An Experimental Study of Open Innovation using MASTERMIND®[1]

Benjamin Hak-Fung Chiao[2]

University of Michigan 8/30/04 Version 6.000

Abstract: This paper presents the first experimental results on open innovation, which is defined to be a method to solve problems with other people by revealing some or the complete history of algorithm already used. An important example is open source. Our data from human subjects show that non-modular payoff structure drives the convergence to a Nash equilibrium, in which commission price to helpers converge to zero but helpers will not stop solving problems for others. By non-modularity, we mean that the total production (or payoff) of a team is zero if either one of its members fails to produce at least at a certain level. In the experiment, subjects produce by solving a variant of a popular board game called MASTERMIND. Theoretically, free-riding leads to zero commission price. This removes a signaling function of price for the difficulty levels of work remaining. Empirically, however, it is not sufficient to cause the catastrophic outcome of zero payoff. This provides a basis for us to hypothesize that open innovation is a key explanation because it allows subjects to directly observe the history of work already done and potentially direct more resources to the more difficult tasks.

---

[1] MASTERMIND is a registered trademark of Hasbro International Inc.

[2] I gratefully acknowledge the advice from Yan Chen and Andrew Schotter. I also benefited from the discussions with Charlie Holt, Josh Lerner, Jeffrey Mackie-Mason, Theo Offerman, Charlie Plott, and Neslihan Uler. Sam Matti provides excellent programming. The software and experiment are designed with the help of Vicky Chang, Doug Halket, Seung-ju Paik, and Anwar Ruff. This research is made possible by the financial support from the Center for Experimental Social Science at New York University.

1. Introduction

We report some theoretical and experimental results on open innovation. Open innovation is defined here as a method to solve problems with other people by revealing some or the complete history of algorithm already used[3]. An important example is open source[4].

An economic analysis is provided in which open innovation is operated within the structures of modular and non-modular team production[5]. By non-modularity, we mean that the total production (or payoff) of a team is zero if either one of its members fails to produce at least at a certain level. We further derive a Nash equilibrium for the non-modular payoff structure.

The experiment is conducted on networked computers with human subjects in the laboratory of the Center for Experimental Social Science at New York University. The author

[3] Open innovation partially overlaps with the concept of user innovation (there is a website dedicated to this area organized by Franke, Karls, Lakhani, and von Hippel, http://userinnovation.mit.edu, visited August 2, 2004). One difference is that in open innovation, not only end users but developers and firms can also participate in the innovation process.

[4] A simplistic view of open source is that one can access and modify the source code of a computer program. Quite a few recent publications offer more detailed discussions. (Weber 2004) argues that open source development conforms to economic and political principles and its development is guided by standards, rules, decisionmaking procedures, and sanctioning mechanisms. (Lerner and Tirole 2002) highlight the extent to which labor economics, especially the literature of career concerns, can explain many of these project features. (Chiao 2003) compares the open source problem to the economic property right development of China's transition during the last twenty-five years from a communist regime to a market-based economy. (von Krogh, Spaeth, and Lakhani 2003) develop a theory to explain the strategies and processes by which new people join the existing community of software developers, and how they initially contribute code. (von Hippel and von Krogh Forthcoming) propose that open source software development is an exemplar of a compound model of innovation that contains elements of both the private investment and the collective action models prevalent in organization science. They point out that the private investment model assumes returns to the innovator results from private goods and efficient regimes of intellectual property protection, while the collective action model assumes that under conditions of market failure, innovators collaborate in order to produce a public good.

[5] (Baldwin and Clark 2003) also study the modularity issue in open source and claim that modularity reduces free-riding.

develops a software that extends a popular board game called MASTERMIND[6]. MASTERMIND captures some characteristics of computer source code because both work histories are observable, stepwise verifiable, reversible, path independent in causing failures, not unique to the solution, and considered fun to work on for some people[7].

The experiment retains the rules of MASTERMIND except that subjects can post games to the public pool–the place where everyone in the lab can see the complete history of moves of the posted games. Each game posted must be accompanied by a (possibly different) commission. The one who solves the posted game gets the commission. Any one who solves a posted game must clone the posted game first so a copy of the history of work already done will be displayed on the screen for this person to continue the work. In each session, everyone is allotted three games in each of the ten periods. There is a $1 potential reward for either solving a game by oneself or having other people to solve it through the public pool. Each period ends when the time limit is reached.

We compare two treatments. In the first treatment, the potential reward is immediately credited to the subjects' earnings. In the second treatment, the potential reward is credited only if all allotted games for every subject are solved. A total of two sessions are conducted. In Session A, the second treatment is run for five periods followed by five periods using the first treatment. In Session B, the order of the treatments is reversed.

The major findings that we have are:

[6]There are previous works that extend ready-made games to run experiments. For instance, (Andreoni and Varian 1999) ran an experiment using a card game a few years ago.

[7]See Section 3 for a more elaborate discussion of these characteristics.

