score so far.

However, further data analysis, as well as the interviews, reveal that this coding is not as clear-cut as it might seem at a first glance. A number of respondents indicated that although they did judge their *performance* based their best score so far (i.e. “I always benchmarked my score against the solution where I did best, to see if I was improving”), they generated new solutions with their latest submission in mind (“I paid attention to what I did last, since that gave me information about how my moves would affect my score”). Therefore, the coding was adjusted to allow for the reference point to be either the solution giving the player’s best score so far or the latest solution. As expected, the number of distant search moves decreased (See Table 1)⁵. The distribution of strategies is similar to the one found by Billinger et al. (2013), where they report values ranging from 33 to 40 percent for local search, contingent on the problem complexity.

[Insert Table 1 here]

Turning to what determines the two search strategies: the solvers engage in local search progressively as they advance in the game — a kind of exploitation. The probability of engaging in local search is increased if solvers receive positive feedback with respect to the reference point or with respect to their previous move. That is, as long as they are performing above their reference level, the more they have played, the more solvers will be risk averse and engage only in minor tweaks of their solutions. There is also robust evidence pointing at a curvilinear relationship with

⁵ The issue of reference points will be addressed in a future study, but this coding was preferred since it is the most conservative with respect to distant search moves, thus allowing for the smallest number of false positives for the variable of interest.

time: i.e. the solvers were less likely to engage in local search in the last couple of moves. Qualitatively similar results were obtained from fitting the same models onto the NK dataset.

[Insert table 2 here]

Mirroring the results for local search determinants, players seem more likely to engage in distant search, early on in the game. Positive feedback results in a lower probability of solver engaging in distant search, and with a long streak of unsuccessful moves, solvers will be more likely to break off the search and engage in distant search. This seems to be the “exploratory” behaviour expected.

3.1.4. Model-based distant search

But are all distant-search moves the same and are they random? While the experimental setup cannot decisively answer this question, the qualitative interviews as well as the coding show that solvers engage in a more systematic exploration of the search space. Specifically in later stages, solvers do seem to make (on average) less distant moves that violate the underlying problem structure, suggestive of the fact that in later stages they have a better representation of the problem interdependence structure; the number of moves that violate the interdependence structure decreases with time, $p < 0.01$ stdv.=0.002⁶. Such moves, without being “wrong” as such, yield little useful information to a player, since making changes across the interdependence structure confounds the interpretation of feedback. Note that this approximate understanding does involve decomposing the problem, but the patterns in the data alongside the interviews suggest that this is not merely a result of players attempting to manage the complexity of the problem, but rather a result of them being able to form an approximate understanding of the underlying function.

⁶ For example, a move from 1001 0000 to 1011 0001 would be considered such a violation since, according to the transformation function, the correct problem decomposition involves “chunking” the first and last four positions.

Indeed, in the interviews, while some respondents showed an obvious preference for local search (“I kept everything but one symbol constant because that was the only way to understand what impacted what”) others were able to detect and exploit the symmetries of the hierarchical landscape (“I started by playing some different combinations [...] and then [...] by looking how the score changed, I tried to understand the pattern. I was [making moves] trying to see if it was symmetry or adjacency that was [important] ”).

Consequently, I continue by making a further distinction in what concerns distant search moves: model-based search and random search. Consistent with the evidence presented earlier, moves were coded as “model-based search” if they involved distances of exactly 2 in the Hamming space⁷ and if they didn’t involve changes that violate the interdependence structure⁸. This coding means that at any given time step, only twelve out of 255 possible new combinations can be considered “model-based search”. Thus, if players engaged exclusively in local and random search, the odds that a move coded as “model-based search” would be randomly chosen by a respondent are 12/255, yielding a ratio of 4.7% such “model-based” moves. In effect, the ratio is considerably higher. According to this coding the distributions of three search strategies are⁹: model-based search: 17.6%, local search: 37.6% and random search: 44.8%.

This differentiation allows for a further investigation of the two different types of distant search. Random distant search does not seem to be influenced by feedback alone (whether immediate or with respect to the reference point), but primarily by the number of successive unsuccessful trials: the longer the unsuccessful streak, the more likely will be that a player will engage in random

⁷ The Hamming distance is a measure of the number of positions at which the values are different. E.g. a move from (0000) to (0011) corresponds to a Hamming distance of 2.

⁸ Identifying and generating solutions according to the correct interdependence structure is not the only possible way a player could generate “model-based” moves, but this normative coding allows for a precise definition of model-based search.

⁹ Since, according to this coding scheme, a move can only be considered “model-based” after time=2, all reported results refer to the remaining 23 attempts.

search (Table 3). Since no significant relationship exists between the likelihood of engaging in random search and the feedback variables (immediate or with respect to the reference point) this seems to suggest that players will primarily engage in random search only if more such failed attempts are made. Indeed, Table 4 illustrates the relationship between choosing any of the three strategies, with model-based search used as a reference category. The likelihood of engaging in local search, relative to model-based search, decreases with the number of successive unsuccessful trials, while the likelihood of engaging in random search relative to model-based search, increases with the number of successive unsuccessful trials.

