Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

THE EFFECTS OF TEAMS' INITIAL CHARACTERIZATIONS OF INTERACTIONS ON PRODUCT DEVELOPMENT PERFORMANCE

Mohsen Jafari Songhori

Department of Computer Science
Tokyo Institute of Technology
Modori-Ku Yokohama, 226-8502, JAPAN

Mohammad S. Jalali

MIT Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Takao Terano

Department of Computer Science
Tokyo Institute of Technology
Modori-Ku Yokohama, 226-8502, JAPAN

ABSTRACT

Coordinated search processes are pervasive in both organizations and product development projects. In such processes, designers with different specialties learn about their interdependent alternatives through a mutual adjustment process. In the context of a product development with several teams developing the new product's subsystems, and using reinforcement learning and agent-based simulation modeling, this study looks at the performance effects of design teams' initial mental characterizations about subsystem interactions. The focus is on two initial mental models, one in which teams over-weight their own subsystem's element interactions, and another, in which teams over-weighting interactions between subsystems. The results indicate that both initial representations have critical performance consequences for product development. Specifically, teams prioritizing their interactions of their own subsystem's elements gain short-run performance benefits as they converge to a local optimum in a short time period. Contrarily, over-weighting between-subsystem interactions leads to a tendency for teams to have long-run performance advantages.

1 INTRODUCTION

According to both organizational and product development (PD) literature, search processes often involve joint efforts by specialists from different domains (Knudsen and Srikanth 2014), and such coordinated or simultaneous searches are ubiquitous in these two fields (Mihm, Loch, and Huchzermeier 2003, Knudsen and Srikanth 2014). Designers are usually engaged in coupled learning or mutual adjustment processes based on trial and error (Puranam and Swamy 2016). Despite the importance of and challenges to such learning processes, however, organizational search models usually conceptualize firms to be unitary actors with cognitive limitations (Cyert and March 1963, Levinthal 1997, Siggelkow and Levinthal 2003), and PD models assume that engineers can communicate their design decisions to the other engineers who are working on interdependent components (Mihm, Loch, and Huchzermeier 2003).

Theoretical models for coordinated and simultaneous search problems have only recently been developed and analyzed (Knudsen and Srikanth 2014, Puranam and Swamy 2016). However, coupled search models have not been examined for the PD context, which often includes several teams and designers. Along these lines, and following Puranam and Swamy (2016), the present study investigates the following research question: How do design teams' initial mental characterizations about their subsystem interactions affect PD performance? In order to address this question, we develop an agent-based simulation model of a PD project.

The rest of this paper is organized as follows. The relevant organizational and PD literature is discussed in Section 2. In Section 3, we formalize our agent-based model of PD project teams and their learning process. Then, in Section 4, we describe our simulation experiments and report our results. We discuss implications of our model and the results for PD managers in Section 5.

2 LITERATURE REVIEW

2.1 Coordinated Searches in Organizational Context

Organizational searches often involve joint efforts by specialists from different domains (Knudsen and Levinthal 2007, Knudsen and Srikanth 2014). In such processes, several specialists jointly conduct searches on their own domains, and each one's payoff is a function of both their own choices and others' choices. Moreover, each specialist has limited knowledge about the others' search domains. Coordinated exploration problems include epistemic interdependence in which one agent's optimal choices depend on accurately predicting another agent's actions (Puranam, Raveendran, and Knudsen 2012). In such coordinated search problems, unknown interdependencies and communication constraints exist that make those problems highly challenging.

The literature of organizational search models (Siggelkow and Levinthal 2003, Fang, Lee, and Schilling 2010), however, has often ignored epistemic interdependence and sidestepped issues of coordination (Knudsen and Srikanth 2014). Moreover, theoretical models for coordinated, simultaneous, and explorative search problems have only recently been developed and analyzed. Knudsen and Srikanth (2014) developed an agent-based simulation to examine differences between coordinated searches by specialists and unitary searches. They found two features of coordinated searches: (i) mutual confusion (i.e., agents unable to learn from feedback); and (ii) join myopia (i.e., feedback on one's actions is confounded by the actions of another). In another study, Puranam and Swamy (2016) argued that organizational systems actively shape coupled learning processes by modifying designers' mental models through a variety of centralized processes (e.g., planning) and their outputs (e.g., plans). They used agent-based simulation, and investigated how initial representations held by the learners in coupled learning processes affect the success of such learning processes, particularly when communication is constrained and individual rates of learning are high.

2.2 Coordinated Searches in Product Development Context

The characteristics of coordinated exploration problems that include epistemic interdependence are also present in product design and development projects. Usually, each designer is working on one component, and since components have technical interdependencies (e.g., spatial limitations), engineers should communicate their successive decisions about their respective components (Mihm et al. 2010).

In addition to these challenging features of coordinated searches, unknown interdependencies and communication constraints also exist in product development contexts. Such unknown interactions have been empirically examined; for instance, in a design project for a commercial aircraft engine, where between subsystem interdependencies were invisible to system architects (Sosa, Eppinger, and Rowles 2004, Sosa, Eppinger, and Rowles 2007). Therefore, designers are usually engaged in a mutual adjustment process based on trial and error (Alexander 1964, Thompson 1967, Thomke 1997, Eppinger 2001), which is called coupled learning (Puranam and Swamy 2016). Such coupled learning processes are manifested as unplanned design iterations where, due to the presence of subsystem interactions, decisions made for other interdependent subsystems make the choices made for a focal subsystem highly unstable (Alexander 1964, Thomke 1997). In this line of work, frequent design iterations and changes are seen in several industries (Allen 1966, Terwiesch and Loch 1999).