1. In the second treatment, a Nash equilibrium is for everyone to set commission to zero and everyone solves any games posted. From the data, the average commission tends to decrease in the second treatment. In Session B, the commission converges quickly to zero with almost all commission converges to zero in the last period. In Session A, the convergence is a bit slower but with a significant portion of commission set at zero.

2. Monotonous prices imply that subjects fail to coordinate to achieve a seemingly pareto dominating equilibrium in which the commission is ranked according to the difficulty of the game posted. Effort will then be more efficiently directed to solve the difficult games within the time limit. However, it is sometimes difficult to assess the difficulty of a game. Price perhaps can only be a rather imperfect signal for the difficulty level. This problem is further aggravated when the payoff is non-modular. The non-modular structure drives the convergence to a Nash equilibrium of monotonous zero commission, which completely removes a signaling function of price as a measure of difficulty levels. However, the data show that it is not sufficient to cause the catastrophic outcome of zero payoff. This provides a basis for us to hypothesize that open innovation is a key explanation because it allows subjects to directly observe the history of work already done and potentially direct more resources to the more difficult tasks.

In the next section, we discuss the rules of MASTERMIND. Then we explore the similarities and dissimilarities between MASTERMIND and computer source code in Section 3. Section 4 presents a Nash equilibrium for a non-modular payoff structure. Section 5 describes the experimental setting. Data are analyzed in Section 6. Section 7 concludes. In Appendix 1, we review the mathematics literature of the strategies that solve the traditional (standalone) MASTERMIND. Appendix 2 contains some graphs.

2. The Rules of MASTERMIND

The rules for MASTERMIND are simple. In a popular version of MASTERMIND, there are pegs in 6 colors used by a combination–breaker and pegs in black and white used by the combination–setter. For each game, a combination of 4 color pegs is set secretly by the combination–setter so the search space is 6^4. The combination–breaker's task is to guess the secret combination. For each guess, the combination–setter uses black and/or white pegs to give hints to the combination–breaker. A black peg means that the color and position of a guessed peg are correct; a white peg means that the color of a guessed peg is correct but not the position.

Mathematically, the secret combination is defined as $S = [s_1, s_2, ..., s_N]$, where s_n is the n^{th} peg with color s, $s \in [1, k]$, k = the number of colors, and N = the length of the combination or the number of slots.

The t^{th} guess is defined as $h^t = [h_1^t, h_2^t, ..., h_N^t]$, where h_n^t is the n^{th} peg with color h, $h \in [1, k]$. $H = [h^1, ..., h^t]$ is a history of guesses[8].

At this point, the interested reader may read Appendix 1 to see how mathematicians solve

[8] Each hint is the pair of numbers $b(h^t, S)$ and $w(h^t, S)$, that is, the number of black and white pegs. $b(h^t, S)$ is the number of subscripts n such that $s_n = h_n$. $w(h^t, S) = [\max_p b(h^t, p)] - b(h^t, S)$, where p is a vector from the set of vectors containing all permutations of the S vector. See (Chvatal 1983).

this game. In Appendix 1, there is some indication that one can have a different assessment of difficulty level of the same posted game. This is because the number of remaining moves and time needed to find the secret combination varies, which depend on the strategy one uses.

3. Open Source and MASTERMIND

We argue that the computer source code is similar to G in MASTERMIND in the following aspects[9]:

1. **Observable History**. The source code is a written algorithm to solve a problem; a history of guesses is a revealed method to break the secret combination. This is not to say that the algorithm written is necessarily sufficient to solve the problem nor that others can continue the same algorithm by studying the written algorithm[10].

2. **Stepwise Verifiability**. Both provide a means to verify the algorithms through steps. The hints directly reduce the search space. There are various hint equivalents in the source code that may mean different things. The hint equivalents could be debug messages or the direct output of a computer program or of its individual modules.

3. **Reversible Work**. In MASTERMIND, any inefficient guesses, those that do not

[9] Since the code is a form of intellectual property, a more general research question is to investigate links between G and intellectual property.

[10] We will need to distinguish between complete and incomplete algorithms. This is because some open source software requires more than just the source code to make it work, which we call the corresponding code an incomplete algorithm. In addition, in software engineering, code obsfucation converts typical code into code that is readable, compilable, but non-comprehensible by human.

eliminate the search space, will not make any subsequent guesses harder. Subsequent combination–breakers can always do at least as good if they ignore the inefficient guesses[11]. This is perhaps with the exception that one may need more time to eliminate(mentally) inefficient guesses if there are too many guesses on the screen. In the source code, any digital information can be re-initialized.

4. **Path Independence of Failures**. In the source code, an error in, say, the last line of code can produce completely erroneous output irrelevant of how many lines of excellent code are written. For example, an omission of a punctuation can render a program non-compilable. In our experiment, the reward only depends on the correct guess irrelevant of how close are the previous guesses to the secret combination.

5. **Non-Unique Strategies**. The strategies for both are not unique.

