Selectionism and Learning in Complex and Ambiguous Projects

Svenja C. Sommer, INSEAD
Christoph H. Loch, HP Labs and INSEAD

December 16, 2002

Abstract

Project management literature has increasingly recognized that established project management methods work well for projects with moderate complexity and uncertainty, but have limitations in projects with ambiguity (unknown influences; events and actions cannot be planned ahead of time) and high complexity (optimal actions cannot be assessed beforehand).

There are two fundamental strategies to manage projects with ambiguity and complexity: learning and selectionism. Learning involves a flexible adjustment of the project approach to changes in the environment as they occur, rather than at planned trigger points. Selectionism involves pursuing several approaches independently of one another and picking the best one *ex post*.

There are proponents of both approaches, but no comparison between them. We build a model of a complex project with ambiguity, simulating problem-solving as a local search on a rugged landscape. We compare the project payoff performance under learning and selectionism, based on a priori identifiable project characteristics: whether ambiguity is present, how high the complexity is, and how much learning and parallel trials cost. We find that if ambiguity is present and the team cannot run trials in a realistic user environment (reflecting the project's true market performance), learning becomes more attractive relative to selectionism as the project's complexity increases. Moreover, the presence of ambiguity may reverse an established result from computational optimization: without ambiguity, the optimal number of parallel trials increases in complexity. But with ambiguity, the optimal number of trials may decrease because the ambiguous factors make the trials less and less informative as complexity grows.

Key Words: Project management, complexity, ambiguity, selectionism, learning, project infrastructure.

1 Introduction

In May 1999, Cliffs Associates Limited (CAL), a joint venture of Cleveland Cliffs Inc. and Lurgi GmbH, started up a new facility in Trinidad to convert iron ore to pure iron briquettes, using a revolutionary new technology, Circored. But although they had performed extensive risk management for the project, it took them over two years to get the facility running. "The potential problems that we had anticipated did not occur, and the problems that did occur had not been foreseen." Moreover, after the facility finally ran reliably, iron prices dropped to $75/ton, the lowest in recorded history, after the terrorist attack of September 11, 2001. CAL was finally written off in September 2002 (see Loch and Terwiesch 2002).

This example demonstrates a limit of classic project management methods: they rely on anticipating events and carefully planning contingent policies (risk management, or dynamic programming in Operations Management terms). As CAL kept running into unanticipated problems and targets were missed, the pressure mounted and finally, people were exchanged and careers suffered. But in highly novel and complex projects, it is simply infeasible to anticipate all relevant possible events and responses to them. All plans go sour, and the people involved are punished.

This limitation of established project management methods is increasingly being recognized by project management literature (e.g., Williams, 1999; Pich *et al.*, 2002). The use of classic methods for highly novel projects often causes project performance problems (e.g., Morris and Hough, 1987; Wideman, 1992; Tatikonda and Rosenthal, 2000).

The challenge is caused by a combination of *ambiguity* and *complexity*. Ambiguity is defined as the inability to recognize and articulate relevant variables and their functional relationships (Schrader *et al.*, 1993). Complexity stems from "a large number of parts that interact in non-simple ways, ... [such that] given the properties of the parts and the laws of their interactions, it is not a trivial matter to infer the properties of the whole" (Simon, 1969, 195). In particular, a complex system represents a "rugged landscape", in which "adjacent points in the landscape are weakly correlated and local [performance] peaks proliferate" (Rivkin, 2000, 830).

Two fundamental strategies exist to manage projects with ambiguity and complexity: *learning* and *selectionism*. Learning involves a flexible adjustment of the project approach to the changing environment as it occurs, rather than at planned trigger points. This approach has characterized the evolution of many breakthrough technologies, which had to be redefined several times over as much as a decade before they became successful, e.g., Motorola's portable, Corning's fiber optics (Leonard-Barton, 1995; Lynn *et al.*, 1996) or Sun's Java (Bank, 1995). Selectionism involves pursuing several approaches independently of one another and picking the best one *ex post*. Car companies apply this approach in design concept competitions and early prototypes (e.g., Sobek *et al.*, 1999); similarly, advertising agencies and television companies (e.g., MTV) often try out products (Leonard-Barton, 1995).

Thus, CAL could have put a Circored team in place with the competence and power to do technical problem-solving, and make important decisions on the spot, as the characteristics of the facility emerged, rather than insisting on the execution of plans and punishing the team when the plans failed. Of course, this would have required an expensive supporting infrastructure - resources, experienced personnel, and decision power (in the extreme, a pilot facility could have been built. This was not done, due to prohibitive cost). Alternatively, the team could have pursued parallel approaches for the entire facility or for parts of it. Again, this was not done because it would have been very expensive. In spite of its cost, either approach might have brought the facility to stability before the 9-11 price crash, and thus allowed its survival.

It is an unanswered question for project managers as to *which approach to use* - learning or selectionism. There are proponents of both approaches, but no comparison has been made between them. This comparison is the first contribution of this article. (In practice, the two approaches may be combined, but we look at them separately to clarify tradeoffs.)

Secondly, we recommend a *different approach to planning* for an ambiguous project than previous work has done. The philosophy of project management has been to define the state space of what can possibly happen, and then estimate and refine over time the probabilities of all these events. This thinking underlies widely used risk management approaches as well as theoretical work (e.g., dynamic programming or Bayesian updating).

However, for highly novel and complex projects, this *cannot be done*, no matter how large the team tries to draw the state space and how well it does its homework (e.g., Schrader *et al.*, 1993; Pich *et al.*, 2002). Rather, we propose that the team should assess whether ambiguity is present ("are unknown unknowns lurking out there?"), how high the complexity is ("we cannot assess all the effects of our actions"), and how much learning and parallel trials cost. These estimates are typically available at the outset: a project team has at least a good qualitative feeling about them. They allow the team to either plan as before or (with ambiguity or complexity) to set an appropriate project strategy (learning and/or selectionism), with a supporting infrastructure.

2 Literature Review

Highly novel innovation projects often involve both high complexity and ambiguity (Leonard-Barton, 1995; Lynn *et al.*, 1996; McGrath 2001). Project management literature discusses two contrasting approaches to cope with ambiguity and complexity: (1) variety generation with *ex post* selection and (2) iterative adjustments based on newly arriving information. Following Pich *et al.* (2002), we call the two approaches *selectionism* and *learning* (Leonard-Barton (1995) calls them "Darwinian selection" and "product morphing"). The term "learning" refers to Garvin's definition of learning organizations: "A learning organization is an organization skilled at creating, acquiring, and transferring knowledge, and at modifying its behavior to reflect new knowledge and insights" (Garvin, 1993, 80).

Selectionism refers to generating variety (via independent parallel trials) and then choosing the solution with the most favorable outcome. This approach assumes that "success depends upon generating enough variations that at least some will prove *ex post* to yield desirable results" (McGrath, 2001, 118).

Ex post can mean two things: (1) firms generate variations at the prototype level and slowly narrow them down to a final solution (e.g., Ward *et al.*, 1995, Sobek *et al.*, 1999). This approach emphasizes technical complexity. (2) Firms can go as far as introducing multiple product variants into the market (Lynn *et al.*, 1996), emphasizing market uncertainty.

Learning consists of making adjustments based on information obtained during the development process. This requires screening for new information, e.g., by paying attention to the sensation of surprise as a signal of new information (Isenberg, 1984, 88). Firms can also gather information more actively with market prototypes (Leonard-Barton, 1995, 210) or technical prototypes (Thomke, 1998); or they may use "market feedback from the launch of their first product" to improve the next product generation (Leonard-Barton, 1995, 207). Lynn *et al.* (1996) provide empirical evidence for the success of this iterative approach with market introduction.

No guidelines have been proposed for choosing between selectionism and learning. If they are mentioned at all, they are considered in isolation. Leonard-Barton (1995) describes both approaches but does not compare them. Tushman and O'Reilly (1997) simply claim that generating variations is optimal for highly novel innovations early in a product's life cycle. Pisano (1994) empirically finds that laboratory experiments in the pharmaceutical industry are useful only in the case of a "deep theoretical and practical knowledge of the process technology". The aim of the current article is to compare selectionism and learning, based on *a priori* identifiable project characteristics.