Previous theoretical models and arguments have described the effects of knowledge and information sharing on performance of coordinated search efforts, like empirical and controversial results about PD success effects of specialists' intense communication (e.g., see pages 413-414 in Knudsen and Srikanth

(2014)). However, extending these models for PD projects where a number of teams conduct searches has potential implications, not only for PD, but also for operations management (OM).

3 AGENT BASED SIMULATION MODEL

In this section, we set up the mathematical model used to simulate the search process. The model constituents are: (i) characterization of the performance landscape over which the PD teams conduct searches; (ii) design teams' coupled learning process by which teams learn about subsystems interactions.

3.1 The Product Landscape

We model the product performance landscape payoff function using an NK landscape (Kauffman 1993). In our model, there are N design elements that the PD teams must develop. In addition, these decisions are considered as binary choices. Consequently, the design search space consists of a total of 2^N possible configurations of product design alternatives. Since a product is often decomposed into subsystems, each of which are being developed by one team, we assume that the product is decomposed into z subsystems. In addition, each subsystem $s \in \{1,2,..,z\}$ is being developed by team $e \in \{1,2,..,z\}$, and as a result, there are also z teams in the project. Furthermore, since we assume that all subsystems have the same number of elements, in the NK terminology, the landscape of subsystem s consists of $\frac{N}{z}$ interacting binary elements that are in state 0 or state 1 at any given time.

Let's assume vector $a = \cup_{s=1}^{s=z} w^s$ represent one of 2^N possible design alternatives in which $w^s = (a_1^s, a_2^s, \ldots, a_{\frac{N}{z}}^s)$ is the design choice of subsystem s. When a particular design configuration vector a is selected, it results in pay-off/fitness $f(a)$. The fitness value $f(a)$ is defined as the average of the contributions of all design elements: $f(a) = \frac{1}{N}\sum_{s=1}^{s=z}\sum_{j=1}^{j=\frac{N}{z}} C(a_j^s|a)$. The contribution of each design element a_j^s when the design choice vector is a, is shown by $C(a_j^s|a)$.

In the classical NK landscape model, parameter K indicates the degree of interaction among elements, that is, the contribution of one element $C(a_j^s|a)$ depends on the state of K other randomly selected elements. In our simulation method, we take a different approach to control the interdependencies within and between subsystems, we specify interactions using two parameters instead of one as shown below.

Since the design elements of a subsystem are more likely to interact, then, each element of subsystem s is defined to interact with K_h other elements of the same subsystem. We call these within-subsystem interactions. Also, since subsystems are interdependent, when two subsystems s and s' interact, each element of subsystem s has K_b interactions with elements of subsystem s'. Two important aspects of these between-subsystem interactions need further clarification. First, subsystem-level interactions, or which subsystem interacts with which others, are determined according to a procedure that is described in the "Experiment" section. Second, managers and engineers may hold high (low) level architectural knowledge, and hence, modules might (or might not) defined properly, and within-subsystem interactions can be more (low) intense than between-subsystem ones, or mathematically $K_h > K_b$ ($K_h < K_b$).

As an example, consider the case of a product with $z = 4$ subsystems, and $N = 12$ design elements. The design space of each subsystem is consists of $\frac{N}{z} = 3$ design elements. Assume that subsystem 1 interacts with subsystem 4, and also, within-subsystem and between-subsystem interaction parameters are defined as follows: $K_h = 1$ and $K_b = 1$. Then, the performance contribution of decision a_1^1, or $C(a_1^1)$, would have eight possible values, depending on how the other two interdependent decisions (i.e., one decision of subsystem 1, or a_2^1 and one element of subsystem 4, or a_3^4) are resolved. These contribution values, for all decisions, are drawn randomly from a uniform distribution over the unit interval $[0,1]$:

$$\begin{aligned} C_1(a_1^1,\{a_2^1,a_3^4\}) \in \{&C_1(0,0,0),C_1(0,0,1),C_1(0,1,0),C_1(1,0,0),\\ &C_1(0,1,1),C_1(1,0,1),C_1(1,1,0),C_1(1,1,1)\}. \end{aligned} \tag{1}$$

Once, by using the aforementioned procedure, the set of design elements of the same or other subsystems that interact with element a_j^s of subsystem s are specified, the landscape function is generated. Depending on the number of elements of that set, it follows that there are a number of possible contribution values (i.e., all possible values of $C(a_j^s|a)$). These contribution values of element $C(a_j^s|a)$ are drawn from a uniform $[0,1]$ distribution. We remark that the properties of the fitness landscape are not sensitive to the distribution applied to generate the landscape (Weinberger 1991).

3.2 Designers Coupled Learning Process

We conceptualize the design process as an engineers' parallel collective search process as follows. At any time t, each team e's design choice corresponds to a position over the product landscape with N decisions. In addition, at time $t = 0$, each team (or agent, problem-solver) e is randomly assigned a state, and its fitness value (i.e., performance) is calculated. Let $f_t^e(a)$ be the fitness of team e at time t that has design vector a at time t. Thereafter, at each subsequent time t, each agent e engages in a learning process—that is a combination of both social and individual learning (March 1991, Lazer and Friedman 2007, Fang, Lee, and Schilling 2010, Barkoczi and Galesic 2016). We define the overall PD performance as the average fitness (payoff) of all design teams.

