



# A general equilibrium analysis of software development: Implications of copyright protection and contract enforcement

Larry D. Qiu\*

*Department of Economics, Hong Kong University of Science and Technology, Kowloon, Hong Kong*

Received 14 May 2004; accepted 29 June 2005

Available online 22 September 2005

---

## Abstract

We develop a general equilibrium model to study the implications of a legal environment on the organization of software production. We show that contract enforcement determines the organizational mode (i.e., in-house versus outsourcing) of customized software development while copyright protection affects both packaged software as well as customized software development. We obtain some testable results: when copyright protection is weak, only customized software will be developed; when copyright protection is strong, both customized software and packaged software will be developed; environment changes in one software market affect the equilibrium in the other software market.

© 2005 Elsevier B.V. All rights reserved.

*JEL classification:* L22; L24; L86; D23

*Keywords:* Software; Piracy; Copyright protection; Contract enforcement; Vertical integration; Contract; Organization; Outsourcing

---

---

\*Tel.: + 852 2358 7628; fax: + 852 2358 2084.

*E-mail address:* [larryqiu@ust.hk](mailto:larryqiu@ust.hk).

*URL:* <http://www.bm.ust.hk/~larryqiu>.

## 1. Introduction

We are now living in the new economy in which information technology is playing a more and more important role. For example, in the United States, computer software is vital to both the domestic economy and external trade. In 1998, the software industry became the second largest industry group in manufacturing.<sup>1</sup> In 1997, packaged software (PS) alone contributed a surplus of \$13 billion to the U.S. trade balance, without which the U.S. trade deficit (excluding U.S. military and government transactions) would have been 36% higher (BSA, 1999b, p. 16). However, software development is very uneven across countries. In 1994, the U.S. controlled about 75% of the global software market. Europe had 20% of the market, and Japan had 4.3% (Fortune, 1994).

Unlike most products and services, though, software is protected by copyright laws against piracy. Software piracy is a serious problem all over the world, but it varies tremendously from country to country. According to a report by the Business Software Alliance (BSA, 1999a), in 1998, 38% of the business software applications in the world were pirated, and, by countries, the software piracy rate ranged from 25% (in the U.S.) to 97% (in Vietnam). It is not a coincidence that the U.S. has the lowest software piracy rate and at the same time has the largest market share of software in the world. The BSA concludes that “[p]rotecting the intellectual property rights that are the basis of packaged software distribution is conducive to greater international trade and increased industry investment in the economies that provide such protections” (BSA, 1998, p. 24).

In this paper, we develop a model to analyze software development formally with a focus on the influence of the legal environment, namely contract enforcement and copyright protection policies. Software products are commonly classified into two groups: PS which is designed for general purposes, like word processing, and *customized software* (CS, in short), which is designed for special use, like an accounting program for a particular company.<sup>2</sup> We show that the degree of contract enforcement affects the organizational *mode* of the CS development, i.e., whether the software is developed in-house (vertical integration) or obtained by outsourcing (contracting). We define a *type-B contract* (B stands for breach) as a contract that will be breached by one contracting party, and a *type-H contract* (H stands for honor) as a contract that will be always honored by all contracting parties. We show that as contract enforcement becomes stronger, the optimal mode of organization switches from vertical integration to type-B contract, then to vertical integration, and finally to type-H contract. For PS, since piracy reduces legitimate consumption, copyright protection is essential to ensure investment in software development. As a result, when copyright protection is weak, only CS will be developed. When

---

<sup>1</sup>The software industry tied with the electronic components and accessories industry for second place, after the motor vehicles and equipment industry, according to a report by the Business Software Alliance (BSA, 1999b, p. 6).

<sup>2</sup>Examples of PS include operating systems, word processing programs and spreadsheets. Examples of CS include some specific software designed for banks and some professional training programs. See Torrisi (1998) for more about this classification.

copyright protection is strong, both CS and PS will be developed. Moreover, we demonstrate the cross-product effects: environment changes in one software market affect the equilibrium in the other software market.

This paper makes a contribution to the literature by developing an economic model that is suitable for a general equilibrium analysis of software industry. The model developed in this paper has several distinguishing features that are crucial for analyzing software development. The industry is assumed to produce two distinct products, and we examine the implications of contract enforcement and copyright protection for these products. The production of CS and production of PS are linked via the labor (software programmers) market in a general equilibrium model. We endogenize the types of software produced as well as the organization model of production. In this way, the present study is related to but significantly different from others in the literature that we discuss below.

First, consider CS. There are three alternative modes through which a piece of software is developed and later delivered to the user: vertical integration, a contract and the market. To emphasize the difference between CS and PS, we assume that any CS is so specific that it has only one user. This immediately eliminates the market as a plausible development and delivery mode. Therefore, our analysis focuses on a comparison between vertical integration and contracts. Unlike us, in analyzing upstream and downstream relationships, McLaren (2000) and Grossman and Helpman (2002) take the incomplete contract approach and assume away contracts in their models. Naturally, they do not touch on the issue of imperfect enforcement. Their focus is a comparison of vertical integration with the market.<sup>3</sup>

In the present model, contracts are complete but enforcement is imperfect, similar to the approach taken by Anderson and Young (2003). Unlike us, however, they consider contracts on trade of ordinary goods between two parties from different countries and examine the implications of imperfect enforcement for international trade. We examine whether contracts are preferred to vertical integration as an organizational mode for CS development between two parties within the same country.

Second, the development of PS is similar to product innovation in that the sunk R&D costs are high, the marginal costs of production are low, and the rents legally accruing to the software developers or product innovators are easily appropriated without sufficient copyright protection. However, software development still differs considerably from innovation in many ways. For example, in product innovation, imitators are also producers who compete against the innovator in the market. In contrast, software is copied by consumers. As a result, the pricing strategy and equilibrium profit of the product innovator could be very different from those of the software developer. In a recent study, Chen and Png (2003) examine how a software publisher should optimally choose its price and degree of copyright enforcement.

---

<sup>3</sup>McLaren (2000) shows that there exist multiple equilibria about the production modes, but international openness makes the market (or outsourcing) more likely than vertical integration to be the equilibrium. Grossman and Helpman (2002) characterize the general equilibrium of industrial organization in the presence of downstream competition.

Our analysis of PS differs from theirs in many aspects. For example, in addition to pricing, we also consider the PS developer's entry decision in the presence of imperfect copyright protection.

Third, to our knowledge, this paper provides the first *general* equilibrium analysis of the software industry. Software programmers are employed in both product markets and therefore environment changes in one product market can affect the equilibrium in the other product market. A general equilibrium analysis allows us to examine cross-product effects. Such an approach is important for policy analysis. For example, stronger copyright protection in the PS market may result in the firms in the CS market to switch to vertical integration from type-H contracts due to the resulting higher costs of programmers. This calls for raising the level of contract enforcement in the CS market. Similarly, as demand for CS increases, the PS market may be closed down, again due to the resulting higher costs of programmers. Strengthening copyright protection would help reduce this negative externality.

Finally, the results obtained from this paper are testable and the model's prediction is consistent with many empirical observations. The U.S. is the dominant developer of PS in the world. In 1993, the U.S. producers of PS controlled about 60% of the world market (Siwiek and Furchtgott-Roth, 1993, p. 60). It is also clear that the U.S. is far ahead of the EU and Japan in promoting legal protection in software innovation.<sup>4</sup> The U.S. extended copyright protection to software in 1980 (the Software Amendment is based on the U.S. Copyright Act enacted in 1976), but it was not until 1991 that the European Commission first issued a directive concerning the application of copyright to software.