We model the development process as a local search on a rugged landscape. In doing so, we follow a number of authors who use local search as an analogy for strategic decision-making (Levinthal, 1997; Rivkin, 2000; Gavetti and Levinthal, 2000). Recent work has also used this analogy in the context of technology development: Kauffman and Lobo (2000) modelled the development of process technologies, and Frenken the innovation network in the aircraft industry (2000) and innovative activities in general (2001).

All these papers use the NK-model, first developed for a biological context by Kauffman (1993). This model is extreme in the sense that problem instances are generated randomly, and thus the causal relationship between action and performance is unknown. In addition, we will look at a complex system in which the causal relationship between action and outcome is well-understood, namely the Travelling Salesman Problem (TSP). By looking at two structurally different cases, our comparison between selectionism and learning is more robust. A novel feature of our model is the explicit representation of ambiguity: the decision maker is completely unaware of some influence variables and thus searches a lower-dimensional projection of the true landscape.

3 A Formal Project Model

3.1 Project Ambiguity

Following Pich *et al.* (2002), we model a project as a performance function $\Pi(\omega', A)$ and a causal mapping $\omega' = M(\omega, A)$. $\omega \in \Omega$ denotes a state of the world. The performance function maps an ending state of the world, $\omega' \in \Omega$, and a network of activities, $A \in \mathcal{A}$, to a project payoff. The ending state, ω', is itself determined by a starting state, ω, and by the network of activities executed. The causal mapping $M : \Omega \text{ x } \mathcal{A} \Rightarrow \Omega$ denotes the effects of the actions on the state of the world.

The state of the world $\omega = (x_1, \ldots, x_n)$ contains all the factors x_i that might possibly influence the project outcome. The activities may influence the state of the world through the causal mapping M. Influence factors (x_i) may include feature requirements, customer tastes, resource costs, competitor intentions and actions, technological issues, regulatory changes, or compatibility issues, to name a few.

The project team does not perfectly know either the state of the world or the mapping. The approach of work to date has been to assume that the team has a fairly accurate approximation of M[1], and to summarize the team's knowledge of the state of

[1]See the discussion in Pich *et al.*, 2002; they call this assumption "transition adequacy" of M.

the world via a probability space $(\Omega, \mathcal{F}, \mathcal{P})$. $\mathcal{F}$ denotes a sigma field that represents all possible events, and P is a probability measure on $\mathcal{F}$, indicating the likelihood of events. If $\mathcal{F}$ is *decision-adequate* (see Marschak and Radner, 1972; Pich *et al.*, 2002), which roughly means that it contains all events that in reality make a difference in the project payoff, the team can apply Bayesian updating of the probability measure P when it receives signals from the environment. Bayesian updating allows the team to apply the powerful methods of *risk management* (Wideman, 1992; see also Dynamic Programming (Bertsekas, 1995) and Partially Observed Markov Decision Processes (Lovejoy, 1991)), to develop optimal action policies, which optimize the *expected* project payoff $E[\Pi(\omega', A)]$.

However, a *complete representation of all possible events is not available* in highly novel or complex projects, the case we are interested in (e.g., Schrader *et al.*, 1993; Pich *et al.*, 2002). Let R be a *projection* that maps the n-dimensional state space Ω into a lower-dimensional space Ω_R of d dimensions: $R : \Omega \rightarrow \Omega_R$, with typical elements $\omega_R = (x_1, x_2, ..., x_d)$, the states of the world that the team can anticipate and plan for. Ω_N is the orthogonal subspace, containing the *ambiguous variables* $\omega_N = (x_{d+1}, ..., x_n)$, that the project team cannot see. Thus, $\omega = (\omega_R, \omega_N)$ (Rudin, 1976, 228).

The projected performance function $\Pi_R(\omega, A) = \Pi((\omega_R, \overline{\omega_N}), A)$ as seen by the project team, is a function of fewer variables. The team is aware of the d-dimensional substate of the world, ω_R, but not of the dimensions in the "ambiguous" space. They are taken as parameters, without being recognized as such. To take a simple example, imagine a team that recognizes two dimensions of the state of the world, (x_1, x_2), which it can set directly (i.e., an action consists of directly setting the state of the world variable). The performance function is a regression curve $\Pi_R(x_1, x_2) = \sum_{i=1}^{2} \alpha_i x_i^{\beta_i} + \epsilon'$. The team does not realize that there are two ambiguous variables, one affecting the sum term multiplicatively, and one being additive. In other words, $\Pi(\omega) = x_3 \sum_{i=1}^{2} \gamma_i x_i^{\beta_i} + x_4 + \epsilon$, and the ambiguous variables are taken as fixed and empirically contained in the parameters $\alpha_i = x_3\gamma_i$, and $\epsilon' = \epsilon + x_4$, without being recognized as variables. The team simply observes (or knows from experience) the connection among the known variables.

3.2 Action Ambiguity, Unk Unks and Complexity

The action set known to the team is a subset $\mathcal{A}_{\mathcal{R}}$, and its effects are recognized only with respect to Ω_R. The risk-neutral team has an associated probability space $(\Omega_R, \mathcal{F}_{\mathcal{R}}, \mathcal{P}_{\mathcal{R}})$ and chooses its actions $A \in \mathcal{A}_{\mathcal{R}}$ to maximize the expected payoff $E[\Pi_R]$. The connection between actions and performance $\Pi(M(\omega, A), A)$ may be so complex that the team cannot find a globally optimal policy of actions, but only a local optimum.

For the comparison below, it is useful to distinguish two ways in which the ambiguous parameters are determined. First, the actions chosen by the team influence not only the known dimensions of the state of the world, ω_R, but also the ambiguous dimensions. In other words, the ω_N may be *unconsciously set* by the project team, which implies that they are determined randomly, not necessarily optimally. We call this *action ambiguity*. For example, a tire cord manufacturer, after a long time of quality problems, discovered that the composition (and thus strength) of the steel threads was influenced by the ambient temperature in the manufacturing facility, which had been set "by default", not in consideration of its effect on product quality.[2]

Alternatively, the team proceeds under implicit and possibly wrong "default" assumptions about the ambiguous dimensions of the state of the world. In effect, the

[2]Taguchi experiments are designed to explore the effects of *known* variables in ω_R, but not of ambiguous variables in ω_N (Phadke, 1989).

team proceeds as if the ambiguous dimensions were set *randomly to values which may differ from the true values that the project will later encounter* in the market environment. This is very common – ambiguity often manifests itself as factors that we take as given, but which are not given. In engineering, a term for this is *unk unks* (from "unknown unknowns").

For example, an Internet startup tried to copy Priceline's reverse auctioning system in Germany. An implicit assumption in the company's business plan was that German consumers would behave in a similar way to US customers in their search behavior (which was never spelled out, nor exposed by initial market surveys). But when they experienced much lower "click-to-purchase" conversion rates, they observed consumers and found that Germans were very reluctant to commit to a purchase without having a feeling for how good their deal was, and were less willing to give a credit card number over the Internet (etc.). The company started to fundamentally revamp the logic of its sales process (allowing customers to step away from the "commitment" after indicating a price, paying via a bill, etc.), but had to cease business before it could fully decipher German consumers' reactions.

3.3 Learning and Selectionism

The project team does not know the complete state space but only (a) the projected state space Ω_R and performance function Π_R, (b) whether ambiguity exists (whether $d < n$), and (c) the complexity of the projected performance function, assuming that this complexity can be *extrapolated* to the full performance function.

In the presence of complexity, the highest performance peaks cannot be identified beforehand (Kauffman, 1993; Gavetti and Levinthal, 2000). The project team can only perform local search and find a local optimum.[3] Local search starts from a random starting event and approach (ω_R, A) and finds a payoff $\Pi_R(M_R(\omega_R, A), A)$.