Based on the learning process, the focal agent improves its own design choice by a local search or considers the design choices of the interdependent teams. This is different from similar collective organizational search models (Lazer and Friedman 2007, Fang, Lee, and Schilling 2010), since, in the design context, each team is responsible for the design choices of its allocated subsystem, and focal team e's individual learning is accomplished over its own subsystem design elements (which is $(a_1^s, a_2^s, \ldots, a_{\frac{N}{z}}^s)$ for $e = s$). However, team e's social learning aims to improve its allocated subsystem's interactions with the other subsystems, and hence, its social learning is conducted over all other subsystems that are perceived as interdependent subsystems by team e (which includes design spaces like $(a_1^{s'}, a_2^{s'}, \ldots, a_{\frac{N}{z}}^{s'})$ and $s' \neq e$).

When a focal team conducts social learning, it conducts searches among its interdependent teams and, based on the subsequent learning process, it selects one of them, and, partially imitates that selected team's subsystem design choice. In other words, a focal team selects one of its interdependent teams, and imitates one design choice of the selected team's subsystem choices. Teams can also use an individual learning strategy in which a team examines the resulting payoff of modifying the current design choice by only changing a single design element of its subsystem, and adopts the resulting design choice if it has a higher payoff.

While organizational collective search models appropriately assume social/individual learning frequency to be fixed (Lazer and Friedman 2007, Barkoczi and Galesic 2016), we need a different approach, since each team has a given search domain (i.e., design space of its subsystem), and also it has limited knowledge of interactions among subsystems (i.e., architectural knowledge of design space). Consequently, one crucial aspect of our simulation model is that teams have partial and incomplete knowledge of interactions among subsystems, and they learn about those interactions during their search process.

Since each team has incomplete knowledge about all possible interactions, we assume that, at first, each team e has to make a team-level decision, and specify to which subsystem's interactions it will allocate its resources. The options for that team-level decision are the focal team itself (team e), and all other teams (any other team e') as well. If team e selects itself (or more precisely, its own subsystem), it conducts individual learning and local search. Whereas, when it selects other team $e' \neq e$, it follows a social learning process. More formally, we define the probability of selecting any team (e.g., team e') to be a function of the expected payoffs for that option (e.g., team e') , relative to the expected payoffs for other options (e.g., all other teams except team e'), using the softmax functional form (Sutton and Barto 1998) that has been seen empirically in trial and error learning situations (Camerer and Hua Ho 1999):

$$p^e_{e'} = \frac{\exp^{\frac{\pi^e_{e',t}}{\tau}}}{\sum\limits_{e'' \neq e, e'} \exp^{\frac{\pi^e_{e',t}}{\tau}}}, \quad \forall e'. \tag{2}$$

In equation 2, $p^e_{e'}$ is the probability by which team e selects team e' to attend to its subsystem interactions with the latter's subsystems. When the selected team is any team other than team e (or $e' \neq e$), then team e imitates one of the subsystem $s' = e'$ design choices (i.e., one of design elements' state $(a^{s'}_1, a^{s'}_2, \ldots, a^{s'}_{\frac{N}{z}})$). However, if the focal team selects itself, or formally $e = e'$, then team e examines the resulting payoff of modifying a single digit of its current design choice only over its subsystem design space. Afterwards, team e adopts the resulting design, if it has a higher payoff. Otherwise, the team retains its current design choice at the next time step.

Also, in equation 2, the expected payoff of team e, in selecting team e' at time t is shown by $\pi^e_{e',t}$. In addition, by parameter τ, the exploration level of search is controlled (Sutton and Barto 1998). Low values of τ result in higher probabilities of selecting teams with higher expected payoffs, whereas, with high values of τ, teams will have equal selection likelihoods. Lastly, the defined probabilities, $p^e_{e'}$ are used in the roulette wheel algorithm to determine the subsequent decisions of team e at time t: (i) which learning type (i.e., social or individual) to enforce, and (ii) which team e''s subsystem design choices to consider.

Each learning endeavor by any team can be seen as a pilot project, a product prototype, or even an individual attempt to solve a component of the joint problem. Each team learns from the feedback, and modifies its beliefs about the weights/possibilities of the presence of design interactions (within its own subsystem or among its elements and those of the other teams), and this process repeats over time (Allen 1966, Terwiesch and Loch 1999). To materialize this learning process, we assume that the teams engage in reinforcement learning over the simulation time period (Sutton and Barto 1998):

$$\pi^e_{e',t} = \pi^e_{e',t-\bar{t}} + \phi\left[(f^e_t(a') - f^e_{t-\bar{t}}(a)) - \pi^e_{e',t-\bar{t}}\right]. \tag{3}$$

An important part of equation 3 is the difference between team e's fitnesses at the current (t) and previous time periods ($t - \bar{t}$), or mathematically $(f^e_t(a') - f^e_{t-\bar{t}}(a))$. This can be considered as received or missed rewards when this term is positive or negative, respectively. A critical parameter in the reinforcement learning equation 3 is ϕ, which specifies the rate at which expected payoffs are adjusted (both upward and downward) based on the received or missed rewards. A high value for ϕ indicates the team's ability to recognize and adapt rapidly to positive and negative feedback.

