7-24-2011

Functionality-rich Versus Minimalist Platforms: A Two-sided Market Analysis

Soumya Sen
Princeton University, soumyas@princeton.edu

Roch A. Guérin
University of Pennsylvania, roch@guerinfamily.org

Kartik Hosanagar
University of Pennsylvania

Functionality-rich Versus Minimalist Platforms: A Two-sided Market Analysis

Soumya Sen*
EE, Princeton University
soumyas@princeton.edu

Roch Guérin
ESE, U.Pennsylvania
guerin@ee.upenn.edu

Kartik Hosanagar
Wharton, U.Pennsylvania
kartikh@wharton.upenn.edu

ABSTRACT

Should a new "platform" target a functionality-rich but complex and expensive design or instead opt for a bare-bone but cheaper one? This is a fundamental question with profound implications for the eventual success of any platform. A general answer is, however, elusive as it involves a complex trade-off between benefits and costs. The intent of this paper is to introduce an approach based on standard tools from the field of economics, which can offer some insight into this difficult question. We demonstrate its applicability by developing and solving a generic model that incorporates key interactions between platform stakeholders. The solution confirms that the "optimal" number of features a platform should offer strongly depends on variations in cost factors. More interestingly, it reveals a high sensitivity to small relative changes in those costs. The paper's contribution and motivation are in establishing the potential of such a cross-disciplinary approach for providing qualitative and quantitative insights into the complex question of platform design.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Design Studies; K.6 [**Management of Computing and Information Systems**]: Economics

General Terms

Design, performance

Keywords

Platforms, two-sided markets, economics of networks

1. INTRODUCTION

Communication platforms from the Internet to social networks (*e.g.,* Facebook, Twitter), as well as a plethora of computing platforms from personal computing (*e.g.,* Windows, Apple OSX, Linux), to mobile devices (*e.g.,* Android, iOS, Symbian), to cloud computing solutions (*e.g.,* Google, Amazon, Microsoft Azure), are emerging as the main drivers of our digital economy. Together with this emergence often comes a period of transformation during which platforms face major design decisions that affect their future success and eventual survival.

For example, many attribute the Internet's success to its original minimalist design, which allowed it to adapt to new technologies and foster the creation of

---

*This work was done while the author was at Penn

a wealth of applications. However, as it matures and transforms from a "physical" network platform to a broader ecosystem of software and services, the question of whether or not to abandon this minimalist principle is increasingly raised [18, 7, 28]. Answering it is non-trivial, especially given the lack of a systematic framework for identifying and evaluating the underlying design trade-offs. The paper does not claim an answer to this complex and multi-faceted question. Instead, our aim is to highlight the availability and relevance of tools and methodologies from the field of economics to explore this complex issue. In support of this claim, we offer an initial quantitative step and illustrate through partial and preliminary results the kind of insight it can yield.

In general, a platform's success is based on its ability to "connect" *consumers* of applications and services to *developers* of those applications and services. The platform entices developers to join by providing access to functionality through built-in APIs, modules, tools, etc., which make it easier to innovate new applications and services of interest to consumers. The platform (development) costs, however, grow with the richness of the functionality it offers. The main question faced by a platform provider is, therefore, to decide what level of functionality to offer, or in other words how many "features" to include in the platform[1] so as to maximize its own profit.

A minimalist platform has a low cost but makes developing services and applications more complex, which limits the number of application developed for it. This makes the platform less attractive to consumers and lowers revenues. Conversely, a feature-rich platform is expensive to build, but this cost may be offset by facilitating the development of more applications, therefore attracting more consumers. Hence, developing tools to explore this trade-off is of interest to platform providers[2]. In the rest of this paper, we demonstrate how a *two-sided market* [24] formulation can be used to investigate the problem. The platform is the 'market' and consumers and developers are the 'two sides' of the market.

The investigation illustrates how a two-sided market model can capture the decision problem of a platform provider. It also affords initial insight that provides some credibility that the approach holds promises as a quantitative tool in support of platform design. In particular, the model identifies the ratio of the *rate*

---

[1]We use the terms features and functionality interchangeably throughout the paper.
[2]The paper assumes a monopoly platform setting.

of change in the cost (to the platform) of adding new features relative to the cost of developing applications given a number of platform features, as a key factor in determining the optimal (for the platform) number of features to offer. This optimal choice is, however, highly sensitive to small changes in this ratio, with minor differences producing drastically different outcomes, *i.e.,* shifting the optimal operating point from a minimalist to a feature-rich platform. This negative result not-withstanding, the model provides a framework for reasoning about the impact of introducing more features to a platform. In addition, in cases where the costs of developing new features and their benefits in lowering application development costs can be estimated, the model offers quantitative tools that can assist decision makers. Section 3.5 reviews some scenarios for which such estimation may be feasible, and for each broadly characterize the "shape" of the cost functions as the number of features that the platform offers varies.

The rest of the paper is organized as follows. Section 2 reviews prior works in two-sided markets and e-commerce platform intermediaries, which provide useful background on techniques of potential relevance to platform design. Section 3 casts the problem of platform design using a two-sided market model. Section 4 outlines a solution methodology, while Section 5 presents a preliminary analysis based on this solution method, and uses its results to investigate the impact of different factors. Section 6 discusses possible extensions to the preliminary results of the paper, and more generally argues for the applicability of models from the economics literature to a variety of technology design issues.

2. RELATED LITERATURE

The purpose of this section is two-fold. First, it offers a brief overview of two sets of works relevant to the type of models the paper considers. Second, it seeks to position the approach used in the paper in the context of these related works. The two sets of works relevant to the paper are (i) platform intermediaries in e-commerce markets; and (ii) two-sided markets.

As discussed earlier, the problem faced by the platform is that its costs and benefits are coupled through cross-externalities involving its two customer types, *i.e.,* users and application developers. This has arisen in other settings, and in particular in e-commerce markets, which therefore boast a large body of relevant works that we briefly discuss. Similarly, the modeling of the platform as a two-sided market with users and application developers as the two sides of the market, calls for at least a cursory review of works in that area. In both cases, the primary difference between this paper and previous works is its focus on the trade-off between the platform and application developers costs, and how it affects the platform's design choice.

Platform intermediaries in e-commerce markets: The bulk of the literature in electronic intermediaries has focused on how they lower search costs for buyers and increase price competition among sellers [2]. However, a number of works [3, 2, 29] have explored the impact of infrastructural investments by the intermediary platform on cross-network externalities. For example, [3] shows that it is optimal for an intermediary to invest in network externalities asym-

metrically to maximize the network benefits for one market side. In contrast, we consider scenarios where the platform provider does not have the means to directly impact cross-externalities. Instead, the platform can invest in adding features that make application development easier for developers, and thus indirectly influence adoption levels. Such scenarios are typical for many web services and social network platforms where the level of functionality offered by the platform determines how costs and benefits are shared by the platform and developers. In that context, we investigate how different factors affect the platform's decision to be minimalist or feature-rich. The results contribute to the literature on e-commerce intermediary investments and platform design.

Two-sided markets: Two-sided markets are made of two interdependent groups of customers (*e.g.,* sellers and buyers) and a platform intermediary. The platform facilitates interactions between these two customer groups and generates its revenue by charging them a price for joining the platform. The focus of most works on the topic has to-date been on how pricing and pricing structure affects the platform adoption and its success, *e.g.,* see [15, 24, 22, 14, 29] for relevant discussions and pointers to other related works. A few more recent works [9, 20] have relied on two-sided markets to investigate net-neutrality. Our work builds on the existing literature, but rather than focusing on pricing it uses a two-sided market model to explore the effect of platform functionality on its adoption. As mentioned earlier, the question is relevant to several contemporary plaforms from the Internet, to Facebook, to Amazon Web Services (AWS), etc. For example, in the case of Internet, the question is cast in terms of whether or not it should depart from its original minimalist design. A formulation as a two-sided market can provide a quantitative handle on what is arguably a complex question.