The team can pursue any number m of parallel local searches (from different starting events, i.e., by pursuing several solution concepts). Without complexity, only one local (and global) optimum exists, and hence all searches will lead to the same solution. If the performance function is very complex, the searches are likely to lead to different local optima.

The team can then, *ex post*, choose the best among the local optima found. For the comparison below, it is useful to distinguish two ways in which the team evaluates the trial results. First, it may perform *laboratory tests* in a controlled and confined environment, in which, indeed, only the projected performance Π_R is revealed (Figure 1, middle). A "lab test" costs c_{lat} per generated solution. As the evaluation is based on the projected performance function, lab tests may prompt the team to choose a solution that is not optimal in the true market.

Alternatively, the team can perform *market tests* in a realistic user environment, at a cost c_{market} per solution (Figure 1, right). This test allows the team to observe the true performance of the trials. The ambiguous influences ω_N are fixed but *correctly observed* (even if implicitly, without exposing the ambiguous variables themselves). Typically, $c_{market} >> c_{lat}$ – the correct identification of true market requirements comes at a price.

Instead of choosing selectionism, the project team may invest c_{learn} to recognize the ambiguous variables $\omega_N \in \Omega_N$. For simplicity, we assume that learning opens

[3] This is well established in literature on computational solutions to complex problems, see, e.g., Ferreira and Zerovnik, 1993; Fox, 1993 and 1994; Jacobson and Yücesan, 1998.

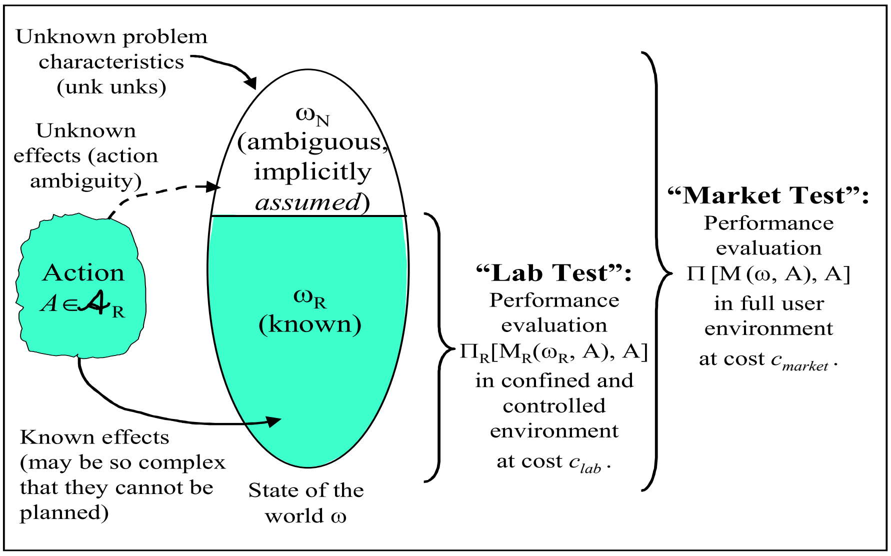


Figure 1: Determination of Ambiguous Parameters

the full state space and performance function to the team (not just an expanded projection). Once the new (previously ambiguous) variables have been recognized, a search over the full set of variables and actions, using the true performance function, may improve performance. Actions may be available to change the new variables, optimizing over the additional dimensions. If the ambiguous variables are *unk unks* fixed by the market environment, their recognition may still change the optimal actions with respect to the project's ending state space.

Clearly, if $c_{learn} >> c_{market}$, selectionism is preferred. Our question is: does the combination of ambiguity and complexity itself have an influence on the choice of learning or selectionism? To answer this question, we look at two opposite situations. In the first, the performance function is represented by Kauffman's (1993) NK-model, which means that the project team has *no meaningful causal mapping* $\Pi(M(\omega, A), A)$ – the effects of any action are essentially unpredictable. The opposite extreme case is a Travelling Salesman Problem (TSP), in which the team has a very good causal mapping of the effects of an action – the effect of a tour change on the total route length can be calculated exactly (although the optimal tour is difficult to find). In both cases, we compare learning with selectionism, while systematically varying complexity.

3.4 Relationship of the Model to Previous Work

One might argue that factors that are "not on the team's horizon" could be incorporated into the sigma field as events with a zero probability. However, this would prevent Bayesian updating as zero probabilities cannot be updated. More fundamentally, it is simply not a useful representation of highly novel projects to suppose that the team can list all the events thought to be impossible. It is the essence of novel projects that "unthinkable" things happen, no matter how diligently the team has planned.

Instead, our model suggests that the project team should give up hoping to have a true representation of all possible events and to maximize an expectation. Rather, the team should *build its choice of a project strategy on its belief of whether or not major ambiguity is present and whether the performance function is complex.* This *can* typically be assessed by the team at the outset. If ambiguity and complexity are important, a combination of learning and selectionism should be applied. In our model, we force the team to choose between the two, as we want to compare them.

4 An Unstructured Project: the NK-Model

4.1 Model Description

As in previous work (Levinthal, 1997; Rivkin, 2000; Kauffmann and Lobo, 2000), we model the performance function as characterized by two parameters: N, the number of actions, and K, the number of interactions among the actions. Increasing K increases the ruggedness of the performance landscape. Action a_i consists of choosing one of S different values: $a_i \in \{1, 2, \ldots, S\}$. All action combinations are allowed, thus, the action network $A \in \{1, 2, \ldots, S\}^N$.

A performance landscape is created by generating individual performance contributions ϕ_i for each of the N actions, which interact: each ϕ_i depends not only on action a_i itself, but also on K other (randomly selected) actions, denoted by $(a_i, a_{i1}, a_{i2}, ..., a_{iK})$. Each performance contribution $\phi_i(a_i, a_{i1}, a_{i2}, ..., a_{iK})$ is drawn from a uniform distribution $U(0, 1)$. Thus, the state space Ω is $[0, 1]^N$, with an N-dimensional uniform distribution as a probability measure. The performance ϕ_i is simply a random redraw for each combination of actions, it cannot be "controlled" by the team. Thus, the NK-model describes a project where market or technology understanding (etc.) are so poor that the team does not know how its actions affect ω' and Pi. The "draw" is different for each action network A, but, *a priori*, all actions look the same. There is no causal model, the performance associated with A is *random*.

The overall performance function is a weighted average of the individual performance contributions: $\Pi(M(\omega, A), A) = \frac{1}{N} \sum_{i=1}^{N} \phi_i(a_i, a_{i1}, a_{i2}, ..., a_{iK})$. For $K = 0$, $\phi_i = \phi_i(a_i)$ and thus the actions are independent of one another. For $K = N - 1$, the performance contribution of one action depends on the values of all the others, and thus $\phi_i = \phi_i(A)$.

We take $N = 10$ as fixed and consider *one* ambiguous dimension, action N. We assume that the team correctly observes the complexity of the problem K. The projected performance function is $\Pi_R(M(\omega, A), A) = \Pi(M((\omega_R, \omega_N), (A_R, A_N)), (A_R, A_N))$ $= \frac{1}{N} \sum_{i=1}^{N} \phi_i(a_i, a_{i1}, a_{i2}, ..., a_{iK})$. The ambiguous dimension may be either an unk unk, fixed by the market environment, or an ambiguous action, set unconsciously by the team. When the team performs a lab test facing unk unks, it assumes the unk unk to obey an implicit default value: a_N is a random variable with $Prob(a_N = l) = \frac{1}{S}$, where $l \in \{1, 2, \ldots, S\}$.

