ICIS 2006 Proceedings

International Conference on Information Systems (ICIS)

December 2006

Pricing Open Source Software

Byungcho Kim
Carnegie Mellon University

Pei-yu Chen
Carnegie Mellon University

Tridas Mukhopadhyay
Carnegie Mellon University

Recommended Citation

Kim, Byungcho; Chen, Pei-yu; and Mukhopadhyay, Tridas, "Pricing Open Source Software" (2006). *ICIS 2006 Proceedings*. 24.
http://aisel.aisnet.org/icis2006/24

PRICING OPEN SOURCE SOFTWARE

Economics and Information Systems

Byung Cho Kim
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA
bckim@andrew.cmu.edu

Pei-yu Chen
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA
pychen@andrew.cmu.edu

Tridas Mukhopadhyay
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA
tridas@andrew.cmu.edu

Abstract

In this paper, we examine the issue of pricing open source software. We compare three different pricing mechanisms: commercial software, open source software dual-licensing, and open source software support. We investigate whether the open source software pricing models are viable under monopoly and duopoly when an open source software vendor competes with a commercial software vendor. Our model considers the motivation for and the barrier to open source software adoption, which provides a better picture of the open source software market. We identify the factors that affect the optimal pricing strategy of the commercial and the open source software vendors. Our results can give pricing guidelines to the open source software vendors in the case of monopoly and duopoly, which is not clear in the current state.

Keywords: Open source software, commercial software, pricing, competition

Introduction

Open Source Software (OSS) is software for which the source code is available to the public, enabling anyone to copy, modify, and redistribute the source code (Varian and Shapiro 2003). Commercial software, by contrast, is software that is distributed under commercial license agreements, usually for a fee. Open source was a service mark of the Open Source Initiative (OSI), a non-profit organization that continues to provide an official "Open Source Definition". According to OSI definition, open source software is software whose source code can be freely modified and redistributed[1]. The redistribution rights do not preclude a company selling such software for profit.

[1] The Open Source Definition. http://opensource.org/docs/definition.php

Recently open source software is getting more attention from not only software developers but also users, including governments and large enterprises. According to the Economist (2003), many countries in European Union and Asia are funding open source software initiatives outright. For example, Japan announced that it would collaborate with China and South Korea to develop open source alternatives to Microsoft's software, and it has already allocated nine million US dollars to the project. According to BBC, the Brazilian government announced that it would adopt open source software. As a result, Brazil's government ministries and state-run enterprises are abandoning Windows in favor of open source software, like Linux (Kingstone 2005). Open source adoption is rising in the industry sector as well. A recent survey by Forrester Research shows that 46 percent of the 140 North American firms were using open source software and 14 percent had a plan to use it in 2004. In 2005, open source usage was even higher. Among 128 respondents, 56% were using open source software and 19% were planning (Gloude 2005). Especially, Linux is already adopted by 74% of the respondents and is considered for adoption by 22%. Other widely adopted open source software includes Apache, MySQL, and JBoss.

Why are firms using open source software? The Free/Libre and Open Source Software (FLOSS) Survey 2002 found that enterprises are adopting open source software on the grounds of higher stability, cost savings, and flexibility due to the open and modifiable source code (Wichmann 2002). Independence from pricing and licensing policies of big software companies and better availability of IT specialists are also considered to be motivations for open source software usage. Another survey by the International Data Corporation (IDC) shows that firms are using open source software because of its quality and flexibility, rather than merely considering it "good enough" because it is inexpensive (Broersma 2005). Such firms consider the customizability of open source software to be important, since they believe that vendors of pre-packaged, proprietary software routinely downplay the customizability of open source software, arguing customers are not interested in extending software functions themselves. On the other hand, a user survey by Forrester Research finds that the top barriers to open source adoption by firms include the lack of support and the availability of applications (Reid 2004). Companies are also concerned about not having skilled workers to support or use open source applications, licensing costs, and security. Another survey by Actuate, conducted in 2006, reflects opinions from 141 respondents in the financial services sector. It shows that the majority of respondents cited the availability of long-term support (58.2%) as the main barrier to adopting open source software and the availability of long-term maintenance was a close second-place barrier (44.7%). The benefits include cost savings (55%), vendor independence (49.3%), and flexibility (47.1%) (McCarthy 2006).

Lately a new movement in the open source software industry has been observed. As some open source software products become popular and their market shares approach critical mass, the vendors of such software seek to capitalize on the publicity and popularity of their software products. Two approaches have been successful in commercializing open source software: the dual-licensing model and the support model. With the dual-licensing model, a software vendor offers the very same software under two different licenses. Open source license allows the licensees to modify, distribute, and use the software for free, but it requires the release of any modifications under the same open source license. The commercial license permits using and developing the software under standard commercial terms. MySQL and Openoffice are the examples of the dual-licensing model. A second approach is the support model, which allows users to get the software for free, but the software vendor charges users for support. Although the viability of this model is still questioned, RedHat and JBoss have been successful commercializing their products by adopting the support model.

There is a lively debate going on among industry experts about the viability and the scalability of the commercial model of the open source software (Moczar 2005; Vaughan-Nichols 2005). Recently the issue of open source software is receiving substantial attention from academic researchers. Raghunathan et al. (2005) examine the quality debate in open source software by setting up an analytical model. They show that open source software quality is not necessarily lower than commercial software quality. Casadesus-Masanell and Ghemawat (2006) analyze a dynamic mixed duopoly in which a for-profit proprietary software vendor interacts with an open source software vendor that prices at zero in the presence of demand-side learning effects. Economides and Katsamakas (2006) analyze the optimal two-sided pricing strategy of a platform firm and compare industry structures based on a proprietary platform such as Microsoft Windows with those based on an open source platform such as Linux. All the papers that examine the issue of open source software from an economic perspective assume that the open source software vendor prices at zero and makes no profit. From a legal perspective, some research has been done on this for-profit open source software (Gomulkiewicz 2004; Välimäki 2003). The issue of pricing information goods has also been extensively studied (Choudhary et al. 2005; Dewan and Mendelson 1990; Hitt and Chen 2005; Mendelson and Whang 1990; Png and Chen 2003; Sundararajan 2004; Westland 1992). To our best knowledge, no research has examined the issue of open source software pricing from an economic perspective. In this paper, we examine the optimal pricing strategies of both the commercial and the open source software vendors and find the conditions under which a for-profit open source model is viable.

We only consider the businesses as the customers in this paper, since the commercial open source software models do not target individual software users. Our model captures two sources of customer heterogeneity in their valuation of software: taste and tech-savviness. Customers value the same software differently. For example, a company in an industry using information technology heavily may have higher valuation of the software than other firms, since a significant portion of their business may depend on information systems powered by the software. Taste parameter captures this heterogeneity in customer taste. We consider two types of customers in terms of their tech-savviness. Along with cost savings, benefits from customizability and flexibility are considered as the top reasons for open source adoption. When an in-house IT management team with skilled workers is available, the firm can enjoy such benefits, since it has the capability of customizing and managing the open source software. When a firm does not have such capability of managing technology, open source software incurs significant support cost, which is cited as the main barrier to open source adoption. We consider the firms with a strong in-house IT management team to be high-type customers and the others to be low-type customers. Reflecting reality, we assume that high-type customers enjoy benefit from flexibility and customizability whereas low-type customers suffer from support cost when they use open source software.

We compare three different pricing mechanisms: commercial software, open source dual-licensing, and open source support. We first examine the optimal pricing strategies for each pricing scheme under a monopoly setting. We find that the monopoly price for open source software under dual-licensing is higher than the commercial software price, hence higher profit, if the total flexibility benefit for the high-type customers is higher than total support cost for the low-type customers. Under the support model, the open source vendor charges for support at higher price than the commercial software price, but it makes lower profit than the commercial software vendor. Then we investigate whether the open source software vendor prefers a certain pricing mechanism in the monopoly market. Our result shows that the dual-licensing model is more profitable when the difference between flexibility benefit and support cost exceeds a certain threshold. Otherwise, the open source vendor prefers the support model. The optimal choice between the dual-licensing model and the support model also depends on other factors such as marginal cost for support and proportion of the high-type and low-type customers. Our paper can give guidelines to the open source software vendors that may have difficulty in making pricing decisions in the current environment.

We extend our model to a duopoly competition setting. The first scenario models competition between a commercial software vendor and an open source software vendor with a dual-licensing model. We analyze the optimal pricing strategies of each software vendor. We find the conditions under which an equilibrium exists. At equilibrium, the commercial and the open source software vendors split the market. The open source software vendor with a dual-licensing model serves high-type customers only, charging higher price than the commercial software vendor that covers the low-type segment. We find that the open source software vendor charges more than the commercial software when flexibility benefit is high. When the open source software does not provide much flexibility benefit, the commercial software vendor sells its software at higher price than the open source software.

Our result proves the viability of the dual-licensing model of the open source software in the presence of competition with commercial software. Next, we study the duopoly software market when there is competition between a commercial software vendor and an open source software vendor with a support model. The profitability of the support model has been questioned since all the customers can get the software for free and the vendor can sell its support only to the customers in need, who are low-type customers in our model. We find that the commercial software vendor competing against the open source support model charges price equal to the open source vendor's marginal cost for support. With this pricing strategy of the commercial software vendor, the support model brings zero profit to the open source software vendor.

