A Dynamic Model of Mixed Duopolistic Competition: Open Source vs. Proprietary Innovation*

Suat Akbulut† Murat Yılmaz‡

August 2015

Abstract

Open source software development has been an interesting investment model of production and innovation in recent years, especially in the last few decades. Unlike the private investment model, open source innovators freely share the proprietary software that they have developed at their private expense. Also, open source development is usually subject to certain licenses, one of which being the General Public License (GPL), the most popular open source license. In this thesis, we study the competition dynamics between a proprietary firm and an open source firm, Windows and Linux, for instance. We model the the competition between such two firms by incorporating the nature of the GPL, investment opportunities by the proprietary firm, user-developers who can invest in the open source development, and a ladder type technology. We use a two period dynamic mixed duopoly model, in which a profit- maximizing proprietary firm competes with a rival, the open source firm, which prices the product at zero, with the quality levels determining

*Marie Curie Career Integration Grant, FP7PEOPLE-2011CIG is acknowledged.

†Boğaziçi University, Department of Economics, Bebek, Istanbul, 34342.

‡Boğaziçi University, Department of Economics, Bebek, Istanbul, 34342. Email: *muraty@boun.edu.tr.* Phone: +90 212 3597656. Fax: +90 212 287 2453. Web site: *http://www.econ.boun.edu.tr/yilmaz/*

their relative positions over time. We ask the following questions. How does the existence of open source firm affect the investment and the pricing behavior of the proprietary firm? Does the social welfare increase with the existence of the open source development? How are the users/consumers affected from the open source firm being available? Is it possible that an open source firm, which is behind the proprietary firm in the technology ladder, catches up with the proprietary firm? We also discuss the limitations of our model and possible extensions.

1 Introduction

A software is called open source, if its source code is open in the sense that anyone has free access to it. Open Source movement aims to bring programmers not concerned with proprietary ownership or any financial gain together to produce a more useful and bug-free product for everyone to use. By revealing its source code, an open source can be refined by many independent developers all around the world. The source code of an open source product is made available free of charge to the public. So, developers on the Internet read, redistribute and modify the source code, forcing an advantageous evolution of it.

Although there are many licenses used to distribute an open source projects, is GNU General Public License (GPL)the most commonly used one as of late 2014, by a share above 51%[1]. GPL has two main features. The first feature that GPL has is that although every user has the right to use and modify the the code freely, the modifications must be distributed under the terms of the same license, if they are to be distributed at all. That is to say, GPL is a copy-lefted license. Although the rationale behind the open source movement is that larger group of programmers who are not interested in the ownership produces a better, faster, more useful and, bug-free software, the second aspect of GPL allows the commercial exploitation of the program. Hence, the users have to sustain the free access to the source code, yet, as long as they maintain the free access, they are allowed to make profit. For example, according to its 2015 income statement, Red Hat, the world's largest commercial distributor of the Linux operating system, made a total net income of $ 180.20 million in 2014.

Open source software (OSS), particularly Linux, has gained significant impact on software industry, thereby it has attracted noticeable interest of the researchers, as well. Having observed the provision of OSS and its ongoing developments are costly, and moreover it is almost always publicly available at a price of zero,which does not reflect the

[1]See https://www.blackducksoftware.com/resources/data/top-20-open-source-licenses/

economic costs of user-developers incur to enhance it, economists have tried to understand the motives that might encourage the user-developers to involve in such a costly activity.

The success of open source software has led a literature on it, which has been flourishing since early 2000s. Lerner and Tirole (2002, 2005) introduce a broad sense discussion on economics of open source development. They indicate two reasons that might lead the developers to contribute to open source evolution. First reason that might make developers involve in this costly activity is that they receive a direct benefit in the form of improved software. Secondly, they get an indirect benefit by signaling their abilities in the job market. They also point out that the literature mostly considers individual incentives to adopt open source software.

Contributing to open source innovations brings the public good nature onto the surface. A considerable amount of literature focuses on open source development as public good in a static approach. Johnson (2002) uses public good approach in a static environment, where private provisions of user-developers to a public good -the open source- diminishes as the number of user-developers increases because of free riding problem, and presents some comparative statistics and welfare results. Modica (2012) takes a two period oligopoly game using a circular city approach in order to model the open source innovations from a public good perspective.

Some of the open source literature focuses on the competition between proprietary firm and open source firm. Casadesus-Masanell and Ghemawat (2006) study the competition between proprietary firm and open source firm in a dynamic mixed duopolostic industry with the demand side learning, and show that it is better to have the proprietary firm as a monopoly when the total welfare is considered. Casadesus-Masanell and Llanes (2011) use a mixed duopoly structure, where a for-profit proprietary firm competes with an open source firm, which tries to maximize the value of its open software. Our model differs from these studies in the way that it combines the open source innovation and the competition between proprietary firm and open source firm in a dynamic environment.

This study tries to examine the effects of the existence of an open source firm that is competing with the proprietary firm on the proprietary firm's investment in innovation and production behavior, and how it affects the total welfare in the market. We set up a dynamic model with two periods, the first of which has two stages: competition and investment. In the second period, being the last period of the model, there is no investment stage. In the two competition stages, proprietary firm and open source firm compete in a mixed duopolistic industry, where the former charges a price to maximize its overall expected profit, whereas, the latter is freely available. At the beginning of each period, a new cohort of potential users enter into the model. At the beginning of the competition stage, they observe the quality levels and the price of proprietary firm's product, and they decide which operating system to use during their life time of one period. As long as they have some valuation for the open source, everyone will have an operating system, at least the free open source. In the next stage of the first period, the investment stage, while proprietary firm invests in probability to increase its products quality level, user-developers' incentives for involving this costly development activity is to signal their abilities.