Local search is performed by varying one action dimension at a time, taking the first a_i that leads to an improvement of the projected performance function. Once no such action exists, a local optimum has been found.[4]

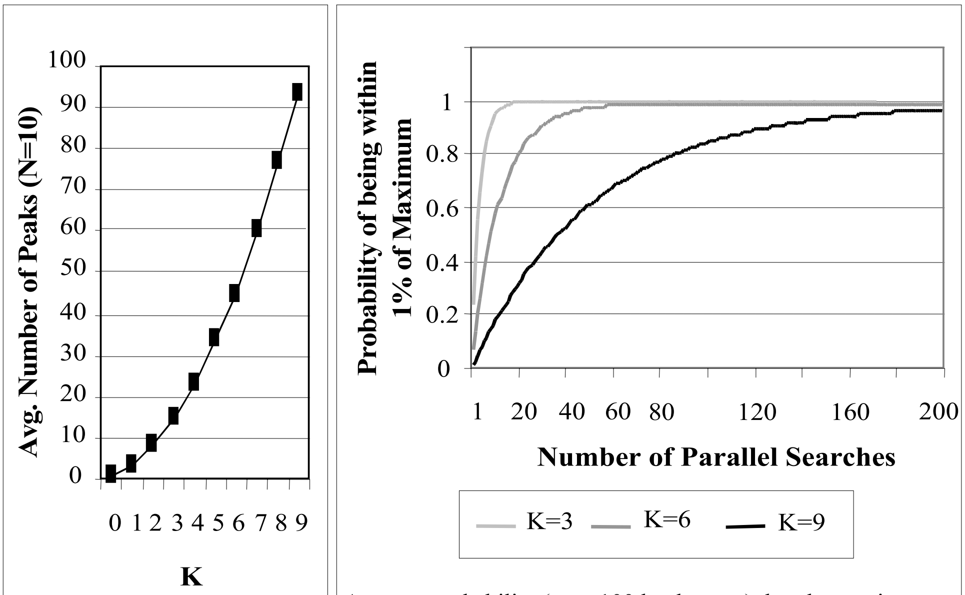
Gavetti and Levinthal (2000) also used a lower dimensional representation of the true performance function in the NK-model in order to symbolize cognition. Rather than working with a projection, they assigned each point in the cognitive representation "a fitness value equal to the average value of the set of points in the actual fitness landscape that are consistent with this point" (p. 121), i.e., by setting $\Pi_R(M(\omega_R, A), A) = \frac{1}{|A_N|} \sum_{a_N \in A_N} \Pi(M((\omega_R, \omega_N), (A_R, A_N)), (A_R, A_N))$. Thus, Gavetti and Levinthal's decision maker knows all the dimensions, which is reasonable in their context. Our model, in contrast, has the critical feature that some variables are completely unknown to the project team. Averaging is impossible; the ambiguous actions a_N are (unconsciously) fixed.

4.2 Complexity of the NK-Model

The complexity of an NK landscape is typically measured by the average number of peaks (left panel of Figure 2; see also Gavetti and Levinthal, 2000; Rivkin, 2000). As

[4] We could, of course, use a more sophisticated search, e.g., varying 2 or 3 actions at a time. However, as long as this is still local search, which does not find the global optimum, our results will remain the same.

a multi-peaked function is not very complex if the peaks are of equal height, we add a second measure that includes the peak height variance (right panel in Figure 2). It is the probability of finding a value within 1% of the global optimum for a given number of parallel searches. Increasing K increases both complexity measures in Figure 2.



Source: Gavetti and Levinthal, 2000

Average probability (over 100 landscapes) that the maximum value found in a given number of parallel searches (from random starting points) is within 1 % of the global maximum.

Figure 2: Complexity in the NK-Model

4.3 Optimal Selectionism in the NK Model

We compare selectionism and learning by simulating their respective project payoffs (averaging over 10,000 runs within one landscape and 100 simulations across landscapes). We consider $N = 10$, one ambiguous action a_N, $S = 2$, and vary the complexity K of the project.[5] We examine how the optimal number of parallel trials depends on the level

[5] We had to set $S = 2$ for computational reasons. For a realistic large S, it is very unlikely that the unk unk is unconsciously set to the true value. To approximate this effect in our simulation with $S = 2$, we set the ambiguous action in the search environment to the "wrong" value.

of complexity and the presence of ambiguity.

For any positive search cost, an optimal number of parallel searches exists, which decreases as runs become more costly. This is true because the local optimum found in a given run is an i.i.d. random draw among the same set of all local optima (Levinthal, 1997, 941). The expectation of the maximum of m i.i.d. random variables is concave increasing in m, while search costs grow linearly in m. Thus, there is a unique optimal number of trials.[6]

For $K = 0$ (no complexity), the team should optimally perform no more than *one* local search. As each variable's contribution $\phi_i = \phi_i(a_i)$ is independent of the others, the problem becomes separable. All parallel searches lead to the same global optimum, and the search cost should be incurred only once.

If the project faces an unk unk (the ambiguous action is set by the market environment), the team can choose between market tests and lab tests to evaluate their selectionist trials. In the face of action ambiguity, both types of evaluation lead to the same result. Hence, we will consider only lab tests for action ambiguity.

We turn to unk unks first. Market tests evaluate the solutions with respect to the true performance function. In this case, the optimal number of parallel searches increases with complexity, both with and without ambiguity (Figure 3). This is consistent with well-kown results from computational complexity theory (e.g., Fox, 1993 and 1994; Rivkin, 2000, 835). There are more and more low peaks, and therefore, more parallel searches are needed to find a value close to the global optimum.

The optimal number of searches is higher when ambiguity is present. Without ambiguity,

[6] Clark (1961) showed this for several specific distributions. We confirmed concavity in our model with simulations.

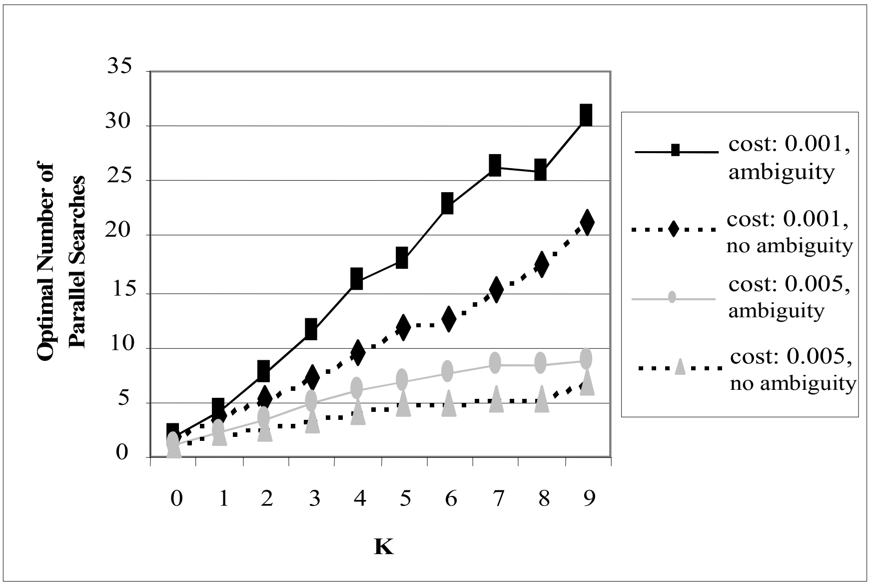


Figure 3: Optimal Number of Parallel Searches with Market Tests in the NK-Model

each search finds a maximum in the true performance landscape. But with ambiguity, the solutions found in the projected landscape might not be maxima in the true landscape, which provides more improvement potential for additional trials.

If lab tests are carried out to evaluate the trials, the best one is chosen in the projected performance landscape, possibly rendering the choice incorrect. Here, the optimal number of parallel searches first increases with complexity, but then decreases (Figure 4). This is due to two conflicting effects. On the one hand, increasing complexity requires more parallel searches to find a good maximum trial. However, increasing complexity also makes the lab results less and less correlated with the true results (in the full landscape), because an incorrectly assumed unk unk influences an increasing number of performance contributions ϕ_i. This decreases the value of additional searches. We can show analytically that when $K = N - 1$ and S is high enough, there is no correlation between the lab tests and the true performance function, and thus, only *one* local search should be performed (the formal result is shown in the Appendix).