6. **Elements of Fun**. In the real world, consumers need to purchase MASTERMIND to play it. In the programmer world, there is a sentiment that solving problems is fun, see (Torvalds and Diamond 2001).

On the other hand, they differ in that MASTERMIND usually allows for a limited number of guesses whereas there is no limit to the amount of code that can be written. However, the experiment is designed such that there is no limit in the number of guesses.

[11] These reversible works differ from the collaboration on a painting in that the whole picture may be ruined if, say, the oil is spilled by a person.

4. A Nash Equilibrium

We will show that if we abstract out the repeated game consideration due to multiple periods in a session, a Nash equilibrium for the non-modular payoff structure is for everyone to set commission to zero and everyone solves any games posted[12]. The use of modular structure in the experiment is basically used to contrast the change of commission when we change the treatment into a non-modular structure. Therefore, we will omit the proof of the Nash equilibria for the modular production.

For the non-modular production, the payoff for each individual in each period is:

$$Y_i = (\prod_{j}^{M} g_j)(\sum_{j \in I} g_j + \sum_{k}^{N_1+N_2} p_k)$$

where

1. g_j is an indicator variable for the j^{th} allotted game, $j = 1, ..., M$, where M is the sum of allotted games for all subjects in the same period. g_j equals to 1 if the j^{th} allotted game is solved. Otherwise, g_j equals to 0. So $\prod_j^M g_j$ takes the value of zero if any one of the allotted games in the period are not solved. This is the so called O–ring property (Kremer 1993)[13].

2. $\sum_{j \in I} g_j$ is the number of games in the set of solved games, which were originally allotted

[12] We will not proceed to find other Nash equilibria for the non-modular structure, which call for additional assumptions that are not essential to the main points of the paper.

[13] Mathematically, in (Kremer 1993), the value of the O–ring production is $Y = \prod_i q_i$, where $q_i \in [0, 1]$, which is the effort or quality level of individual i. It is so named to capture the idea that a very insignificant part of a system can cause a complete failure. This is exemplified by the explosion of the space shuttle *Challenger* in 1986, which was caused by the failure of the O–rings.

to individual i. Note that the games are not necessarily solved by individual i because he or she can post games to the public pool for others to solve.

3. $p_k \in [-\$1, \$1]$ is the commission received from and pay to other subjects for the $N_1$and N_2 transactions, respectively. Note that $N_2 \leq \|I\|$, which is the number of elements in I. This is to ensure that subjects cannot pay more than what they could possibly have.

Theorem 1 *In a non-modular payoff structure, a Nash equilibrium is for everyone to set* $p_k = 0, \forall k$ *and solve all games in the public pool. This is called the zero-price monotone equilibrium.*

Proof. Suppose that all games are solved in the public pool, then the lower bound of Y_i is zero, which is equal to the value of Y_i when at least one game is not solved. Given this, for all i, setting $p_k = 0$ for individual i increases Y_i because everyone solves any game in the public pool at any price. ■

The zero price captures the idea of a free-riding problem in open source discussed in (Baldwin and Clark 2003). In addition, there is a significant problem with this type of equilibrium. In the situation in which the games can be ranked according to their difficulty levels (for example, through the commission price), the zero-price monotone equilibria (in fact, any equilibria with monotone and non-zero price) fails to provide this signaling function of price. Especially under a time limit, it is harder to direct more resources to the more difficult games without this signal. It is then tempting to conclude that this equilibrium is not pareto optimal because difficulty level could have been ranked by price. However, this ranking could simply be not feasible because price may not be a good measure of the difficulty levels. On the other hand, since open innovation provides an additional means to measure the difficulty of the games because people can directly browse the posted games, this extra measure will be one of the key questions that we will explore for the rest of the paper.

5. Experimental Procedures

Sixteen subjects are recruited for each of the two sessions. They are mostly undergraduate students at New York University. Students already subscribed to the mailing list receive notice of the experiment. The interested students then sign up for the experiment through an online recruitment system[14] on a first-come-first-served basis. The experimenter has no control in the selection of students in particular disciplines because the mailing list comprises of students across disciplines.

In each period, each subject is allotted three games. For each session, there are ten periods, which are seven minutes long. Each session lasts for two hours. Almost all students finish the instructions in half an hour. A quiz is administered before the experiment begins. The experimenter goes to the carrels to check the answers. If there is a mistake, the experimenter explains to him or her individually. Individual anonymity is maintained

[14] The recruitment is conducted using the open source software eRecruit (http://eRecruit.ust.hk or http://cess.nyu.edu/experiments) developed by the author in 2000 with funding provided by the Center for Experimental Business Research at the Hong Kong University of Science and Technology.

through out the experiment. There is a two-minute trial period before Period 1. In session A, Treatment 2 is run from Period 1 to Period 5. Then a new instruction sheet, which highlights the changes in the new treatment, is delivered to the students before Period 6 begins. Treatment 1 is run from Period 6 to Period 10. The order of the treatment is reversed in Session B.

The average payoffs, which include a show-up fee of $5, are $19.06 and $22.44 for Sessions A and B, respectively.