[Insert table 3 here]

[Insert table 4 here]

There is no significant relationship ($p=0.23$) between the hamming distance and the number of successive unsuccessful trials in the hierarchical problem. However, such a relationship is found in the NK problem ($p<0.0005$). This further endorses the claim that in a structured landscape, players are capable to detect and exploit patterns and that model-based search and random distant search are two distinct strategies.

The data does not show significant relations between model-based search and any of the variables of interest: feedback, immediate feedback or the number of successive unsuccessful trials. There is, however, similar to the case of local search, to be noted a significant ($p<0.001$) relationship between the type of search behaviour the player was using in his previous attempt. I thus find evidence of “strategic persistence” (Lant et al. 1992), pointing to the fact that, controlling for other factors, a player will be more likely to show some consistency in his search behaviour.

3.2. An agent based model of search

The agent-based model examines how search behaviours are influenced by embedding additional knowledge in the search heuristic and how that in turn affects the likelihood of finding an optimal or near optimal solution. The aim is to explore the interplay between different problem solving strategies, problem formulation and performance. Following the analysis of the experiments, I calibrate the agent-based model, using the relationships and distributions from the experimental data. Therefore, for the parametrization, I used both the descriptive results from the lab experiments as well as the determinants of each strategy to generate rules for agent behaviour.

By running simulations on different model parameterizations, several virtual experiments are conducted in the attempt to identify significant dynamics that can describe the problem solving processes. Specifically, the variable of interest is the maximum performance reached (averaged across multiple runs) in several distinct scenarios.

3.2.1 Problem representation and problem restructuring

One of the reasons behind this particular choice of rugged landscape is the fact that it allows for a simplified, albeit non-trivial implementation of problem representations as well as how they might be formed. As the upper limit of their memory is reached, agents attempt to detect regularities about the landscape, relying on their best attempts so far and the solutions stored in memory. As such, if an agent's best solution so far has "0" and "1" for the first two positions of the solution and all the other solutions he "remembers" have the values of either "01" or "10" for these positions, the two loci are "chunked" (Iclanzan and Dumitrescu 2007). Thus, chunks are formed by attempting to detect regularities in the best solutions the agent has encountered so far. For subsequent model-based distant search moves the agent will only use the values these values as possible values.

[Insert figure 2 around here]

If an agent starts with chunks of size 1, for an 8 bit HXOR of bias=1, all possible strings will yield the same payoff =8 and are as such completely un-informative with respect to feedback. Finding the right solution is rendered to sampling all 256 possible combinations. For chunks of size 2, however, the landscape has only 2^4 local optima, since the agent can optimize at this level of representation all strings of length 2 i.e. “10” and “01” yield better payoffs than “11” and “00”. The task of finding the right solution is easier, albeit there are still 16 possible combinations that, at this level of representation are equally good. Finally, for chunks of size 4, the search is further constrained and the agent needs to navigate through only 2^2 local optima (Watson and Pollack 1999).

This type of problem representation accounts, for example, for competency traps (Siggelkow and Levinthal 2005) in organizational learning as well as a closer mapping between organizational problems and model (Goldberg 2000). A hierarchical structure of the problem also allows for a more cyclical problem solving process to be modelled. As such, the computational agents start “tabula rasa”, but as they explore the landscape, they encode the feedback they receive and form cognitive maps of the space. These imperfect representations are used in subsequent searches and are refined.

[Insert figure 3 around here]

3.2.2. Model-based search

Random distant search occurs in early stages ($\text{time} < 2$) (before any patterns are detectable). *Persistence* is a variable that measures the number of consecutive moves that lead to scores lower than the current best performance an agent has made so far: if at time $t-1$ an agent makes a move that leads to decreasing performance, the variable gets incremented by 1. However, if at time t the same agent makes a move that marks an improvement of his best performance the persistence

counter is set again to 0¹⁰. This parameter captures the variable “successive unsuccessful trials” and was initially set to 3 (the median value obtained from the laboratory experiments).

In later stages ($\text{time} \geq 2$) only if the persistence threshold has been reached, the agent engages in random search. More specifically, if a previous move has generated positive feedback, the agent will be more likely to engage in local search. With a lower probability, if a previous move has generated negative feedback the agent will still engage in local search. Otherwise he engages in model-based distant search, the details of which are explained below.

In this model, the implementation of model-based distant relies on an evolutionary search heuristic developed by Iclanzan and Dumitrescu (2007). Technical details aside, the algorithm relies on a simple associative learning rule and is similar to Hayes and Simon’s (1974) description of problem solving where a solver first begins by creating a problem representation and generates a solution which he tries to improve. When a solver’s memory is full, he uses the information he has collected in the search process thus far to generate a different problem representation.

