Research in the Sociology of Organizations

Emerald Book Chapter: Garbage Can in the Lab

Thorbjørn Knudsen, Massimo Warglien, Sangyoon Yi

GARBAGE CAN IN THE LAB

Thorbjørn Knudsen, Massimo Warglien and Sangyoon Yi

ABSTRACT

We develop an experimental setting where the assumptions and predictions of the garbage can model can be tested. A careful reconstruction of the original simulation model let us select parameters that leave room for potential variations in individual behavior. Our experimental design replicates these parameters and thereby facilitates comparison of human behavior with the original model. We find that the majority strategy of human subjects is consistent with the original model, but exhibits some behavioral diversity. Human subjects exhibit fluid diverse behaviors that improve coordination in the face of uncertainty, but hinder collective learning that can improve group performance.

INTRODUCTION

The "garbage can model" of organizational choice (Cohen, March, & Olsen, 1972) is an early agent-based formalization of the decision making process. It formally describes how decision makers allocate their efforts to choice opportunities that shift as a function of the way agents interact. As the

The Garbage Can Model of Organizational Choice: Looking Forward at Forty
Research in the Sociology of Organizations, Volume 36, 189–226

ISSN: 0733-558X/doi:10.1108/S0733-558X(2012)0000036011

actual interaction patterns are jointly determined by the organizational structure and the behavioral propensities of the agents, the model provides useful insights about the relationship between individual behavior and organizational performance, given a particular organizational structure. In this regard, the garbage can model presents an ideal case for experimental investigation of the way individual behaviors jointly influence organizational choice and performance. As the model provides precise predictions for a wide range of organizational structures, it lends itself to natural tests in the laboratory.

What is more important, while the limelight has been concentrated on the structural conditions that the model tries to capture and their effects on decision making patterns, little attention has been devoted to the behavioral assumptions embedded in the model – whether they are plausible, and whether changes in choice heuristics would affect the conclusions of the model itself. Will decision makers actually respond to the (shifting) demands of choice opportunities in the way hypothesized by the model? In particular, some of the conclusions of the garbage can model depend on the assumption that all agents adopt the same behavioral rule – an assumption that appear at odds with the findings from many experiments, that show a widespread heterogeneity of individual behaviors (Camerer, 2003), and that suggest that such heterogeneity may actually contribute to improve coordination among agents (Rapoport, 1995). If diverse behaviors should appear in the garbage can setting, what is their potential impact on organizational performance? Are there ways to improve organizational decision making in garbage can situations? Would subjects over time learn, and thereby improve organizational performance? Our aim is first and foremost to address these open questions and thereby provide an empirically grounded extension of the original model. But it is also to explore if a laboratory test of the garbage can model more generally may inspire experimental research that examines the relationship between individual choice behavior and organizational performance.

Experiments are certainly not new to organizational research – some actually date back to the very origins of the "Carnegie Mellon" school. One of the first laboratory studies published in a management journal showed that biased estimates and other anomalies at the microlevel are not necessarily reflected at the macro level (Cyert, March, & Starbuck, 1961). Still, organization science has until now seen relatively few attempts to bring theories to the lab, challenge their assumptions, and test whether interactions among individuals influence theory predictions at the organizational level. Over the last five decades, about 50 laboratory studies were published in the major management and organization journals (*Academy of Management Journal*, *Administrative Science Quarterly*, *Management Science*, and *Organization Science*).[1] However, most of these studies tested behavior at the individual or unstructured group level but rarely the organizational level of analysis. We believe it is time to move to the next level, and therefore aim to submit a theory at the organizational level to a test in the laboratory.

In this chapter, we develop a laboratory setting in which the predictions of the garbage can model can be experimentally tested. A careful reconstruction of the original simulation model enables us to select a set of structural parameters that leave room for potential variations in individual behavior. We then set up a laboratory experiment that replicates these parameters and thereby allows us to compare human behavior with the predictions from the original simulation model. Do human agents exhibit the same patterns of collective decision making as the computational agents? Do human agents solve more problems than the computational agents? On the ground of the observed results, we extend the original model, explore alternative behavioral hypotheses, and check their implications for organizational performance in garbage can settings.

THE ORIGINAL GARAGE CAN MODEL AND ITS USE IN A LABORATORY EXPERIMENT

The garbage can model is a true organizational classic, but despite being widely cited, it has rarely been challenged on empirical grounds. It has even been suggested that the model "has led a charmed life-to its disadvantage" (Bendor, Moe, & Shotts, 2001, p. 169). Indeed, a veil of confusion has often surrounded debates about the garbage can model, and both critics and supporters have often ended up seeing, in the brilliant metaphor of organizational decision making, the image of what they wanted to find – a veritable Rorschach test for social scientists.

A preliminary distinction between the broader theory of choice in "organized anarchies" and the model stricto sensu is helpful (Bendor et al., 2001). The broader theory is set in qualitative terms, and characterized by the idea that the ordinary "consequentialist" structure of decision making (both in the rational and bounded rationality views) has to be relaxed in favor of a more fluid one: An organization is "…a collection of choices looking for problems, issues and feelings looking for decision situations in which they might be aired, solutions looking for issues to which they might be the answer…" (Cohen et al., 1972, p. 1). In the tradition of bounded rationality, decision making is equated with problem solving, but the usual goal-driven sequence of problem solving is subverted in favor of a flow where all elements stand on equal footing and affect each other in a nonhierarchical way: "… one can view a choice opportunity as a garbage can into which various kinds of problems and solutions are dumped by participants as they are generated. The mix of garbage in a single can depends on the mix of cans available, on the labels attached to the alternative cans, on what garbage is currently being produced, and on the speed with which garbage is collected and removed from the scene." (Cohen et al., 1972, p. 2). The qualitative theory that has grown from the garbage can metaphor has been very influential in establishing a nonconsequentialist view of decision making (March & Olsen, 1984, 1989), and has provided an interpretive framework for numerous field studies. However, its very qualitative, verbal nature makes it compatible with a very broad range of behavioral assumptions and thereby limits its predictive power. On the other hand, the computer model that lies at the core of Cohen et al.'s (1972) original paper makes predictions about the outcomes of organizational decisions, given rather precise assumptions about individual behavior and a set of well-defined structural constraints. While it can only be taken as an instance or illustration of the broader framework of "organized anarchies," it provides an ideal background for experimental explorations. A closer look at the model will allow us to clarify the main issues of experimental inquiry and motivate our experimental design.

A Dynamic Coordination Problem

At its core, the garbage can computer model can be described as an instance of a coordinated effort allocation problem. As is well known, the basic elements of the model are decision makers, endowed with some capability to supply effort that will help solve problems (energy supply), problems that require effort to be solved, and "choice opportunities" (garbage cans) that define arenas where agents and problems meet. In each period during the process, decision makers allocate their effort among competing choice opportunities, and the coordination of their joint effort allows problem solving. Just like individuals are endowed with capabilities to supply effort, problems carry energy (effort) requirements. The basic principle is that a problem can be solved only when the (cumulative) energy spent on it exceeds its energy requirement. However, problems don't get solved in insulation.

They can only be resolved when the energy requirement of the whole set of problems in a choice opportunity (in a can) can be matched by the effort spent on the same choice opportunity. In that case, the choice opportunity is removed from the scene together with the problems attached to it. Thus, in Fig. 1, the can (a) will be removed and problems 1, 2, and 5 will be solved, while problems 3 and 4 will persist, as will can (b). The outcome is that one problem is solved by expenditure of the required amount of energy.