3. MODEL FORMULATION

A platform provider attracts developers and consumers by creating value that entices them to join the platform. This 'value' depends on a number of factors, such as the platform's intrinsic value, the subscription fees to join it, the cost of developing applications for it, and *externalities* that affect the value that developers and consumers derive from joining the platform. When modeling a platform as a two-sided market, externalities are usually classified as *same-side* externalities and *cross-side* externalities. Same-side externalities arise in each side of the market from the presence of other users [19, 9, 29]. Cross-side externalities measure benefits that one side of the market derives from the other. These are usually positive, *i.e.,* consumers benefit from more applications offered by developers, and conversely developers benefit from being able to target more consumers.

The adoption of the platform by either developers or consumers depends on the overall value they derive from it. As commonly done, we measure this value through a *utility function* that incorporates the different factors that contribute to it. Similarly, the impact of the decisions that the platform provider makes, *i.e.,* pricing and selection of the platform's functionality, are also reflected through the platform provider's utility function. The utility functions for the platform,

the developers, and the consumers are described in Subsections 3.2, 3.3, and 3.4, respectively. However, before introducing these utility functions, we briefly review a number of assumptions we make in the model and their implications on its applicability.

3.1 Assumptions and Implications

As with most models, we make assumptions for analytical tractability and hope that the results offer qualitative insight applicable in practice. Following [3, 9, 22], the model considers only cross-side externalities[3], as they typically impact adoption the most.

We also assume that developers generate revenue from advertisements and not consumers purchases, *i.e.,* free downloads and minimal transaction costs. This is reasonable in many settings, *e.g.,* when applications are offered for free and the bulk of the developers' revenue comes from location based and personalized advertising [17, 25], a trend that is expected to grow [10]. The revenue generated by an application is further assumed to be linear in the number of users[4].

Two other important assumptions are that (i) applications all make use of the *same set of platform features*; and (ii) the functionality embedded in these features can be built by *either the platform or the developers,* with possibly different costs but the same quality. We briefly expand on both assumptions.

Assumption (i) implies homogeneous development needs across applications (services). In other words, they rely on the same platform application programming interfaces (APIs) or independent features created by developers. They can still be differentiated, but this clearly limits the range of their differences. Assumption (ii) calls for the platform provider to know application development needs ahead of time, and for application developers to be able to independently develop features that the platform decides not to incorporate. This is reasonable for many software products and services, where platform and applications share a common technology. However, it excludes hardware features whose presence or absence determines the feasibility (or not) of certain applications, *e.g.,* a graphic processor is mandatory for certain rendering effects. Implicit in assumption (ii) is that the development quality (and cost) of a feature by either the platform or the developers, is fixed and not a decision variable.

In general, assumptions (i) and (ii) limit the model's applicability to platforms that are software ecosystems, *e.g.,* cloud computing, web services, OSes, etc. Next, we introduce the utility functions that drive the decisions of the platform provider, application developers, and consumers.

3.2 Platform Utility

The platform provider's goal is to maximize its profit, which depends on revenue from the two market sides and the cost of the platform features it offers.

We use x_c and n_d to denote the fraction of a large population of consumers and developers who join the platform. As in [3, 9], the platform charges flat fees of p_c and b_d to the consumers and to the developers, re-

spectively[5]. These fees may be incurred as a monthly membership fee for consumers and as a licensing fee for developers.

The revenue for the platform is, therefore[6],

$$p_c x_c + b_d n_d \,.$$

As mentioned earlier, the set of platform features of potential benefits to application developers is assumed known to the platform provider. Embedding more features in the platform incurs a greater cost, and we denote as $C(F)$ the cumulative cost of incorporating F features. We assume that the set of possible features is large. Hence, when mapped on to an interval $[0, F^{\max}]$, they result in a differentiable, monotonically increasing function for $F \in [0, F^{\max}]$. An integrality constraint on F is, therefore, not considered explicitly. In Subsection 3.5, we discuss specific, real-world examples to illustrate possible behaviors for $C(F)$, *i.e.,* concave or convex.

The profit (utility) of a platform with F built-in features and fees of p_c and b_d is given by

$$U_p = p_c x_c + b_d n_d - C(F) \qquad (1)$$

As discussed in Section 4, Eq. (1) together with similar expressions for the utility of consumers and application developers will guide the decisions of how many features to embed in the platform and how to price it.

3.3 Developer Utility

Developing applications incurs a cost that depends on the level of support provided by the platform (number of features). A feature-rich platform will usually have higher subscription fees, but afford lower application development costs. Conversely, the revenues generated by an application depend on the number of consumers of the platform, and grow in proportion to that number. Eq. (2) captures the combined effect of these factors on the developers' utility.

$$U_d = \alpha x_c - b_d - (K(F) + \tau\phi) \qquad (2)$$

The first component of Eq. (2) represents the application revenues generated from the x_c consumers that joined the platform (the factor α is the marginal value, *e.g.,* ad revenue, that a consumer generates for the developer). The second component of Eq. (2) is the flat-fee, b_d, a developer pays the platform, *e.g.,* license fee for certification.

The last component of Eq. (2) accounts for development costs. They depend on the number F of features provided by the platform, as captured by $K(F)$. The function $K(F)$ is assumed differentiable, and monotonically decreasing for $F \in [0, F^{\max}]$. For a given $F, K(F)$ is the same for all developers, and can be interpreted as the base cost of developing applications on a platform with F built-in features. This assumes comparable skill levels across developers, who can however exhibit heterogeneity in their overall development costs, *e.g.,* because of different fixed costs. This heterogeneity is captured in the factor $\tau\phi$ of Eq. (2), where as is commonly assumed [3, 9, 27, 29] ϕ is uniformly distributed on a unit interval[7], and τ is a normalization constant.

---

[3]Appendix E.1 shows that the paper's main results are qualitatively unaffected by same-side externalities.
[4]Non-linearity has a quantitative, but not a qualitative effect.

[5]p_c or b_d can be positive or negative (subsidy).
[6]See Appendix B for relabeling of parameters to account for consumer and developer population sizes.
[7]Results typically extend to other distributions [5, 12] that share with the uniform distribution the impor-

Section 3.5 provides again illustrative, real-world examples of possible $K(F)$ functions. In particular, $K(F)$ can be convex or concave depending on how additional platform features translate into marginal reductions in development costs.

3.4 Consumer Utility

The value that consumers derive from joining a platform depends on the subscription fees charged, and on the number of applications and services accessible through the platform. Consumers are typically heterogeneous, and this heterogeneity manifests itself in how they value the platform, applications and services (cross-side externality), or both. For simplicity and analytical tractability, we focus on a model where heterogeneity is present only in how consumers value access to applications. Appendix E.2 presents an alternative utility function and its analysis, where consumers are instead heterogeneous in how they value the platform. The results under both utility functions are qualitatively similar.

The consumer utility function is of the form:

$$U_c = \theta \beta n_d - p_c \qquad (3)$$

The first component, $\theta \beta n_d$, captures the cross-side externality benefits that consumers enjoy from accessing applications available on the platform. Consistent with earlier works [29, 3, 27], those benefits are assumed linear in the number of available applications and, therefore, developers (n_d) under the assumption that developers are homogeneous in the number of applications they create. The factor β denotes the marginal externality benefit associated with each developer. The term $\theta \in [0,1]$ is a random variable that accounts for heterogeneity in how users value access to applications. As with the random variable ϕ of Eq. (2), we assume that θ is uniformly distributed in $[0,1]$. The last element of Eq. (3) is the price p_c, which is a flat membership fee paid to the platform provider.