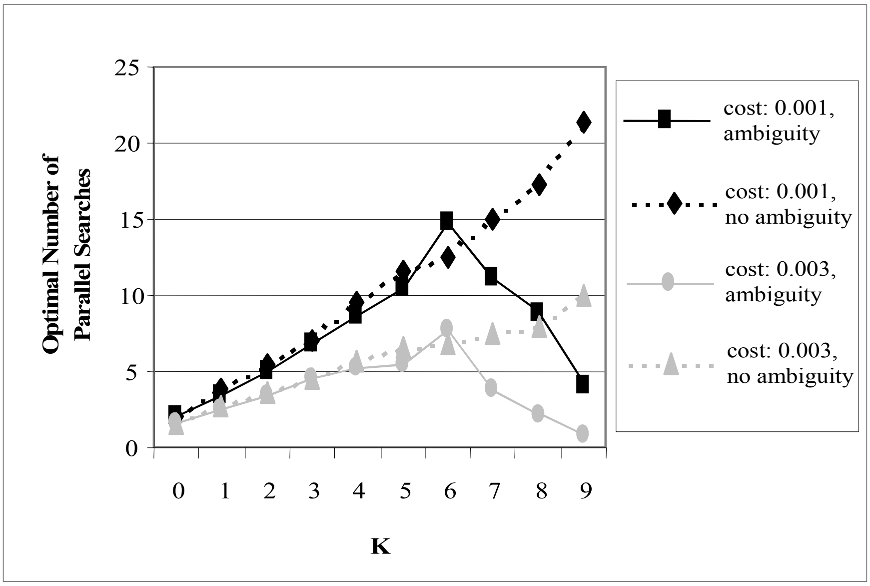


Figure 4: Optimal Number of Parallel Searches with Lab Tests

As long as complexity is low, lab tests exhibit almost no difference in the optimal number of parallel searches between the presence and absence of ambiguity (Figure 4). A good solution in the projected performance function also leads to a good solution in the true performance. With increasing complexity, this correlation drops, making parallel searches less and less interesting. The optimal number of parallel searches in the lab is always lower than in the market. The reason for this is that, via lab tests a trial is chosen based on an incorrect performance function, thus reducing the expected search benefit.

So far, we have discussed ambiguity in the form of unk unks. If, instead, the team faces action ambiguity (action N is chosen unconsciously), the differentiation between market and lab tests disappears – the team now does not assume incorrect parameter values. In this case, higher complexity increases the optimal number of parallel searches, both with and without ambiguity.

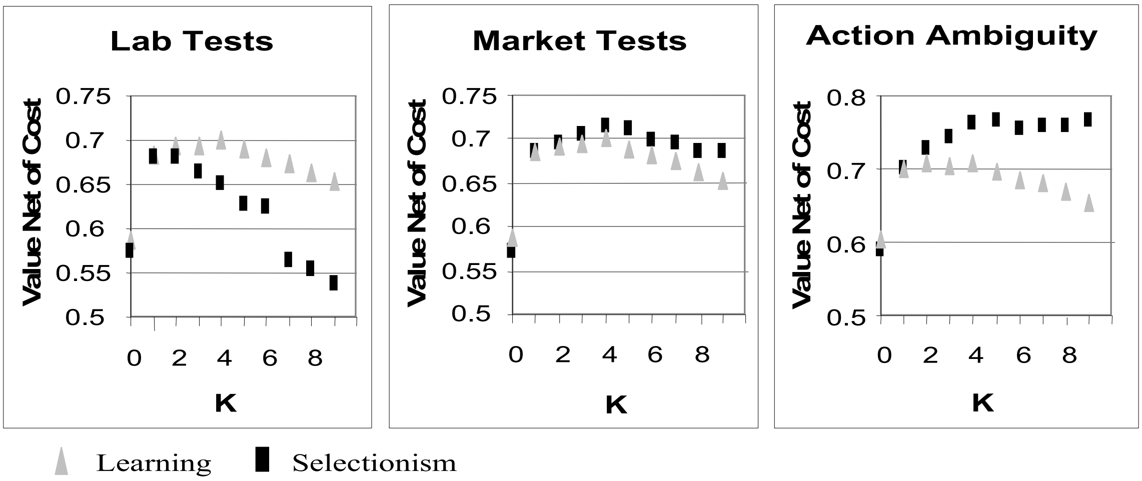
Action ambiguity *reduces* the optimal number of parallel trials. Note the contrast with unk unks: with action ambiguity, the project team searches a lower dimensional performance landscape (than without ambiguity), whereas with unk unks, the evaluation and choice of parallel market trials happens in the full landscape (although a trial may not find a true local peak). Thus, the trials with action ambiguity *de facto* solve a less complex problem when ambiguity is present. Hence, fewer trials are needed to find a value close to the global optimum of this lower dimensional search space (which is not necessarily the global optimum of the full space).

4.4 Selectionism versus Learning in the NK Model

We now turn to the comparison of learning and selectionism. We have already discussed the fact that relative costs have an influence – if the cost of learning is high, selectionism is preferred, and *vice versa*. The comparison below is "net of costs", with the interesting result that learning is especially advantageous when the team cannot correctly evaluate the true performance function.

If the team has market trials at its disposal in the face of unk unks, it *can* evaluate true performance, and learning is advantageous only if it is cheap or if the complexity is low (Figure 5, middle). In the case of action ambiguity, the value of learning is low, particularly at increasing levels of complexity (Figure 5, right) because a single search easily gets stuck in a low local optimum, even if one knows all state dimensions.

However, learning is a powerful alternative if the team cannot evaluate the true performance (has to to use lab tests to check for unk unks). While the project payoff resulting from both learning and selectionism decreases as complexity grows (not surprisingly, as the project becomes more difficult with complexity), the selectionism payoff deteriorates faster, and learning offers a growing advantage. At the extreme ($K = N - 1$), selectionism corresponds to a random pick in the landscape, while learning at least finds a random local peak. The reason for this is that with increasing complexity, the trials become less representative of the true performance, as the ambiguous action affects the optimal choice of more and more known decision actions.



Parallel search costs have been taken out of the performance comparison - it is "net of costs". However, costs still play an indirect role by influencing the optimal number of selectionist runs. We have assumed the lowest cost in our range (c=0.0005) to make the comparison most favorable to selectionism (more runs increase the expected performance of the best run; because of concavity, an even lower cost ceases to improve the selectionist performance).

Figure 5: Comparison of Learning and Selectionism

5 A Structured Project: the Symmetric TSP

5.1 Model Description

We now turn to a project that is in contrast to the NK-model described in Section 4, a project in which the team has a good causal model of the performance effects of its actions. An example of such a structured project is the Travelling Salesman Problem (TSP): the team has to develop a tour that visits a number of "cities" (or generally, places) at the lowest cost. Ambiguity is present if cities emerge that the team was not aware of at the outset.

The symmetric TSP is described by an undirected graph $G = (N, E)$ with n nodes and cost $c(e)$ for every edge $e \in E$. We generate it by randomly placing cities in a five unit square, according to a uniform distribution. Cost $c(e)$ is the Euclidean distance between the two respective cities. We vary the complexity by changing the number of cities.[7]

There is no uncertainty in the TSP (the performance of a given tour is known), only complexity and ambiguity. Thus, there is only one state of the world, ω, defined by the graph $G = (N, E)$ and the costs $c(e)$. An action network A is a sequence of edges making up a tour; the action space consists of all possible tours. Let a_i be the arc chosen for the i-th link of the tour. As in the NK-model, the performance function is the sum of individual performance contributions $c(a_i)$, the cost of each link chosen. The goal is to find a cycling tour that visits all cities at minimal cost. The interactions among the elements of the action network A do not reside in the performance function (as in the NK-model), but in the constraint that A must connect all cities in a tour (Bertsimas and Tsitsiklis, 1997). The TSP has more causal structure than the NK-model, and a change in one city tends to have more locally confined effects.