We find that under some circumstances, the proprietary firm supplies less and invests more in the presence of the open source rival, which leads the proprietary firm to make less profit in the duopolistic industry compared to its monopoly, suggesting that a duopoly is likely to dominate the proprietary firm's monopoly in terms of total welfare generation. However, this is not always true, i.e it might be better for the total welfare when there is only proprietary firm in the market.

2 The Model

In order to model such a competition, where the competitors have heterogeneous objectives, we will make use of the literature on "mixed duopolies". Throughout this study, w and ℓ will stand for Windows and Linux, respectively.

There are two periods, the first of which has two stages: competition and investment. There is only one stage in the second period, which is competition stage. At the beginning of period t, for $t = 1, 2$, the quality level of an operating system (OS) $s \in \{w, \ell\}$, is denoted as $k_t^s \in \mathbb{Z}_+$. Although the initial quality levels k_1^w and k_1^ℓ will be given, their levels at the beginning of the second period, k_2^w and k_2^ℓ, is determined endogenously by the investment decisions of both Windows and user-developers -Linux users. The evolution of quality levels follow a ladder type technology. For this reason, in second stage of the first period, investment stage, Windows invests in probability in order to climb one step on the technology ladder, while, user developers involve in this costly development activity to signal their abilities to the job market. The realizations of the developments occur at the end of the period. Hence, if Windows invests $i_w \in [0, 1]$, its quality level at the beginning of the second period will be:

$$k_2^w = \begin{cases} k_1^w + 1 & \text{with prob. } i_w \\ k_1^w & \text{with the remaining prob. } (1 - i_w) \end{cases}$$

During the same stage, those user-developers, who decided to get a free copy of Linux in the previous stage, simultaneously with Windows' investment decision, decide whether and how much to invest in probability to get the exogenous bonus $b \in (0, 1)$. Let i_j be the user-developer j's investment level. Assuming the open source does not have the modular nature[2], if at least one user developer succeeds in development stage, because of the terms

[2]If a software has modular nature, then it is possible to break up the large projects, such as developing it, into small modules. Improving the modules independently will accomplish the project. However, it

of GPL, Linux will move up one step in the technology ladder. Hence, then Linux' quality level at the beginning of the second period will be:

$$k_2^\ell = \begin{cases} k_1^\ell + 1 & \text{, if at least one user-developer succeeds} \\ k_1^\ell & \text{, if no user-developers succeed} \end{cases}$$

However, investment is a costly activity for both Windows and user-developers. In order to invest i, they must incur a cost of $\frac{1}{2}i^2$. Here we assume that users-developers' skills are homogeneous. Yet, in Section 6.3, we introduce heterogeneity among user-developers in their development skills.

To specify the demand side, in each period, a new cohort of N potential users enter into the market. They observe the quality levels of both proprietary software and open source. Let k_t denote the quality differences between Windows and Linux, thus $k_t = k_t^w - k_t^\ell$. Let $\alpha_s(k_t) > 0$ denote the OS s's value given by the cohort entering at time t. $\alpha_s(k_t)$ is called as OS s's technological trajectory, which is a function of the quality level of OS s, k_t^s, and the quality level of the competing OS, k_t^{-s}. Even though the initial levels of these technological trajectories are exogenously given in the model, how they evolve while moving from period 1 to period 2 is endogenous. In the beginning of the first stage of period 1, and period 2, Windows charges a price of P_t, where $t \in \{1, 2\}$, to attract new customers. Let q_t be the number of users in period t, who buy Windows, then $N - q_t$ is the number of user-developers in the same period since Linux is freely available and $\alpha_\ell(\cdot)$ is positive.

Assumption 1 *We assume linear demand function. In period τ, for tau $= 1, 2$:*

the software does not have the modular nature, then its development cannot be divided into pieces, and consequently, there may not be sufficient sophisticated user-developers who can customize the software to their own needs. Section 6.3 tries to deal with the question when the open source software has the modular nature by introducing a contribution dimension to the model. You can see Lerner and Tirole (2002) for details about modules.

Windows value to a user $q_\tau \in \{1, 2, \ldots, N\}$ *is:*

$$\alpha_w(k_\tau)\frac{N - q_\tau}{N} \tag{1}$$

and, similarly, let the value of Linux be:

$$\alpha_\ell(k_\tau)\frac{N - q_\tau}{N}$$

Figure 1 illustrates the Assumption 1. The demand is drawn as a straight line as if N was a real number, but the readers are well aware of that it must be a set of N points on that line instead.

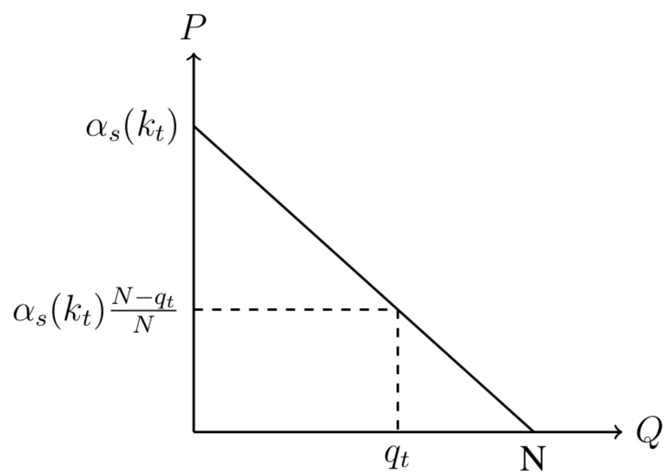


Figure 1: Demand of operating system s by cohort τ

Assumption 2 $\alpha_s(k_t) \geq 0$. *Since we assume OS s' value is never negative, technological trajectories are to be bounded below, which is required to have well defined demand functions.*