We also investigate whether the open source support model is viable in other settings. By setting up a two-period model, we examine the role of switching cost and study the optimal pricing strategies of both the commercial and the open source software vendors. The result is consistent with the case with no switching cost in the sense that the commercial software vendor sets up its price to prevent positive profit for the open source vendor. When the switching cost is large, the commercial vendor charges a low price in the first period and increases its price in the second so that it can have a large installed base in the first period and lock in the customers in the second. However, with small switching cost, the commercial vendor charges higher price in the first period than in the second since increasing its installed base is not very effective in raising switching cost to the customers. Finally, we examine the case of quality asymmetry. Interestingly, we find that the open source support model is viable in the presence of quality asymmetry no matter whether the quality of the open source software is higher or lower than the commercial software quality. The case of the open source software with higher quality is reasonable. The opposite case is interesting in the sense that the open source software with lower quality is profitable while the open source software with the same quality as the commercial software is not. This can be explained as market expansion for the commercial software. When both vendors offer the same quality software, the high-type customers have no incentive to buy the commercial software, since the open source software is free and they do not need any support. However, when the commercial software is superior to the open source software, some high-type customers may buy the commercial software, implying that the commercial software vendor can target such high-type customers as well.

The rest of the paper is organized as follows. We present our model in Section 2 and examine each of the three pricing mechanisms under monopoly in Section 3. In Section 4, we extend our model to a duopoly competition between a commercial software vendor and an open source software vendor with each of the open source pricing models. We investigate whether the open source support model is viable in the presence of other factors such as switching cost and quality asymmetry in Section 5. Finally, Section 6 concludes the paper.

Model

We analyze the optimal pricing decisions of an open source software vendor and investigate whether the emerging pricing schemes for open source software are viable in both monopoly and duopoly cases where the open source software vendor competes against a commercial software vendor. We consider two models of open source software pricing: the dual-licensing model and the support model. The dual-licensing model provides the same open source software under two different licenses: open source license and commercial license. Users who want to donate their source code to the Open Source community can license their software under the General Public License (GPL), which is the most popular free software license. Under the open source license, the licensees can freely modify, distribute, and use the software at no charge. On the other hand, any users who want to use the open source software for profit seeking purpose must purchase a commercial license. MySQL and Openoffice are following the dual-licensing model. Under the support model, users can get the software for free, but the software vendor charges the users for support. RedHat and JBoss are examples of successful open source support model.

Customers

Customers in our model are firms, not individual software users. Potential customers divide into two segments. A proportion μ of customers are technically savvy, and we call them high-type customers. These customers have a capable in-house IT management team. We assume that the high-type customers can take advantage of the flexibility of the open source software by customizing the source code and that their in-house IT management team can handle maintenance so that they do not need any further support from a vendor or a third party. The high-type customers may adopt the open source software not only because open source software has cost advantage but also it gives flexibility. Customers in the remaining fraction $(1 - \mu)$ are low-type customers that may incur support cost if they adopt the open source software. The low-type customers' incentive for open source adoption may be cost savings only.

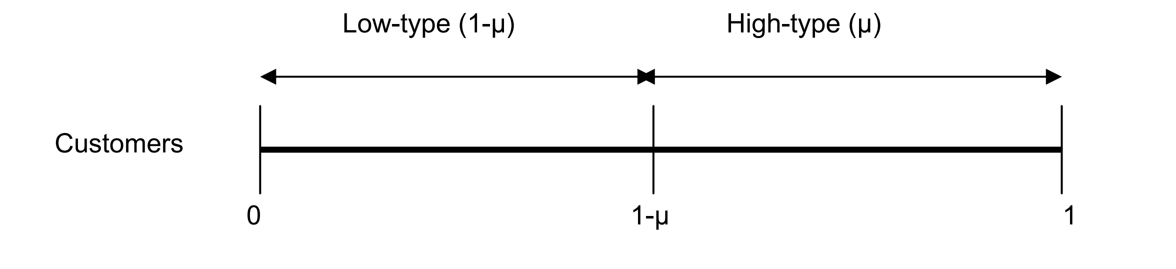


Figure 1. Fractions of Customers in Software Market

No matter which segment a customer is in, the customer does not enjoy any flexibility benefit or suffer from support cost when it uses commercial software. Within each segment, potential customers are characterized by their value, v, from the software. We assume that v is uniformly distributed on [0, 1], which leads to a linear demand curve. We also assume risk neutrality of the customers.

Software Pricing Schemes

We examine a commercial software model and two open source pricing schemes: the dual-licensing model and the support model. In this section, we present our model and derive customer utility and vendor profit under each of the three software pricing mechanisms.

Commercial Software

When a customer buys the commercial software, her benefit would be the same no matter which type the customer is. In other words, commercial software incurs neither flexibility benefit nor support cost to the customer. In reality, the commercial software may incur some level of flexibility benefit and/or support cost, which is not significant compared to the case of the open source software. In the open source pricing models, we consider the relative benefit and cost from the open source software. *C* in the subscript means commercial software. Let p_C be the price of the commercial software. The customer's utility would be

$$u_C = v - p_C.$$

Let q_C be the demand for the commercial software. Then the profit for the commercial software vendor would be

$$\pi_C = q_C p_C.$$

Open Source Software: Dual-Licensing Model

We label the high-type customers *H* and the low-type customers *L*. The proportion of the high-type customers is μ and the remaining proportion $(1 - \mu)$ is the low-type segment. *D* in the subscript represents the dual-licensing model of open source software. When adopting open source software, a high-type customer enjoys the flexibility benefit whereas a low-type customer suffers from the support cost. Let f be the flexibility benefit and s be the support cost. p_D denotes the price of the open source software with a commercial license. The net benefits for a high-type customer and a low-type customer are as follows:

$$\begin{aligned} u_{DH} &= v - p_D + f \\ u_{DL} &= v - p_D - s. \end{aligned}$$

Let q_{DH} and q_{DL} be the demand for the open source software under dual-licensing pricing scheme from the high-type and the low-type segments, respectively. The profit for the open source software vendor adopting the dual-licensing model would be

$$\pi_D = \{\mu\, q_{DH} + (1-\mu)\, q_{DL}\}\, p_D.$$

Open Source Software: Support Model

Let *S* in the subscript represent the support model for open source software. Under this pricing scheme, the open source software vendor provides the software at no charge. The vendor makes profit by selling its support service to the users in need. Let p_S be the price for the support. The net benefits for a high-type and a low-type customer are

$$\begin{aligned} u_{SH} &= v + f \\ u_{SL} &= v - p_S \quad \text{given} \quad p_S \le s. \end{aligned}$$

Note that p_S is bounded by the support cost s. If p_S exceeds s, no customer would buy support service. Since the high-type customers use the software for free and would never ask for any support, the vendor makes profit from the low-type customers only. Let q_S be the demand for the support. We assume that support and service incur marginal cost, c_S. The profit for the open source software vendor adopting the support model would be

$$\pi_S = (1-\mu)\, q_S\, (p_S - c_S)\,.$$

Table 1 summarizes utility that a customer enjoys from software under each of the three mechanisms.

Table 1. Utility from Software

	High-Type	Low-Type
Commercial	$v - p_C$	$v - p_C$
OSS: Dual-Licensing	$v - p_D + f$	$v - p_D - s$
OSS: Support	$v + f$	$v - p_S$

Monopoly

In this section, we examine the optimal pricing strategies for each pricing scheme under monopoly. We compare prices and profits of the three pricing models: commercial, open source dual-licensing, and open source support. We investigate whether each mechanism is viable when the software vendor monopolizes the market. Since a business model that is not viable under monopoly is not likely to survive competition, examining the monopoly case is relevant, although monopoly may not be realistic for most for-profit open source software.

Price

Commercial Software

Customers who get positive utility would buy the software. The demand for the commercial software would be

$$q_C = \int_{p_C}^{1} dv = 1 - p_C\,.$$

The profit for the commercial software vendor is

$$\pi_C = q_C\, p = (1 - p_C)\, p_C\,.$$

The first-order condition for π_C is

$$\frac{\partial \pi_C}{\partial p_C} = 1 - 2p_C = 0,$$

resulting in the profit-maximizing price as

$$p_C^* = \frac{1}{2}.$$

Open Source Software: Dual-Licensing Model

The demand for the open source software from the high-type and the low-type segments are

$$q_{DH} = \int_{p_D - f}^{1} dv = 1 - p_D + f$$

$$q_{DL} = \int_{p_D + s}^{1} dv = 1 - p_D - s.$$

The open source software vendor adopting a dual-licensing gets the following profit:

$$\begin{aligned}\pi_D &= \{\mu q_{DH} + (1-\mu) q_{DL}\} p_D \\ &= \{\mu(1-p_D+f) + (1-\mu)(1-p_D-s)\} p_D.\end{aligned}$$

