ASAC 2005
Toronto, Ontario

Min Lu
Economics Department
University of British Columbia

Yanbin Tu

Y. Alex Tung
School of Business
University of Connecticut

REPEATED TRIALS BEFORE BUY:

A STUDY OF COMPUTER SOFTWARE SAMPLING

Product sampling is an important marketing strategy in the software industry to promote product sales. Unlike other conventional physical goods sampling, software samplers typically require more than one trial before a final purchasing decision is made. This paper attempts to model the process of limited-trial-time and unlimited-trial-time dynamic software sampling.

Introduction

The penetration speed of global Internet access has been extraordinary in the past decade. This phenomenon results in a near zero-time distribution of digital goods from companies to their customers. It also further reduces the already low "production" costs, where digital goods such as computer software and music songs are stored in CD/DVD, cassette tapes, floppy disks or other similar media, to a negligible zero level. Many information goods companies are utilizing this fast, zero marginal cost distribution channel to conduct product sampling, through the releasing of free samples, to increase user base and boost product sales.

Besides the zero-time distribution and zero marginal cost characteristics, information/digital goods sampling possesses another feature differing from other physical product sampling, namely, durability. Traditional product sampling applies mostly to non-durable goods such as shampoo, cheesecake, and laundry detergent, while information goods are durable. Since there is possible substitution or cannibalization effect between samples and the products themselves, various approaches are taken to minimize such effect. For non-durable physical products, while the sample attains full quality of the product, companies usually set a restriction on quantity. For example, a shampoo manufacturer gives only one small bag of shampoo sample to each household. Digital goods sampling adopts a different approach in that it either limits its sample quality (functionality) or restricts the trial duration. For example, a partial song, a first-three-page e-book, limited functionality software, or fully functional software with 30-day trial time.

Due to the aforementioned differences, the strategies for physical product sampling and digital product sampling may be fundamentally different. While the majority of marketing literature on product sampling focuses primarily on physical, non-durable goods, increasingly more studies on the information goods sampling can also be found in recent years (Cheng and Tang 2004). Our focus in this work is on the economic analysis of software sampling.

Generally, we can divide information goods sampling into static sampling such as music songs sampling and dynamic sampling such as computer software sampling. In static sampling, a consumer tries information goods and derives her utility for the product sample. Since samples are typically free, there is implicitly a filtering mechanism within each consumer that converts her utility for the product sample into an expected utility for the real product. This expected utility takes into account all the exogenous factors such as the price and other costs for the real product. It is based on this utility a customer makes a purchase decision. Let's define U_S as the utility derived from information goods sampling, and U_G as the expected utility for the real information goods, ρ is the expectation filter that measures how a consumer converts her utility from sampling into expected utility of the real product. That is, $U_G = \rho U_S$. The consumer will buy the information goods only under the condition where $U_G - P \geq U_S$ where P is the total costs (including price) for the information goods. Hence, under monopolistic scenario and assuming price is the only cost, the maximum price the information goods vendor can charge is $P = U_G - U_S$. Figure 1 depicts the process of static information goods sampling.

