ABSTRACT

Title of dissertation: INTEGRATING SOCIAL NETWORK EFFECTS in PRODUCT DESIGN and DIFFUSION

Dilek Günneç, Doctor of Philosophy, 2012

Dissertation directed by: Professor Subramanian Raghavan
Robert H. Smith School of Business

Connectivities among people are amplified with recent advancements in internet technology increasing the number of communication channels. Information spread over these networks strengthen the social influence among individuals and affect their purchasing decisions. In this thesis, we study three problems in the product design and diffusion context by integrating such social network effects where influence takes place over neighborhood relationship ties among the users of the product. We consider the setting where peer influence plays a significant role in a consumer's product choice or there is a tangible benefit from using the same product as the rest of one's social network.

Building upon the well-known Share-of-Choice problem, we model an influence structure and define the Share-of-Choice problem with Network Effects. It is an NP-Hard combinatorial optimization problem which we solve using a Genetic Algorithm. Using simulated data we show that ignoring social network effects in the design phase of a product results in a significantly lower market share for a product. Our genetic algorithm obtains near-optimal solutions and is very robust in terms of its running time, scalability, and ability to adapt to additional constraints/variations of the model. In this setting, we introduce a product diffusion problem, the Least Cost Influence Problem, which increases the market share of a product by intervening the natural diffusion of it over the social network. This intervention is in the form of incentive supply to a group of people in a least costly way while maximizing the spread of the product.

We generalize the Least Cost Influence Problem by moving away from the marketing setting and by treating the previous product as any piece of "information" that can spread over a social network by adoption. We show that this problem is polynomially solvable over tree networks under some conditions. We provide a Dynamic Programming algorithm to solve this problem and show that it can be interpreted as a greedy algorithm that gives incentives starting with the people that are least influenced by their neighbors, albeit the definition of susceptibility to influence from neighbors is updated throughout the algorithm.

We introduce a two dimensional influence model and extend our modeling and solution methods for the product line design problem which involves designing multiple products within the same product line with the objective of appealing to the heterogeneous structure of the market. The first dimension of influence is the affection of individuals from using the same product, and the second dimension is the influence of using a similar product from the same product line which has a lower intensity of influence. We reexamine the Least Cost Influence Problem in the product line setting.

INTEGRATING SOCIAL NETWORK EFFECTS in PRODUCT DESIGN and DIFFUSION

by

Dilek Günneç

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2012

Ailem Şeniz, Erol, Şenol Günneç ve Şahin Danış'a.

Acknowledgments

Studying for a Ph.D. has been one of the most satisfying and challenging responsibilities in my life. It could never have been possible without the many who have helped and supported me throughout these five years.

I cannot justly express my sincere appreciation to my advisor, Professor Raghu Raghavan. I thank him for his guidance and for sharing his knowledge and experience. I have always felt challenged, inspired and enthusiastic about research after our meetings. He taught me a lot about approaching a problem, asking the right questions, managing time, and overall about how to be a good researcher.

I am thankful to Professor Bruce Golden for his valuable and motivating feedback on my thesis and on my research in general. I would like to thank Professor Yi Xu and Professor William Rand for critical reading of this thesis and for their helpful comments, and Professor Shapour Azarm for agreeing to be on my dissertation committee.

Most of a Ph.D. student's time is spent at either school or home. I thank all my friends for all the good times we had together and for making this an enjoyable experience; My classmates Ali, Yongjae, my officemate Gisela, my dear friends Merve and Ahmet, who were always generous to open their houses to me, and my dear roommate Didem, whom I thank for her care and friendship and who has become a part of my family.

Distance did not prevent Şahin from being there for me whenever I needed him. I have always felt his love and encouragement, and I cannot thank him enough.

Last but not the least, I would like to thank my family for their never ending love and support. I'm excited to go back home and to be with them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Problem Description

Can consumer buying decisions be isolated from their social networks? We consider the setting where peer influence plays a significant role in a consumer's product choice or there is a tangible benefit from using the same product as the rest of one's social network. Take for example communication tools. Family plans encourage using the same phone network by offering large discounts to customers for in network calls. When a person has to choose between products, it is natural to take into account the positive network effects (or peer influence) from their social networks in addition to a product's attributes. In the extreme case, if the communication is carried over the internet, such as voice or video chat, it is necessary for users to install the same application (e.g., Skype). We are focused on influence over neighborhood relationships which takes into account the social ties among the users of the product and therefore is different from the traditional models with positive network externalities described in Katz and Shapiro (1985) or Parker and Van Alstyne (2005). We use the term "neighbor" throughout the dissertation to represent the individuals who are connected to the person of interest and the terms "link" and "edge" to represent these connections. When presented on a network, the individuals are referred to as "nodes".

Recent advancements in social media allow better access to social networks of consumers. Interactive information sharing among customers is becoming fast and convenient over the internet with the online social networks, blogs and review sections of shopping websites. Online and offline reviews of products emerge as the first source of reference before making a purchase decision when interested in a newly launched product. A friend's (who has similar preferences or concerns) experience with a product can ease the searching process among a variety of product alternatives, especially for complicated technological products. Innovations such as mobile applications ask the users to log in using their existing Facebook accounts. The online gaming industry is definitely a good example of the business model which harvests social network effects among their consumers into their business. Zynga (with a value of more than $7 billion in December 2011) offers a large variety of games which can be played by people over different online social media. The games emulate a real life experience where players are allowed to communicate (send a message or a drink) with each other over an online social network. It is one of the examples of platforms where the users of a product (a game in this case) are explicitly connected with each other over a social network which makes it easier to collect information on the relationship network among the customers.

The exploding reach of the web and the prevalence of social networking sites, which in turn have made large amounts of data on social networks easily available to marketers, has only recently resulted in their recognition as an important tool for marketing (Van den Bulte and Wuyts, 2007). Because the market is shifting to the online environment and because of the competitive nature of the industry, it is important for marketing departments to benefit from such information with appropriate marketing strategies. Being able to analyze a social network provides marketers a competitive advantage in terms of forecasting the spread of product influence and intervening at times with promotions or incentives to strengthen this process. Predictions on market share (the number of people who will purchase the product) are critical when a new product enters the market and can create a difference for businesses.

1.1.1 Social Influence on Purchase Decisions

Since the early 1950s, it is accepted that people behave similar to a certain frame of reference that is produced by the groups they belong to (Merton and Rossi, 1949). More specifically, a reference group is defined as a person or a group of people that significantly influences an individual's behavior (Bearden and Etzel, 1982). People change their ideas or behavior according to different reference groups. In terms of opinion formations, according to Kelman (1961), social influence has three distinct processes. *Internalization* happens when an individual accepts influence because it is perceived as inherently helpful to maximize his values/goals. *Identification* occurs when an individual adopts a behavior because this behavior is associated with a self-defining relationship with that adopter group. *Compliance* is observed when the individual conforms with the rest of the group for a reward or to avoid punishment.

These social effects have been studied by researchers from different disciplines on different types of behavior. Such influence has been termed variously such as bandwagon effect, peer influence, neighborhood effect, conformity and contagion (Iyengar et al., 2009). For purchasing decisions, the reference groups considered previously were very general groups such as the social class a person belongs to or a group of people that had a similar educational background. These groups were too large to observe purchase decision effects, and therefore not reflective of the real life. Product influences are more about the people that interact frequently and have closer relationships, family members, friends and colleagues, and role models. The most recognized marketing discussion of reference group influence is by Bourne (1957) in which the distinction between product and brand decisions is stressed. Bourne (1957) argued that the most general attribute that is susceptible to reference group influence on product and brand decisions is through conspicuousness and stated the two forms of conspicuousness as the degree of exclusivity and the extent to which it is identifiable, respectively. Park and Lessig (1977) defined three dimensions of influence from a reference group. i) *Informational:* People facing uncertainties seek information from their reference group to make their decisions. ii) *Utilitarian:* People comply with the wishes of others to achieve rewards or avoid punishment. iii) *Value-expressive:* People need to feel psychologically associated with a person or a group and this is achieved by the acceptance of positions expressed by others. Based on the framework by Bourne (1957), Bearden and Etzel (1982) conducted a mail survey and investigated reference group influence on product and brand purchase decisions by examining the interrelationships among two forms of product use conspicuousness and the three types of reference group influence (Park and Lessig, 1977).

Burnkrant and Cousineau (1975) showed that people may use the product evaluations of others as a source of information about products. They conducted an experiment with participants who saw evaluations of other people about a coffee type as a test group. The authors found that after observing other peoples' positive evaluations of a product, people perceive the product more favorably themselves than they would in the absence of this observation and suggested that the same phenomenon occurs regularly in shopping situations.

The influence of reference groups on consumers' connections to brands are studied by Escalas and Bettman (2003). The authors showed that reference groups can be a source of brand associations reflecting a mental description of self for a consumer. They conducted two studies examining undergraduate students attitudes and beliefs about prototypical student types and their connections to brands. They indicated that the students build a connection to a brand when there's a connection with that brand and a reference group which the students associate their self images with.

A change in the evaluation of a product can also be caused by affective reasons. An interesting study is conducted by Howard and Gengler (2001) on emotional contagion. The authors investigated whether and how one person's emotional state can influence another person's evaluation of a product. In their experiment, they manipulated the emotion of a gift sender (happy vs. neutral) and a receiver was led or not led to believe that the sender provided her with a gift, inducing liking of the sender (liking vs. neutral). The sender and the buyer evaluated the product when brought together. The results showed that emotional contagion can lead to a positive bias on consumers' product evaluation, i.e., when senders were happy and receivers liked the senders, receivers' evaluations were most favorable.

Diffusion of innovation is the process by which an innovation is communicated through certain channels over time among the members of a social system (Rogers, 1983 First Ed., 1962). The model proposed by Bass (1969) has been one of the most important and influential diffusion models and it has been used frequently in practice. According to the Bass Model, the conditional probability of adoption at time t is increasing in the fraction of the population that has already adopted, so with Bass Model one can predict the number of users who will adopt an innovation at a given time t. In Bass Model, the rate of adoption is a function of the current proportion of the population who have adopted (Hill et al., 2006). Whereas the Bass Model portrayed a collective approach, there are some later studies in the 1980s that developed diffusion models by specifying adoption decisions at the individual level. Mahajan et al. (1990) provided a comprehensive review of the early stages of new product diffusion models in marketing. In these models, an adopter's utility for an innovation is dependent on what he thinks of the innovation in terms of its performance, value or benefits. But people's perceptions change over time as they learn more about the innovation from external and internal sources and when the utility is greater than the utility from the status quo, they adopt. In this respect, the Bass Model does not take into account stages of adoption, treating it as a binary process by avoiding the intermediate stages such as awareness and knowledge of the product.

Berning and Jacoby (1974) examined the decision making process before purchasing an innovation and the process before purchasing an established alternative. They concluded that there is a difference and that this difference lies primarily in the search for information from friends for the innovative products. They claimed that the patterns of information acquisition were surprisingly similar across product categories.

As social networks are in the center of social influence, graph theory is an important tool for understanding the structural properties. Nair et al. (1995) discussed this in their paper and argued that although many studies talk about social networks, more precise definitions of graphs, network structures and types of social relationships were required.

Taking a graph theoretical approach, they studied brand choices of individuals in a specified social relationship by comparing it with those individuals who were not in that social relation. For the social network in their paper, they used a real social network of a sorority group. In a later study, Reingen and Kernan (1986) studied network structures concentrating on strength of ties in a social network. Graph theory analysis were also used for addressing recommendation algorithms by Mirza et al. (2003).

Wu and Lee (2008) studied social influence on online impulsive purchases, i.e., unplanned purchases, in an online setting to see if the social network effects persist. In their experiments, two worlds are created as physical (friends, reference group) and virtual (online friends, reviews, online information) and they measured happiness, disappointment, willingness to buy and willingness to pay of the potential customers. Their empirical results showed that people in social groups generally perceive stronger willingness to buy, happiness, disappointment, and willingness to pay than the non-social group. They claimed that these results clearly demonstrated that the effects of social comparison could create a significant influence on consumers online purchasing behavior.

In a working study, Iyengar et al. (2009) studied whether friends influence purchases in a social network. They used two different types of data collection method to collect information from a Korean social network site. The first one is the geographic proximity and the second one is self reports. The authors used a choice and quantity model that captures the effect of social influence on a member's decision to purchase. They quantified the social influence in terms of changes in purchase probability and revenues.

More recently on experimental studies, Trusov et al. (2010) found that, on average, approximately one-fifth of a user's friends actually influence his or her activity level on the site. Iyengar et al. (2011) analyzed a social network among physicians and showed that there is a positive contagion effect at work in physicians' decisions to adopt a new drug which explicitly considered the role of social network structure and opinion leadership.

Social influences have been studied in a variety of topics in the economics literature. Leibenstein (1950) described the bandwagon effect as the increase in demand for a commodity because some or all other individuals in the market also demanded more of the commodity. However, the author also proposed a "diminishing marginal external consumption effect" which means that for every individual there is a point where the external effect equals zero. Theoretical models have also been developed in the economics literature where social interactions have been studied as a coordination process among individuals. Blume (1993) studied stochastic strategy revision processes in the interaction among players in large populations.

Katz and Shapiro (1985) studied network externalities, discussed examples for possible sources of positive consumption externalities and modeled them in a competition setting. Besides the direct physical effect of the number of people using the same communication technologies, they discussed indirect effects such as compatibility issues and post purchase services (which depend on the experience and the size of the service network) for durable goods.

Especially with the increased use of technological tools, the reasons why people imitate the behavior of others is explained as informational effects and direct-benefit effects, (Easley and Kleinberg, 2010). The first type of effect is similar to previous definitions of people seeking to learn more about the product. The second type of effect includes an extra benefit when you align with the behavior of others. In traditional economics, a person makes a decision to buy when the price of the product is less than his reservation price. The authors argued that when network effects are present (especially for technological tools, where interaction and compatibility with others can be important), a potential purchaser takes into account both her own reservation price and the total number of users of the good. They modeled this using two functions; when a z fraction of the population is using the good, the reservation price of consumer x is equal to $r(x)f(z)$, where $r(x)$ is the intrinsic interest of consumer x in the good, and $f(z)$ measures the benefit to each consumer from having a z fraction of the population use the good. The function $f(z)$ is increasing in z controlling how much the value increases as more people use it and they use a multiplicative model so that those who place a greater intrinsic value on the good benefit more from an increase in the fraction of the population using the good than those who place a smaller intrinsic value on the good. The authors also discussed that to convince a first group of people to buy, some cost can be incurred to give free samples/coupons to customers or to the fashion leaders who will promote the product simply by using it.

In this thesis, we acknowledge the social interaction and the influence among people over social networks and propose an influence model which specifically concentrates on utility increase for an individual as a result of the influence from other adopters which are directly connected to this individual.

1.1.2 Product Design Problem

Although product design is a well-studied subject in marketing and the implications of network externalities on several aspects of marketing (including customer behavior and market structure, product-related decisions such as pre-announcements, timing of product introductions and product differentiation and market entry) have been explored (Srinivasan et al., 2004), social network effects have generally been ignored within the product design process. In terms of social network effects, the organizational structure and the management of new product development teams have been studied (Leenders et al., 2003; Sosa et al., 2004) with respect to their relation with the design of a product and the creativity involved in the process.

Product design without network effects has been of significant interest both in academic studies and industry applications. Market share forecasts and estimated customer utilities for a product profile allow for a better understanding of the market needs and can lead to better product designs for the companies. Calculation of perceived values for product features have drawn attention from market researchers for many years. Conjoint analysis is one of the most popular tools in new product design for identifying customer preferences and utilities for attribute levels of a product called *part-worth utilities* (Green and Rao, 1971). It has been studied widely in the marketing literature and has been used to design many different products in practice. Broadly, there are two main steps in a conjoint analysis. The first is the data collection from consumers and the second is the analysis of this data to obtain part-worth utilities for each customer on each attribute level. After this analysis, part-worth utilities data are used to design the product. Earlier literature on marketing has focused on these two steps, especially in valid data collection and improved statistical estimation. However the natural next step of optimally using such data via conjoint *optimization* to design a product with maximum market share has been given relatively less attention (Camm et al., 2006). The first optimization approach using conjoint data was proposed by (Zufryden, 1977). In this regard, the share-of-choice (SOC) problem is defined in the product design process where the design corresponds to the selection of levels for each attribute of a product. This is necessary if the number of attributes and levels of the attributes are large and most product designs arising from different attribute level combinations are technologically and economically feasible (Nair et al., 1995). The objective is to create the product profile that will return the largest market share. It finds the best (optimal) design among different possible product profiles.

Part-worth utilities are used as inputs for the share-of-choice problem to optimize the selection of levels for each attribute. In this problem, one buys a product if and only if the utility the person gets by using the product is greater than or equal to her/his "hurdle". The "hurdle" is the utility value at which one would be indifferent between making a purchase or not making a purchase. It could also be seen as a status-quo product. In this dissertation we model the social network influence effects on product design by making adjustments to the well studied share-of-choice problem. The following notation is used in the model to state the share-of-choice problem. Let $V = \{1, 2, \ldots, n\}$ denote the set of people in the market, h_s denote the hurdle utility of person $s \in V$, K be the number of attributes, L_k be the number of levels for attribute $k = 1, 2, \ldots, K$, and u^s_{kl} denote the part-worth utility for person s if level l is chosen for attribute k. There are two types of binary decision variables; x_{kl} equals 1, if level l has been chosen for attribute k and is 0, otherwise; and y_s equals 1, if person s decides to buy the product and is 0, otherwise. The share-of-choice (SOC) problem is then formulated mathematically as follows.

$$\textbf{SOC:} \quad \text{Maximize} \quad \sum_{s=1}^{n} y_s, \tag{1.1}$$

$$\text{subject to} \quad \sum_{k=1}^{K}\sum_{l=1}^{L_k} u_{kl}^{s} x_{kl} \geq h_s y_s \qquad s = 1, 2, \ldots, n, \tag{1.2}$$

$$\sum_{l=1}^{L_k} x_{kl} = 1 \qquad k = 1, 2, \ldots, K, \tag{1.3}$$

$$y_s \in \{0, 1\} \qquad s = 1, 2, \ldots, n, \tag{1.4}$$

$$x_{kl} \in \{0, 1\} \qquad k = 1, 2, \ldots, K, \quad l = 1, 2, \ldots, L_k \tag{1.5}$$

In this share-of-choice model, the objective is to maximize the market share. (If the objective is to maximize the total utility of the customers or the firm's marginal return, the problem is called the "buyer's welfare" or "seller's welfare" problem.) Constraint (1.2) guarantees that people buy the product if and only if their utilities from the product exceed their hurdles. Constraint (1.3) ensures that each attribute is assigned only to a single level. Note that no network effects are taken into account in this model.

The share-of-choice problem has been studied in the marketing literature and shown to be an NP-hard problem (Kohli and Krishnamurti, 1989). Several heuristics have been used as solution approaches including a divide-and-conquer heuristic (Green and Krieger, 1989), greedy search and dynamic programming based heuristics (Kohli and Krishnamurti, 1987, 1989), a genetic algorithm (Balakrishnan and Jacob, 1996), a pruning heuristic (Tarasewich and McMullen, 2001) and a nested partitioning algorithm (Shi et al., 2001). An exact branch-and-bound algorithm (Camm et al., 2006) has been applied more recently for the share-of-choice problem.[1] For the product line design problem (designing multiple products of the same product line) Belloni et al. (2008) provided a Lagrangian relaxation method followed by a branch-and-price approach by Wang et al. (2009). Belloni et al. (2008) also provided a review of methods for product line optimization in general. None of the previous studies consider integrating the network effects. To our knowledge, this is the first study to explicitly incorporate social network (or peer influence) effects among potential customers in the share-of-choice problem and product design.

In a recent pioneering study, Narayan et al. (2011) considered three behavioral mechanisms of how consumers' product choice decisions may be affected from influence of their neighbors. We explain these mechanisms in Chapter 2 when we introduce our influence model.

From the product design literature, more recently Aral and Walker (2011) acknowledged the effects of viral marketing and argued that such viral features can be engineered during the launch of the product (We discuss a brief literature on viral marketing in Chapter 3). The authors differentiate the "viral characteristics" and "viral features" of a product. The first relates to the content of the product whereas the second corresponds to how the product is shared and how the features allow relationships with the other consumers. Concentrating on the viral features they showed that whereas the personalized referrals are more effective in encouraging adoption, passive-broadcast viral messaging is used more often and therefore causes a larger overall adoption. In this dissertation, we focus more on the viral characteristics of a product implicitly by looking at the changes in

[1]We should note that some commercial packages for conjoint analysis now provide optimization capabilities using some of the heuristics mentioned (Sawtooth Software, 2003).

the utility consumers get from using the same product with their neighbors. We consider the adoption as a passive-broadcast viral message which increases utilities for the other consumers.

We discuss the *combinatorial optimization* problem at the design phase of a product (after conjoint analysis has been completed for the product) with the same objective, maximizing market share, but we also *explicitly include social network effects* that take place at the product adoption stage, after the product is launched. For example, in a Zynga environment, the share-of-choice model with network effects would provide an applicable framework for designing new games as the characteristics of the problem in terms of the number of feasible attributes and levels are large and product use by people over online social networks are explicitly present. In this case the design that best fits the customers can increase the market share significantly. We aim to integrate the social network effects within the share-of-choice problem (both single and multiple product cases) and propose a genetic algorithm to provide high-quality solutions for these problems.

We contribute to the product design literature by showing with computational studies that the integration of the social network effects in the design phase of a product may lead to a different optimal product design and consequently to a larger market share. And this increase can be guaranteed and further strengthened by intervening in the diffusion process by providing tailored incentives to a group of individuals over the network. We then provide an efficient method to optimize the diffusion process. The innovation in the methodology of this dissertation is the integration of an exact mathematical modeling to the genetic algorithm to solve the product design problem and introducing two algorithms (dynamic programming and an iterative procedure) to solve a computationally challenging problem arising in product adoption. The model with network effects is complicated due to the dynamic relationship of the people (whether a person buys the product or not changes depending on the buying decision of their connections) and hence the traditional methods are inadequate to evaluate a product profile. We use a Matheuristic approach, which we explain in the following chapters, to overcome this problem. The second problem (Least Cost Influence Problem) introduced in Section 2.2 addresses a problem of a larger scope which may be applicable in different environments such as epidemiology, as explained in Chapter 5.

1.2 Overview of the Dissertation

We discuss the three main problems in this dissertation which lie at the interface of operations management and marketing research. The dissertation is organized as follows.

In Chapter 2, we develop our model on how to incorporate social network effects into the share-of-choice problem. We show that by modeling the product design problem with network effects, a product profile that reaches a much larger market share can be designed. We introduce a complementary secondary problem called the Least Cost Influence Problem (LCIP), which takes place when the product is on the market and its adoption is spreading over the network. We show that to attain the desired market share, it may be necessary to provide some incentives (such as free samples or discount coupons) to a group of people over the network during this adoption period. For both of these computationally challenging problems, we propose smart solution methods; a genetic algorithm to solve the first problem and an iterative approach that preserves optimality to solve the second problem. Using extensive computational studies, we discuss the performance of the methods.