The first-order condition for π_D is

$$\frac{\partial \pi_D}{\partial p_D} = \mu f - (1-\mu)s + 1 - 2p_D = 0.$$

Thus, the optimal price under the dual-licensing model is

$$p_D^* = \frac{\mu f - (1-\mu)s + 1}{2}.$$

Open Source Software: Support Model

Under the support model, the open source software vendor makes profit out of low-type customers only. The demand for the support model would be

$$q_S = (1-\mu)\int_{p_S}^{1} dv = (1-\mu)(1-p_S).$$

The profit for the open source software is then

$$\pi_S = q_S\,(p_S - c_S) = (1-\mu)(1-p_S)(p_S - c_S)$$

where c_S is the marginal cost for support. The first-order condition is

$$\frac{\partial \pi_S}{\partial p_S} = (1-\mu)(1-2p_S+c_S) = 0.$$

Thus, the optimal price is

$$p_S^* = \frac{1+c_S}{2}.$$

Profit

In this section, we compare the profits of the three different software pricing models and investigate how much profit the open source software pricing scheme brings compared to commercial software. We examine the viability of the open source software pricing models in a monopolized software market. Note that the commercial software vendor makes the following profit at the optimal price:

$$\pi_C^* = (1-p_C^*)p_C^* = \frac{1}{4}.$$

The profit for the open source software vendor with a dual licensing model at the optimal price would be

$$\begin{aligned}\pi_D^* &= \{\mu(1-p_D^*+f) + (1-\mu)(1-p_D^*-s)\} p_D^* \\ &= \{\mu f - (1-\mu)s + 1 - p_D^*\} p_D^* \\ &= \frac{1}{4}\{\mu f - (1-\mu)s + 1\}^2.\end{aligned}$$

Finally, with a support model, the open source software vendor makes its profit as follows:

$$\begin{aligned}\pi_S^* &= (1-\mu)(1-p_S^*)(p_S^*-c_S)\\ &= (1-\mu)\left(1-\frac{1+c_S}{2}\right)\left(\frac{1+c_S}{2}-c_S\right)\\ &= \frac{1}{4}(1-\mu)(1-c_S)^2.\end{aligned}$$

Comparison of the optimal prices leads to the following Proposition.

Proposition 1: *Under monopoly, the open source software with a dual-licensing model charges higher price than the commercial software vendor* $(p_D^* > p_C^*)$ *and makes higher profit* $(\pi_D^* > \pi_C^*)$ *if total flexibility benefit for the high-type customers is higher than total support cost for the low-type customers* $(\mu f > (1-\mu)s)$. *Otherwise* $(\mu f < (1-\mu)s)$ *, the commercial software is more expensive* $(p_D^* < p_C^*)$ *and more profitable* $(\pi_D^* < \pi_C^*)$ *than the open source software with a dual-licensing model. With a support model, the open source software vendor provides support at higher price than commercial software price* $(p_S^* > p_C^*)$ *with making less profit* $(\pi_S^* < \pi_C^*)$.

Proof. See the Appendix.

When an open source software vendor with a dual-licensing model monopolizes the market, the open source software price depends on flexibility benefit, support cost and proportion of high-type customers. If the total flexibility benefit for the high-type customers exceeds the total support cost for the low-type customers, open source software can be more expensive than commercial software. Otherwise, open source software charges a price lower than commercial software. Interestingly, with a support model, an open source vendor sells its support at higher price than the commercial software price. Since the profitable customers for the open source support model is only the low-type segment, the open source vendor's optimal strategy is charging high price for support. This may contradict the results from the survey on motivation for open source adoption, which cites cost advantage on the top of the list. In reality, monopoly may not be the case for most for-profit open source software. Proposition 1 shows the optimal pricing strategies of the open source software as a benchmark. We also find that both pricing models for open source software are viable when the software vendor monopolizes the market. Interestingly, an open source vendor can make higher profit than a commercial software vendor under monopoly by adopting a dual-licensing model. However, with a support model, the profit for the open source is always lower than the profit from commercial software. Then which pricing scheme is better for the open source software vendor that monopolizes the market? Dual-licensing model or support model? We next examine the strategic incentive of the open source software vendor for each pricing scheme and find the conditions under which one mechanism is better than the other.

Proposition 2: *Under monopoly, a dual-licensing model is more profitable than a support model* $(\pi_D^* > \pi_S^*)$ *when* $\mu f - (1-\mu)s > \sqrt{1-\mu}(1-c_S) - 1$. *The open source software vendor makes more profit with a support model than with a dual licensing model* $(\pi_D^* < \pi_S^*)$ *when* $\mu f - (1-\mu)s < \sqrt{1-\mu}(1-c_S) - 1$.

Proof.

Note that

$$\pi_D^* - \pi_S^* = \frac{1}{4}\{\mu f - (1-\mu)s + 1\}^2 - \frac{1}{4}(1-\mu)(1-c_S)^2.$$

Since $\mu f - (1-\mu)s + 1 > 0$ and $c_S \le 1$,

$$\pi_D^* > \pi_S^* \quad \text{if} \quad \mu f - (1-\mu)s > \sqrt{1-\mu}(1-c_S) - 1$$

$$\pi_D^* < \pi_S^* \quad \text{if} \quad \mu f - (1-\mu)s < \sqrt{1-\mu}(1-c_S) - 1.$$

QED.

Proposition 2 shows the conditions under which the monopolist open source vendor makes a choice between the dual-licensing model and the support model. The result implies that the dual-licensing model is better than the support model if the total net benefit from open source, which is the difference between the total flexibility benefit and total support cost, is greater than a certain threshold. If the net benefit does not reach the threshold, the support model is more profitable than the dual-licensing model.

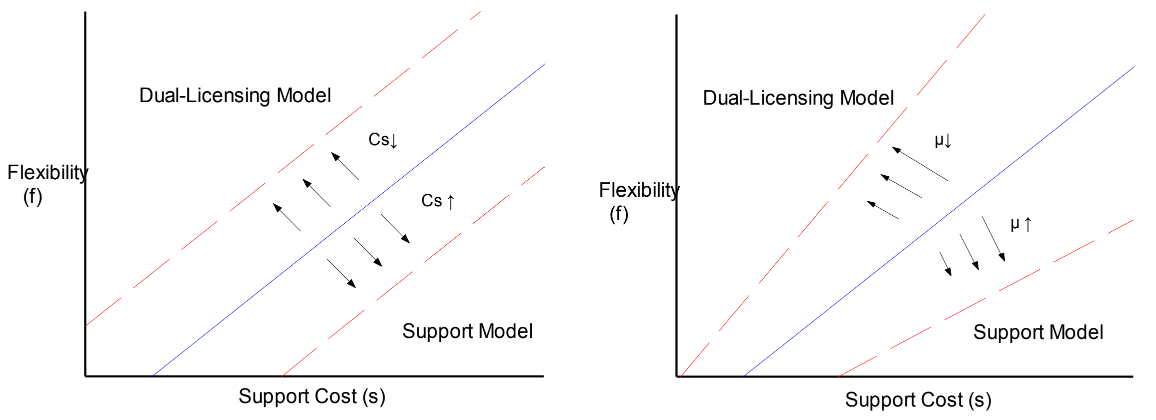


Figure 2. Open Source Pricing under Monopoly

Figure 2 illustrates the role of the marginal cost for support and the proportion of the high-type customers to the framework. It is shown that the dual-licensing model is preferred in the upper-left region while the support model is chosen in the lower-right region. As the marginal cost for support increases, the region for the dual-licensing model becomes wider. This is reasonable in the sense that the support model is less viable when the marginal cost for support is higher. The proportion of the high-type customers tilts the line while the marginal cost shifts it. As the high-type segment becomes bigger, the dual-licensing model becomes more preferable. This makes sense because the high-type customers are profitable only to the vendor with the dual-licensing model. Thus, an increase in the proportion of the high-type customers makes the dual-licensing model more attractive to the open source software vendor.

Competition

In the real world, there are successful examples of for-profit open source software competing with commercial software. With a dual-licensing model, MySQL competes against Oracle, IBM DB2, and Microsoft SQL server. RedHat Enterprise Linux, which is a commercial version with support and service, competes against Microsoft Windows. In this section, we study the optimal pricing strategies of the commercial and the open source software vendors in more realistic cases. We extend our model to a duopoly competition setting. First, we examine the competition between a commercial software vendor and an open source software vendor with a dual-licensing model. Then we study the case where an open source software vendor with a support model competes against a commercial software vendor.

Open Source Software with Dual-Licensing versus Commercial Software

Recall that a customer enjoys the following utility by adopting commercial software:

$$u_C = v - p_C .$$

The open source software vendor with a dual-licensing model provides the following benefits to a high-type customer and a low-type customer, respectively.

$$u_{DH} = v - p_D + f$$
$$u_{DL} = v - p_D - s.$$

A further analysis of the customer benefits leads to Proposition 3.

Proposition 3: *In a duopoly market with a commercial software vendor and an open source software vendor with a dual-licensing model, the commercial software vendor serves low-type customers while the open source software vendor serves high-type customers. An equilibrium exists in the presence of sufficient flexibility benefit and support cost* ($f \geq \sqrt{\mu}$ *and* $s \geq \frac{1}{2}$)*. When flexibility benefit is high and support cost is low* ($f \geq \sqrt{\mu}$ *and* $0 < s < \frac{1}{2}$)*, the condition for an equilibrium to exist is as follows:*

$$\mu(1+f)^2 \geq (2\mu f + 2\mu s + 1)(1-2s).$$

The optimal prices are

$$p_C^* = \frac{1}{2} \quad and \quad p_D^* = \frac{1+f}{2}.$$

Proof. See the Appendix.