The outline of the decisions that the subject can make is already discussed in Section 1. Note also that the subjects do not observe the number of games unsolved for all subjects as a whole. The detailed instructions to subjects can be downloaded on the Internet[15].

6. Empirical Analyses

Since there are three allotted games in each of the ten periods for both sessions with sixteen subjects each, there are 960 allotted games in total. Therefore, we have 960 data points on the decisions whether to post a game and set a commission. There are 159 games posted by the subjects.

In Graphs 1 and 2, the area of each circle is proportional to 1 plus the number of clones made from the posted game identified by the pair of commission (y-axis) and posted time (x-axis). The numbers next to some circles to the left and to the right indicate these numbers. It follows that if a circle is of size 1, it has no clones. If everyone clones a game, it will

[15] The instructions to subjects are available at http://www-personal.umich.edu/~bchiao/oi/ which includes the screenshots of the software used.

have size 16. Note that a small size does not necessarily mean that the posted game is not attractive to cloning, it may just mean that it is solved very quickly or that the period is ended. This is usually not the case because the data show that cloning can be very fast before anyone can solves the posted game. However, some very small circles seem to indicate unattractiveness perhaps because of its low commission or the early posting time in which people are concentrating on learning how to solve their own allotted games. Besides different sizes, there are three colors for the circles. Transparent means that a clone of the posted game is solved. It also indicates that the original is solved if the posted game is a clone itself. Grey means that the posted game is not solved. Black means that the posted game is eventually solved by the poster.

Graph 1 refers to data in Session A. In Period 1, subjects seem to be testing the water. The standard deviation for the commission is 2.07, which is higher than all other periods in the same treatment (i.e., up to Period 5). The sizes of the circles are at the small to medium levels, with the largest at 8. In Period 2, the number of circles declines a little, with a reduction of 37.5% compared to the last period. In Periods 3 and 4, the commission clusters around \$0.2. The sizes of circles generally increase, except for games posted early in the period because the games are solved without being cloned very much (perhaps because most people are still learning how to solve their own allotted games). In Period 5, there is a dramatic increase of commission set at zero, which constitutes 61.5% of the total posted games in that period. And they are solved quickly because the sizes of the circles are small no matter how early or late the games are posted. This is an indication of the convergence to the zero-price monotone equilibrium.

The O-ring requirement has not been achieved for the first 5 periods.

Starting Period 6, there is no more O-ring requirement. In Period 6, there is an immediate drop of activities in the public pool. The number of games posted drops 54% compared with Period 5. It is a bit puzzling why there are three out of six games cloned at zero commission. There is rarely any activity in the public pool in Period 7 but the commission at $0.5 seems to have set an example for later periods in which 5 out of 29 (17%) of games posted at this price are solved by others. In Period 8, the zero commission resurfaces for two times but one of them is not cloned at all. In Period 9, quite a lot of posted games (9 out of 13 or 69%) were posted at zero but 56% (5 out of 9) is either not solved or solved by the original poster, indicating a lack of interest for other subjects to work for free. Few cloning at zero price indicates that the zero-price monotone equilibrium is not achieved. This perhaps drives the commission to increase in Period 10, with the median at $0.2.

Table 1 shows the number of games solved:

Table 1: Games Solved in Both Sessions

	Session A	**Session B**
Period	Games Solved	Games Solved
1	37	31
2	39	31
3	44	40
4	42	39
5	43	38
1 to 5 Subtotal	**205**	**179**
6	43	46
7	46	47
8	45	44
9	46	46
10	45	48
6 to 10 Subtotal	**225**	**231**
1 to 10 Total	**430**	**410**

The following tables show more detailed statistics for Session A:

Table 2: All Posted Games in Session A				
Period	Std Error	Mean	Median	Obs
1	2.07	3.50	4	8
2	1.87	4.00	4	5
3	1.57	2.86	2	7
4	1.79	2.20	2	5
5	1.71	1.08	0	13
6	0.82	0.67	0.5	6
7	2.83	3.00	3	2
8	2.38	1.50	0.5	4
9	2.03	1.15	0	13
10	1.91	1.64	2	11

Table 3: Posted Games Solved by Others in Session A				
Period	Std Error	Mean	Median	Obs
1	2.14	3.17	4	6
2	1.87	4.00	4	5
3	1.72	2.83	2	6
4	1.00	1.50	2	4
5	1.53	0.83	0	12
6	0.82	0.67	0.5	6
7	0.00	1.00	1	1
8	2.65	2.00	1	3
9	2.36	1.88	0.5	8
10	1.64	3.20	2	5

Table 4: Posted Games Solved by the Posters in Session A				
Period	Std Error	Mean	Median	Obs
1	0.00	1.00	1	1
2	na	na	na	0
3	0.00	3.00	3	1
4	na	na	na	0
5	na	na	na	0
6	na	na	na	0
7	0.00	5.00	5	1
8	0.00	0.00	0	1
9	0.00	0.00	0	4
10	0.00	0.00	0	2