Let y_e be an indicator of whether edge e has been chosen, that is, $y_e = 1$ if $\exists\, a_i \in A$ with $a_i = e$, and $y_e = 0$ otherwise. Let $\delta(T)$ be the subset of edges that bridge the node subset T and its complement $(N - T) : \delta(T) = \{e \quad e = \{j, k\}, j \in T, k \notin T\}$ with $T \subset N$, $T \neq \emptyset, N$. Now we can write the performance function as

[7] We considered other ways of varying complexity. Varying the network density does not significantly change complexity. Neither does changing the variance of the distances (as suggested by Cheeseman *et al.*, 1991) affect the complexity when the distances are not constrained to be integers.

$$\text{Minimize over } A : \Pi(A) = \sum_{i=1}^{n} c(a_i), \quad \text{subject to:} \quad (1)$$

$$\sum_{e \in \delta(j)} y_e = 2 \ \ \forall j \in N \ \ \text{(each node must touch two edges;)}$$

$$\sum_{e \in \delta(T)} y_e \geq 2 \ \ \forall T \subset N (T \neq \emptyset, N) \ \ (\geq 2 \text{ edges connect any } T \text{ and its complement.})$$

The known state ω_R is the projection of the graph $G = (N, E)$ on a subgraph $G_R = (N_R, E_R)$ with a smaller number of cities and their edges. The nodes $\notin N_R$ are ignored in the tour, and thus the actions concerning the ambiguous cities a_R are set to link each of these cities to itself, so that $c(a_i) = 0 \ \forall \, i \in \{(d+1), \ldots, n\}$. The projected performance function is thus the same as in Equation 1, only within the subgraph G_R.

While the TSP is a classic model in computational complexity, it also demonstrates project management challenges. Different parts of a project often interact via constraints, due to limited resources (financial, man hours, etc.) or to component constraints (e.g., space, force, or energy). As in the exchange of cities in a tour, local improvements may affect other parts of the project. Ambiguity may manifest itself in unanticipated requirements (and thus components to fulfill them) and interactions.

For the search of the minimum cost tour, we use the 3-opt procedure with a neighborhood list implementation, which performs reasonably well (Aarts and Lenstra, 1997). While there are stronger heuristics available, the aim of our model is not to find the optimal TSP tour, but rather to compare selectionism and learning in the face of ambiguity. Thus, we believe that the use of this procedure is justified.[8]

After the project team has successfully constructed a tour in the projected

[8] We also performed simulations with a simpler optimization heuristic, 2-opt, and obtained the same results, only with more local peaks because the heuristic is less powerful.

landscape, the true quality of this solution must be evaluated in the full landscape. In other words, the ambiguous cities must be taken into account. For each ambiguous city, we do this by considering its five nearest neighbors (the five shortest edges adjacent to the city) and choosing the one that yields the best tour (without changing the tour otherwise). This represents a limited ability (even without learning) of the team to "scramble" after the unanticipated cities emerge, without the full capability of re-optimizing. This also amounts to considering only unk unks, not action ambiguity (as setting the ambiguous links a_R randomly would not produce a legitimate tour).

If the team invests in learning, it can identify the ambiguous cities sufficiently early to still be able to re-optimize the tour. Re-optimization may take two forms. (1) The team searches again, starting with the solution found in the initial problem-solving over the cities known at the outset. This corresponds to a situation where earlier decisions are very expensive to be changed wholesale. (2) Alternatively, the team may start from scratch with a new random tour including previously ambiguous cities. This is sometimes possible when contracts are not yet signed and analyses can easily be redone from templates.

It turns out that the latter option, on average, yields a slightly better result. This is consistent with previous work – by starting at a good prior solution, one tends to get stuck in a relatively bad local minimum (Aarts and Lenstra, 1997, 236). In our simulations, we report case (2) – the learning team can re-optimize with a new starting tour. The difference between the two cases, however, is small and does not change our results.

5.2 Complexity of the TSP

We vary complexity by changing the number of cities. Both the number of locally optimal tours found and the number of parallel trials necessary to get within 1% of the optimal solution increase exponentially in the number of cities (Figure 6). According to the second measure, a TSP with 20 cities has a similar complexity as the ($N = 10$) NK-model with $K = 3$; a TSP with 30 cities corresponds to the NK-model with $K = 6$, and a TSP with 40 cities approaches the complexity of the NK-model with $K = 9$.

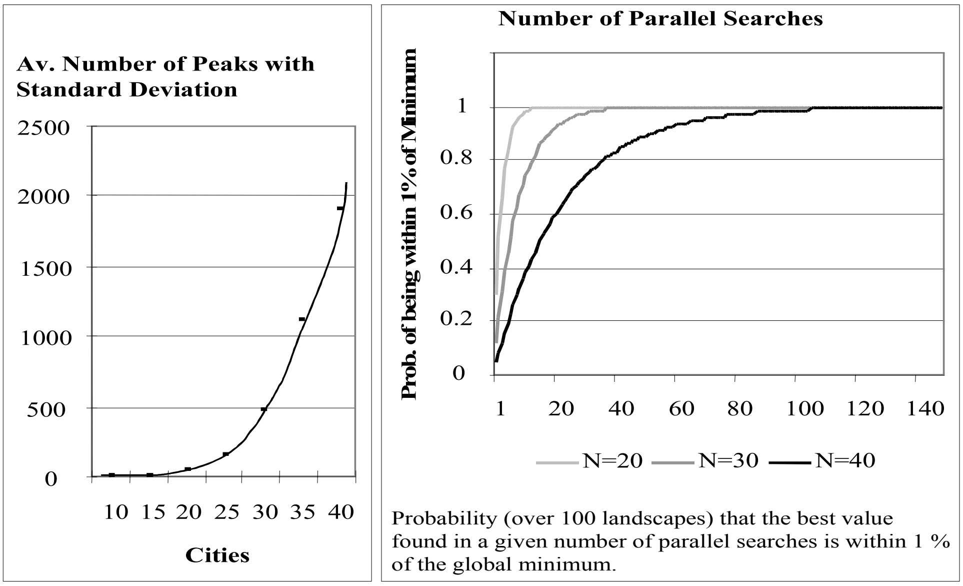


Figure 6: Complexity in the TSP Model

5.3 Optimal Selectionism in the TSP

While we held the number of ambiguous variables fixed at one in the NK-model, we take a fixed percentage (10 percent) of the cities to be ambiguous in the TSP. This is due to the fact that we vary complexity via the network size. If we kept the number of ambiguous cities fixed, their influence would be diluted by the increasing network size, due to the more local causal structure of the TSP.[9] Again, we first examine how the optimal number of parallel searches depends on ambiguity (only unk unks) and complexity (averaging over 100 simulations within a landscape and across 200 landscapes).

The optimal number of parallel searches increases with complexity, both for market tests and lab tests (Figure 7). The optimal number of parallel lab trials does not fall off as it does in the NK-model because the influence of the ambiguous cities is more local and confined. There is always some value in doing parallel searches, even when the choice of the best trial is based on the incorrect projected performance function.

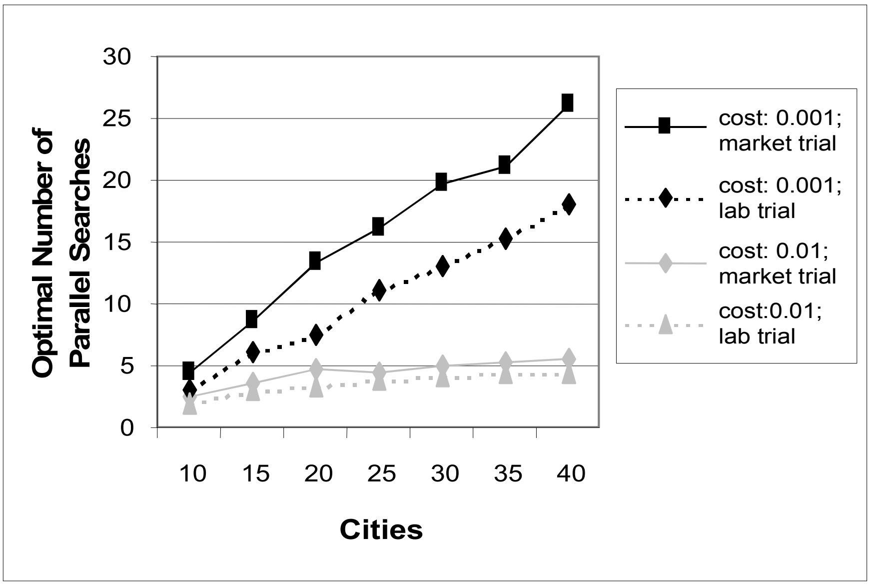


Figure 7: Optimal Number of Parallel Searches in the TSP

As in the NK-model, the optimal number of trials is lower for lab tests than for market tests because market tests take the influence of the unknown cities into account (albeit not optimally) and offer, thus, a higher search benefit.