We find that a commercial software vendor and an open source software vendor with a dual-licensing model target different segments of the customers. When they compete against each other, a commercial software vendor serves the low-type customers while an open source software vendor serves the high-type customers. This is consistent with the literature on product discrimination and versioning in the sense that the high-type customers choose high-value products, open source software in this case, since it provides flexibility benefit in addition to the original value of the product to the high-type customers. Our findings support viability of the dual-licensing model for open source software.

Open Source Software with Support versus Commercial Software

We examine a duopoly software market where an open source software vendor with a support model competes with a commercial software vendor. We investigate whether the support model is viable in this market and examine the impact of competition on each vendor's profit. Recall that net benefit from commercial software would be

$$u_C = v - p_C.$$

On the other hand, open source software with a support model gives different levels of benefit to the customers in different types:

$$u_{SH} = v + f$$
$$u_{SL} = v - p_S \quad \text{given} \quad p_S \leq s.$$

Proposition 4: *In a duopoly market with a commercial software vendor and an open source software vendor with a support model, the commercial software vendor charges price at the open source vendor's marginal cost for support* ($p_C^* = c_S$)*. The commercial software vendor makes a positive profit* ($\pi_C = (1-\mu)(1-c_S)c_S$) *while the open source software vendor makes zero profit. When the marginal cost for support is large* $\left(\frac{1}{2} < c_S < 1\right)$*, the commercial software vendor charges its monopoly price.*

Proof. Consider the high-type customers. Since $u_C = v - p_C < v + f = u_{SH}$, high-type customers always choose the open source software. Thus, both vendors compete for the low-type segment only. The demand for the commercial and the open source software would be

$$q_{CL}(p_C,p_S)=\begin{cases} q_L(p_C) & \text{if} \quad p_C < p_S \\ \frac{1}{2}q_L(p_C) & \text{if} \quad p_C = p_S \\ 0 & \text{if} \quad p_C > p_S \end{cases} \quad \text{and} \quad q_{SL}(p_S,p_C)=\begin{cases} q_L(p_S) & \text{if} \quad p_S < p_C \\ \frac{1}{2}q_L(p_S) & \text{if} \quad p_S = p_C \\ 0 & \text{if} \quad p_S > p_C. \end{cases}$$

Note that

$$\pi_C = (1-\mu)q_{CL}p_C$$
$$\pi_S = (1-\mu)q_{SL}(p_S - c_S).$$

First, no vendor charges less than marginal cost to avoid making negative profit. That is,

$$p_C \ge 0 \quad \text{and} \quad p_S \ge c_S.$$

Second, there cannot be an equilibrium in which both p_C and p_S are strictly above c_S. Consider the case where $p_C > p_S > c_S$. Then the commercial software vendor can obtain the entire demand and increases its profit by charging at $p_S - \varepsilon$ where ε is a small and positive number. If $p_S > p_C > c_S$, the open source software vendor has an incentive to lower its support charge to $p_C - \varepsilon$. Suppose that $p_C = p_S > c_S$. Both vendors still want to lower their prices by ε so that they can increase profit.

Finally, the commercial software vendor can obtain the entire demand by charging $c_S - \varepsilon$. The equilibrium prices can be obtained as

$$p_C^* = p_S^* = c_S.$$

The commercial software has the whole market and the open source software vendor makes zero profit. The profit for the commercial software vendor would be

$$\pi_C^* = (1-\mu)q_{CL}^* p_C^* = (1-\mu)(1-c_S)c_S.$$

When $\frac{1}{2} < c_S < 1$, the commercial vendor charges its monopoly price $p_C^{Monopoly} = \frac{1}{2}$ and

$$\pi_C^{Duopoly} = \frac{1}{4}(1-\mu) < \frac{1}{4} = \pi_C^{Monopoly}.$$

QED.

Unlike the dual-licensing model, the open source support model is not viable in a duopoly market. The commercial software vendor's best strategy is not to allow the open source vendor to make any positive profit by aggressive pricing. Since there is an asymmetry in marginal cost, the open source software vendor is not able to survive the competition. Our result shows that the commercial software vendor provides its software at the open source software vendor's marginal cost for support. Consequently, the open source vendor makes zero profit while the commercial software vendor is able to make positive profit.

Viability of Open Source Software Support Model

Our findings show that the support model for open source software is not viable in a duopoly market. However, some open source software vendors such as RedHat and JBoss have been successful in commercializing their software by adopting a support model. In this section, we consider other possible factors that may affect the competition framework. We investigate whether the open source support model is viable in other settings. First, we examine the role of switching cost to the competition by setting up a two-period model, and study the optimal pricing strategies of both the commercial and the open source software vendors. Then we examine the different quality case and study how asymmetry in quality affects the viability of the support model under competition.

Competition in the Presence of Switching Cost

In reality, an open source software vendor enters the market which has been dominated by a monopolist commercial software vendor such as Microsoft. In addition to software price, software adoption incurs significant switching cost when a user wants to switch from one software product to another. For example, in addition to direct cost for software adoption, changing from Microsoft Windows to Linux requires time and effort for learning Linux and cost for Linux-compatible applications. Such costs are considered as switching costs. In this section, we investigate whether the presence of switching costs changes the competition framework by setting up a two-period model. The decision flow is explained as below.

Period 1: Commercial Software Only

- Commercial software vendor introduces software at price p_1 for period 1 licensing.
- Customers decide whether or not to buy the software.
- The demand for commercial software (x_1) is determined.

Period 2: Commercial versus Open Source

- Period 2 game is based on the customers who purchased commercial software in period 1.
- A commercial software vendor offers software at price p_2 for period 2 licensing.
- At the same time, an open source software vendor gives the software for free to both types and provides support at p_S to customers in need.
- Customers choose either of the following strategies.
  - Stay with the commercial software and pay p_2.
  - Switch to open source software and incur switching cost wx_1

    (Switching cost depends on the scale parameter w and the market size x_1)
  - Exit, i.e., do not buy any software.

We consider the case where there is a sufficient switching cost, in other words, the maximum possible level of switching cost is greater than the flexibility benefit $(w > f)$. In this game, a customer has four strategic choices. A customer can buy commercial software in both period 1 and 2 (CC), or buy commercial software in period 1 and switch to open source software in period 2 (CO). It is also possible that the customer who buys the commercial software in period 1 chooses not to buy any in period 2 (CX). Finally, a customer can choose not to buy any software in any period (XX). Then high-type and low-type customers enjoy the following utility from each choice.

	High - Type (μ)	Low - Type $(1 - \mu)$
CC	$v - p_1 + v - p_2$	$v - p_1 + v - p_2$
CO	$v - p_1 + v + f - wx_1$	$v - p_1 + v - p_S - wx_1$
CX	$v - p_1$	$v - p_1$
XX	0	0

A further analysis leads to Proposition 5.

Proposition 5: *In the presence of the switching cost, a commercial software vendor sets up its price to prevent positive profit for the open source vendor with a support model. The optimal first and second period prices for the commercial software are as follows:*

$$p_1 = \frac{1}{3w^2}\left(3w^2 - 2wf - 2 + 1 - \sqrt{w^2(f^2 + f + 4) - 2w(1 + 2f) + 1}\right)$$

$$p_2 = \frac{1}{3}\left(1 - f - \frac{1}{w} + \frac{1}{w}\sqrt{w^2(f^2 + f + 4) - 2w(1 + 2f) + 1}\right).$$

Proof. See the Appendix.

The result is consistent with the case where no switching cost is present in the sense that the open source software vendor cannot make any profit with a support model. In the presence of switching cost, the commercial software vendor follows a pricing strategy not to allow positive profit for the open source vendor. Consequently, the open source software vendor does not have any incentive to enter the market. This result implies that the open source software support model is not viable when the open source software vendor competes with the commercial software vendor no matter whether switching cost is present or not.

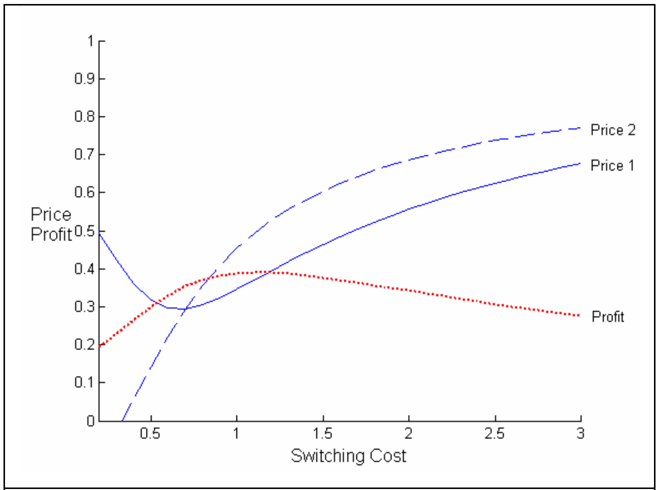


Figure 3. Price, Profit, and Switching Cost

Figure 3 illustrates the optimal first and second period prices and the profit for the commercial software. When the scale parameter for the switching cost is sufficiently large, the commercial software vendor charges low price in the first period and increases its price in the second so that it can have a large installed base in the first period and lock in the customers in the second. When the scale parameter is small, the commercial vendor cannot increase switching cost in the second period efficiently by having a large installed base in the first period. Thus, the optimal strategy is to charge higher price in the first period than in the second to maximize its profit. Interestingly, we find that the profit does not always increase with the scale parameter for the switching cost.