Another important feature of equation 3 (and also equation 2) is the frequency of receiving feedback and learning from it. In other words, we assume that both the team selection softmax function in equation 2, and the reinforcement learning function in equation 3 are applied only at particular time interval. In particular, the frequency of those selection and learning processes is determined by parameter $\bar{t}$, and its low values (high values) indicate high (low) frequency and learning rate. Thus, equations 2 and 3 can be written more precisely, as follow (t mod $\bar{t}$ is the remainder when dividing t by $\bar{t}$):

$$p^e_{e'} = \begin{cases} \frac{\exp^{\frac{\pi^e_{e',t}}{\tau}}}{\sum\limits_{e'' \neq e, e'} \exp^{\frac{\pi^e_{e',t}}{\tau}}}, & \forall e', \quad t \ mod \ \bar{t} = 0 \\ 0, & t \ mod \ \bar{t} > 0 \end{cases}, \tag{4}$$

$$\pi^e_{e',t} = \begin{cases} \pi^e_{e',t-\bar{t}} + \phi\left[(f^e_t(a') - f^e_{t-\bar{t}}(a)) - \pi^e_{e',t-\bar{t}}\right], & t \ mod \ \bar{t} = 0 \\ \pi^e_{e',t-\bar{t}}, & t \ mod \ \bar{t} > 0 \end{cases}. \tag{5}$$

Consider the previous example, in which there are $z = 4$ subsystems, and the design space of each subsystem consists of $\frac{N}{z} = \frac{12}{4} = 3$ design elements. Assume team 2 has choice $a_1^2 = [0,0,0,0,0,0,0,0,0,0,0,0]$ (and its fitness, at time $t = 1$, is $f_1^2(a) = 0.51$), and the feedback frequency interval is $\bar{t} = 10$ time periods. Initially, at time $t = 1$, using equations 4 and 5, and by conducting the described procedures, team 2 selects itself, and thus, it conducts individual learning (or local search) for the next $\bar{t} = 10$ time periods. According to individual learning, team e examines the resulting payoff of modifying a single digit of its current design choice only over its subsystem design space. Then team e adopts the resulting design, if it has a higher payoff. Otherwise, the team retains its current design choice at the next time step.

In the example, if team 2 follows individual learning at time $t = 2$, then by modifying a single digit of the current design over its subsystem space (i.e., the first three bits), it may generate new choice $a_{new}^2 = [0,1,0,0,0,0,0,0,0,0,0,0]$ that has fitness $f(a_{new}^2) = 0.53$. Then, the team accepts design choice a_{new}^2 as its fitness is higher than the current design choice $f(a_{new}^2) > f(a_3^2)$. The main ingredients of this selection procedure are the design team 2' initial representation of interactions that are manifested as $\pi_{1,1}^2, \pi_{2,1}^2, \pi_{3,1}^2$, and $\pi_{4,1}^2$. As an example, team 2 may have a very biased perception about between-subsystem interactions (i.e., allocating low weights to them), and these values are considered as $\pi_{1,1}^2 = 0.025, \pi_{2,1}^2 = 0.9, \pi_{3,1}^2 = 0.025$, and $\pi_{4,1}^2 = 0.025$.

By such a local search process, team 2's design choice is likely to improve, and be at, for example, $a_{10}^2 = [1,0,0,1,0,0,0,0,0,0,0,0]$ with fitness $f_{10}^2(a) = 0.61$. At time $t = 10$, as $t \bmod \bar{t} = 0$, using equations 4 and 5, team 2 selects interactions of which subsystem it will attend during the next $\bar{t} = 10$ time periods. Particularly, using $f_1^2(a) = 0.51$ and $f_{10}^2(a) = 0.61$, the values of expected payoffs are revised: $\pi_{1,10}^2 = 0.025, \pi_{3,10}^2 = 0.025, \pi_{4,10}^2 = 0.025$ and $\pi_{2,10}^2 = 0.9 + \phi((0.61 - 0.51) - 0.9)$. This process is repeated for all PD teams, and during the whole simulation time, and the results are recorded.

4 SIMULATION EXPERIMENTS

In this section, we describe the experimental setup and report the results. We first detail the simulation procedure and ingredients, and then discuss the results of the experiments. Parameter N in the NK simulation model is the total number of design elements in the landscape. In our experiments, we assume that the project organization consists of five teams and that each team controls three elements; hence $N = 5 \times 3 = 15$. We also study a four-team project with each team controlling four elements $N = 4 \times 4 = 16$. Two other landscape parameters are K_h and K_b, considered to be either $K_h = \frac{N}{z} - 1, K_b = 1$ or $K_h = 0, K_b = \frac{N}{z}$. The former setting represents PD projects with proper architectural knowledge since subsystems are defined properly, and elements' interactions mainly occur among elements of one subsystem. Contrarily, the latter arrangement shows PD teams having a limited level of architectural knowledge.

Subsystems in a PD project may relate to each other in different architectures (patterns). For example, Table 1 shows the two patterns we use in our experiments for five teams. In the random pattern, any two subsystems s and s' are interdependent (i.e., each element of subsystem s has K_b interactions with elements of subsystem s') with probability $PS \in [0,1]$. A centralized pattern is a concept where subsystems have mutual interactions (i.e., team e' actions affects payoff of team e' and vice versa), and some subsystems have significantly higher numbers of interactions than the other subsystems (see (Ghemawat and Levinthal 2008)). These two patterns cover both symmetric (e.g., random) and asymmetric (e.g., centralized) distribution of interactions among subsystems.

Although there are particular interaction patterns among subsystems (see Table 1), and due to bounded-rationality and limited product architectural knowledge, PD organizations are assumed to shape teams' mental models about interactions in one of the following ways: (i) Each teams is biased toward over-weighting interactions among elements of its own subsystems, or (ii) Each teams over-weights interactions among its own subsystem and those of the other teams. We refer to the former and latter strategies as within-subsystem integration bias, and between-subsystem bias, respectively. Comparison of these two

Table 1: Interaction network among five teams.