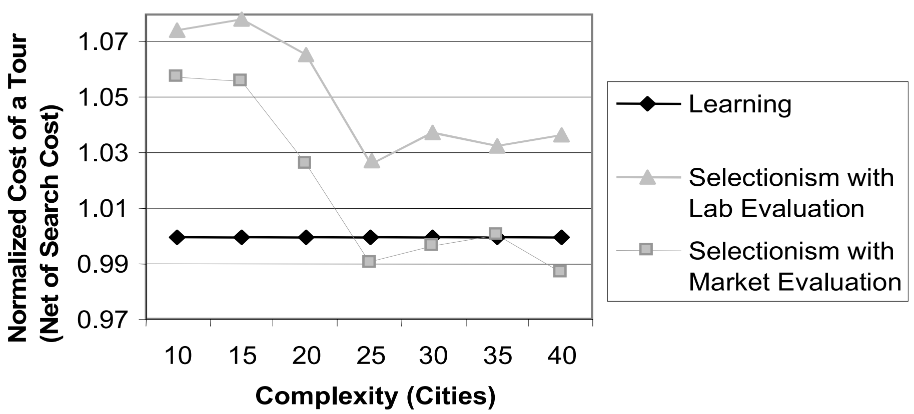
The effect of ambiguity on the optimal number of trials is similar to that in the NK-model. Ambiguity increases the optimal number of parallel trials for market

[9]We performed simulations with a fixed number of ambiguous cities and obtained the expected results: the effect of ambiguity and the advantage of learning "washed out" at high complexity.

tests but decreases it for lab tests. As the choice of the best selectionist lab test is based on the projected performance (without the unknown cities), it is suboptimal, making additional trials less interesting.

5.4 Selectionism and Learning in the TSP Model

As in the NK-model, learning is especially advantageous if the team has to perform lab tests, and thus cannot correctly evaluate the true performance function (Figure 8). While its advantage decreases with increasing complexity, it does not disappear even at high complexity levels. This implies that finding a better tour for the known cities through parallel runs does not fully compensate for giving up the improvement potential of learning and re-optimizing over the ambiguous cities after they are discovered. If the team *can* evaluate the true performance via market tests, learning still offers a relative benefit at low complexity (net of costs).



As complexity is varied via network size, the performance (cost of the tour) is not comparable in absolute terms across different complexity levels. Thus, we normalize the performance (average over 100 runs in 100 networks) of the learning approach, and compare selectionism in relative terms.

(The differences in tour costs from 25 to 40 cities for both selectionist curves are not statistically significant.)

Figure 8: Learning versus Selectionism in the TSP

6 Discussion and Conclusion

The model of a project that we have presented above is, to our knowledge, among the first to *formalize ambiguity* (Schrader *et al.*, 1993), the inability to recognize and articulate relevant influence variables. Our model is the first to *compare the relative attractiveness of the two fundamental project approaches, selectionism and learning* (Pich *et al.*, 2002) under the combined influence of project complexity and ambiguity. Selectionism involves pursuing several solution candidates independently of one another and picking the best one, *ex post*, and learning refers to a flexible (unplanned) adjustment of the project approach to the changing environment as it occurs.

This formalization and comparison are highly relevant for project teams executing novel projects, for which classic project management methods have proven insufficient. Our results on the attractiveness of selectionism and learning allow the project team to look at its task in a fundamentally new way, right at the outset, when important decisions about the approach must be taken.

Instead of a futile attempt to foresee all possible events that may be relevant to the project and then develop optimal (if contingent) policies, our model suggests that the project team ask itself five questions:

1. Is there major ambiguity present or not? Does it take the form of *unk unks* (unknown influence parameters that may differ from our tacit assumptions) or of *action ambiguity* (overlooked effects of actions we take)?
2. How good is the *causal mapping* that we have about the effects of our actions – do we understand them or not? If ambiguity is unimportant and good causal mapping is available, the team should use conventional risk management methods.

3. Can we perform solution trials that reveal the true project performance? In other words, can we perform reasonably exhaustive tests of unanticipated effects of our actions, or can we run "market tests" in a realistic user environment, which check for important unk unks?
4. How high is the complexity, or interrelatedness of actions, in our project?
5. And finally, how much does it cost to run selectionist parallel trials as compared to flexibly learning and redefining the project as it goes along (possibly having to go all the way into implementation to reveal all relevant influences)?

These five questions *can be answered at the outset* of the project; teams usually have at least a qualitative feeling about them. If ambiguity and complexity are important (yes to questions one and two), the team should use some combination of selectionism and learning. Our model makes suggestions as to when which approach may be most suitable (Figure 9). First, if one is much more expensive than the other (it should be noted that both the pursuit of multiple solutions and learning from true market feedback may be very expensive indeed), the cost effective approach should be chosen.

Net of costs, learning, or flexible project redefinition, appears as attractive relative to selectionism if the team cannot design parallel trials that reflect the true (unknown and impossible to express) performance, that is, if all possible trials are stylized "lab tests" that do not reveal unk unks. This is true both when the team has a good causal model of the effects of its actions (as in the TSP model) and when the team does not have a good causal model (as in the NK-model).

However, learning falls far short of selectionism when the project faces a combination of action ambiguity (unanticipated action effects), a poor action-effect model, and high complexity. In this situation, a single trial may get stuck in a bad local peak, no matter

	Trial evaluation of true performance available (market tests, action ambiguity)	Trial evaluation available only in projected landscape (lab tests)
Structured causal mapping available from actions to performance (TSP)	• Learning performs better in the case of low complexity • Advantage vanishes with increasing complexity levels	• Learning performs better than selectionism • Advantage diminishes with increasing complexity but does not disappear
No structured causal mapping available from actions to performance (NK-Model)	• Learning performs as well as selectionism for unk unks • Learning performs worse for action ambiguity	• Learning performs better than selectionism • Advantage increases with complexity • Optimal number of selectionist trials may *decrease* with complexity

(All comparisons between learning and selectionism are net of search cost)

Figure 9: Choice of Project Infrastructures

how well the team learns, while multiple trials have the chance of escaping from a bad region of the solution space.

Our model suggests another interesting finding: the well-known principle that more complexity necessitates a larger number of parallel trials hinges on the absence of ambiguity (complete knowledge about all states of the world). This principle may be reversed in truly novel situations, where some problem characteristics (unk unks) are unknown. If the team does not have a good causal model or the ability to evaluate the performance correctly (e.g., market tests are too expensive), high complexity reduces the predictive power of the trials and thus their value, suggesting *fewer* trials.

It is of very high importance for a project team to get these fundamental decisions about the project approach right at the beginning. Diligent execution depends on setting the right environment at the outset. Our model helps the project manager, conceptually, to think about these decisions more intelligently.

In this first study, we have, of course, considered only a situation where the team must choose between selectionism and learning, and cannot combine them.

Empirical work is under way to test the recommendations of our model and to explore modes of combining the two approaches. This will also inspire further theoretical work.

The five questions and prescriptive results summarized in Figure 9 contribute to project management theory. We are not aware of any work that has compared selectionism and learning without assuming full information about the project's state space. By taking a new angle on the key decisions the project manager must make at the outset, we hope to open up new research directions about decision-making in project management.