Thus, the garbage can model could be reinterpreted as a dynamic coordination game, in which multiple agents with no conflict of interests have to choose moves that jointly contribute to solving as many problems as possible. Since there are multiple ways in which agents may combine their actions to achieve that goal, each agent is exposed to a great amount of uncertainty about which actions to choose. This feature of the garbage can model is no different from static coordination games, where the agent's choice set not only depends on her own actions but also on the action of others. The added ingredient that sets the garbage can model apart, and in our view makes it a dynamic coordination game, is that the strategy sets change dynamically as a function of the agents' actions and the way problems move. It is therefore highly challenging for each agent to prefigure what the current relevant strategies are, to assess what the other agents prefigure, and how to pick a successful action given this uncertainty.

Thus, at least two features make the garbage can model very different from the kind of games that have received large attention both in game theory and experimental economics (Kagel & Roth, 1995). First, it is a genuinely dynamic coordination game, where choices at time t are affected

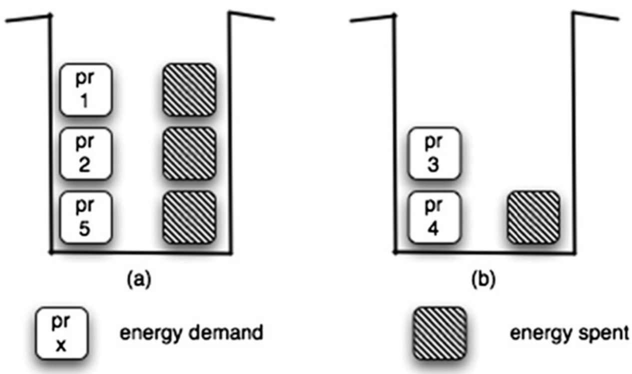


Fig. 1. Illustration of Problem Solving in the Garbage Can Model.

by choices at time $t-1$ and, in turn, affect those at time $t+1$. This intertemporal dependence is a defining feature of the garbage can model. It is present because effort is cumulative (once exerted, it transfers across periods) and because the choice opportunities available at any period, as well as the distribution of problems among available choices, depends on the agents' past behavior. Second, it is (to use a favorite adjective from Cohen et al., 1972) a ''fluid'' game where problems and choices enter the choice arena in unpredictable ways – trying to solve the game by backward induction would be a remarkably frustrating effort!

Organizational Structure

The fluidity of the coordination problem is shaped by some structural factors that give the garbage can model a peculiar organizational flavor. In particular, both the agents' and the problems' access to choice opportunities are constrained by organizational arrangements that define which agents (or which problems) can be attached to a given choice opportunity. Three types of structure are considered: ''unsegmented'' (Fig. 2a), ''hierarchical'' (Fig. 2b), and ''specialized'' (Fig. 2c). The same type of structure can apply to both decision makers and problems.

Finally, two additional sets of parameters affect the model behavior: (1) the sequential entry order of decision makers, choice opportunities, and problems and (2) the energy load (how the energy requirement of the problems relate to the energy capacity of decision makers).

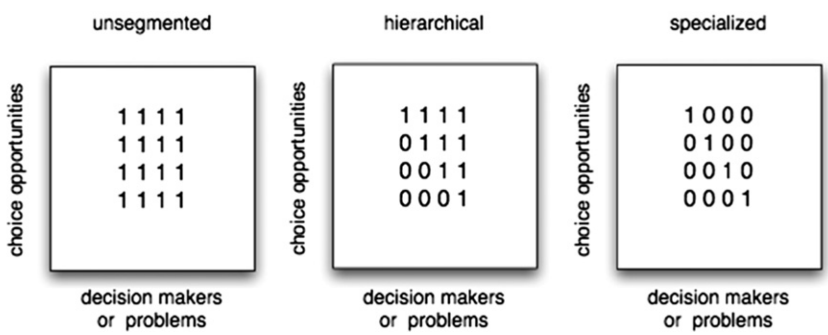


Fig. 2. The Definition of Access Structure in the Garbage Can Model.

Behavioral Assumptions

While the "verbal" theory of organized anarchies makes very loose assumptions about the behavior of the agents (preferences are ill-defined and possibly inconsistent, and discovered through action), the garbage can computer model makes simple and homogeneous assumptions about agent behavior. The agents are all assumed to follow the same rule of behavior, which can be stated as "allocate your effort to the (available) choice opportunity where the problems appear closest to solution" (or, in the language of the model, where the energy deficit is lowest). Clearly, this decision rule can be labeled as a simple decision making heuristic. The decision rule is not really motivated in the original garbage can paper, but can plausibly be associated with the idea of myopic adaptive behavior (look at the closest solution, ignore the future, and the intentions of the others), an idea that has a long tradition in adaptive models of organizational choice in the behavioral tradition. Remarkably enough, this is the same rule that problems follow in migrating from one choice opportunity to the other. The fact that problems have a degree of agency symmetric to that of decision makers is one of the most peculiar and intriguing aspects of the garbage can theory – but one on which we will not elaborate in this chapter.

Patterns of Decision Making

The garbage can model occupies a place of its own in the literature on decision making, not only because of its accent on dynamic, fluid coordination problems, but also for its peculiar characterization of the outcomes of organizational choice processes, which are described not only in terms of effectiveness and efficiency (how many problems get solved, with how much energy waste) but also in terms of patterns of choice – the decision making styles that emerge as the result of the interactions among decision makers and problems. Cohen et al. (1972) distinguish three characteristic patterns.

The first pattern is the most traditional pattern of choice and problem solving: after working on a choice opportunity (a) for some time, the choice is made ($t=2$), and both problems and the can disappear ($t=3$). As illustrated, this happens because the joint effort allocated to the problems in the can (gray) meets the requirement in that arena (white). This pattern is labeled as decision making by resolution (Fig. 3).

The second pattern emerges when (some) decision makers deal with choice opportunities that are devoid of problems (because these are attached somewhere else). Of course, in this case even a minimal effort will remove the choice opportunity from the stage, but will solve no problems (Fig. 4). This is referred to as decision making by oversight.

The third pattern is favored by the migration of problems to another choice opportunity, which reduces the energy demand associated with the "old" choice. This process captures the idea that choice opportunities are solved by diminishing requirements instead of increasing efforts. This pattern is appropriately labeled as decision making by flight (Fig. 5).[2]

One of the main results of the garbage can model is to show that non-resolution patterns of choice emerge and may constitute a significant portion of the overall decision output of an organization – and to show how

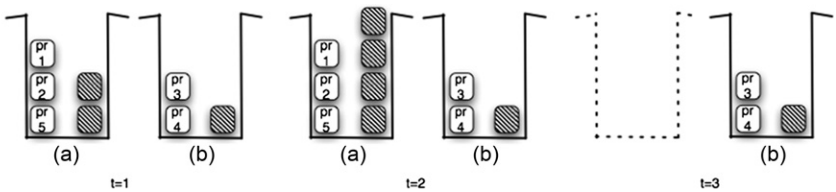


Fig. 3. Decision Making by Resolution in the Garbage Can Model.

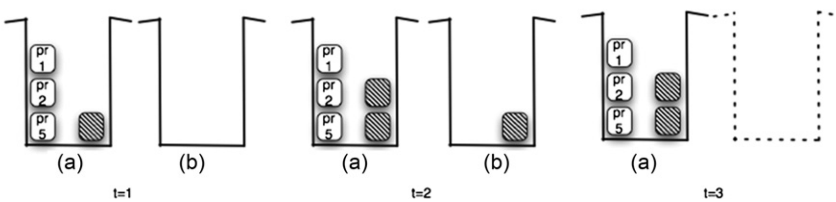


Fig. 4. Decision Making by Oversight in the Garbage Can Model.