All other countries, including those in the EU, more or less concentrate on developing CS. For example, CS sales accounted for 84% of total Japanese software sales according to a 1990 survey by the MITI (Ministry of International Trade and Industry).<sup>5</sup> In-house development of CS is popular. According to another survey by MITI in 1988, 58% of software engineers in Japan worked for user companies (i.e., in-house development), and 35% worked for software houses. Among the software companies in Japan, user spin-off software houses accounted for 31.3% of market sales (representing semi-in-house development).<sup>6</sup> Most software firms in China, India and Latin America are reported to produce primarily CS (see Zhang and Wang, 1995, for China; Correa, 1996, p. 172, for India). As described by Zhang and Wang (1995, p. 66), in China, “[m]ost of the local software companies were generally limited to developing software for individual customers according to their specifications, on a customer-by-customer basis. Consequently, these software companies had hardly any market-oriented product of their own and acted only as subcontractors.”<sup>7</sup> Although India has a large pool of highly skilled computer

---

<sup>4</sup>This also leads Torrisi (1998, p. 156) to conclude that the differences in copyright protection, together with some other factors, “explain why US firms were the first to enter the market of PS, pre-empting the entrance of European producers.”

<sup>5</sup>See Baba et al. (1995, p. 478).

<sup>6</sup>See Baba et al. (1995).

<sup>7</sup>In China, “the first law to protect copyright, the ‘Copyright Law of the People’s Republic of China,’ came into effect in June 1991. Due to the inadequate legal framework, it is understandable that the

programmers, the country does not develop a significant amount of PS (see Arora and Asundi, 1999; Arora et al., 1999). This facts are consistent with the prediction of our analysis.

The rest of the paper is organized as follows. The model is set up in Section 2. In Section 3, we analyze the general equilibrium under various conditions of legal protection. Concluding remarks are presented in Section 4.

## 2. The model

There are  $M$  computer programmers, who can do all the work required for software development (e.g., system design, coding, testing, etc.). The reservation wage is normalized to 1, which a programmer can earn from a nonsoftware job. When engaging in software development, each programmer has to expend additional effort, denoted  $e$ . We describe the CS and PS development separately below.

### 2.1. Customized software and contract enforcement

There are  $N$  firms that are potential users of CS. Each firm needs one CS specifically designed for its own use. Software  $i$  ( $i = 1, \dots, N$ ) can generate a value equal to  $u$  for firm  $i$ , but has no value for firm  $j$  ( $\neq i$ ).<sup>8</sup> This is assumed in order to capture and emphasize the product specificity of CS, as opposed to the generic nature of PS. By choosing units properly, we can assume that one programmer is needed to develop software  $i$ . We consider and compare two organizational modes of obtaining software  $i$ . First, firm  $i$  can hire a programmer to develop it. This is referred to as in-house development or vertical integration. In this case, we call the firm a v-firm. A v-firm's net value from software development is  $\pi^v = u - w_v - v$ , where  $w_v$  is the wage paid to the programmer and  $v$  is the cost associated with vertical integration.<sup>9</sup>

Second, firm  $i$  can offer a contract to a programmer for developing software  $i$ . This is referred to as outsourcing or as a contractual relationship. In this case, we call the firm a c-firm. To emphasize the role of contract enforcement, we consider the simplest contracting environment.<sup>10</sup> Assume the programmer can either put in effort  $e$  to produce the software with value  $u$  or put in zero effort to produce a software

*(footnote continued)*

software market simply was not ready to emerge in the 1980s." (Zhang and Wang, 1995, p. 43). "With only a few software package products available in the market, most of the application systems were either developed in-house or ordered from outside on an individual customer-design basis." (Zhang and Wang, 1995, p. 48).

<sup>8</sup>We can consider that firm  $i$  earns a fixed amount of profit in its industry, but if it uses software  $i$ , its profit will increase by  $u$ .

<sup>9</sup>It has been well argued that vertical integration inevitably incurs many types of costs, such as monitoring costs. See Williamson (1971) and Stuckey and White (1993) for some discussions.

<sup>10</sup>See Whang (1992) for a useful discussion of a typical contract for CS development based on his study of five real-world cases. We ignore those details, since our focus is not on optimal contract design.

without any value. The c-firm can judge the quality of the software. A contract specifies a payment, denoted  $c$ , from the c-firm to the programmer after the delivery of the software with value  $u$ .<sup>11</sup> Assume no renegotiation.<sup>12</sup> Since payment is required only after the programmer has invested the irreversible effort  $e$  to develop the software, this contract inevitably leads to the hold-up problem. After the delivery of the software with value  $u$ , the c-firm may take one of two actions. First, it may honor the contract by paying the programmer the contracted amount  $c$ . Second, it may breach the contract by paying any amount less than  $c$ . We use a parameter  $\alpha \in [0, 1]$  to capture imperfect contract enforcement. That is, for any contract in default, there is probability  $\alpha$  that the contract will be enforced by a court. When the court enforces a default contract, the programmer receives  $c$  from the c-firm and the c-firm pays an additional penalty  $t > 0$ , which may include both monetary and moral losses to the society at large. When a default contract is not enforced by the court, the programmer receives nothing and the c-firm pays nothing. The two parameters,  $\alpha$  and  $t$ , jointly represent the degree of contract enforcement, given which, a c-firm's (expected) net value is given by

$$\pi^c = \begin{cases} u - c & \text{if it honors the contract,} \\ u - \alpha(c + t) & \text{if it breaches the contract.} \end{cases} \quad (1)$$

Hence, in contrast to settings of incomplete contracting, complete contracts are possible in the present model because of possible enforcement.<sup>13</sup> We focus on complete contracts, since we are concerned with the issue of legal enforcement and interested in examining its implications for the equilibrium organizational mode of CS development.

There is a third potential mode, in which firm  $i$  buys software  $i$  from the market. This is referred to as arm's length relationship or market. However, it is clear that this is not a viable mode in the present setup, since no programmer will invest his/her effort to produce a very specific product without any certainty of getting a reward. This mode may be considered as an extreme case of the contractual relationship, namely when contract enforcement is completely absent ( $\alpha = 0$ ).

## 2.2. *Packaged software*

There is only one PS and it has multiple potential users. For simplicity, assume that the PS user group is different from the CS user group and there is no overlap between the two groups. Also assume that the potential users of PS are evenly and

<sup>11</sup>There are various types of contracts in reality. As cited by Arora et al. (1999), fixed fee contracts account for 58% of all contracts. Our contract is a fixed fee contract.

<sup>12</sup>We could have considered other types of contracts and allowed renegotiation. But this would significantly complicate the analysis, and the qualitative results obtained from our simple contract are unlikely to change. See Whang (1992), Wang et al. (1997), and Png and Tao (2000) for some interesting analyzes of optimal (limited) contract design for CS development.

<sup>13</sup>Here we rely on legal enforcement to combat the holdup problem. In contrast, Klein et al. (1978) emphasize long-term contracts as the market enforcement mechanism.

continuously distributed in the interval  $[0, 1]$  in such a way that users with higher  $i$  derive lower utilities from using the software. Let  $v(i)$  denote user  $i$ 's utility level when the software's quality is equal to unity. Then  $v'(i) < 0$ . Furthermore, assume that  $v(1) = 0$ ,  $v''(i) < 0$ , and if the PS's quality is  $q$ , then user  $i$ 's utility is  $v(i)q$ .

There is only one PS developer. In the working paper version (Qiu, 2004), we consider the case of multiple PS developers racing to get the patent of PS production. The basic results are the same as in this paper and so we focus on a single PS developer case to simplify the analysis and exposition. The PS developer hires programmers to develop the software. The quality of the software is an increasing function of the number of programmers, denoted  $x$ , hired by the PS developer. For convenience, we treat  $x$  as a continuous variable. Assume the following properties for the quality function:  $q(x) \geq 0$ ,  $q'(x) > 0$ , and  $q''(x) < 0$ . Without loss of generality (and realism), we can assume that the marginal cost of production (after the first copy) is zero.

The PS developer incurs two types of irreversible investment costs. First, it needs to spend a fixed amount on physical capital, such as computer hardware.<sup>14</sup> This cost is assumed equal to  $f$ . Second, it needs to pay the programmers for developing the software. Let  $w_p$  denote the wage rate paid by the PS developer to its programmers. Software development and production exhibits very strong increasing returns to scale.