References

Aarts, E. and J.K. Lenstra (eds.). 1997. *Local Search in Combinatorial Optimization*, New York: John Wiley & Sons.

Bank, D. 1995. The Java Saga. *Wired* December, 166-169 and 238-246.

Bertsekas, D. P. 1995. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA.

Bertsimas, D. and J.N. Tsitsiklis. 1997. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA.

Cheeseman, P., B. Kanefky and W.M. Taylor. 1991. Where the really hard problems are. In: Mylopoulos, J. and R. Reiter, editors. *Proceedings of IJCAI-91* San Mateo, 331-337.

Clark, C.E. 1961. The greatest of a finite set of random variables. *Oper. Res.* 9, 145-162.

Ferreira, A.G. and J. Zerovnik. 1993. Bounding the probability of success of stochastic methods for global optimization. *Computer Math Applic.* 25, 1-8.

Fox, B.L. 1993. Random restarting versus simulated annealing. *Computer Math. Applic.* 27, 33-35.

Fox, B.L. 1994. Integrating and accelerating tabu search. *Annals of Operations Research* 41, 47-67.

Frenken, K. 2000. A complexity approach to innovation networks. The case of the aircraft industry (1909-1997). *Research Policy* 29, 257-272.

Frenken, K. 2001. Modelling the organization of innovative activity using the NK-model. *Paper prepared for the Nelson-and-Winter Conference, Aalborg, June 2001.* 1-24.

Garvin, D.A. 1993. Building a Learning Organization. *Harvard Business Review* July-August, 78-91.

Gavetti, G. and D. Levinthal. 2000. Looking Forward and Looking Backward: Cognitive and Experiential Search. *Administrative Science Quarterly* 45 113-137.

Isenberg, D.J. 1984. How Senior Managers Think. *Harvard Business Review* 62(6) 81-90.

Jacobson, S.H., and E. Yücesan. 1998. Performance results for generalized hill-climbing algorithms. INSEAD Working Paper.

Kauffman, S.A. 1993. *The Origins of Order.* Oxford: Oxford University Press.

Kauffman S.A. and J. Lobo. 2000. Optimal Search on a Technology Landscape. *Journal of Economic Behavior and Organization* 43(2), 141-166.

Leonard-Barton, D. 1995. *Wellsprings of Knowledge.* Boston: Harvard Business School Press.

Levinthal, D. 1997. Adaptation on Rugged Landscapes. *Management Sci.* 43(7), 934-950.

Loch, C.H., and C. Terwiesch. 2002. The Circored Project. *Insead Case Study.*

Loch, C.H., C. Terwiesch, and S. Thomke. 2001. Parallel and Sequential Testing of Design Alternatives. *Management Science* 47 (5), 663 - 678.

Lovejoy, W.S. 1991. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research* 28, 47-66.

Lynn, G.S., J.G. Morone and A.S Paulson. 1996. Marketing and Discontinuous Innovation: The Probe and Learn Process. *California Management Review* 38(3), 8-37.

Marschak, J. and R. Radner. 1972. *Economic Theory of Teams.* New Haven: Yale University Press.

McGrath, R.G. 2001. Exploratory Learning, Innovative Capacity, and Managerial Oversight. *Academy of Management Journal*, 44(1), 118-131.

Morris, P.W.G. and G.H. Hough. 1987. *The Anatomy of Major Projects.* Chichester: Wiley.

Phadke, M.S. 1989. *Quality Engineering using Robust Design.* Engelwood Cliffs: Prentice Hall.

Pich, M.T., C.H. Loch and A. De Meyer. 2002. On Uncertainty, Ambiguity and Complexity in Project Management. *Management Sci.*, 48(8), 1008-1023.

Pisano, G.P. 1994. Knowledge, Integration, and the Locus of Learning: An Empirical Analysis of Process Development. *Strategic Management Journal*, 15, 85-100.

Rivkin, J.W. 2000. Imitation of Complex Strategies. *Management Science* 46(6), 824-844.

Rudin, W. 1976. *Principles of Mathematical Analysis*, 3rd Edition, Mc Graw-Hill.

Sabbagh, K. 1996. *21st Century Jet.* New York: Scribner.

Schrader, S., W.M. Riggs and R.P. Smith. 1993. Choice over uncertainty and ambiguity in technical problem solving. *Journal of Engineering and Technology Management* 10, 73-99.

Simon, 1969. *The Science of the Artificial.* 2nd. ed., Boston: MIT Press.

Sobek II, D.K., A.C. Ward and J.K. Liker. 1999. Toyota's Principles of Set-Based Concurrent Engineering. *Sloan Management Review* 40, 67-83.

Tatikonda, M.V. and S.R. Rosenthal. 2000. Technology novelty, project complexity, and product development execution success. *IEEE Transactions on Engineering Management* 47, 74-87.

Thomke, S.H. 1998. Simulation, learning and R&D performance: Evidence from automotive development. *Research Policy.* 27, 55-74.

Tushman, M.L. and C.A. O'Reilly. 1997. *Winning through innovation: a practical guide to leading organizational change and renewal.* Boston: Harvard Business School Press.

Ward, A., J.K. Liker, J.J. Cristiano and D.K. Sobek II. 1995. The Second Toyota Paradox:

How Delaying Decisions Can Make Better Cars Faster. *Sloan Management Review*, Spring, 43-61.
Wideman, R.M. 1992. *Project and Program Risk Management.* Project Management Institute, Newton Square, PA.
Williams, T.M. 1999. The need for new paradigms for complex projects. *International Journal of Project Management* 17 (5), 269-273.

Appendix

Theorem 1. If the ambiguous variables in ω_N have been assumed incorrectly in the lab environment, then for $K = N - 1$, the true expected value of the solution (A^*) found in the lab evaluation with any number or parallel runs is $E(\Pi(M((\omega_R, \omega_N), A^*), A^*)) = 0.5 = E(\Pi(M(\omega, A), A)) \; \forall \; A$, or the expected value of a random point in the landscape. Thus, for $S \to \infty$, the optimal number of parallel searches approaches 1 for any positive search cost.

Proof. Assume that the value of the ambiguous factors set in the lab $A_N^{lab} \neq A_N^{true}$ (value of the ambiguous factor encountered in the market). The parallel searches find a local maximum for the actions A^* in the perceived performance function with one unknown: $\Pi_R(M_R(\omega, A^*), A^*) = \frac{1}{N} \sum_i \phi_i(A_R, A_N^{lab})$. The true performance of the best solution A^* with one unknown is given by $\Pi(M(\omega, A^*), A^*) = \frac{1}{N} \sum_i \phi_i(A_R, A_N^{true})$. Since $A_N^{lab} \neq A_N^{true}$ $\phi_i(A_R, A_N^{lab}) \neq \phi_i(A_R, A_N^{true})$ $\forall \; \phi_i$, and $\phi_i(A_R, A_N^{true})$ are different random variables distributed $U(0,1) \; \forall \; i$.

Thus, $E(\Pi(M(\omega, A^*), A^*)) = E(\frac{1}{N} \sum_i \phi_i(A_R, A_N^{true})) = \frac{1}{N} \sum_i E(\phi_i(A_R, A_N^{true})) = \frac{1}{N} * N * 0.5 = 0.5 \; \forall \; A^*$. Hence, for $K = N - 1$ the expected value in the true landscape of any maximum found in the perceived performance landscape equals the expected value of a random point in the landscape.

The probability of setting the ambiguous factor correctly in the lab environment is $\frac{1}{S+1}$. Thus, the expected value of true performance (Π) of the found solution, given its expected performance (Π_R), is given by $E(\Pi|\Pi_R) = \frac{1}{S+1} * \Pi_R + \frac{S}{S+1} * 0.5$ and $lim_{S \to \infty}(E(\Pi|\Pi_R)) = 0.5$. With increasing S, the value of additional searches decreases. In the limit, any number of searches has an expected value of 0.5, and is thus, in expectation, no better than a single search or even a random point in the landscape. Given positive search costs, only one search should be performed.