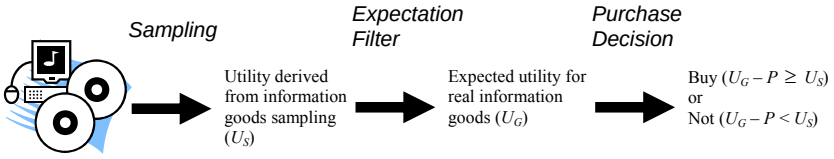

Figure 1. Process of Static Information Goods Sampling

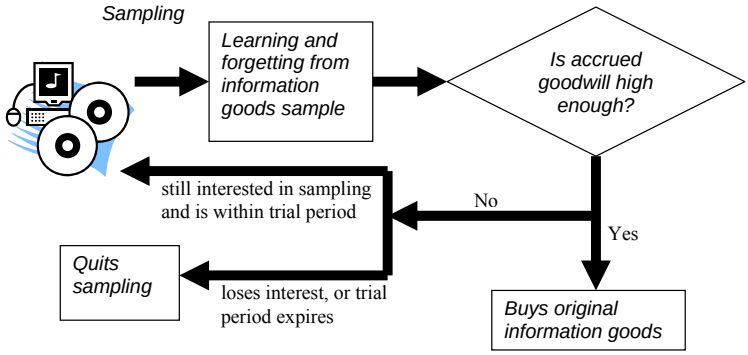

Figure 2. Process of Dynamic Information Goods Sampling

The process of dynamic information goods sampling is more complicated than that of the static one. After a consumer tries an information goods sample, she neither buys the information goods nor quits sampling right away. She keeps on trying the sample to learn more about the information goods. As this process continues, learning and forgetting effects occur. Learning is the process of increasing the goodwill toward a product, while forgetting is the depreciation of this goodwill (Heiman et al. 2001). Therefore, both U_G and U_S are constantly changing through the sampling process. Once the accrued goodwill reaches a level, the consumer will choose to buy the information goods, or, she might lose her interest during the sampling process and quits. In limited-trial-time sampling, if the sampler's goodwill has not been accumulated high enough to buy the product before the expiry date, she will be forced to drop out the sampling process. The process of dynamic information goods sampling is depicted in Figure 2.

Software sampling is considered dynamic since the trial of particular software does not typically start and end within one trial round. The potential consumer who tries the software sample needs to try the product for some duration before making purchase decisions. Other

product sampling, such as music songs and most physical goods sampling (e.g. shampoo, cheesecake, laundry detergent), however, results in buy or no buy decisions after one trial. This study tries to answer the following research questions: For software vendor, how to decide the optimal sample quality and the optimal legitimate trial period so that he can achieve highest sampling effectiveness? How do the parameters such as the free-rider tendency, the threshold utility of purchasing, the sampling time period, the learning speed and the forgetting speed affect the software sampling process?

Brief Literature Review

Freedman (1986) and Jain et al. (1995) find sampling useful when introducing a new product. Sampling is particularly important in the early stages of a product's life because the initial samplers might not only become the future consumers, but also potentially help to diffuse the new product information through "words of mouth" (Holmes and Lett, Jr., 1977). Stanton (1998) states that it is advisable for the food industry if consumers could taste samples of new food products while they are introduced into the market. Sampling not only raises the public's awareness and improves perception of a product, but also brings a direct experience that reduces the risks of new product uncertainties (Heiman, 2001; McGuinness et al., 1995; Conrad, 1976). Besides new products, Bettinger et al. (1979) argue that sampling can also be used to introduce existing products to new consumers. Some researchers argue that sampling is a more effective marketing technique than advertising. Wright and Lynch, Jr. (1995) demonstrate that direct product experience has an advantage over exposure to newspaper advertising. Marks and Kamins (1988) find that belief and attitudinal confidence are higher for consumers who are exposed only to product sampling than those who are exposed to product advertising alone. They also find product sampling creates higher order beliefs than indirect product experience does. Kempf and Laczniak (2001) show that samplers form more confidently held beliefs, and have higher expect value and purchase intention. Paine-Andrews et al. (1996) find product sampling with prompting and price reductions can increase customer purchases of certain lower-fat products. McGuinness et al. (1995) show that combined use of product samples and coupons promotes product more effectively than used individually. Rothschild and Gaidis (1981) and Stanton (1998) further suggest that sampling is a necessarily integral part of successful marketing campaign. Though research on product sampling abounds in marketing literature, little has looked at computer software sampling. Cheng and Tang (2004) study free trial software issues by focusing on network externalities and cannibalization effect. They do not consider software trial as a dynamic phenomenon. Although our study relates to Heiman et al.'s (2001) as it explores the dynamic sampling problem, ours differs from theirs as we focus on information product sampling while Heiman et al. discuss physical product sampling.