3.5 Representative Examples

Before presenting how the three utility functions just introduced combine in the platform provider decision process, we pause to introduce two examples[8] (see Fig. 1) that illustrate possible combinations of the cost functions $C(F)$ and $K(F)$. In both examples, there is an inherent ordering of the features the platform provider is considering offering, *i.e.*, from basic features to more advanced ones, with the latter building on the former. The examples differ in the relative cost (to the platform) of more advanced features compared to basic ones, and the impact of additional feature on application development costs.

1. Amazon Web Services Platform: Amazon Web Services (AWS) is a cloud computing platform that offers functionality (features) which third-party developers can use to create services for clients (consumers). These features include Amazon EC2 (computation), SimpleDB (database), Amazon S3 (storage), etc. Consumers and developers of services and applications on the AWS platform enjoy cross-side externalities, for which they pay subscription fees.

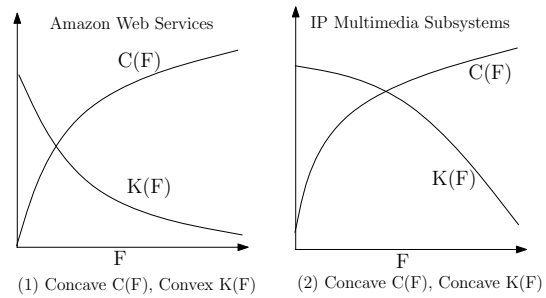(1) Concave C(F), Convex K(F) (2) Concave C(F), Concave K(F)

Figure 1: Examples of different $C(F)$ and $K(F)$.

The introduction of features on the AWS platform proceeded in two steps. Between 2006-2007, Amazon introduced a set of core features (EC2, FPS, SimpleDB, etc.) that offered basic capabilities such as computation, database, and payment for its AWS platform. Additional features (*e.g.,* SQS, SNS, DevPay, etc.) that built on these core capabilities were subsequently introduced.

Adding each feature to the AWS platform came at a cost. Using API complexity[9] as a proxy for the platform's development cost together with data from [13], it can be observed that capabilities such as EC2, FPS, and SimpleDB came at a higher cost than that of follow-on enhancements such as SNS and DevPay. From this data, one can infer that the AWS platform has a feature development cost function, $C(F)$, which is a *concave increasing* function of F. Conversely, the benefits of each feature can be estimated based on its "popularity" among developers, *i.e.,* presuming that more useful features are more likely to be used by developers. Using again [13], we see that most core features are significantly more popular than subsequent enhancement features[10]. In other words, the features that were the most costly to develop and incorporate in the platform were also the most useful to developers; at least based on how often developers took advantage of them. As a result, one can conclude that while the development cost function $C(F)$ of the AWS platform is a concave increasing function, the benefits that developers derive from those features, as captured by the function $K(F)$, is a *convex decreasing* function, *i.e.,* the more expensive initial features are also the most useful.

2. IP Multimedia Subsystems (IMS) Platform: The IMS platform is meant to facilitate the development of new integrated multimedia applications and services. Both applications developers and subscribers (consumers of services) pay a fee to the IMS platform. The platform offers a number of built-in capabilities such as a registration mechanism, co-location of multiple IMS services, quality of service, etc. These capabilities are exposed to developers through APIs using Java specifications (JSRs). There are multiple "layers" of JSRs [16, 21], from low-level JSRs such as JSR-180, to high-level JSRs such as JSR-186/187, to more

_____

tant property of a non-decreasing hazard-rate function $\mathcal{F}'(\phi)/(1 - \mathcal{F}(\phi))$.

[8] See Appendix A for a third example.

[9] The development cost of a feature can be approximated through the complexity of its API. [13] measured API complexity using the number of operations that the feature supports, as captured in the data required in specifying WSDL.

[10] *e.g.,* compare EC2, SDB, FPS to SQS, SNS and DevPay.

developer friendly APIs for Communication Services such as JSR-281+. Each layer builds on those below, with the base layer that implements the core capabilities of the platform the most expensive to develop. Additional layers are typically "wrappers" meant to hide low-level details from developers, and therefore typically easier for the platform to implement. The development cost function $C(F)$ of the IMS platform is, therefore, again a *concave increasing* function of the number F of features (JSRs) it offers.

On the developer side, application development costs are high when only low-level APIs are available, mostly because of greater programming complexity. As APIs that hide many of the platform's low-level intricacies are made available, development costs decrease rapidly. In other words, the function $K(F)$ is a *concave decreasing* function, *i.e.,* low-level APIs have little effect on developers costs, while higher-level ones deliver significant benefits.

In the next section, we introduce the methodology used by the platform provider to decide on the "optimal" number of features to incorporate.

4. SOLUTION METHODOLOGY

The platform provider's objective is to decide on the number of features to include in the platform, and what to charge consumers and developers to maximize profit. This can be realized through the three-stage sequential process of Fig. 2. In the first stage, the platform provider chooses the number F of features built into the platform. Given a choice for F, prices for the two market sides are chosen in the second stage. Equilibrium adoption levels of consumers and developers are simultaneously realized in the third stage. The three stages are referred to as the *Design Stage, Pricing Stage*, and *Adoption Stage,* respectively.
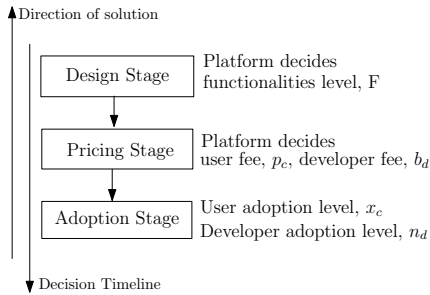

Figure 2: Sequential decision process

This sequential decision process is solved in reverse order. Equilibrium adoption levels for users and developers are first computed for a given choice of prices and number of built-in features. Next, given a number of features, 'optimal' prices are computed based on the equilibrium adoption levels of the previous step. The results characterize the platform's profit for any given number of features. This can then be used to find the 'optimal' number of features, F^*, that maximizes the platform's profit. These steps are detailed next.

4.1 Adoption Stage

Consumers and developers both make rational and incentive-compatible decisions, and join the platform only if they derive positive utility. We let x_c^* and n_d^*

denote the expected fraction of consumers and developers joining at equilibrium.

Given p_c, b_d, and F, the value $\widehat{\theta}$ of the marginal consumer who is indifferent (derives zero utility) between joining the platform or not is

$$\widehat{\theta} = 1 - x_c = p_c / \beta n_d^*. \tag{4}$$

Similarly, the value $\widehat{\phi}$ of the marginal developer who is indifferent between joining the platform or not is

$$\widehat{\phi} = n_d = \alpha x_c^* - b_d - K(F). \tag{5}$$

The parameters in Eq. (5) are normalized and relabeled following the procedure of Appendix B. At equilibrium, $x_c^* = x_c$ and $n_d^* = n_d$, so that equilibrium adoption levels satisfy

$$p_c = (1 - x_c^*)\beta n_d^* \tag{6}$$
$$b_d = \alpha x_c^* - n_d^* - K(F) \tag{7}$$

4.2 Pricing Stage

Given a number of features F, the decision problem of the platform provider is to select fees p_c and b_d that maximize its profit, subject to constraints on the fractions of consumers and developers joining the platform. This yields

$$\max_{x_c^*, n_d^*} U_p = p_c x_c^* + b_d n_d^* - C(F) \tag{8}$$

$$s.t. \quad 0 \le x_c^* \le 1; \ 0 \le n_d^* \le 1$$

Using Eqs. (6) and (7) in Eq. (8), optimal fees and corresponding adoption levels are obtained. Proposition 1 gives expressions for interior solutions, *i.e.,* $0 < x_c^*, n_d^* < 1$), which arise when $\alpha < \beta$ and $4\beta K(F) < (\alpha + \beta)^2 < 4\beta(2 - K(F))$, under which the second order conditions of the Hessian also hold. Boundary solutions, *i.e.,* $x_c^* = 0, 1$ or $n_d^* = 0, 1$ solutions are given in Appendix D. Derivations can be found in Appendix C.

