AFRL-RI-RS-TM-2008-2
Final Technical Memorandum
January 2008

THE STRUCTURE OF COMPLEX PROBLEMS

Columbia University

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TM-2008-2 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

ROGER J. DZIEGIEL, JR.
Work Unit Manager

/s/

JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government’s approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE	*Form Approved* **OMB No. 0704-0188**

1. REPORT DATE *(DD-MM-YYYY)* JAN 2008	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* Jun 06 – Aug 07

4. TITLE AND SUBTITLE

THE STRUCTURE OF COMPLEX PROBLEMS

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8750-06-1-0180

5c. PROGRAM ELEMENT NUMBER
62304E

6. AUTHOR(S)

John B. Lee
Duncan J. Watts

5d. PROJECT NUMBER
AC52

5e. TASK NUMBER
00

5f. WORK UNIT NUMBER
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Columbia University
Department of Sociology
1180 Amsterdam Ave
New York NY 10027

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RIED
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TM-2008-2

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. WPAFB #08-0087

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The design of collective problem solving experiments requires that experimenters be able to control for task difficulty and problem structure across a variety of potential experimental tasks. Controlling for these factors allows experimenters to isolate the effects of organizational and communication structure from the nature of the task, and allows for comparison between experimental results and real-world problem solving efforts. This report develops a model of problems that accounts for the major features of "complex problems" and shows that the model is capable of generating "hard" problems.

15. SUBJECT TERMS
Organizational architectures, automatic network configuration, modeling network configuration

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Roger J. Dziegiel, Jr.
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UL	13	**19b. TELEPHONE NUMBER (*Include area code*)** N/A

Table of Contents

List of Figures

Introduction

Many scholars of organizational behavior have long thought of organizations as information processing machines (Shaw 1932; Bavelas 1950; Leavitt 1951; Shaw 1954; Guetzkow and Simon 1955; Becker and Baloff 1969; Dodds, Watts et al. 2003; Kearns, Suri et al. 2006). In other words, the activity of organizations is to take some inputs and filter those inputs through the constituent parts of the organization in order to generate an output. Unlike other information processing machines and tools, however, organizations are often tasked with solving problems that are ill-defined, display complex (or even unknown) sets of interdependencies, or for which there is no known method of directly constructing a solution (Miller, Galanter et al. 1986; Hargadon and Bechky 2006). Nevertheless, anecdotal evidence suggests that some organizations are capable of solving such problems; manufacturing companies and investment banks are able to recover from disasters that catastrophically damage their physical infrastructure (Nishiguchi and Beaudet 1998; Buenza and Stark 2003), engineering firms are able to construct complex machines (Kidder 1981; Hamilton 2001), and technology firms are able to take what they've learned in one product sector to make inroads into others.

What makes these organizations different from other, less successful or less responsive, organizations? One possible explanation is that the way in which information processing and decision-making are structured in organizations with high problem solving capability is different from less capable organizations. It is also plausible that seemingly hard problems are not so hard, or that organizations which appear to flourish despite uncertainty or volatility "get lucky." In order to disentangle these alternative explanations, however, requires that we be able to differentiate between *classes* of problems and organizations. Some types of organizations may be better at some classes of problems than others. Therefore, to gain a complete picture of problem solving capability requires understanding the "space" created by organizational structures and problem classes.

While case studies can reveal much about particular organizations solving particular problems in particular circumstances, it is not clear to what degree such reports are generalizable or representative. Controlled experiments, in which subjects perform stylized tasks that distill the key features of real-world tasks, would allow researchers to systematically explore how organizational structure interacts with problem structure to produce different levels of problem solving capability. This report presents a precise model of problems that can be interpolated between "simple" and "complex" problems. We will also present results from simulations comparing two simple optimization algorithms with simulated annealing, a method commonly used in combinatorial optimization. We will begin by defining the key features of "complex problems." We will then present a model of problems that can be tuned to generate problems ranging from "simple" to "complex." Finally, we will present our simulation results.