PD Team	1	2	3	4	5
1	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	
4	✓	✓	✓	✓	
5	✓	✓			✓

Centralized

PD Team	1	2	3	4	5
1	✓			✓	
2		✓			✓
3	✓		✓		
4	✓		✓	✓	
5	✓				✓

Random-$PS = 0.2$

simplified and generic strategies enables us to investigate and compare performance consequences of designers' biased mental models in a couple search arrangement.

To materialize the two strategies, at first, we assume summing the initial expected payoff of each team for attending interactions of all teams (including itself) to be unit, or mathematically $\sum_{e'=1}^{e'=z} \pi^e_{e',1} = 1$. Then, the first strategy is defined by setting the initial expected payoff of any team e in selecting its subsystem at time $t = 1$, to be a very high value $\pi^e_{e,1} = 0.9$, and the expected payoff for selecting any subsystem, other than its own subsystem, to be a very low value $\pi^e_{e',t} = \frac{0.1}{n-1}, \;\; e \neq e'$. Conversely, the second strategy is implemented as follows. The initial expected payoff of any team e in selecting its subsystem at time $t = 1$, is considered a very low value $\pi^e_{e,1} = 0.1$, and the expected payoff for selecting any subsystem, other than its own subsystem, is set to a very high value $\pi^e_{e',t} = \frac{0.9}{n-1}, \;\; e \neq e'$.

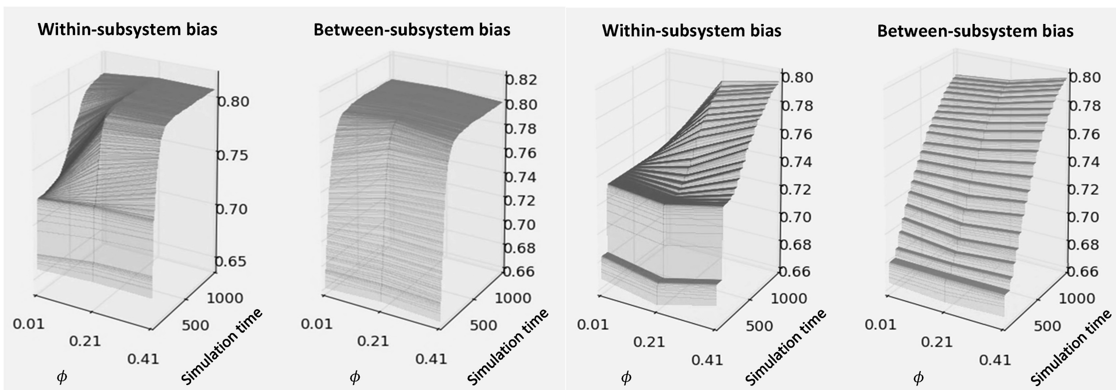
We arrange two PD systems, one with within-subsystem integration bias, and another one with between-subsystem bias, and for each system, the following simulation experiments are conducted. We generate 50 NK landscapes, and run each PD scenario over each landscape four times (in each of the four runs over the same landscape, different initial design choices or positions over landscape are used), resulting in 200 simulation experiments. Then, each of the following scenarios is simulated over these 200 simulation experiments. In each scenario, we simulate the PD project for $T = 1000$ time periods, and the roulette wheel algorithm parameter is set as $\tau = 0.1$. In addition, each scenario is constructed by selecting reinforcement learning parameter ϕ and learning frequency parameter $\bar{t}$ in the following ranges, respectively: $\phi \in [0.01, 0.21, 0.41]$ and $\bar{t} \in [5, 25, 45]$. The defined range for learning frequency rate parameter is defined such that, on the one hand, a sufficient number of learning occurs, and on the other hand, the learning frequency time interval is long enough so that learning is effective (i.e., when $\bar{t} = 1$, the variations of the resulting fitnesses over search course is high, and learning is highly ineffective).

Among all these scenarios, in below, we report and discuss only the results of some selected scenarios. These results are adopted such that the overall seen patterns among all simulation scenarios can be precisely and properly discussed. In Figure 1, the average performance PD systems with within-subsystem bias and that of those PD arrangements with between-subsystem bias are plotted. We observe that when each team over-weights interactions among its own subsystem elements (left graph in each panel), in comparison to its subsystem's interactions with the other subsystems, the overall PD performance increases quickly (e.g., $t = 100$). However, and contrarily, if teams over-weight and integrate interactions between subsystems (right graph in each panel), the overall performance increases, slowly in short term (e.g., $t = 100$), but achieves higher performance in the long run (e.g., $t = 900$).

It is worth noting that, in fact these short and long term horizons are relative to the total simulation time ($T = 1000$), and rather than being elaborated horizons, they indicate the expected overall PD performance level when teams go through either a small (short run) or large number of feedback and learning. More precisely, since those feedbacks are essentially costly (e.g., conducting meeting), with those short and long term horizons, overall PD performance become relative performances. Hence, in short run like time

$t = 100$, only $\frac{100}{\bar{t}}$ learning and feedback updates are happened, and it can be relevant to PD projects with very high learning and feedback costs. However, in the long run like $t = 900$, nearly $\frac{900}{\bar{t}}$ learning updates have occurred, and can be more related to the PD projects with low learning costs.

According to the graphs in Figure 1, the diverging performance behaviour of the former PD system (i.e., with within-subsystem bias) converges toward the latter system (i.e., with between-subsystem bias), or they tend to have less discrepancy in PD performance, as learning capability of teams increases (i.e., ϕ increases). Differently, also, as learning frequency decreases (i.e., $\bar{t}$ increases), both PD systems tend to have more discrepancy in PD performance patterns. This becomes more clear when we compare the left (right) graph in the panel (a) with the corresponding left (right) graph in the panel (b).