The Basic Model: Limited-Trial-Time Software Sampling

The software vendor wants the sampler to get to know the software features or functionality so that she accumulates her expected utility high enough, and finally makes a purchase decision on his product. During the sampling process, the software vendor hopes that the sampler' sampling efforts can bring her high total amount of the expected utility so that the sampler is enthusiastic at software sampling. However, the free software sample might have a free-rider problem. As the legislative trial period is longer and the sampler's expected utility is not high enough, she would rather uses the product sample instead of buying the original product. We use mq^2 to characterize the free-rider effect, where m measures the magnitude of the sampler's tendency to be a free-rider and q is the quality that the sampler perceives during sampling

process. q^2 is a convex function which suggests that it is a marginal increasing function. The sampling process can be interpreted as the process that the sampler discovers or explores the quality of the software. We can write the expected utility growth as $\dot{U}_G = a + bq - cU_G$ where a is a natural growth rate of the expected utility due to network effects, q is the quality perceived or discovered by the sampler, b measures the contribution rate of quality to the expected utility, c is a parameter to measure the forgetting (attrition) rate. We define the threshold utility of purchasing a copy of software while sampling as $\overline{U}$. That is, if the sampler's expected utility for the original software U_G reaches over $\overline{U}$, she has an incentive to buy the software product. Suppose the sampler's $U_G = 0$ at time 0, and the software vendor wants the sampler's $U_G = M > \overline{U}$ at time T. At the same time, the software vendor wants to maximize the sampler's expected utility for the real software product and minimize the free-rider tendency effect of the sampler. We use the combination of the expected utility minus the free-rider effect to measure the sampling effectiveness. Briefly, the software vendor tries to maximize the following objective function:

Max: $\int_0^T (U_G - mq^2)dt$ subject to: $\dot{U}_G = a + bq - cU_G$; $U_G = 0$ at time 0, that is, $U_G(0) = 0$; and $U_G = M > \overline{U}$ at time T, that is, $U_G(T) = M > \overline{U}$.

The above dynamic optimization problem can be explained this way: Given a fixed period time T, the software vendor tries to maximize the total sampling effectiveness (the accrued expected utility minus the free-rider effect) according to the sampler's expected utility growth rule. At the same time, the sampler's expected utility should be increased from 0 at the beginning of sampling to M ($>\overline{U}$) at time T, which means the sampler has an incentive to buy the product after her sampling from time 0 to T.

To solve this dynamic optimization problem, we need to set up the Hamiltonian function $H = U_G - mq^2 + \lambda(a + bq - cU_G)$, and get the following maximization conditions: (i) $\frac{dH}{dq} = -2mq + b\lambda = 0$ ($\frac{dH^2}{d^2q} = -2m < 0$ implies the dynamic optimization problem is the maximum one); (ii) $-\frac{dH}{dU_G} = \dot{\lambda} = -1 + c\lambda$; (iii) $\dot{U}_G = a + bq - cU_G$; (iv) $U_G(0) = 0$; (v) $U_G(T) = M$. We use *Maple 8* to solve the optimal time paths for both q and U_G.

$q^*(t)$ is a function of time t, and tells us the optimal route where the sampler discovers the quality of software prodcut through sampling. $U_G^*(t)$ is a function of time t as well, and describes the process where the sampler's expected utility for the real software prodcut changes with time t. The optimal time phrase of the discovered quality and expected utility can be depicted in Figure 3.

In the current software sampling practice, instead of dynamically releasing the quality of software, the software vendor usually releases the sample software with static quality. The optimal time path $q^*(t)$ clearly suggests the low boundary of the quality that the software vendor should release. From Figure 3, we can see that in order to obtain U_G equal M at the time T, the

software vendor must release at least the quality level $q^*(T) = Q$. This suggests that in order to let the sampler's expected utility be greater than the threshold $\overline{U}$, the software vendor should release the quality of software at least at the level of Q. Suppose he releases the quality of Q' instead, the sampler's expected utility only reaches the level of M' after the time T' till T, which is lower than M at the time T. This implies that the samplers cannot accrue the expected utility high enough to buy the original software product.