The agents start without any problem representation (i.e. looking at all possible combinations of bolts and screws, in this case 1s and 0s) so they use their first moves to acquire a basic “understanding” of the problem. After a first random move, agents engage either in local search or in model-based distant search. In their model-based search they attempt solutions that incorporate previously gained knowledge about optimal configurations of the components they have identified as interdependent. E.g. If an agent has identified a correlation between the first and the second loci, those and only those configurations which have been deemed successful in the past are possible model-based moves. If the resulting score is an improvement, they add the respective solution to

¹⁰ “Persistence” is similar to the “moderate obsession” described in Winter, S. G., G. Cattani, et al. (2007). “The value of moderate obsession: Insights from a new model of organizational search.” *Organization Science* **18**(3): 403-419. and it is calibrated based on the empirically derived “number of successive unsuccessful trials”

their memory. Agents receive real feedback after submitting a solution and this feedback is used to compute their best score so far.

As agents engage in different learning actions, they remember the solutions they encounter. The memory an agent has is a fixed ratio of the size of the current problem representation reflecting human capacity limitations within attention and working memory¹¹ (Collins and Koechlin 2012). Once an agent's memory is full, he will attempt to develop his own map of the search space: i.e. which components are interdependent with which.

3.2.3 Simulation results

The following experiments were conducted on various parameter configurations and the search strategies performance was compared both in terms of speed and performance. The eight-bit version of the problem was used, the task, for the computational agents, similar to that of the lab participants was to find the optimal configuration of 1s and 0s for a problem with 8 binary attributes. For each reported result 2000 virtual experiments were conducted to account for the stochastic nature of the model and ensure that results do indeed reflect underlying dynamics of modelled processes. Analyses of variance suggest that 2000 runs were sufficient.

Drawbacks and virtues of local search

The first round of experiments attempted to benchmark the adaptive search described in earlier paragraphs against local search. For simplicity, I refer to this algorithm as *model-based distant search*, but I want to caution the reader that, as explained in the previous section, it describes a

¹¹ For different parametrizations for memory size, except the limit case of size of memory=0, the results detailed below are robust.

complex search algorithm that relies on model-based distant search moves, *alongside* local and random moves.

For smooth landscapes, local search outperforms in terms of efficiency – that is, for *simple* problems, local search operates faster, and reaches the global optimum, even for problems of increasing size¹² (e.g. for a problem of 16 bits: Speed average local search = 56.52; std. dev= 7.12; Speed average model-based = 260; std. dev= 0; sig at $p < 0.001$). In order to make a meaningful comparison between model-based and local search, speed here denotes the number of “steps” and agent makes, not the number of function evaluations.

However, as the ruggedness of the landscape increases, much like in a traditional NK scenario, relying on local search traps the solver in sub-optimal solutions. For increasing size and complexity, an exclusively local strategy (local search) quickly gets stuck in local optimum.

[Insert Figure 4 around here]

As illustrated in Figure 4, in the short run, even for more complex problems ($\text{bias} > 0$) local search quickly improves performance and can outperform model-based distant search. This is because it is relatively easy in hierarchical problems to identify local optima, but without a proper problem representation, the agent is unable to escape those local optima.

But wouldn't a mix of local and random search be enough to account for the results seen in the lab sessions? Virtual experiments suggest the answer is “no”. I have parametrized “a naïve” agent based model (Figure 5) such that local search would occur with increased likelihood if positive feedback had been received, while random jumps would occur if local search steps had not been made.

¹² Since for any bias value, the HXOR function does not generate smooth landscape, with respect to local search, results reported here are from a different function from the same hierarchical family (HIFF).

[Insert figure 5 around here]

A parameter sweep allowed for the identification of the parameters such that the distribution of search behaviours matches the empirical data (a local search to random search ratio of 1:2). Still, for a comparable simulation performance (Simulation performance = 26.93; Average lab performance = 26.86; std. dev.=4.59), the hamming distance is significantly higher than the one found in the lab and close to random (Average hamming distance simulation = 3.95; Average hamming distance lab = 2.7). This suggests that human players are “less random” when it comes to their distant search moves. Indeed, running simulations with the full model yields an average performance that is not significantly different from the empirical results with a comparable hamming distance (Simulation performance = 27.06; std. dev.=5.318, $p>0.05$, average hamming distance = 2.4)¹³.

As such, on a given problem of any level of complexity (i.e. a non-trivial hierarchical problem), applying a simple heuristic, that ignores the structure of the problem yields a search landscape which has many local optima. Allowing for random distant search does improve the performance, but the path towards the solution becomes very long, because, without adequate decomposition, the search involves evaluating numerous solutions.

Persistence and problem representation

To explore previous results further, a batch of simulations was conducted on different levels of “persistence” the agents might have. The answer seems to be quite counter-intuitive: agents who are *less* persistent perform better.

There is no significant difference between the groups with respect to the degree of chunking. When an agent gets stuck on a suboptimal peak it is not necessarily as a consequence of operating with

¹³ As an additional robustness check, I have parametrized a model where each successful move increases the likelihood that the respective strategy will be chosen. In this parametrization agents quickly converge on a “random only” strategy, with an average performance of 25.56 and a hamming distance of 4.

high dimensional chunks because (similar to Baumann and Siggelkow (2013)’s chunky search) the agents do not start with a coarse problem representation, but gradually learn it. Rather, prior successful experience is trapping agents in suboptimal solutions. A computational agent might spuriously correlate the first and the seventh loci of the problem. This in turn will restrict the space of possible model-based moves. Since this correlation is not consistent with the accurate problem decomposition, the feedback is not informative (leading to more potential spurious correlations) and the likelihood of being stuck in a sub-optimal solution increases. Computational agents who are less persistent have an increased likelihood to build a “database” with sufficiently diverse information that allows them to find the accurate solution.