Assumption 3 *$\alpha_w(k_t)$ is increasing in k_t, whereas, as k_t increases, $\alpha_\ell(k_t)$ decreases. Holding the competent OS' quality level constant, $\alpha_j(\cdot)$, where $j \neq i$, the value of OS i, $\alpha_i(\cdot)$, will increase as its quality level increases.*

Let β be defined as $\beta(k_t) = \alpha_w(k_t) - \alpha_\ell(k_t)$, representing the difference between the trajectories of w and ℓ.

Assumption 4 *$\beta(k_t)$ is assumed to be concave in k_t.*

Assumption 4 is needed to be assumed to have a well defined maximization problem for Windows. It ensures that the difference between the technological trajectories of Windows and Linux would not explode. Otherwise, after some level of the quality difference, Windows would become a monopoly-like firm in the market.

3 Monopoly

In a market, where there is no substitute for Windows, and every user of any cohort has positive willingness to pay, inverse demand function is directly obtained by Equation 1. Figure 2 illustrates the timing of the events in a monopoly industry.

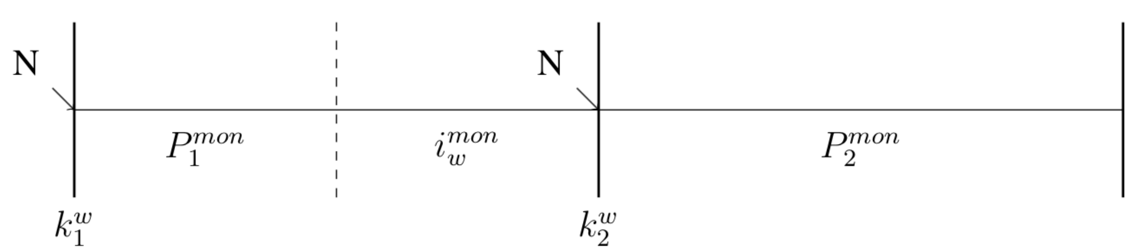


Figure 2: Timing of events in monopoly benchmark

To solve the equilibrium of monopolistic structure, we employ the Backward Induction methodology in this two-period game.

3.1 Second Period

Since Windows is the only operating system producer and this is last period of the game, having monopoly power, it produces the profit maximizing amount of $N/2$, and sets its price to $\alpha_w(k_2^w)/2$ in accordance with the demand structure. As a result, it generates a profit of:

$$\pi_2^{mon}(k_2^w) = \frac{N}{4}\alpha_w(k_2^w) \tag{2}$$

3.2 Second Stage of The First Period

How much investment is optimal for Windows? Investment affects Windows' profit through the quality level, which determines the valuations of the users, with trajectory function. An investment level i_w will increase its quality level by 1 with the probability i_w. Having known it will generate a profit of $\pi_2^{mon}(k_2^w)$ in the next period by Equation 2, the optimal monopoly investment strategy for Windows require it to choose an investment level, i_w^{mon} be in the following set:

$$i_w^{mon} \in \underset{i_w}{\operatorname{argmax}} \left\{ i_w \frac{N}{4}\alpha_w(k_1^w + 1) + (1 - i_w)\,\frac{N}{4}\alpha_w(k_1^w) \right\}$$

Since the above term is linear in i_w, the monopoly investment level will be:

$$i_w^{mon} = \begin{cases} \frac{N}{4}\left[\alpha_w(k_1^w + 1) - \alpha_w(k_1^w)\right] & \text{, if } \alpha_w(k_1^w + 1) - \alpha_w(k_1^w) \leq \frac{4}{N} \\ 1 & \text{, otherwise} \end{cases} \tag{3}$$

3.3 First Stage of The First Period

At the beginning of the game, knowing its optimal strategies for the second phase and the next period, Windows chooses a price level, P_1^{mon} (or, equivalently, quantity level, q_1^{mon}) that maximizes its following overall expected profit:

$$\max_{q_1}\left\{\alpha_w(k_1^w)\frac{(N-q_1)\,q_1}{N}-\frac{1}{2}\,(i_w)^2+i_w\frac{N}{4}\alpha_w(k_1^w+1)\,(1-i_w)\,\frac{N}{4}\alpha_w(k_1^w)\right\}$$

When we take the first order derivative with respect to q_1, we obtain the following optimality condition since second order condition holds:

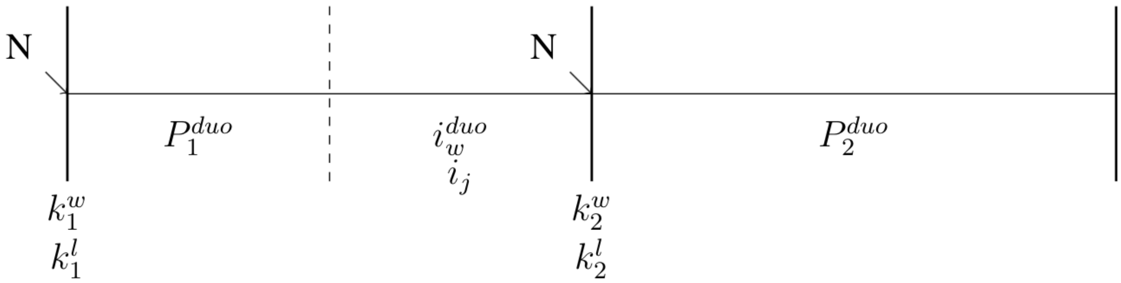
$$0=\frac{\alpha_w(k_1^w)}{N}\,(N-2q_1)\quad\Rightarrow\quad q_1^{mon}=\frac{N}{2}\qquad\text{and}\qquad P_1^{mon}=\frac{\alpha_w(k_1^w)}{2}$$

Hence, Windows produces is used by one half of the number of users in each period, and the users should pay a price that is equal to the one half of the maximum value given to itself by their cohort.