Complex Problems

A situation is problematic (is a problem) when the current state of a system differs from some defined goal state. Problem solving is simply the movement of the system from the current state to the goal state either directly or via some number of intermediary states (Dunbar 1988). The simplest problems can be solved directly. Such problems are well understood, and moving the system to the goal state is simply a matter of applying a series of known operations to the inputs. Such problems are easily solvable by individuals, or can be broken down or "chunked" into a set of sub-problems that can be solved completely independently by a group of individuals.

Multiplying a large set of numbers is an example of such a problem. Although an individual can construct a solution directly, very large sets may require a prohibitive amount of time. A group of individuals could solve this problem by (1) dividing the set of numbers amongst themselves, (2) independently computing the product of each subset of numbers (in parallel), and then (3) aggregating the final result by computing the product of the intermediary products. Although the input size for this example problem is large and benefits from the collective capabilities of a group of individuals, input size, alone, is not the sole differentiator of "complex problems." Designing an automobile or a microprocessor are also problems with large "input sizes" but one would not put them in the same class as computing the product of a set of numbers.

In addition to input size, complex problems are also characterized by partial decomposability and ambiguity. Partial decomposability refers to a property of complex problems wherein problems can be chunked into smaller but partially interdependent sub-problems. Sub-problems are partially interdependent in the sense that the optimal solution to each sub-problem depends on the solution selected for some, but much fewer than all, other sub-problems. The problem of multiplying a large set of numbers is decomposable, but completely so — the product of each subset of numbers depends only on the numbers in the subset and on no other subset. In contrast, the optimal design for an automobile engine may depend on the desired range of transmissions being offered which may, in turn, be influenced by other requirements. However, the design of the engine is unlikely to be directly determined by such things as upholstery choices or headlight design. The web of sub-problems that make up complex are neither completely independent nor are they each completely dependent on all other sub-problems.

In addition to being partially decomposable, complex problems are also ambiguous in the sense that (1) the initial problem statement may not contain enough information to directly construct a solution and (2) the "payoff landscape" is unknown and, typically, rugged. Many problems begin vaguely, particularly product design problems. Software development, for example, typically begins with a set of "functional requirements" that must be developed in detail before technical design can even begin. For example, a requirement such as "easy to use" needs to be turned into a set of rules for how users will interact with the software before developers can begin to design the code underlying the final software product. Likewise, automobile designers must go through a process of benchmarking competitors' products and isolating key features when attempting to satisfy requirements such as "best in class braking."

The payoff landscape describes the payoff or fitness of each possible solution. In complex problems, problem solvers must search the solution space in some way. Rugged payoff/fitness landscapes have many local optima that typical hill-climbing methods can get trapped on (Kauffman and Levin 1987; Levinthal 1997; Levinthal and Warglien 1999). Hill-climbing describes a heuristic for searching solution spaces for optima. Under hill-climbing, problem solvers starting with an arbitrary or random solution look at similar nearby solutions and select the best solution. By repeatedly applying this rule, problem solvers can eventually find optima. However, a weakness of hill-climbing heuristics is that they get "stuck" on local optima — solutions that are better than all of the nearby solutions.

The three features described above — input size (or "scale"), partial decomposability, and ambiguity — characterize complex problems. Together, these three properties make complex problems difficult by requiring too many resources to be solved by an individual problem solver, but at the same time creating coordination problems for groups of problem solvers. The large input size and partial decomposability of complex problems means that, although such problems can be broken up and distributed amongst a group of individuals, those individuals will need to somehow coordinate their solutions. Furthermore, individuals may need to select locally sub-optimal solutions (e.g., a braking system with slightly less power or a microprocessor with less performance), in order to find the globally optimal solutions (e.g., a lightweight fuel efficient vehicle or a device that runs for extended periods on battery power). Any model of problems will need to account for these features. In the next section, we will describe a model, which we call the Correlated Payoff Model, that can describe a broad range of problems ranging from very simple problems that can be solved by an individual, through large problems that can be broken up and solved in parallel, to so-called complex problems.