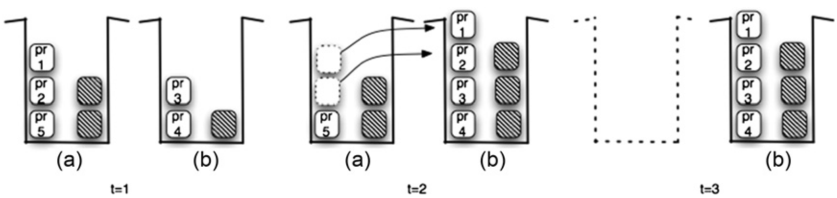


Fig. 5. Decision Making by Flight in the Garbage Can Model.

the mix of these three patterns varies with the structural arrangements of the organization.

Summary of Results from the Original Model

The original garbage can model provides three sets of notable results. First, decision making by flight and oversight is quite common – that is, decisions are made that do not solve any problems. The model generally suggests that resolution of problems is not the most common style of decision making. Rather, decision making by flight and oversight appears to be a major feature of the process of organizational choice. Second, higher problem difficulty (energy load) tends to increase problem activity, decision maker activity, decision difficulty, and the uses of flight and oversight. Third, the effect of alternative organization structures is characterized in terms of critical trade-offs. Specifically, three aspects of the decision processes define trade-offs that contribute to the overall efficiency of the organization: problem activity, problem latency, and decision time. Segmentation of the access structure tends to reduce the number of unresolved problems active in the organization, but at the cost of increasing the latency period of problems and, in most cases, the time devoted to reaching decisions. On the other hand, segmentation of the decision structure tends to result in decreasing problem latency, but at the cost of increasing problem activity and decision time. These results imply a fundamental trade-off that organization designers face in balancing different aspects of decision making efficiency. Especially, the issue of problem latency becomes important in situations where overlooked problems can seriously damage organizational performance. Another notable result is the tendency of decision makers and problems to track each other through choices. This result is consistent with the observation that decision makers might have a feeling that they are always working on the same problems in somewhat different contexts, mostly without results.

THE EXPERIMENTAL GARBAGE CAN AND THE SIMULATION BENCHMARK RESULTS

The garbage can model offers a set of distinct predictions about the choice patterns that would emerge from alternative structural constraints. Unfortunately, the space of parameters of the original simulations is too large for any realistic experimental design. The combinations of access structures for both decision makers and problems, their possible orders of entry, and levels of "energy load" are so numerous that it would be infeasible to organize a laboratory experiment on such a scale. Furthermore, it might not be particularly fruitful. For example, a specialized access structure, in the sense of the original simulation model, is an extreme that leaves little room for discretionary action of decision makers, since each agent is assigned to a specific choice opportunity (and so is each problem). Thus, in such case little or nothing could be added by an experiment to what is already shown by a computer simulation. Thus, we had to choose a set of parameters that is directly comparable to the ones in the original simulation model, but at the same time provide enough interesting alternatives to human subjects and nontrivial problem migrations among choice opportunities.

Choice of Access Structures

In the interest of examining human behavior in the context of dynamic coordination, we decided to explore the unsegmented access structure for decision makers (all subjects can act on all cans). This access structure allows interesting variation in individual behavior and also maximizes the challenge associated with achieving a coordinated outcome at the organizational level. As regards the access structure for problems, we were also motivated by our desire to maximize the challenge of achieving coordination at the organizational level. To capture this challenge, we used a hierarchical structure, which induces differential patterns in the way problems move across cans (since each problem has different constraints). This feature makes problem behavior hard to predict. Under the unsegmented structure, all problems would systematically mass-migrate to the can closest to solution, making problem behavior very predictable.

Downscaling the Model

The original model was downscaled to facilitate the implementation of the garbage can model in the laboratory.[3] Specifically, we defined an organization that has four decision makers and faces eight choice opportunities and 16 problems sequentially. An experimental run consists of 10 periods (against the original 20 periods), and in each of the first 8 periods one choice opportunity and two problems arrive at the organization. Each group of experimental decision makers participates in 10 consecutive runs.

In addition to scaling the model down, we also fixed the entry sequences of choice opportunities and problems. This was done to make it easier for subjects to learn, and for us to analyze if any learning was happening. We used the following two random entry sequences: {3, 5, 8, 2, 4, 6, 1, 7} for choice opportunities and {6, 4, 5, 12, 14, 2, 13, 10, 1, 3, 9, 11, 8, 16, 7, 15} for problems.

The Docking Problem

To get benchmark predictions that are directly comparable with the experimental results, we reconstructed the original computer model and ran it under the parameters of the experiment. Below, we briefly discuss the considerable difficulties that we met, and overcame, during our engagement with this reconstruction effort – a challenge that among *aficionados* is known as "the garbage can docking problem." Having obtained a reliable and almost perfect replica of the original model,[4] we could verify that the experimental version of the model predicted outcomes qualitatively similar to those obtained from using the original parameters (see Table 1). Furthermore, simulations confirmed that during each round there would be enough alternative choice opportunities available to decision makers – thus, the behavioral assumptions of the model could be tested in the lab.

Benchmark Results

Table 1 summarizes the simulation results for the "downscaled model" obtained by averaging over random entry sequences exactly as in the original model. All parameters are identical to those that we have used in the experiment.

Table 1. Simulation Results from the Simplified Garbage Can Model.

	Least Energy Deficit		
	Mean	SD	Range
Problems solved	2.64	1.00	2–6
Choices made	5.82	0.45	5–7
Proportion of choices by flight/oversight	0.60	0.07	0.43–0.67
Mean problem activity	5.61	0.04	5.50–5.63
Mean can lifetime	3.74	0.12	3.50–3.88

PERFORMING THE EXPERIMENT

In order to keep the experimental design manageable in the laboratory, we used the downscaled version of the model described in the previous section. The experiment was performed in a computerized lab where subjects could interact with a computer interface representing concurrent information on (1) available choice opportunities, (2) effort required to achieve resolution for each can (energy demand), (3) collective effort allocated to each can after the last period, and (4) the cumulative performance of the group of decision makers, measured in terms of the number of problems solved. Fig. 6 shows a snapshot of the interface (see Appendices A–C for a detailed description of the instructions). There are eight boxes on the screen, each of which

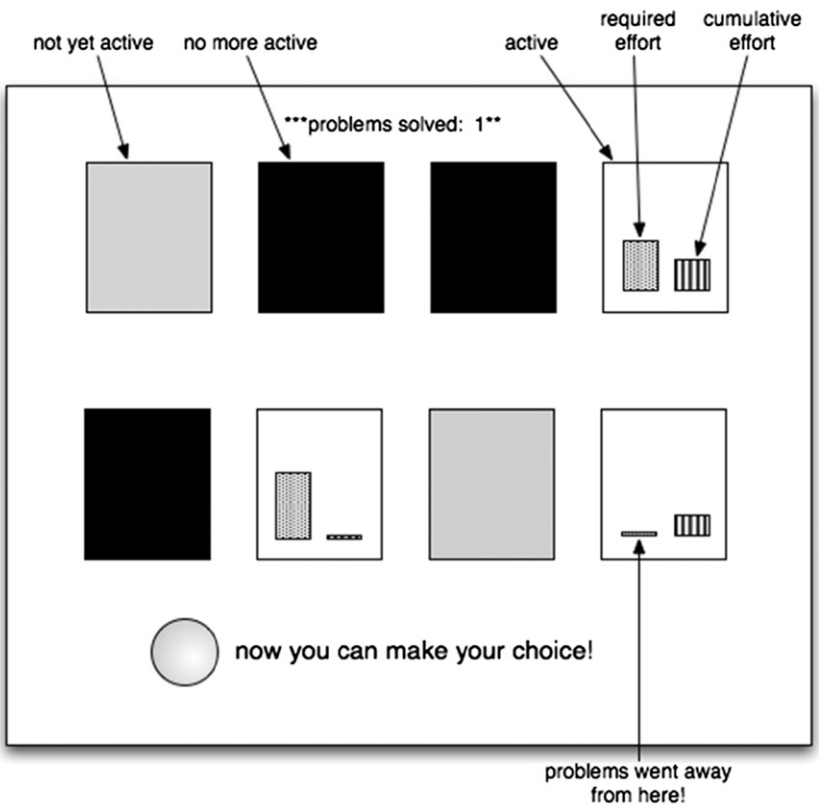


Fig. 6. A Snapshot of What the Subjects See on Their Computer Monitor.

represents a choice opportunity. A white box represents an active choice opportunity, while a gray box indicates latent opportunities that have not yet arrived. A black box represents a decision that has already been made – it is therefore not active anymore.