In Chapter 3, we build upon the second problem introduced in Chapter 2 by describing it for general networks and a more general setting. It addresses contagion models and methods to determine critical nodes over a social network, which are of significant interest in marketing and epidemiological settings. We show that this problem reduces to the well-known NP-Hard knapsack problem for some cases, and propose a dynamic programming algorithm to solve it for a special case; when the social network is a tree. Our dynamic programming is a polynomial time algorithm which benefits from the star-shaped sub-network structures that are frequently observed on social networks (such as networks of authors of Wikipedia articles and networks of retweets from Twitter users). The least cost influence problem (LCIP) has a correspondence with the influence maximization problem (Kempe et al., 2003) that has been studied in computer science literature. Both problems attempt to identify a set of nodes over a given network such that the diffusion over the network is maximized. However the LCIP differentiates from the existing influence maximization literature as it allows for tailored incentives (incentive-discrimination) to be given to potential consumers.

In Chapter 4, we extend our model for the single product design problem to the product line design problem. There are multiple products in a product line and the product line design problem finds the optimal combination of products to appeal to the heterogeneous choice structures within the market. For a review of methods for product-line optimization in general, the reader is referred to Belloni et al. (2008). In that setting, there is a constraint of type (1.2) for each product. Since the influence structure can be augmented by the additional influence among users of different product types the model is slightly more complicated than that of a single product design. We model it as a two dimensional influence structure. We explain how our genetic algorithm solution method can be adapted to work with multiple products at various levels of the algorithm. We also explain the changes that will be observed when we encounter the least cost influence problem.

In Chapter 5, we discuss how the least cost influence problem that we study in a marketing setting can be altered to make it amenable to analysis in an epidemiological setting. We also develop two models as extensions to our product design model with network effects including a model with budgetary constraints and a comprehensive model that combines the product design and diffusion problems.

In Chapter 6, we discuss directions for future research and conclude by summarizing the dissertation and its contribution to the existing literature.

Chapter 2

Incorporating Social Network Information in Product Design

In this chapter, we discuss our analytical model to incorporate social network effects in the share-of-choice problem and elucidate why this can be critical in product design. We also explain why past solution procedures for the share-of-choice problem unfortunately cannot be easily adapted for the share-of-choice problem that incorporates network effects.

In Section 2.3, we present a genetic algorithm to solve the share-of-choice problem with network effects. A novelty of this heuristic procedure is that the fitness (or quality) of a solution is ascertained via an integer program (i.e., by using an exact solution procedure) which allows it to achieve very high-quality solutions. Section 2.4 provides the results of a computational study demonstrating both the speed of the calculations and the high quality of the solutions provided by the genetic algorithm.

A secondary problem, the least cost influence problem (LCIP), is also introduced as a complementary model using a subset of the network used for the share-of-choice problem. It is the problem of identifying the cheapest way of influencing individuals in a social network to achieve the market share associated with a given product design. This problem is of independent interest, as it addresses contagion models and the issue of determining influential nodes in a social network, which are of significant interest in marketing and epidemiological settings.

2.1 Share-of-Choice Problem with Network Effects

Consistent with social contagion research (Bell and Song, 2007; Manchanda et al., 2008), we follow a multiattribute linear utility-maximization approach. We focus on cases where an individual can only observe the outcome of a consumer's purchase decision, but not the relative preferences among each attribute level for three reasons. First, it is less complicated to collect data on purchase decisions among large social network groups using sales data than information about consumers' relative preferences among attributes (in fact due to privacy concerns it may not be feasible to do so). Secondly, potential consumers are exposed to peers' decisions about a product over online and offline social networks frequently since such information sharing requires less proximity or intimacy among consumers. Thirdly, this approach allows for a tractable and solvable model for the product design problem while still providing a well-established model to include social network effects. Our mechanism to model social network effects processes the influence from neighbors as an additional attribute of the product. Under this mechanism, the influence from other consumers adds to (or detracts from) product utility of a consumer in a linear additive manner.

In their recent study, Narayan et al. (2011) considered three behavioral mechanisms of how consumers' product choice decisions may be affected from influence of their neighbors. The first is a Bayesian mechanism where a consumer's updated preference for an attribute of the product is a weighted average of her initial (prior) preference and the preferences of their neighbors for the same attribute. The second mechanism is a more generalized Bayesian mechanism and allows for a more flexible process of preference revision. Finally, the third mechanism, which is based on the literature on social contagion and identical to our approach, abstains from updating the relative attribute preferences. Although, they suggested that their first model fits their particular data set (one study of MBA students in the same class) best, Narayan et al. (2011) also found that the mean extent of influence of neighbors' choices on consumer utility is positive and significant. Thus, the choice of a product profile by an influencer leads to an increase in the utility of that profile for the influenced consumer. Narayan et al. (2011) agreed that their study might be less representative of the cases where influence among individuals exclude choice-related information sharing and when the number of peers is large. Most importantly, data burdens in the first and the second methods are significant and it is not clear if the required data would be available in a social network setting. In this respect, for the computational studies of this dissertation we propose a linear influence model (Model 1) where the influence between people is dependent only on the person being influenced and the number of neighbors of this person. We also propose two different models where the influence is dependent on the influencer as well as the influenced and the structure of the influence is non-linear in the number of influencers.

Model 1: In our model, communication among friends strengthens the inclination towards buying the same product which others have also purchased. The counterpart of this behavior in our model is the decrease in one's hurdle (this could alternatively be viewed as an increase in utility from the product which is the third mechanism in Narayan et al., 2011). So, with every neighbor buying the product and individual's hurdle is updated. To avoid negative hurdles, the amount of decrease is limited within a hurdle span which we define as the interval between the hurdle in the traditional product design problem where no social network effects are considered (high hurdle), and a low hurdle which corresponds to the smallest value of utility a hurdle can take (i.e., when all friends buy the product). For this study, motivated by privacy concerns[1] prevalent on social networks, we assume a decrease structure that does not depend on the identity of the neighbor for the influence effect in our model. Therefore an individual is affected the same from each neighbor, but this effect may be different for each individual. We first introduce the model using a linearly decreasing influence effect.

Let h_s^H denote the "high" (H) and h_s^L denote the "low" (L) hurdle for person s. The amount of decrease in high hurdle depends on the number of neighbors who purchases the product. Due to the linear decrease structure, we calculate the unit decrease in hurdle for person s, Δ_s, as the ratio of the hurdle span and the degree (number of neighbors) of each node ($deg(s)$): $\Delta_s = \frac{h_s^H - h_s^L}{deg(s)}, \ \forall s \in V$. Using this definition for Δ_s, the share-of-choice model incorporating social network effects can be formulated as using the previous model, SOC, where constraint (1.2) is replaced with the following new constraint.

$$\sum_{k=1}^{K}\sum_{l=1}^{L_k} u_{kl}^s x_{kl} \geq h_s^H y_s - \Delta_s \sum_{j\in V} a_{js} y_j \qquad s = 1, 2, \ldots, n. \tag{2.1}$$

In constraint (2.1), the right hand side represents the *current* hurdle for person $s \in V$. Each entry of the social network adjacency matrix is shown by a_{js}, and it is 1 if there is an edge (j, s) between nodes j and s. We refer to the formulation with objective (1.1),

[1]In 2007, due to protests from users about privacy concerns, Facebook retreated on a tracking program called Beacon which sent messages to users friends about what they are buying on websites like Travelocity.com (Story and Stone, 2007). To address this, our model assumes that the anonymity of the users' friends will be preserved and only information on the number of friends purchasing the product is provided. In an environment where such privacy concerns are not important, our model is easily modified as described in Model 3.

constraints (1.3), (1.4), (1.5) and (2.1) as the share-of-choice model incorporating social network effects (SOCSNE).

Model 2: In an environment where the influence effects are negative and have a linear increase structure on the hurdles, Δ_s would simply be negative in constraint (2.1). For *any other influence structure* (nonlinear, diminishing returns, etc.), we propose the following model. Let $f_{si}, i, s \in V$, be the amount of change in hurdle of person s for the i^{th} additional neighbor who has purchased or adopted the product, and let g_{si} be a binary decision variable which equals 1 if there are at least i neighbors of node s who buy the product, and 0 otherwise. Then, constraint (2.1) would be replaced by constraints (2.2), (2.3) and (2.4). Constraint (2.2) is similar to constraint (2.1) in terms of calculating the current hurdle for node s. Here the effect equals to the sum of the effects from neighbors where each neighbor has a different effect at different times. Constraint (2.3) sets the number of g_{si} variables that are 1 equal to the number of neighbors who adopt the product. Constraint (2.4) establishes an ordering on the g_{si} variables to correctly compute the change in hurdle in accordance with the buyer neighbor. With this model, both the relationship with the neighbor and the number of previous owners matters in terms of influence and these can be regulated within the model using the f function.

$$\sum_{k=1}^{K}\sum_{l=1}^{L_k} u_{kl}^{s} x_{kl} \geq h_s^H y_s - \sum_{i=1}^{deg(s)} f_{si} g_{si} \qquad s = 1, 2, \ldots, n. \tag{2.2}$$

$$\sum_{j \in V} a_{js} y_j = \sum_{i=1}^{deg(s)} g_{si} \qquad s = 1, 2, \ldots, n. \tag{2.3}$$

$$g_{s1} \geq g_{s2} \geq \ldots \geq g_{s(deg(s))} \qquad s = 1, 2, \ldots, n. \tag{2.4}$$

Model 3: In a setting where neighbors of a node have different influences (this may be the case when the privacy concerns discussed earlier are moot), we suggest modifying the previous model as follows. Let Δ_{js} represent the influence of neighbor node j on node s (i.e., the amount by which node j's product adoption reduces node s's hurdle). To calculate Δ_{js}, we propose a weighted directed graph model, where (j, s) represents the directed link from node j to node s and $a_{js} = 1$ if there is a directed link from node j to node s with $A(s) = \{j | a_{js} = 1\}$ being the set of neighbors of node s. Further, the relative influence of node j on node s is denoted by the weight w_{js} (analogous to Narayan et al., 2011). For each arc (j, s), Δ_{js} can now be calculated using the formula;

$$\Delta_{js} = (h_s^H - h_s^L) \frac{w_{js}}{\sum\limits_{j \in A(s)} w_{js}}. \tag{2.5}$$

If there is no arc from node j to node s, then Δ_{js} is set to 0. After Δ_{js} is calculated, constraint (2.1) in the SOCSNE model is replaced with the following inequality so that the influence depends on the neighbor as well the node itself.

$$\sum_{k=1}^{K} \sum_{l=1}^{L_k} u_{kl}^s x_{kl} \geq h_s^H y_s - \sum_{j \in V} \Delta_{js} a_{js} y_j \qquad s = 1, 2, \ldots, n. \tag{2.6}$$

Note that Δ_{sj} and Δ_{js} are not necessarily equal in all three models, which is reasonable as the influence among people is not necessarily symmetric.

2.1.1 Why do Social Network Effects Matter in Product Design?

When social network effects are present, taking them into account in the share-of-choice model can lead to product designs with a higher market share. The following example is constructed to emphasize how large the difference can be among market shares with and without social network effects taken into account in the product design process. Consider a simple social network with 3 people where all customers are connected to each other, i.e., a fully connected network with three nodes. The problem is to design a product with a single attribute and maximize the market share. For simplicity, assume that there are only two possible levels for this attribute, level 1 and level 2. The corresponding

Table 2.1: Utility, hurdle and Δ_s values for the example.

Person	Utilities		High Hurdle	Δ_s
	Level 1	Level 2		
1	200	280	300	20
2	320	280	320	20
3	205	240	250	20

data on utility, high hurdle and Δ values for each person are provided in Table 2.1[2]. The utilities are $200, 320, 205$ for using level 1; $280, 280, 240$ for using level 2, and the high hurdles are $300, 320, 250$ for the three people, respectively. So person 1 would prefer level 2 to level 1 but would not buy the product in either case because his hurdle is larger than his utilities from both levels. Again for simplicity, let Δ_s values be the same for each person, 20, i.e., for each additional neighbor purchasing the product, a person's hurdle decreases by 20. When the original share-of-choice formulation (without the social network effects) is solved, it is easy to see by simply comparing utilities with hurdles that

[2]We should note that this data is generated only for this example and does not represent the data range used for the computational studies.

the optimal solution is to choose level 1; the market share is 1 and only person 2 purchases the product. Although the social network effects have not been taken into account in the design phase, if they are allowed after the product is launched, the purchase by person 2 decreases the hurdles for person 1 and 3. Their current hurdles become 280 and 230 but still are greater than their utilities (insufficient to induce them to buy the product). So the market share does not change. When the problem is solved taking social networks into account in the design phase (SOCSNE), the optimal solution is to choose level 2 and the market share is 3 where all 3 people buy the product. In this case, each hurdle is lowered by 20x2=40 because 2 neighbors of each person are purchasing the product. Now the comparison (300-40=260 vs. 280, 320-40=280 vs. 280 and 250-40=210 vs. 240) shows that all hurdles are less than the utilities. In this example, neglecting the social network effects in the product design leads to a solution that is three times worse than the solution obtained by the model that includes social network effects. This example can be generalized (simply add customers to the market identical to person 3) to show that in the worst case, the loss of market share when social network effects are ignored can be as large as the size of the market!

2.2 Least Cost Influence Problem (LCIP) over the Buyer Network

From Table 2.1 in the previous example, we see that the utilities for level 2 for each person are less than their high hurdles which implies no one would buy the product (designed with level 2) at the outset. Yet, the market share equals three when this level is selected for the attribute since the model implicitly allows for simultaneous buying.

Assuming that people are not making buying decisions together, to obtain this market share we need at least one person to purchase the product first and start a diffusion. But how should that person be selected? The second problem in this chapter concentrates on this question of diffusion and the ordering of buyers over a finite time period.

To obtain the market share solution from the product design of the proposed SOCSNE model, some people may need to be provided with incentives to make a purchase. In practice, such intervention to the product diffusion process is not costless, and for a business they can be contemplated as advertising or marketing costs of distributing free samples or discount coupons to a select group of people. We consider an incentive as an additional utility provided from the company selling the product to the potential customer. Once a person makes a purchase, hurdles for her/his neighbors are updated and checked to see if they also adopt (buy)[3] the product. Every time a new person adopts the product, this update and comparison is repeated. If at some point there is no new buyer and the market share has not yet reached the amount dictated by the SOCSNE model, a new person needs to be given incentives to purchase the product and to restart the diffusion. Although providing incentives result in a larger market share (equal to the amount dictated by the SOCSNE model), it could be expensive, therefore the set of people to give incentives need to be selected with care. This trade-off is the subject matter of the second problem in this chapter.

The objective of the least cost influence problem (LCIP) is to accomplish the market share of the SOCSNE model while minimizing the total amount of incentives given. To choose the set of people to give incentives to, we analyze the order of buying. We

[3]We use the terms adopt and buy interchangeably throughout the thesis.

introduce the time dimension, $t = 0, 1, ..., T$ (where T is the number of time periods), to capture the ordering of buyers. The product profile, the market share, and the individuals who will buy the product are inputs for the LCIP model because this problem is solved *after* the solution to the SOCSNE model is obtained. The social network in this problem, G', is a subset of the network in the SOCSNE model and includes only the people, V', that adopt the product as a result of the product profile chosen after solving the SOCSNE model (i.e., it includes only those nodes for which $y_s = 1$ in the SOCSNE model solution) and the edges connecting them, E'. There are two types of decision variables; z_s, $s \in V'$, represents the amount of incentive given to person s and y_{st}, $s \in V', t = 0, 1, \ldots, T$, is a binary variable which is 1, if person s buys in period t and is 0, otherwise. Since the attribute levels have been chosen by the first model (SOCSNE), the utility one gets by using the product can simply be represented as one parameter, U_s, for each person $s \in V'$. It is the summation of utilities from each level selected in each attribute. The mathematical formulation is as follows:

$$\textbf{LCIP:} \text{ Minimize } \sum_{s \in V'} z_s, \tag{2.7}$$

$$\text{subject to } \quad U_s \geq h_s^H y_{s0} \qquad \forall s \in V', \tag{2.8}$$

$$U_s \geq h_s^H y_{st} - z_s - \Delta_s \sum_{j \in V'} a'_{js} y_{j,t-1} \quad \forall s \in V', \forall t \geq 1, \tag{2.9}$$

$$y_{st} \geq y_{s,t-1} \qquad \forall s \in V', \forall t \geq 1, \tag{2.10}$$

$$y_{sT} = 1 \qquad \forall s \in V', \tag{2.11}$$

$$y_{st} \in \{0, 1\} \qquad \forall s \in V', \forall t \geq 0, \tag{2.12}$$

$$z_s \geq 0 \qquad \forall s \in V'. \tag{2.13}$$

The objective in this model is to minimize the total amount of incentives given to all the people in the buyer network. Constraint (2.8) identifies the people who buy the product in period 0 without requiring any incentives or influence from neighbors. Constraint (2.9) is similar to constraint (2.1) in the SOCSNE model; the right hand side captures the current hurdle. The current hurdle of a person in any time period is calculated as the remainder after the amount of incentives *and* the social network effects are subtracted from the high hurdle. Note that in the first model (SOCSNE), the hurdle of a person decreases only with social network effects. In the LCIP model, in addition to the decrease from social network effects, hurdles are also decreased by the incentives people receive. The second term in the right hand side of the constraint (2.9), z_s, represents this incentive amount for person s.

Observation 1: The amount of incentive one receives equals to the difference between its current hurdle and utility from the product. If the incentive is less than this difference, the person will not buy the product. If it's greater than this difference, it will hurt our objective function of minimizing the total amount of incentives. This consequently suggests a weak upper bound for the objective function as $\sum\limits_{s \in V'} (h_s^H - U_s)$, which is the summation of all the differences between hurdles and utilities for each node.

Observation 2: The number of periods is less than or equal to the number of people: $T \leq |V'|$. Note that once $y_{st} = 1$ for $t = k$, $y_{st} = 1$ for $t \geq k$ by constraint (2.10) and constraint (2.11) guarantees that every person on the network buys the product by forcing the last period decision to be 1. By this definition of a time period, there may be more than one buyer in a period.

Given the solution of the SOCSNE problem, the LCIP aims to identify the people whose purchase decisions constitute a bottleneck in the diffusion process and providing them incentives will strengthen product adoption to reach the market share of SOCSNE. In a general setting, the LCIP looks for the nodes to catalyze (or proliferate) a diffusion process. In the literature, the contagion or diffusion over a network is modeled fundamentally using two models, the linear threshold model (Granovetter, 1978) and the independent cascade model (Goldenberg et al., 2001). The basic difference between the two models is the way each individual is affected from individuals in its neighborhood. In both models, we start with an initial set of nodes which have already adopted. In the linear threshold model, a node is "affected" from diffusion when the sum of the influences from its neighbors exceeds a certain threshold. A person buys the product when his total influence is greater than a threshold. As the number of people that adopted the product increase, a person becomes more likely to adopt the product. In the Independent Cascade Model, the process still unfolds as a discrete process while the influence follows a randomized rule, i.e., when a person buys, he is given a single chance to influence his neighbors with a certain probability independent of the history thus far. Both processes run until there are no new adopters. Our model effectively follows the linear threshold model since a node adopts the product if and only if the sum of the influences from its neighbors and any incentives from outside the social network is greater than the difference, $h_s^H - U_s$. The problem of finding nodes to maximize the spread or influence over a social network has been studied previously, and is of increasing interest in a variety of disciplines including marketing, computer and information science. The most prominent research in this area is the "influence maximization problem" (Kempe et al., 2003). Here the objective is, for a given parameter *k*, to find a set of *k*-nodes to maximize influence over the network. The k individuals are seeded (i.e., given the product for free) at the beginning of the diffusion process and the diffusion process takes place (with these k-nodes exerting an influence on their neighbors). Our LCIP deviates from this approach; the incentives in our model are not seed products and involve partial inducements tailored for a set of individuals. We introduce incentive-discrimination which in a marketing setting could correspond to 10%, 50%, etc. discount coupons rather than free samples of a product. In this approach, an incentive is used to resolve a knot in the diffusion process that will lead to a larger spread with more influenced nodes at the end. Such catalysation addresses the trade-off of minimizing the amount of incentives given and reaching each individual in the network.

The LCIP is a computationally challenging problem which can take significant amount of time to solve a relatively small problem. In fact for a model where $T = |V'|$, it is not feasible to solve the LCIP model as it renders the computer out of memory in less than three hours for the smallest size data set. In Section 2.5, we provide an iterative approach to solve the LCIP optimally and in much shorter time.

2.3 A Genetic Algorithm for the Share-of-Choice Problem with Network Effects

A genetic algorithm (GA) (Holland, 1975) is an evolutionary search algorithm that imitates natural selection of species in order to reach near-optimal solutions. We use a GA approach for the SOCSNE model for three reasons. First, the time required to solve the presented integer programming model optimally increases rapidly with the size of the problem (making it a nonviable computational option). Second, due to influence effects, even for the small integer programs state of the art solvers like CPLEX have numerical instability and one is unable to find optimal solutions (specifically CPLEX finds incorrect optimal solutions!) to the problem. Third, specialized approaches to solve the original SOC problem cannot be extended easily to the SOCSNE problem. Our GA generates high quality solutions and is robust in terms of computational time.

For the original share-of-choice problem, Camm et al. (2006) use Lagrangian relaxation with branch-and-bound method to solve the problem exactly. In this method, a search tree is developed, where each level of the search tree corresponds to an attribute of the product. So a node down a path in the tree would have some levels of attributes fixed already. The search tree is pruned using logic-based rules. With these rules, a node is fathomed if the path starting at this node cannot produce a feasible solution superior to one that is already known. This is evaluated by checking whether people's hurdles fall in the range of the minimum and maximum utilities a person may have if that path is followed for the product design. The network relationship among prospective customers in our study prevents such logical inferences since comparison of utilities from the product and the hurdles include social network effects which depend on the buying status of one's neighbors. The calculation of the objective value at each node would still require the solution of the integer programming model proposed in this work which would significantly slow down the methodology and actually make it computationally intractable. For the product-line design case, efficient methods are presented and compared by Belloni et al. (2008). Their findings are consistent with the literature in terms of supporting the superiority of a genetic algorithm over other methods such as dynamic programming, beam search or a greedy heuristic. While simulated annealing is as successful, they report that it has a running time that is one or two orders of magnitude larger than other methods. Balakrishnan and Jacob (1996) also demonstrated the use and advantages of genetic algorithms for solving product design problems. Their study provides a starting point for the genetic algorithm we present here. We use a similar approach to obtain the product profile with the highest market share. However, our genetic algorithm varies significantly from Balakrishnan and Jacob (1996) in several regards including the fitness evaluation. The outline and the details of the genetic algorithm are given next. In our description, we assume the reader has some familiarity with genetic algorithms. A good introduction to genetic algorithms is the text by Goldberg (1989).

Outline of the Genetic Algorithm for the SOCSNE Problem

Input: *Parameters:* population size, mutation rate, number of generations.

Data: number of attributes, number of levels for each attribute, social network of people, high and low hurdles for each person, utilities for each person.

Output: Recommended product design, market share of the chosen product design.

Step 1 [GENERATE] Generate an initial population of q product profiles. Set $t = 0$.

Step 2 [EVALUATION] Calculate fitness of each product profile and let BEST = the profile with the largest fitness.

Step 3 [CROSSOVER] Perform single-point crossover operation to generate q offsprings.

Step 4 [MUTATION] Perform mutation.

Step 5 Calculate fitness of each product profile. If the largest fitness $>$ BEST, update BEST.

Step 6 [REDUCTION] Reduce the population to half by choosing the ones with greatest fitness. $t = t + 1$.

Step 7 If $t <$ number of generations, then go to Step 3.
Else, STOP.