PROPOSITION 1. *Optimal price levels (p_c^*, b_d^*) and optimal adoption levels (x_c^*, n_d^*) that maximize the platform provider's profit are given by*

$$p_c^* = (\beta - \alpha)((\alpha + \beta)^2 - 4\beta K(F))/16\beta \tag{9}$$
$$b_d^* = ((3\alpha - \beta)(\alpha + \beta) - 4\beta K(F))/8\beta \tag{10}$$
$$x_c^* = (\alpha + \beta)/2\beta \tag{11}$$
$$n_d^* = ((\alpha + \beta)^2 - 4\beta K(F))/8\beta \tag{12}$$

Proposition 1 reveals properties that are consistent with prior works in two-sided markets [3, 29, 6]. In particular, optimal pricing is typically asymmetric, *i.e.,* different prices are levied on the two market sides, and in some cases one market side may be subsidized, *e.g.,* $b_d^* < 0$ while $p_c^* > 0$.

4.3 Design Stage

Using the results of Proposition 1 in Eq. (8), the platform provider can determine the optimal number of features F^* that maximizes profits. Solving for $\frac{\partial U_p}{\partial F} = 0$, F^* can be shown to verify the conditions of Proposition 2.

PROPOSITION 2. *The optimal number F^* of features that should be built into the platform to maximize profit*

satisfies

$$\frac{C'(F^*)}{K'(F^*)} = \frac{K(F^*)}{2} - \frac{(\alpha+\beta)^2}{8\beta} \qquad (13)$$

$$\Rightarrow \quad \frac{C'(F^*)}{K'(F^*)} = -n_d^*(F^*) \qquad (14)$$

$$\text{and} \quad C''(F^*) > -n_d^*(F^*)K''(F^*)$$
$$+ (1/2)[K'(F^*)]^2 \qquad (15)$$

where Eq. (14) is obtained using Eq. (12) in Eq. (13).

Note that the condition $C'(F^*)/K'(F^*) = -n_d^*(F^*)$ of Eq. (14) implies that at F^*, *the marginal increase in the cost to the platform of adding more features equals the marginal decrease in development costs across all developers subscribed to the platform*[11].

Note also (see Section 5) that selecting the optimal number of features still calls for evaluating profits at *all* F^* that satisfy Eqs. (13) and (15), *and* at the boundaries $F = 0$ and F^{max}.

Using the above results, Section 5 explores how different system parameters, *i.e.,* externality benefits and costs, affect the optimal number of platform features.

5. ANALYSIS

In this section, we use Proposition 2 to explore properties of F^* and the influence of system parameters. We begin with Eqs. (13) and (15), which we use to characterize how changes in (cross-side) externality benefits affect F^*. The results are in Proposition 3, whose proof is in Appendix C.

PROPOSITION 3. *For interior solutions* $(0 < x_c^*, n_d^* < 1)$, *increases in the cross-side externality benefits* α *and* β *favor adding functionality to the platform. In other words,* $\frac{\partial F^*}{\partial \alpha} > 0$ *and* $\frac{\partial F^*}{\partial \beta} > 0$.

Fig. 3 illustrates this behavior through a representative example[12]. It shows that F^* increases as developer's cross-side benefits increase from $\alpha = 0.65$ to $\alpha = 0.67$. We note that Proposition 3 is consistent with arguments in favor of expanding the Internet's functionality at a time where the services it offers become more valuable, and the providers of those services derive more value than previously feasible.

Next, we investigate how F^* is affected by changes in the relationship between the cost of adding new features to the platform and the benefits that application developers derive from them. The platform development costs $C(F)$ increase with F, while application development costs $K(F)$ decrease. The relative rates of these increases and decreases ultimately determine F^* and the optimal prices, $p_c^*(F)$ and $b_d^*(F)$. The dependency of F^* on the relative rate of change of $C(F)$ and $K(F)$ is captured in Eq. (13).

In general, note that the platform utility function of Eq. (1) includes product terms of the form $p_c x_c$ and $b_d n_d$, which imply complex dependencies on the functions $C(F)$ and $K(F)$. Hence, the function $U_p(F)$ that the platform provider seeks to maximize can exhibit a wide range of variations, *e.g.,* multiple maxima

---

[11]Eq. (14) remains valid for boundary cases where either market sides is at full market penetration.

[12]The parameters in all the figures of this section are assumed to be normalized with respect to populations of size $N_c = N_d = 10^3$, and maximum fixed cost for developers, $\tau = \$10^3$.
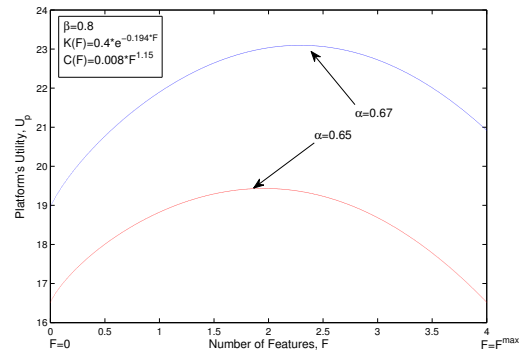

Figure 3: Impact of α on F^*.

and/or minima, even when $C(F)$ and $K(F)$ are individually "well-behaved," *e.g.,* concave or convex. Clearly, the possibility of several F values satisfying Eqs. (13) and (15) complicates the platform provider's decision process. Furthermore, the optimal decision also depends on how profits at these local maxima compare to "boundary" profits, *i.e.,* for $F = 0$ and $F = F^{max}$.

Numerical investigations easily produce combinations of $C(F)$ and $K(F)$ for which the platform utility exhibits local maxima at different F values. In general small adjustments in the relative rate of increase and decrease of $C(F)$ and $K(F)$ are sufficient to yield drastic shifts in outcome. This is to a large extent borne by Eqs. (13) and (15), which indicate that the F^* value at which both equations are satisfied can change substantially through small changes in the functional expressions of either $K(F)$ or $C(F)$. The implications are that the optimal investment in features for a platform cannot be easily inferred from general properties of $C(F)$ and $K(F)$, *e.g.,* concavity or convexity. Instead, it calls for a fine-grain comparison of the costs of developing features and their benefits to application developers.
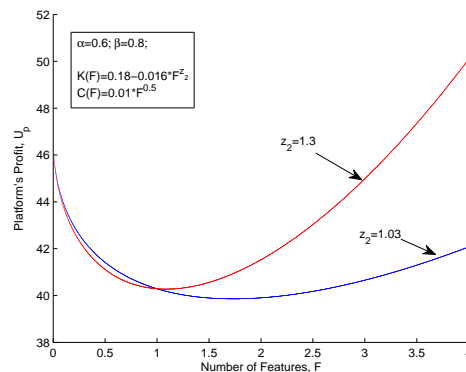

Figure 4: Boundary solutions ($F = 0$ or $F = F^{max}$) are optimal for concave $K(F)$ and $C(F)$.