The height of the dotted bar shows how much effort is needed to solve the associated problems, that is, the sum of their required energy. The striped bar indicates the amount of accumulated energy devoted by the participants so far. Once the striped bar becomes as high as, or taller than, the dotted bar, the decision is made by solving the associated problems. A choice can also be made in a can if all the associated problems move to other active choice opportunities and the height of the dotted bar falls to zero (i.e., no dotted bar), while some effort has been accumulated in the same can – decision making by flight.

The players' decisions and feedback on the decision outcomes were aggregated and synchronized in each discrete period through the local network. For each discrete period, feedback on choice outcomes would only appear after *all* decision makers had made their choices. As in the original model, subjects had a fixed amount of energy to spend in each period, and this energy could be allocated to only one choice opportunity at a time (by selecting the appropriate can number on the computer keyboard). To allow subjects to become familiar with the task, the entry sequences of problems and cans were the same in all 10 rounds (as described in the previous section).[5]

Subjects received a fixed monetary reward for their participation plus a variable reward proportional to the group performance in terms of number of problems solved (thus there were no conflicting incentives). We report results from a total of 16 experiments (for a total of 64 subjects) that were run in the laboratories of the University of Southern Denmark, Odense, and Università Ca' Foscari, Venezia, during the fall of 2011. No significant differences in the two subject pools were observed, so we treated them as a unique subject pool. Prior to the experimental runs, we tested and calibrated the computer interface and the effectiveness of the instructions in three pilot studies that were conducted in the spring of 2011.

RESULTS FROM THE LABORATORY EXPERIMENT

Problem Solving Patterns

Table 2 provides a summary of the aggregate results that were extracted from all of the 16 experimental groups. The problem solving capabilities of

Table 2. Aggregate Results from Human Subjects in the Laboratory.

	Human Subjects		
	Mean	SD	Range
Problems solved	2.98	1.49	0–13
Choices made	6.31	0.56	5–7
Proportion flight/oversight	0.62	0.13	0–1
Mean problem activity	5.58	0.59	2.7–6.5
Mean can lifetime	2.71		

the experimental groups appear to be stunningly similar to those of the original computer model. At the end of the ten experimental runs, groups had on average solved 2.98 out of the16 problems that were introduced during each run. The standard deviation of problems solved was 1.49 and the range within a minimum of 0 and an impressive maximum of 13 solved problems. In the "downsized" computer model (Table 1), the average number of problems solved was 2.64 (within a range of 2–6 problems solved).

On average, 6.31 out of 8 choices were made, but resolution of problems was not the most common style for making decisions. Rather, decision making by flight and oversight accounted for 62% of the choices (against 60% of the computer model!). Interestingly, the decision style varied across trials, with extremes spanning the entire range of problem activity, with flight and oversight accounting for 0–100% of all choices that were made in a trial. The mean problem activity was 5.58 (number of periods a *problem* was active and attached to some choice), and the can lifetime was 2.71 (number of periods a *choice arena* was active).

To summarize, the human subjects performed fairly similarly to the computer version of the garbage can model. T-tests on the difference in means between the experimental results and the computer simulations show that the humans solve slightly more problems ($p = 0.037$), make notably more choices ($p < 0.001$), and have a lower can lifetime ($p < 0.001$). However, there is no difference as regards the proportion of choices made by flight and oversight ($p = 0.447$) or the mean problem activity ($p = 0.462$).

Interestingly, learning at the group level across the 10 rounds does not seem to account much for differences in performance across groups. In fact, there is no learning appearing from the aggregate data, nor do the best performing groups appear to learn more than the worst performing ones.

Fig. 7 shows evidence on experiential learning across trials in terms of problems solved averaged across all groups. The hypothesis that there is a

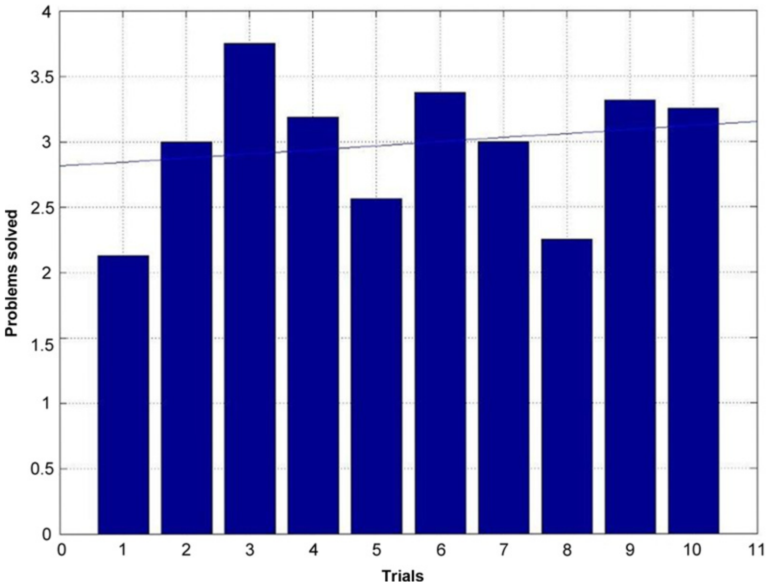


Fig. 7. Test of Experiential Learning in the Garbage Can. ($R^2 = 0.0322$, $p = 0.62$.)

learning effect can be rejected with $p = 0.62$. A closer examination of individual groups does not reveal any differences in this pattern across groups.

Individual Behavior

While the predictions of the garbage can model appear to be fairly successful, we have yet to examine individual behavior and whether it conforms to the assumptions of the garbage can model. As recalled above, the fundamental assumption of the model about individual behavior is that decision makers look for the choice opportunities where the energy deficit is lowest. Is this the strategy human subjects employ in the experiment? Of course, we have to restrict our analyses to the periods in which agents had at least two cans available, so that it is possible to discriminate among alternative behaviors. We have coded three behavioral strategies that apply to such cases. The first one, labeled "garbagecanness," corresponds to the application of the behavioral rule from the original model: look for the can with the *lowest* energy deficit. The second measure is the mirror of garbagecanness: look for the can with the *highest* energy deficit. We labeled this measure "optimism." Finally, we defined a "residual" category that captures any other moves than seeking cans with least energy deficit (garbagecanness) and cans with maximal energy deficit (optimism).

Table 3 summarizes the distribution of strategies among experimental subjects. It shows that garbagecanness is indeed the modal behavior, but accounts for only 57% of the choices. Thus, there is much room for behavioral diversity in individuals and groups. This result points to an interesting difference between human subjects and computer agents.