GENERATE: The population consists of product profiles which are represented by binary strings, sized $\sum_{k=1}^{K} L_k$ where L_k is the number of levels for attribute k. An *initial population* is generated randomly by assigning one level for each attribute. For example, if the product has 2 attributes, color and size with 2 and 3 levels as (black, white) and (small, medium, large) respectively, then the product with color white and size small would be represented as $(01\ 100)$.

EVALUATION: After the population is generated, each product profile in the population is evaluated for fitness. *Evaluation* of a product profile corresponds to calculating the market share if that product profile is launched in the market. The exact value of market share is easily determined with the given x_{kl} values for that profile (for example, by solving the integer program SOCSNE model with the x_{kl} values fixed). Methods used to calculate the fitness should be chosen carefully. In an earlier approach, we used an approximate fitness function (where the number of people adopting the product was calculated simply by comparing hurdles with the utilities while hurdles are updated after each purchase) for which the results of the genetic algorithm were significantly worse. Using the exact evaluation via an integer program (for example with the solver CPLEX) corresponds to a hybrid approach called MATHEURISTICS (Maniezzo et al., 2009) marrying mathematical programming with metaheuristic approaches.

CROSSOVER: In this step, two offspring product profiles are produced by two parent product profiles. Parents are chosen from a given population with respect to their fitnesses using the roulette-wheel mechanism (Michalewicz, 1996). This allows individuals with higher fitnesses to be more likely to get selected as parents. The offspring carry properties of both parents. We use a *single-point crossover* to determine how the heritage is carried to the new generation of individuals. A point is chosen randomly from the points where binary representation of each attribute ends. One of the offspring gets the entries before that point from the first parent and the entries after that from the second parent. The other offspring gets the properties of the attributes after that point from the first parent and the properties of attributes before that point from the second parent. This step is carried out until the number of offspring created is equal to the population size, so the size of the population is doubled at the end of this stage.

MUTATION: *Mutation* in product profiles are used to incorporate a different direction in the search process. It corresponds to making a change in the product profile and creating a profile whose properties are not all inherited from the parents. Each product profile in the population undergoes this step, however mutation occurs with a predefined mutation rate or probability, μ. In this algorithm, mutation is done by changing the level of one of the attributes to another level. If a profile is subject to mutation, each attribute has an equal chance of being changed. Similarly, all other levels are equally likely to be selected to be the new level. When a profile is mutated, only the mutated version stays in the population. So, at the end of this phase, the size of the population stays the same.

REDUCTION: The population size is halved by eliminating the individuals with the least market share. The other half of the profiles with greater market share are carried to the next generation.

STOPPING CONDITION: The process is repeated until either there is no significant improvement over multiple generations or a predefined number of iterations is reached.

2.4 Computational Results

In this section, we present experimental studies on the following issues; data generation, performance evaluation of the proposed genetic algorithm and the improvement in market share when network effects are considered in product design process.

2.4.1 Data Generation

Similar to earlier work in the literature (Kohli and Krishnamurti, 1987; Balakrishnan and Jacob, 1996; Shi et al., 2001; Camm et al., 2006; Wang et al., 2009) on the share-of-choice problem, we generated our own data which included the generation of a social network, part-worth utilities and a hurdle span for each individual. Keeping along similar lines with the data generation methods used in the previous studies (Kohli and Krishnamurti, 1987; Nair et al., 1995), part-worth utilities are uniformly generated between 0 and 1 and normalized for each individual, meaning for each person there is one level of attribute with utility 0 and one level of attribute with utility 1. The utilities for rest of the levels change between 0 and 1. Our modeling approach requires a hurdle span for each person which is characterized by a high and a low hurdle. To represent a heterogeneous preference behavior among the customers, hurdle spans are selected with the following method; 1000 product profiles are randomly generated and for every person they are ranked in descending order with respect to total utilities they provide for that person. High hurdle, which could correspond to total utilities of a status-quo product, is randomly selected from the first 500 product profiles of this ordered set and low hurdle is randomly selected from the second half of the same set.

Social networks of customers are randomly generated. However, the random generation of social networks is more complex than typical random graph generation. In the past, random network generation has typically been done by using the Erdös and Rényi (1959) method. In this model, given a set of vertices, two vertices have a connecting edge with independent probability p. However, the degree distribution of the generated network becomes a Poisson distribution and is found to be inaccurate in modeling real-world social networks. Experimental studies also support the fact that random networks generated with respect to the Erdös and Rényi (1959) model do not represent actual social networks. Random social networks are different from other random networks because they show some sort of an order even while they have randomness. Several methods are studied to generate random social networks. The most frequently studied ones are the preferential attachment, exponential random graphs, power-law distributed graphs and small-world phenomenon. Barabasi and Albert (1999) used the growing model (preferential attachment) for generating social networks. In this model, the nodes are added to a network one by one and these new nodes prefer to connect to the nodes with higher degrees. The second model is the power-law model. The probability $P(k)$ that a vertex in graph interacts with k other vertices decays as a power-law distribution. The third model is the exponential random model *(p* model)* which is a generalization of Markov random graphs. In our test data, we use the small-world network generation model in R programming (R Development Core Team, 2010). In the small-world model every node ends up being only a few (about six) connections away from each other. To generate such a graph, Watts and Strogatz (1998) proposed the following rewiring method. In this method, the network starts with a ring of n nodes, each connected to its k nearest neighbors by undirected links. A node and the link that connects it to its nearest neighbor in a clockwise sense are chosen. With probability p, the edge is reconnected to a node chosen uniformly at random over the entire ring, with duplicate edges forbidden; otherwise the edge is left in place. This is repeated until each edge in the original lattice has been considered once. For $p = 0$, the original ring is unchanged; as p increases, the graph becomes increasingly disordered until for $p = 1$, all edges are rewired randomly, as illustrated in Figure 2.1. The connection topology for social networks lies somewhere between the two extremes of completely regular and completely random (Watts and Strogatz, 1998).

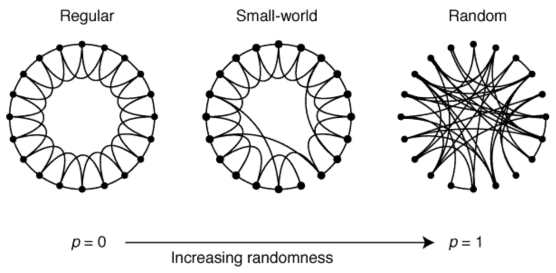

Figure 2.1: Random rewiring procedure.

Since (i) the clustering coefficient[4] starts to decrease sharply at around $p = 0.3$, and (ii) Watts and Strogatz (1998) showed that for intermediate values of p the graph is a small-world network; we used $p = 0.1, 0.2$ and 0.3 as the rewiring probabilities when generating the nine social networks with 100, 200 and 300 people, as illustrated in Figures

[4]For friendship networks, the intuitive meaning is that this coefficient measures the cliquishness of a typical friendship circle (Watts and Strogatz, 1998).

2.2, 2.3 and 2.4.

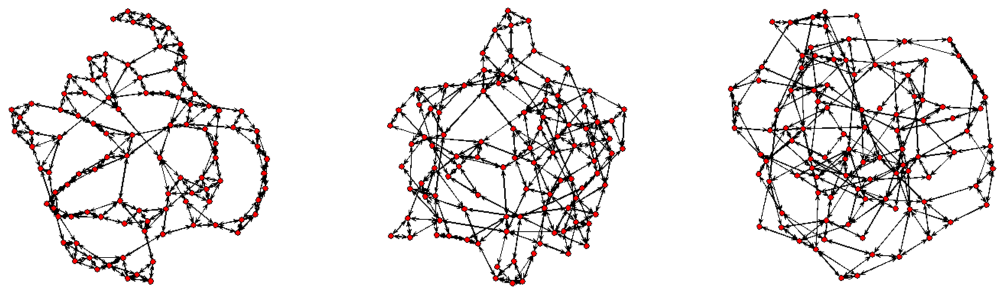
Figure 2.2: Social networks of 100 people for $p = 0.1, p = 0.2, p = 0.3$, respectively.

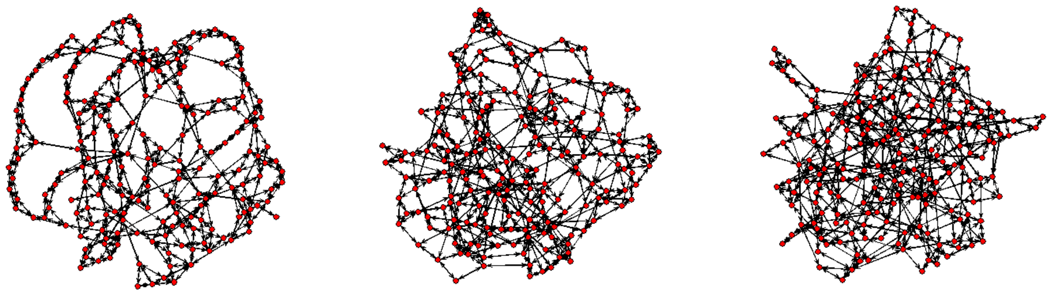
Figure 2.3: Social networks of 200 people for $p = 0.1, p = 0.2, p = 0.3$, respectively.

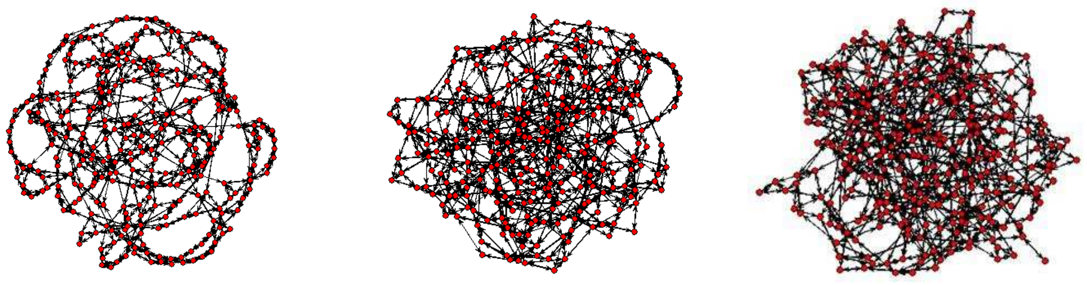
Figure 2.4: Social networks of 300 people for $p = 0.1, p = 0.2, p = 0.3$, respectively.

For the linear influence structure used in our computations, for an individual s, each additional buying neighbor decreases the hurdle an equal amount, Δ_s. When the social network is generated and the neighbors are determined for each person, Δ_s is calculated as the ratio of the hurdle span and the number of neighbors. By this definition, if all neighbors of an individual buy the product, the current hurdle for that individual reaches the low hurdle.

2.4.2 Performance Evaluation of the GA

To evaluate the performance of the proposed GA, we compare it with the optimal solution of the exact integer programming (IP) model SOCSNE (described in Section 2.1) obtained using CPLEX 12.0 on a 3.00 GHz AMD Phenom II X2 B55 Processor with 4.00 GB of RAM. The GA was coded in C++ and run on the same computer. The complete combination of product profiles of (3,4,5,6) attributes, (2,3,4,5) levels, (0.1,0.2,0.3) rewiring probabilities (p) and (100,200,300) number of people in the market constitute the 144 data sets. After considerable experimentation on small data sets, the GA parameters for the population size and the mutation rate are set to 100 and 0.3, respectively and the stopping condition is set as 10 generations. The market shares for the product profiles selected by the GA and SOCSNE IP model for the 144 problems are shown in Table 2.2.

Information on the rewiring probabilities used to generate the social networks and the product profile (in terms of the number of attributes and the number of levels for each attribute) are given in the first three columns of Table 2.2. Comparing the market shares for the GA and the IP model SOCSNE, the GA obtains the optimal solution in 126 out of 144 data sets where the optimal solution is known (from solving the SOCSNE model). The average running time is 106 seconds for the GA and 1305 seconds for the SOCSNE. The worst case for the SOCSNE IP model is as large as 20110 seconds to find the optimal or 11642 seconds before running out of memory. The running time for the GA is only slightly affected by the size of the social network (the worst case is 199 seconds), whereas the SOCSNE IP model is affected significantly both by the size of the network and the size of the set of possible product profiles, as expected.

To explore the GA performance with larger data sets, we ran experiments using product profiles of size (8,9,10) attributes with (2,3,4,5) levels for $p = 0.2$, 100 and 200 people, given in Table 2.3. The running time of the GA stays around 60 seconds for the first network and 87 seconds for the second network whereas the IP model can take around 15 hours before it reaches the optimal solution. Further it renders the computer out of memory (abbreviated as o.o.m.) for nine problems. On average, for the problems that the IP model could solve, the GA is able to get 98.76% of the optimal solution with the running time around one minute.

Table 2.2: Comparison of market shares for the GA and the optimal solution of SOCSNE.

	Num. of	Num.of	100 people		200 people		300 people	
	attributes	levels	GA	SOCSNE	GA	SOCSNE	GA	SOCSNE
p=0.1	3	2	76	76	132	132	191	191
	3	3	76	76	135	135	200	200
	3	4	69	69	133	133	199	199
	3	5	75	75	147	147	221	221
	4	2	76	76	132	132	190	190
	4	3	71	71	140	140	185	185
	4	4	70	70	133	133	198	198
	4	5	80	80	140	140	217	217
	5	2	81	81	140	140	206	206
	5	3	84	84	146	146	208	208
	5	4	78	79	151	151	212	212
	5	5	77	77	144	144	208	212
	6	2	78	78	136	136	196	196
	6	3	77	77	142	142	205	205
	6	4	76	80	150	150	218	218
	6	5	85	85	145	152	216	216
p=0.2	3	2	68	68	129	129	190	190
	3	3	73	73	133	133	191	191
	3	4	81	81	129	129	191	191
	3	5	74	74	128	128	205	205
	4	2	72	72	141	141	191	191
	4	3	73	73	136	136	198	198
	4	4	77	77	139	139	204	204
	4	5	72	73	133	135	198	198
	5	2	70	70	127	127	189	189
	5	3	82	82	140	140	193	193
	5	4	79	79	148	148	203	206
	5	5	82	82	144	144	215	222
	6	2	78	78	140	140	200	200
	6	3	76	76	145	145	206	206
	6	4	81	84	146	146	205	209
	6	5	80	83	145	148	217	217
p=0.3	3	2	70	70	146	146	196	196
	3	3	74	74	134	134	189	189
	3	4	82	82	145	146	202	202
	3	5	71	71	148	148	199	199
	4	2	71	71	142	142	201	201
	4	3	66	66	142	142	200	200
	4	4	72	72	145	145	202	202
	4	5	81	81	141	141	228	228
	5	2	72	72	141	141	199	199
	5	3	76	76	145	145	208	208
	5	4	79	79	145	145	213	213
	5	5	76	76	149	151	206	206
	6	2	69	69	143	143	208	208
	6	3	73	79	143	143	196	196
	6	4	76	76	147	147	209	216
	6	5	83	83	142	145	215	218

Table 2.3: GA and SOCSNE market share results for larger size data sets.

		100 people				200 people			
Num. of	Num. of	GA		SOCSNE		GA		SOCSNE	
attributes	levels	Market share	Time (sec)	Market share	Time (sec)	Market share	Time (sec)	Market share	Time (sec)
8	2	68	65	68	2	138	83	138	4
8	3	87	56	87	122	151	83	o.o.m.	7003
8	4	82	60	82	23002	148	83	o.o.m.	8828
8	5	83	60	84	33454	155	85	o.o.m.	9651
9	2	80	61	80	7	156	89	156	11
9	3	82	59	82	3795	153	90	o.o.m.	2542
9	4	84	60	86	15843	151	92	o.o.m.	10351
9	5	89	60	91	9881	156	93	o.o.m.	11920
10	2	76	64	76	155	137	88	137	27
10	3	81	62	84	5443	153	91	o.o.m.	8412
10	4	80	61	87	55618	155	87	o.o.m.	12139
10	5	83	59	o.o.m.	16115	151	88	o.o.m.	13858

2.4.3 Network Effects on Market Share

In Tables 2.4, 2.5 and 2.6, we compare the market shares for product profiles designed with and without taking network effects into account. The column “SOC” gives the market share for the optimal product design obtained by the exact integer programming model solution of the original SOC model. In this model, network effects are not considered in the design process, incentives are not allowed and people decide to buy the product if and only if their utilities are greater than or equal to their hurdles. The second column “SOC+NE” (NE=network effects) gives the market share for the same product, but it is greater than the first column since this column shows the market share after the product has spread over the network without any intervention from the company. As some people bought the product, others are influenced and some of the influenced also buy the product and this behavior naturally cascades through the network. In the fourth column “GA+IN”, we have the market shares for the product designed using the GA for the SOC-SNE model. However, obtaining this market share may involve promoting the product with incentives (IN), as explained in Section 2.2. If such incentives are not provided, the market shares are as given in column “GA+NE”. Note that for columns 2 and 3, the market share with network effects but no incentives is easily computed in an iterative fashion, while the market share with network effects and incentives given in column 4 is computed by using the product profile and inserting it into the IP model SOCSNE (i.e., the variables corresponding to the product are fixed).

As should be expected, considering network effects, compared to neglecting these effects (SOC vs. SOC+NE), always increases the market share. Similarly, providing incentives, compared to no incentives (GA+NE vs. GA+IN), produces a larger market share. The computational experiments indicate that this increase in market share can be quite significant, and ignoring network effects in the design process leads to a substantially inferior product design. The trade-off between the amount of incentives given and the amount of increase acquired in the market share is analyzed in Section 2.5.3 with the introduction of a social welfare measure.

Table 2.4: Network effects on market share for rewiring probability $p = 0.1$.

Num. of	Num.of	100 people				200 people				300 people			
attributes	levels	SOC	SOC+NE	GA+NE	GA+IN	SOC	SOC+NE	GA+NE	GA+IN	SOC	SOC+NE	GA+NE	GA+IN
3	2	35	56	56	76	68	105	102	132	100	154	164	191
3	3	41	58	61	76	70	115	97	135	100	159	136	200
3	4	40	56	42	69	58	95	88	133	101	169	169	199
3	5	40	66	58	75	70	110	123	147	105	184	152	221
4	2	40	56	66	76	64	111	97	132	96	136	147	190
4	3	38	55	62	71	68	113	104	140	104	154	144	185
4	4	37	57	65	70	64	103	107	133	97	171	148	198
4	5	38	56	61	80	67	104	117	140	96	158	139	217
5	2	36	51	61	81	59	88	89	140	98	173	168	206
5	3	40	53	80	84	65	104	88	146	96	149	143	208
5	4	39	60	66	78	71	116	111	151	95	149	161	212
5	5	38	55	63	77	69	105	117	144	99	160	175	208
6	2	33	52	57	78	65	96	95	136	91	149	141	196
6	3	32	47	44	77	69	98	97	142	105	170	179	205
6	4	41	58	55	76	o.o.m.	-	118	150	104	178	182	218
6	5	o.o.m.	-	68	85	o.o.m.	-	98	145	o.o.m.	-	168	216
Average		37.87	55.73	60.31	76.81	66.21	104.50	103.00	140.38	99.13	160.87	157.25	204.38

Table 2.5: Network effects on market share for rewiring probability $p = 0.2$.

Num. of	Num.of	100 people				200 people				300 people			
attributes	levels	SOC	SOC+NE	GA+NE	GA+IN	SOC	SOC+NE	GA+NE	GA+IN	SOC	SOC+NE	GA+NE	GA+IN
3	2	36	55	42	68	69	84	91	129	103	144	154	190
3	3	33	48	66	73	71	93	97	133	98	161	138	191
3	4	34	43	56	81	68	114	110	129	89	133	140	191
3	5	37	61	53	74	61	88	79	128	103	180	173	205
4	2	40	55	57	72	65	117	111	141	93	131	154	191
4	3	36	51	46	73	69	114	113	136	102	152	122	198
4	4	39	56	48	77	68	110	95	139	93	160	158	204
4	5	35	59	59	72	64	92	111	133	96	161	140	198
5	2	37	65	63	70	62	91	87	127	91	144	137	189
5	3	41	68	75	82	70	103	105	140	97	164	143	193
5	4	39	59	67	79	67	115	112	148	99	155	174	203
5	5	37	55	49	82	77	118	131	144	100	174	146	215
6	2	36	63	72	78	59	91	114	140	97	164	166	200
6	3	35	61	60	76	70	116	103	145	104	139	153	206
6	4	37	53	71	81	69	103	113	146	o.o.m.	-	171	205
6	5	o.o.m.	-	67	80	o.o.m.	-	88	145	o.o.m.	-	164	217
Average		36.80	56.80	59.44	76.13	67.27	103.27	103.75	137.69	97.50	154.43	152.06	199.75

Table 2.6: Network effects on market share for rewiring probability $p = 0.3$.

Num. of	Num.of	100 people				200 people				300 people			
attributes	levels	SOC	SOC+NE	GA+NE	GA+IN	SOC	SOC+NE	GA+NE	GA+IN	SOC	SOC+NE	GA+NE	GA+IN
3	2	36	61	61	70	74	112	112	146	109	157	160	196
3	3	33	50	49	74	67	99	100	134	97	151	136	189
3	4	36	49	59	82	69	121	103	145	102	155	134	202
3	5	38	59	46	71	70	121	110	148	97	145	148	199
4	2	36	67	67	71	69	127	127	142	99	154	141	201
4	3	39	61	55	66	64	94	119	142	103	168	162	200
4	4	36	50	56	72	68	98	117	145	99	167	166	202
4	5	39	55	68	81	72	113	113	141	100	161	161	228
5	2	36	53	66	72	59	100	105	141	93	157	161	199
5	3	36	63	50	76	69	106	110	145	102	172	169	208
5	4	36	52	68	79	72	125	121	145	97	143	167	213
5	5	38	60	43	76	82	120	124	149	100	154	133	206
6	2	33	50	64	69	64	106	115	143	86	150	147	208
6	3	34	57	57	73	73	121	119	143	102	155	161	196
6	4	35	49	39	76	71	108	102	147	105	163	177	209
6	5	o.o.m.	-	72	83	o.o.m.	-	130	142	o.o.m.	-	163	215
Average		36.07	55.73	57.50	74.44	69.53	111.40	114.19	143.63	99.40	156.80	155.38	204.44

2.5 Solving the LCIP Model: Identifying Individuals to Pay Out Incentives to

The solution to the least cost influence problem (LCIP) identifies the set of people who receive some incentives to buy the product and further influence their neighbors. The model for the LCIP was given in Section 2.2 and is computationally intractable to solve even for very small problem sizes. To overcome this problem, we preprocess the model and then use a tractable, iterative, and much faster approach that preserves optimality (i.e., ensures the LCIP is solved to optimality).

In the LCIP model, the influence spreads through the network over a finite number of time periods (Observation 2) and eventually reaches everyone. At the beginning of each period, an individual's hurdle is updated with respect to neighbors' buying decisions. Decisions made in time t affect neighbors' hurdles in period $(t+1)$.

Preprocessing: As a preprocessing step before solving the LCIP, we identify by simple comparison the nodes whose utilities are already greater than or equal to their hurdles and the cascade of nodes that purchase the product after being influenced from previous buyers *without* requiring any incentives. Once these nodes are eliminated, the remaining network of nodes are the ones that need an incentive to start a new cascade of buyers (i.e., people which constitute the market share difference between columns GA+IN and GA+NE in tables in Section 2.4.3).