The Correlated Payoff Model

Decomposable problems, of which complex problems are a subset, can be thought of as collections of elemental sub-problems that cannot be further subdivided. Each elemental sub-problem has an associated local payoff function that maps candidate solutions to payoffs independently of the solution selected for any other sub-problem. Such collections are representative of completely modular problems, in which sub-problems consist of modules that can be changed without affecting the payoff functions of other sub-problems. Problems of this type can be easily divided amongst a group of individuals and solved in parallel. In the correlated payoff model, *local payoff functions* associated with sub-problems are given by the following equation:

$$\pi_i(x) = a_i e^{-(b_i(x-c_i))^2}$$

The equation shown above describes a function with a single maximum of *a* centered on *c* that decays as a function of *b*. Figure 1 illustrates such a function with *a=1* and *c=50*. In a decomposable problem, each individual sub-problem *i* can have its own values of *a*, *b*, and *c*, as indicated by the subscripts. Although each sub-problem may have a distinct optimal solution, the entire problem can nevertheless be broken down into sub-problems and solved piecewise without the need to coordinate between individual problem solvers.

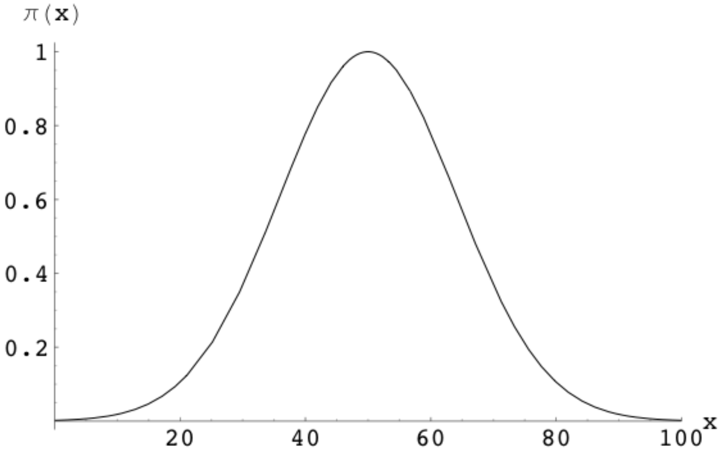

Figure 1 An example local payoff function with a maximum of 1 centered on 50.

Complex problems, however, cannot be solved in such a manner because the optimal solution for a given sub-problem depends on the solutions selected for some number of other sub-problems. To account for this interdependency, we introduce a new *pairwise payoff function* defined as follows:

$$\pi_{ij}(x_i) = a_i e^{-\left(b_i\left(x_i - c_i\left(x_j\right)\right)\right)^2}$$

The pairwise payoff function describes the way in which sub-problem *i* is dependent on sub-problem *j* and has the general form as the local payoff function assigned to sub-problems. However, the function is defined in relation to a sub-problem *j*, on which *i* is dependent. The function $c_i(x_j)$ determines the optimal pairwise solution and is a function of the solution selected for sub-problem *j*.

Because the pairwise payoff function relates two sub-problems, we can now extend the idea of a collection of independent sub-problems to model complex problems. A complex problem is simply a collection of interdependent sub-problems with interdependencies defined by the $c_i(x_j)$s. Complex problems can be described as directed networks (or graphs) of sub-problems with edges indicated interdependent sub-problems. Furthermore, $c_i(x_j)$ may simply evaluate to a constant, in which case *i* is independent of *j*. This model, therefore, is not limited to generating sets of completely independent or completely interdependent sub-problems, but is also capable of generating problems with arbitrary interdependence structures — i.e., what we have called “complex problems.”