4 Duopoly

When we introduce Linux into the market, Windows no longer has its monopoly power. It has to consider the presence of Linux and the user-developers' investment decisions while deciding how much to produce in each period, and to invest in the investment stage. The timing of events in duopoly industry is described in Figure 3.

Figure 3: Timing of events in duopoly industry



To solve duopoly equilibrium, we use the same methodology, backward induction technique, as we utilize in the monopoly case. Since Linux can be downloaded freely and $\alpha_\ell(\cdot) > 0$, i.e. all users are willing to pay something (even only a small amount) for a product that they can get without paying anything, it is guaranteed that every user will get one operating system, at least Linux. Hence, if q_τ is the number of users who buy Windows, then the remaining users of cohort τ, $N - q_\tau$, obtain the Linux at no price, and they become user-developers.

4.1 Second Period

When Windows is sold at price P_2^{duo} at period 2, the indifferent user between Windows and Linux, q_2, is found by the following equation:

$$\alpha_w(k_2)\frac{N - q_2^{duo}}{N} - P_2^{duo} = \alpha_\ell(k_2)\frac{N - q_2^{duo}}{N}$$

Remembering $\beta(k_t) = \alpha_w(k_t) - \alpha_\ell(k_t)$, the inverse demand function for Windows in period 2 would be the following

$$P_2^{duo} = \beta(k_2)\frac{N - q_2^{duo}}{N} \tag{4}$$

As a rational profit maximizer agent, Windows produces q_2^{duo} such that

$$q_2^{duo} \in \underset{q_2}{\operatorname{argmax}} \left\{ \left(\beta(k_2) \frac{N - q_2}{N} \right) \cdot q_2 \right\}$$

Taking the first order derivative with respect to q_2, we get quantity and price levels for the second period as follows:

$$0 = \frac{\beta(k_2)}{N} (N - 2q_2) \quad \Rightarrow \quad q_2^{duo} = \frac{N}{2} \qquad \text{and} \qquad P_2^{duo} = \frac{\beta(k_2)}{2}$$

As a result, Windows will make a profit of

$$\pi_2^{duo} = P_2^{duo} \cdot q_2^{duo} = \frac{N}{4} \beta(k_2)$$

in the second period.

4.2 Second Stage of The First Period

In the investment stage, the actions of the user-developers have impact on Windows' objectives. However, Windows' investment decision does not affect the user-developers' investment strategies since they invest only for the purpose of getting the bonus, b. There would be some other version of the model, in which Windows' and user-developers' decisions affect both their investment strategies. However, Section 6 will be discussing the issues that such models could create and how these issues could be handled. For now, we will analyze the investment decisions of user-developers and Windows separately.

4.2.1 User-Developers' Investment Decisions

We have already mentioned that user-developers are only interested in the bonus, b, when deciding whether and how much to invest to Linux' development. Hence, a user-developer

j, for $j \in \{q_1^{duo}, (q_1^{duo}+1), \ldots, N\}$, chooses and investment level i_j, which maximizes her expected net benefit. Hence, i_j solves the following maximization problem

$$\max_{i_j \in [0,1]} \left\{ i_j \cdot b - \frac{1}{2} {i_j}^2 \right\}$$

In order to find the optimal solution for the problem above, we take the first order derivative of it with respect to i_j. Hence,

$$b - i_j = 0 \quad \Rightarrow \quad i_j^* = b \qquad \text{for} \quad j = q_1^{duo}, (q_1^{duo}+1), \ldots, N$$

Since her expected net benefit, $\frac{b^2}{2}$, is positive, in the equilibrium, she will choose to invest $i_j^* = b$. Due to the symmetry, $i_j^* = b$ for all $j \in \{q_1 + 1, \ldots, N\}$. So, Linux will be developed with probability $1 - (1-b)^{N - q_1^{duo}}$, which is the probability that at least one user-developer succeeds.

4.2.2 Windows' Investment Decision

In contrast to investment decision of user-developers, Windows takes into account what the user-developers strategy is. Thus, it chooses the investment level i_w^{duo} so that i_w^{duo} maximizes its expected future profit $\pi_2^{duo}(k_2)$.

$$\max_{i_w} \left\{ \begin{array}{c} i_w \left((1-b)^{(N-q_1)} \cdot \beta(k_1+1) + \left(1 - (1-b)^{(N-q_1)}\right) \cdot \beta(k_1) \right) \\ +(1-i_w) \left((1-b)^{(N-q_1)} \cdot \beta(k_1) + \left(1 - (1-b)^{(N-q_1)}\right) \cdot \beta(k_1 - 1) \right) - \frac{1}{2} {i_w}^2 \end{array} \right\}$$

By taking the first order derivative with respect to i_w, we obtain

$$i_w^{duo} = \frac{N}{4} \left[(1-b)^{N - q_1^{duo}} \cdot (\beta(k_1+1) + \beta(k_1 - 1) - 2\beta(k_1)) + (\beta(k_1) - \beta(k_1 - 1)) \right]$$

or in a more proper way

$$i_w^{duo} = \min\left(\frac{N}{4}\left[(1-b)^{N-q_1^{duo}} \cdot C + \Delta\right], 1\right)$$

where $C = \beta(k_1+1) + \beta(k_1-1) - 2\beta(k_1)$ and $\Delta = \beta(k_1) - \beta(k_1-1)$.