It should be worthwhile to point out that in the current software sampling practices the software vendor usually just releases the software sample to the potential buyers, but does not care about how the sampler "learns" the software. Our framework shows that there exists a time phase of $q^*(t)$ which is the optimal route for software sampling process. This supplies a very useful tool for the software vendor to design a "learning" tutorial for the sampler. The software vendor should design a software learning tutorial to help the sampler learn the software effectively and efficiently. The optimal time phase shows that if the software vendor designs a software learning tutorial following the optimal path, the sampler can effectively explore the software features and increase her expected utility high enough to buy the original product by the end of the time T.

Now, let us look at how the parameters m, M, T and b to affect the quality exploration and the accrued expected utility in the software sampling process.

Impacts of Higher *m* and *M*: The higher m implies that the sampler has a higher tendency to be a free rider, that is, she would rather use the sample software instead of buying the original product. To induce such a sampler to become a software buyer, she needs to explore higher quality of the software and correspondingly cherish higher expected utility. This is due to: $\frac{dq^*}{dm} = \frac{(e^{ct}-1)b}{2cm^2} > 0$ and $\frac{dU_G^*}{dm} = \frac{(e^{ct}-1)b^2}{2c^2m^2} > 0$.

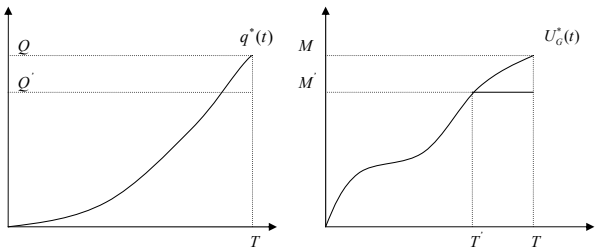

Figure 3. The Time Phase of Perceived Quality and Expected Utility

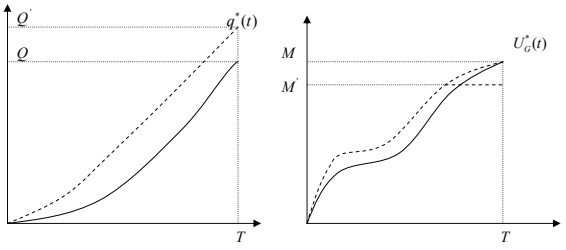

Figure 4. The Time Phase of Perceived Quality and Expected Utility

The impact of higher m can be described in Figure 4. As the sampler has a higher m, the software vendor must design the higher quality exploration route (the left dash-line) and correspondingly have the higher expected utility route (the right dash-line) in Figure 4. Another point is about the sample software quality required for higher m. In Figure 4, if the software vendor releases the quality at the level of Q, the sampler's expected utility only reaches at M'. Once he releases the quality at the level Q', the sampler's expected utility can reach the level of M. This implies that when the sampler has a high tendency to be a free rider, it is more challenging for the software vendor to use the product sampling strategy as he needs to develop and release software sample in higher quality.

The impact of higher M is similar to higher m. Higher M means that $\overline{U}$ is higher, which suggests that the sampler has a higher threshold of expected utility to buy the product. Therefore, if the sampler has a higher M, the software should release sample software in higher quality and design a higher learning tutorial curve for the sampler. Correspondingly, the sampler has a higher expected utility curve under the higher M. The proof is as follow: $\frac{dq^*}{dM} = \frac{ce^{ct}}{b(1-e^{-Tc})} > 0$ and $\frac{dU_G^*}{dM} = \frac{(e^{c(t+T)} - e^{-c(t-T)})}{e^{Tc}-1} > 0$. So, the impact of higher M is the same as the higher m shown in Figure 5.