A natural question thus arises. If similar patterns of choice appear in the presence of differences in individual decision making behavior, is the result just driven by the structure of the choice problem, independently of what individuals do? In other words, to what extent can individual behavior affect the outcome of the decision process? The higher performance variance in human groups versus the computer model hints at the possibility that individual behaviors, and their distribution in groups, could matter. However, there are no clear patterns in the experimental observations that enable detection of factors associated with differential performance among the groups. As observed above, learning effects are negligible, and uncorrelated to group success. The relative incidence of "non-garbagecanness" is also uncorrelated to group success. So is success in facing the challenge of the garbage can just the result of chance?

While the experimental data provide no clear answer, the computer simulations can inform this question. Stimulated by the evidence of heterogeneity in individual behavior, we ran a version of the simulation model in which behavior was entirely random. Remarkably, if the behavior of the computer agents is simple randomization of choices, performance increases dramatically! (Table 4).

Table 3. Individual Choice Strategies.

	Mean
Garbagecanness	0.57
Optimism	0.34
Other choices	0.09
Entropy	0.54
Random benchmark	0.48

Table 4. Comparison of Human Subjects and Computer Agents with Different Behavioral Propensities.

	Mean		
	Experiment	Original Model	Random Behavior
Problems solved	2.98	2.64	6.04
Choices made	6.31	5.82	6.78
Proportion flight/oversight	0.62	0.60	0.42
Mean problem activity	5.58	5.61	4.85
Mean can lifetime	2.71	3.74	3.39

While we observe heterogeneous strategies in the experiment, the behavioral patterns in the laboratory observations cannot be reconciled with random behavior. Rather, the laboratory experiments indicated behavioral diversity within groups, as if individuals would adopt particular roles associated with garbagecanness ("always make the easy choices") or optimism ("always make the difficult choices").

We therefore ran the garbage can simulation model with different mixtures of garbagecanness and optimism. The result is reported in Fig. 8. It is easy to see that the mean problem solving performance increases in the number of agents that are ambitious (optimists). This is because the version of the model we used in the laboratory presents the human subjects with difficult problems, in the sense that the energy requirement is high. The results from the experiment cannot be predicted by any of the computer simulations shown in Fig. 8. However, an average across the four simulations comes remarkably close to the observed results, including the score of 2.98 problems solved.

The main difference that remains to be explained is the higher variance in performance of experimental groups. Since the entropy (related to decision) of the groups appears unrelated to performance, it is likely that this effect can be attributed to the intertemporal sequence of choices that is not captured by our averages. Postexperiment interviews with groups of subjects revealed that the adaptive behavior of unsuccessful subjects produced signals that confused the other players, an effect that resulted in unproductive cascades of failed mutual adaptation. In conclusion, the basis for successful human engagement with the kind of problems that are enshrined in the garbage can model is a *stable division* of strategies among decision makers who exhibit a significant amount of heterogeneity in behavior in terms of the energy deficit of the choices that are pursued.

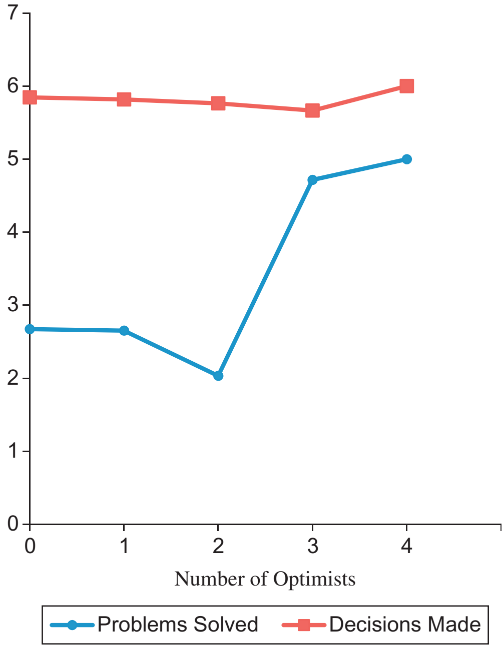


Fig. 8. Computer Simulation of Agents with Heterogeneous Behavior.

CONCLUSIONS

This experimental study, in conjunction with simulation analysis, was motivated by the fact that the garbage can model, despite its notable impact on the literature, has rarely been challenged on empirical grounds. We produced a reliable and almost perfect replica of the original computer model. We then used this clean version of the simulation model to construct the experimental environment in the laboratory and to generate predictions that could serve as a benchmark for the experimental results.

The analysis revealed a number of interesting results. First, we found that individual subjects systematically played strategies that are consistent with the behavioral assumptions made in the original model. Consequently, the human subjects performed fairly similar to the computer version of the garbage can model. This is a notable success story for the underlying theory.

Second, we observed that human subjects did not uniformly adopt the rule of pursuing least energy deficit (i.e., easy choice opportunities) that was used in the original computer model. In contrast, the human subjects followed strategies that preserved a significant amount of heterogeneity in behavior (e.g., pursuing difficult choice or switching between easy and difficult choices). The effect of the observed behavioral diversity was that groups of human decision makers outperformed the garbage can simulation.

This result implies that behavioral diversity (or, different preferences) may be an important factor of successful performance in dynamic coordination contexts. Differences in choice behavior serve as a capacity to flexibly cope with unpredictable arrivals of decision opportunities and associated problems. If the nature of future choice situations can be predicted, and if desirable behaviors are known ex ante, it would be possible to design effective incentive systems that could solve coordination problems. The reality, however, often deviates from this ideal situation. In a context characterized by an unpredictable environment and imperfect control of individual behaviors (i.e., fluidity in the garbage can model), behavioral diversity is a source of flexibility that promotes successful coordination.

We know from prior work in complexity theory that diversity can be beneficial for group performance. For instance, Scott Page (2007) showed how diversity could reduce collective error by improving the estimate of the expected value of some observation. While this idea explains why crowds may outperform individuals in a static environment, it does not explain why diversity can also be beneficial in a dynamic context characterized by uncertainty, mutual adaptation, and intertemporal path dependence. Minority games, like the famous El Farol Bar problem (Arthur, 1994), are more illuminating. In this problem, the chance of success of a strategy decreases with the number of agents adopting it, which in turn forces the agents to form divergent expectations and follow diverse behaviors. This idea is closely related to the emergence of diverse behaviors in our laboratory experiment. Indeed, the garbage can model can be thought of as a variant of the El Farol Bar problem, with the added complication that bars open and close at unpredictable hours.

Third, quite surprisingly the experiments showed no clear indication of learning at the aggregate level. Apparently, the garbage can setting derailed attempts to systematically enhance group performance through individual learning. Through interviews with the subjects, we found that the likely reason for the no-learning result was that subjects often confused each other. Just as heterogeneity in behaviors could increase performance, frustrating performance may induce changes in choices and increase behavioral heterogeneity. Thus, heterogeneity in behavior could be both the source and the consequence of performance.

This endogeneity between behavioral diversity and group performance has both theoretical and empirical implications. Theoretically, collective learning may not be compatible with dynamic coordination. Success in dynamic coordination is associated with a fluid context as it demands freedom to make simultaneous changes in individual choices. By contrast, success in individual learning requires a stable context where learning by someone could be "substituted" with learning by others (Levinthal & March, 1993). It is likely that these two demands are not always compatible. Further investigation into the conditions in which dynamic coordination and collective learning could complement each other will be a promising research topic.

Empirically, our results suggest that observed behavioral heterogeneity could be both a determinant and a consequence of performance. Disentangling cause and effect would require a quantification of the partial amount of behavioral heterogeneity that can be attributed to performance feedback. We suggest that future laboratory studies of interactive human behavior and group performance should be designed in a way that allows such decomposition.

In conclusion, we are impressed by the predictive performance of the simple behavioral theory that undergirds the garbage can model, and we are excited about the prospects of advancing an experimental organization science that submits organizational level phenomena to rigorous tests in the laboratory.