It is then easier to recast the LCIP as follows. For convenience, we will repeat notation and let V' denote the nodes in the problem after preprocessing, b_s denote the difference between the current hurdle and product utility for $s \in V'$ (observe $b_s > 0$ for

$s \in V'$), and $d_{js} = a'_{js}\Delta_s$ denote the influence of node j on node s if node j adopts the product. The LCIP can be rewritten as;

$$\textbf{LCIP:} \quad \text{Minimize} \quad \sum_{s \in V'} z_s, \tag{2.14}$$

$$\text{subject to} \quad z_s + \sum_{j \in V'} d_{js} y_{j(t-1)} \geq b_s y_{jt} \qquad \forall s \in V', \forall t \geq 1, \tag{2.15}$$

$$y_{s0} = 0 \qquad \forall s \in V', \tag{2.16}$$

$$\text{Constraints } (2.10), (2.11), (2.12) \text{ and } (2.13).$$

Iterative approach: Our iterative approach further reduces the size of the LCIP by limiting the number of time periods to obtain an initial solution, and then increments the number of periods by one at each iteration. In the single period model, i.e., in the absence of a successor period, network effects cannot take place. The individuals have to be paid the difference between their hurdles and utilities to adopt. In the next iteration, the LCIP model is re-solved with two time periods. Here the network effects are present but limited to only the "first"- and "second"-generation buyers.

Observation 3: The cost of total incentives can never be worse than costs in the previous iteration. In the additional period, some of the individuals who were given incentives in the previous solution may be covered by influence from neighbors decreasing the total amount of incentives.

Observation 4: The costs can only decrease when there is at least one new buyer in a period. The decrease in the amount of incentives can only be covered by the influence of new buyers in the previous period.

In every successive iteration an additional time period is added allowing peer influence effects to cascade further into the network. The iterations are continued until the cost of the LCIP solution (i.e., incentives paid out) stays the same as the cost of the previous iteration.

Observation 5: The iteration where the objective stays the same is the **optimal solution**. No improvement in the objective value shows that an additional time period is not used for influence.

2.5.1 Computational Results

Table 2.7 shows the LCIP results for the 32 problems where $p = 0.2$ and networks have sizes 100 and 200. Here, the ratio between the number of people who decide to buy as a result of their neighbors' decisions and who receive incentives is given in column "Return". For example, for the problem set (3,2,100), the return is 0.73 where 15 people are given incentives and they influence 11 additional people to change their decisions without receiving incentives. (Return is calculated as the ratio $\frac{11}{15}$.) Note that the number of new buyers in columns 3 and 7 are equal to the difference between columns "GA+NE" and "GA+IN" in Table 2.5. This number is the size of the set of buyers who purchase the product after a subset of the network have been given incentives. For the same example, the market share is increased from 42 to 68 by giving incentives to only 15 people. Overall, the returns for the 32 data sets are all greater than or equal to 0.67 meaning (in the worst case) giving incentives to 3 people leads 1 additional person to buy without receiving incentives. The number of iterations needed to solve the LCIP problems are given in columns 5 and 9 for 100 and 200 people, respectively. Notice that the original

LCIP model for the problem with 6 attributes, 5 levels, 200 people (Table 2.5) has 145 time periods, but our preprocessing reduces this to a problem over 57 nodes and iterative approach solves the problem within 4 iterations (Table 2.7).

Although the problem size is decreased with the iterative approach, the running times for the problem data sets still increase with the size of market. The average run time increases from 4.94 seconds to 47 minutes when the size of the market increase from 100 to 200. For the problem sets of (3,3,200), (3,5,200) and (6,5,200), the computer ran out of memory before reaching the optimal solution. However, we were able to get close to optimal solutions in terms of the number of people that are given incentives and the amount of incentives they have been given. This is the case for almost half of the data sets with 300 consumers, so we don't present those results here.

Table 2.7: LCIP results for rewiring probability $p = 0.2$.

		100 people				200 people			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of iterations	Return
3	2	26	15	5	0.73	38	22	7	0.73
3	3	7	3	3	1.33	36	19	7	0.89
3	4	25	13	4	0.92	19	7	4	1.71
3	5	21	9	7	1.33	49	16	6	2.06
4	2	15	6	2	1.50	30	18	6	0.67
4	3	27	13	4	0.77	23	13	4	0.77
4	4	29	12	6	1.42	44	22	5	1.00
4	5	13	7	3	0.86	22	11	5	1.00
5	2	7	3	3	1.33	40	17	6	1.35
5	3	7	3	4	1.33	35	14	4	1.50
5	4	12	7	3	0.71	36	17	6	1.12
5	5	33	15	7	1.20	13	4	4	2.25
6	2	6	2	3	2.00	26	9	5	1.89
6	3	16	7	3	1.29	42	16	8	1.63
6	4	10	6	4	0.67	33	16	4	1.06
6	5	13	6	4	1.17	57	27	4	1.11

2.5.2 An Illustration of the LCIP solution

To further examine and illustrate the cascading network effects in the LCIP solution, we focus on one data set (6,2,200, $p = 0.2$) as an example. For this example, when incentives are not present, the product profile obtained by the GA captures a market share of 114 customers after the network effects take place (Table 2.5). However, the product adoption of 9 additional people via incentives increases the market share to 140. The influence originating from these 9 people does not reach the additional 17 people in a single period. Figure 2.5 and Table 2.8 show, period-by-period, how the influence spreads over the part of the social network that contains people who will buy the product either after receiving incentives or being influenced by their neighbors. Customers provided with incentives are marked with "+" signs and customers that are influenced by others and buy the product are marked with "•" signs.

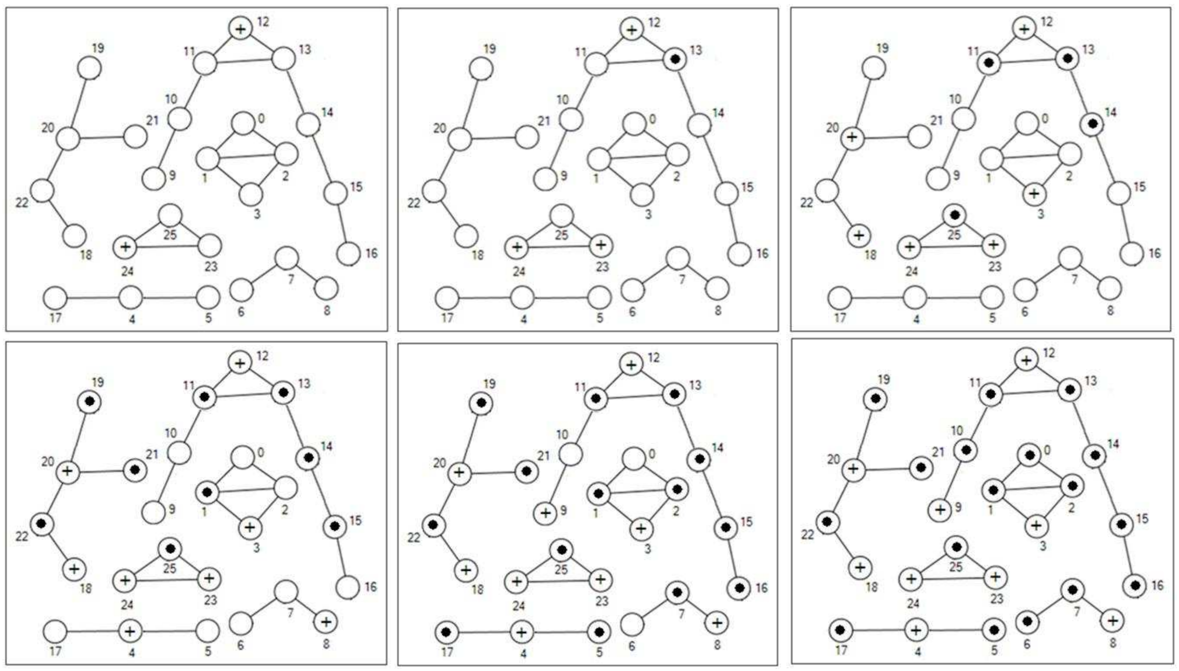

Figure 2.5: LCIP solution for the data set (6,2,200,$p = 0.2$).

To explain the diffusion process, we focus our attention on the largest network block of nodes numbered as $\{9,10,\ldots,16\}$. The diffusion over the remaining nodes can be followed via Table 2.8. In the first period node 12 is provided with incentives. In the next period, hurdles are updated for its neighbors (11 and 13). Although 11 is influenced by 12, it still does not adopt the product. The high hurdle for 13 decreases sufficiently enough to change its decision. In the third period, the new customers affected from the influence (of node 13) are 11 and 14. Both of them adopt the product. In the following two periods, 15 buys after the influence from 14 and 16 buys as a result of the influence from 15. In the fifth period node 9 is given incentives to adopt the product. Finally in period 6, node 10 buys the product with the influence from node 9.

Table 2.8: LCIP example.

				Current Hurdle						
Person	Utilities	Hurdle	Δ	t=1	t=2	t=3	t=4	t=5	t=6	Incentives
0	2.0899	2.5005	0.3666	2.5005	2.5005	2.5005	2.5005	2.1340	**1.7674**	0
1	3.5918	3.8967	0.3801	3.8967	3.8967	3.8967	**3.5166**			0
2	3.6248	3.9235	0.2552	3.9235	3.9235	3.9235	3.6683	**3.4131**		0
3	3.7329	3.8533	0.1208	3.8533	3.8533	**3.7329**				0.1204
4	2.0695	2.5299	0.2614	2.5299	2.5299	2.5299	**2.0695**			0.4604
5	2.3355	2.6251	0.3102	2.6251	2.6251	2.6251	2.6251	**2.3148**		0
6	2.0638	2.2550	0.1922	2.2550	2.2550	2.2550	2.2550	2.2550	**2.0628**	0
7	2.4257	2.7660	0.5966	2.7660	2.7660	2.7660	2.7660	**2.1694**		0
8	3.1668	3.2157	0.2219	3.2157	3.2157	3.2157	**3.1668**			0.0489
9	2.2630	2.3132	0.6612	2.3132	2.3132	2.3132	2.3132	**2.2630**		0.0502
10	3.5868	3.9947	0.2445	3.9947	3.9947	3.9947	3.7502	3.7502	**3.5057**	0
11	3.6349	4.3801	0.3775	4.3801	4.0026	**3.6251**				0
12	2.9853	3.0490	0.4652	**2.9853**						0.0637
13	3.7322	3.8657	0.1374	3.8657	**3.7283**					0
14	3.7534	4.0310	0.5328	4.0310	4.0310	**3.4982**				0
15	3.7941	3.9833	0.5875	3.9833	3.9833	3.9833	**3.3958**			0
16	2.0522	2.0543	0.2552	2.0543	2.0543	2.0543	2.0543	**1.7991**		0
17	1.8873	2.1816	0.4424	2.1816	2.1816	2.1816	2.1816	**1.7391**		0
18	5.0710	5.1173	0.2870	5.1173	5.1173	**5.0710**				0.0462
19	3.4486	3.7521	0.3519	3.7521	3.7521	3.7521	**3.4002**			0
20	2.8955	2.9618	0.1512	2.9618	2.9618	**2.8955**				0.0664
21	2.2459	2.6065	0.4770	2.6065	2.6065	2.6065	**2.1295**			0
22	2.5947	3.5842	0.5028	3.5842	3.5842	3.5842	**2.5785**			0
23	2.5196	3.2482	0.4113	3.2482	**2.5196**					0.3172
24	2.7059	2.7566	0.3072	**2.7059**						0.0507
25	2.8436	3.9490	0.6422	3.9490	3.3067	**2.6645**				0

2.5.3 Social Welfare Comparison

The amount of incentives provided and the increase in the market share are not measured in the same units, perhaps making it somewhat hard to analyze the trade-off. To ease the comparison, we use a welfare measure similar to the utilitarian function (selecting a product on the basis of the sum of utilities) as in Gupta and Kohli (1990). Finding the product profile that results in the largest overall consumer utility is referred to as the "buyer's welfare problem". Individual welfare is calculated as the difference between total utility from a product and the current hurdle of the individual. Social welfare is the sum of all the individual welfares. We look at the social welfare when incentives are given in Table 2.9 and calculate the average increase in social welfare for a unit incentive given in column "Return". Note that all returns are positive and the smallest return is more than 4 times the incentive provided!

Table 2.9: Effects of incentives on social welfare.

		100 people				200 people			
Num. of	Num. of	Incentives allowed		No Incentives	Return	Incentives allowed		No Incentives	Return
attributes	levels	Social Welfare	Incentives	Social Welfare		Social Welfare	Incentives	Social Welfare	
3	2	26.6057	1.7895	15.6838	6.1032	59.6522	1.8998	44.4102	8.0231
3	3	45.0922	0.3370	40.7852	12.7817	78.5633	1.6689	58.9479	11.7532
3	4	38.8263	1.1015	28.1265	9.7139	70.0126	0.5986	58.9194	18.5315
3	5	44.6026	1.0084	32.8448	11.6597	65.7663	0.9536	40.0679	26.9477
4	2	43.0862	0.4841	35.9059	14.8316	81.7442	1.8612	67.2795	7.7719
4	3	38.2466	1.2263	26.2343	9.7952	85.6369	1.4974	75.5858	6.7125
4	4	44.1616	0.7542	29.7510	19.1081	78.1816	1.8015	56.2452	12.1769
4	5	48.8871	0.7300	41.6471	9.9178	76.8212	0.6906	65.1930	16.8377
5	2	48.6230	0.6672	45.8226	4.1975	70.9345	1.5533	52.2856	12.0057
5	3	59.3606	0.6552	54.8855	6.8302	76.8809	0.8056	59.6078	21.4417
5	4	56.7058	0.7871	48.3769	10.5820	93.9887	1.1779	72.3549	18.3666
5	5	55.9945	0.9820	34.2808	22.1120	106.4820	0.2837	97.4306	31.8993
6	2	50.8347	0.2571	47.2107	14.0970	82.6756	1.2242	67.1765	12.6608
6	3	43.1903	0.5685	34.5883	15.1297	101.5370	1.6088	76.4503	15.5934
6	4	53.6025	0.5091	46.8800	13.2057	109.6330	1.6954	89.1506	12.0809
6	5	65.1313	0.5529	56.7974	15.0719	93.1654	3.3888	64.6902	8.4028

2.6 Concluding Remarks

We proposed a novel model to include peer influence effects in product design within the framework of the share-of-choice (SOC) problem. Although the share-of-choice problem has been studied in the marketing literature, to our knowledge the peer influence effects have never been explicitly considered previously within the product design process. In this model, we attempt to develop a new product taking social network effects into account, before intervening during the marketing phase with targeted promotions. By taking into account peer influence effects, with our new model one is able to design products with far larger market shares than obtained by the original SOC model. While the effects of peer influence on consumer choice are well documented, previous analysis of conjoint data typically assumed that a consumer's attribute preferences and product choices are independent of choices of others. Narayan et al. (2011) take a significant first step in the development of conjoint estimation models that incorporate peer influence. The SOCSNE model introduced in this chapter allows one to successfully use the results of such conjoint estimation models towards a logical next step—the design of a product with the largest market share.

The new model we constructed remains a computationally challenging NP-Hard problem. However, after extensive computational studies we show that the GA used to solve this problem finds high quality solutions for the simulated data sets. Such evaluation is possible for small size data sets where the optimal is available for comparison. For the larger data sets, where CPLEX requires hours to solve the problem or is unable to do so, the GA is robust and preserves a running time around less than two minutes independent of the size of the problem. As one of the characteristics of the GA, it is flexible with extensions requiring only minor modifications to the algorithm.

Engineering the diffusion of a product is already used by businesses in the form of free samples or discount coupons. The modeling of this process (LCIP) by using the product profile obtained by the SOCSNE model, and then solving the problem with tailored incentives for a group of customers addressing the trade-off between a larger market share and the costs of providing incentives are the other class of contributions of this chapter. We model the LCIP as an integer program. Since this model is computationally very hard to solve, we develop a preprocessing procedure and a simple iterative strategy to solve it. As noted earlier the LCIP is applicable in a larger number of settings. For instance, while it is considered within the framework of the product design process in this study, in practice it may be used tactically and operationally during the marketing phase. Specifically, the population to which the product is marketed to is typically larger than the population used for the conjoint study. Further, social network structures are dynamic and may change over time. Thus, with a product (or product line) in place a marketer could benefit from the LCIP model to analyze the tradeoffs and incentives required to reach a desired fraction of the population. Although the LCIP model considers incentives in terms of customer utilities, one should note that following Miller et al. (2011) it is possible to infer dollar values from customer utilities.

There are several natural directions for future research. Broadly, they all encompass the expansion of product or product-line design problems to settings with social network or peer influence effects. They include (i) pricing of the levels of attributes for a product as part of the design process, (ii) consideration of alternate assumptions of product selection amongst consumers (i.e., instead of the highest choice pattern where each consumer deterministically self-selects the product from the line that provides her the highest surplus one could consider the multinomial choice logit model (Chen and Hausman, 2000)), and (iii) extension of the problem to stochastic network settings (i.e., one in which the social network changes over time).

Chapter 3

The Least Cost Influence Problem (LCIP)

3.1 Introduction

In the previous chapter, we have introduced the least cost influence problem as a complementary piece to the product design problem; the solution of the integer programming model for the share-of-choice problem with network effects implicitly required people to make buying decisions together (see Section 2.2). To overcome this bottleneck, we searched for the best way to persuade one person out of the group to initiate the buying process through incentives. Working through iterations, the objective was to identify the most critical nodes in the network in order to minimize the amount of incentives given.

In this chapter, we model the LCIP in a more generalized setting, independent of the product design problem. Here, the objective is still to find a group of individuals to give incentives to, but the aim here is for the spread to reach to a portion of the whole network. This is relevant when it is desired that a certain threshold fraction in a society/social network be reached (e.g., immunizing at least a certain proportion of the population). In the previous model, every neighbor had the same amount of decrease (increase) in one's hurdle (utility) for the product. The model considered here allows each person to be influenced in a different manner from different neighbors.

In the next section, Section 3.2, we briefly review the related literature on information propagation in the computer science and information systems research areas. In

Section 3.3, we show the mixed integer programming model for the general setting. In Section 3.4, we focus on tree networks and show that LCIP over a tree network can be solved in a polynomial time in certain cases. Next, we provide the algorithms to solve this problem in Section 3.5. We propose a dynamic programming algorithm for the LCIP on tree networks in Section 3.5.1 and show that it is equivalent to a greedy algorithm in Section 3.5.2. In Section 3.6, we show that the LCIP over a tree network becomes a knapsack problem when influence from neighbors are not equal to each other for a person (more specifically, an instance of a knapsack problem can be polynomially transformed to an instance of an LCIP). We conclude this chapter with an example of the proposed algorithms in Section 3.7 and with concluding remarks in Section 3.8.

3.2 Information Propagation

Finding the most critical nodes to achieve a certain spread has found a large place in the computer science and complex networks literature. The abundance of data on social networks and the advancement of technologies that enable their collection, have encouraged computer and information scientists to search for reasoning behind actions of large groups. These researchers are interested in knowing how any “information”, whether it is a disease, a behavior, or an advertisement, spreads. Research on complex networks studies the nonlinear dynamics of observing interactions among humans/cells/websites at the individual level and understanding the emerging collective behavior, and it is a part of a broader research on complex systems (Strogatz, 2001).

3.2.1 Viral Marketing Literature

Influence spread over a network of individuals via electronic sources such as internet or wireless networks to advertise a particular product or service is called Viral Marketing. It was first used in 1997 in a Netscape newsletter, defined loosely as "network-enhanced word-of-mouth". The first idea came from Hotmail where a promotional pitch with a clickable URL was included in every message sent by a hotmail user (Jurvetson, 2000). The main characteristics of viral marketing is that every customer becomes an involuntary salesperson simply by using the product and talking about it to his/her friends. As the name suggests, the origins of the idea is that the spread of influence is similar to a virus carrying a disease over a network of people and affecting them throughout a certain period of time. Viral marketing is also referred to as word-of-mouse or word-of-keyboard. Network-based marketing is another term that refers to a collection of marketing techniques that take advantage of links between consumers to increase sales. Instances of network-based marketing have been called diffusion of innovation, buzz marketing and viral marketing as well. Three modes of network-based marketing have been defined (Hill et al., 2006): *Explicit advocacy:* People become vocal advocates of product. *Implicit advocacy:* Cool members of smaller social groups adopt the product, they do not explicitly speak about the product but use it. *Network targeting:* A firm creates a market with respect to prior purchasers' social-network neighbors.

In a broad sense, viral marketing has been studied from two main points of view. The first one is from the marketing side which is more consumer-oriented and the other is a sociological perspective which focuses on social network structures. With the recent advancements in technology, researchers have access to large online networks and computer science, physics and complex systems research have begun to study real network structures in detail. The first group is more interested in the characteristics of this spread and on how to handle and control for the different components of this marketing structure. On the other hand, the sociological studies are more interested in the structure of the network and how it affects social influence. From the computer science perspective, diffusion models of viral marketing are used in identifying the most critical nodes on a network to maximize the number of affected people from the proposed product, idea or service. The individuals in both types of studies consist of ones who have already adopted the information and the ones who have not yet adopted it. In this respect, the LCIP proposed in this section lies within the computer science perspective in which we try to identify a set of people to persuade to adopt and to maximize the number of adopters as a result of their influence on the others. In this section we will take a look at a nonexhaustive viral marketing literature in the marketing, computer science and complex systems as these are the most related papers to our work.

Starting with the marketing side, viral marketing has been in the focus for quite some time now. Although it has been seen like a random process earlier (Goldenberg et al., 2001), the marketing industry started to see the potential it has and began to direct it rather than only trying to manage it so that they can take control over it and use it as an advertising tool. A study by Reichheld (2003) showed that willingness of consumers to recommend a company to their friends is by far the best predictor of a company's growth. Money et al. (1998) studied the factors that affect the word-of-mouth referral behavior in international business transactions and studied them as native culture of the buyer, location of operation and relative location in general terms. More specifically, they discussed the source nodes, strong ties on the network and measured the effect of centrality on a network influence. Jurvetson (2000) suggested that viral marketing is more powerful than third-party advertising because it conveys an implied endorsement from a friend. The paper also stated the aspects that characterize a successful viral marketing strategies such as; the product is being offered for free only on the internet and via no other distribution channel, the offer contains a real customer value and the first "carriers" are chosen very carefully. The paper also discusses the "first-mover advantages" in viral marketing.

Mobile viral marketing is one of the word-of-marketing communication arenas which emerged as the number of individuals who own a cell phone has increased dramatically. Wiedemann (2007) examined 34 case studies to identify relevant characteristics of mobile viral marketing. In this study, a description model of mobile viral marketing as well as a derivation of four mobile viral marketing standard types are presented.

The nature of viral marketing and the access to large amounts of data through technological advancements allows many experimental studies in marketing literature. Van den Bulte and Stremersch (2004) looked at different contagion mechanisms and provided empirical support for the role of income heterogeneity in diffusion of consumer innovations. Van den Bulte and Joshi (2007) formalized the previous theoretical arguments on social structure and diffusion, and modeled the diffusion structure to generate more refined theoretical insights with empirical analysis of 33 different data series. More recently, Berger and Milkman (2012) took a psychological approach to understanding diffusion and studied how content characteristics affect virality. Determining influential users in internet social networks and finding critical nodes to seed to start an adoption is a topic that is of interest to both marketers and network scientists (Goldenberg et al., 2009; Stonedahl et al., 2010; Bakshy et al., 2011).