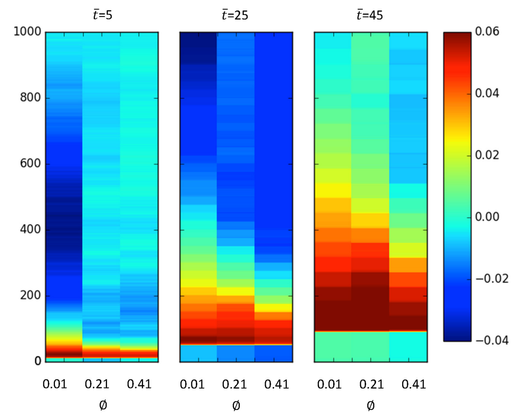


(a) $\bar{t} = 5$ (b) $\bar{t} = 45$

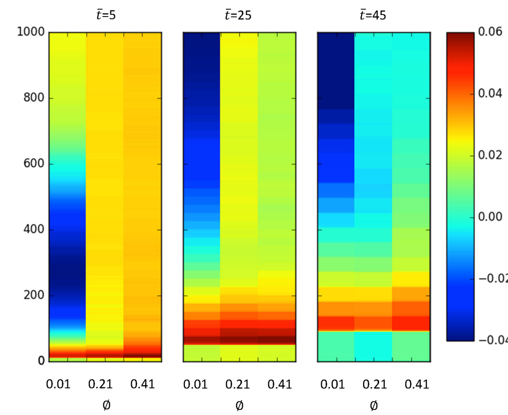
Figure 1: Average performance of 200 simulation experiments. In each panel, the left graph shows PD systems with within-subsystem bias, and the right one illustrates those with between-subsystem bias. Each simulation experiment is run for 1000 time steps, and averages are calculated for each time step. In the PD system, team interaction patterns are random with $PS = 0.2$. Also, the PD system encompasses five PD teams, each developing subsystems with three decision elements, and $K_h = 2$ and $K_b = 1$.

In order to have more abstract comparison between the PD systems, the following color-map graphs, in Figures 2 and 3, show the average performance of PD systems with within-subsystem bias minus those of with between-system bias. Therefore, a positive (negative) value in any cell, that is represented by color-bars at the right of each panel, indicates that PD systems with within-subsystem (between-subsystem) bias outperform those with between-subsystem bias (within-subsystem). In each panel, there are three color-map graphs that are arranged as follows. The PD systems with a high (i.e., $\bar{t} = 5$), medium (i.e., $\bar{t} = 25$), and low (i.e., $\bar{t} = 45$) learning frequencies, are shown at the left, middle, and right color-graphs, respectively. In addition, inside each color-map (e.g., the left plot on the left panel in Figure 2), there are three columns, that represent the teams' learning capability level, increasing in the range $\phi \in [0.01, 0.21, 0.41]$.

In general, according to the both panels in Figure 2, PD systems with within-subsystem bias outperform those with between-subsystem bias in short-run (e.g., $t = 200$). However, this relative performance changes in the long run (e.g., $t = 900$), and the latter strategy appears to be more beneficial for PD performance than the former one. Comparison across color-maps (i.e., learning frequency) on each panel shows an interesting pattern. Specially, short term performance dominance of PD projects with within-subsystem bias tend to remain for longer time, when learning frequency decreases (e.g., when we compare, from the left, the first and second color-maps on panel [a], we see somewhat larger red areas in the latter one). Moreover, comparison across panels reveals that when landscape becomes less rugged (i.e., landscape of PD system with $PS = 0.2$ is less rugged than that of the one with $PS = 0.7$), long-term performance dominance of



(a) Teams randomly interacting $PS = 0.7$



(b) Teams randomly interacting $PS = 0.2$

Figure 2: Average performance of 200 simulation experiments. Each simulation experiment is run for 1000 time steps, and averages are calculated for each time step, and within-subsystem PD systems minus those of with between-system bias is shown. Also, the PD system encompasses five PD teams, each developing subsystems with three decision elements, and $K_h = 2$ and $K_b = 1$.

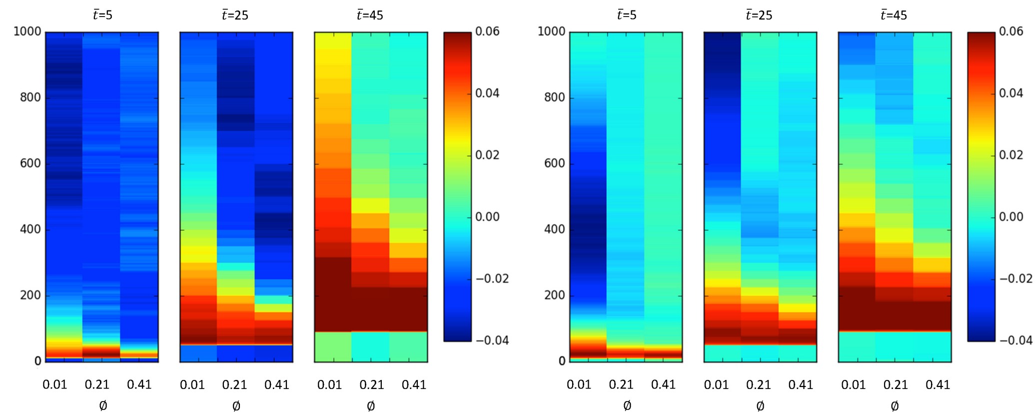
PD projects with between-subsystem bias occurs only when teams have lower learning capability levels (i.e., on panel [a], we observe large areas with blue color, whereas, on panel [b] blue areas occur only in the columns of color-maps related to $\phi = 0.01$). Overall, also, these observations are seen in the panels of Figure 3 that report results of PD projects with the centralized team interaction patterns, and different values for K_h and K_b.