Impact of Shorter *T:* Sometime, the software vendor hopes that the sampler can learn the software and accure higher expected utility within a shorter time period. What does this imply to his product sampling strategy? Our analytical framework shows that if the software vendor wants to shorten the sampling process, he must release a higher quality of software sample, and corespondingly, the sampler's expected utility increase at a higher speed. Mathematicaly: $\frac{dq^*}{dT} = \frac{e^{c(t-T)}Mc^2}{b(e^{-Tc}-1)^2} < 0$ and $\frac{dU_G^*}{dT} = \frac{(e^{-c(t-T)} - e^{c(c+T)})Mc}{(e^{-Tc}-1)^2} < 0$. The impact of a shorter T can be described in Figure 6.

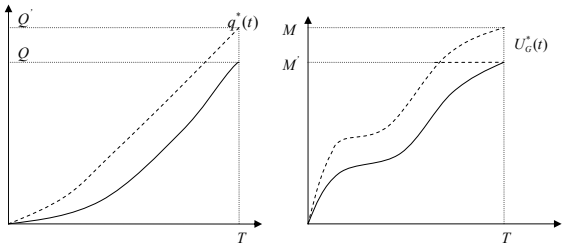

Figure 5. The Time Phase of Perceived Quality and Expected Utility

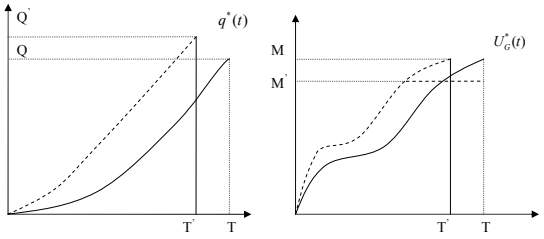

Figure 6. The Time Phase of Perceived Quality and Expected Utility

Due to a shorter T, the software vendor needs to release a higher quality Q' level for the software sample so that at time T the sampler can accrue the expected utility to reach M. Suppose the software vendor only supplies the sample software quality at Q, we can see the sampler gets the expected utility of M', which is lower than M. The higher quality exploration curve indicates that the software vendor needs to design a more dedicated software learning tutorial for the sampler.

Impact of Higher *b:* Higher b means that the quality exploration contributes more to the growth of the sampler's expected utility. The impact of a higher b on the expected utility straightforward: if the quality exploration contributes more to the expected utility, the sampler's expected utility is easier to reach the level of M. In Figure 7, the expected utility curve is lower than before. However, the quality exploration path depends on the value of c. Recall that c measures the attrition (forgetting) speed of expected utility that the sampler drives from the sampling process. Higher c means that the sampler is easy to forget what she learns about the software product, and lower c means she is not easy to forget what she learns about the product. Mathematically, $\frac{dU_G^*}{db} = \frac{(1-e^{ct})b}{mc^2} < 0$ and $\frac{dq^*}{db} > 0$ when c is high enough, and $\frac{dq^*}{db} < 0$ when c is low enough.

Figure 7 shows us the impact of higher *b* depends on the value of *c*. With a high *c*, the software vendor still needs to supply a higher quality for the sample software, and he needs to design a higher quality exploration curve for the sampler. If the software vendor only releases the quality at *Q*, the sampler only gets the expected utility at M' which is undesirable to the software vendor. On the other hand, when *c* is quite low, the software vendor might supply a low quality for the sample software. In this case, the software vendor is flexible to supply software sample. Therefore, how to design sample software so that the sampler is not easy to forget what she learns is an important topic to the software vendor. The software vendor might use multimedia, simulation, picture and other teaching tools to help the sampler better remember what she has learned before.

Unlimited-Trial-Time Software Sampling

In the current software sampling practice, some sample software is of unlimited-trial-time. We need to investigate under which condition the software vendor can adapt the unlimited-trial-time software sampling. The software vendor's objective function is as below:

Max: $\int_0^{\infty}(U_G - mq^2)dt$ subject to: $\dot{U}_G = a + bq - cU_G$ and U_G=0 at time 0, that is, $U_G(0) = 0$, and $U_G(\infty)$ is free.