Using communication data, Hill et al. (2006) showed that consumers linked to a prior customer adopt a service at a rate 3-5 times greater than baseline groups selected by the best practices of the firm's marketing team. Interestingly, Hill et al. (2006) tried to separate word-of-mouth effects from homophily, where homophily suggests that people that are linked to each other are like-minded and like-minded people tend to buy the same product. If we are not sure about the way of communication that takes place, it would be difficult to understand what causes the influence. To overcome this effect, they used propensity scores. They matched the scores in the treatment and control groups using demographic data which helped to understand if homophily was present or not. Flagging the customers NN (network neighbor) indicating whether they had communicated with a current user of the service in a time period prior to the marketing campaign, they then observed purchase behavior.

Viral marketing has received an increasing attention from the computer scientists and engineers who concentrate more on the structure of the network and the relationship among the components of the network. In the early study by Domingos and Richardson (2001), the authors proposed a "network value" for each customer with respect to the expected increase in sales to others that results from marketing to that customer. The paper concentrated on what determines this network value of a customer. Several examples to this are the connectivity of the customer in the network, whether the taste of customer and the product being marketed are similar, the second third and higher degree connectivity of the customer which would lead to faster spread of the influence. By viewing the market as a social network, they modeled it as a Markov random field and seeked for the choice of which customers to market to, and for the way to estimate what customer acquisition cost is justified for each customer. They mined data from the EachMovie collaborative filtering system and showed that their modeling framework have advantages over traditional direct marketing.

In 2002, in another study (Richardson and Domingos, 2002) the authors search for best viral marketing plan by mining data from Epinions knowledge-sharing site, where customers review products and advise each other. Social network here is the trust relationship among the reviewers of products. The knowledge of the network is partial in this paper, and that gathering that knowledge can itself have a cost. Some of the parameters were estimated such as the effect that marketing has on a customer's probability of purchasing, (i.e., the amount of influence between customers). They estimated the purchasing probability using a Bayes model assuming that a user is more likely to purchase a product if it was reviewed by a person he trusts. They extended their previous work (Domingos and Richardson, 2001) and showed how to find the optimal viral marketing plans that maximized positive word-of-mouth among customers using continuously valued marketing actions, and they reduced computational costs. Later Domingos (2005) looked back and presented a general introduction on the importance of viral marketing mentioning that lack of data caused social network models traditionally to be descriptive rather than predictive, and that having the data made the previous two studies possible. The author argued that in traditional direct marketing customers are presented products only if the price is greater than the production cost, however, in viral marketing a free sample which would mean an expense at current time can cause a larger number of people to know about the product and lead to higher sales in the long term.

Structural measures of influence in social networks are studied by several researchers. As mentioned in Section 2.2, Kempe et al. (2003) studied the k-max influence problem; for a parameter k, find a k-node set of maximum influence. Adopters of the innovation is represented as active nodes. The authors focused on two basic diffusion models for the spread of an idea or innovation that have been studied well in social sciences; linear threshold model and independent cascade model. They showed that this problem is NP-Hard and provided the first provable approximation guarantee (within 63% of optimal) for several classes of models, with a natural greedy strategy. Their algorithm outperformed node-selection heuristics based on the well-studied notions of degree centrality and distance centrality in terms of the set of resulting active set size.

Leskovec et al. (2007) concentrated on person-to-person recommendation networks and how recommendations spread. They analyzed how user behavior varies within user communities defined by a recommendation network. The growth of recommendation network over time and the effects of sender and receiver recommendations are studied. Using a stochastic model, they successfully identified communities, product and pricing categories for which viral marketing seems to be very effective. They found that the probability of purchasing a product increases with the number of recommendations received but quickly saturates to a constant and relatively low probability.

Complex systems approaches have also been used to look at the underlying process of word-of-mouth advertising. Goldenberg et al. (2001) used a stochastic cellular automata, generated data and analyzed the results to understand the impact of strong and weak ties between people on the speed of acceptance of a new product. Here, the strength of the ties affected the probability of an agent to be influenced by its neighbors. In another study, Garcia (2005) studied the use of agent-based modeling (ABM) to study innovation and new product development research. The paper explained the benefits of ABM, and presented possible applications to use ABM for new product development and innovation. The paper provided a quiet rich literature review and introduced three potential areas of research for ABMs as, diffusions of innovations, organizations and knowledge/information flows.

While communicating and spreading an idea or the influence about a product, people can also talk about their negative experiences as well. This could lead to a decrease in the sales revenue. In the extreme cases, there are many examples of hate sites. It is usually assumed that negative word-of-mouth is spread further than its positive counterpart (Helm, 2000). Another possible negative effect occurs when customers feel used if they have doubts about whether positive comments are really generated by regular people or by that company. Luo (2009) studied the long-term impact of negative word-of-mouth on cash flows and stock prices and provide insights for marketing research and viral marketing management.

Studies over social networks are sparse in the operations research literature. A recent paper in the operations research literature is by Dawande et al. (2011). In their paper, they studied searches of sets of nodes over social networks and in this respect they defined two integer programs; The Elite Group Problem (EGP) and The Portal Problem (PP) corresponding to the two set-based notions which are influential sets and central sets, respectively. They provide a variety of algorithmic results and discuss computational complexities of both problems.

3.3 Problem Definition in a General Setting

Given an undirected graph $G = (V, E)$, let vertex (node) set $V = \{1, 2, ..., n\}$ denote the set of people in the network and edge (link) set E represent the connections among people. Each node $i \in V$ is associated with a hurdle b_i. Let d_{ij} (defined over all edges $(i, j) \in E$ so that $d_{ij} = 0$ if these two nodes are not adjacent to each other) denote the "influence factor" which captures how much another node j influences node i if node j adopts. This corresponds to $\Delta_i a_{ji}$ in the model in Chapter 2. There the influences Δ_{ik} and Δ_{ij} were equal to each other and therefore were represented as Δ_i where i is the person being influenced by neighbors k and j. We follow the linear influence structure in the previous chapter, and also assume that node i adopts the information if and only if the sum of the influence from neighbors and the incentive he receives is greater than or equal to his hurdle. Remember that the hurdle in the previous chapter was the value at which one would be indifferent between buying a product or not. Here, in a more general setting, hurdle has a similar purpose. It serves as a limit which needs to be surpassed for the node to adopt. However, we do not have utilities in this problem. The reader can read the new hurdle as the remaining from the hurdle when utilities are subtracted. The goal of the problem is to minimize the amount of incentives given while guaranteeing that a fraction (α) of the market will adopt at the end. (We use the terms adopt and buy interchangeably throughout this thesis.)

We model the LCIP in a general setting using a mixed-integer program. Again, to capture the order of buying among customers, we introduce time periods, $t = 1, 2, ..., T$. Since there are $|V|$ nodes, we can limit the time indices, $T \leq |V|$. In this model we have four types of decision variables. Let $x_i = 1$ if node i eventually adopts by the time T, and 0 otherwise. Let $y_{it} = 1$ if node i adopts in time period t, and 0 otherwise. Let $v_{it} = 1$ if node i adopts for the first time in time period t and 0 otherwise. Finally let p_i denote the inducement paid to node i. The mathematical formulation is as follows:

$$\begin{array}{llcllr}
\textbf{LCIP:} & \text{Min} & \sum_{i \in V} & p_i & & (3.1) \\
\text{subject to} & p_i + \sum_{j \in V} d_{ij} y_{j,(t-1)} & \geq & b_i y_{it} & i \in V, t = 2, ..., T & (3.2) \\
& v_{it} & = & y_{it} - y_{i,(t-1)} & i \in V, t = 2, ..., T & (3.3) \\
& \sum_{t=1}^{T} v_{it} & = & x_i & i \in V & (3.4) \\
& \sum_{i \in V} x_i & = & \alpha |V| & & (3.5)
\end{array}$$

where $x_i, y_{it} \in \{0, 1\}$ and $v_{it}, p_i \geq 0 \quad i \in V,\ t = 1, 2, ..., T.$

The objective of this formulation is to minimize the sum of the incentives given over the network. The first constraint provides the buying condition for each person. The hurdle, b_i, for person i is compared with the sum of the total influence over all neighbors, $\sum_{j \in V} d_{ij} y_{j,(t-1)}$, and the amount of incentive received, p_i. With constraint 3.3 we introduce the variable v_{it} and keep track of when a person adopts for the first time. We assume there is no churn. That is, adopters do not switch to not adopting at later points of time. By summing over all periods in constraint 3.4, we decide if a person adopts. With constraint 3.5, we make sure the number of adopters constitute the required portion of the population.

3.4 Focus on Tree Networks

A tree network is a cycle free undirected graph where there is only one simple path between any two vertices. In our solution procedure we utilize the star subnetworks within a tree network. A star network consists of one root node for the tree and possibly many leaf/end nodes connected to it. The degrees of leaf nodes are equal to one, whereas the degree of the root node is equal to the number of leaf nodes. Figure 3.1 shows an example of a tree network with 10 nodes. We have one root node (though it is not unique). On the right, we show one way to decompose the tree into its star subnetworks. There are three star subnetworks in this example. Nodes 2 and 4 become the root nodes for the other two subnetworks.

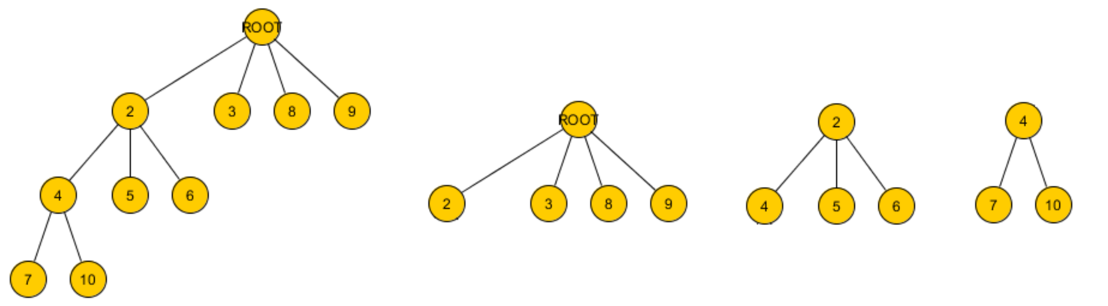

Figure 3.1: An example of a tree and its star subnetworks.

The mixed integer program for the LCIP becomes intractable as the size of the network increases. The LCIP is an NP-Hard problem for general networks. However, we show that it is polynomially solvable on tree networks under the assumption that all neighbors of a node exert equal influence.

Although the focus on trees as social networks may seem limited, they are prevalent. Recent visualization tools allow network data to be better read for data scientists. Popular visualizations include data from Facebook, Twitter and Wikipedia articles. Various aspects of these different environments may lead to visualizations of star subnetworks and tree networks, including infection trees on Blogsphere as described by Adar and Adamic (2005), Facebook page trees as described by Sun et al. (2009), and "retweet trees" as described by Kwak et al. (2010) and Sadikov et al. (2011). Therefore, our focus on tree networks is still applicable to a wide variety of real life information on networks. Further when a network can be decomposed into trees, our algorithm can be used to develop a fast dynamic programming (though not polynomial) algorithm for sparse networks.

In the next section we propose a dynamic programming approach to solve this problem when the influence factor, d_{ij}, is the same for all neighbors, i.e., $d_i = d_{ij}$, $\forall i \in V$. With this assumption, each node in the network is associated with only two numbers, b_i and d_i. We also assume the desired market share is the entire population size of the network, i.e., $\alpha = 1$. In this respect, we can express the LCIP problem over a tree network as follows. Given the network below (Figure 3.2) and given the corresponding hurdles and influence factors, choose the set of nodes, S, that will let the information be spread over the whole network while receiving the minimum incentives.

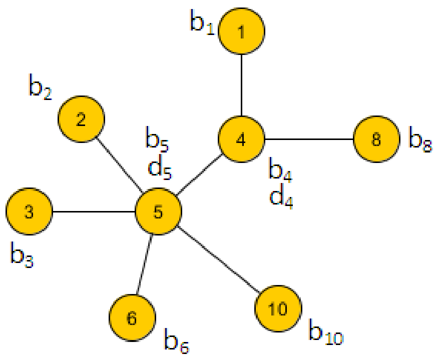

Figure 3.2: The LCIP over a tree network.

This figure shows all the nodes before adoption started over this network. Hurdles are shown with b_i and influence factors are shown with d_i. If none of the nodes is given incentives, no adoption will take place in this network. If, for example, node 3 is paid incentives in the amount of b_3, node 3 would adopt and the hurdle for node 5 would decrease by d_5. Then paying $b_5 - d_5$ to node 5 would lead node 5 to adopt. This would be followed by nodes 2, 6, and 10 to adopt in the next period without any incentives. It would also decrease the hurdle for node 4. This is an example of how the diffusion takes place over the network. In this example, nodes other than 4 and 5 are leaf nodes[1]. In terms of solving the problem, paying incentives in the amount of b_4 and b_5 to nodes 4 and 5 at the beginning of the adoption would suffice to reach all nodes. However, there may be a cheaper solution where some of the leaf nodes are paid their current hurdles, and nodes 4 and 5 are only partially paid. In the next section, we will look at some properties of the problem and the solution to understand how we can reach a solution for the LCIP over a tree.

3.4.1 Properties of the problem and the solution procedure

Without loss of generality, we assume that if all nodes in the set of neighbors, N_i, of a node i adopt then the total influence from these neighbors will be greater than or equal to the node's hurdle, i.e., she will adopt too. Otherwise, it would mean the only way she could adopt would be when an incentive of $b_i - |N_i|d_i$ is given at the outset. This difference would be independent of the solution of the problem and therefore can be seen as a sunk cost and omitted. Consequently, we can say $b_i \leq |N_i|d_i \quad \forall i \in V$.

[1]See Section 3.4.1 property 2 for why we do not show d_i for the leaf nodes.

Letting ρ represent any root node in a tree network, we now describe additional properties about the problem and its solution:

1- In the initial network, $b_i \geq d_i \quad \forall i \in V$.

(The inequality $b_i \leq d_i$ says that influence on node i is larger than i's hurdle meaning a single buyer neighbor will be sufficient for the node to buy. This is the same case if b_i is set to be equal to d_i since the extra influence of $d_i - b_i$ is not required.)

2- We can assume $b_i = d_i \quad \forall i$ where i is a leaf node.

(Since $\alpha = 1$, we know the whole network will buy at the end. A leaf node has a single link connected to it (the root node) and since without loss of generality we have assumed that $b_i \leq |N_i| d_i$, $b_i \leq d_i$ and d_i cannot be less than b_i, where $|N_i| = 1$. Then property 2 follows from property 1.)

3- In a star network, if the root adopts, then all the leaf nodes connected to it will adopt too.

(Since $b_i = d_i$ for a leaf node, the influence from the root node will be sufficient to persuade the leaf node to adopt.)

4- In a star network, if a leaf node i is paid incentives, it is paid in the amount of its hurdle, b_i.

(If there is influence from the root node, then no incentives would be required. If there is no influence from the root node and the incentive given to the leaf node is less then b_i, then the leaf node would not adopt.)

5- The initial network can be truncated by eliminating leaf nodes i for which $b_i \geq b_\rho$.

(Since $b_i \geq d_i \quad \forall i \in V$ in the initial network, $b_\rho \geq d_\rho$ and so $b_i \geq b_\rho \geq d_\rho$. This means if the leaf node receives an incentive of b_i, its influence on the root node will be d_ρ, which is less than or equal to b_i. So the cost (incentives given) of paying such leaf nodes is larger than the benefit (the influence on the root node) we get.)

6- If there are only two people in a network that are connected to each other, it is always cheaper to give incentives to the one whose current hurdle amount is the less than the other one.

(The other one will automatically buy from the influence.)

7- One can think a natural upper bound would be to give incentives to everyone but actually, we never need to give incentives to all because we can always leave at least one person out.

(Since all her neighbors would already have adopted, she will buy due to the influence from them without requiring any incentives.)

8- A naive lower bound can be obtained by giving incentives to the person with the least current hurdle.

(It may be the case that once she buys, all will be affected and buy.)

3.5 Algorithms to Solve the LCIP over a Tree Network

In this section, we explain our dynamic programming algorithm to solve the LCIP over a tree network when the influence factor d_{ij} is the same for all neighbors j of i, and then show how it can be viewed as a greedy algorithm. The dynamic programming algorithm uses the advantage of the star subnetworks in a tree network and solves the problem over each star network and suggests a method to compress them into a single node. This promises that if we can iteratively reduce the tree network into a single star, we can solve the problem for the tree. A consequent product of this dynamic algorithm is the greedy algorithm, which is computationally a lot easier than the dynamic algorithm. In the greedy algorithm, nodes are selected in the ascending order of $\min\{b_i, d_i\}$, they are paid in the amount of their current hurdles and the hurdles for the neighbors of the selected node are updated at each iteration. After giving the details on both algorithms, we explain how the greedy algorithm is an interpretation of the dynamic algorithm.

3.5.1 Dynamic Programming Algorithm

In the dynamic programming algorithm, we fragment the tree network into subproblems. Each subproblem is solved optimally leading to an optimal solution for the problem. Subproblems are defined on star subnetworks which are collectively exhaustive. Solving a subproblem corresponds to *identifying* the nodes to be given incentives and *calculating* the total cost of incentives required for the adoption to spread over that star subnetwork. Before solving the problem over a star, an important step is to consider the link that connects the root node of the star to the rest of the network, which we will call the "connector" link. We solve the problem over a star as if the influence over this connector has been realized, meaning the influence from the rest of the network has traveled over this link and the hurdle of the root of the star has been updated (reduced). In this way, we will calculate the cost of solving this part of the problem assuming there is an external influence from the rest of the network. Next, this star is compressed into a single node (with a new hurdle and a new influence factor) and it becomes a leaf node for the star subnetwork in the next iteration. When we compress the star, we fix the assumption over the connector link. We repeat this process until we are left with a single star network. Once we solve the problem over the last star, a backtracking method is applied to identify the nodes which have been given incentives.

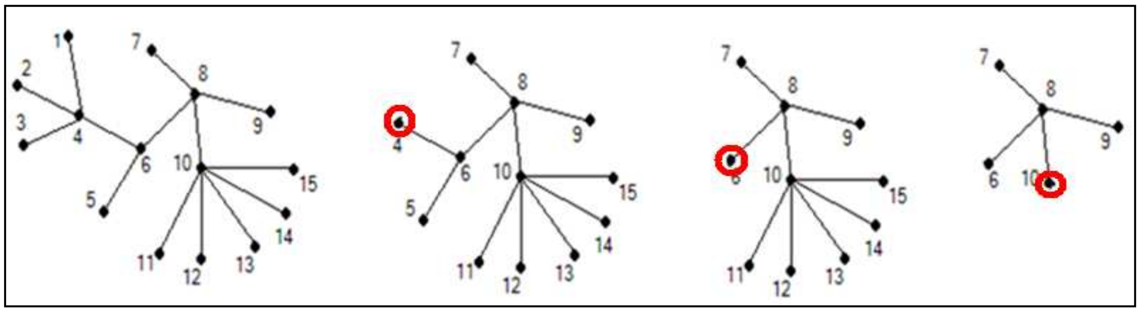

Figure 3.3: The main idea behind the dynamic programming algorithm is compressing the tree into a star network.

Figure 3.3 is an example of this process. In this figure, we start with a tree network on the left, and the stages of the dynamic programming for this network are shown as we move to the right. In the first iteration, the star with leaf nodes 1, 2 and 3 are compressed into a single node, node 4. In the next iteration, leaf nodes 4 and 5 are compressed to a single node. In the last iteration, leaf nodes 11, 12, 13, 14 and 15 become node 10, and we are left with a single star with root node 8.

The pseudocode of the algorithm is as follows. We explain each part in detail next.

```
1.  begin
2.      for each star network do,
3.          SolveStar
4.          CompressStar
5.      TotalCost
6.  end
```

SolveStar: In a star network if the root node, ρ, is persuaded to adopt, all the leaf nodes will definitely adopt in the following period (see Section 3.4.1 property 3). Therefore, an upper bound for the cost of a star network is the current hurdle of the root node. However, a cheaper solution may also exist. In the SolveStar part of the dynamic programming we search for this solution.

The first step is to let the influence over the connector link to be realized and update the hurdle of the root node. To solve the problem over a star, we know we need to concentrate on the root node and try to get it to adopt. Since all the leaf nodes have the same influence over the root node, when incentives need to be given leaf nodes would be preferred in the order of their hurdles starting from the smallest, since it will be cheaper this way. Actually, leaf nodes with hurdles greater than the minimum of the root's hurdle and influence factor can be neglected. It is easy to see why this is the case (similar to property 5 in Section 3.4.1). If the minimum of b_ρ and d_ρ is the influence factor, d_ρ, then the influence from paying b_i to the leaf node i, will be d_ρ and since d_ρ is less than b_i, we will be gaining less than what we spend. We could as well give this incentive in the amount of d_ρ directly to the root node and secure the same decrease in the hurdle. If the minimum of b_ρ and d_ρ is the hurdle, b_ρ, and we give the incentive to the leaf node i in the amount of b_i, then we are spending b_i and decreasing the hurdle of the root node by b_ρ (since this is all the amount the root node needs to adopt). Since $b_i > b_\rho$, again we will be spending more than what we are gaining. In either case, we could be better of by giving the incentive directly to the root node, and never use such leaf nodes. This also means we can eliminate these nodes (whose hurdle values are greater than or equal to the minimum of the hurdle and the influence factor of the root node) before solving the problem. We collect the remaining nodes (with hurdles less than the minimum of b_ρ and d_ρ) in set S. Providing incentives to these nodes is advantageous (less expensive) in terms of decreasing the hurdle of the root node. This is why we choose nodes from this set and eliminate all leaf nodes j, for which $b_j \geq \min\{b_\rho, d_\rho\}$. Other nodes are put in set S and selected in the ascending order of their hurdles.

To calculate the cost of the star, we calculate the number of adopter neighbors that is sufficient for the root node to decide to adopt, which is $\lceil \frac{b_\rho}{d_\rho} \rceil$. If there are sufficient number of nodes in set S, we get the cost of the star by comparing the costs under two cases. The difference between the cases is whether the root node is paid partial incentives or not. In the first case, the root is convinced to adopt by the influence of her neighbors. So the first $\lceil \frac{b_\rho}{d_\rho} \rceil$ leaf nodes in set S are paid. In the second case, we pay $\lceil \frac{b_\rho}{d_\rho} \rceil - 1$ nodes in set S and try to cover the remaining amount by paying directly to the root node. This partial payment may be smaller than $\min\{b_\rho, d_\rho\}$, and may also be smaller than the b_i of any of the nodes in set S. So we are better off by paying it directly to the root node. The result of the comparison between these two cases changes depending on the difference between the hurdle and the influence factor of the root node. When $S = \emptyset$ or $|S| < \lceil \frac{b_\rho}{d_\rho} \rceil$, all nodes in S are selected and the root node is given incentives to complement the total amount. The following two networks are examples where the minimum is different for the two cases.