### 2.3. *Sequence of moves*

There are three stages. At the first stage, each of the  $N$  firms determines its organizational mode of CS development. The PS developer determines whether to make the fixed investment to develop the software. At the second stage, each c-firm sets the terms of its contract and offers it to a programmer, each v-firm hires a programmer, and the PS developer hires a desired number of programmers. At the final stage, each of the  $N$  firms receives its CS. The PS developer sets the price for its products and sells them in the market. Wage rates are endogenously determined in the whole process. All parties are assumed to be risk neutral.

### 2.4. *Wage rate*

In this model, there are a large number of firms plus the PS developer to hire programmers and there are a large number of programmers waiting to be employed. Accordingly, we assume that wage rates are determined by the total demand and supply of labor. That is, parties from both sides of the labor market are price takers. Although in reality the skills for CS development and PS development could be somewhat different, we assume that programmers can move freely (without adjustment costs) between CS and PS labor markets. The implication is that programmers will get equal wage rates in equilibrium in both CS and PS employment. Hence, we can set  $w_v = w_p = w$ . We will treat  $w$  as an exogenous

<sup>14</sup>For more discussion on this type of costs see Siwek and Furchtgott-Roth (1993) and Torrisi (1998).

parameter first (in Sections 3.1 and 3.2) and later (in Section 3.3) derive the equilibrium  $w$ . At the end of this paper, it should be clear that none of these assumptions are crucial for the main results of the paper. What is crucial is the assumption that programmers can move (with or without a cost) between the PS and CS sectors. This assumption is realistic because similar skills are required in both PS and CS development. This movement provides a link between these two sectors although wage rates may not be exactly identical.

### 3. A general equilibrium analysis

#### 3.1. Customized software

Since the  $N$  firms are not related in any respect except that they rely on the same labor pool to develop software, we can analyze each firm's decision independently for any given  $w \geq w_0$ , where  $w_0 \equiv 1 + e$ . As all firms are identical, we need to focus on just one firm.

##### 3.1.1. Contracts

It is useful to define three critical values that are important in determining whether a contract will be honored or breached:

$$m_+ \equiv \alpha t / (1 - \alpha), \quad m_- \equiv u / \alpha - t \quad \text{and} \quad m_0 \equiv u / (u + t).$$

Note that an increase in  $m_+$ , or a decrease in  $m_-$ , implies that contract enforcement gets stronger. Moreover, the following three conditions are equivalent:  $m_+ \geq u$ ,  $m_- \leq u$ , and  $\alpha \geq m_0$ , all indicating stronger enforcement.

As defined in the introductory section, we call a contract a type-H contract if it will be honored by the c-firm, and a contract a type-B contract if it will be breached by the c-firm. The firm never chooses outsourcing unless  $\pi^c \geq 0$ , which holds if  $c \leq u$  for a type-H contract, and if  $c \leq m_-$  for a type-B contract. Suppose  $\pi^c \geq 0$  in the following analysis.

Eq. (1) immediately implies that the c-firm prefers honoring the contract to breaching the contract if and only if  $c \leq m_+$ . As a result: (i) if  $\alpha \geq m_0$ , then the c-firm always honors the contract, and (ii) if  $\alpha < m_0$ , then the c-firm honors the contract when  $c \leq m_+$  and breaches it when  $m_+ < c < m_-$ . Hence, a programmer can correctly predict whether or not a contract will be honored. The next question is whether the c-firm should offer a type-H or type-B contract and how to determine the contract term  $c$ . On the one hand, the c-firm should set  $c$  as low as possible so long as the contract is acceptable to a programmer. On the other hand, in order to induce a programmer to accept the contract, the c-firm has to make it at least as attractive as the programmer's outside option, i.e., receiving  $w$  for being hired by a v-firm or the PS developer. Note that by accepting the c-firm's contract, the programmer's expected net return is  $c - e$  for a type-H contract, but  $\alpha c - e$  for a type-B contract. Hence, the optimal term is  $c = w$  for a type-H contract (provided  $w \leq u$ ) and  $c = w / \alpha$  for a type-B contract (provided  $w \leq u - \alpha t$ ).

When it is certain or almost certain that a contract will be breached ( $\alpha = 0$  or  $\alpha \approx 0$ ), no programmer will accept the contract. To see this, note that in this case both the price of the contract ( $w/\alpha$ ) and the penalty in the case of contract enforcement  $[(w/\alpha) + t]$  are infinitely large, which becomes unrealistic. Accordingly, we rule out this possibility by assuming that there exists  $\alpha_{\min} > 0$  such that no contract can be made for  $\alpha < \alpha_{\min}$ .

Lemma 1 below describes the optimal contract at a given wage rate.

**Lemma 1.** *Suppose  $w \in [w_0, u)$ . Then for any given  $\alpha \geq \alpha_{\min}$  and  $t$ ,*

- (i) *if  $\alpha \geq m_0$ , then the c-firm offers a type-H contract with  $c = w$ ;*
- (ii) *if  $\alpha < m_0$ , then the c-firm offers a type-H contract with  $c = w$  for  $w \leq m_+$ , it offers no contract for  $w > \max\{m_+, \alpha m_-\}$ , and it offers a type-B contract with  $c = w/\alpha$  if and only if  $m_+ < \alpha m_-$  and  $w \in (m_+, \alpha m_-]$ .*

**Proof.** See the appendix.

The intuition behind the above results is easy to understand by noting that the conditions indicate various degrees of contract enforcement. There are three messages contained in Lemma 1. First, an increase in  $\alpha$  raises the occurrence of type-H contracts and reduces that of type-B contracts or no contract. Second, in the case of weak contract enforcement ( $\alpha < m_0$ ), as the wage increases, the optimal contract switches from type-H to type-B and then to no contract. Finally, and most importantly, although imperfect contract enforcement may tempt the c-firm to breach its contract, this may actually hurt the firm. To see this, note that since the programmer has an outside option and can infer the type of the contract, he discounts the value of any type-B contract. As a result, if the c-firm is to offer a type-B contract,  $c$  must be sufficiently attractive, equal to  $w/\alpha$  as opposed to  $w$  for a type-H contract.<sup>15</sup> Consequently, a type-B contract results in the expected value  $\pi_b^c \equiv u - w - \alpha t$ , lower than  $\pi_h^c \equiv u - w$ , which is the value of a type-H contract. When enforcement is weak (i.e.,  $\alpha < m_0$ ) and  $w \in (m_+, \alpha m_-]$ , should the c-firm have been able to commit not to breach the contract, it would have been able to make a type-H contract and so obtain  $\pi_h^c$ . However, without the ability to commit, the c-firm is forced to offer a type-B contract (Lemma 1(ii)). This is the negative externality existing in the environment with weak contract enforcement.

### 3.1.2. Optimal mode of organization

If the firm chooses in-house development, then given  $w \in [w_0, u - v]$ , it hires a programmer and receives  $\pi^v$ . Comparing  $\pi^v$  to  $\pi^c$ , we immediately obtain Lemma 2, which describes the optimal choice of organizational mode.

<sup>15</sup>This result, due to legal risk, is consistent with that observed by Whang (1992, p. 310) in contracts subject to technical and financial risks. Whang points out that in anticipation of the risks, the developer will demand a high premium for the contract for compensation.

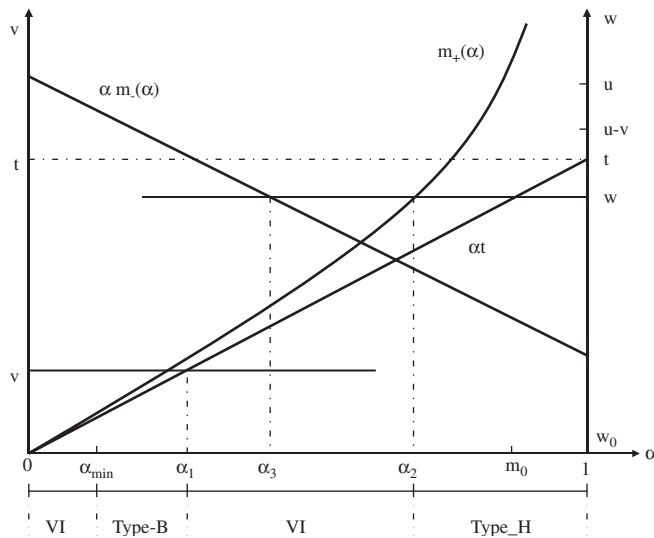


Fig. 1. Customized software.