It is worth discussing the micro-mechanisms that drive short and long term performance patterns. On the one hand, in PD projects with within-subsystem bias, design teams tend to concentrate their search process effort on their own subsystem design domains. Since such search domains are smaller than when all subsystems design domain is considered, consequently, teams converge to a local optimum in a short time period. That short convergence time is also reinforced as teams tend to focus on their own subsystem, and mainly ignore subsystems interactions. On the other hand, in PD projects with between-subsystem bias, each design team mainly concentrates its search process effort on its subsystem interactions with the other subsystems. Thus, their search domain becomes quite large, and with their limited learning capability, they need longer time to find superior design solutions. Such long time search, in comparison to PD systems with within-subsystem bias, also results in higher payoff for them.

We also observe that when landscape becomes less rugged, long-term performance dominance of PD projects with between-subsystem bias occurs only when teams have lower learning capability levels. That is likely to be associated with the presence of a high number of local optima on rugged landscape, on which teams with high learning capability, converge quickly to one of the local optima. That is similar to competency traps that has been discussed in the literature (Siggelkow and Levinthal 2003).

5 DISCUSSION

Two challenging features of coordinated search process are unknown interdependencies and communication constraints, which exist in both organizational and product development contexts (Sosa, Eppinger, and Rowles 2004, Sosa, Eppinger, and Rowles 2007, Puranam and Swamy 2016). Therefore, designers are usually engaged in a mutual adjustment process based on trial and error (Puranam and Swamy 2016). Such coupled



(a) $z = 5$, $K_h = 0$ and $K_b = 3$

(b) $z = 4$, $K_h = 3$ and $K_b = 1$

Figure 3: Average performance of 200 simulation experiments. Each simulation experiment is run for 1000 time steps, and averages are calculated for each time step, and within-subsystem PD systems minus those of with between-system bias is shown. In the PD system, team interaction patterns are centralized. Also, the PD system encompasses five/four PD teams, each developing subsystems with three/four decision elements.

learning processes are mainly shaped by organizational systems by modifying designers' mental models through a variety of centralized processes (e.g., planning) and their outputs (e.g., plans) (ibid).

This paper proposes a model for studying how PD teams' initial mental perceptions about interactions among subsystems affect the performance of PD projects. Initial mental models of PD teams about subsystem interactions are likely to affect the trajectory of coupled searches, and hence, performance. However, neither the extent of such effects nor the performance of different mental models have been examined. To fill some of this gap in the literature, the present study conceptualizes PD teams conducting collective searches over a performance landscape, and having incomplete knowledge of subsystem interactions. In this context, the initial mental models of teams affect their tendency toward paying attention to (and addressing) interactions of their elements with their own or other teams' subsystems. Specifically, such tendencies affect search domains that are highly selected and searched by teams. Two initial mental models are studies, one in which teams over-weight interactions of their own subsystem's elements, and another, in which each focal team over-weights interactions between its subsystem and other teams' subsystems.

The model ingredients and assumptions of this study are consistent with and relevant to other PD and organizational search models. First, assumption that PD teams have incomplete product architectural knowledge is consistent with key concepts. Ambiguity (Schrader, Riggs, and Smith 1993) and unforeseeable uncertainty (Sommer and Loch 2004) are features of new projects and are defined as the inability to identify and articulate the relevant variables and their effects. Product development teams developing a new product are unlikely to identify all of the project's consequential events (Pich, Loch, and Meyer 2002). Second, the proposed PD model, in which teams learn over time to allocate their search efforts to their own subsystems or potentially interdependent subsystems, is relevant to how organizational search processes should be structured, but for which only unitary search models are provided (Baumann and Siggelkow 2013). Third, the PD literature acknowledges that allocation of scarce resources to complex PD projects is highly challenging (Yassine and Naoum-Sawaya 2017). Also, PD investment efforts can be separated into modular improvements or establishing design rules—these design rules increase the modularity of product architecture (ibid). In the proposed model, teams learn endogenously over time to allocate their search efforts to their own subsystem (i.e., module improvement), or alternatively, teams may use their search efforts for interactions among subsystems (i.e., develop design rules and increase product modularity).

Overall, our findings indicate that when teams give more weight to paying attention to their own subsystem than to between-subsystem interactions, they concentrate their search efforts on small search domains, and converge to a local optimum in a short time period. However, in PD projects in which each design team mainly concentrates its search process efforts on interactions of its subsystem with the other subsystems, the search domains covered by teams becomes quite large, and with their limited learning capability, they take longer to find superior design solutions.

While current coordinated search models include only two agents, the proposed model extends those models to a setting with more than two agents and endogenous learning processes. Moreover, there are several opportunities to investigate previous PD models with designers learning about component interactions (Mihm, Loch, and Huchzermeier 2003).

REFERENCES

Alexander, C. 1964. *Notes on the Synthesis of Form*, Volume 5. Harvard University Press.

Allen, T. J. 1966. “Studies of the Problem-Solving Process in Engineering Design”. *IEEE Transactions on Engineering Management* (2): 72–83.