NOTES

1. Based on a simple JSTOR-search on these journals, conducted on February 12, 2012. Our search included any article with the term "laboratory experiment" in the abstract.

2. This figure actually shows a choice involving both flight and resolution – the flight of some problems diminishes the energy demand so that the remaining problem can be solved (Cohen et al., 1972, p. 8). In the extreme case of decision making by flight, all problems leave the choice opportunity and the decision solves no problems.

3. A simplified version of the garbage can model is used to reliably extract behavioral patterns from laboratory subjects. As the arrival rate of choice opportunities (one per period) and problems (two per period) are held identical to the original model, however, the situation (i.e., available choice opportunities and associated problems) from the perspective of an individual decision maker is equivalent to that of the original model. Note that a decision maker should make her choice without knowledge of others' choices, and hence the number of decision makers has little impact on the decision process at the individual level and the qualitative results at the organization level.

4. Perfection is measured by minimal deviation from the original model on vital statistics such as rate of problem resolution, mean problem activity, and proportion of choices by flight and oversight.

5. The software used for the experiments is available on request. It has been written in Processing, a dialect of the Java language.

ACKNOWLEDGMENTS

The authors thank Alessandro Lomi, an anonymous reviewer, and participants at the Garbage Can Model 40th Anniversary Conference, Stanford University, June 24–25, 2011. MW acknowledges support by the Fondazione Ca'Foscari-Swiss & Global.

REFERENCES

Arthur, B. (1994). *Increasing returns and path dependence in the economy*. Ann Arbor, MI: University of Michigan Press.

Bendor, J., Moe, T. M., & Shotts, K. W. (2001). Recycling the garbage can: An assessment of the research program. *American Political Science Review*, *95*(1), 169–190.

Camerer, C. (2003). *Behavioral game theory: Experiments in strategic interaction*. Princeton, NJ: Princeton University Press.

Cohen, M. D., March, J. G., & Olsen, J. P. (1972). A garbage can model of organizational choice. *Administrative Science Quarterly*, *17*(1), 1–25.

Cyert, R., March, J., & Starbuck, W. (1961). Two experiments on bias and conflict in organizational estimation. *Management Science*, *7*(3), 254–264.

Kagel, J. H., & Roth, A. E. (1995). *The handbook of experimental economics*. Princeton, NJ: Princeton University Press.

Levinthal, D. A., & March, J. G. (1993). The myopia of learning. *Strategic Management Journal*, *14*(S2), 95–112.

March, J., & Olsen, J. (1984). The new institutionalism: Organizational factors in political life. *American Political Science Review*, *73*(September), 734–749.

March, J., & Olsen, J. (1989). *Rediscovering institutions: The organizational basis of politics*. New York, NY: Free Press.

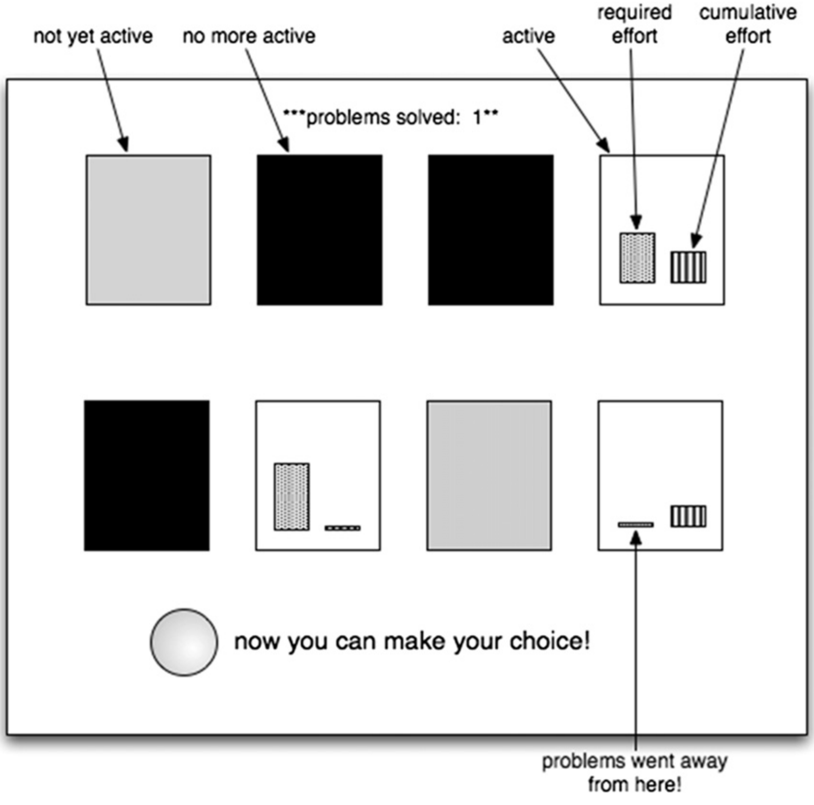
Page, S. E. (2007). *The difference – How the power of diversity creates better groups, firms, schools, and societies*. Princeton, NJ: Princeton University Press.

Rapoport, A. (1995). Individual market strategies in a market entry game. *Group Decisions and Negotiation*, *4*(2), 117–133.

APPENDIX A: INSTRUCTIONS TO SUBJECTS

Together with three other persons, you are going to participate in an experiment on decision making.

There will be 10 blocks of 10 rounds. At each round you will see 8 boxes on the screen. Each of these boxes represents a "task." Tasks can be active or inactive. Active tasks are indicated with a white color. Gray rectangles are the tasks that have not yet become active. Black tasks are "gone" – they are no longer active.



Within each task, there can be dotted bars and striped bars. Dotted bars show how much effort is needed to solve the problems currently attached to the task. Striped bars show how much (cumulative) effort has been spent by you and by the others to try to fulfill this particular task.

At each round you will be asked to make a decision on where to allocate your "effort" to help solving problems. For each round, you have a fixed amount of effort, which can be allocated to just one task in each round. To make a decision, you must press a key with the number corresponding to the task you are choosing. If you choose a task that is not active (i.e., which corresponds to a gray or black rectangle), your effort will be wasted.

You can decide when there is a green light on your screen. When there is a red light, you have to wait – others still have to make their own decision!

A task is fulfilled when the amount of cumulative effort (the striped bar) is at least as high as the dotted bar. When a task is fulfilled, it becomes black and is no longer active. Also, problems attached to a fulfilled task will disappear. However, problems may move from task to task, so the height of the dotted bar could change over time. Thus, you might find that there are no more problems attached to a task you have been working on!

The screen will report how many problems have been solved up to that moment.

Your goal as a group is to solve as many problems as possible. The number of problems solved will determine the probability to win 14 euros. The probability to win will be given by the number of problems you solved divided by 160 (which is the max).

You will also be rewarded with a fix amount of 7 euros for your participation.

Thank you for your participation!