4.3 First Stage of The First Period

In the competition stage of the first period, in order for the user q_1 to be indifferent between Windows and Linux, her net benefit from buying Windows and downloading a free copy of Linux must be equal. In the equilibrium, choosing to get a free copy of Linux ensures a user to get an expected benefit of $\frac{b^2}{2}$ in the investment stage. So, when Windows is sold at price P_1^{duo} at period 2, the indifferent user between Windows and Linux, q_1, is found by the following equation:

$$\alpha_w(k_1)\frac{N-q_1}{N} - P_1 = \alpha_\ell(k_1)\frac{N-q_1}{N} + \frac{b^2}{2}$$

Hence, the inverse demand for Windows in period 1 would look like the following

$$P_1^{duo} = \beta(k_1)\frac{N-q_1^{duo}}{N} - \frac{b^2}{2} \tag{5}$$

The optimal pricing/quantity strategy for Windows must be a solution of the following maximization problem, which simply is Windows' overall expected profit when it chooses to produce q_1.

$$\max_{q_1}\left\{\begin{array}{rcl} P_1 \cdot q_1 - \frac{1}{2}(i_w)^2 & + & i_w \cdot (1-b)^{(N-q_1)} \cdot \frac{N}{4} \cdot \beta(k_1+1) \\ & + & i_w \cdot \left(1-(1-b)^{(N-q_1)}\right) \cdot \frac{N}{4} \cdot \beta(k_1) \\ & + & (1-i_w) \cdot (1-b)^{(N-q_1)} \cdot \frac{N}{4} \cdot \beta(k_1) \\ & + & (1-i_w) \cdot \left(1-(1-b)^{(N-q_1)}\right) \cdot \frac{N}{4} \cdot \beta(k_1-1) \end{array}\right\}$$

The first order condition of above maximization problem is

$$-\beta(k_1)\frac{2q_1^{duo}}{N} + \beta(k_1) - \frac{b^2}{2} - \frac{N^2}{16}C^2\left((1-b)^{N-q_1^{duo}}\right)^2 \ln(1-b) - \frac{N}{4}\Delta(1-b)^{N-q_1^{duo}} \ln(1-b)\left(1+\frac{N}{4}C\right)$$

Although this condition does not have a closed form analytical solution in q_1^{duo}, we are capable of comparing it with the first period quantity in the monopoly case, q_1^{mon} since we know that the overall expected profit function is concave in q_1, and maximized at q_1^{duo}. When we evaluate the above first order condition at $q_1^{mon} = N/2$, we obtain the following function:

$$f(b) = -\frac{b^2}{2} - \frac{N^2C^2}{16}(1-b)^N \ln(1-b) - \frac{N}{4}\Delta(1-b)^{N/2}\ln(1-b)\left(1+\frac{N}{4}C\right)$$

Proposition 1 *For large enough bonus b, proprietary firm produces less in the first period of the duopolistic competition as opposed to the case in which it is a monopoly.*

Proof. $f(b)$ is continuous in $[0,1)$. $f(0) = 0$, and $\lim_{b\to 1} f(1-b) < 0$. Therefore, $\exists\, \hat{b} \in [0,1)$ such that $f(b) < 0$, $\forall\, b \in [\hat{b}, 1)$. Because the first order condition is negative at point $q_1 = N/2$ for large b's and, the overall expected profit function is concave in q_1,

$$q_1^{duo} < q_1^{mon} = \frac{N}{2} \tag{6}$$

■

Proposition 1 shows that the existence of an open source rival affects the monopoly production as existence of any other rivalry for-profit firm, in the sense that the monopoly firm decreases its production level.

Proposition 2 *Proprietary firm makes more investment in the duopoly industry competition as opposed to the case where it is a monopoly.*

Proof. Let $(1-b)^{(N-q_1)} = x$. Observe that $x \in (0,1)$. Since $\beta(\cdot)$ is concave, $C = \beta(k_1+1)+\beta(k_1-1)-2\beta(k_1)$ is negative, and $\Delta = \beta(k_1)-\beta(k_1-1)$ is positive. Hence,

$$\begin{aligned} & \frac{N}{4}\cdot(x-1)\cdot(\beta(k_1+1)+\beta(k_1-1)-2\beta(k_1)) \;>\; 0 \\ \Rightarrow\quad & \frac{N}{4}\,(x\cdot C+\Delta-(\beta(k_1+1)-\beta(k_1))\;)\;>\;0 \\ \Rightarrow\quad & \frac{N}{4}\,(x\cdot C+\Delta)\;>\;\frac{N}{4}\,(\beta(k_1+1)-\beta(k_1)) \\ \Rightarrow\quad & i_w^{duo}\;>\;i_w^{mon} \end{aligned}$$

■

As Proposition 2 suggests, competition results in Windows to increase its investment level. However, this result is not special to having a open source rival.