Table 5: Posted Games Not Solved in Session A				
Period	Std Error	Mean	Median	Obs
1	na	na	na	0
2	na	na	na	0
3	na	na	na	0
4	0.00	5.00	5	1
5	0.00	4.00	4	1
6	na	na	na	0
7	na	na	na	0
8	na	na	na	0
9	0.00	0.00	0	1
10	1.00	0.50	0	4

Graph 2 refers to data in Session B. The pattern is relatively more clear-cut. As in Session A, subjects are testing the water in Period 1. There are quite a lot of posted games that are not solved (4 out of 11 or 36%). There is a lack of activities in the public pool from Periods 2 to 5, with the number of circles range from 2 to 6 and commission ranges from \$0.1 to \$0.5.

When the O-ring requirement is enabled starting Period 6, there is a dramatic increase of activities in the public pool with a total of 57 posted games from Period 6 to Period 10 (compared with 28 from Period 1 to Period 5 and 36 from Period 1 to 5 in Session A). In Period 6, the number of circles is the same as in Period 5 but the number of clones increases. One circle is of size 16 even at zero commission. In Period 7, 65% (or 11 out of 17) of posted games are at zero commission and are solved quickly. The situation is similar for Period 8. In Period 9, only two posted games are not set at zero commission. In the last period, 80% (or 8 out of 10) of the posted games are set at zero commission, which is a strong indication of convergence to the zero-price monotone equilibrium. Finally the subjects are able to achieve the O-ring requirement in Period 10. Though the number of games solved already increased dramatically when the second treatment began in Period 6. This increase could be due to the following factors:

1. People work faster because of the pressure of jeopardizing the group
2. People learned in earlier periods about how to solve games faster
3. Increased use of the public pool

For the first factor, the supporting evidence is that with the absence of the O-ring requirement, the number of games solved in the second half of Session A increases 9.8% from the first half, which is less than the increase for the second half of Session B when the O-ring requirement is present, which is 29.1%. Also, in the presence of weaker learning effect for the first five periods of each session, the number of games solved is higher in Session A (205 games) than in Session B (179 games).

For the second factor, there is clear evidence that the games are solved faster. In Session A the number of games solved within five minutes (two more minutes till the end of the period) in Periods 1 and 10 are 26 and 41. The respective numbers for Session B are 20 and 41.

For the third factor, the use of the public pool indeed increases dramatically. In Session B, the numbers of posted games in the first half and second half of the session are 28 and 57. The respective numbers for Session A are 38 and 36.

The following tables show more detailed statistics for Session B:

Table 6: All Posted Games in Session B				
Period	Std Error	Mean	Median	Obs
1	2.41	3.73	3.00	11
2	1.41	4.00	4.00	2
3	2.11	3.67	3.00	3
4	1.47	2.83	2.50	6
5	1.37	2.67	3.00	6
6	1.70	3.29	3.00	7
7	2.06	1.35	0.00	17
8	2.33	1.88	0.50	16
9	2.70	1.57	0.00	7
10	1.64	0.70	0.00	10

Table 7: Posted Games Solved by Others in Session B				
Period	Std Error	Mean	Median	Obs
1	2.29	3.71	3.00	7
2	1.41	4.00	4.00	2
3	1.15	3.67	3.00	3
4	0.71	4.50	4.50	2
5	1.15	1.67	1.00	3
6	2.65	3.00	4.00	3
7	2.10	1.44	0.00	16
8	2.13	1.60	0.00	15
9	2.04	0.83	0.00	16
10	1.64	0.70	0.00	10

Table 8: Posted Games Solved by the Posters in Session B				
Period	Std Error	Mean	Median	Obs
1	na	na	na	0
2	na	na	na	0
3	na	na	na	0
4	na	na	na	0
5	0.71	3.50	3.50	2
6	0.00	3.00	3.00	2
7	0.00	0.00	0.00	1
8	na	na	na	0
9	na	na	na	0
10	na	na	na	0

Table 9: Posted Games Not Solved in Session B				
Period	Std Error	Mean	Median	Obs
1	2.99	3.75	3.00	4
2	na	na	na	0
3	na	na	na	0
4	0.82	2.00	2.00	4
5	na	na	na	0
6	1.41	4.00	4.00	2
7	na	na	na	0
8	na	na	na	0
9	na	na	na	0
10	na	na	na	0

7. Conclusions

In recent years, a challenge is for managers to decide whether to adopt an open innovation process to develop products. Flagship companies like IBM, and RedHat inject huge amount of resources to open source each year. To provide some insights to these decisions, this paper presents the first experimental evidence collected explicitly in the context of open innovation.