[Insert figure 6 around here]

[Insert figure 7 around here]

These simulations highlight a connection between two previously reported but uncorrelated results: problem representation and persistence (e.g. “moderate obsession” (Winter, Cattani et al. 2007)). For complex problems, less persistent agents (i.e. agents who switch to random search after a lower number of moves that do not outperform their best result thus far), acquire more diverse information and are consequently less likely to form incorrect high order representations of the problem. It is then not that chunking in itself is a good/bad predictor of performance, but rather that performance is driven by developing an accurate problem decomposition. Indeed, running the naïve algorithm across the different types of hierarchical landscapes, reveals that a moderate level of random distant search moves is desirable to obtain the optimal performance. Note that while the underlying logic is the same, the two models allow for different intuitions with respect to the desired combination of search strategies, for low persistence translates directly into a *higher* likelihood of engaging in random search.

[Insert figure 8 around here]

4. Discussion

This study aimed to advance our understanding of how human problem solving processes aggregate and whether they can be reliably designed to successfully solve problems of varying complexity. The overall approach was bottom-up: first, the experiments allowed the study of human problem solving behaviour and second, the simulations attempted to qualify how different search strategies affect the propensity of finding the optimal solution. Even if the model is agnostic with respect to intrinsic or extrinsic motivation, I find that focusing on immediate results (by not engaging in more risky, distant moves) has the potential to diminish learning and the likelihood to solve complex problems.

To begin with, the experimental findings are consistent with previous research on problem solving under uncertainty (e.g. Billinger, Stieglitz et al. (2013)). Consistent with previous work on the balance between exploration and exploitation (March 1988), I find that as solvers are more likely to engage in risk seeking behaviour if they consistently perform below their aspiration level. However, I also distinguish a third behaviour that solvers engage in: model-based distant search. In line with psychology and cognitive science literature (Gläscher, Daw et al. 2010; Doll, Simon et al. 2012), but also organizational literature (Gavetti and Levinthal 2000), I find that solvers do not engage merely in local or random-distant search, but attempt to identify and exploit patterns as they navigate the rugged problem landscape. This different type of distant search is not influenced directly by feedback. Works such as Gavetti and Levinthal (2000) or Winter et al. (2007) have already brought cognition into the spotlight, but to the best of my knowledge, empirical work has not yet focused on the mechanisms or the propensity with which, when solving complex problems, solvers are likely to engage in model-based distant search. Additionally, while this finding is in itself interesting, being able to investigate what determines different types of searching behaviour is

an important first step towards being able to aggregate these behaviours and to identify macro-level patterns.

Consequently, the findings from the experimental study were used in calibrating an agent-based model of adaptive search. The simulations carried allowed for different propositions to be investigated.

First, I explored whether indeed model-based distant search, can be a reliable tool in the search for the optimal solution. Virtual experiments pit local search against the agent-based model search. Results show that local search can be an efficient tool for simple problems and can lead to minor quick improvements for more complex problems. Still, in the case of (even mildly) rugged landscapes, relying on an exclusively local search strategy is inefficient. By relying on (often imperfect) model-based search, agents are able to greatly shorten the paths towards the global optimum as well as to increase their likelihood of finding it.

Secondly, I looked into what determines the success or failure of model-based distant search. “Fail early, fail often” is a phrase that has gathered quite a bit of traction in recent years as the mantra of one of the most successful design companies in the world (IDEO). This idea is captured in the information systems practice (and literature) by agile software development (Martin 2003), scrum (Conboy 2009) etc. The results suggest that a lower level of “persistence” (the willingness to abandon earlier on unsuccessful paths) translates into improved performance. These results add to previous research regarding solving complex problems. As such, results show, low persistence becomes important insofar as computational agents fail to detect an accurate problem representation. Simulations endorse empirical studies suggesting that for complex problems concentrating on immediate performance (and focusing on either local or model-based moves) increases the likelihood that solvers focus too little on learning (e.g. Manso 2011; Gardner 2012), and consequently have an increased likelihood of making spurious correlations which trap them into

suboptimal peaks. This intuition is not perfectly in line with the “moderate obsession” (Winter et al. 2007) results. I argue that while in unstructured environments (or when solvers are unable to form a problem representation) moderate obsession (and its converse: a moderate degree of random exploration) *is* desirable, but in the case of structured environments, solvers often find themselves trapped by their own experience, thus a low degree of persistence is desirable to introduce a sufficient degree of information variability.

Additionally, I put forward a different way to model landscape ruggedness (the HXOR function) that is closer to the near-decomposability which is often described in the Simonian tradition as being a characteristic of organizational problems. In particular, although, in principle, the value of a given attribute depends on the values of all other attributes (thus equivalent to a $K=N-1$, in the NK tradition) the structured nature of the landscape used in this paper allows both solvers and agents to detect and exploit patterns as they advance their search.