5 Welfare Comparison

Proposition 1 & 2 concludes that the proprietary firm makes less profit in the duopoly industry, which suggests that a duopoly is likely to dominate proprietary firm's monopoly in terms of total welfare generation. In this section, we analyze the welfare implications of the two industry structure that we studied above. Instead of finding the absolute level of total welfare in the duopoly industry, we will compare the total welfare levels under the assumptions that $\alpha_w(\cdot)$ and $\alpha_\ell(\cdot)$ are linear with slope γ_w and γ_ℓ, respectively. $\alpha_w(\cdot)$

and $\alpha_\ell(\cdot)$ being linear with slope γ_w and γ_ℓ causes $\beta(\cdot)$ to be a linear function, as well, with slope $\gamma_w - \gamma_\ell$, that is, $C = 0$ and $\Delta = \gamma_w - \gamma_\ell$. Assumption 3 ensures that $\gamma_w > 0$ and $\gamma_\ell < 0$. Thereby, Δ is positive.

Proposition 3 *If $\frac{N}{4}\left(|\gamma_\ell| + 2\gamma_w\right) < 1$ and $\frac{N}{4}\gamma_w > (1-b)^{N/2}$, then total welfare is higher in proprietary firm's monopoly than the total welfare in duopoly industry.*

Proof. To prove Proposition 3, we divide the total welfare into pieces and compare them piece-wise instead of measuring them as wholes. And, when comparing the two welfare levels, we interpret the absence of Linux in the monopoly industry as k_t^ℓ and $\alpha_\ell(k_t)$ being zero. Therefore, $\beta(k_t) = \alpha_w(k_t)$.

Expected total welfare in the Windows' monopoly, W^m is

$$\mathrm{W}^m = \sum_{j=1}^{N/2}\left(\alpha_w(k_1)\frac{N-j}{N}\right) - \frac{\left(i_w^{mon}\right)^2}{2} + i_w^{mon}\sum_{j=1}^{N/2}\left(\alpha_w(k_1+1)\frac{N-j}{N}\right) + \left(1-i_w^{mon}\right)\sum_{j=1}^{N/2}\left(\alpha_w(k_1)\frac{N-j}{N}\right)$$

$$= \underbrace{\overbrace{\alpha_w(k_1)\left(\frac{3N-2}{8}\right)}^{a^{mon}} - \overbrace{\frac{\left(i_w^{mon}\right)^2}{2}}^{c^{mon}}}_{\text{first period welfare}} + \underbrace{\overbrace{i_w^{mon}\alpha_w(k_1+1)\left(\frac{3N-2}{8}\right) + \left(1-i_w^{mon}\right)\alpha_w(k_1)\left(\frac{3N-2}{8}\right)}^{d^{mon}}}_{\text{second period welfare}}$$

Expected welfare in the first period of the duopoly industry, W^d:

$$\mathrm{W}_1^d = \sum_{j=1}^{q_1}\left(\alpha_w(k_1)\frac{N-j}{N}\right) + \sum_{j=q_1+1}^{N}\left(\alpha_\ell(k_1)\frac{N-j}{N}\right) - \frac{\left(i_w^{duo}\right)^2}{2}$$

$$= \overbrace{\beta(k_1)\left(q_1 - \frac{1}{N}\frac{q_1(q_1+1)}{2}\right)}^{a^{duo}} + \overbrace{\alpha_\ell(k_1)\frac{N-1}{2}}^{e} - \overbrace{\frac{\left(i_w^{duo}\right)^2}{2}}^{c^{duo}}$$

And the expected total welfare generated in the second period of the duopoly will be:

$$
\mathrm{W}_2^d = \begin{array}{llll}
i_w \cdot (1-b)^{(N-q_1)} & \cdot & \left(\beta(k_1+1)\left(\frac{3N-2}{8}\right)\right. & \left.-\alpha_\ell(k_1+1)\frac{N+1}{2}\right) \\
+i_w \cdot \left(1-(1-b)^{(N-q_1)}\right) & \cdot & \left(\beta(k_1)\left(\frac{3N-2}{8}\right)\right. & \left.-\alpha_\ell(k_1)\frac{N+1}{2}\right) \\
+(1-i_w) \cdot (1-b)^{(N-q_1)} & \cdot & \left(\beta(k_1)\left(\frac{3N-2}{8}\right)\right. & \left.-\alpha_\ell(k_1)\frac{N+1}{2}\right) \\
+(1-i_w) \cdot \left(1-(1-b)^{(N-q_1)}\right) & \cdot & \left(\beta(k_1-1)\left(\frac{3N-2}{8}\right)\right. & \left.-\alpha_\ell(k_1-1)\frac{N+1}{2}\right)
\end{array}
$$

Hence,

$$
\mathrm{W}_2^d = \overbrace{\frac{3N-2}{8}\left[\left(i_w^{duo} + (1-b)^{(N-q_1)}\right)(\gamma_w - \gamma_\ell) + \beta(k_1-1)\right]}^{d^{duo}} \\
\underbrace{-\frac{N+1}{2}\left[\left(i_w^{duo} + (1-b)^{(N-q_1)}\right)\gamma_\ell + \alpha_\ell(k_1-1)\right]}_{f}
$$