**Lemma 2.** *A firm chooses vertical integration when contract enforcement is weak, or wage rate is high, or cost of vertical integration is low. More precisely, supposing  $w < u - v$ , then a necessary and sufficient condition for vertical integration is*

- (i)  $\alpha < \alpha_{\min}$ , or
- (ii)  $\alpha \in [\alpha_{\min}, m_0)$  and  $w > \max\{m_+, \alpha m_-\}$ , or
- (iii)  $\alpha \in [\alpha_{\min}, m_0)$ ,  $m_+ < \alpha m_-$ ,  $w \in (m_+, \alpha m_-]$  and  $v < \alpha t$ .

Lemma 2 can be stated in a different way: contracts are preferred to vertical integration when enforcement is very strong (more precisely,  $\alpha \geq m_0$ ), or when it is not so strong but the wage rate is low ( $w \in [w_0, m_+]$ ).

Because of the extra cost associated with vertical integration, contracts are more efficient. However, contracts are not always adopted due to imperfect enforcement, which actually imposes a cost on contracting. This cost does not exist for type-H contracts. As indicated by Lemmas 1 and 2, whenever a type-H contract dominates, vertical integration is never optimal.<sup>16</sup>

### 3.1.3. Contract enforcement and optimal mode of organization

To analyze fully the effects of contract enforcement, we first depict recurring Lemmas 1 and 2 graphically in Fig. 1. We measure  $\alpha$  using the horizontal axis,  $v$  using the left-hand-side vertical axis, and  $w$  using the right-hand-side vertical axis.

<sup>16</sup>Lemma 2 also predicts that vertical integration and contracts may coexist only in a very special case, viz., when  $v = \alpha t$ . However, it is not difficult to see that if  $v$  or  $u$  is not homogeneous across the firms, then in many cases vertical integration and contracts coexist: vertical integration for some firms and contracts for the others.

We focus on  $v \in (0, t]$ , since when  $v > t$ , contracts always dominate vertical integration regardless of the level of  $\alpha$  (provided  $\alpha \geq \alpha_{\min}$ ). The curves  $m_+(\alpha)$  and  $\alpha m_-(\alpha)$  are plotted against the right vertical axis, while the straight line  $\alpha t$  is plotted against the left vertical axis. Two critical points are automatically defined:  $\alpha_1 \equiv v/t$ , at which the two straight lines  $v$  and  $\alpha t$  intersect, and  $\alpha_2 \equiv w/(w+t)$ , at which the two curves  $w$  and  $m_+(\alpha)$  intersect. Using Lemmas 1 and 2, together with Fig. 1, we prove a general pattern of the optimal organizational mode in Proposition 1.

**Proposition 1.** *As contract enforcement becomes stronger, the optimal mode of organization switches from vertical integration to type-B contract, then to vertical integration, and finally to type-H contract.*

*Specifically, given  $v \leq t$  and  $w \in [w_0, u - v)$ , there exist unique  $\alpha_1$  and  $\alpha_2$  such that the optimal mode of organization for CS development is vertical integration when  $\alpha \in [0, \alpha_{\min})$ , a type-B contract when  $\alpha \in [\alpha_{\min}, \min\{\alpha_1, \alpha_2\}]$ , vertical integration when  $\alpha \in (\min\{\alpha_1, \alpha_2\}, \alpha_2)$ , and a type-H contract when  $\alpha \in [\alpha_2, 1]$ .*

**Proof.** See the appendix.

It is possible that  $\alpha_1 < \alpha_{\min}$ , in which case  $[\alpha_{\min}, \alpha_1] = \emptyset$ , i.e., there exists no type-B contract for all levels of  $\alpha$ . It is also possible that  $\alpha_1 \geq \alpha_2$ , in which case  $(\min\{\alpha_1, \alpha_2\}, \alpha_2) = \emptyset$ , i.e., there is no vertical integration for intermediate values of  $\alpha$ . But in any case, the optimal mode of organization is always vertical integration for sufficiently small  $\alpha$  ( $\leq \alpha_{\min}$ ) and always a type-H contract for sufficiently large  $\alpha$  ( $\alpha \geq \alpha_2$ ). One may ask why, as  $\alpha$  decreases, the optimal mode switches from vertical integration to a type-B contract. The reason is that as  $\alpha$  decreases, the cost associated with vertical integration ( $v$ ) becomes higher than the cost associated with a type-B contract ( $\alpha t$ ).

Given Proposition 1, we can easily examine the effects of changing  $v$  and  $w$  on the optimal mode of organization. From Fig. 1, an increase in  $v$  shifts  $\alpha_1$  to the right. As a result, the range of  $\alpha$  for a type-B contract expands and that for vertical integration shrinks. As  $v$  continues to increase, the interval for vertical integration ( $\alpha_1, \alpha_2$ ) eventually vanishes. The intuition is quite clear: an increase in the cost of vertical integration makes vertical integration less attractive.

An increase in  $w$  shifts  $\alpha_2$  to the right in Fig. 1. This reduces the range of  $\alpha$  for type-H contracts, to the benefit of vertical integration. To understand this, simply consider the point at  $\alpha = \alpha_2$ . An increase in  $w$  reverses the inequality  $w \leq m_+$  at this point and so the c-firm can no longer commit to a type-H contract. As explained before (after Lemma 1), the negative externality forces the c-firm to offer a type-B contract or no contract, reducing the c-firm's value. Thus, at  $\alpha = \alpha_2$ , the optimal choice switches from a type-H contract to vertical integration as a result of wage increases.

From the above analysis, we summarize how the optimal mode of organization responds to changes in the cost of vertical integration and wage rates.

**Corollary 1.** *Suppose  $v \leq t$  and  $w \in [w_0, u)$ . An increase in  $v$  reduces the case for vertical integration and raises the case for type-B contracts. An increase in  $w$  reduces the case for type-H contracts and raises the case for vertical integration or type-B contracts.*

**Proof.** See the appendix.

### 3.2. *Packaged software*

We now analyze the PS developer's decision at the second-stage and the third-stage. The analysis of entry at the first stage is postponed to Section 3.3.

#### 3.2.1. *Users' behavior, copyright protection and demand for PS*

Due to piracy, demand for PS is very different from demand for ordinary commodities. Suppose the PS's quality is  $q$  and the price is  $p$ . Then, if a potential user  $i$  buys the software, he will receive a net benefit equal to  $v(i)q - p$ . However, the software can be copied at zero cost. When using the pirated software, the probability of being caught is  $\mu \in (0, 1)$  and the penalty is equal to  $p$ .<sup>17</sup> Hence, the net expected benefit to user  $i$  who uses the pirated software is  $(1 - \mu)v(i)q - \mu p$ . Government agents are responsible for detecting piracy and collecting the fines, which become part of the government's revenue. A person's gross and net benefit is zero if he does not use the software. The degree of copyright protection is fully captured by the magnitude of  $\mu$ . It is clear that in addition to its generic use, piracy is another distinguishing feature of PS, as opposed to CS.

We are now ready to examine the users' decisions on buying, copying and not using PS. The decision is affected by the price-adjusted value  $v(i)q/p$  and copyright protection  $\mu$ . First, for user  $i$ , buying is preferred to copying, if and only if

$$v(i)q/p \geq (1 - \mu)/\mu. \quad (2)$$

Second, buying is preferred to not using if and only if  $v(i)q/p \geq 1$ . Finally, copying is preferred to not using, if and only if

$$v(i)q/p \geq \mu/(1 - \mu). \quad (3)$$

Combining (2) and (3) yields a necessary condition for the existence of copying:  $\mu < \frac{1}{2}$ . Lemma 3 below gives the conditions under which copying exists.