Quality Difference

We examine the different quality case and study how asymmetry in quality affects the viability of the open source software support model in competition. Let α be the quality difference parameter, which is a proportion of the value of the commercial software to the value of the open source software. We assume that quality is the only factor that affects the value of software. We use software value and software quality interchangeably. Then the customer's utility from the commercial software would be

$$u_C = \alpha v - p_C .$$

Recall that the net benefit from the open source software with a support model is

$$\begin{aligned} u_{SH} &= v + f \\ u_{SL} &= v - p_S \quad \text{given} \quad p_S \le s. \end{aligned}$$

There are two possible cases of asymmetric quality. The commercial software offers higher quality than the open source software ($\alpha > 1$) and the open source software quality is higher than the commercial software ($0 < \alpha < 1$). We start with the former case.

Case 1: Better Commercial Software ($\alpha > 1$)

High-Type Customers

Note that $u_C - u_{SH} = \alpha v - p_C - v - f$. The customer in the high-type segment would buy the commercial software $u_C - u_{SH} > 0,$ i.e. $(\alpha - 1)v > p_C + f$. Since $\alpha > 1$,

$$\frac{p_C + f}{\alpha - 1} < v < 1 \quad \Rightarrow \quad \text{Buy commercial software}$$

$$0 < v < \frac{p_C + f}{\alpha - 1} \quad \Rightarrow \quad \text{Get open source software for free}$$

Thus, the demand for the commercial and the open source software would be

$$q_{CH} = \int_{\frac{p_C + f}{\alpha - 1}}^{1} dv = 1 - \frac{p_C + f}{\alpha - 1}$$

$$q_{SH} = \int_{0}^{\frac{p_C + f}{\alpha - 1}} dv = \frac{p_C + f}{\alpha - 1}.$$

Low-Type Customers

The customer in the low-type segment would buy the commercial software $u_C - u_{SL} > 0,$ i.e. $(\alpha - 1)v > p_C - p_S$.

$$\frac{p_C - p_S}{\alpha - 1} < v < 1 \quad \Rightarrow \quad \text{Buy commercial software}$$

$$p_S < v < \frac{p_C - p_S}{\alpha - 1} \quad \Rightarrow \quad \text{Get open source software and buy support}$$

$$0 < v < p_S \quad \Rightarrow \quad \text{Not buy any software}$$

Thus, the demand for the commercial and the open source software would be

$$q_{CL} = \int_{\frac{p_C - p_S}{\alpha - 1}}^{1} dv = 1 - \frac{p_C - p_S}{\alpha - 1}$$

$$q_{SL} = \int_{p_S}^{\frac{p_C - p_S}{\alpha - 1}} dv = \frac{p_C - p_S}{\alpha - 1} - p_S.$$

The profit for the commercial vendor is

$$\begin{aligned}\pi_C &= \{\mu q_{CH} + (1-\mu) q_{CL}\} p_C \\ &= \left\{\mu\left(1 - \frac{p_C + f}{\alpha - 1}\right) + (1-\mu)\left(1 - \frac{p_C - p_S}{\alpha - 1}\right)\right\} p_C \\ &= \frac{1}{\alpha - 1}\{(1-\mu)p_S - p_C + \alpha - \mu f - 1\} p_C.\end{aligned}$$

The first-order condition is

$$\frac{\partial \pi_C}{\partial p_C} = \frac{1}{\alpha - 1}\{(1-\mu)p_S - 2p_C + \alpha - \mu f - 1\} = 0.$$

The profit for the open source software vendor comes from only low-type customers. Thus,

$$\begin{aligned}\pi_S &= (1-\mu) q_{SL}(p_S - c_S) \\ &= (1-\mu)\left(\frac{p_C - p_S}{\alpha - 1} - p_S\right)(p_S - c_S) \\ &= \frac{1}{\alpha - 1}(1-\mu)(p_C - \alpha p_S)(p_S - c_S).\end{aligned}$$

The first-order condition is

$$\frac{\partial \pi_C}{\partial p_C} = \frac{1}{\alpha - 1}(1-\mu)(p_C - 2\alpha p_S + \alpha c_S) = 0.$$

Solving for the following equations give optimal prices.

$$(1-\mu)p_S - 2p_C + \alpha - \mu f - 1 = 0$$
$$p_C - 2\alpha p_S + \alpha c_S = 0.$$

The optimal prices are

$$p_C^* = \frac{\alpha\{2\alpha + (1-\mu)c_S - 2(1+\mu f)\}}{4\alpha + \mu - 1}$$
$$p_S^* = \frac{\alpha(1+2c_S) - (1+\mu f)}{4\alpha + \mu - 1}.$$

Case 2: Better Open Source Software ($0 < \alpha < 1$ **)**

High-Type Customers

Note that $u_C - u_{SH} = \alpha v - p_C - v - f = -(1-\alpha)v - p_c - f < 0.$ Thus, high-type customers always choose the open source software. When the open source software provides higher quality than the commercial software, both software vendors target the low-type customers only.

Low-Type Customers

The customer in a low-type segment would buy open source software $u_{SL} - u_C > 0$, i.e. $(1-\alpha)v > p_S - p_C$.

$\frac{p_S - p_C}{1-\alpha} < v < 1 \quad \Rightarrow$ Get open source software and buy support

$p_C < v < \frac{p_S - p_C}{1-\alpha} \quad \Rightarrow$ Buy commercial software

$0 < v < p_C \quad \Rightarrow$ Not buy any software

Thus, the demand for the commercial and the open source software would be

$$q_{CL} = \int_{p_C}^{\frac{p_S - p_C}{1-\alpha}} dv = \frac{p_S - p_C}{1-\alpha} - p_C = \frac{p_S - p_C(2-\alpha)}{1-\alpha}$$
$$q_{SL} = \int_{\frac{p_S - p_C}{1-\alpha}}^{1} dv = 1 - \frac{p_S - p_C}{1-\alpha}.$$

The profit for the commercial software vendor is

$$\pi_C = (1-\mu)q_{CL}p_C$$
$$= \frac{1-\mu}{1-\alpha}\{p_S - p_C(2-\alpha)\}p_C.$$

The first-order condition is

$$\frac{\partial \pi_C}{\partial p_C} = \frac{1-\mu}{1-\alpha}\{p_S - 2p_C(2-\alpha)\} = 0.$$

The profit for the open source software vendor is

$$\pi_S = (1-\mu)q_{SL}(p_S - c_S)$$
$$= \frac{1-\mu}{1-\alpha}(1 - \alpha - p_S + p_C)(p_S - c_S).$$

The first-order condition is

$$\frac{\partial \pi_S}{\partial p_S} = \frac{1-\mu}{1-\alpha}(1-\alpha-2p_S+p_C+c_S)=0.$$

Solving for the following equations give optimal prices.

$$p_S - 2p_C(2-\alpha)=0$$
$$1-\alpha-2p_S+p_C+c_S=0.$$

The optimal prices are

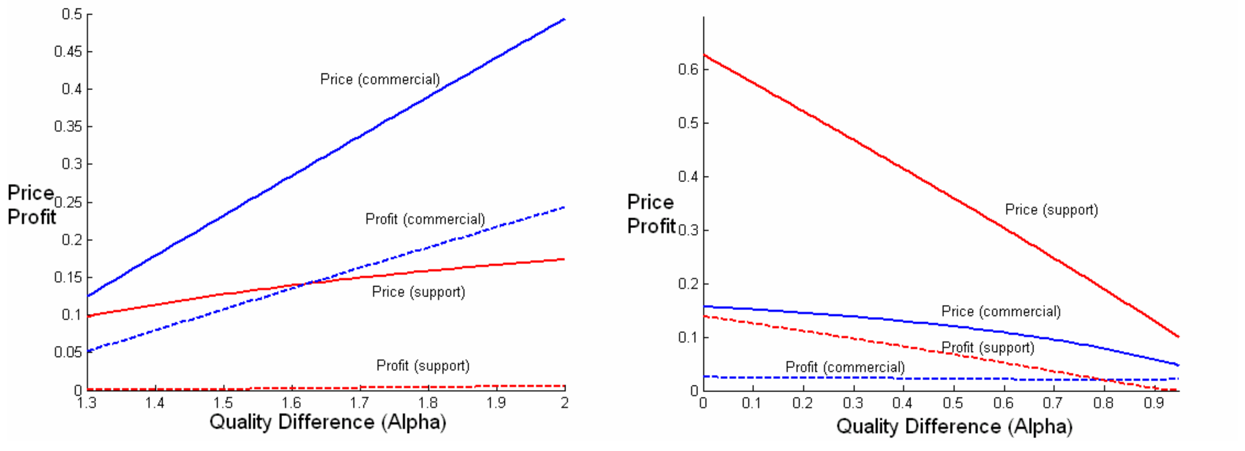
$$p_C^* = \frac{c_S+1-\alpha}{7-4\alpha}$$
$$p_S^* = \frac{2(c_S+1-\alpha)(2-\alpha)}{7-4\alpha}.$$

A further analysis of the prices leads to the Proposition 6.