Now let us start to compare the pieces marked by lower case letters. For $q_1 < N$,

$$
\frac{d\left(q_1 - \frac{1}{N}\frac{q_1(q_1+1)}{2}\right)}{dq_1} > 0
$$

Thus,

$$
\beta(k_1)\left(q_1^{duo} - \frac{1}{N}\frac{q_1^{duo}(q_1^{duo}+1)}{2}\right) < \beta(k_1)\left(\frac{N}{2} - \frac{1}{N}\frac{\frac{N}{2}(\frac{N}{2}+1)}{2}\right) = \beta(k_1)\frac{3N-2}{8}
$$

which implies

$$
a^{duo} < a^{mon} \tag{7}
$$

As a consequence of Proposition 2, we have $i_w^{duo} > i_w^{mon}$ which implies $-\frac{1}{2}\left(i_w^{duo}\right)^2 < -\frac{1}{2}\left(i_w^{mon}\right)^2$, which in turn implies

$$
c^{duo} < c^{mon} \tag{8}
$$

Since $i_w^{mon} > (1-b)^{(N-q_1)}$ and, i_w^{duo} cannot be more than 1,

$$i_w^{duo} + (1-b)^{(N-q_1)} - 1 < i_w^{mon}$$
$$\left(i_w^{duo} + (1-b)^{(N-q_1)} - 1\right)(\beta(k_1) - \beta(k_1-1)) < i_w^{mon}(\beta(k_1) - \beta(k_1-1))$$
$$\frac{3N-2}{8}\left[\left(i_w^{duo} + (1-b)^{(N-q_1)}\right)(\gamma_w - \gamma_\ell) + \beta(k_1-1)\right] < \frac{3N-2}{8}\left[i_w^{mon}(\beta(k_1) - \beta(k_1-1)) + \beta(k_1)\right]$$

which implies

$$d^{duo} < d^{mon} \tag{9}$$

Combining e and f we get:

$$\begin{aligned} e + f &< \frac{N-1}{2}\left[\alpha_\ell(k_1) - \left(i_w^{duo} + (1-b)^{(N-q_1)}\right)\gamma_\ell - \alpha_\ell(k_1-1)\right] \\ &= \frac{N-1}{2}\left(1 - i_w^{duo} - (1-b)^{(N-q_1)}\right)\gamma_\ell \end{aligned}$$

Note that $\left(1 - i_w^{duo} - (1-b)^{(N-q_1)}\right)$ is positive due to the assumptions $\frac{N}{4}(|\gamma_\ell| + 2\gamma_w) < 1$ and $\frac{N}{4}\gamma_w > (1-b)^{N/2}$. To see this, note

$$\begin{aligned} &\frac{N}{4}(|\gamma_\ell| + 2\gamma_w) < 1 \\ \Rightarrow\quad &\frac{N}{4}|\gamma_\ell| + \frac{N}{4}\gamma_w + \frac{N}{4}\gamma_w < 1 \\ \Rightarrow\quad &\frac{N}{4}\gamma_\beta + \frac{N}{4}\gamma_w < 1 \\ \Rightarrow\quad &\frac{N}{4}\gamma_\beta + (1-b)^{N/2} < 1 \\ \Rightarrow\quad &\frac{N}{4}\gamma_\beta + (1-b)^{N-q_1^{duo}} \end{aligned}$$

Thus, the summations of the terms including α_ℓ in the duopoly welfare is negative, that is,

$$e + f < 0 \tag{10}$$

Combining Equations (7), (8), (9) & (10), we conclude that the total welfare that the monopoly proprietary generates is higher than the total welfare in the duopoly industry. ■

Proposition 3 shows that the competition does not necessarily increase the welfare in an oligopoly industry when compared to the monopoly market. This is because the presence of a rival induces the proprietary firm to set lower prices and those users who do not buy the proprietary firm's product are not left empty handed; they can get the open source freely, which increases the total surplus. However, the decrease in proprietary firm's and its users' surpluses do not, always, need to be compensated by the increase in user-developers' surpluses.

6 Discussion

This paper is about the alternative models that could have been used to capture the effects of an open source firm's presence on the behavior of a proprietary firm. We also tried to examine changes in our results when the question is modeled in different ways, and summarized the reasons behind the fact that why we end up with not using them. As our future work, we will improve the last alternative in order to increase the period number to finitely or even infinitely many because it might be useful to have infinitely many periods in order to study long-run behavior of the two firms and to question the lifespan of proprietary firm, whose faith might be releasing its source codes and becoming an open source, as well.

6.1 $T \geq 3$ Periods

When we tried to set up a model, where the number of periods is three or more, or infinitely many, we end up with technical problems of solving the first order condition of proprietary firm's maximization problem. This problem occurs because there is no analytical solution, for sure, to the number of proprietary users at period t, q_t, when the number of potential users, N, exceeds three. Employing the known methods to solve the Bellman Equation that captures the recursive nature of the dynamic game problem is not helpful since transition matrix that should govern the evolution of the state variables are determined by the choice variables in each period, i.e the transition matrix is not stationary.

6.2 Endogenous Bonus with OLG

One other possibility could be utilize the first investment incentive for the user-developers that Lerner and Tirole (2002) mention, i.e user-developers involve in the development activity because they receive a direct benefit in the form of improved software. We set up a model, where users lived two periods. They could buy an operating system only when they are young. User-developers could develop the open source when they are young, and enjoy the appreciation of its quality when they are old, if at least one of them succeeds due to General Public License. When we model the user-developers investment incentives in this framework, with allowing the investment levels to be in $[0, 1]$ interval, we faced difficulties while solving the optimal investment levels of user-developers since the optimal decisions include N^{th} order equations. To overcome such difficulties, one could think of forcing the possible investment level choices of the user developers to be binary, i.e they would be either 0 or 1. However, there occurs a free rider problem that Johnson (2002) finds, too. Since it is guaranteed for the open source to be improved when one user-developer chooses to invest in 1, it is optimal for every user-developer to let someone else do it.