To solve this dynamic optimization problem, set up the Hamiltonian equation: $H = U_G - mq^2 + \lambda(a + bq - cU_G)$. From the Hamiltonian equation, we get the following maximization conditions: (i) $\frac{dH}{dq} = -2mq + b\lambda = 0$; (ii) $-\frac{dH}{d\dot{U}_G} = \dot{\lambda} = -1 + c\lambda$; (iii) $\dot{U}_G = a + bq - cU_G$; (iv) $U_G(0) = 0$; (v) $U_G(\infty)$ is free. From conditions (i), (ii) and (iii), we get the following system of equations: $\dot{\lambda} = -1 + c\lambda$ and $\dot{U}_G = a + \frac{b^2\lambda}{2m} - cU_G$. Correspondingly, we get the dynamic systems: $\dot{q} = -\frac{b}{2m} + cq$ and $\dot{U}_G = a + bq - cU_G$. The equilibrium of the systems is $q^* = \frac{b}{2mc}$ and $U_G^* = \frac{a}{c} + \frac{b^2}{2mc^2}$. The phase diagram is shown in Figure 8. From the phase diagram, we can see the equilibrium is a saddle point. We get the following proposition:

Proposition 1: The dynamic system of $\dot{q}$ and $\dot{U}_G$ has a saddle point.

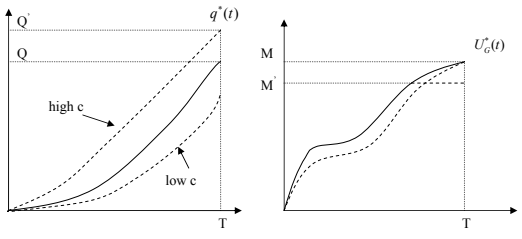

Figure 7. The Time Phase of Perceived Quality and Expected Utility

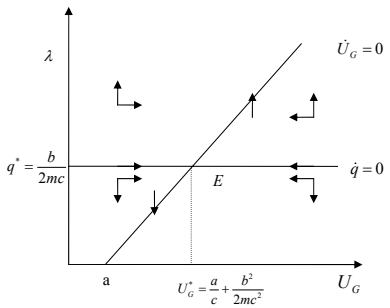

Figure 8. The Phase Diagram

Proposition 1 tells us that if the software vendor designs a software quality learning tutorial along the optimal time path of $q^*(t)$, the sampler will reach a stable point of $q^* = \frac{b}{2mc}$ and $U_G^* = \frac{a}{c} + \frac{b^2}{2mc^2}$. Obviously, whether $U_G^* = \frac{a}{c} + \frac{b^2}{2mc^2}$ is greater than $\overline{U}$ or not depends on the values of parameters *a, b, c* and *m*. Higher *a* and *b*, or lower *m* and *c*, will help to increase the stable value of U_G^*. The logic is straightforward: higher *a* means the higher natural growth rate of the expected utility; higher *b* means more contribution of quality exploration to the expected utility; lower *m* means the lower tendency to be free rider of the sampler; and lower *c* means the sampler has lower attrition or forgetting rate. This finding justifies the importance of anti-free rider solution and optimal software learning tutorial design when the software vendor implements the infinite horizon software sampling. From the above analysis, we can conclude that if U_G^* is greater than $\overline{U}$, it is worthwhile for the software vendor to release a sample software without trial time limitation. Otherwise, such the unlimited-trial-time product sampling strategy will be ineffective.

Conclusions

Product sampling is an important marketing strategy to promote product sales. Software is information good and durable good. Sampling software is quite different from sampling physical good which is non-durable good with quantity restriction such as shampoo. It is also different from sampling other information good which is static such as digital music.