A special feature of open innovation in the real world is that it is at times free. One can download the source code of a program without charge and one can post a program to a website for volunteers to work on. It is then perplexing to economists about how to allocate resources efficiently and whether the supply is transient. Traditionally, price (both pecuniary and non-pecuniary) drives the allocation of resources. However, it is sometimes difficult to measure the difficulty of a task, even less so by a central authority. If pecuniary price were used as a measure, it can only be a rather imperfect signal. This problem is further aggravated when the production function is non-modular, that is, the production is positive only when every worker produces at least at a certain level. This leads to a free-riding problem in which self-interest drives some workers to rely on others to help them free of charge. Our experiment indicates that the commission to helpers converges to zero and everyone solves all posted games. This in turn completely removes a signaling function of price for the difficulty levels. However, our data show that this removal is not sufficient to lead to the catastrophic outcome of zero payoff. This provides a basis for us to hypothesize that open innovation is a key feature to prevent this undesirable outcome to happen because it allows subjects to directly observe the history of work already done and potentially direct more resources to the more difficult tasks.

One implication is that this hypothesis explains why open source software is developed predominately, at least in the 1980's and 1990s, in the operating systems area. This is because early operating systems tend to have a non-modular structure.

If the implications of the paper are valid, managers should identify whether their production processes are modular or not. Or whether their products are considered essential parts of a non-modular system across firms. Such products include virus protection software and firewall[16]. If yes, should the managers free-ride?

Some questions remained though.

What drives the increased use of the open innovation processes? Why more and more open source software is being developed in all areas of software types? Is the modular structure sufficient? If yes, then why do we need time for the convergence of price? Is it because there is a cost in searching for the equilibrium price? Or is it because of a hold-

[16] A famous system intrusion detection software, Tripwire, is open source.

up problem if subjects view the whole session as a repeated game[17]? If no, the innovation process may be further facilitated if there are mechanisms to accelerate the equilibriation of price. In an experimental setting, does it help if the experimenter displays more explicitly the number of clones and the number of games left for a period?

Lastly, it is conceivable that open innovation is conducive to certain strategies that may be hard to imitate in traditional production processes[18] but it is not fully understood or directly attacked in the current study.

References

Andreoni, J., and H. Varian (1999): "Preplay Contracting in the Prisoners' Dilemma," *Proceedings of the National Academy of Sciences*, 96, 10933–10938.

Baldwin, C. Y., and K. B. Clark (2003): "The Architecture of Cooperation: How Code Architecture Mitigates Free Riding in the Open Source Development," Discussion paper, Harvad Business School.

Bestavros, A., and A. Belal (1986): "Mastermind, a Game of Diagnosis Strategies," *Bulletin of the Faculty of Engineering, Alexandria University*.

[17]Note that there is no hold-up problem once the price reaches zero; there is nothing to lose to post games at this price.

[18]In our experiment, there is clear evidence that one particular form of strategy, the rainbow strategy, is increasingly used in later periods. This strategy is a scaled down version of (Pope 1995), which only retains the part of Pope's method to find out the number of colors used without further restricting what one should do afterwards. This strategy is particular easy to notice just by browsing the public pool because of its close resemblance to a rainbow. This is, however, specific to MASTERMIND; we will not infer more than saying that the open innovation process could possibly facilitate the transfer of knowledge for some classes of strategies.

CHIAO, B. H.-F. (2003): "An Economic Theory of Free and Open Source Software: A Tour from Lighthouse to Chinese-Style Socialism," *Proceedings of the International Conference on Open Source*, pp. 61–80.

CHVATAL, V. (1983): "Mastermind," *Combinatorica*, p. 325 to 329.

KNUTH, D. E. (1976–77): "The Computer as Mastermind," *Journal of Recreational Mathematics*, pp. 1–6.

KOOI (2004): "Just Another Mastermind Strategy," mimeo.

KOYAMA, K. (1994): "An Optimal Mastermind Strategy," *Journal of Recreational Mathematics*.

KREMER, M. (1993): "The O-Ring Theory of Economic Development," *The Journal of Economic Development*.

LERNER, J., AND J. TIROLE (2002): "Some Simple Economics of Open Source," *Journal of Industrial Economics*, pp. 197–234.

MERELO, J., J. CARPIO, P. CASTILLO, V. RIVAS, AND G. ROMERO (1999): "Finding a Needle in a Haystack using Hints and Evolutionary Computation: the Case of Genetic Mastermind," http://geneura.ugr.es/~ jmerelo/newGenMM/node3.html.

NEUWIRTH, E. (1982): "Some Strategies for Mastermind," *Zeitschrift fur Operations Research*, pp. 257–278.

POPE, S. (1995): "Basic Mastermind Decryption," http://www.math.niu.edu/~ rusin/uses-math/games/mastermind/usenet.posts.

Strobl, W. (1998): "How to Build a Calculator for Master Minds," http://ntklotz.gmd.de/pic/mm47.

Torvalds, L., and D. Diamond (2001): *Just for Fun–the Story of an Accidental Revolutionary*. HarperCollins Publishers, New York.

von Hippel, E., and G. von Krogh (Forthcoming): "Open Source Software and the Private-Collective Innovation Model: Issues for Organization Science," *Organization Science*.

von Krogh, G., S. Spaeth, and K. R. Lakhani (2003): "Community, Joining, and Specialization in Open Source Software Innovation," *Research Policy*, 32, 1217–1241.