APPENDIX B: PSEUDO CODE OF THE DOWNSCALED GCM FOR THE EXPERIMENT

Attributes and Actions of Agents

Agents	Attributes	Actions
Decision Maker	*Choice Opportunity*: The choice opportunity currently participating in, if any *Available Energy*: Available energy per period	*Search*: Find the choice opportunity with least energy deficit (excluding the contribution of her energy) among available ones *Move*: Participate in the found choice opportunity and update the *Choice Opportunity*.
Decision Problem	*Choice Opportunity*: The choice opportunity currently attached to, if any *Required Energy*: Required energy for resolution	*Search*: Find the choice opportunity with least energy deficit (excluding the contribution of its energy) among available ones *Move*: Attach itself to the found choice opportunity and update the *Choice Opportunity*. *Solve*: Remove itself from the *Attached Problems* of its *Choice Opportunity* and the Decision Problems of the *Organization*.
Choice Opportunity	*Attached Problems*: Currently attached decision problems	*Update*: Update the energy deficit based on currently *Attached*

	Participants: Currently participating decision makers *Energy Deficit*: Sum of the required energy by the attached problems minus the cumulative sum of the devoted energy by current and previous participants	*Problems* and *Participants*. *Make*: If having any attached problem, count as a "decision by resolution" and let *Solve* the *Attached Problems*; otherwise, count as a "decision by flight/oversight." Remove itself from the *Choice Opportunities* of the *Organization*.
Organization	*Decision Makers*: A fixed number of decision makers *Decision Problems*: Problems that came into the organization but not solved yet *Choice Opportunities*: Choice opportunities that came into the organization but not made yet *Access Structure*: Mapping decision problems to accessible choice opportunities *Decision Structure*: Mapping decision makers to accessible choice opportunities	*Allocate* 1st step: Each of the *Decision Problems* and *Decision Makers* does *Search*. 2nd step: Each of the *Decision Problems* and *Decision Makers* does Move. *Make Choices*: Each of the *Choice Opportunities* does *Update* its energy deficit, and *Make* if its new energy deficit is equals to or below zero.

Model Procedure

Initialization

Create an organization with four decision makers and set the organizational structure (access structure, decision structure).

Set model parameters: Available energy for each of the decision makers, required energy for a decision problem, and the entry sequence of the 16 problems and 8 choice opportunities.

Repetition (for 10 time periods)
At each time period, create a decision problem of the next kind according to the entry sequence, assign it the predefined required energy, and let it come into the organization and add itself to the set of decision problems.

At each second time, create a choice opportunity of the next kind according to the entry sequence, assign it an energy deficit of zero, and let it come into the organization and add itself to the set of choice opportunities.

The organization does allocate the decision problems and the decision makers within its boundary and make choices.

Finalization
Update model statistics: Problems solved, choices made, decisions by resolution, decisions by flight/oversight, mean problem activity, mean can lifetime, etc.

APPENDIX C: COMPUTER CODE FOR THE DOWNSCALED GCM FOR THE EXPERIMENT

Organization.java

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Random;

public class Organization{

   // Fixed parameters
   static final int numParticipants = 4, numDecisions = 8, numProblems = 16;
   static final int energyPar = 3, energyPro = 10;
   static final int endTime = 10;

   // Variable parameters
   public int accessStructure;     // (0) unsegmented, (1) hierarchic, (2)
specialized
   public int decisionStructure;    // (0) unsegmented, (1) hierarchic, (2)
specialized

   // Agents
   public ArrayList<Decision> decisions;
   public ArrayList<Participant> participants;
   public ArrayList<Problem> problems;

   // Statistics
   public int numResolution = 0;
   public int numOversightFlight = 0;
   public int problemLatency = 0;        // Periods of being latent in the
                                           organization
   public int problemActivity = 0;    // Periods of attaching to a
                                           decision
   public int participantActivity = 0;  // Number of flights of
                                           participants
   public int decisionDifficulty = 0;    // Active time within the
                                           organization

   // Others
   public int time = 0;
   public Random rand;
```

```
public Organization(int accessStr, int decisionStr, Random random){

    // Model Parameters
    accessStructure = accessStr;
    decisionStructure = decisionStr;
    rand = random;

    // Lists of agents
    decisions = new ArrayList<Decision>();
    participants = new ArrayList<Participant>();
    problems = new ArrayList<Problem>();

    // Participants in the organization
    for(int i=1; i<=numParticipants; i++) addParticipant
    (i, energyPar);

 }



/** Model procedure **/

public void proceed(){

    time ++;

    // Fly of participants searching for better decision opportunity
    for(int index=participants.size()-1; index>=0; index--){
       Participant aParticipant = participants.get(index);
       aParticipant.search();
    }

    // Fly of problems searching for better decision opportunity
    for(int index=problems.size()-1; index>=0; index--){
       Problem aProblem = problems.get(index);
       aProblem.search();
    }


    // Decision-making process
    for(int index=decisions.size()-1; index>=0; index--){
       Decision aDecision = decisions.get(index);
       aDecision.proceed();
    }

}
```

```
/** Move-In of Agents **/

public void addDecision(int importance){
    Decision aDecision = new Decision (this, importance);
    decisions.add(aDecision);

 }
public void addParticipant(int importance, double energy){
    Participant aParticipant = new Participant(this,
    importance, energy);
    participants.add(aParticipant);
}

public void addProblem(int importance, double energy){
    Problem aProblem = new Problem(this, importance, energy);
    problems.add(aProblem);
}


/** Simulation Experiment: Given Entry Sequence **/

public static void main (String[] args) {

    // Write results to the file
    File file = new File("GCM_experiment - given sequence
    (10 periods).txt");

    try{

     BufferedWriter writer = new BufferedWriter(new FileWriter(file,
     false));
     Random rand = new Random(System.currentTimeMillis());

     writer.write("run"+"\t"+"structureA"+"\t"+"structureD"+"
     \t"+"energyPro"+"\t"+"solvedProblems"+"\t"+"madeDecisions"+
     "\t"+"resolutionRate" +"\t"+"proLatency"+"\t"+"proActivity"+
     "\t"+"parActivity"+"\t"+"decDifficulty"); writer.newLine();

     // Simulation runs
     for(int run=1; run<=160; run++){

      // Given entry sequence
      int[] sequenceDec = {3,5,8,2,4,6,1,7};
      int[] sequencePro = {6,4,5,12,14,2,13,10,1,3,9,11,8,16,7,15};

      /** Variable model parameters **/
      for(int as=0; as<3; as++){     // Access structure
      for(int ds=0; ds<3; ds++){     // Decision structure
```

```
      for(int ep=8; ep<13; ep+=2){     // Energy per problem
       // Create an organization
       Organization org = new Organization(as, ds, rand);
       System.out.println("["+run+"] structureA = "+as+", structureD =
"+ds+", energyPro = "+ep);

       // Model procedure
       while(org.time < endTime){

             // In-stream of decisions and problems
             if(org.time < numDecisions){

                 // Entry of one decision
                 org.addDecision(sequenceDec[org.time]);

                 // Entry of two problems (mapping id to importance)
                 org.addProblem((sequencePro[2*org.time]+1)/2, ep);
                 org.addProblem((sequencePro[2*org.time+1]+1)/2, ep);

        }

        // GCM decision-making process
        org.proceed();

     }


     // Write results to the file
     writer.write(run+"\t"+as+"\t"+ds+"\t"+ep
          +"\t"+(numProblems - org.problems.size())
          +"\t"+(numDecisions - org.decisions.size())
          +"\t"+(double)org.numResolution/(numDecisions -
org.decisions.size())
          +"\t"+(double)org.problemLatency/numProblems
          +"\t"+(double)org.problemActivity/numProblems
          +"\t"+(double)org.participantActivity/numParticipants
          +"\t"+(double)org.decisionDifficulty/numDecisions);
     writer.newLine();

    }
    }
    }
          writer.flush();
          }
          writer.close();

      }catch(IOException ex){
```

```
            System.out.println("Couldn't write to the file! Check if it's in
            use.");
            ex.printStackTrace();
        }

    System.out.println("Simulation done.");