In this study, we discuss the sampler's quality exploration route and the expected utility path during her sampling process while the software vendor tries to maximize the sampling effectiveness. Our analysis shows that the optimal quality exploration route supplies a guideline for the software vendor to release a low boundary quality level of sample software and design an optimal learning tutorial for the sampler. We also discuss the impacts of different values of the parameters such as the free rider tendency, the threshold utility of purchasing, the sampling time period, the learning speed and the forgetting speed in the dynamic sampling process. Generally, when the sampler has a high tendency to be a free rider or higher threshold utility of purchasing, the software vendor should release a higher quality and design a higher learning curve. When the software vendor wants a short sampling period, he needs to release a higher quality and design a higher learning curve as well. The impact of s higher learning rate depends on the forgetting rate, or we needs to consider the combined effect of the learning rate and forgetting rate. When the forgetting rate is high enough, the software vendor needs to release a higher quality and design a higher learning curve. On the other hand, when the forgetting rate is low, the software is quite flexible to release the sample software quality. Therefore, our analysis justifies the importance of software learning tutorial and effective software sample design to help the sampler remember what she learns during the sampling process. For the unlimited-time software sampling, we show that the equilibriums of the perceived quality and expected utility form a saddle point. We conclude that only if the expected utility in equilibrium is greater than the threshold utility of purchasing, it is worthwhile for the software vendor to release sample software without trial time limitation. Otherwise, unlimited-trial-time software sampling strategy will be ineffective.

References

Bettinger, C.O., Dawson, L.E. and Wals, H.G. "The Impact of Free-Sample Advertising," *Journal of Advertising Research*, 19, (1979), 35-39.

Cheng, H.K. and Tang, Q.C. "Free Trial or No Free Trial: Optimal Software Product Design with Network Externalities," *Working Paper*, University of Florida, 2004.

Conrad, S.A., "Sampling Information in New Product Marketing," *Omega*, 4(1), (1976), 93-96.

Freedman, A.M., "Use of Free Product Samples Wins New Favor as Sales Tool," *Wall Street Journal*, 28, (1986), 19.

Heiman, A., Mcwilliams, B., Shen, Z. and Zilberman, D. "Learning and Forgetting: Optimal Product Sampling Over Time," *Management Science*, 47(4), (2001), 532-546.

Holmes, J.H. and Lett, J.D. Jr., "Product Sampling and Word of Mouth," *Journal of Advertising Research*, 17(5), (1977), 35-40.

Jain, D., Mahajan, V. and Muller, E. "An Approach for Determining the Optimal Product Sampling for the Diffusion of a New Product," *Journal of Product Innovation Management*, 12(2), (1995), 124-135.

Kempf, D.S. and Laczniak, R.N. "Advertising's Influence on Subsequent Product Trial Processing," *Journal of Advertising*, 30(3), (2001), 27-38.

Marks, L.J. and Kamins, M.A. "The Use of Product Sampling and Advertising: Effects of Sequence of Exposure and Degree of Advertising Chain Exaggeration on Consumers' Belief Strength, Belief Confidence, and Attitudes," *Journal of Marketing Research*, 25(3), (1998), 266-281.

McGuinness, D., Brennan, M. and Philip, G. "An Empirical Test of Product Sampling and Couponing," *Journal of the Market Research Society*, 37(2), (1995), 159-170.

Paine-Andrews, A., Francisco, V.T. , Fawcett, S.B., Johnston, J., and Coen, S. "Health Marketing in the Supermarket: Using Prompting, Product Sampling, and Price Reduction to Increase Customer Purchases of Lower-Fat Items," *Health Marketing Quarterly*, 14(2), (1996), 85-99.

Rothschild, M.L. and Gaidis, W.C. "Behavior Learning Theory: Its Relevance to Marketing and Promotions," *Journal of Marketing*, 45, (1981), 70-78.

Stanton, J.L., "Let Consumers Ample the Good!" *Food Processing*, 59(12), (1998), 71-72.

Wright, A.A. and Lynch, J.G. Jr., "Communication Effects of Advertising versus Direct Experience when both Search and Experience Attributes are Present," *Journal of Consumer Research*, 21(4), (1995), 708-718.