Finally, another key contribution of this paper is the agent based model which is the backbone of this study. Overall, this paper contributes to the strategic organization literature and to the innovation-as-search literature in particular by proposing a model that allows for more plausible (and empirically validated) assumptions regarding solver behaviour to be taken into account.

Although these results do not contribute directly to the ongoing discussion about prizes on crowdsourcing platforms e.g. Archak and Sundararajan (2009), they do offer managers a different lens in making that decision. For example, they confirm that in the context of problem solving performance pressure indeed can act like a “double edged sword” (Gardner 2012) and complement work regarding incentive systems for innovation that suggests failure should be tolerated in highly innovative settings (Holmstrom 1989).

A limitation of this study is its focus on *adaptive* search. For solving complex problems, there seems to be evidence that relying on multi-shot submissions (a set-up where solvers receive feedback from the broadcasting organization and can subsequently modify their solutions) is preferable to the classical broadcast search (Vuculescu and Bergenholtz 2014) and certainly there is little controversy regarding the usefulness of feedback in learning environments (Bangert-Drowns, Kulik et al. 1991). The current study is indirectly making the same argument: without having the possibility to receive and incorporate feedback, solvers cannot form an accurate problem representation and their performance depends on their prior knowledge (which is what the traditional Innocentive challenge is trying to exploit). While this is certainly not the norm, there are several instances where online platforms resorted to multi-shot submissions (e.g. the Netflix prize) and, recently, even Innocentive has resorted to setting up two or multiple phases for its challenges. Finally, this study aims to bring forward a novel approach to problem solving research that takes into account the dynamic and complex nature of the phenomenon, while at the same time being informed by empirical research. Previous research on search and problem-solving has been either purely conceptual or relied on either qualitative research or quantitative designs. Using an agent based model offers both the advantages of conducting research that allows for a greater level of detail and answers the “how” question and for a higher degree of generalization (moving away from a particular empirical setting). Furthermore, the model is easily adaptable to other avenues of research. So far I have focused on external search and tailored the model accordingly, but the model could be adjusted to replicate and advance knowledge regarding internal search, organizational learning and even collective cognition.

References

- Archak N and Sundararajan A (2009) Optimal Design of Crowdsourcing Contests. *ICIS 2009 Proceedings*: 200:229.
- Baldwin C and von Hippel E (2011) Modeling a Paradigm Shift: From Producer Innovation to User and Open Collaborative Innovation. *Organization Science* 22(6): 1399-1417.

- Bangert-Drowns R L, Kulik CLC, Kulik JA & Morgan M (1991) The instructional effect of feedback in test-like events. *Review of educational research* 61(2): 213-238.
- Baumann O (2010) Search, Failure, and the Value of Moderate Patience. *Schmalenbach Business Review* 62: 234-259.
- Baumann O and Siggelkow N (2013) Dealing with complexity: integrated vs. chunky search processes. *Organization Science* 24(1): 116-132.
- Benkler Y (2002) Coase's penguin, or, Linux and The Nature of the Firm. *Yale Law Journal* 112(3): 369-+.
- Billinger S, Stieglitz N et al. (2013) Search on rugged landscapes: An experimental study. *Organization Science* 25(1): 93-108.
- Boudreau KJ, Lacetera N, et al. (2011) Incentives and problem uncertainty in innovation contests: An empirical analysis. *Management Science* 57(5): 843-863.
- Brusoni S, Marengo L, et al. (2007) The value and costs of modularity: a problem-solving perspective. *European Management Review* 4(2): 121-132.
- Chen W R (2008). Determinants of firms' backward-and forward-looking R&D search behavior. *Organization Science* 19(4): 609-622.
- Cohen MD and Bacdayan P (1994) Organizational routines are stored as procedural memory: Evidence from a laboratory study. *Organization Science* 5(4): 554-568.
- Collins A and Koechlin E (2012) Reasoning, learning, and creativity: frontal lobe function and human decision-making. *PLoS Biol* 10(3): e1001293.
- Conboy K (2009). Agility from first principles: reconstructing the concept of agility in information systems development. *Information Systems Research* 20(3): 329-354.
- Dolan RJ and Dayan P (2013) Goals and habits in the brain. *Neuron* 80(2): 312-325.
- Doll BB, Simon DA, et al. (2012). The ubiquity of model-based reinforcement learning. *Current Opinion in Neurobiology* 22(6): 1075-1081.
- Eggers JP and Kaplan S (2013) Cognition and Capabilities: A Multi-Level Perspective. *Academy of Management Annals* 7(1): 295-340.
- Ericsson KA, Charness N, et al. (2006) *The Cambridge handbook of expertise and expert performance* Cambridge:University Press.
- Exadaktylos F, Espín AM, et al. (2013) Experimental subjects are not different. *Scientific Reports* 3: 1213.
- Gardner HK (2012) Performance pressure as a double-edged sword enhancing team motivation but undermining the use of team knowledge. *Administrative Science Quarterly* 57(1): 1-46.
- Gavetti G (2005) Cognition and Hierarchy: Rethinking the Microfoundations of Capabilities' Development. *Organization Science* 16(6): 599-617.
- Gavetti G and Levinthal D (2000) Looking forward and looking backward: Cognitive and experiential search. *Administrative Science Quarterly* 45(1): 113-137.
- Gavetti G, Levinthal D et al. (2007) Neo-carnegie: The carnegie school ' s past, present, and reconstructing for the future. *Organization Science* 18(3): 523-536.
- Gavetti G and Rivkin JW (2007) On the origin of strategy: Action and cognition over time. *Organization Science* 18(3): 420-439.
- Gigerenzer G and Edwards A (2003) Simple tools for understanding risks: from innumeracy to insight. *BMJ: British Medical Journal* 327(7417): 741.
- Gläscher J, Daw N, et al. (2010) States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. *Neuron* 66(4): 585-595.
- Goldberg DE (2000) The design of innovation: Lessons from genetic algorithms, lessons for the real world. *Technological Forecasting and Social Change* 64(1): 7-12.