The one instance for which the range of possible outcomes can be narrowed is when both $C(F)$ and $K(F)$ are concave, *e.g.,* the IMS platform example. In this case, the optimal number of features can be shown to always be at one of the two boundaries, *i.e.,* $F = 0$ or $F = F^{max}$ (see Appendix C for a proof). The op-
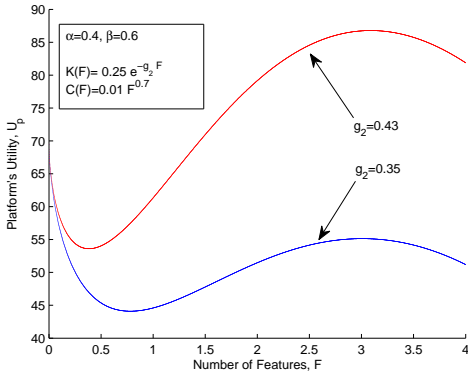
Figure 5: Multiple local maxima for convex $K(F)$ and concave $C(F)$.

timal solution can still be either a minimalist or a functionality-rich platform, but the number of alternatives to evaluate is considerably reduced. We illustrate this with a numerical example in Fig. 4. The figure shows the platform's profit as a function of the number of features it offers for two different configurations. In both configurations, the platform cost grows like $\sqrt{F}$, while application developers experience a super-linear cost decrease in F parametrized by z_2 (see the legend). The figure shows that when $z_2 = 1.03$ (the decrease is nearly linear), a minimalist platform is more efficient, while when the decrease if steeper ($z_2 = 1.3$) a functionality-rich platform is preferred. This is obviously intuitive, but the model offers a quantitative validation of this insight.

For all other combinations of convex or concave $C(F)$ and $K(F)$, more complex outcomes can arise, including instances where an intermediate number of features is optimal. We illustrate this with a numerical example in Fig. 5, which corresponds to a scenario where much like the AWS example of Section 3.5, $C(F)$ is concave increasing and $K(F)$ is convex decreasing in F. As before, the figure plots the platform's (profit) utility U_p (y-axis) as a function of $F \in [0,4]$ (x-axis). The figure displays results for two different convex decreasing $K(F)$, while $C(F)$ is kept equal to $C(F) = 0.01F^{0.7}$ (concave increasing). The two chosen functions for $K(F)$, $K(F) = 0.25e^{-g_2 F}$; $g_2 = \{0.35, 0.43\}$, differ in their rate of decrease with F. The figure reveals the following three interesting behaviors.

First, it shows that in this scenario there are two local maxima for the platform provider's utility; one corresponding to a minimalist choice ($F = 0$), and the other to an intermediate optimal number of features F^* that satisfies Eqs. (13) and (15). Selecting the globally optimal solution calls not only for computing F^*, but also for comparing profits at $F = 0$ and $F = F^*$.

Second, it shows that a small change in the rate of decrease in the developers' cost $K(F)$ can produce drastically different choices for the platform provider. In the case of $K(F) = 0.25e^{-0.35F}$, the provider's optimal choice is a minimalist design (*i.e.,* $F = 0$), while for $K(F) = 0.25e^{-0.43F}$, the provider should create a platform with a large number of built-in features (*i.e.,* $F = F^* \approx 3$). This illustrates the dependency of the decision process on the rate at which development costs decrease as the number of features increases. A sim-

ilar outcome can be obtained by keeping $K(F)$ constant, and changing the rate of increase in the platform provider's cost (*i.e.,* $C'(F)$).

Third, the figure illustrates a behavior that may at first seem counter-intuitive. Consider the two developers cost functions $K(F) = 0.25e^{-0.35F}$ and $K(F) = 0.25e^{-0.43F}$. The rate of decrease of $K(F)$ is higher in the second, so that the biggest benefits are realized when adding the first features. In contrast, the slower decrease in the first case implies that more features need to be added to realize a similar benefit. This would seem to suggest that a larger number of features would be preferable. The figure shows that the opposite is actually true, *i.e.,* a minimalist choice ($F = 0$) is preferred when $K(F) = 0.25e^{-0.35F}$, while $K(F) = 0.25e^{-0.43F}$ calls for a platform with a relatively large number of features. The reason is that when $K(F) = 0.25e^{-0.43F}$ and costs drop fast, adding features ultimately yields a lower *absolute* value of $K(F)$, which encourages more developers to join and ultimately produces a higher profit. In contrast, the slower cost decrease of $K(F) = 0.25e^{-0.35F}$ is such that the smaller number of developers that join is not sufficient to produce a higher profit than when $F = 0$.

Finally, we should point out that while the scenario of Fig. 5 showed only one interior maximum, it is possible to have more than one. This is illustrated in Fig. 7 in Appendix F, which involves a convex decreasing $K(F)$ function and a convex increasing $C(F)$ function. As in the previous example and for essentially the same reasons, the choice of which maximum yields the highest overall profit depends on the relative rates of change of the two cost functions.

6. CONCLUSIONS

The paper applies models from the economics literature to the problem of platform design, and uses them to explore whether a a minimalist or a feature-rich design should be used. The question is formulated using a two-sided market model, with the platform as the market and service developers and consumers as its two sides. The model is solved using a three stage sequential decision process, and results of a preliminary investigation are presented.

The investigation confirms a number of properties traditionally present in two-sided markets, *e.g.,* the benefits of asymmetric pricing, and the effect that cross-externalities have in shaping the outcome. More importantly, it illustrates how the platform decision is highly dependent on the relative rate of change of its own cost structure (how cost increases with the number of features it offers) and that of application developers (how they benefit from new features). An unfortunate corollary of this finding is that very minor changes in either cost structures can translate into very different (optimal) outcomes. This argues that given the inherent inaccuracy in estimating cost structures, identifying a platform's optimal (how feature-rich) design point remains challenging, and this in spite of the analytical handle that the paper offers.

This limited success notwithstanding, the paper has hopefully illustrated the applicability of the model on which it is based. Given the preliminary nature of the investigation, there are obviously many directions in which it can and should be extended. Empirical validations are obviously at the forefront, and exploring if

this can be done for one of the examples of Section 3.5 is of interest. Relaxing the modeling assumptions of Section 3.1 is also worth pursuing. In particular, applications should be able to use different "subsets" of the platform's features, with the fraction of applications to which a (new) feature is useful drawn from a probability distribution. Similarly, some features may be available *only* from the platform, *i.e.,* developers cannot implement a substitute, and their availability or unavailability would then determine the feasibility of some applications. Finally, another "natural" extension is to introduce multiple platforms and allow them to compete. In such a context, an attractive direction is to explore possible combinations with models from "evolutionary science," as recently done in [1]. Evolutionary behaviors could, for example, be incorporated to determine which features survive based on their usefulness to new applications, and how this affects a platform own survival.

7. REFERENCES

[1] S. Akhshabi and C. Dovrolis. The evolution of layered protocol stacks leads to an hourglass-shaped architecture. In *Proc. ACM SIGCOMM Conference*, Toronto, ON, Canada, August 2011.

[2] Y. Bakos. Reducing buyer search costs: Implications for electronic marketplaces. *MIS Quarterly*, 15(3):295–310, 1991.

[3] Y. Bakos and E. Katsamakas. Design and ownership of two-sided networks: Implications for Internet platforms. *Journal of Management Information Systems*, 25(2):171–202, 2008.