Proposition 6: *In a software market where the commercial software vendor provides higher quality than the open source software with a support model* $(\alpha > 1)$ *, the commercial software vendor charges more for the software* $(p_C \uparrow)$ *and the open source software vendor charges more for the support* $(p_S \uparrow)$ *as the quality difference between the commercial software and the open source software increases* $(\alpha \uparrow)$*. The price difference* $(p_C - p_S)$ *increases with the quality difference. When the open source software quality is higher than the commercial software* $(0<\alpha<1)$ *, the open source software vendor charges more on support* $(p_S \uparrow)$ *as the quality difference increases* $(\alpha \downarrow)$ *. When the marginal cost for support is low* $(0<c_S<\frac{3}{4})$, *commercial software price increases* $(p_C \uparrow)$ *with quality difference* $(\alpha \downarrow)$*. Otherwise* $(\frac{3}{4}<c_S<1)$, *commercial software price decreases* $(p_C \downarrow)$ *with quality difference* $(\alpha \downarrow)$*.*

Proof. See the Appendix.

Proposition 6 shows that the open source support model is viable in the presence of quality asymmetry. No matter whether the quality of the open source software is higher or lower than the commercial software, the open source software vendor makes profit with a support model in a duopoly market. When the open source software has higher quality, it is reasonable that the open source software is profitable. The opposite case is interesting in the sense that the open source software with lower quality is also profitable although we find that the open source software with same quality as the commercial software is not. This can be explained as the market expansion of the commercial software vendor. When the quality is identical for both vendors, the high-type customers have no incentive to buy the commercial software, since the open source software is free and they do not need any support. However, when the commercial software offers higher quality than the open source software, some high-type customers will buy the commercial software, which implies that the commercial software vendor can target such high-type customers as well.



(a) Better Commercial Software ($\alpha > 1$)

(b) Better Open Source Software ($0 < \alpha < 1$)

Figure 4. Software Price and Profit

Figure 4 illustrates the optimal prices and profits with respect to the quality difference. Unlike the case of symmetric quality, the open source software vendor with a support model can make positive profit in the presence of quality asymmetry. Even when the open source software provides worse quality, the open source software vendor makes a thin but positive profit. As mentioned earlier, this can be a result of market expansion. This quality difference can be one possible way to explain the successful open source support model by some open source software vendors such as RedHat and JBoss. Our findings support the viability of the open source software support model.

Conclusion

In this paper, we compare three different software pricing mechanisms: commercial software, open source dual-licensing, and open source software support. We start with examining the optimal pricing strategies of a software vendor with each pricing scheme under monopoly. We find that the monopoly price for open source software under dual-licensing is higher than the commercial software price, hence higher profit, if total flexibility benefit for the high-type customers exceeds total support cost for the low-type customers. With a support model, the open source vendor charges for support at higher price than the commercial software price, but it makes a lower profit than the commercial software vendor. Then we investigate whether the open source software vendor prefers one pricing mechanism over the other in the monopoly market. Our result shows that the dual-licensing model is more profitable when the difference between flexibility benefit and support cost is high. Otherwise, the support model is preferred. We identify the factors that affect the open source software vendor's decision.

We extend our model to a duopoly competition setting. In a market with competition between a commercial software vendor and an open source software vendor with a dual-licensing model, the open source software vendor serves high-type customers only by charging more than the commercial software vendor that covers low-type segment only. Our result supports the viability of the dual-licensing model for the open source software in the presence of competition. When there is a competition between a commercial software vendor and an open source software vendor with a support model, the commercial software vendor competing against the open source support model charges a price equal to the open source vendor's marginal cost of support. With this pricing strategy of the commercial software vendor, the support model brings zero profit to the open source software vendor.

We focus on the viability of the support model for the open source software. We examine the role of switching cost to the competition between commercial and open source software. The result is consistent with the case where no switching cost is present in the sense that the open source software vendor cannot make any positive profit. The optimal pricing strategy of the commercial software vendor is interesting. When the switching cost is large, the commercial software vendor charges a low price in the first period, and increases its price in the second so that it can have a large installed base in the first period and lock in the customers in the second. However, with small switching cost, the commercial software vendor charges a higher price in the first period than in the second, since increasing its installed base is not very effective in raising switching cost to the customers. Next, we examine the different quality case. Interestingly, we find that the open source software support model is viable in the presence of quality asymmetry no matter whether the quality of the open source software is higher or lower than the commercial software. Even when the open source software provides lower quality, the support model is profitable. This can be explained as market expansion of the commercial software vendor.

Our paper contributes to the literature in the following ways. First, this paper is the first study that examines the issue of pricing open source software through an economic lens. In spite of the growing interest in the commercial open source software among industry experts and jurists, no academic study has shown the viability of the pricing models of for-profit open source software. We identify the factors that affect the viability of the pricing models for open source software and find the conditions under which each model can be successful. Second, our result can give pricing guidelines to the open source software vendors, which is not clear in the current state. Finally, we model the motivation for and the barrier to open source adoption, which provides a better picture of the open source software market. Considering such factors, consistent with survey statistics, may allow us to better understand the issue of open source software.

Appendix

Proof of Proposition 1

Note that

$$p_D^* - p_C^* = \frac{\mu f - (1-\mu)s + 1}{2} - \frac{1}{2}$$
$$= \frac{1}{2}\{\mu f - (1-\mu)s\}.$$

Thus,

$$p_D^* > p_C^* \quad \text{if} \quad \mu f > (1-\mu)s$$
$$p_D^* < p_C^* \quad \text{if} \quad \mu f < (1-\mu)s.$$

The profit difference is

$$\pi_D^* - \pi_C^* = \frac{1}{4}\{\mu f - (1-\mu)s + 1\}^2 - \frac{1}{4}.$$

Since $p_D^* = \dfrac{\mu f - (1-\mu)s + 1}{2} \geq 0,$

$$\mu f - (1-\mu)s + 1 \geq 0.$$

Thus,

$$\pi_D^* - \pi_C^* > 0 \quad \text{if} \quad \mu f - (1-\mu)s + 1 - 1 > 0.$$

Therefore,

$$\pi_D^* > \pi_C^* \quad \text{if} \quad \mu f > (1-\mu)s$$
$$\pi_D^* < \pi_C^* \quad \text{if} \quad \mu f < (1-\mu)s.$$

The price difference between open source software support and commercial software is

$$p_S^* - p_C^* = \frac{1+c_S}{2} - \frac{1}{2} = \frac{c_S}{2} > 0.$$

Therefore, $p_S^* > p_C^*$. Note that

$$\pi_S^* \text{ - } \pi_C^* = \frac{1}{4}(1-\mu)(1-c_S)^2 - \frac{1}{4}.$$

The marginal cost is bounded by price, i.e. $c_S \leq p_S^* = \dfrac{1+c_S}{2}$. Therefore, $0 < c_S \leq 1$. Since $0 \leq \mu \leq 1$, we have $0 \leq (1-\mu)(1-c_S)^2 < 1$. Thus,

$$\pi_S^* \text{ - } \pi_C^* = \frac{1}{4}(1-\mu)(1-c_S)^2 - \frac{1}{4} < 0.$$

Therefore, $\pi_S^* < \pi_C^*$. ***QED.***

Proof of Proposition 3

Recall that a customer's utility from commercial software would be

$$u_C = v - p_C.$$

By using open source software sold under a dual-licensing model, a high-type and a low-type customer enjoys the following benefits:

$$u_{DH} = v - p_D + f$$
$$u_{DL} = v - p_D - s.$$

Thus, the demands for commercial and open source software from high-type and low-type customers are as follows:

High-Type Customers

$$q_{CH}(p_C, p_D) = \begin{cases} q_H(p_C) & \text{if} \quad p_C < p_D - f \\ \frac{1}{2} q_H(p_C) & \text{if} \quad p_C = p_D - f \\ 0 & \text{if} \quad p_C > p_D - f \end{cases} \quad \text{and} \quad q_{DH}(p_D, p_C) = \begin{cases} q_H(p_D) & \text{if} \quad p_D - f < p_C \\ \frac{1}{2} q_H(p_D) & \text{if} \quad p_D - f = p_C \\ 0 & \text{if} \quad p_D - f > p_C. \end{cases}$$

Low-Type Customers

$$q_{CL}(p_C, p_D) = \begin{cases} q_L(p_C) & \text{if} \quad p_C < p_D + s \\ \frac{1}{2} q_L(p_C) & \text{if} \quad p_C = p_D + s \\ 0 & \text{if} \quad p_C > p_D + s \end{cases} \quad \text{and} \quad q_{DL}(p_D, p_C) = \begin{cases} q_L(p_D) & \text{if} \quad p_D + s < p_C \\ \frac{1}{2} q_L(p_D) & \text{if} \quad p_D + s = p_C \\ 0 & \text{if} \quad p_D + s > p_C. \end{cases}$$

Note that the commercial software vendor loses all the demand if it charges price p_C above $p_D + s$. Thus, $p_C \in (p_D + s, 1)$ is a dominated strategy. The commercial software vendor would charge price at $p_C \in (0, p_D + s)$ to guarantee the demand from the customers in the low-type segment. From the open source software vendor's perspective, charging price at p_D such that $p_D - f > p_C$ is a dominated strategy since such pricing would lead to zero demand for the open source software vendor. Thus, the open source software vendor would charge price at $p_D \in (0, p_C + f)$ and serve the high-type customers. If an equilibrium exists, at equilibrium, the commercial software vendor uses a pricing strategy to serve the low-type customer segment while the open source software vendor serves the high-type customer segment. Consider the profit for the commercial software vendor:

$$\pi_C = (1 - \mu)(1 - p_C)p_C.$$

The optimization problem for the commercial software vendor is

$$\underset{p_C}{\text{Max}}\, \pi_C \quad \text{subject to} \quad p_C \le p_D + \text{s}.$$