- Graybiel AM (2008) Habits, rituals, and the evaluative brain. *Annual Review of Neuroscience*. 31: 359-387.
- Greeno JG (1977) Process of understanding in problem solving. *Cognitive Theory* 2: 43-83.
- Hayes JR. and Simon HA (1975) Psychological differences among problem isomorphs. *Carnegie-Mellon University, Department of Psychology*.
- Hodgkinson GP, Langan-Fox J, et al. (2008) Intuition: A fundamental bridging construct in the behavioural sciences. *British Journal of Psychology* 99(1): 1-27.
- Holmstrom B (1989) Agency costs and innovation. *Journal of Economic Behavior & Organization* 12(3): 305-327.
- Iclanzan D and Dumitrescu D (2007) Overcoming hierarchical difficulty by hill-climbing the building block structure. *Proceedings of the 9th annual conference on Genetic and evolutionary computation, ACM*.
- Kahneman D (2011). *Thinking, fast and slow*. New York: Macmillan.
- Kahneman D and Tversky A (1979) Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the Econometric Society*: 263-291.
- Kaplan CA. and Simon HA (1990) In search of insight. *Cognitive Psychology* 22(3): 374-419.
- Kauffman SA (1993) *The origins of order: Self-organization and selection in evolution*, Oxford: Oxford University Press.
- Klein GA (1998) *Sources of power: How people make decisions*. Boston: MIT press.
- Lakhani KRH. Lifshitz-Assaf, et al. (2013) Open innovation and organizational boundaries: Task decomposition, knowledge distribution, and the locus of innovation *Handbook of economic organization: Integrating economic and organizational theory*: 355-382.
- Lant TK, Milliken FJ et al. (1992) The role of managerial learning and interpretation in strategic persistence and reorientation: An empirical exploration. *Strategic Management Journal* 13(8): 585-608.
- Levinthal DA (1997) Adaptation on rugged landscapes. *Management Science* 43(7): 934-950.
- Levitt B and March JG (1988) Organizational Learning. *Annual Review of Sociology* 14: 319-340.
- Lopez-Vega HF, Tell, et al. (Forthcoming). Where and how to search? Search paths in open innovation. *Research policy*
- Luo J and Knoblich G (2007) Studying insight problem solving with neuroscientific methods. *Methods* 42(1): 77-86.
- Manso G (2011). Motivating innovation. *The Journal of Finance* 66(5): 1823-1860.
- March JG (1991) Exploration and exploitation in organizational learning. *Organization Science* 2(1): 71-87.
- Marengo L (2014) Representation, search, and the evolution of routines in problem solving. *Industrial and Corporate Change*
- Martin RC (2003) *Agile software development: principles, patterns, and practices*. Prentice Hall PTR.
- Mason W and Watts DJ (2012) Collective problem solving in networks. *Proceedings of the National Academy of Sciences*
- Mckelvey BL. Meng, et al. (2013). Re-thinking Kauffman's NK fitness landscape: From artifact & groupthink to weak-tie effects. *Human Systems Management* 32: 17-42.
- Nelson RR. and Winter SG (2002) Evolutionary Theorizing in Economics. *Journal of Economic Perspectives* 16(2): 23.
- Nickerson JA. and Zenger TR (2004) A knowledge-based theory of the firm—The problem-solving perspective. *Organization Science* 15(6): 617-632.
- Rivkin JW and Siggelkow N (2007) Patterned interactions in complex systems: Implications for exploration. *Management Science* 53(7): 1068-1085.