Weber, S. (2004): *The Success of Open Source*. Harvard University Press, Cambridge.

8. Appendix 1–Literature Review

I borrow heavily from (Kooi 2004) and (Merelo, Carpio, Castillo, Rivas, and Romero 1999) to complete the literature review. I thank Kooi for sending me the working paper.

8.1. Stepwise Optimal Strategy

Each guess is optimal in the sense that it is possibly the right answer, that is, it is consistent with all guesses already played. Authors may define the term "consistency" differently. This strategy does not have an expected number of guesses, but it is guaranteed to find the solution in finite time.

A. Exhaustive Search by (Koyama 1994) In a 4 peg, 6 color game, there are $6^4 = 1296$ possible combinations. The strategy is to generate guesses with a computer in sequence and then rule out the combinations that are inconsistent with the guesses already played. For instance, if we start with AAAB, the next moves would be AAAC, AAAD... up to AAAF and then AABA, and so on. As we move along, we will be able to rule out combinations that receive no white or black pegs. The advantage of this strategy is that it runs over the search space only once.

B. Random Search by (Strobl 1998): This approach is similar to (Koyama 1994) except that it generates combinations randomly. Like the exhaustive search, there are combinations that can be ruled out before being played, but it runs over the search space more than once.

8.2. Analytical Optimal Strategy

Combinations that are known to be incorrect are played to reduce the search space. Examples:

8.2.1. (Pope 1995)

In a 4 peg, 4 color game, we can determine what color is used and how many times each color is used by making single-color moves in the first three rows and using a different color each time (e.g. 1st move: AAAA; 2nd move: BBBB; 3rd move: CCCC).

This approach leaves us with five possible scenarios:

- Single-color combination (e.g. AAAA). In this case, we have solved the game by the 4^{th} move.

- 2-color combination with one color used three times and the other used once (e.g. ABBB). This scenario presents $4!/3! = 4$ possibilities. Let's say we learn that the colors are A and B (from the previous three moves). By placing A in a different position each time in the next three moves, we can determine if the combination is ABBB, BABB, BBAB, or BBBA and solve the game by the 7^{th}move.

- 2-color combination with each color used twice (e.g. AABB). There are $4!/2!2! = 6$ possible combinations. Similar to the above scenario, we can use the 4^{th}, 5^{th}, and 6^{th} moves to determine the position of A and solve the game in a maximum of 7 moves.

- 3-color combination (e.g. AABC). In this scenario, we likewise use the next three moves to determine which two slots color A is in. Let's say if color A is in slots 2 and 3, we are left with two possibilities: BAAC or CAAB. We will figure out the positions of B and C in the 7^{th} move and solve the game in 8 moves.

- 4-color combination. This is the most complicated scenario. We can determine which color is in slot 4 by making the 4^{th} move AAAB and, if necessary, the 5^{th} move CCCD.WLOG, let's say color D is in slot 4. Then, in the 6^{th} move we will use the combination, AABD to get either 1, 2, or 3 positional matches. One match means A is in slot 3 and thus leaves us with two possibilities: BCAD or CBAD; Two matches indicates that the combination is either ABCD or BACD; and three matches means it is either ACBD or CABD. In each case, we will be able to identify the positions of the two remaining colors in the 7^{th} move and solve the game in a maximum of 8 moves.

8.2.2. (Knuth 1976–77)

This is the first paper published on solving MASTERMIND. Knuth's approach allows us to identify the combination after four moves. The expected number of moves is 4.478. The strategy is to choose a guess that minimizes the maximum number of remaining possibilities at each stage. If several guesses satisfy this condition, we will used the one that is a "valid pattern" and receives four black pegs.

Table 10 adopts largely from a table in (Kooi 2004). It shows us the number of possible secret combinations that are consistent with the hints for each possible optimal guess. The first column lists the hints which are combinations of black and white pegs. The top row lists five possible choices for an optimal first guess.

Table 10: Uneliminated Secret Combination Candidates Consistent with Hints Given to All Possible First Guesses					
	AAAA	AAAB	AABB	AABC	ABCD
(0,0)	625	256	256	81	16
(0,1)	0	308	256	276	152
(0,2)	0	61	96	222	312
(0,3)	0	0	16	44	136
(0,4)	0	0	1	2	9
(1,0)	500	317	256	182	108
(1,1)	0	156	208	230	252
(1,2)	0	27	36	84	132
(1,3)	0	0	0	4	8
(2,0)	150	123	114	105	96
(2,1)	0	24	32	40	48
(2,2)	0	3	4	5	6
(3,0)	20	20	20	20	20
(4,0)	1	1	1	1	1

The underlined cells in Table 10 represent the worst scenario for all possible types of first guesses. According to Knuth, AABB should be played first since it minimizes the worst scenario.