However, the pairwise payoff function only defines payoffs for *pairs* of sub-problems. In order to allow for any interdependence structure, we need to account for the possibility of sub-problems having multiple dependencies on other sub-problems. To do this, we introduce a *neighborhood payoff function* defined as follows:

$$\phi_i(x_i)=\frac{1}{k_i+1}\left(\pi_{ii}(x_i)+\sum_{\{j|(i,j)\in E\}}\pi_{ij}(x_i)\right)$$

where k_i is the number of dependencies *i* has on other sub-problems and *E* is the set of edges in the directed graph describing the interdependency structure of the sub-problems. When a sub-problem has no dependencies, the neighborhood payoff function is the same as the local payoff function. When a sub-problem is dependent on other sub-problems, the neighborhood payoff function is the average of the pairwise payoff functions in which a sub-problem is involved. This has the effect of reducing the maximum achievable payoff for a sub-problem and reflects the reality that any sub-problem solution is a tradeoff between the various other sub-problems on which it depends.

We still need a *global payoff function* that describes the fitness of the complete solution. A natural global payoff function is simply the sum of some neighborhood payoffs. We define the global payoff function as follows:

$$\Phi(x_1,x_2,\text{☺},x_N)=\sum_{i=1}^{N}\phi_i(x_i)$$

For a decomposable (but not complex) problem consisting of completely independent sub-problems, the global payoff is simply the sum of the local payoffs. Such problems are easily maximized by independently optimizing each local payoff function. In contrast, problems in which sub-problems are interdependent cannot be maximized in such a fashion because the interdependencies make neighborhood optima contingent on other sub-problems. The degree of sub-problem interdependence is determined by the structure of the interdependency graph as well as by $c_i(x_j)$. In the following section, we will simulate and compare three optimization heuristics on complex problems with stochastic $c_i(x_j)$s.

Heuristic Optimization of Complex Problems

In the previous section, we described a model of problems that accounts for the three primary features of complex problems: scale, partial decomposability, and ambiguity. Together, these features make it difficult to find optimal solutions. Individuals working alone are faced with a large number of possible solutions, while groups of problem solvers must contend with the need to coordinate their actions in order to address the effects of sub-problem interdependency. In this section, we compare two simple optimization heuristics to simulated annealing, a widely used optimization method intended to avoid getting trapped at local optima — a danger that is particularly prevalent in the sorts of payoff landscapes associated with complex problems.

Simulated annealing is a combinatorial optimization method which attempts to avoid getting trapped in local optima by probabilistically accepting moves that result in less-optimal states — “uphill” moves if one is minimizing, “downhill” moves if one is maximizing. In simulated annealing, systems are gradually “cooled” by decreasing the probability of accepting a payoff-decreasing move. An annealing algorithm consists of three main components:

1. A way to generate a new configuration *Y* from a current configuration *X*
2. A cost or payoff function which can be evaluated for any configuration
3. A cooling schedule that “anneals” the system from a random solution to a “frozen” solution

New configurations are generated stochastically, often by an underlying Markov chain. The simulated annealing algorithm takes these three rough steps:

1. Generate a new configuration Y from the current configuration X
2. If $\Phi(Y) \geq \Phi(X)$, then accept the new configuration
3. Otherwise, accept the new configuration with probability $\alpha = Exp\left[-\beta\left(\Phi(Y) - \Phi(X)\right)\right]$

$\Phi(X)$ is the payoff associated with configuration X, and β is a tunable parameter representing “cooling schedule” which is selected so that α decreases over the course of the simulation. In the early parts of a simulated annealing run, most or all payoff-decreasing moves will be accepted, effectively moving the system randomly around the configuration space. As the simulation runs, the amount of the configuration space gets smaller and smaller as payoff-decreasing moves are increasingly rejected. In our simulations, we defined the cooling schedule as follows:

$$\beta_t = \frac{1}{r^t t_0}$$

where *t* is the timestep, *r* is a constant ratio (in this case, 0.998) and t_0 is the initial temperature (1000).