**Lemma 3.** *There is no copying if  $\mu \geq \frac{1}{2}$ . For  $\mu < \frac{1}{2}$ , copying occurs if and only if there exists some  $i$  such that the price-adjusted value satisfies condition (3).*

When  $\mu \geq \frac{1}{2}$ , a further increase in  $\mu$  will not have any effect on the users' behavior. Without loss of generality, in the rest of the paper, we confine the analysis to  $\mu \leq \frac{1}{2}$ , letting  $\mu = \frac{1}{2}$  capture the case for all  $\mu \geq \frac{1}{2}$ .<sup>18</sup> We first consider demand for legitimate software at any given  $\mu \leq \frac{1}{2}$  and  $q$ . Let  $Q \in (0, 1)$  denote the marginal user who is

<sup>17</sup>According to Gilman (1992), the Software Publishers Association (SPA) conducts audits, and if illegal software is found, then it "charges a fine equal to the price of the product and then destroys the copy." Alternatively, we could have considered a more general case where the fine may be different from the price, in which case the combination of  $\mu$  and the fine would represent the degree of copyright protection.

<sup>18</sup>This critical point,  $\frac{1}{2}$ , of course depends on many factors, including risk neutrality, equal quality of legitimate software and pirated software and the magnitude of the fines. However, the present study does not emphasize the numerical value of this point, but only its existence. The qualitative results do not depend on whether this value is  $\frac{1}{2}$  or  $\frac{3}{4}$ .

indifferent between buying and copying the software, i.e.,  $v(Q)q/p = (1 - \mu)/\mu$ , which defines the demand function  $Q(p; q, \mu)$ . Users with  $i \leq Q$  purchase the software, and others either copy or do not use it. Using this definition and the properties of the value function  $v(i)$ , we can derive the properties of the demand function:  $\partial Q/\partial p < 0$ ,  $\partial Q/\partial q > 0$ , and  $\partial Q/\partial \mu > 0$ .

We now turn to examining copying. Let  $Q^c \in [0, 1]$  denote the marginal user who is indifferent between copying and not using the software, i.e.,  $v(Q^c) = \mu/(1 - \mu)$ , by (3), which defines a function  $Q^c(p; q, \mu)$  with the following properties:  $\partial Q^c/\partial p < 0$ ,  $\partial Q^c/\partial q > 0$  and  $\partial Q^c/\partial \mu < 0$ . Recall from (2) and (3) that users with the price-adjusted values greater than or equal to  $\mu/(1 - \mu)$  but less than  $(1 - \mu)/\mu$  will use pirated software. Hence, users  $i \in (Q, Q^c]$  use pirated software. People with  $i > Q^c$  do not use the software (legitimate or pirated).

3.2.2. PS developer’s decision on quality and price

We shall work backward by first looking at the third-stage pricing decision. Suppose the PS developer has hired  $x$  programmers at stage two. Then, it faces the demand  $Q(p; q(x), \mu)$ . Thus, the PS developer’s operating profit at the third stage is  $Q(p; q(x), \mu)p$ . It chooses the price level  $p$  to maximize this profit. At stage two, given wage rate  $w$ , the PS developer chooses its hiring level  $x$  to maximize its expected profit. As a result, the PS developer’s profit at the second stage is given by  $Q(p; q(x), \mu)p - wx$ . Because the PS developer has a monopoly, it is easy to see that the above sequential decisions are equivalent to a simultaneous decision on  $x$  and  $p$ , which is obtained from the following maximization problem:

$$\max_{(p,x)} \pi(p, x; w, \mu) \equiv Q(p; q(x), \mu)p - wx. \tag{4}$$

There is a unique interior solution (see proof of Proposition 2 in the Appendix), denoted  $x^*(w, \mu)$  and  $p^*(w, \mu)$ , that satisfies the following first-order conditions:

$$\begin{aligned} \pi_x &= Q_q q'(x^*)p^* - w = 0, \\ \pi_p &= [Q_p p^* + Q(p^*; q(x^*), \mu)] = 0, \end{aligned} \tag{5}$$

where and hereafter the subscripts of  $Q$  represent derivatives with respect to the corresponding variables. Note that the wage rate affects the equilibrium consumption through quality and price:

$$\partial Q^*/\partial w = Q_q q'(x^*)(\partial x^*/\partial w) + Q_p(\partial p^*/\partial w). \tag{6}$$

**Proposition 2.** *Given  $\mu$ , an increase in wage rate (i) reduces employment in the PS development, (ii) reduces the PS’s quality, and (iii) lowers the PS’s price. However, the final consumption for legitimate PS increases (is unchanged, decreases) if the quality is higher than (equal to, lower than) unity. More precisely,*

$$\frac{\partial x^*}{\partial w} < 0, \quad \frac{\partial q^*}{\partial w} < 0, \quad \frac{\partial p^*}{\partial w} < 0 \quad \text{and} \quad \frac{\partial Q^*}{\partial w} \begin{cases} < 0 & \text{if } q^* < 1, \\ = 0 & \text{if } q^* = 1, \\ > 0 & \text{if } q^* > 1. \end{cases} \tag{7}$$

**Proof.** See the appendix.

The first three results of the proposition are easy to understand. As the cost of hiring programmers rises, the PS developer has to reduce its employment ( $x^*$ ). As a result, the quality of the software decreases. Lower quality also results in lower demand for the product, and so the PS developer’s optimal price is reduced accordingly. However, the effect on consumption  $Q^*$  is less clear cut. There are two competing forces as shown in (6) where the first term is negative and the second term is positive. On the one hand, for the same price level, lower quality reduces demand and so causes  $Q^*$  to decrease. On the other hand, for a given quality, the lower price generates higher demand and so causes  $Q^*$  to increase. This result is interesting in that it conveys a message that better quality software may or may not be more popular. Since the PS developer charges a higher price for high-quality software than for low-quality software, the resulting consumption for the high-quality software may end up lower than that for the low-quality software. The proposition shows that whether  $Q^*$  increases or decreases depends on the level of the quality in equilibrium.

### 3.3. Equilibrium

In this section, we analyze the equilibrium wage rate  $w^*$ . Let us first examine the demand for programmers in CS development. So long as  $w < u$ , all firms strictly benefit from using their respective CS, regardless of the modes of organization adopted. Therefore, demand for programmers is equal to  $N$ . At  $w = u$ , since the only viable mode of organization is a type-H contract and the firms are indifferent between using and not using CS, demand for programmers is between 0 and  $N$ . Obviously, there is no demand for programmers when  $w > u$ . Such a demand curve (formed by three straight lines) is depicted on the left panel of Fig. 2.

We now examine the PS development and depict the results in the right panel of Fig. 2. First, we derive the supply curve  $S$ . The supply of programmers is zero for wage rates below  $w_0$ . At  $w = w_0$ , the maximum number of programmers left over for

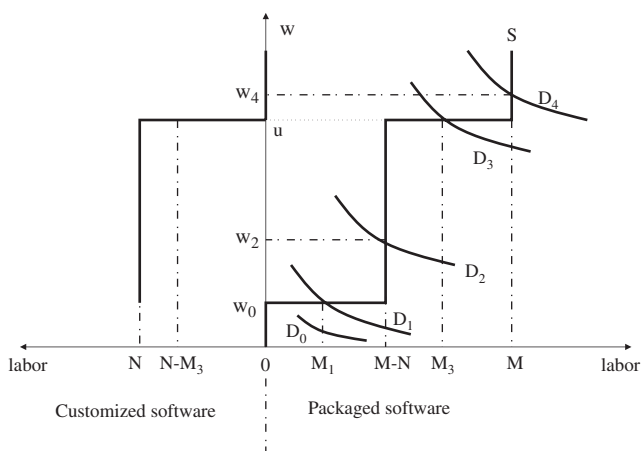


Fig. 2. Labor market equilibrium.

PS is equal to  $M - N$ . Since the programmers are indifferent between being employed and not at this wage rate, the supply is between 0 and  $M - N$ . For  $w_0 < w < u$ , all programmers want to be employed and so the supply is equal to the number of programmers left over from CS, i.e.,  $M - N$ . At  $w = u$ , the supply is between  $M - N$  and  $M$ . For  $w > u$ , the supply equals  $M$ .