6.3 Contribution Game with Infinitely Many Users

When Lerner and Tirole (2002) explain the favorable characteristics for an open source production, they mention about its modularity, whether the overall project is divided into smaller and well-defined tasks (modules) that individuals can handle independently from other modules. Sufficiently modular nature of an open source software, whose different portions can be improved by independent user-developers, might turn the investment stage to a contribution game for open source user-developers. To do so, one other helpful way could be having infinitely many users distributed on $[0, 1]$. Although in our original model, that would create some compatibility problems while finding the open source firm's development probability, since it has a multiplication part, which is not a good way to use when there are infinitely many users, that would provide a well defined demand, and is a better way to model the investment stage as a contribution game, where the probability of open source firms' development is affected by a fraction of the measure of user-developers that contribute or all users. Such a model might also capture the direct benefit incentives of the user-developers, which would result in having different optimal investment strategies for different user-developers. To incorporate the direct benefit, a successful development of a user-developer could be rewarded by enjoying the appreciation of her own operating system before the quality increase become public.

7 Conclusion

It is impressive that a costly investment based upon not having the property rights has produced such a useful and reliable software. In this study, a simple two-period model of open source innovation has been presented to understand the difference of the behavior of the proprietary firm's production, pricing and investment strategies and to facilitate welfare comparisons between the presence of it and the traditional, profit driven method of development, where the quality levels of the two follow a ladder type technology framework.

It has been shown that the proprietary firm decreases its production level when there is an open source rival, and in order to better compete with the open source firm, it invests more. However, that the proprietary firm loosing some of its profit cannot be concluded as a duopoly is likely to dominate the proprietary firm's monopoly in terms of total welfare generation because it has been shown that for some levels of the linear formed technological trajectory functions' slopes, the total welfare is higher in proprietary firm's monopoly than the total welfare in duopoly industry.

REFERENCES

Admati, A. R., & Perry, M. (1991). Joint projects without commitment. *The Review of Economic Studies*, *58*, 259-276.

Aoki, R. (1991). R&D competition for product innovation: An endless race. *The American Economic Review*, *81*, 252-256.

Atal, V., & Shankar, K. (2014). Open source software: Competition with a public good. *Atlantic Economic Journal*, *42*(3), 333-345.

Bitzer, J. (2004). Commercial versus open source software: The role of product heterogeneity in competition. *Economic Systems*, *28*, 369-381.

Casadesus-Masanell, R., & Ghemawat, P. (2006). Dynamic mixed duopoly: A model motivated by Linux vs. Windows. *Management Science*, *52*(7), 1072-1084.

Casadesus-Masanell, R., & Llanes, G. (2011). Mixed source. *Management Science*, *57*(7), 1212-1230.

Caulkins, J. P., Feichtinger, G., Grass, D., Hartl, R. F., Kort, P. M., & Seidl, A. (2013). When to make proprietary software open source. *Journal of Economic Dynamics & Control*, *37*, 1182-1194.

Economides, N., & Katsamakas, E. (2006). Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. *Management Science*, *52*(7), 1057-1071.

Erkal, N., & Minehart, D. (2014). Optimal technology sharing strategies in dynamic games of R&D. *Journal of Economics & Management Strategy*, *23*(1), 149-177.

Fershtman, C., & Markovich, S. (2010). Patents, imitation and licensing in an asymmetric dynamic R&D race. *International Journal of Industrial Organization*, *28*(2), 113-126.

Grossman, G. M., & Shapiro, C. (1987). Dynamic R&D competition. *The Economic Journal*, *97*, 372-387.

Haruvy, E., Sethi, S. P., & Zhou, J. (2008). Open source development with a commercial complementary product or service. *Production and Operations Management*, *17*(1), 29-43.

Hasnas, I., Lambertini, L., & Palestini, A. (2014). Open innovation in a dynamic cournot duopoly. *Economic Modelling*, *36*, 79-87.

Jaisingh, J, See-To, E. W. K., & Tam K. Y. (2008). The impact of open source software on the strategic choices of firms developing proprietary software. *Journal of Management Information Systems*, *25*(3), 241-276.

Johnson, J. P. (2002). Open source software: Private provision of a public good. *Journal of Economics & Management Strategy*, *11*(4), 637-662.

Kort, P. M., & Zaccour, G. (2011). When should a firm open its source code: A strategic analysis. *Production and Operations Management*, *20*(6), 877-888.

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. *Journal of Industrial Economics*, *50*(2), 197-234.

Li, C., & Ji, C. (2010). Innovation, licensing, and price vs. quantity competition. *Economic Modelling*, *27*, 746-754.

Llanes, G., & de Elejalde, R. (2013). Industry equilibrium with open-source and proprietary firms. *International Journal of Industrial Organization*, *31*, 36-49.

Marx, L. E., & Matthews, S. A. (2000). Dynamic voluntary contribution to a public project. *The Review of Economic Studies*, *67*, 327-358.

Modica, S. (2012). Open source without free-riding. *Economia Politica*, *2*, 247-260.

Mustonen, M. (2005). When does a firm support substitute open source programming?. *Journal of Economics & Management Strategy*, *14*(1), 121-139.

Reisinger, M., Ressner, L., Schmidtke, R., & Thomes, T. P. (2014). Crowding-in of complementary contributions to public goods: Firm investment into open source software. *Journal of Economic Behavior & Organization*, *106*, 78-94.

Suh, J., & Yılmaz, M. (2015). *Economics of open source: A dynamic approach.* Manuscript in preparation.

Yıldırım, H. (2006). Getting the ball rolling: Voluntary contributions to a large-scale public project. *Journal of Public Economic Theory*, *8*(4), 503-528.