The first-order condition for π_C is

$$\frac{\partial \pi_C}{\partial p_C} = (1 - \mu)(1 - 2p_C) = 0,$$

resulting in the profit-maximizing price as

$$p_C^* = \begin{cases} \frac{1}{2} & \text{if} \quad \frac{1}{2} \le p_D + \text{s} \\ p_D + \text{s} & \text{if} \quad \frac{1}{2} > p_D + \text{s}. \end{cases}$$

The open source software vendor's profit would be

$$\pi_D = \mu(1 - p_D + f)p_D.$$

The optimization problem for the commercial software vendor is

$$\operatorname*{Max}_{p_D} \pi_D \quad \text{subject to} \quad p_D \le p_C + f.$$

The first-order condition for π_D is

$$\frac{\partial \pi_D}{\partial p_D} = \mu(1 - 2p_D + f) = 0,$$

leading to the optimal price as

$$p_D^* = \begin{cases} \dfrac{1+f}{2} & \text{if} \quad \dfrac{1+f}{2} \le p_C + f \\ p_C + f & \text{if} \quad \dfrac{1+f}{2} > p_C + f. \end{cases}$$

Now, we investigate whether $(p_C^*, p_D^*) = (\frac{1}{2}, \frac{1+f}{2})$ is a Nash equilibrium. Consider the case when the open source software vendor offers price at $p_D^* = \frac{1+f}{2}$. We investigate whether the commercial software vendor has incentive to deviate from $p_C^* = \frac{1}{2}$. Since $f > 0$ and $s > 0$,

$$p_D^* - f = \frac{1-f}{2} < p_C^* = \frac{1}{2} < p_D^* + s = \frac{1+f}{2} + s.$$

Suppose that $f \ge 1$. Then $p_D^* - f = \frac{1-f}{2} \le 0$. Thus, the commercial software vendor cannot offer any price below $p_D^* - f$, meaning that the commercial software vendor cannot serve any customer in the high-type segment. Since $p_C^* = \frac{1}{2}$ is the price that maximizes profit from serving low-type segment, the commercial software vendor has no incentive to deviate from $p_C^* = \frac{1}{2}$. Thus, $p_C^* = \frac{1}{2}$ is a best response to $p_D^* = \frac{1+f}{2}$. Now, suppose that $0 < f < 1$. Then $p_D^* - f = \frac{1-f}{2} > 0$. In this case, the commercial software vendor has two choices as below:

Strategy 1: Stay at $p_C^* = \frac{1}{2}$ and serve low-type customers only.

Strategy 2: Deviate to $p_C^* = \frac{1-f}{2}$ and serve both segments.

Denote π_C^1 and π_C^2 be the profit for the commercial software vendor playing strategy 1 and 2, respectively. Then

$$\pi_C^1 = (1-\mu)(1-p_C^*)p_C^* = (1-\mu)(1-\frac{1}{2})\frac{1}{2} = \frac{1-\mu}{4}$$
$$\pi_C^2 = (1-p_C^*)p_C^* = (1-\frac{1-f}{2})\frac{1-f}{2} = \frac{1-f^2}{4}.$$

If $\pi_c^1 \ge \pi_c^2$, the commercial software vendor has no incentive to deviate. Note that $\pi_C^1 \ge \pi_C^2 \Leftrightarrow f \ge \sqrt{\mu}$. Therefore, $p_C^* = \frac{1}{2}$ is a best response to $p_D^* = \frac{1+f}{2}$ if $f \ge \sqrt{\mu}$.

Now, we investigate $p_D^* = \frac{1+f}{2}$ is a best response of the open source software vendor to $p_C^* = \frac{1}{2}$ offered by the commercial software. Suppose that $s \geq \frac{1}{2}$. The open source software vendor cannot charge any price p_D such that $p_D + s < p_C^* = \frac{1}{2}$. Thus, the open source software vendor cannot serve any customer in the low-type segment. Since $p_D^* = \frac{1+f}{2}$ is the optimal price when serving low-type customers only, the open source software vendor has no incentive to deviate from $p_D^* = \frac{1+f}{2}$. Thus, $p_D^* = \frac{1+f}{2}$ is a best response to $p_C^* = \frac{1}{2}$. Now, suppose that $0 < s < \frac{1}{2}$. Then $p_C^* - s = \frac{1}{2} - s > 0$. Thus, the open source software vendor has two choices as follows:

Strategy 1: Stay at $p_D^* = \frac{1+f}{2}$ and serve high-type customers only

Strategy 2: Deviate to $p_D^* = \frac{1}{2} - s$ and serve both segments.

Denote π_D^1 and π_D^2 be the profit for the commercial software vendor playing strategy 1 and 2, respectively. Then

$$\pi_D^1 = \mu(1 - p_D^* + f)p_D^* = \mu(1 - \frac{1+f}{2} + f)\frac{1+f}{2} = \frac{\mu(1+f)^2}{4}$$

$$\pi_D^2 = \{\mu(1 - p_D^* + f) + (1-\mu)(1 - p_D^* - s)\}p_D^* = \{\mu(1 - \frac{1}{2} + s + f) + (1-\mu)(1 - \frac{1}{2} + s - s)\}(\frac{1}{2} - s)$$

$$= \frac{1}{4}(2\mu f + 2\mu s + 1)(1 - 2s).$$

Note that $\pi_C^1 \geq \pi_C^2 \Leftrightarrow \mu(1+f)^2 \geq (2\mu f + 2\mu s + 1)(1-2s)$. Thus, $p_D^* = \frac{1+f}{2}$ is a best response to $p_C^* = \frac{1}{2}$ if the following conditions hold:

$$s \geq \frac{1}{2} \text{ or } \left\{0 < s < \frac{1}{2} \text{ and } \mu(1+f)^2 \geq (2\mu f + 2\mu s + 1)(1-2s)\right\}.$$

Therefore, $(p_C^*, p_D^*) = (\frac{1}{2}, \frac{1+f}{2})$ is a Nash equilibrium when

$$\left\{f \geq \sqrt{\mu} \text{ and } s \geq \frac{1}{2}\right\} \text{ or } \left\{f \geq \sqrt{\mu},\ 0 < s < \frac{1}{2} \text{ and } \mu(1+f)^2 \geq (2\mu f + 2\mu s + 1)(1-2s)\right\}.$$

QED.

Proof of Proposition 5

Note that a customer who chooses to play CC enjoys positive utility: $v - p_1 \geq 0$ and $v - p_2 \geq 0$. For such customers, XX and CX are dominated by CC. Comparison of CC and CO is not trivial. Consider period 2. A high-type customer prefers the commercial software to the open source software if

$$v - p_1 + v + f - wx_1 \leq v - p_1 + v - p_2 .$$

Note that x_1 is the period 1 demand for the commercial software, which is, $1 - p_1$. Thus, the above condition can be written as

$$wp_1 + p_2 \leq w - f\,.$$

A low-type customer stays with the commercial software if

$$\begin{aligned} & v - p_1 + v - p_S - wx_1 \leq v - p_1 + v - p_2 \\ \Leftrightarrow\ & wp_1 + p_2 \leq w + p_S\,. \end{aligned}$$

Since $w > f$, the commercial software vendor can serve both high-type and low-type customers in both periods by setting up its prices satisfying $wp_1 + p_2 \leq w - f$. Then, the open source software's best strategy is to price its support at the marginal cost and to make zero profit. The commercial software vendor's profit is

$$\begin{aligned} \pi_C &= q_1 p_1 + q_2 p_2 \\ &= (1 - p_1)p_1 + (1 - p_1)(1 - p_2)p_2\,. \end{aligned}$$

The optimization problem is then

$$\underset{p_1, p_2}{\text{Max}}\, \pi_C \quad \text{subject to } wp_1 + p_2 \leq w - f.$$

The Lagrangean function can be defined as

$$L(p_1, p_2, \lambda) = (1 - p_1)p_1 + (1 - p_1)(1 - p_2)p_2 - \lambda(wp_1 + p_2 - w + f).$$

The Kuhn-Tucker conditions are

$$\begin{aligned} \frac{\partial L}{\partial p_1} &= 1 - 2p_1 - (1 - p_2)p_2 - w\lambda = 0 \\ \frac{\partial L}{\partial p_2} &= (1 - p_1)(1 - 2p_2) - \lambda = 0 \\ \frac{\partial L}{\partial \lambda} &= wp_1 + p_2 - w + f = 0. \end{aligned}$$

From $\dfrac{\partial L}{\partial \lambda}$, we have $p_2 = w - f - wp_1$. From $\dfrac{\partial L}{\partial p_1}$ and $\dfrac{\partial L}{\partial p_2}$,

$$\begin{aligned} & 1 - 2p_1 - (1 - p_2)p_2 = w(1 - p_1)(1 - 2p_2) \\ \Leftrightarrow\ & p_2^2 + p_2(2w - 2wp_1 - 1) + p_1(w - 2) + 1 - w = 0. \end{aligned}$$

Plugging $p_2 = w - f - wp_1$ in the above equation leads to

$$\begin{aligned} & (w - f - wp_1)^2 + (w - f - wp_1)(2w - 2wp_1 - 1) + p_1(w - 2) + 1 - w = 0 \\ \Leftrightarrow\ & 3w^2 p_1^2 - 2p_1(3w^2 - 2wf - w + 1) + (3w^2 - 4wf - 2w + f^2 + f + 1) = 0. \end{aligned}$$

Solving the above equation gives the equilibrium prices as follows:

$$\begin{aligned} p_1 &= \frac{1}{3w^2}\left(3w^2 - 2wf - 2 + 1 - \sqrt{w^2(f^2 + f + 4) - 2w(1 + 2f) + 1}\right) \\ p_2 &= \frac{1}{3}\left(1 - f - \frac{1}{w} + \frac{1}{w}\sqrt{w^2(f^2 + f + 4) - 2w(1 + 2f) + 1}\right). \end{aligned}$$

QED.