[4] R. Beaubrun, B. Moulin, and N. Jabeur. An Architecture for Delivering Location-Based Services. *International Journal of Computer Science and Network Security*, 7(7):160–166, 2007.

[5] H.K. Bhargava and V. Choudhary. Economics of an Information Intermediary with Aggregation Benefits. *Information Systems Research*, 15(1):22–36, 2004.

[6] B. Caillaud and B. Jullien. Chicken and egg: Competing matchmakers. *Rand Journal of Economics*, 34(2):309–328, 2003.

[7] J. Catone. eBay Launches Dev Platform - Too Little, Too Late?, 2008. www.readwriteweb.com /archives/ebay_launches_ dev_platform.php.

[8] K. Currier. *Comparative Statics Analysis in Economics.* World Scientific Publishing Company, August 2000.

[9] N. Economides and J. Tag. Net neutrality on the Internet: A two-sided market analysis. *Working Paper no. 07-14, NET Institute, New York University*, 2007.

[10] K. Flanigan. $1.5 million ad revenue suggests free apps could be the future, 2010. http://148apps.biz/1-5-million-ad-revenue-suggests-free-apps-could-be-the-future/.

[11] WPI Center for Wireless Information Network Studies (CWINS). Location based applications are driving innovation in wireless communication industry, 2010. http://www.wpi.edu/news/20101/2010wireless.html.

[12] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press„ Cambridge, MA, 1991.

[13] M. Garnatt. AWS by the numbers, 2010. http://www.elastician.com/2010/06/aws-by-numbers.html.

[14] A. Hagiu. Pricing and commitment by two-sided platforms. *The RAND Journal of Economics*, 37(3):720–737, 2006.

[15] A. Hagiu. Multi-Sided Platforms: From Microfoundations to Design and Expansion Strategies. *Harvard Business School Strategy Unit Working Paper No. 09-115*, 2009.

[16] JSR: IMS Communication Enablers (ICE), 2010. http://jcp.org/en/jsr/detail?id=325.

[17] M. Maisto. Mobile app revenue soaring, although most are free, 2010. www.eweek.com /index2.php?option=content&do_ pdf=1&id=58530.

[18] K. Martin and I. Todorov. How Will Digital Platforms Be Harnessed in 2010, and How Will They Change the Way People Interact with Brands? *Journal of Interactive Advertising*, 10(2):61–66, 2010.

[19] M.L.Katz and C. Shapiro. Network externalities, competition and compatibility. *American Economic Review*, 75(3):424–440, 1985.

[20] J. Musacchio, G. Schwartz, and J. Walrand. A two-sided market analysis of provider investment incentives with an application to the net-neutrality issue. *Review of Network Economics*, 8(1):22–39, 2009.

[21] U. Olsson and M. Stille. Communication services - the key to IMS service growth. *Ericsson Review*, 2008.

[22] G.G. Parker and M.W. Van Alstyne. Two-sided network effects: A theory of information product design. *Management Science*, 51(10):1494–1504, 2005.

[23] R. Reed. Facebook places: The de facto platform for LBS, 2010. http://momentfeed.com/2010/11/facebook-places-platform-lbs/.

[24] J.-C. Rochet and J. Tirole. Two-sided markets: a progress report. *The RAND Journal of Economics*, 37(3):645–667, 2006.

[25] E.I. Schwartz. Behind the surge in mobile advertising, 2010. http://www.technologyreview.com /business/26648/.

[26] S. Sen, R. Guérin, and K. Hosanagar. Functionality-rich versus minimalist platforms: A two-sided market analysis. Technical report, University of Pennsylvania, July 2011. Available at http://repository.upenn.edu/ese_papers/xxx.

[27] S. Sen, Y. Jin, R. Guerin, and K. Hosanagar. Modeling the dynamics of network technology adoption and the role of converters. *IEEE/ACM Trans. Netw.*, 18(6):1793–1805, 2010.

[28] I. Wilkes. Cloud platform choices: a developer's-eye view, 2010. arstechnica.com/business/news/2010/02 /cloud-basics.ars.

[29] B. Yoo, V. Choudhary, and T. Mukhopadhyay. A Model of neutral B2B intermediaries. *Journal of Management Information Systems*, 19(3):43–68, 2002.

APPENDIX

A. ADDITIONAL MODEL EXAMPLES

Social Network Platform with location-based services (LBS) support: A social network platform such as Facebook provides application developers access to basic capabilities, *e.g.,* APIs to access the users' social graph, database of user interests, affiliations, etc. However, it also offers more sophisticated functionalities such as real time updates and location-based services (LBS). These have enabled the rapid growth of applications that offer personalized services, *e.g.,* Facebook's Recommendation and Places [23].
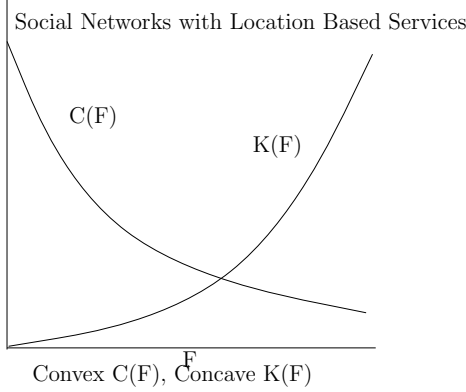
Social Networks with Location Based Services

Convex C(F), Concave K(F)

Figure 6: Convex $C(F)$ and concave $K(F)$

Adding this level of sophistication to the platform is, however, technically challenging. It calls for integrating capabilities such as spatial database management, location tracking, real time generation of cryptographic data [4], all of which are significantly more complex than the basic functionalities at the core of a social network platform, *e.g.,* access to the underlying social graph or to a user database. In other words, the function $C(F)$ that captures the cost of adding new (sophisticated) capabilities to a social network platform such as Facebook is a *convex increasing* function of F. This is shown in Fig. 6.

On the other hand, the benefits to application developers of those advanced features can be very high. For example, in the absence of LBS support from the platform, developers would need to build this capability into their application, *e.g.,* by interfacing to the GPS service built into the user's mobile device, when available. That those development costs are high is readily seen from the growth in the number of applications that rely on location information once LBS became available. Specifically, in spite of the large revenue potential of location-based services [23], there were relatively few applications that used location information before LBS became readily accessible to application developersr [11]. In other words, the substantial decrease in development costs that this produced, enabled many more developers to offer such applications. As a result, it can be argued that a social network platform such as Facebook is an environment where while sophisticated features are expensive to add, they are the ones that deliver the most benefits (reduction in development costs) to application developers. This means that the corresponding function $K(F)$ is a *concave decreasing* function of F.