How well does simulated annealing compare to regular hill-climbing on complex problems? We generated problems using the correlated payoff model and compared a simulated annealing algorithm with two simple hill-climbing algorithms. In the simultaneous hill-climbing algorithm, each individual makes an uphill “move” assuming that all other individuals will not change their solutions. At each timestep, all individuals simultaneously update their solutions. In the sequential hill-climbing algorithm, each individual makes a greedy uphill move in turn. Rather than every individual simultaneously updating their solutions, each individual updates in turn and therefore has perfect information about the solutions contributed by other individuals

The generated problems consisted of 80 elemental sub-problems with random interdependencies. Each sub-problem was dependent on five other sub-problems, chosen at random. We defined $c_i\left(x_j\right)$ stochastically. For each value of x_j we selected a c_{ij} randomly selected from a Normal distribution with mean c_i and a standard deviation of 50. For each algorithm, we generated 1,000 such random problems, and ran each algorithm for 2,000 timesteps. Figure 2 shows the cumulative distribution of global payoffs achieved in the last timestep for each of the three algorithms. the sequential heuristic does not perform as well as the simultaneous heuristic, even when all individuals are permitted to update their solution in each timestep, effectively multiplying the number of timesteps evaluated. Going from simultaneous to sequential updating changes the sort of information available to individuals. Both heuristics are "greedy" in the sense that individuals only make "uphill" moves. However, in the sequential heuristic, individuals are presented with perfect information about the solutions contributed by others. In contrast, when the simultaneous heuristic is used, individuals have an imperfect or "noisy" view of others' solutions. This imperfect information seems to help with problem solving, moving the system as a whole out of local maxima.

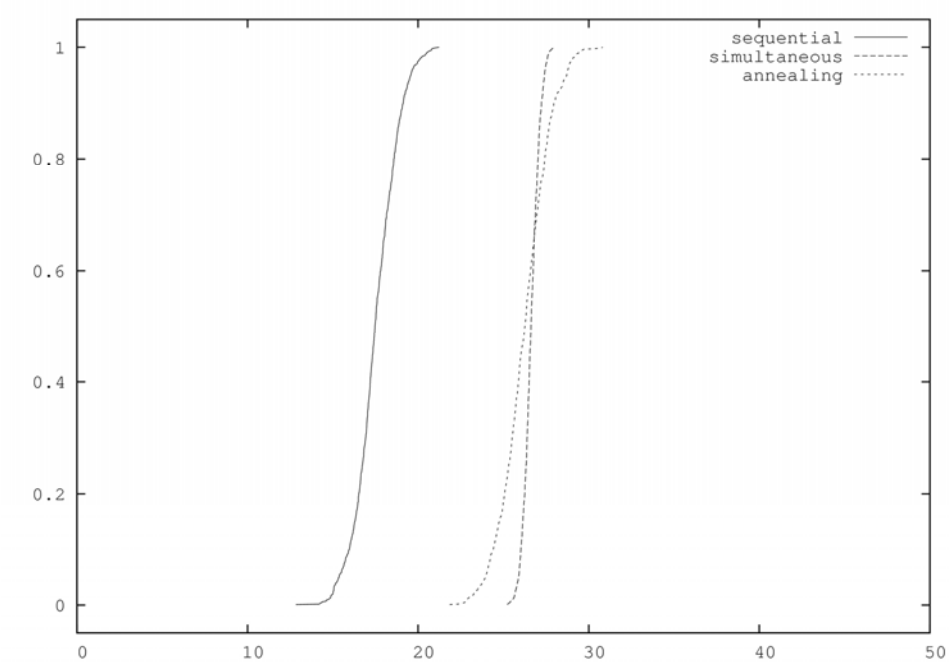

Figure 2 Cumulative distribution of final global payoffs for three optimization strategies.