Proof of Proposition 6

Suppose that $\alpha > 1$. Recall that

$$p_C^* = \frac{\alpha\{2\alpha + (1-\mu)c_S - 2(1+\mu f)\}}{4\alpha + \mu - 1}$$

$$p_S^* = \frac{\alpha(1+2c_S) - (1+\mu f)}{4\alpha + \mu - 1}.$$

Note that

$$\begin{aligned}\frac{\partial p_C^*}{\partial \alpha} &= \frac{1}{(4\alpha+\mu-1)^2}[(4\alpha+\mu-1)\{4\alpha + (1-\mu)c_S - 2(1+\mu f)\} - 8\alpha^2 - 4\alpha\{(1-\mu)c_S - 2(1+\mu f)\}] \\ &= \frac{1}{(4\alpha+\mu-1)^2}[4\alpha(2\alpha+\mu-1) - (1-\mu)\{(1-\mu)c_S - 2(1+\mu f)\}]\end{aligned}$$

Since $\alpha > 1$ and $0 \le \mu \le 1$,

$$4\alpha(2\alpha + \mu - 1) > 0.$$

Note that $c_S \le 1$ and $f \ge 0$. Thus,

$$(1-\mu)c_S - 2(1+\mu f) < 0.$$

Therefore, $\dfrac{\partial p_C^*}{\partial \alpha} > 0.$

Note that

$$\begin{aligned}\frac{\partial p_S^*}{\partial \alpha} &= \frac{1}{(4\alpha+\mu-1)^2}\{(4\alpha+\mu-1)(1+2c_S) - 4\alpha(1+2c_S) + 4(1+\mu f)\} \\ &= \frac{1}{(4\alpha+\mu-1)^2}\{4(1+\mu f) - (1-\mu)(1+2c_S)\} \\ &= \frac{1}{(4\alpha+\mu-1)^2}\{\mu(4f+1+2c_S) + 3 - 2c_S\}.\end{aligned}$$

Since $\mu > 0$, $f > 0$, and $0 < c_S < 1$,

$$4f + 1 + 2c_S > 0 \quad \text{and} \quad 3 - 2c_S > 0.$$

Therefore, $\dfrac{\partial p_S^*}{\partial \alpha} > 0.$

Recall that the first-order condition for the commercial software vendor's profit is

$$\frac{\partial \pi_C}{\partial p_C} = \frac{1}{\alpha - 1}(1-\mu)(p_C - 2\alpha p_S + \alpha c_S) = 0.$$

Thus,

$$p_C^* = \alpha(2p_S^* - c_S).$$

Let δ be the difference between the commercial software price and the support charge for open source, i.e.

$$\delta = p_C^* - p_S^* = (2\alpha - 1)p_S^* - \alpha c_S.$$

Then

$$\frac{\partial \delta}{\partial \alpha} = 2p_S^* + (2\alpha - 1)\frac{\partial p_S^*}{\partial \alpha} - c_S.$$

Since $c_S \le p_S < 2p_S$ and $\alpha > 1$,

$$2p_S^* - c_S > 0 \text{ and } 2\alpha - 1 > 0.$$

We show that $\frac{\partial p_S^*}{\partial \alpha} > 0$. Therefore, $\frac{\partial \delta}{\partial \alpha} > 0$.

Now suppose that $0 < \alpha < 1$. Note that

$$p_C^* = \frac{c_S + 1 - \alpha}{7 - 4\alpha}$$
$$p_S^* = \frac{2(c_S + 1 - \alpha)(2 - \alpha)}{7 - 4\alpha}.$$

Thus,

$$\frac{\partial p_C^*}{\partial \alpha} = \frac{1}{(7-4\alpha)^2}\{-(7-4\alpha) + 4(c_S + 1 - \alpha)\} = \frac{4c_S - 3}{(7-4\alpha)^2}.$$

Therefore, $\frac{\partial p_C^*}{\partial \alpha} > 0$ if $\frac{3}{4} < c_S < 1$ and $\frac{\partial p_C^*}{\partial \alpha} < 0$ if $0 < c_S < \frac{3}{4}$.

Note that

$$\begin{aligned}\frac{\partial p_S^*}{\partial \alpha} &= \frac{2}{(7-4\alpha)^2}[(7-4\alpha)(2\alpha - c_S - 3) + 4\{\alpha^2 - \alpha(c_S + 3) + 2(c_S + 1)\}] \\ &= \frac{2}{(7-4\alpha)^2}(-4\alpha^2 + 14\alpha + c_S - 13) \\ &= \frac{2}{(7-4\alpha)^2}\{-2(2\alpha - 5)(\alpha - 1) + c_S - 3\}.\end{aligned}$$

Since $2\alpha - 5 < 0$, $\alpha - 1 < 0$, and $c_S - 3 < 0$, $\frac{\partial p_S^*}{\partial \alpha} < 0$. Finally, $p_S^* = 2(2-\alpha)p_C^* > 2p_C^* > p_C^*$. ***QED.***

References

Broersma, M. “Companies Buy Open Source Because It’s Better, Not Cheaper”, *Techworld*, April 21, 2005.

Casadesus-Masanell, R., and Ghemawat, P. “Dynamic Mixed Duopoly: A Model Motivated by Linux vs. Windows”, *Management Science* 52(7), 2006, pp. 1072 – 1084.

Choudhary, V., Ghose, A., Mukhopadhyay, T., and Rajan, U. “Personalized Pricing and Quality Differentiation”, *Management Science* 51(7), 2005, pp. 1120 – 1130.

Dewan, S., and Mendelson, H. “User Delay Costs and Internal Pricing for a Service Facility.” *Management Science* 36(12), 1990, pp. 1502–1517.

Economides, N., and Katsamakas, E. “Two-Sided Competition of Proprietary vs. Open Implications for the Software Industry”, *Management Science* 52(7), 2006, pp. 1057 – 1071.

Economist “Microsoft at the Power Point”, September 11, 2003.

Gloude, M. “Open Source Usage Is Up, But Concerns Linger”, *Forrester Research*, June 23, 2005.

Gomulkiewicz, R. W. “Entrepreneurial Open Source Software Hackers: MySQL and Its Dual Licensing”, C*omputer Law Review and Technology Journal* (9), 2004, pp. 203 – 212.

Hitt, L., and Chen, P. “Bundling with Customer Self-Selection: A Simple Approach to Bundling Low Marginal Cost Goods”, M*anagement Science* 51(10), 2005, pp. 1481 – 1493.

Kingstone, S. “Brazil Adopts Open-Source Software”, June 2, 2005.

McCarthy, V. “Survey: Financial IT Execs Say Jury Still Out Open Source”, *OpenEnterpriseTrends.com*, March 24, 2006.

Mendelson, H., and Whang, S. “Optimal Incentive-Compatible Priority Pricing for the *M/M/*1 Queue.” *Operations Research* 38(5), 1990, pp. 870–883.

Moczar, L. “The Economics of Commercial Open Source”, *Galatea Information Strategies*, 2005.

Png, I., and Chen, Y. “Information Goods Pricing and Copyright Enforcement: Welfare Analysis.”, *Information Systems Research* 14(1), 2003, pp. 107–123.

Raghunathan, S., Prasad, A., Mishra, B., and Chang, H. “Open Source Versus Closed Source: Software Quality in Monopoly and Competitive Markets”, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* 35(6), 2005, pp. 903 – 918.

Reid R. “Forrester: More Firms Using Open Source”, *ITWorldCanada.com*, March 17, 2004.

Sundararajan, A. “Nonlinear Pricing of Information Goods.”, *Management Science* 50(12), 2004, pp. 1660-1673.

Välimäki, M. “Dual Licensing in Open Source Software Industry”, *Systemes d´Information et Management* 8(1), 2003, pp. 63-75.

Varian, H., and Shapiro, C. “Linux Adoption in the Public Sector: An Economic Analysis”, *Working Paper*, 2003.

Vaughan-Nichols S. J. “Making Money from Free Software”, *eWeek*, July 13, 2005.

Wichmann, T. “Free/Libre and Open Source Software: Survey and Study”, *Berlecon Research GmbH*, Berlin, Germany, 2002.

Westland, C. “Congestion and Network Externalities in the Short Run Pricing of Information Systems Services.”, *Management Science* 38(7), 1992, pp. 992–1099.