- Sieg JH, Wallin MH et al (2010) Managerial challenges in open innovation: a study of innovation intermediation in the chemical industry. *R&D Management* 40(3): 281-291.
- Siggelkow N and Levinthal DA (2005) Escaping real (non-benign) competency traps: Linking the dynamics of organizational structure to the dynamics of search. *Strategic Organization* 3(1): 85-115.
- Simon HA (1956) Rational choice and the structure of the environment. *Psychological Review* 63(2): 129-138.
- Simon HA (1962). The architecture of complexity. *Proceedings of the American philosophical society* 106(6): 467-482.
- Takeuchi H. and Nonaka I (1986) The new new product development game. *Harvard Business Review* 64(1): 137-146.
- Terwiesch C and Xu Y(2008) Innovation contests, open innovation, and multiagent problem solving. *Management Science* 54(9): 1529-1543.
- Tripsas M. and Gavetti G (2000) Capabilities, cognition, and inertia: Evidence from digital imaging. *Strategic Management Journal* 21(10-11): 1147-1161.
- Vuculescu O and Bergenholtz C (2014) How to Solve Problems with Crowds: A Computer-Based Simulation Model. *Creativity and Innovation Management* 23.2 (2014): 121-136.
- Ward TB (1994) Structured imagination: The role of category structure in exemplar generation. *Cognitive Psychology* 27(1): 1-40.
- Watson RA and Pollack JB (1999) Hierarchically consistent test problems for genetic algorithms. *Evolutionary Computation* 1999. CEC 99. Proceedings of the 1999 Congress on, IEEE.
- Wiley J (1998) Expertise as mental set: The effects of domain knowledge in creative problem solving. *Memory & Cognition* 26(4): 716-730.
- Winter SG, Cattani G et al. (2007) The value of moderate obsession: Insights from a new model of organizational search. *Organization Science* 18(3): 403-419.
- Zhang J (1997) The nature of external representations in problem solving. *Cognitive Science* 21(2): 179-217.

Figures and tables

Game 1

See instructions again

Solutions

refresh submitted solutions

Solution	Score	Number
□△△△△△□□	10	1

<<<1>>>

Create

Figure 1. Screen-shot of the game

Hierarchical			NK		
Reference	Reference1	Reference2		Reference1	Reference2
Local Search	25.10%	35.20%	Local Search	23.50%	35.50%
Distant search	74.90%	64.80%	Distant search	76.50%	64.50%

Table 1. Distributions of search behaviour (Reference1 refers to the coding which involves the best solution so far, while Reference2 refers to the coding which involves both the best solution so far and the latest solution)

Determinants of local search						
Independent variables	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Feedback(ref=0)	-0.3988*** (0.08291)	-0.6594*** (0.08687)	-0.3254** (0.0967)	-0.2793*** (0.09766)	-0.3008** (0.09856)	
Immediate_feedback (ref=0)		-0.8691*** (0.06461)	-0.8341*** (0.06510)	-0.8795*** (0.06610)	-0.8844*** (0.0638)	-0.8384* (0.06444)
Number of successive unsuccessful trials			-0.06224*** (0.008234)	-0.1141*** (0.009909)	-0.1110*** (0.01001)	-0.1232* (0.09865)
Trial				0.05572*** (0.005618)	0.1758*** (0.02124)	0.1740*** (0.02114)
Trial*Trial					-0.00454*** (0.000771)	-0.00445*** (0.000768)
Constant	-0.2033** (0.07593)	0.5154*** (0.09388)	0.4863*** (0.09401)	-0.02327 (0.1071)	-0.6024*** (0.1462)	-0.8249*** (0.1266)
-2 Log Likelihood	6091.56	5906.76	5845.98	5745	5710	
-2 Res Log Pseudo-Likelihood						20487.86
N	4646	4646	4646	4646	4646	4646

*p<0.05, **p<0.005, ***p<0.001

Models 1-5 are logistic regressions. The last model fitted is marginal logistic regression model: $\log \left\{ \frac{\Pr(LS_{ij}=1)}{\Pr(LS_{ij}=0)} \right\} = \beta_1 + \beta_2 \text{immediate_feedback}_{ij} + \beta_3 \text{trial}_{ij} + \beta_4 \text{unsuccessful}_{ij} + \beta_5 \text{trial}_{ij} * \text{trial}_{ij} + e_{ij}$, where β_1 - β_5 are the fixed-coefficients (intercept β_1) and e_{ij} is the error for observation j of subject i . The within-subject association among the vector of responses is modelled by specifying time as an R-side effect for each participant, with a standard compound symmetric structure. To account for potential autocorrelation in the observations, I included a multiplicative over-dispersion parameter ($\phi \neq 1$). Since -2 Res Log Pseudo-Likelihood is not comparable even among nested models different modelling assumptions with respect to within subjects variance were tested and reported results are robust. Immediate_feedback and feedback are binary variables which capture whether a participant has received negative (coded as 0) or positive (1) feedback with respect to either his latest (immediate feedback) or his best (feedback) submission. The number of successive unsuccessful trials is a continuous variable that captures the number of trials since the latest improve in performance, with respect to the best achieved score so far.