In the networks in Figure 3.4, there are three nodes. Nodes 1, 2 and 3 form the star. Hurdles and influence factors are given next to each node. For example in Case (a), for node 1, hurdle equals 18 and influence factor is 10. These numbers are equal to each other for the leaf nodes (see Section 3.4.1 property 2), so only one number is shown on the picture, e.g., hurdle and influence factor equals 5 for node 2. In both cases (a) and (b), the number of adopters needed is equal to 2 and set S contains nodes 2 and 3. Nodes

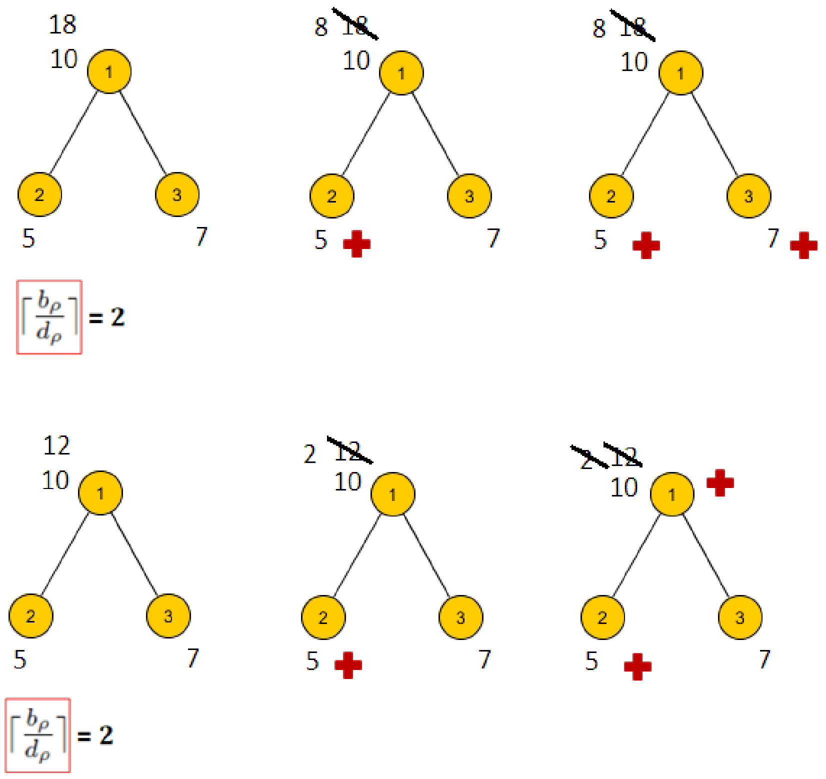

Figure 3.4: Cost of a star network in the SolveStar algorithm, (Case (a) and Case (b)).

which are given incentives are marked with a plus sign. In case (a), when the incentives are given to the two nodes in S, the total cost is 5+7=12. If the incentive is given to a single node and the remaining is covered by the root node, the cost becomes 5+(18-10)=13. Thus, it is less costly to give incentives to both leaf nodes in set S. In case (b), when the incentives are given to the two nodes in S, the total cost is 5+7=12. However, if the incentive is given to a single node and the remaining is covered by the root node, the cost becomes 5+2=7 which is the better choice.

The algorithm SolveStar[2] can be summarized as follows. For ease of exposition, we use the same notation and do not use indices to differentiate each star. Let V be the set of nodes, L be the set of leaf nodes, S^* be the set of leaf nodes that are given incentives,

[2]If the tree network is a star at the outset, then Step 1 of the SolveStar algorithm can be omitted.

and ρ be the root node of the star network. The cost of a star equals C. The records of partial incentives are kept in an array P. The pointer $B[i] = j$ shows that when we see node i in the algorithm after a star has been compressed, it actually represents (points to) node j. This will be useful when we calculate the total cost and identify the nodes with incentives. At the beginning of the algorithm, $P[i]$ is set to 0 for each i and each node points to itself, i.e., $B[i] = i$.

```
algorithm SolveStar
1.    Update hurdle for the root, $b_\rho = b_\rho - d_\rho$.
2.    Let $S = \{i \mid b_i < \min\{b_\rho, d_\rho\},\ \ i \in L\}$.
3.    Order nodes in $S$ with respect to $b_i$, $i \in S$ in ascending order
4.    If $|S| \geq \lceil \frac{b_\rho}{d_\rho} \rceil$, cost of the star, $C$, equals to the minimum of the following;
5.        Select the first $\lceil \frac{b_\rho}{d_\rho} \rceil$ nodes in $S$ and give incentives equal to their
          hurdle.
6.        Select the first $\lceil \frac{b_\rho}{d_\rho} \rceil - 1$ nodes in $S$ and give incentives equal to their
          hurdle. Pay the remaining to the root, $P[\rho] = b_\rho - (\lceil \frac{b_\rho}{d_\rho} \rceil - 1)d_\rho$.
7.    Else cost of the star, $C$, equals,
8.        For all nodes in $S$, give incentives equal to their hurdle. Pay the
          remaining to the root, $P[\rho] = b_\rho - |S|d_\rho$.
9.    Remove the selected nodes from set $S$ and place them in set $S^*$.
```

CompressStar: Once the cost of the star is determined, the star is converted into a single node with hurdle, b'_ρ and influence factor, d'_ρ. In this part of the algorithm, the connector link between the star and the rest of the network plays an important role. Remember that the influence over this link is assumed to be realized when the problem is solved over the star. So the cost calculated for the star is the cost assuming this influence. When the star is compressed, we take into account the fact that this influence has actually not been realized and we include it as the hurdle of the new node. Determining this hurdle is a bit tricky and is determined by considering the following two cases;

Case 1) Remaining hurdle at the root node.

Case 2) Hurdle of a leaf node that is in S.

The pseudocode for the CompressStar algorithm is given as follows.

```
algorithm CompressStar
1.    New hurdle $b' = \min\{d_\rho, b_\rho + d_\rho - |S^*|d_\rho, \min_{i \in S} d_i\}$.
2.        If $b' = \min_{i \in S} d_i$, then $B[\rho] = B[i]$.
3.    New influence factor $d' = b'$.
```

In Case 1, we basically put the assumed influence over the connector back into the problem and calculate the remaining hurdle. There are two types of solutions in Case 1 due to the solution of the StarSolve algorithm. In the first type of solution, there's no extra influence on the root node from the leaf nodes and the hurdle for the compressed node is equal to the influence factor of the root node, d_ρ which was assumed over the connector link. In the second type of solution, the influence from the leaf nodes exceeded the hurdle of the root node and this excess influence is subtracted from the hurdle of the root node after the influence over the connector link is put back, $b_\rho + d_\rho - |S^*|d_\rho$. This classification among the two solutions can also be done by looking at whether the root node is paid any incentives in the SolveStar solution. When the root node receives a partial payment, this suggests that the hurdle for the root node is covered exactly and the hurdle of the new node is simply the influence factor, d_ρ. However, if there is no partial payment, it means the hurdle for the root node is covered through the influence from the leaf nodes and this influence may actually be larger than the hurdle itself. In this second situation (when the root node does not receive partial payment), the excess influence from the leaf nodes are subtracted from the influence factor of the root node when calculating the hurdle of the new node. We give examples to both of these types of solutions below in Figure 3.5 and 3.6 with explanations below.

In Case 2, hurdle of a leaf node in S is carried onto the next iteration as the hurdle of the new compressed node. This is possible when there is a leaf node in set S (i.e., it has not exerted influence on the root node). Since any node i in set S has a hurdle less than the influence factor of the root node, it may be less costly to pay the amount d_i to the node i in set S than to pay the remaining hurdle to the root node. Consequently when this is the case, we select the hurdle of the new compressed node as d_i. We give an example for this case in Figure 3.7. Since the compressed node becomes a leaf node, the influence factor of this node is equal to the hurdle of the node in both cases.

In these examples, in Figure 3.5, 3.6 and 3.7, there are four nodes. Nodes 1, 2 and 3 form the star and the link between 1 and 4 is the connector link that connects this star to the rest of the network. The numbers next to the nodes represent the hurdle and the influence factor for that node. The four networks in each figure represent the evolution of the star as the hurdles are updated. The “+” sign on a link in the star represents that a diffusion takes place through that link.

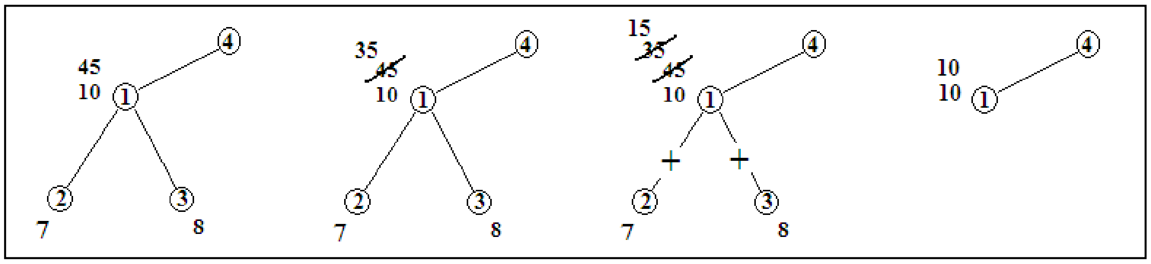

Figure 3.5: Example for Case 1 with partial incentives given to the root node (type 1).

The following is an example of the first type of solution in Case 1. In Figure 3.5, the hurdle for the root node, 1, is equal to 45 and the influence factor is 10. The hurdles for nodes 2 and 3 are 7 and 8, respectively. Initially, the influence from link (1,4) is temporarily assumed to exist and the hurdle for node 1 is updated, 35=45-10. Next, following the algorithm SolveStar, nodes 2 and 3 are given incentives and their influence decreases node 1's hurdle again. This decrease equals to 20 since there are two adopters. The remaining amount, 15, is paid as a partial incentive to the root. The summation of the influence from the leaf nodes and the partial payment to the root node add up to exactly the hurdle of the root node (the root node is paid an incentive). When we bring back the influence over the connector link, the hurdle of the compressed node is exactly equal to the amount of the assumed influence factor, $d_\rho = 10$.

The following is an example for the second type of solution in Case 1. Hurdles for the three nodes are equal to 25, 2 and 4, respectively. Again following SolveStar, the two leaf nodes are selected and given incentives. As mentioned above, here is a slight difference on the hurdle of the compressed node. In this solution, the total influence from the leaf nodes exceed the current hurdle of the root node (there is no need to give any partial payments to the root node). The excess influence from the leaf nodes covers a portion of the hurdle and therefore the remaining becomes the new hurdle, $b_\rho + d_\rho -$

$|S^*|d_\rho = 15 + 10 - 20 = 5$.

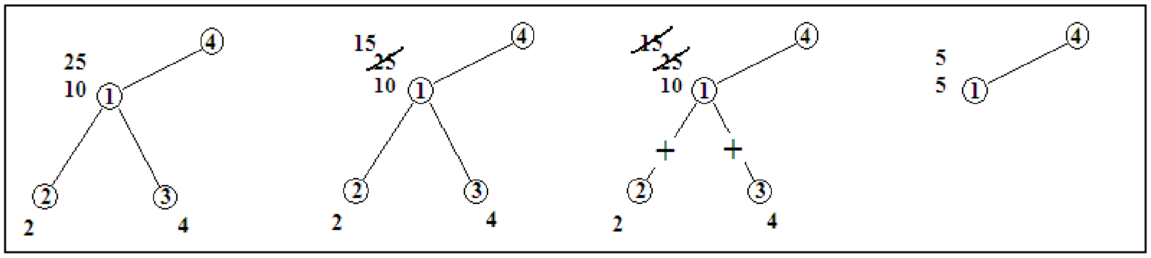

Figure 3.6: Example for Case 1 with no partial incentives given to the root node (type 2).

In the second case, nodes in set S (nodes that were not selected in SolveStar), if any, can still be in the optimal solution. In Figure 3.7, the solution of the SolveStar selected node 2, but there is a leaf node (node 3) in set S which is not used to solve the problem over the star network. The cost of influencing this leaf node 3 is $b_3 = 4$. The cost under Case 1 is equal to $b_\rho + d_\rho - |S^*|d_\rho = 5 + 10 - 10 = 5$. Comparing the two cases, it is cheaper to select node 3 and the hurdle for the new node equals to 4. This implies that if the new compressed node is given any partial payments in the future iterations, they are actually paid to node 3.

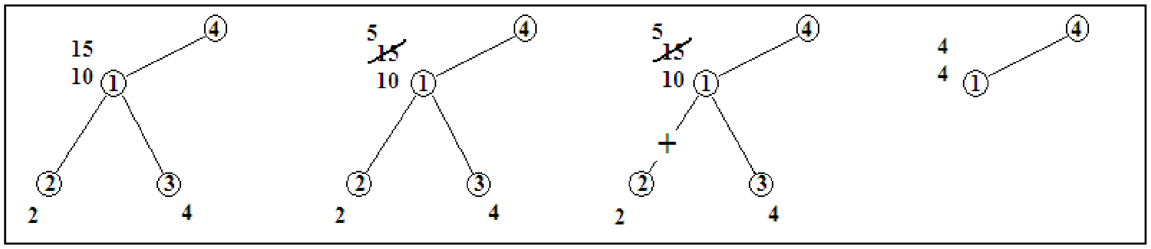

Figure 3.7: Example for Case 2.

TotalCost: Star networks are solved one at a time and compressed to be the leaf node of the next star until a single star is reached. This star is solved in the same way with the other star networks, except now there is no external influence from the outside, so the hurdle does not need to be updated at the beginning. Once the cost of this star is obtained, the algorithm is traced back to calculate the total cost (the objective of LCIP) of the problem and to identify how this cost is distributed among the nodes. We name this part of the algorithm TotalCost. The total cost of the problem is simply obtained as the summation of all costs for each star network. To trace back which nodes have been given incentives to, we use the array B and the set $\overline{S}$ which is the union of all sets of selected nodes for each star. Remember that, the pointer $B[i] = j$ shows that node j is represented by node i in the algorithm after a star has been compressed. Leaf nodes, when selected in the algorithm, always get paid their hurdle (see Section 3.4.1 property 4). If a node i is in $\overline{S}$ and it points to a leaf node j, i.e., $B[i] = j$, in the initial tree network, then the amount of incentive for j is its hurdle, b_j. However, if node j is not a leaf node on the initial tree network, that means it has been the root node for some star network at some iteration and was selected when the star was compressed (Case 1 in CompressStar algorithm). The amount of incentives j receives would be equal to its current hurdle when the star was compressed, b'_j. The nodes which received incentives can be identified by the P vector and are collected in set $\overline{P}$. They are paid in the amount of $P[i]$.

```
algorithm TotalCost

1.        Cost equals $\overline{C} = \sum_{\text{all stars}} C$.
2.        Let $\overline{S} = \bigcup_{\text{all stars}} S^*$.
3.            If $i \in \overline{S}$ and $B[i] \in L$, then pay $b_{B[i]}$ to node $B[i]$.
4.            Else Pay $b'_{B[i]}$ to node $B[i]$.
5.        Let $\overline{P} = \{i | P[i] > 0\}$.
6.            If $i \in \overline{P}$, Pay $P[i]$ to node $i$.
```

3.5.2 Greedy Algorithm

In the greedy algorithm, nodes to give incentives to are selected one by one in the ascending order of $\min\{b_i, d_i\},\ i \in V$. The selected node is paid incentives in the amount of its current hurdle, b_i and the hurdles for the neighbors of the selected node are updated at each iteration. We use the dynamic programming algorithm to show that the greedy algorithm solves the LCIP optimally over a tree network.

Proposition 1: At the outset, suppose node $k = \arg\min_{i \in V} d_i$ is chosen as the root of the last star network in the algorithm.[3] Then, all leaf nodes in the last star network will have a larger d_i than d_k.

Proof: Hurdles and influence factors of the leaf nodes in the last star are determined when stars in the previous iterations are compressed into single nodes. Consider the first compressed star in the algorithm. At this stage, the influence factor of the compressed node, d', is equal to the minimum of the following three; $(d_\rho), (b_\rho + d_\rho - |S^*|d_\rho), (\min_{i \in S} d_i)$ as we have seen in CompressStar algorithm. Since both $(d_\rho) \geq d_k$ and $(\min_{i \in S} d_i) \geq d_k$, if one of these is the minimum, we can easily say $d' \geq d_k$. So we focus on the second case where $(b_\rho + d_\rho - |S^*|d_\rho)$ is the minimum, which can only take place if the root has not been given partial payments (type 2 of Case 1 in CompressStar). This means if $\lceil \frac{b_\rho}{d_\rho} \rceil$ people are required to buy, $\lceil \frac{b_\rho}{d_\rho} \rceil$ people are given incentives. So when costs are compared in SolveStar, the cost of the first $\lceil \frac{b_\rho}{d_\rho} \rceil$ nodes in S is less than the cost of the first $\lceil \frac{b_\rho}{d_\rho} \rceil - 1$ nodes in S plus the partial payments to the root which is $P[\rho] = b_\rho - (\lceil \frac{b_\rho}{d_\rho} \rceil - 1)d_\rho$. So $\exists$ a leaf node, j s.t. $b_\rho - (\lceil \frac{b_\rho}{d_\rho} \rceil - 1)d_\rho > d_j$. We know $d_k < d_j$, $\forall j \in V$, then $b_\rho - (\lceil \frac{b_\rho}{d_\rho} \rceil - 1)d_\rho > d_k$ which is equal to the current hurdle of the new leaf node since

[3]Remember that in the initial graph $b_i \geq d_i\ \forall i \in V$ (see in Section 3.4.1 property 1).

$(b_\rho + d_\rho - |S^*|d_\rho) = b_\rho - (|S^*| - 1)d_\rho = b_\rho - (\lceil \frac{b_\rho}{d_\rho} \rceil - 1)d_\rho$. So in all the cases, d_k is still the smallest.

Corollary 1: Node $k = \arg\min_{i \in V} d_i$ will always be in the optimal solution.

Theorem 1: The greedy algorithm solves the LCIP on a tree optimally.

Proof: Select $k = \arg\min_{i \in V} d_i$ as the root node for the last star. It will be in the optimal solution by Corollary 1. When k is removed from the network and the hurdles are updated, Proposition 1 still holds for each network that appears. So, we would select the node with $\min d_i$, *however* when the hurdles for the neighbor nodes are being updated if $b_j < d_j$ for some node j (thus the hurdle for this node becomes less than its influence factor), then this would effectively mean that the influence factor for node j is equal to the hurdle (see Section 3.4.1 property 1) because the extra influence would be unnecessary. At this point, if the hurdles for nodes connected to node k are processed between two iterations of the greedy algorithm to be $b_i \geq d_i$, the new node with $\min\{d_i\}, i \in V$ can be selected next. Otherwise, we would choose the node with $\min\{b_i, d_i\}, i \in V$. This can be repeated until no further incentives are necessary.

The running time for the greedy algorithm is linear with respect to the number of nodes since each iteration of the algorithm only involves selecting the node that has the minimum of hurdles and influence factors (which is linear) and the number of iterations are limited by the total number of nodes.

3.6 When influence factor depends on the neighbors, $d_{ij} \neq d_{ik}$

We have modeled the influence from neighbors to be equal on an individual mainly due to privacy concerns. In this section we show that the LCIP model reduces to the well-known NP-Hard Knapsack Problem in a different influence model. We have introduced this model in Section 2.1 as Model 3 where the influence is dependent not only on the person being influenced but also the influencer. In this model, the influence was represented with d_{ij} for nodes i and j and $d_{ij} \neq d_{ik}$ for node k, i.e., nodes j and k have different amount of influence on node i. In the LCIP on a tree network with this model, each node can not be represented with two numbers as the current hurdle and the influence factor, but each link would be assigned an influence factor for each direction (Figure 3.8).

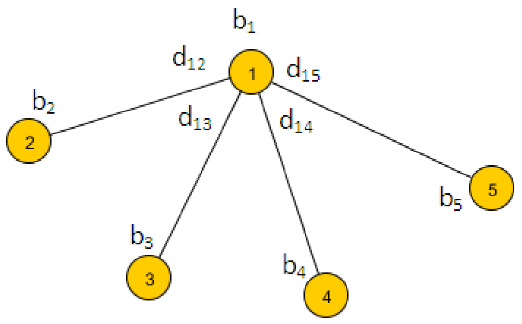

Figure 3.8: LCIP with Influence Model 3.

Now to solve the problem over a star, it is not possible to identify the nodes which have hurdles less than the influence factor of the root node since this factor is not unique anymore. We need to select the nodes which are the cheapest and also have the largest effect on the hurdle of the root node. This problem is similar to the knapsack problem where we want to select items such that the sum of the weights do not exceed a certain threshold while the total value of the selected items is the largest. In the dynamic programming algorithm, we need the solution of a knapsack problem to solve the problem over each star. The following formulation shows how the LCIP over a star network can be modeled using a mixed integer program.

$$\text{Min} \quad \sum_{i \in L} b_i x_i + y \tag{3.6}$$

$$\text{s.t.} \quad \sum_{i \in L} d_{\rho i} x_i + y \geq b_\rho \tag{3.7}$$

$$y \geq 0, \quad x_i \in \{0, 1\} \qquad \forall i \in L \tag{3.8}$$

In this model, letting $L =$ {Set of leaf nodes} decision variables are x_i, $i \in L$ and $y \geq 0$. Let $x_i = 1$ if the leaf node i is given incentives, and $x_i = 0$ otherwise and let y be the amount of incentives given to the root node. This corresponds to $P[\rho]$ in the dynamic programming algorithm given in Section 3.5.1. b_i and b_ρ are the current hurdles for all $i \in L$ and the root node ρ, respectively. The influence of leaf node i on the root node ρ is $d_{\rho i}$. The objective is to minimize the total amount of incentives paid to both the root node, y, and the leaf nodes, $\sum_{i \in L} b_i x_i$. The only constraint forces the root node to adopt if and only if the sum of the partial payment and the influence from the neighbors exceeds the hurdle for the root node. For example, in this model if we decide to give incentives only to leaf node i, i.e., $x_i = 1$, this would mean we pay b_i to node i and the hurdle b_ρ would decrease by $d_{\rho i}$. The remaining amount $b_\rho - d_{\rho i}$ (if > 0) would need to be paid directly to the root node and would set the y variable to be equal to $b_\rho - d_{\rho i}$.

3.6.1 LCIP on a tree network is NP-Hard when $d_{ij} \neq d_{ik}$

To prove that the LCIP on a tree network is NP-Hard, we reduce a known NP-Hard problem to our problem. In this respect, we use a *mixed 0-1 knapsack problem* (Marchand and Wolsey, 1999) which is a knapsack problem with a single continuous variable (KPC) (Büther and Briskorn, 2012).

Given a set of items (N) with profit p_j and weight w_j for each item $j \in N$, the objective is to maximize the profit of a selection of items where the capacity of the knapsack is limited by b. However, the capacity value b can be adjusted with a certain cost. The adjustment is represented with s and the value per unit of flexible capacity is c. The parameters can be restricted to the following: $p_j > 0$ and $w_j > 0$, $j \in N$. Moreover, in order to avoid trivial cases it is assumed that $\sum\limits_{j \in N} w_j > b$. The mathematical formulation is as follows.

$$\begin{array}{llll} \textbf{KPC}: & \text{Max} & \sum\limits_{j\in N} & p_j x_j - c \cdot s \\ & \text{s.t.} & \sum\limits_{j\in N} & w_j x_j \leq b + s \\ & & & l \leq s \leq u \\ & & & x_j \in \{0,1\} \qquad \forall j \in N \end{array}$$

This maximization problem can be transformed to the following equivalent minimization problem where if $x_1, ..., x_n$ is an optimal solution for the maximization problem, then the variables $y_j := 1 - x_j$ $(j = 1, ..., n)$ is an optimal solution of the minimization problem.

$$\textbf{KPC'}: \quad \begin{aligned} \text{Min} \quad & \sum_{j \in N} p'_j y_j + s' \\ \text{s.t.} \quad & \sum_{j \in N} w_j y_j + s' \geq -b + \sum_{j \in N} w_j - l \\ & s' \leq u \\ & y_j \in \{0,1\} \quad \forall j \in N \end{aligned}$$

The transformation is done as follows.

$$\begin{aligned} \text{Max} \sum_{j \in N} p_j(1-y_j) - c \cdot s &\equiv \text{Min} \sum_{j \in N} -p_j(1-y_j) + c \cdot s \\ &= \text{Min} \sum_{j \in N} p_j y_j - \sum_{j \in N} p_j + c \cdot s \end{aligned}$$

And minimizing $\sum_{j \in N} p_j y_j - \sum_{j \in N} p_j + c \cdot s$ is equivalent to minimizing $\sum_{j \in N} (\frac{p_j}{c}) y_j + s$. Similarly for the first constraint;

$$\begin{aligned} \sum_{j \in N} \quad w_j(1-y_j) &\leq b + s \\ \sum_{j \in N} \quad -w_j(1-y_j) &\geq -b - s \\ \sum_{j \in N} \quad w_j y_j + s &\geq -b + \sum_{j \in N} w_j \end{aligned}$$

Let $p'_j = (\frac{p_j}{c})$ and let $s' = s - l$ for the second constraint, and we reach the model KPC'.