Second, we turn to the demand for programmers by the PS developer. The total demand, denoted  $D(w; \mu)$ , is equal to  $x^*(w, \mu)$ , which is decreasing in  $w$ , as shown by (7). Without more specifications of the model, we cannot exactly pin down the position of the demand curve in Fig. 2. However, we can draw five possible representative demand curves, labeled  $D_i$ ,  $i = 0, \dots, 4$ . When demand is very low ( $D_0$ ), we have  $x^* = 0$  for  $w \geq w_0$ . In this case, PS does not exist in the economy. The equilibrium wage rate, denoted  $w^*$ , is totally determined by CS and so  $w^* = w_0$ . At a higher level of demand ( $D_1$ ), PS emerges, but since demand is not sufficiently strong, the wage rate remains the same and employment in CS and in PS development are  $N$  and  $M_1$ , respectively. As demand continues to increase, the wage rate eventually starts to rise, and it continues to rise (e.g.,  $w^* = w_2$  when demand is captured by  $D_2$ ) until it is equal to  $u$ . Within this range of demand,  $w^* \in [w_0, u]$  and employment in CS and in PS are  $N$  and  $M - N$ , respectively. As demand becomes higher, at  $D_3$ , we have  $w^* = u$  and employment in the PS development reaches  $M_3$ , leaving the remaining  $M - M_3$  for CS. As demand increases further, the equilibrium wage rate remains equal to  $u$  before the employment in PS absorbs all  $M$  programmers. After this point, the wage rate has to go up, since the labor supply has reached its maximum. At a demand level such as  $D_4$ , the equilibrium wage ( $w^* = w_4$ ) is totally determined by PS, and CS does not exist.

The above graphical analysis clearly indicates the importance of demand  $D(w; \mu)$  in affecting the equilibrium wage rate and employment. We next show how copyright protection affects such demand.

Using the equilibrium values in  $\pi$  given in (4), we obtain the expected profit as a function of  $\mu$ :  $\pi^*(\mu)$ . On the one hand, an increase in  $\mu$  results in higher demand for PS, and so the expected profit is also higher. On the other hand, this higher demand for PS translates to higher demand for programmers. Wage rates may or may not rise, depending on the actual position of  $D$ . In any case, this wage effect is secondary, and we can show (see the appendix for a proof) that

$$d\pi^*(\mu)/d\mu > 0. \quad (8)$$

We are now ready to analyze the PS developer's entry decision. Given  $\mu$ , entry occurs if and only if  $\pi^*(\mu) - f \geq 0$ . Denote  $\bar{f} \equiv \pi^*(\frac{1}{2})$ , and assume that  $f < \bar{f}$ . That is, the fixed cost is not too large to deter entry when copyright protection is very strong.

**Proposition 3.** *Suppose  $f < \bar{f}$ . Then there exists a unique  $\mu_0 \in (0, \frac{1}{2})$  such that a country produces PS if and only if  $\mu \geq \mu_0$ .*

**Proof.** See the appendix.

The intuition behind the above result is very simple. Without sufficient copyright protection, demand for PS is not guaranteed, and so the PS developer will not pay

the fixed entry cost to enter the nonprotected industry. Although there are increasing returns to scale, weak copyright protection limits the scale and thus discourages entry.

### 3.4. Cross-product effects

Our model is a general equilibrium one in the sense that the two types of software are linked via the labor market and programmers can move freely between the two labor markets. Wage rate is determined jointly by the total labor supply and the labor demand from both markets. Hence, it is natural to expect that exogenous changes from one product market can transmit to the other product market via their effects on the labor market. The above equilibrium analysis and results allow us to conduct such a study easily. The cross-product analysis also indicates important policy implications.

Let us first explore how changes in copyright protection affect the organizational mode of CS production. In light of Propositions 1 and 3, we focus on the most interesting equilibrium situation where  $w^* \in (w_0, u - v)$  and the case of medium degree of copyright protection with  $\mu \geq \mu_0$ . That is, PS exists in the market and programmers are employed in both CS and PS production. Given  $\mu$  and  $\alpha$ , there will be a corresponding equilibrium production mode of CS. We ask how the equilibrium production mode would differ should the copyright protection be stronger. To answer this question, we suppose  $\mu$  increases and just redo the analysis conducted before. First, we have known that  $\partial Q / \partial \mu > 0$  from Section 3.2. That is, stronger copyright protection induces higher demand for PS. But will higher demand for PS translate to higher demand for labor? We need to check the PS developer's optimal decisions on  $p$  and  $x$ , determined by the first-order conditions (5), in response to such a change. Differentiating (5) with respect to  $\mu$  and then using Cramer's rule, we obtain  $\partial x^* / \partial \mu = \pi_{xp} Q_\mu / \Delta > 0$  because  $\Delta > 0$ ,  $\pi_{xp} > 0$  and  $Q_\mu > 0$  as shown in the proof of Proposition 2 (in the appendix).

From Fig. 2, we can see that an increase in labor demand by the PS developer ( $x^*(w)$  is just the demand function  $D_2(w)$ ) strictly raises the equilibrium wage rate. Then from Fig. 1 and Lemma 2, we know that the critical point between vertical integration and type-H contract  $\alpha_2$  increases in response to a higher wage rate. That is, the range of  $\alpha$  for having contracts in CS production is reduced and the range of  $\alpha$  for using vertical integration as the production mode in CS is enlarged. In particular, if  $\alpha$  is slightly above  $\alpha_2$  before  $\mu$  increases, then an increase in  $\mu$  results in a switch of CS production mode from contract to vertical integration.

Therefore, a stronger copyright protection not only affects the PS production (higher quality, higher price, and higher employment), but also the organizational mode of CS production. The cross-product transmission effect comes from the changes of wage rate for programmers. As shown in Proposition 1 and explained after that proposition, for some range of contract enforcement, an increase in wage rate makes the firms to switch from offering a type-H contract to using vertical integration. This is a negative spillover in the sense that type-H contract is socially preferred to vertical integration because the latter is associated with additional

organizational costs. There are two policy options to reduce this negative spillover. One is to raise the level of contract enforcement so that  $\alpha$  is above the new critical level  $\alpha_2$  and hence the firms still offer type-H contracts. Another option is to start importing programmers (i.e., skillful IT talent) from overseas, as many countries have been doing. This policy could prevent wage rate from going up and so the negative spillover will not exist.

The same approach can be adopted to examine the transmission effects from the CS market to the PS market. In the present model, changes in contract enforcement do not affect wage rate and so will have no transmission effect on PS. However, other cases can be investigated. Suppose  $N$  increases, i.e., demand for CS increases. From Fig. 2, we can see that the labor supply curve in the PS market shifts to the left, resulting in a higher wage rate ( $w_2$ ). Proposition 2 indicates that both the prices and quality of PS will drop. In some more drastic cases, a large increase in wage rate or if the PS developer just slightly prefers entering to not entering in the previous equilibrium situation, the PS will respond by reversing its entry decision, from entry to no entry. This can be easily seen since the PS's profit decreases in wage rate (applying the Envelop Theorem to problem (4)). Again, an increase in demand in the CS market generates a negative externality to the PS market. This effect could be so strong to lead to a close down of the PS market. This calls for policy changes in order to keep both markets active. For example, the government can further improve copyright protection or start importing programmers. Both policies help restore the profitability of the PS industry.

#### 4. Conclusion

We have developed a model to analyze software development. We emphasize the effects of contract enforcement on the organizational mode of CS development and the effects of copyright protection on the pattern of both PS and CS production. We also stress the importance of a general equilibrium analysis in examining cross-product effects.

Software has several distinguishing features compared with common commodities like automobiles and therefore deserves special attention. In this paper, we have emphasized the product specificity of CS, which results in holdups, and the low (or zero) marginal cost of reproducing PS, which leads to piracy. Because of these features and the resulting problems, legal environments are very crucial in shaping this industry's development.