It is surprising to find that the simultaneous hill-climbing and simulated annealing algorithms performed so closely. That simulated annealing *only* works as well as a standard hill-climbing strategy, and not better, suggests that the model at the very least generates problems that actually are difficult to solve.

Conclusion

In this report, we have described a flexible model of problems that can account for features of problems ranging from very simple problems to very complex problems. We have also presented some simulation results that suggest the model does indeed generate “hard” problems. Although we have not gone so far as to design a laboratory experiment, in detail, the work presented here addresses the main issues with doing human-subjects experiments in collective problem solving and lays the foundation for a future empirical work. Without a model of problems, it is difficult to compare potential experimental tasks, or to compare results between different human-subjects studies.

A model of problems provides a means to compare the experimental tasks subjects are asked to do. Such a model also provides a starting point to compare stylized experimental tasks with problems encountered in the real-world. In fact, the model of problem presented in this report provides a framework within which to build experimental tasks. Alternatively, existing task designs can be “fit” to the model and compared or adjusted to achieve the desired problem structure.

Finally, although the organizational structures studied in our simulations mirrored the problem structure, it is not necessarily the case that organizations must share the same structure as the problems they are trying to solve. Future human-subjects experiments will need to compare organizational structures not only against a single problem structure, but against a range of problem structures.

References

Bavelas, A. (1950). "Communication Patterns in Task-Oriented Groups." Journal of the Acoustical Society of America **22**(6): 725-730.

Becker, S. W. and N. Baloff (1969). "Organization Structure and Complex Problem Solving." Administrative Science Quarterly **14**(2): 260-271.

Buenza, D. and D. Stark (2003). "The Organization of Responsiveness: Innovation and Recovery in the Trading Rooms of Lower Manhattan." Socio-Economic Review **1**: 135-164.

Dodds, P. S., D. J. Watts, et al. (2003). "Information Exchange and the Robustness of Organizational Networks." Proceedings of the National Academy of Sciences **100**(21): 12516-12521.

Dunbar, K. (1988). Problem Solving. A Companion to Cognitive Science, Blackwell**:** 289-298.

Guetzkow, H. and H. A. Simon (1955). "The Impact of Certain Communication Nets upon Organization and Performance in Task-Oriented Groups." Management Science **1**: 233-250.

Hamilton, D. P. (2001). "Intel gambles it can move beyond PC with new microprocessor." The Wall Street Journal: 1.

Hargadon, A. B. and B. A. Bechky (2006). "When Collections of Creatives Become Creative Collectives: A Field Study of Problem Solving at Work." Organization Science **17**(4): 484-500.

Kauffman, S. and S. Levin (1987). "Toward a General Theory of Adaptive Walks on Rugged Landscapes." Journal of Theoretical Biology **128**: 11-45.

Kearns, M., S. Suri, et al. (2006). "An Experiment Study of the Coloring Problem on Human Subject Networks." Science **313**: 824-827.

Kidder, T. (1981). The Soul of a New Machine, Atlantic Monthly Press.

Leavitt, H. J. (1951). "Some Effects of Certain Communication Patterns on Group Performance." Journal of Abnormal and Social Psychology **46**: 38-50.

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

Levinthal, D. A. and M. Warglien (1999). "Landscape Design: Designing for Local Action in Complex Worlds." Organization Science **10**(3): 342-357.

Miller, G. A., E. Galanter, et al. (1986). Plans and the Structure of Behavior, Adams, Bannister, Cox.

Nishiguchi, T. and A. Beaudet (1998). "The Toyota Group and the Aisin Fire." Sloan Management Review **40**: 49-59.

Shaw, M. E. (1932). "A Comparison of Individuals and Small Groups in the Rational Solution of Complex Problems." The American Journal of Psychology **44**(3): 491-504.

Shaw, M. E. (1954). "Some Effects of Problem Complexity Upon Problem Solution Efficiency in different Communication Nets." Journal of Experimental Psycoholgy **48**: 211-217.