Now, given an instance of KPC', we have n items with profits $p'_1, p'_2, ..., p'_n$ and weights $w_1, w_2, ..., w_n$. Let $p'_i = b_i$ and $w_i = d_{\rho i}$ $\forall i \in N$. We want to select a set of nodes to maximize the total profit while satisfying a capacity of $b = \sum_{j \in N} w_j - b_\rho - l$. When there is no upper bound on the s', we reach the following model.

$$\begin{aligned} \text{Min} \quad & \sum_{i \in N} b_i y_i + s' \\ \text{s.t.} \quad & \sum_{i \in N} d_{\rho i} y_i + s' \geq b_\rho \\ & s' \geq 0, \quad y_i \in \{0,1\} \qquad \forall i \in N \end{aligned}$$

Solving this problem is equivalent to solving the LCIP on a star network.

3.7 Example of an LCIP solution

We solve the following example using both the dynamic programming approach and the greedy algorithm to illustrate the algorithms and compare the solutions. The network of this example was given in Figure 3.3 and the data on the hurdles and the influence factors are given below.

Table 3.1: Hurdle and influence factor data for the example.

Node i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hurdle b_i	7	5	9	35	8	20	16	14	13	29	11	18	10	13	15
Influence Factor d_i	7	5	9	12	8	10	16	10	13	11	11	18	10	13	15

3.7.1 Dynamic programming algorithm

In this example, we apply the dynamic algorithm by working through the SolveStar, CompressStar and TotalCost algorithms. The numbers in each line represent the corresponding step of that algorithm.

$B[i] = i, P[i] = 0 \qquad \forall i \in V.$

SolveStar

$V = \{1, 2, 3, 4\}, \ L = \{1, 2, 3\}, \ \ \rho = 4.$

1. $\qquad b_4 = 35 - 12 = 23.$

2. $S = \{1, 2, 3\}$.

3. $S = \{2, 1, 3\}$.

4. $|S| = 3 > \lceil \frac{23}{12} \rceil = 2$.

5. $b_2 + b_1 = 5 + 7 = 12 \quad$ vs.

6. $b_2 + (b_4 - (2-1)d_4) = 5 + (23 - 12) = 16$.

   $\min\{12, 16\} = 12$, then $C = 12$.

9. $S = \{3\}$, $S^* = \{2, 1\}$.

CompressStar

1. $b' = \min\{d_4, b_4 + d_4 - |S^*|d_4, d_3\} = \min\{12, 23 + 12 - 24, 9\} = 9$.

2. $B[4] = B[3] = 3$.

3. $d' = b' = b'_4 = d'_4 = 9$.

SolveStar

$V = \{4, 5, 6\},\ L = \{4, 5\},\quad \rho = 6$.

1. $b_6 = 20 - 10 = 10$.

2. $S = \{4, 5\}$.

3. $S = \{5, 4\}$.

4. $|S| = 2 \geq \lceil \frac{10}{10} \rceil = 1$.

5. $b_5 = 8 \quad$ vs.

6. $0 + (b_6 - (1-1)d_6) = 0 + (10 - (1-1)10) = 10$.

   $\min\{8, 10\} = 8$, then $C = 8$.

9. $S = \{4\}$, $S^* = \{5\}$.

CompressStar

1. $b' = \min\{d_6, b_6 + d_6 - |S^*|d_6, d'_4\} = \min\{10, 10 + 10 - 10, 9\} = 9$.

2. $B[6] = B[4] = 3$.

3. $d' = b' = b'_6 = d'_6 = 9$.

SolveStar

$V = \{10, 11, 12, 13, 14, 15\},\ L = \{11, 12, 13, 14, 15\},\quad \rho = 10$.

1. $b_{10} = 29 - 11 = 18$.

2. $S = \{13\}$.

7. $|S| = 1 < \lceil \frac{18}{11} \rceil = 2$.

8. $C = b_{13} + P[10] = b_{13} + (b_{10} - (1)d_{10}) = 10 + (18 - 11) = 17$.

9. $S = \emptyset$, $S^* = \{13\}$.

CompressStar

1. $b' = \min\{d_{10}, b_{10} + d_{10} - |S^*|d_{10}\} = \min\{11, 18 + 11 - 11\} = 11$.

3. $d' = b' = b'_{10} = d'_{10} = 11$.

SolveStar

$V = \{7, 8, 6, 10, 9\},\ L = \{7, 6, 10, 9\},\quad \rho = 8$ *(The last star).*

1. $b_8 = 14$ *(No update is necessary).*

2. $S = \{6\}$.

7. $|S| = 1 < \lceil \frac{14}{10} \rceil = 2$.

8. $C = b_6 + P[8] = b_6 + (b_8 - (1)d_8) = 9 + (14 - 10) = 13$.

9. $S = \emptyset$, $S^* = \{6\}$.

TotalCost

1. $\overline{C} = 12 + 8 + 17 + 13 = 50$.

2. $\overline{S} = \{2, 1, 5, 13, B[6] = 3\}$.

3. Pay nodes 2,1,5,13,3 full incentives in the amount of 5,7,8,10,9.

5. $\overline{P} = \{10, 8\}$.

6. Pay nodes 10,8 partial incentives in the amount of 7,4.

3.7.2 Greedy algorithm

We solve the same problem using the greedy algorithm. Each iteration of the greedy algorithm is shown with a "·" at the beginning of the line.

· $\min\{b_i, d_i\}, i \in V = 5$.

Set of buyers: $\{2\}$.

Update the hurdles: $b_4 = 35 - 12 = 23$.

· $\min\{b_i, d_i\}, i \in V - \{2\} = 7$.

Set of buyers: $\{2, 1\}$.

Update the hurdles: $b_4 = 23 - 12 = 11$.

· $\min\{b_i, d_i\}, i \in V - \{2, 1\} = 8$.

Set of buyers: $\{2, 1, 5\}$.

Update the hurdles: $b_6 = 20 - 10 = 10$.

· $\min\{b_i, d_i\}, i \in V - \{2, 1, 5\} = 9$.

Set of buyers: $\{2, 1, 5, 3\}$.

Update the hurdles: $b_4 = 0$.

Set of buyers: $\{2, 1, 5, 3, 4\}$.

Update the hurdles: $b_6 = 0$.

Set of buyers: $\{2, 1, 5, 3, 4, 6\}$.

Update the hurdles: $b_8 = 14 - 10 = 4$.

· $\min\{b_i, d_i\}, i \in V - \{2, 1, 5, 3, 4, 6\} = 4.$

Set of buyers: $\{2, 1, 5, 3, 4, 6, 8\}$.

Update the hurdles: $b_7 = b_9 = 0, b_{10} = 29 - 11 = 18.$

Set of buyers: $\{2, 1, 5, 3, 4, 6, 8, 7, 9\}$.

· $\min\{b_i, d_i\}, i \in V - \{2, 1, 5, 3, 4, 6, 8, 7, 9\} = 10.$

Set of buyers: $\{2, 1, 5, 3, 4, 6, 8, 7, 9, 13\}$.

Update the hurdles: $b_{10} = 18 - 11 = 7.$

· $\min\{b_i, d_i\}, i \in V - \{2, 1, 5, 3, 4, 6, 8, 7, 9, 13\} = 7.$

Set of buyers:$\{2, 1, 5, 3, 4, 6, 8, 7, 9, 13, 10\}$.

Update the hurdles: $b_{11} = b_{12} = b_{14} = b_{15} = 0.$

Set of buyers: V.

Total Cost $= 5 + 7 + 8 + 9 + 4 + 10 + 7 = 50.$

Set of nodes with partial or full incentives:$\{2, 1, 5, 3, 8, 13, 10\}$.

Both solution methods give the same total cost of 50, with the same set of nodes that receive incentive, though the greedy algorithm is much easier to apply.

3.8 Concluding Remarks

In this chapter, we investigated the LCIP model independent of the product design problem by proposing a model for general networks. As this model is NP-Hard, we identified a polynomially solvable case, which is the problem on a tree network with equal influences from one's neighbors. We proposed a dynamic programming approach as the solution method. This solution approach concentrates on the star subnetworks of a network and first solves the problem on these subnetworks. We also show that this algorithm reduces to a greedy algorithm.

The LCIP is a combinatorial optimization problems that can be used in a variety of environments, from marketing to epidemiology. It is an interesting problem in diffusion of information as it offers a rich set of problems for future research. One of such problems is the LCIP under a stochastic influence structure, rather than a discrete and linear one used in this thesis. Another direction of problems can include the time dependent social network, i.e., one which changes over time as the connections among nodes break or new connections are built. Understanding the structural relationship (depending on the location of the node over a network) of the optimal set of nodes is another direction for targeting. Exploring both exact algorithms and heuristics to expand the solution approaches for these problems is a future research direction for the LCIP.

Chapter 4

The Product Line Design Problem with Social Network Effects

4.1 Introduction and Literature Review

Product line design is an important problem companies face when they want to create a selection of products to appeal to heterogenous consumer segments in the market. Such products may be manufactured goods as well as a service good such as a cell phone plan. It is a very well studied subject in the marketing literature. In this chapter we focus more on the optimization approaches and therefore refer the reader to Belloni et al. (2008) for a detailed description and comparison of methods in the literature for product line optimization in general. More recently, work by Wang et al. (2009) proposed a branch-and-price algorithm, Luo (2011) provided an integrated marketing and engineering approach, and Tsafarakis et al. (2011) used a particle swarm optimization approach to solve the product line design problem. Although we solve a product line design problem, our problem is quite different from the existing literature as, to our knowledge, this is the first study to incorporate social network effects in product line design.

As mentioned in Chapter 1, the product line design problem with social network effects is more complicated than the single product design problem since the peer influence effects among users of different products of the same product line also have a role in the diffusion of products. We model this situation using two dimensions as first and second order peer influence effects. The first order effects are the peer influences from consumers of the same segment, i.e., using the same product, and the second order effects are the peer influence among all product users across the product line.

In this model, the utility one gets from a product is dynamic under peer influence, i.e., increases with the number of neighbors who also bought the same or a different product (as the decrease in hurdle is mathematically equivalent to an increase in utility). This fluctuation (change) in the utilities due to peer influence, before a purchasing decision is made, prevents one from making a list of preference ordering among the products before solving the product line design problem. (Note that the same is also true for the single product model.) The model by Belloni et al. (2008) does not take into account such peer influence effects and therefore can determine an a priori preference ordering among the different product profiles which simplifies their model. It should be clear that it cannot be used to model our setting. Wang et al. (2009) also modeled the product line design for the share-of-choice problem without requiring an a priori preference ordering among the product profiles for the consumers, however, their model does not require the identification of the highest utility product for each person. Determining the highest utility product is important in the setting where peer influences vary across products (as the peer influence is related specifically to the product chosen/adopted) and is more realistic in terms of the customer preference. Thus, we significantly expand upon the Wang et al. (2009) model by incorporating the peer influence effects and ensuring the customer picks the product with the highest utility.[1]

[1]We should note that the branch-and-price approach in Wang et al. (2009) relies on the procedure developed in Camm et al. (2006) to solve the pricing problem. As explained in Section 2.3 that approach cannot be applied in the setting with peer influence effects.

4.2 Integer Program for the Product Line Design with Network Effects

To include the first order network effects that a person is positively affected from a neighbor who uses the same product (as the influence model, Model 1, in Section 2.1) we construct the following integer program. Let M be the number of products in the product line and purchase among multiple products be represented by the additional decision variable type y_{sq} which is 1 if person s buys the q-th product in the product line. x_{klq} is 1 if level l is selected for attribute k of the q-th product and is 0, otherwise. The share-of-choice problem with network effects for the product line design problem (SOCNEPL) is given below.

SOCNEPL: Maximize
$$\sum_{s=1}^{n} y_s, \tag{4.1}$$
subject to
$$\sum_{k=1}^{K}\sum_{l=1}^{L_k} u_{kl}^{s} x_{klq} + \Delta_s^1 \sum_{j\in V} a_{js} y_{jq} = U_{sq} \tag{4.2}$$
$$h_s^H y_{sq} \leq U_{sq} \tag{4.3}$$
$$s = 1,2,\ldots,n, \quad q = 1,2,\ldots,M,$$
$$\sum_{l=1}^{L_k} x_{klq} = 1 \tag{4.4}$$
$$k = 1,2,\ldots,K, \quad q = 1,2,\ldots,M,$$
$$\sum_{q=1}^{M} y_{sq} = y_s \quad s = 1,2,\ldots,n, \tag{4.5}$$

(The integer program continues on the next page.)

$$\left.\begin{aligned} U_{sq} - U_{sr} &\le B(1 - c^s_{qr}) \\ U_{sr} - U_{sq} &\le B c^s_{qr} \\ y_{sq} - y_{sr} &\le (1 - c^s_{qr}) \\ y_{sr} - y_{sq} &\le c^s_{qr} \\ c^s_{qr} &\ \text{binary} \end{aligned}\right\} \begin{array}{l} s = 1, 2, \ldots, n, \\ \forall\, q, r \text{ pairs} \in \{1, 2, \ldots, M\}, \end{array} \tag{4.6}$$

$$\left.\begin{aligned} y_s, y_{sq}, x_{klq} &\ \text{binary} \\ U_{sq} &\ge 0 \end{aligned}\right\} \begin{array}{l} s = 1, 2, \ldots, n,\ l = 1, 2, \ldots, L_k, \\ k = 1, 2, \ldots, K, q = 1, 2, \ldots, M. \end{array} \tag{4.7}$$

The objective of the problem maximizes the total number of buyers. Constraints (4.2) and (4.3) correspond to constraint (2.1) in the SOCSNE model and include only the first-order network effects. One level is selected for each attribute with constraint (4.4) and at most one product is allowed to be purchased by a customer with constraint (4.5), represented by the binary variable y_s which is 1 if person s buys a product and 0, otherwise. As mentioned above, we need to ensure a customer purchases the product that provides them the highest utility after taking social network effects into account. This is somewhat tricky to model and the constraint group (4.6) guarantees that if the utility from the q-th product is greater than the utility from the r-th product for a customer then the q-th product is preferred over the r-th product by that person. Here B is a positive large integer number, though it is easily seen that it is bounded by the total number of levels and c^s_{qr} is a binary variable required for the logical argument.

To differentiate the two dimensional network effects, we introduce a second order effect, Δ_s^2, which includes network effects from buyers of products in the product line other than the one that person owns and is of a smaller magnitude than the first order effect, Δ_s^1. So the utility person s gets from using product q represented by U_{sq} in constraint (4.2) would now be replaced with the following (Δ_s^2 is subtracted from Δ_s^1 to eliminate double counting of the effect from the same consumer).

$$\sum_{k=1}^{K}\sum_{l=1}^{L_k} u_{kl}^s x_{klq} + (\Delta_s^1 - \Delta_s^2)\sum_{j\in V} a_{js}y_{jq} + \Delta_s^2 \sum_{j\in V} a_{js}y_j = U_{sq}. \tag{4.8}$$

From here on, when we discuss the integer program for the product line design with network effects, we refer to the model with constraint (4.8), i.e., including both first and second order effects, instead of constraint (4.2).

4.3 Genetic Algorithm for Product Line Design with Network Effects

The SOCNEPL model for the product line design problem with network effects is not computationally tractable as the LP relaxation is weak. Genetic algorithms have been used previously in the literature to solve the product line design problem without the network effects and shown to have significant potential to solve product line design problems (Alexouda and Paparrizos (2001), Steiner and Hruschka (2003), Balakrishnan et al. (2004), Fruchter et al. (2006)). Genetic algorithms have also been used in practice for a product line design problem as part of an optimization study by Intel (Kempf and Rash, 2011) that also won the Daniel H. Wagner Prize Competition. We will follow the natural extension of our genetic algorithm for the single product design for the product line design problem. We follow the same steps; Generation of an initial population, Evaluation, Crossover, Mutation and Reduction. The GA requires a few but major modifications to adapt it to the product line model.

First, in the product line problem, the representation of an individual in the population will need to include the different products in a product line. Here we expand our approach in the single case by simply placing product profiles next to each other in an individual representation. Note that with this approach there are more than one representation of the same product line, which can be obtained by different permutations of products in the product line. To overcome any problems that we may face due to this ordering at the crossover and mutation stages, we adapt these stages taking this into account as we explain next.

An initial population is randomly generated. Fitness of a product line is the market share of the product line and is calculated using the new integer program, SOCNEPL which includes the first and second order effects. We do not necessarily prevent the same product from appearing more than once in a representation. In this setting, increasing the number of different products offered in a product line cannot worsen the solution (on the other hand, there may be new buyers for the additional product). So we know that solutions which contain the same product more than once will be eliminated through generations in the genetic algorithm because of their fitness values and are unlikely to appear in the final solution. In the crossover stage, we use a single-point crossover, however the solutions with the single point crossover as it is used in the single product case would be dependent on the ordering of the products in the product line representation. To avoid this, we repeat the crossover for each product in the representation, keeping the point of crossover random for each product. Similarly, in the mutation step a level is changed with a predetermined probability for each product in the product line. At the end of the mutation step, when the population size is doubled, it is halved by selecting the product lines which have the largest total fitnesses.

4.4 Computational Results for Product Line Design with Network Effects

In this section, we describe our experiments on simulated data with our genetic algorithm. We start with the data we used in Chapter 2. However, the market shares with a single product are already high with this data and as we increase the number of products, the market share quickly matches the size of the network, preventing us from observing the effects of adding more products to the product line. To overcome this, we generate a second set of data for the product line design problem with network effects where we try to keep the market share below a certain percentage of the market size when a single product is offered.

4.4.1 Data Generation

The data simulated in Chapter 2 does not include second order effects, so we generate Δ_s^2 for each node s. We generate Δ_s^2 proportional to Δ_s^1 for each node, as we expect it to be more likely for a person who is more easily influenced by the first order effects to be influenced from the second order effects. We keep the secondary effects less than the first order effects and more specifically we select them to be within the interval of 33.3% and 66.6% of the first order effects. To keep a rather heterogeneous structure, the second order effects are randomly generated from this interval.

For the new set of simulations in this chapter, utilities are generated uniformly between 0 and 1 and normalized for each person. In order to reach a market solution that can clearly demonstrate the difference in market shares for different number of products, we experimented with different hurdles and peer effects. To determine the hurdles, 1000 products are randomly generated and ranked for each person (in descending order of their utilities) and their high hurdle is set as the utility value of a product profile selected randomly from the top 5% of this ranking. This product profile is assumed to represent a status-quo product in the market. Then the lower hurdle is selected randomly, within the range determined by the corresponding high hurdle. We set the lower bound of this range for the low hurdle as 60% of the high hurdle and the upper bound to be the same as the high hurdle. This structure allows a person to have the same high and low hurdle with a small probability. In other words, some individuals in the social network have a lot of positive utility when their neighbors use the product and others have less. The second part of the simulation includes the random generation of the first and second order effects. The first order effects are calculated as the ratio of the difference between high and low hurdles and the degree of a node, as in the single product case. We want the second order effects to be less than the first order effects and we set them to be within 50%-70% (chosen randomly) of the first order effects.

We kept the same parameters in the GA for the product line design. However, as most steps are repeated for each product in a product line, the GA running times for the product line are noticeably longer than single product cases. Therefore we limit the population size of the algorithm to 50 (instead of 100).

Alexouda and Paparrizos (2001) and Steiner and Hruschka (2003) used 2 to 4 products for their computational studies, Balakrishnan et al. (2004) used 4 to 7 products, and Belloni et al. (2008) used 3 to 5 products as the size of the product line. For our study, taking into account the running times of the integer program (as the problem gets larger with network effects), we use 2 to 5 products (which is comparable in size to the state-of-the-art work by Belloni et al. (2008)).

4.4.2 Performance of the GA

We start our computational experiments using the data from Chapter 2 and show with one of the data sets, with $p = 0.2$ and 100 nodes, that the market share reaches the whole population quickly as the number of products is increased. In Table 4.1, we show the market shares obtained with the GA for product lines with different number of products. In these examples, by providing 2 different products we can reach almost all the population, and when a third or a fourth product is added to the product line, we reach the whole market where each individual can find an appealing product design for themselves. Market shares for product lines with 4 products, as shown in Column 6, are expected to be 100 when the results for 3 products are 100. Increasing the number of products at this point (once the whole population is reached) cannot change (increase or decrease) the market share, albeit the product a person buys may be different in the two cases.

We next present, in Table 4.2, the GA solutions with the new simulated data (for $p = 0.2$) and compare it to the integer programming (SOCNEPL) solutions. Looking at the computational time to reach these solutions, the average running time is 150 seconds

Table 4.1: GA market shares for product lines of different sizes (with data in Chapter 2).

Num. of	Num.of	100 people			
attributes	levels	1 product	2 products	3 products	4 products
3	2	68	100	100	100
3	3	73	98	100	100
3	4	81	97	100	100
3	5	74	96	100	100
4	2	72	100	100	100
4	3	73	96	100	100
4	4	77	98	100	100
4	5	72	97	100	100
5	2	70	100	100	100
5	3	82	99	100	100
5	4	79	99	100	100
5	5	82	99	100	100
6	2	78	100	100	100
6	3	76	98	100	100
6	4	81	99	100	100
6	5	80	97	100	100

for the 100 people network and 291 seconds for the 200 people network for the GA. As the size of the integer program is large, the time to calculate the exact integer programming solution is huge (renders the computer out of memory in 6 out of 28 cases) as well. For the 22 cases where we could compare the GA solution with the exact IP solution, we hit the optimal solution in 17 cases. When the optimal is missed, the GA solution is %96.34 of the optimal solution, on average. In tables 4.4, 4.5 and 4.6, we give the extensive GA market share results for different networks and different sizes of product lines.

Looking at the market shares under different product line sizes, we see that the market shares increase as product line is expanded with more products, as expected. We want to explore if there is a significant difference in how the market share changes within networks formed by different rewiring probabilities, $p = 0.1, p = 0.2, p = 0.3$. Note that as p gets closer to 1, randomness of the graph increases. In Table 4.3 we observe the

Table 4.2: Comparison of market shares for the GA and the SOCSNEPL (2,3 products).