B. SIMPLIFYING UTILITY PARAMETERS

This appendix explains briefly the parameter relabeling of parameters and normalization method used in the utility function of Eq. (1). We denote the total customer population for the two market sides, developers and consumers, as N_d and N_c, respectively. Eq. (2) and Eq. (3) can be written as

$$
\begin{aligned}
U_d &= \alpha x_c N_c - b_d - K(F) - \phi\tau &\quad (16)\\
U_c &= \theta\beta n_d N_d - p_c &\quad (17)
\end{aligned}
$$

The platform's utility function can then be written as

$$
U_p = p_c x_c N_c + b_d n_d N_d - C(F) \quad (18)
$$

The above three utility functions parallel those of Eq. (2), Eq. (3), and Eq. (1). Following the three stage solution methodology explained in Section 4, we have $p_c = (1 - x_c)\beta n_d N_d$ and $b_d = \alpha x_c N_c - K(F) - n_d\tau$. Using these prices, the platform utility U_p in the pricing stage is

$$
\begin{aligned}
U_p = &\Big((1 - x_c)\overline{\beta}n_d x_c &\quad (19)\\
&+ (\overline{\alpha}x_c - n_d - \overline{K(F)})n_d - \overline{C(F)}\Big)\tau N_d
\end{aligned}
$$

where $\overline{\beta} = (\beta N_c)/\tau$, $\overline{\alpha} = (\alpha N_c)/\tau$, $\overline{K(F)} = K(F)/\tau$ and $\overline{C(F)} = C(F)/(N_d\tau)$. With these relabeling of parameters and proper normalization, the platform's optimization problem reduces to an equivalent maximization problem, $max_{0 \le n_d^*, x_c^* \le 1} U_p$, where

$$
\begin{aligned}
U_p &= ((1 - x_c^*)\overline{\beta}n_d^*)x_c^*\\
&+ (\overline{\alpha}x_c^* - n_d^* - \overline{K(F)})n_d^* - \overline{C(F)}\\
&= \overline{p_c}x_c^* + \overline{b_d}n_d^* - \overline{C(F)} &\quad (20)
\end{aligned}
$$

where $\overline{p_c} = (1 - x_c^*)\overline{\beta}n_d^*$ and $\overline{b_d} = \overline{\alpha}x_c^* - n_d^* - \overline{K(F)}$. Note that the simplified equivalent function in Eq. (20) is structurally identical to Eq. (1) used in Section 3.2, and similarly, $\overline{p_c}$ and $\overline{b_d}$ are equivalent to p_c and b_c of Eq. (6) and Eq. (7) in Section 4.1. Examples of such standard utility functions after relabeling of parameters can be found in traditional economics model, *e.g.,* [9].

C. PROOFS

Proof of Proposition 1:

PROOF. The utility functions of the developers and consumers at the indifferent points provides the price levels, p_c and b_d, as in Eq. (6) and Eq. (7). Substituting these prices into the profit function of the platform provider in Eq. (1), we get the first order condition for each market side, $\frac{\partial U_p}{\partial x_c^*} = 0$ and $\frac{\partial U_p}{\partial n_d^*} = 0$, which gives

$$
\frac{\partial U_p}{\partial x_c^*} = (1 - 2x_c^*)\beta n_d^* + \alpha n_d^* = 0
$$

$$
\frac{\partial U_p}{\partial n_d^*} = (1 - x_c^*)\beta x_c^* + \alpha x_c^* - 2n_d^* - K(F) = 0
$$

Simultaneously solving the above equations gives x_c^* and n_d^*, which yields the optimal prices p_c^* and b_d^*.

$$
\begin{aligned}
x_c^* &= \frac{\alpha + \beta}{2\beta}\\
n_d^* &= \frac{(\alpha + \beta)^2 - 4\beta K(F)}{8\beta}
\end{aligned}
$$

The conditions for interior solutions, *i.e.,* $0 < x_c^*, n_d^* < 1$, are satisfied when $\alpha < \beta$ and $4\beta K(F) < (\alpha+\beta)^2 < 4\beta(2 - K(F))$. The second order maximization conditions for a negative definite Hessian (principal minors have alternate signs), *i.e.,* $|H_1|_{(x_c^*, n_d^*)} = -2\beta n_d^* < 0$ and $|H_2|_{(x_c^*, n_d^*)} = 4\beta n_d^* > 0$, also hold true for interior solutions. $\square$

Proof of Proposition 2:

PROOF. Using the expressions for p_c^*, b_d^*, x_c^* and n_d^* from Proposition 1, the platform's optimization problem in Eq. (8) gives

$$
\begin{aligned}
C'(F^*) &= \left[-\frac{\beta^2 - \alpha^2}{8\beta} - \frac{(\alpha+\beta)^2}{16\beta} \right. \\
&\quad \left. -\frac{(3\alpha - \beta)(\alpha+\beta)}{16\beta} \right] K'(F^*) \\
&\quad + \frac{K(F^*)K'(F^*)}{2} \\
\Rightarrow \frac{C'(F^*)}{K'(F^*)} &= \frac{-(\alpha+\beta)^2}{8\beta} + \frac{K(F^*)}{2}
\end{aligned}
$$

$\square$

Proof of Proposition 3:

PROOF. The proof relies on the conjugate pair theorem [8], which readily gives

$$
\begin{aligned}
sign\frac{\partial F^*}{\partial \alpha} &= sign\frac{\partial^2 U_p}{\partial F \partial \alpha} > 0 \\
sign\frac{\partial F^*}{\partial \beta} &= sign\frac{\partial^2 U_p}{\partial F \partial \beta} > 0
\end{aligned}
$$

$\square$

Proof of boundary conditions when $C(F)$ and $K(F)$ are concave:

PROOF. Since both $C(F)$ and $K(F)$ are concave, we have $C'' < 0$ and $K'' < 0$. Also, we had $C'(F) > 0$ (platform's cost increases in F) and $K'(F) < 0$. Therefore, the slope of $\frac{C'(F)}{K'(F)}$ in Eq. (13) is always positive, *i.e.,* $\frac{C'(F)}{K'(F)}$ is monotonically increasing in F. But the slope of $K(F)$ is monotonically decreasing in F, and hence there can be at most one $F^* \in [0, F^{max}]$ that will satisfy Eq. (13). Moreover, since $C''(F^*) < 0 < -n_d^*(F^*)K''(F^*) + 0.5(K'(F^*))^2$, this F^* will be a minimum. Therefore, the optimal choice of F^* is either at $F = 0$ or $F = F^{max}$. $\square$

D. BOUNDARY SOLUTIONS

Boundary solutions arise when the fraction of customers joining the platform on either market side is either 0 or 1. Such outcomes are typically associated with less interesting configurations, but are provided here for completeness.

Eqs. (6) and (7) are easily seen to imply that solutions of the form $(0, n_d^* > 0)$ or $(x_c^* > 0, 0)$ are not feasible. This is because $n_d^* = 0$ forces $p_c^* = 0$, which results in a negative profit for the platform. Similarly, when $x_c^* = 0$, the platform needs to subsidize developers $b_d^* < 0$, which again translates into a negative profit. So $(0,0)$ is the only feasible equilibrium in such cases, *i.e.,* the system parameters are such that the platform cannot be profitable.

Other possible boundary solutions arise when one side of the market reaches full penetration. There are three sub-cases to consider:

i. Both market sides are at full penetration ($x_c^* = 1, n_d^* = 1$): In this scenario, Eqs. (6) and (7) give $p_c^* = \beta$ and $b_d^* = \alpha - K(F) - 1$.

ii. Only the consumers side of the market is at full penetration ($x_c^* = 1, 0 < n_d^* < 1$): The adoption level on the developers side of the market is then given by $n_d^* = (\alpha + \beta - K(F))/2$, and the prices for the two sides are $b_d^* = (\alpha - \beta - K(F))/2$ and $p_c^* = \beta(\alpha + \beta - K(F))/2$, respectively. The constraints $0 < n_d^* < 1$ imply $K(F) < \alpha + \beta < K(F) + 2$, *i.e.,* cross-side benefits, as measured by $\alpha + \beta$, cannot be either too large or too small compared to development costs $K(F)$. When they are large, $(1,1)$ is the equilibrium.

iii. Only the developers side of the market is at full penetration ($0 < x_c^* < 1, n_d^* = 1$): The adoption level on the consumers side of the market is then given by $x_c^* = (\alpha + \beta)/2\beta$ (which requires $\alpha < \beta$), and the prices for the two sides are $b_d^* = (\alpha(\alpha+\beta) - 2\beta(K(F) - 1))/2\beta$ and $p_c^* = (\beta - \alpha)/2$.