Table 2. Determinants of local search

Determinants random distant serch (RS)				
Independent variables	Model 1	Model 2	Model 3	Model 4
Feedback	0.3547*** (0.08320)	0.6048*** (0.08711)	0.1753 (0.09817)	
Immediate_feedback		0.8301*** (0.06572)	0.7804*** (0.06646)	0.7517 (0.06433)
Number of succesive unsuccesful trials			0.07665*** (0.008559)	0.08367** (0.007655)
Constant	0.2033*** (0.08320)	-0.4830*** (0.09430)	-0.4417*** (0.09453)	-0.3103*** (0.05928)
-2 Log Likelihood	5863.97	5701.28	5614.53	
-2 Res Log Pseudo-Likelihood				-19350.54
N	4646	4646	4646	4646

*p<0.05, **p<0.005, ***p<0.001

Models 1-3 logistic regressions. Model 4 is a marginal logistic regression model:

$\log \left\{ \frac{\Pr(RS_{ij}=1)}{\Pr(RS_{ij}=0)} \right\} = \beta_1 + \beta_2 \text{immediate_feedback}_{ij} + \beta_3 \text{unssuccessful}_{ij} + e_{ij}$, where where β_1 - β_3 are the fixed-coefficients (intercept β_1) and e_{ij} is the error for observation j of subject i . The within-subject association among the vector of responses is modelled by specifying time as an R-side effect for each participant, with a standard compound symmetric structure. To account for potential autocorrelation in the observations, I included a multiplicative over-dispersion parameter ($\phi \neq 1$). Since -2 Res Log Pseudo-Likelihood is not comparable even among nested models different modelling assumptions with respect to within subjects variance were tested and reported results are robust. Immediate_feedback and feedback are binary variables which capture whether a participant has received negative (coded as 0) or positive (1) feedback with respect to either his latest (immediate feedback) or his best (feedback) submission. The number of successive unsuccessful trials is a continuous variable that captures the number of trials since the latest improve in performance, with respect to the best achieved score so far.

Table 3. Determinants of random search.

Multinomial model of search strategies					
Effect	strategy	Estimate	Standard Error	t Value	Pr > t
Intercept	local	11.255	0.08139	13.83	<.0001
Intercept	random	0.7995	0.07913	10.10	<.0001
Number of succesive unsuccessul trials	local	-0.2832	0.03591	-7.89	<.0001
Number of succesive unsuccessul trials	random	0.1272	0.02798	4.55	<.0001
N=4646					
In modeling category probabilities, strategy='Model-based' serves as the reference category.					

In this formulation, the dependent variable has three possible outcomes corresponding to the three possible search strategies: local, model-based and random search. I use the second strategy (model-based) as the reference category. I further model subject variance as a random effect. Thus the model becomes: $\log \left\{ \frac{\pi_{ijr}}{\pi_{ij1}} \right\} = \beta_1 + \beta_2 \text{unssucessful}_{ij} + b_i + e_{ij}$, where $\pi_{ijr} = P(Y_{ij} = r)$ are the response probabilities for individual i to choose strategy r at time j . The random effect b_i is assumed to have a univariate normal distribution with zero mean and compound symmetric covariance matrix.

Table 4. Multinomial model of search strategies

```

IF memory is full
  WHILE there are still chunks not checked
    GO through all chunks from the current best solution (BB(i))
    GO through all other chunks from the current best solution (BB(j))
    IF the set of values corresponding to the chosen chunks are either identical or
    the bitwise negation to the values associated with those chunks stored in
    memory
      FORM new chunk -> BB(i)+BB(j) with the values retrieved.
      Set chunk as checked
    End
    ELSE Set chunk as checked
  End
End
End

```

Figure 2. Pseudo code for the chunking algorithm

```

WHILE the current solution is different from the global optimum OR the number of steps is <25
  IF feedback is positive and  $a > p_1$  OR IF feedback is negative and  $a > 10 - p_1$ 
    Attempt to improve solution by local search
  ELSE IF the number of successive unsuccessful trials > persistence
    Random search
  ELSE
    Model-based search
  End
  Add generated solution to memory
  IF memory is full, attempt to learn new problem structure //chunking algorithm
  End
END

```

Figure 3. Pseudocode for the model-based search algorithm. P_1 is the probability of engaging in local search and a is an integer drawn at each step from a uniform distribution ranging from 0 to 10

Figure 4. Illustrative run ($N=8$, HXOR, bias=1): Local search can outperform model-based distant search on short time intervals (vertical axis= performance, horizontal axis = time steps)


```
WHILE the current solution is different from the global optimum OR the number of steps is <25
  IF feedback is positive and  $a > p_1$  OR IF feedback is negative and  $a > 10 - p_1$ 
    Attempt to improve solution by local search
  ELSE IF feedback is negative and  $b > p_2$  OR IF feedback is positive and  $b > 10 - p_2$ 
    Random search
  Increment the number of steps
END
```

Figure 5. Pseudocode for the naïve model. p_1 and p_2 are the two parameters determined by calibration and a and b are two integers drawn independently, at each step from a uniform distribution ranging from 0 to 10.

Figure 6: Model based algorithm: Average performance at different levels of persistence, for different problem types (bias=1, 0.8, 0.6, 0.4, 0.2, 0)

Figure 7: Model based algorithm: Average number of interdependence violations at different levels of persistence for different problem types (bias=1, 0.8, 0.6, 0.4, 0.2, 0)

Figure 8: Naïve algorithm: Average performance at different levels of persistence for different problem types (bias=1, 0.8, 0.6, 0.4, 0.2, 0)

Author biographical notes:

Oana Vuculescu is a research fellow at the Department of Management, Aarhus University. She recently received her PhD from the School of Business and Social Sciences at Aarhus University. Her research is centered around search as problem solving, with a particular focus on human problem solving strategies and her interests span across cognitive science, computer science and management.