8.2.3. Others

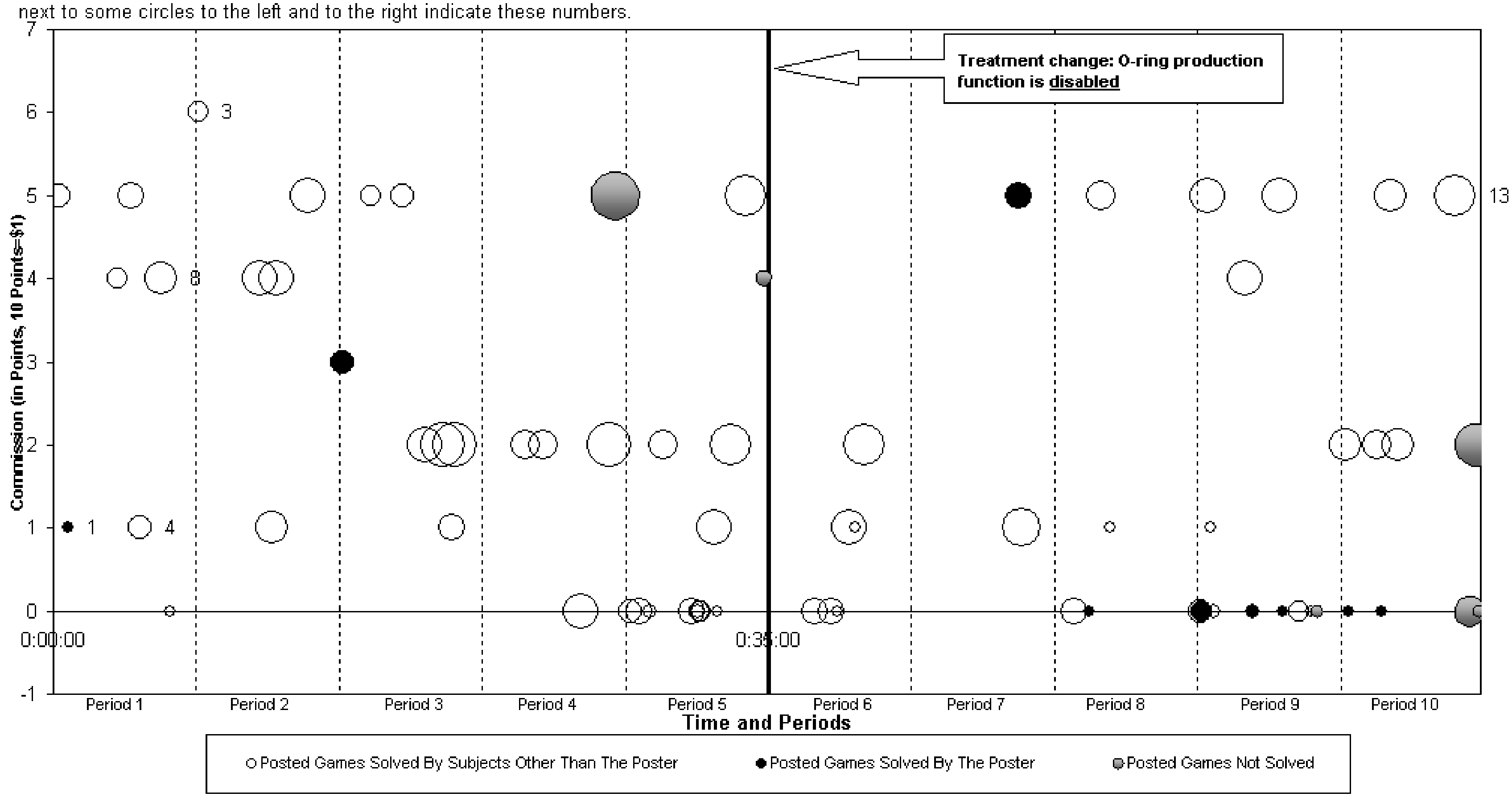
In (Bestavros and Belal 1986), the strategy is based on information theory. The technique is to obtain as much information as possible on the secret combination with each chosen guess, be it on the average or in the worst case. With this algorithm, the secret combination can be found in 3.9 ± 0.5 or 3.8 ± 0.6 average combinations in a 4-peg and 6-color game. This strategy is an improvement on Knuth's approach.

In addition, one can also minimize the number of parts described in (Kooi 2004). Or one can minimize the entropy as in (Neuwirth 1982).

9. Appendix 2–Graphs

Graph 1 Commission, Solved And Unsolved Clones Of Posted Games Across Periods in SESSION A

Note: The area of the circle is proportional to **1 plus** the number of clones made from the posted game identified by the pair of commission and posted time. The numbers next to some circles to the left and to the right indicate these numbers.



Graph 2 Commission, Solved And Unsolved Clones Of Posted Games Across Periods in SESSION B

Note: The area of the circle is proportional to **1 plus** the number of clones made from the posted game identified by the pair of commission and posted time. The numbers next to some circles to the left and to the right indicate these numbers.