As in the case of interior solutions, pricing is typically asymmetric and instances where the platform subsidizes one side and charges the other also occur. Boundary solutions arise mostly when cross-side externalities dominate other system parameters. In the remaining of the paper, the focus is on (the more interesting) scenarios where neither market-side has achieved full market adoption.

E. ALTERNATE UTILITY FUNCTIONS

E.1 Competition among Developers: Same-side Externalities

In our model we consider that the applications that developers innovate, although differentiated in their offering and characteristics, share some degree of homogeneity in terms of the underlying functionalities they use. Consequently, developers may have to compete for same advertisement revenue sources and consumers, and thus incur negative same-side externalities from each other's presence. To account for such situations, we show that our model can be easily generalized to include competition among the developers.

In the utility function of developers given in Eq. (2), we include a term $-\eta n_d$ where n_d is the fraction of developers on the platform and η captures the marginal externality that each developer imposes on the other. Therefore, the new utility function becomes

$$U_d = \alpha x_c - \eta n_d - b_d - (K(F) + \tau\phi) \tag{21}$$

In using Eq. (21) to compute the equilibrium adoption levels (as done in Section 4.1), the developer who is indifferent between joining and not joining the platform ($U_d = 0$) is obtained from $\widehat{\phi} = n_d$:

$$
\begin{aligned}
\tau\widehat{\phi} = \tau n_d &= \alpha x_c^* - \eta n_d - b_d - K(F) \\
\Rightarrow n_d(\tau + \eta) &= \alpha x_c^* - b_d - K(F)
\end{aligned} \tag{22}
$$

If all the parameters are normalized with respect to $\tau + \eta$, the resulting expression is identical to Eq. (5). Therefore, the qualitative outcomes observed from our earlier model do not change upon introducing competition among developers, only the quantitative values need to be adjusted with respect to the new normalization coefficient.

E.2 Consumer Heterogeneity in Intrinsic Benefits

In this section, we consider alternative consumer utility function that capture scenarios where all consumers have similar valuation for the number of available applications, but are heterogeneous in how they evaluate the platform's intrinsic qualities (*e.g.,* reliability, performance, brand name etc). This is typically true for platforms that deliver a strong core value and the availability of software applications are added bonuses. For example, consider a gaming platform (*e.g.,* Xbox 360) where almost all users value the availability of interesting games equally but are quite heterogeneous in how they perceive the platform's intrinsic quality, as determined by the screen resolution, controls, loading time etc. The utility function for the consumers in this scenario is:

$$U_c = \theta q + \beta n_d - p_c \qquad (23)$$

The term q stands for the intrinsic value of the platform to the consumers, and it is weighted by their individual preference parameter θ. We assume θ to be uniformly distributed in $[0,1]$, and that it value is a private information for each consumer, but its distribution is known. The term βn_d captures the cross-externality benefits that the consumers enjoy from the presence of n_d developers on the platform. As before, β captures the marginal externality benefit that each developer brings to the consumers. p_c is the flat fee paid to the platform. All the parameters of the model are appropriately normalized with respect to the customer population on each side and the maximum fixed cost that developers may incur (*i.e.,* $\tau = 1$). The utility functions for the platform and the developers remain the same, as in Eqs. (1) and (2), respectively.

Using the solution methodology outlined in Section 4, we get the following outcomes in the three stages:

Adoption Stage:

For a given functionality level, F, and a set of prices, p_c and b_d, for the consumers and developers respectively, the marginal consumer who is indifferent between joining and not joining the platform is $\widehat{\theta} = 1 - x_c = \frac{p_c - \beta n_d^*}{q}$ and similarly, the marginal indifferent developer is $\widehat{\phi} = n_d = \alpha x_c^* - b_d - K(F)$ At equilibrium, we have $x_c^* = x_c$ and $n_d^* = n_d$.

Pricing Stage:

As before, we solve the platform provider's profit maximization problem to find the 'optimal' prices for the two market sides and the equilibrium adoption levels at these prices. The interesting situation is one where neither market side has reached full adoption, *i.e.,* where the outcome of the equilibrium adoption is an interior solution of the maximization problem. The results for it are provided below.

The optimal price levels $(p_c^*, \ p_d^*)$ and the optimal adoption levels of consumers and developers $(x_c^*, \ n_d^*)$ of the two-sided market, which maximize the platform

provider's profit are given by

$$p^* = \frac{q(2q + (\alpha - \beta)K(F) - \alpha(\alpha + \beta))}{4q - (\alpha + \beta)^2} \qquad (24)$$

$$b^* = \frac{(\alpha - \beta)q - (2q - \beta(\alpha + \beta))K(F)}{4q - (\alpha + \beta)^2} \qquad (25)$$

$$x_c^* = \frac{2q - (\alpha + \beta)K(F)}{4q - (\alpha + \beta)^2} \qquad (26)$$

$$n_d^* = \frac{(\alpha + \beta - 2K(F))q}{4q - (\alpha + \beta)^2} \qquad (27)$$

For the above outcome to be an interior solution, $0 < \{2q - (\alpha + \beta)K(F), (\alpha + \beta - 2K(F))q\} < 4q - (\alpha + \beta)^2$ needs to be satisfied. The second order condition for the Hessian to be positive definite is also satisfied when the above inequality holds.

Design Stage:

The optimal level of built-in functionalities (F^*) for the platform which maximizes its profit is given by

$$\frac{C'(F^*)}{K'(F^*)} = \frac{2q}{4q - (\alpha + \beta)^2}K(F^*) - \frac{(\alpha + \beta)q}{4q - (\alpha + \beta)^2} \qquad (28)$$

$$\Rightarrow \frac{C'(F^*)}{K'(F^*)} = -n_d^*(F^*) \qquad (29)$$

$$C''(F^*) > -n_d^*(F^*)K''(F^*) + \frac{2q[K'(F^*)]^2}{4q - (\alpha + \beta)^2} \qquad (30)$$

As before with the utility functions of Section 4, we have the following result by using the conjugate pair theorem. The level of functionality investment by the platform increases with increase in cross-externality benefits enjoyed by either customer side, *i.e.,*

$$sign\frac{\partial F^*}{\partial \alpha} = sign\frac{\partial^2 U_p}{\partial F \partial \alpha} > 0 \Rightarrow \frac{\partial F^*}{\partial \alpha} > 0$$

$$sign\frac{\partial F^*}{\partial \beta} = sign\frac{\partial^2 U_p}{\partial F \partial \beta} > 0 \Rightarrow \frac{\partial F^*}{\partial \beta} > 0.$$

F. ADDITIONAL RESULTS

As alluded to in Section 5, Figure 7 shows that multiple interior local maxima may arise when the developer cost function $K(F)$ is convex and platform's cost function $C(F)$ is convex. Additionally, it shows that small changes in the relative rates of change in these cost functions can lead to drastically different outcomes in the the selection of the optimal number of features that a platform should have.
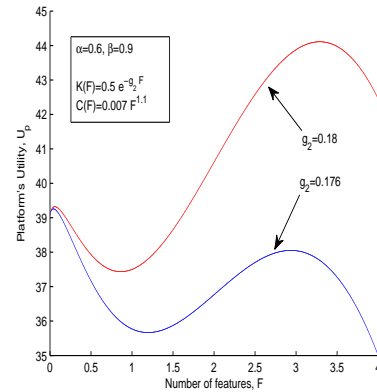

Figure 7: Multiple interior maxima for convex $K(F)$ and $C(F)$.