    }
   }
```

Decision.java

```
import java.util.ArrayList;
import java.util.Iterator;

public class Decision{

    public Organization org;
    public int importance;

    public ArrayList<Participant> participants;
    public ArrayList<Problem> problems;

    public double energyDeficit;        // Current energy deficit
    public double cumulativeEnergy;     // Accumulated energy so far


    public Decision(Organization p_org, int p_importance){

        org = p_org;
        importance = p_importance;

        energyDeficit = 0;
        cumulativeEnergy = 0;

        participants = new ArrayList<Participant>();
        problems = new ArrayList<Problem>();

    }

/** Behave as a decision opportunity **/

public void proceed(){
    // Update statistics (periods of staying in the organization)
    org.decisionDifficulty++;
```

```
    // Update energy accumulation and deficit
    updateState();

    // Attempt to make a decision (no decision without any contribution)
    if(cumulativeEnergy > 0 && energyDeficit <= 0) makeDecision();
}


/** Update energy states **/
public void updateState(){

    // Update accumulated energy
    Iterator<Participant> itPar = participants.iterator();
    while(itPar.hasNext()) cumulativeEnergy += itPar.next().energy;

    // Update energy deficit
    energyDeficit = -cumulativeEnergy;
    Iterator<Problem> itPro = problems.iterator();
    while(itPro.hasNext()) energyDeficit += itPro.next().energy;

}


/** Decision making **/
public void makeDecision(){

    // Update statistics
    if(problems.isEmpty()) org.numOversightFlight++;  // No problem so
                                                      far, or all flew
                                                      out
    else org.numResolution++;      // With problem(s) resolved

    // Release participants
    Iterator<Participant> itPar = participants.iterator();
    while(itPar.hasNext()){
       Participant aParticipant = itPar.next();
       aParticipant.latent = true;
       aParticipant.decision = null;
    }

    // Release problems
    Iterator<Problem> itPro = problems.iterator();
    while(itPro.hasNext()){
       Problem aProblem = itPro.next();
       aProblem.latent = true;
       aProblem.decision = null;
     org.problems.remove(aProblem);
    }
```

```
    // Remove from the decision list
    org.decisions.remove(this);

   }

}
```

Participant.java

```
import java.util.ArrayList;
import java.util.Iterator;

public class Participant{

   static final int UNSEGMENTED = 0, HIERARCHIC = 1, SPECIALIZED = 2;

   public int importance;        // The smaller, the more important
   public double energy;         // Available energy

   public boolean latent;        // Whether attaching to a decision
                                    opportunity
   public Decision decision;    // Current decision opportunity where
                                    participating

   public Organization org;      // Organization

   public Participant(Organization p_org, int p_importance, double
   p_energy){

      org = p_org;
      importance = p_importance;
      energy = p_energy;

      latent = true;
      decision = null;
}


/** Find and Go for a more attractive decision opportunity **/
public void search(){

    // Identify accessible decision opportunities
    ArrayList<Decision> accessibleDecisions = new ArrayList
    <Decision>();

    Decision aDecision;
    Iterator<Decision> it = org.decisions.iterator();
```

```
    while(it.hasNext()){

       aDecision = it.next();

       if((org.decisionStructure == UNSEGMENTED) ||
          (org.decisionStructure == HIERARCHIC  &&  aDecision.
importance >= this.importance) ||
          (org.decisionStructure == SPECIALIZED  &&  aDecision.
importance == this.importance)){

              accessibleDecisions.add(aDecision);
       }
    }


    // If no accessible decision, stay latent
    if(accessibleDecisions.isEmpty()){
       latent = true;
       decision = null;
    }
    // If only one accessible decision, fly to it
    else if(accessibleDecisions.size() == 1){
       flyTo(accessibleDecisions.get(0));
    }
    // Otherwise, identify and fly to the most attractive one
    else{

       // Exclude current contribution for fair comparison
       if(!latent) decision.energyDeficit += energy;

       // Identify the decision with least energy deficit
       ArrayList<Decision> targets = new ArrayList<Decision>();

       it = accessibleDecisions.iterator();
       Decision best = it.next();
       targets.add(best);

       while(it.hasNext()){
          aDecision = it.next();
          if(aDecision.energyDeficit < best.energyDeficit){
             best = aDecision;
             targets.clear();
             targets.add(aDecision);
          }
        else if(aDecision.energyDeficit == best.energyDeficit){
             targets.add(aDecision);
        }
     }
```

```
      // Turn it back to the original value
      if(!latent) decision.energyDeficit -= energy;

      // Fly to the chosen decision (it could be the current one)
      flyTo(targets.get(org.rand.nextInt(targets.size())));

     }

}


/** Fly to the (new) decision opportunity **/
public void flyTo(Decision newDecision){

     // If it's the current one, just stay there.
     if(!latent && (decision == newDecision)) return;

     // Move to the new decision opportunity
     newDecision.participants.add(this);
     if(!latent) decision.participants.remove(this);

     // Update status
     decision = newDecision;
     latent = false;

     // Count the number of flies(activities)
     org.participantActivity++;
    }


}
```

Problem.java

```
import java.util.ArrayList;
import java.util.Iterator;

public class Problem{

   static final int UNSEGMENTED = 0, HIERARCHIC = 1, SPECIALIZED = 2;

   public int importance;     // The smaller, the more important
   public double energy;      // Required energy

   public boolean latent;     // Whether attaching to a decision
                                 opportunity
```

```
   public Decision decision;   // Current decision opportunity where
                                  attached

   public Organization org;     // Organization


   public Problem (Organization p_org, int p_id, double p_energy){

      org = p_org;
      importance = p_id;
      energy = p_energy;
      latent = true;
      decision = null;
}
/** Find and go for a more attractive decision opportunity **/
public void search(){

    // Identify accessible decision opportunities
    ArrayList<Decision> accessibleDecisions = new ArrayList
    <Decision>();

    Decision aDecision;
    Iterator<Decision> it = org.decisions.iterator();
    while(it.hasNext()){

       aDecision = it.next();

       if((org.accessStructure == UNSEGMENTED) ||
         (org.accessStructure == HIERARCHIC  &&  aDecision.importance
>= this.importance) ||
        (org.accessStructure == SPECIALIZED  &&  aDecision.importance
== this.importance)){

            accessibleDecisions.add(aDecision);
     }
    }


    // If no accessible decision, stay latent
    if(accessibleDecisions.isEmpty()){
       latent = true;
       decision = null;
       org.problemLatency++;
    }
    // If only one accessible decision, fly to it
    else if(accessibleDecisions.size() == 1){
       flyTo(accessibleDecisions.get(0));
    }
```

```
    // Otherwise, identify and fly to the most attractive one
    else{

       // Exclude current contribution for fair comparison
       if(!latent) decision.energyDeficit -= energy;

       // Identify the decision with least energy deficit
       ArrayList<Decision> targets = new ArrayList<Decision>();
       it = accessibleDecisions.iterator();
       Decision best = it.next();
       targets.add(best);

       while(it.hasNext()){
          aDecision = it.next();
          if(aDecision.energyDeficit < best.energyDeficit){
             best = aDecision;
             targets.clear();
             targets.add(aDecision);
          }
          else if(aDecision.energyDeficit == best.energyDeficit){
               targets.add(aDecision);
        }
     }

     // Turn it back to the original value
     if(!latent) decision.energyDeficit += energy;

     // Fly to the chosen decision (it could be the current one)
     flyTo(targets.get(org.rand.nextInt(targets.size())));
  }

}


/** Fly to the (new) decision opportunity **/
public void flyTo(Decision newDecision){

    // Update model statistics
    org.problemActivity++;

    // If it's the current one, just stay there.
    if(!latent && (decision == newDecision)) return;

    // Move to the new decision opportunity
    newDecision.problems.add(this);
    if(!latent) decision.problems.remove(this);
```

```
    // Update status
    decision = newDecision;
    latent = false;

   }

}
```