Barkoczi, D., and M. Galesic. 2016. “Social Learning Strategies Modify the Effect of Network Structure on Group Performance”. *Nature communications* 7.

Baumann, O., and N. Siggelkow. 2013. “Dealing with Complexity: Integrated vs. Chunky Search Processes”. *Organization Science* 24 (1): 116–132.

Camerer, C., and T. Hua Ho. 1999. “Experience-Weighted Attraction Learning in Normal Form Games”. *Econometrica* 67 (4): 827–874.

Cyert, R. M., and J. G. March. 1963. “A Behavioral Theory of the Firm”. *Englewood Cliffs, NJ* 2.

Eppinger, S. D. 2001. “Innovation at the Speed of Information”. *Harvard Business Review* 79 (1): 149–158.

Fang, C., J. Lee, and M. A. Schilling. 2010. “Balancing Exploration and Exploitation through Structural Design: The Isolation of Subgroups and Organizational Learning”. *Organization Science* 21 (3): 625–642.

Ghemawat, P., and D. Levinthal. 2008. “Choice Interactions and Business Strategy”. *Management Science* 54 (9): 1638–1651.

Kauffman, S. A. 1993. *The Origins of Order: Self Organization and Selection in Evolution*. Oxford University Press, USA.

Knudsen, T., and D. A. Levinthal. 2007. “Two Faces of Search: Alternative Generation and Alternative Evaluation”. *Organization Science* 18 (1): 39–54.

Knudsen, T., and K. Srikanth. 2014. “Coordinated Exploration Organizing Joint Search by Multiple Specialists to Overcome Mutual Confusion and Joint Myopia”. *Administrative Science Quarterly* 59 (3): 409–441.

Lazer, D., and A. Friedman. 2007. “The Network Structure of Exploration and Exploitation”. *Administrative Science Quarterly* 52 (4): 667–694.

Levinthal, D. A. 1997. “Adaptation on Rugged Landscapes”. *Management Science* 43 (7): 934–950.

March, J. G. 1991. “Exploration and Exploitation in Organizational Learning”. *Organization science* 2 (1): 71–87.

Mihm, J., C. Loch, and A. Huchzermeier. 2003. “Problem–solving Oscillations in Complex Engineering Projects”. *Management Science* 49 (6): 733–750.

Mihm, J., C. Loch, D. Wilkinson, and B. A. Huberman. 2010. “Hierarchical Structure and Search in Complex Organizations”. *Management Science* 56 (5): 831–848.

Pich, M. T., C. H. Loch, and A. D. Meyer. 2002. "On Uncertainty, Ambiguity, and Complexity in Project Management". *Management Science* 48:1008–1023.

Puranam, P., M. Raveendran, and T. Knudsen. 2012. "Organization Design: The Epistemic Interdependence Perspective". *Academy of Management Review* 37 (3): 419–440.

Puranam, P., and M. Swamy. 2016. "How Initial Representations Shape Coupled Learning Processes". *Organization Science* 27 (2): 323–335.

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 (1-2): 73–99.

Siggelkow, N., and D. A. Levinthal. 2003. "Temporarily Divide to Conquer: Centralized, Decentralized, and Reintegrated Organizational Approaches to Exploration and Adaptation". *Organization Science* 14 (6): 650–669.

Sommer, S. C., and C. H. Loch. 2004. "Selectionism and Learning in Projects with Complexity and Unforeseeable Uncertainty". *Management Science* 50 (10): 1334–1347.

Sosa, M., S. Eppinger, and M. Rowles, Craig. 2004. "The Misalignment of Product Architecture and Organizational Structure in Complex Product Development". *Management Science* 50 (12): 1674–1689.

Sosa, M. E., S. D. Eppinger, and C. M. Rowles. 2007. "A Network Approach to Define Modularity of Components in Complex Products". *Journal of Mechanical Design* 129 (11): 1118–1129.

Sutton, R. S., and A. G. Barto. 1998. *Reinforcement Learning: An Introduction*, Volume 1. MIT press Cambridge.

Terwiesch, C., and C. H. Loch. 1999. "Measuring the Effectiveness of Overlapping Development Activities". *Management Science* 45 (4): 455–465.

Thomke, S. H. 1997. "The Role of Flexibility in the Development of New Products: An Empirical Study". *Research policy* 26 (1): 105–119.

Thompson, J. D. 1967. *Organizations in Action: Social Science Bases of Administrative Theory*. Transaction Publishers.

Weinberger, E. D. 1991. "Local Properties of Kauffman's N-k model: A Tunably Rugged Energy Landscape". *Physical Review A* 44 (10): 6399–6413.

Yassine, A. A., and J. Naoum-Sawaya. 2017. "Architecture, Performance, and Investment in Product Development Networks". *Journal of Mechanical Design* 139 (1): 011101.

AUTHOR BIOGRAPHIES

MOHSEN JAFARI SONGHORI is a Postdoc Research Fellow at Department of Computer Science, Tokyo Institute of Technology. His research interests include new product development, and innovation, supply chain management. His email address is mj2417@gmail.com.

MOHAMMAD S. JALALI is a research faculty at MIT Sloan School of Management. He is interested in simulation and model estimation methodologies, and the applications of dynamic modeling for organizational and socio-technical problems. His email address is jalali@mit.edu.

TAKAO TERANO is a Professor in the Department of Computer Science, Tokyo Institute of Technology. His research interests include genetic algorithm-based machine learning, case-based reasoning, analogical reasoning, distributed artificial intelligence, cooperative agents, computational organization theory, and knowledge system development methodology. His email address is terano@dis.titech.ac.jp.