As an extension to check the robustness of the results obtained in this paper and to derive new results, we could introduce some other features of software into the present model. One example is the network externality in the PS. When using the software, a user's utility,  $v(i)$ , depends on the number of people who are also using it (legally or illegally). This externality will affect the PS developer's pricing strategy in an interesting way. Inevitably, the investment and entry decision will also be affected. However, it is conceivable that the results obtained in this paper about the PS are unlikely to be altered qualitatively.

Another direction of extensions is to explore more closely the copyright spillover effect from PS to CS. In the present model, the cross-product effect is realized through changes in the wage rate. We could also consider the case when productivity in CS development may be enhanced by the existence and the quality of PS.

The model can be easily extended to analyze international trade of software. Qiu (2004) has examined how copyright protection becomes a country's comparative advantage in producing and exporting PS.

### Acknowledgements

I have benefited from comments by Leonard Cheng and Brian Copeland and discussions with Son Ku Kim, Zhigang Tao, Susheng Wang, Oliver Williamson, two referees, Esther Gal-Or (Editor) and seminar participants at HKUST, the "International Economics and Asia" Workshop held at the City University of Hong Kong, the Second Annual Conference of the European Trade Study Group held in Glasgow and the Mid-west International Economics Meeting held at the University of Wisconsin (Madison). I am grateful for financial support by a grant from HKUST (DAG03/04.BM28).

### Appendix

**Proof of Lemma 1.** The proof for Lemma 1(i) is straightforward. For (ii), we can easily check that a type-H contract with  $c = w$  when  $w \leq m_+$  always dominates a type-B contract regardless of the ranking of  $m_+$  and  $\alpha m_-$ . Now, consider  $w > m_+$ , in which case breaching the contract is always preferred to honoring the contract from the c-firm's point of view. Hence, we simply need to consider if a type-B contract is profitable to the c-firm and acceptable by the programmers.

Note that the net value of a type-B contract is  $\pi_b^c = u - \alpha(w/\alpha + t) > (=, <) u - \alpha m_- - \alpha t = 0$  for  $w < (=, >) \alpha m_-$ . Hence,  $\pi_b^c > 0$  iff  $m_+ < \alpha m_-$  and  $w \in (m_+, \alpha m_-]$ . Summarizing the above analysis yields the result in Lemma 1(ii).

**Proof of Proposition 1.** We refer to Fig. 1 for visual help, but the proof does not depend on the special case drawn in the figure. Given  $v \leq t$  and  $w \in [w_0, u - v)$ ,  $\alpha_1$  and  $\alpha_2$  have been defined in the main text, and now we define  $\alpha_3 \equiv \min\{(u - w)/t, 1\}$ , at which the two curves  $w$  and  $\alpha m_-(\alpha)$  intersect.

Let us first check when vertical integration dominates contracts using Lemma 2. Obviously, this is the case when  $\alpha < \alpha_{\min}$ . Note that  $w > m_+ \iff \alpha < \alpha_2$ ,  $w < \alpha m_- \iff \alpha < \alpha_3$ , and  $v < \alpha t \iff \alpha > \alpha_1$ . Thus, by Lemma 2(ii), vertical integration is optimal for  $\alpha \in (\alpha_1, \alpha_3]$ , and by Lemma 2(iii), it is optimal for  $\alpha \in (\alpha_3, \alpha_2)$ . If  $w$  is lower (than that shown in the figure) so that  $\alpha_3 \geq \alpha_2$ , then Lemma 2(ii) does not apply and by Lemma 2(iii), vertical integration is optimal for  $\alpha \in (\alpha_1, \alpha_2]$  provided  $\alpha_1 < \alpha_2$ . If  $v$  is sufficiently high or  $w$  is sufficiently low, it is possible that  $\alpha_1 \geq \alpha_2$ , in which case there will be no vertical integration for  $\alpha \geq \alpha_{\min}$ .

The case for contracts is easily proved.

**Proof of Corollary 1.** The proof is straightforward by using Proposition 1 and Fig. 1 with the following observations: (1)  $\partial\alpha_1/\partial v \geq 0$ , but  $\partial\alpha_2/\partial v = 0$ , and (2)  $\partial\alpha_2/\partial w \geq 0$  but  $\partial\alpha_1/\partial w = 0$ .

**Proof of Proposition 2.** Let us first derive the properties of the demand:

$$\left\{ \begin{array}{l} Q(0; q, \mu) = 1, \quad Q(v(0)q\mu/(1 - \mu); q, \mu) = 0, \quad Q_\mu = -p/\mu^2qv' > 0, \\ Q_{\mu p} = Q_{p\mu} = -[(v')^2\mu q - (1 - \mu)pv'']/(v')^3\mu^3q^2 > 0, \\ Q_{\mu q} = Q_{q\mu} = p[(v')^2 - vv'']/(v')^3\mu^3q^2 < 0, \\ Q_p = (1 - \mu)/\mu qv' < 0, \quad Q_{pp} = -v''(1 - \mu)^2/(v')^3q^2\mu^2 < 0, \\ Q_q = -v/qv' > 0, \quad Q_{qq} = v[2(v')^2 - vv'']/(v')^3q^2 < 0, \\ Q_{pq} = Q_{qp} = -[(v')^2 - vv''](1 - \mu)/\mu(v')^3q^2 > 0. \end{array} \right. \tag{A.0}$$

Calculating the second-order derivatives of  $\pi$  given in (4) yields

$$\pi_{xx} = p[Q_q q''(x) + (q'(x))^2 Q_{qq}] < 0, \quad \pi_{pp} = (2Q_p + pQ_{pp}) < 0, \tag{A.1}$$

$$\pi_{xp} = \pi_{px} = q'(x)(Q_q + pQ_{qp}) > 0. \tag{A.2}$$

Then we use (A.0), (A.1), and (A.2) to define and obtain  $\Delta \equiv \pi_{xx}\pi_{pp} - \pi_{xp}\pi_{px} = \Delta_1/\mu^2q^4(v')^6$ , where  $\Delta_1 \equiv (1 - \mu)p\{-v''(v')^2vqq'' + v(q')^2[2(v')^2 - vv'']\}[2(v')^2\mu q - (1 - \mu)pv''] - (q')^2\{(v')^2\mu vq + (1 - \mu)p[(v')^2 - vv'']\}^2$ . Noting that by definition

$$v(Q)q = (1 - \mu)/\mu p, \tag{A.3}$$

and so using  $\mu vq$  to substitute  $(1 - \mu)p$  in  $\Delta_1$ , we obtain  $\Delta_1 = -(\mu vqv')^2 qq'' [2(v')^2 - vv'']$ . Hence,

$$\Delta = -v^2q''[2(v')^2 - vv'']/(v')^4qK^2 > 0. \tag{A.4}$$

Thus, the second-order conditions represented by (A.1), (A.2), and (A.4) are satisfied, and there exists a unique interior solution to the first-order conditions (5).

Differentiating the first-order conditions (5) with respect to  $w$  and then using Cramer’s rule, we immediately obtain, following (A.1) and (A.2),

$$\partial x^*/\partial w = \pi_{pp}/\Delta < 0 \quad \text{and} \quad \partial p^*/\partial w = -\pi_{px}/\Delta < 0. \tag{A.5}$$

Moreover,  $\partial q^*/\partial w = q'(x^*)(\partial x^*/\partial w) < 0$ .

Finally, we examine the effect of a wage rate increase on the equilibrium consumption  $Q^*$ . Using (A.5) in (6), we get

$$\partial Q^*/\partial w = (Q_q\pi_{pp}q'(x^*) - Q_p\pi_{px})/\Delta. \tag{A.6}$$

Next, using (A.1) and (A.2) to substitute  $\pi_{pp}$  and  $\pi_{px}$  in (A.6) and then making use of (A.0) to simplify the expression, we obtain

$$\partial Q^*/\partial w = (1 - \mu)q'[(1 - \mu)p/\mu - v]/\mu q(v')^2\Delta = (q - 1)vq'(1 - \mu)/\mu q(v')^2\Delta,$$

where the second equality is obtained using (A.3). This completes the proof.

**Proof of inequality (8).** Applying the envelope theorem to (4) gives

$$d\pi^*(\mu)/d\mu = (\partial\pi/\partial w)(\partial w^*/\partial\mu) + \partial\pi/\partial\mu = -x^*(\partial w^*/\partial\mu) + p^*Q_\mu. \quad (\text{A.7})$$

Using (A.0), we have the second term  $\partial\pi/\partial\mu = -p^2/\mu^2 qv' > 0$ . Since  $\partial w^*/\partial\mu = 0$  in the cases of  $D_0, D_1$  and  $D_3$ , the right-hand side of (A.7) is obviously positive in these cases.

Let us now turn to the case of  $D_2$  to derive  $\partial w^*/\partial\mu$ . Differentiating the labor market equilibrium condition,  $x^*(w, \mu) = M - N$ , with respect to  $\mu$  gives  $\partial w^*/\partial\mu = -(\partial x^*/\partial\mu)/(\partial x^*/\partial w)$ .

We now fix  $w$  to get the partial effect of  $\mu$  on  $x^*$ . This can be obtained by first differentiating the first-order condition (5) with respect to  $\mu$  and then using Cramer's rule:

$$\partial x^*/\partial\mu = [-pq'Q_{q\mu}\pi_{pp} + (pQ_{p\mu} + Q_\mu)\pi_{xp}]/\Delta.$$

We then substitute  $\pi_{pp}$  and  $\pi_{xp}$  in the above expression using (A.1) and (A.2), to obtain

$$\partial x^*/\partial\mu = q'[-pQ_{q\mu}(2Q_p + pQ_{pp}) + (pQ_{p\mu} + Q_\mu)(Q_q + pQ_{qp})]/\Delta.$$

Then, using (A.0) to substitute for  $Q_\mu, Q_q, Q_p, Q_{q\mu}, Q_{p\mu}$  and  $Q_{qp}$ , we can easily simplify the above result to

$$\partial x^*/\partial\mu = pq'v[2(v')^2 - vv'']/(v')^4 \mu^2 q^2 \Delta > 0. \quad (\text{A.8})$$

Note that  $\pi_{pp} = (2Q_p + pQ_{pp}) = (1 - \mu)[2(v')^2 - vv'']/\mu q(v')^3$ . Using (A.4), (A.8), and the expression for  $\pi_{pp}$ , we also get

$$(\partial x^*/\partial\mu)/(\partial x^*/\partial w) = pq'v[2(v')^2 - vv'']/\pi_{pp}\mu^2 q^2 (v')^4 = pq'v/(1 - \mu)\mu qv'.$$

Using the above results in (A.7) and substituting  $(1 - \mu)p/\mu$  with  $vq$  [according to (A.3)] gives

$$d\pi^*(\mu)/d\mu = -pv(q - xq')/(1 - \mu)\mu qv' > 0, \quad (\text{A.9})$$

where the inequality holds because  $q - xq' = 0$  at  $x = 0$  and  $\partial(q - xq')/\partial x = -xq'' > 0$ .

The same result (A.9) holds for the case of  $D_4$ , for an obvious reason.

**Proof of Proposition 3.** Because of (8), we need only to prove that for sufficiently small  $\mu$ , the PS developer will not enter.

First, since  $Q < 1$  (for  $p > 0$ ),  $Qp < p$ . Thus, if  $p \leq f$ , the PS developer will not enter because entry leads to negative profit:  $\pi^*(\mu) - f < 0$ .

Second, suppose  $p > f$ . Since  $q(x)$  is bounded above, denote  $\bar{q} \equiv \sup\{q(x)\}$ . By definition, the demand  $Q$  is determined by  $v(Q) = (1 - \mu)p/\mu\bar{q}$  and the right-hand side is greater than  $(1 - \mu)f/\mu\bar{q}$ . Recall that  $v'(Q) < 0$  and  $v''(Q) < 0$ . That is,  $v(Q)$  is bounded above. Hence, there exists  $\mu_1$ , independent of  $p$  and  $q$ , such that

$Q \leq f/v(0)\bar{q}$  for all  $\mu \leq \mu_1$ . As a result, for  $\mu \leq \mu_1$ , the revenue of the PS developer is  $Qp \leq fp/v(0)\bar{q}$ .

Third, note that if  $p \geq v(0)\bar{q}\mu/(1 - \mu)$ , then  $Q = 0$ . Again  $\pi^*(1, \mu) - f < 0$ . To avoid this case, we have to focus on  $p < v(0)\bar{q}\mu/(1 - \mu)$ . The right-hand side is less than or equal to  $v(0)\bar{q}$  since  $\mu \leq \frac{1}{2}$ . Therefore, for  $p \in (f, v(0)\bar{q})$ , using (A.3), we have

$$Qp \leq fp/v(0)\bar{q} < f \implies \pi^*(\mu) - f < 0.$$

To summarize, the PS developer will not enter if  $\mu \leq \mu_1$ , which is independent of  $p$  and  $q$ . This  $\mu_1$  gives a sufficient condition for no entry, and so  $\mu_1$  may be smaller than  $\mu_0$  in the proposition.

## References

- Anderson, J.E., Young, L., 2003. Trade and contract enforcement. Mimeo., Boston College.
- Arora, A., Asundi, J., 1999. Quality certification and the economics of contract software development: a study of the Indian software service companies. NBER Working Paper 7260.
- Arora, A., Arunachalam, V.S., Asundi, J., 1999. The Indian software service industry. Mimeo, The H. John Heinz III School of Public Policy and Management.
- Baba, Y., Takai, S., Mizuta, Y., 1995. The Japanese software industry: The hub structure approach. *Research Policy* 24, 473–486.
- BSA, 1998. The contribution of the packaged software industry to the European economies. A study conducted by PricewaterhouseCoopers, commissioned by the Business Software Alliance (<http://www.bsa.org>).
- BSA, 1999a. 1998 global software piracy report. A study conducted by International Planning and Research Corporation for the Business Software Alliance (<http://www.bsa.org>).
- BSA, 1999b. Forecasting a robust future: An economic study of the U.S. software industry. A study conducted by PricewaterhouseCoopers, commissioned by the Business Software Alliance (<http://www.bsa.org>).
- Chen, Y., Png, I., 2003. Information goods pricing and copyright enforcement: Welfare analysis. *Information Systems Research* 14 (1).
- Correa, C.M., 1996. Strategies for software exports from developing countries. *World Development* 24 (1), 171–182.
- Fortune, 1994. Competitiveness: How U.S. companies stack up now. April 18.
- Gilman, J., 1992. When the SPA comes, a-knockin. *Computerworld* (December 7, 1992), 108.
- Grossman, G., Helpman, E., 2002. Integration versus outsourcing in industry equilibrium. *Quarterly Journal of Economics* 117 (1), 85–120.
- Klein, B., Crawford, R., Alchian, A., 1978. Vertical integration, appropriable rents, and the competitive contracting process. *Journal of Law and Economics* 21 (2), 297–326.
- McLaren, J., 2000. Globalization and vertical structure. *American Economic Review* 90 (5), 1239–1254.
- Png, I., Tao, Z., 2000. Installment payments in contracts for systems development. Mimeo., National University of Singapore.
- Qiu, L., 2004. Software Development and Trade. Mimeo, Hong Kong University of Science and Technology.
- Siwek, S.E., Furchtgott-Roth, H.W., 1993. *International Trade in Computer Software*. Quorum Books, London.
- Stuckey, J., White, D., 1993. When and when *not* to vertically integrate. *Sloan Management Review*, Spring, 71–83.
- Torrisi, S., 1998. *Industrial Organisation and Innovation: An International Study of the Software Industry*. Edward Elgar, Cheltenham, UK.

- Wang, E.T., Barron, T., Seidmann, A., 1997. Contracting structures for custom software development: The impacts of informational rents and uncertainty on internal development and outsourcing. *Management Science* 43 (12), 1726–1744.
- Whang, S., 1992. Contracting for software development. *Management Science* 38 (3), 307–324.
- Williamson, O., 1971. The vertical integration of production: Market failure considerations. *American Economic Review* LXI (2), 112–123.
- Zhang, J.X., Wang, Y., 1995. *The Emerging Market of China's Computer Industry*. Quorum Books, London.