		2 products				3 products			
		100 people		200 people		100 people		200 people	
Num. of attributes	Num.of levels	GA	SOCNEPL	GA	SOCNEPL	GA	SOCNEPL	GA	SOCNEPL
3	2	55	55	106	106	75	75	147	147
3	3	31	31	40	41	47	47	65	68
3	4	25	25	41	42	39	39	61	o.o.m.
3	5	25	25	37	38	34	o.o.m.	55	o.o.m.
4	2	41	41	64	64	61	61	100	100
4	3	28	28	50	50	42	42	76	76
4	4	29	31	42	o.o.m.	38	o.o.m.	60	o.o.m.

average (over all number of attributes and number of levels) market shares in each network in columns 2, 3 and 4, and the percentage increases in the next two columns. According to this table, for the 200 people network with $p = 0.1$ the market share has more than doubled (25 to 52.75) when a second product is introduced and it is further increased by another almost 48% with the addition of the third product (52.75 to 77.88). However, the patterns of increase in the market share across differently generated networks are very similar; more than doubled with the introduction of the second product, and with the introduction of the third product the increase is less but still almost three times the market share obtained with a single product.

Table 4.3: Average market shares and % increases for product lines of sizes 1, 2 and 3, respectively.

	Num. of people	Avg m.s.	Avg m.s.	Avg m.s.	% increase	% increase
p=0.1	100	14.50	31.06	42.94	214.22	138.23
	200	25.00	52.75	77.88	211.00	147.63
	300	34.63	71.31	105.38	205.96	147.77
p=0.2	100	15.56	32.31	46.25	207.63	143.13
	200	24.94	50.81	75.25	203.76	148.09
	300	33.19	73.06	105.75	220.15	144.74
p=0.3	100	16.00	32.81	45.38	205.08	138.29
	200	25.38	51.00	77.31	200.99	151.59
	300	33.13	71.75	104.56	216.60	145.73

When we look at the optimal product designs at product lines of different sizes, we see that a product profile optimal in a product line with q products is not necessarily optimal in a product line with $q + n$ products, $q, n \in Z^+$. Table 4.7 shows an example of product profiles for one of those data sets ($p = 0.2$, 100 people, 3 attributes with 3 levels). The first row corresponds to the single product case. At every row, the number of products in a product line are increased by 1. The first column shows the market share, and the following columns show the optimal product designs within these product lines. For example, the optimal design for the problem with a single product is 3-1-2, i.e., level 3 is selected for attribute 1, level 1 is selected for attribute 2, and level 2 is selected for attribute 3. If the number of products in this product line are increased by 1, then the product design 1-2-3 is included in the optimal solution. Product design 3-1-2 is carried onto the solutions of the next four cases, but is absent in the product line with 5 products. Similarly, product design 1-2-3 appears only in the product line with 2 products. In the product line design problem with network effects, we aim to select the best combination of product profiles, taking into account the interactions between customers.

4.4.3 Network Effects

Since we are using an integer problem within the evaluation process of the genetic algorithm for the product line design problem with network effects, we encounter the same incentives problem as in the single product case (see Section 2.2 for a comprehensive explanation) and we want to minimize the total amount of incentives given. In this section, we look at the market share results of the genetic algorithm with and without the

Table 4.4: GA market shares for product lines of different sizes, for $p = 0.1$.

	Num. of attributes	Num.of levels	1 prod.	2 prod.	3 prod.
100 people	3	2	27	52	74
	3	3	13	29	43
	3	4	10	28	40
	3	5	8	20	32
	4	2	21	42	58
	4	3	13	30	44
	4	4	13	29	34
	4	5	11	22	34
	5	2	13	30	47
	5	3	13	28	38
	5	4	12	27	36
	5	5	13	29	36
	6	2	15	33	45
	6	3	18	32	44
	6	4	16	33	43
	6	5	16	33	39
200 people	3	2	42	96	131
	3	3	30	54	78
	3	4	19	41	64
	3	5	18	39	59
	4	2	33	73	112
	4	3	21	40	69
	4	4	18	37	60
	4	5	18	37	56
	5	2	29	65	93
	5	3	23	49	74
	5	4	26	55	74
	5	5	23	48	69
	6	2	24	63	93
	6	3	22	47	73
	6	4	27	53	75
	6	5	27	47	66
300 people	3	2	70	137	209
	3	3	30	67	104
	3	4	29	61	92
	3	5	25	53	76
	4	2	36	86	135
	4	3	31	65	96
	4	4	31	58	88
	4	5	21	51	77
	5	2	39	87	130
	5	3	37	73	100
	5	4	36	65	82
	5	5	25	45	68
	6	2	35	85	130
	6	3	40	80	113
	6	4	36	71	104
	6	5	33	57	82

Table 4.5: GA market shares for product lines of different sizes, for $p = 0.2$.

	Num. of attributes	Num.of levels	1 prod.	2 prod.	3 prod.	4 prod.	5 prod.
100 people	3	2	25	55	75	87	93
	3	3	15	31	47	58	68
	3	4	12	25	39	50	56
	3	5	12	25	34	42	48
	4	2	16	41	61	72	80
	4	3	13	28	42	47	53
	4	4	15	29	38	45	60
	4	5	12	22	34	43	50
	5	2	15	32	50	64	75
	5	3	11	29	41	60	70
	5	4	19	32	46	56	66
	5	5	15	29	38	47	53
	6	2	24	43	66	72	76
	6	3	18	39	45	55	70
	6	4	15	32	43	51	62
	6	5	12	25	41	49	51
200 people	3	2	51	106	147	173	182
	3	3	18	40	65	90	106
	3	4	19	41	61	84	103
	3	5	19	37	55	75	81
	4	2	34	64	100	127	146
	4	3	23	50	76	100	102
	4	4	19	42	60	78	93
	4	5	19	37	58	68	80
	5	2	31	62	94	127	142
	5	3	25	44	64	87	101
	5	4	23	47	70	83	91
	5	5	20	46	60	70	87
	6	2	27	55	96	120	141
	6	3	23	45	75	99	107
	6	4	25	51	62	89	106
	6	5	23	46	61	83	93
300 people	3	2	61	140	191	240	262
	3	3	32	69	107	134	168
	3	4	22	51	81	116	140
	3	5	28	61	82	105	128
	4	2	49	109	151	189	216
	4	3	33	73	108	130	154
	4	4	25	52	79	111	127
	4	5	28	51	86	90	115
	5	2	35	81	130	177	195
	5	3	32	73	98	128	152
	5	4	29	63	98	112	135
	5	5	30	65	83	115	141
	6	2	29	70	107	151	168
	6	3	39	84	106	138	164
	6	4	28	63	96	105	146
	6	5	31	64	89	116	140

Table 4.6: GA market shares for product lines of different sizes, for $p = 0.3$.

	Num. of attributes	Num.of levels	1 prod.	2 prod.	3 prod.
100 people	3	2	26	54	75
	3	3	16	31	46
	3	4	13	26	35
	3	5	10	24	33
	4	2	17	41	60
	4	3	14	29	40
	4	4	13	25	36
	4	5	13	23	31
	5	2	16	37	51
	5	3	11	31	51
	5	4	19	34	45
	5	5	15	32	38
	6	2	23	44	60
	6	3	18	34	45
	6	4	17	33	47
	6	5	15	27	33
200 people	3	2	42	85	131
	3	3	19	45	71
	3	4	22	40	62
	3	5	17	37	65
	4	2	39	71	110
	4	3	23	49	75
	4	4	18	39	64
	4	5	20	38	57
	5	2	30	64	97
	5	3	33	58	80
	5	4	20	44	68
	5	5	19	48	65
	6	2	27	56	85
	6	3	25	49	81
	6	4	25	52	67
	6	5	27	41	59
300 people	3	2	61	129	192
	3	3	35	71	112
	3	4	27	57	84
	3	5	26	55	84
	4	2	44	88	140
	4	3	35	74	104
	4	4	32	62	84
	4	5	24	49	74
	5	2	29	86	123
	5	3	28	77	108
	5	4	25	56	82
	5	5	28	60	79
	6	2	29	73	110
	6	3	32	69	101
	6	4	41	80	104
	6	5	34	62	92

Table 4.7: Optimal product profiles for product lines of different sizes.

Market Share	Product 1	Product 2	Product 3	Product 4	Product 5
15	3-1-2				
31	3-1-2	1-2-3			
47	2-2-3	3-1-2	2-2-2		
58	2-2-3	3-1-2	1-2-1	1-3-2	
68	2-2-3	1-2-2	1-1-1	2-3-2	3-3-3

use of incentives. In Table 4.8, 4.9 and 4.10, we compare the market shares when these incentives are given vs. not given. Here GA+NE (GA with Network Effects) columns show what the market shares are when the products designed with the GA is launched in a market and the product is allowed to diffuse over the network in a natural way (through influence over people's connections), i.e., no outside incentives are given to the people. In the adjacent columns, we have GA+IN (IN=incentives) which show the market share when the products designed with the GA is launched in the market, and further interventions (coupons, free samples) are made by the seller to maximize the spread. By comparing columns 5 and 6 (GA+IN for 2 products vs. GA+NE for 3 products), these three tables also allow us to observe that in some cases we can reach a larger market share by providing incentives instead of increasing the number of products in a product line. Here, an analysis should be performed though, among the cost of introducing another product into the line and the amount of resources needed to provide incentives.

Table 4.8: Comparison of market shares with and without incentives, for $p = 0.1$.

	Num. of attributes	Num.of levels	2 prod. GA+NE	2 prod. GA+IN	3 prod. GA+NE	3 prod. GA+IN
100 people	3	2	50	52	71	74
	3	3	16	29	37	43
	3	4	22	28	30	40
	3	5	18	20	21	32
	4	2	24	42	36	58
	4	3	20	30	26	44
	4	4	21	29	20	34
	4	5	22	22	24	34
	5	2	12	30	32	47
	5	3	17	28	21	38
	5	4	17	27	21	36
	5	5	14	29	26	36
	6	2	8	33	23	45
	6	3	18	32	23	44
	6	4	18	33	35	43
	6	5	33	33	33	39
	Average		20.63	31.06	29.94	42.94
200 people	3	2	79	96	122	131
	3	3	36	54	56	78
	3	4	31	41	45	64
	3	5	34	39	52	59
	4	2	44	73	78	112
	4	3	25	40	56	69
	4	4	26	37	42	60
	4	5	21	37	41	56
	5	2	40	65	61	93
	5	3	31	49	53	74
	5	4	42	55	43	74
	5	5	25	48	42	69
	6	2	25	63	48	93
	6	3	32	47	51	73
	6	4	30	53	55	75
	6	5	29	47	57	66
	Average		34.38	52.75	56.38	77.88
300 people	3	2	110	137	194	209
	3	3	41	67	64	104
	3	4	59	61	78	92
	3	5	38	53	60	76
	4	2	57	86	112	135
	4	3	45	65	70	96
	4	4	56	58	64	88
	4	5	31	51	63	77
	5	2	44	87	88	130
	5	3	40	73	65	100
	5	4	50	65	51	82
	5	5	35	45	49	68
	6	2	39	85	57	130
	6	3	52	80	76	113
	6	4	42	71	67	104
	6	5	44	57	67	82
	Average		48.94	71.31	76.56	105.38

Table 4.9: Comparison of market shares with and without incentives, for $p = 0.2$.

	Num. of attributes	Num.of levels	2 prod. GA+NE	2 prod. GA+IN	3 prod. GA+NE	3 prod. GA+IN	4 prod. GA+NE	4 prod. GA+IN	5 prod. GA+NE	5 prod. GA+IN
100 people	3	2	36	55	64	75	79	87	93	93
	3	3	27	31	40	47	51	58	61	68
	3	4	22	25	32	39	39	50	53	56
	3	5	14	25	27	34	31	42	37	48
	4	2	25	41	59	61	66	72	76	80
	4	3	26	28	37	42	43	47	49	53
	4	4	22	29	33	38	40	45	42	60
	4	5	13	22	21	34	30	43	37	50
	5	2	16	32	19	50	30	64	32	75
	5	3	14	29	20	41	30	60	48	70
	5	4	22	32	19	46	46	56	47	66
	5	5	22	29	24	38	33	47	37	53
	6	2	11	43	41	66	40	72	60	76
	6	3	22	39	23	45	38	55	47	70
	6	4	20	32	32	43	36	51	43	62
	6	5	11	25	22	41	32	49	32	51
	Average		20.19	32.31	32.06	46.25	56.13	64.44	49.63	64.44
200 people	3	2	96	106	137	147	173	173	182	182
	3	3	30	40	50	65	78	90	90	106
	3	4	30	41	36	61	68	84	68	103
	3	5	25	37	48	55	55	75	65	81
	4	2	34	64	74	100	95	127	130	146
	4	3	45	50	48	76	67	100	79	102
	4	4	26	42	46	60	57	78	66	93
	4	5	30	37	37	58	52	68	52	80
	5	2	41	62	55	94	93	127	115	142
	5	3	31	44	46	64	62	87	76	101
	5	4	33	47	53	70	66	83	80	91
	5	5	32	46	53	60	48	70	66	87
	6	2	25	55	51	96	95	120	83	141
	6	3	27	45	50	75	75	99	58	107
	6	4	27	51	27	62	57	89	89	106
	6	5	29	46	46	61	59	83	65	93
	Average		35.06	50.81	53.56	75.25	75.00	97.06	85.25	110.06
300 people	3	2	125	140	176	191	229	240	260	262
	3	3	62	69	92	107	120	134	134	168
	3	4	45	51	63	81	95	116	105	140
	3	5	52	61	68	82	84	105	124	128
	4	2	73	109	115	151	159	189	172	216
	4	3	49	73	73	108	87	130	120	154
	4	4	41	52	61	79	83	111	105	127
	4	5	39	51	64	86	62	90	91	115
	5	2	59	81	88	130	111	177	130	195
	5	3	47	73	66	98	96	128	111	152
	5	4	46	63	83	98	72	112	93	135
	5	5	59	65	65	83	80	115	109	141
	6	2	31	70	64	107	90	151	115	168
	6	3	56	84	78	106	106	138	125	164
	6	4	45	63	63	96	81	105	94	146
	6	5	52	64	58	89	77	116	90	140
	Average		55.06	73.06	79.81	105.75	102.00	134.81	123.63	159.44

Table 4.10: Comparison of market shares with and without incentives, for $p = 0.3$.

	Num. of attributes	Num.of levels	2 products GA+NE	2 products GA+IN	3 products GA+NE	3 products GA+IN
100 people	3	2	40	54	69	75
	3	3	24	31	37	46
	3	4	26	26	31	35
	3	5	13	24	17	33
	4	2	37	41	55	60
	4	3	17	29	36	40
	4	4	18	25	30	36
	4	5	19	23	29	31
	5	2	22	37	36	51
	5	3	18	31	28	51
	5	4	24	34	28	45
	5	5	15	32	28	38
	6	2	13	44	31	60
	6	3	16	34	40	45
	6	4	20	33	30	47
	6	5	20	27	16	33
	Average		21.38	32.81	33.81	45.38
200 people	3	2	81	85	111	131
	3	3	34	45	47	71
	3	4	34	40	50	62
	3	5	29	37	40	65
	4	2	50	71	88	110
	4	3	35	49	59	75
	4	4	37	39	42	64
	4	5	31	38	39	57
	5	2	32	64	60	97
	5	3	42	58	56	80
	5	4	37	44	45	68
	5	5	36	48	52	65
	6	2	18	56	62	85
	6	3	43	49	62	81
	6	4	44	52	55	67
	6	5	31	41	47	59
	Average		38.38	51.00	57.19	77.31
300 people	3	2	111	129	165	192
	3	3	54	71	77	112
	3	4	45	57	64	84
	3	5	42	55	52	84
	4	2	61	88	113	140
	4	3	55	74	81	104
	4	4	42	62	61	84
	4	5	27	49	57	74
	5	2	54	86	47	123
	5	3	48	77	68	108
	5	4	36	56	67	82
	5	5	45	60	63	79
	6	2	35	73	56	110
	6	3	43	69	67	101
	6	4	58	80	81	104
	6	5	39	62	62	92
	Average		49.69	71.75	73.81	104.56

4.5 LCIP for the Product Line Design with Network Effects

4.5.1 Model

In the single product case, in order to identify the set of nodes to give incentives to, we zoom into the diffusion process and examine it one period at a time. The incentives are given at points of time where natural diffusion of the product over the network cannot expand anymore because there are no nodes whose utility is greater than their hurdles. In this setting, incentives allow for more number of steps of diffusion to take place until another knot is reached where more incentives are provided. This process is repeated until all the market share (obtained with GA) is reached.

The model for the multiple products is more complicated than the single product model for two reasons. First, people select the product with the highest utility. Secondly, although there is a single hurdle for each node, utilities from each product increase at different rates as time advances, depending on the decisions of other buyer nodes. Therefore the product with the highest utility may change for a node throughout this process.[2] If we follow the same approach as in the single product model with multiple products, we might assign the wrong product to a node in the early stages of the diffusion leading us to find a suboptimal solution for the LCIP. Determining the product profile with the highest utility requires the whole process to be completed and therefore we cannot simply go one time period at a time to determine the buyers and the corresponding products. However,

[2]e.g., Let utilities from product 1 and 2 be 10 and 5, respectively and first and second order effects be 4 and 2, respectively for node s. Assume both products have utilities higher than the hurdle and this node is connected to 10 people who are not buyers yet. At this time, the person would select product 1. In the next period, if all 10 neighbors buy the second product (by the effects of their other neighbors), then the utility of the products for s would become 30 and 45, suggesting the node would select product 2.

since the diffusion problem is solved after the design problem, *we know* which product each person buys before attempting to solve the diffusion problem. This puts us in an advantageous position to solve the LCIP. Once we know the product for each person, the model is simply equivalent to the single product model, though it requires a few initial adjustments to the data. We explain this preprocessing stage next.

Preprocessing: The goal of this stage is to reduce the problem at hand to a single product problem and to use the LCIP integer programming model from Chapter 2 to minimize the incentives given in the product line design problem. (Note that the product will not be the same for each person.) To do this, let's remember the types of variables and data we have in LCIP (Section 2.2). The two types of variables are the amount of incentives given to a node and the binary variable which shows at which time interval a node buys the product. We will keep these same variables here. In the single product case, the objective was the same (to minimize the total amount of incentives given) and the incentives were specified to the nodes. Here, the incentives for the product line design problem are specified with respect to the product profile as well as the node itself. The four types of data we use are 1) Utility from the product (U_s), 2) Hurdle (h_s^H), 3) Adjacency matrix (a_{js}), and 4) Influence factors (Δ_s^1 and Δ_s^2), for each node s.

In the LCIP (in both single product and product line problems), the network only includes the buyers. In the preprocessing stage, the first step is to identify these buyers and classify them according to the products they buy. This allows us to calculate the utility for person s, U_s, from that product without any influence or incentives from outside. The hurdles are distinct for each person and independent of the product profiles, so no modification is required for hurdles. The nodes which are not buyers and the edges connected to these nodes are eliminated from the adjacency matrix. In the product line problem we have two orders of network effects; first order from the users of the same product, and second order from users of any product. Since all nodes in the LCIP are buyers, we have the second order effects between every node which are connected. For the first order effects, we use the sets we have identified at the beginning of the preprocessing step (buyers of the same product) and consider the first order effects between two nodes if and only if they belong to the same set. After this simple preprocessing stage, we can now use the LCIP model in Section 2.2 by replacing constraint (2.9) with the following constraint.

$$U_s \geq h_s^H y_{st} - z_s - (\Delta_s^1 - \Delta_s^2) \sum_{j \in \overline{V}} a_{js} y_{jt} - \Delta_s^2 \sum_{j \in V'} a_{js} y_j \tag{4.9}$$

$$\text{where} \qquad \overline{V} = \{j | Q(j) = Q(s)\}$$

$$\text{and} \qquad Q(j) = q \Leftrightarrow \text{node } j \text{ buys product } q.$$

To reduce the size of the model, as in the solution method of the single product design, we eliminate the nodes that already buy that specified product without incentives before trying to solve the LCIP for the product line design. Next we use the same iterative approach we have introduced in Section 2.5 and solve the problem until no further improvement is obtained in the objective function. We know from Observation 5 in Chapter 2 that this is the optimal solution.

4.5.2 Computational results

We show the LCIP results for the product line design problem for the nine different networks in tables 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19 for product lines with 2 and 3 products. In all of these tables, all rows correspond to one problem set. The number of new buyers in columns 3 and 7 show the network sizes of the LCIP. These are the nodes that were not buyers until at least one node was given incentives. The columns 4 and 8 show the size of the set of nodes that receive incentives. For example, in Table 4.11, for the problem with 2 products which have 6 attributes and 2 levels for each attribute, out of 100 people 10 people are given incentives and in response to this the market share is increased by 25 nodes. The return for each problem set is then calculated as the ratio of the additional number of nodes which become buyers without receiving incentives and the number of nodes that receive incentives; it measures how many additional buyers are reached for each node that receives incentives. Over the nine networks, we calculate the minimum return as 0.25 (where 8 out of 10 people have to be given incentives in the LCIP) and the largest return as 15.50 (where 4 out of 66 people have to be given incentives in the LCIP). The average return over nine networks is 1.32; for each person given incentives, on average at least one additional person buys the product with that influence (i.e., with no incentives). In tables 4.20, 4.21 and 4.22, we look at the solutions of the LCIP for product lines with 4 and 5 products.

Within these tables (Table 4.11-4.22), there are some problem sets where the integer program for the LCIP could not be solved to optimality due to their size, leaving the computer out of memory after a certain period of calculations. We present these cases with a star (*) next to the corresponding number of new buyers in columns 3 and 7, and we show the best (close to optimal) solution we have for these cases. Observing the number of new buyers and the total number of people receiving incentives in each problem set, it is difficult to argue that the ratios of these values to the market share or the market size change in a certain direction as the size of the product line is increased.

Table 4.11: LCIP results for the network of 100 nodes and $p = 0.1$ (2,3 products).

		2 products				3 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	2	1	2	1.00	3	2	2	0.50
3	3	13	5	3	1.60	6	3	4	1.00
3	4	6	3	4	1.00	10	5	3	1.00
3	5	2	1	2	1.00	11	6	4	0.83
4	2	18	4	5	3.50	22	10	5	1.20
4	3	10	5	4	1.00	18	11	5	0.64
4	4	8	3	3	1.67	14	6	4	1.33
4	5	0	-	-	-	10	6	5	0.67
5	2	18	8	4	1.25	15	3	3	4.00
5	3	11	7	4	0.57	17	7	4	1.43
5	4	10	8	7	0.25	15	6	8	1.50
5	5	15	9	4	0.67	10	5	4	1.00
6	2	25	10	6	1.50	22	9	7	1.44
6	3	14	8	4	0.75	21	8	4	1.63
6	4	15	5	4	2.00	8	3	4	1.67
6	5	0	-	-	-	6	3	3	1.00

Table 4.12: LCIP results for the network of 200 nodes and $p = 0.1$ (2,3 products).

		2 products				3 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	17	6	4	1.83	9	4	3	1.25
3	3	18	8	4	1.25	22	10	4	1.20
3	4	10	4	3	1.50	19	13	4	0.46
3	5	5	2	2	1.50	7	3	3	1.33
4	2	29	12	5	1.42	34	11	4	2.09
4	3	15	7	3	1.14	13	6	4	1.17
4	4	11	5	2	1.20	18	9	4	1.00
4	5	16	7	5	1.29	15	7	5	1.14
5	2	25	11	5	1.27	32	13	4	1.46
5	3	18	11	5	0.64	21	7	6	2.00
5	4	13	7	4	0.86	31	12	5	1.58
5	5	23	12	5	0.92	27	14	5	0.93
6	2	38*	16	7	1.38	45*	22	6	1.05
6	3	15	9	4	0.67	22	11	7	1.00
6	4	23	13	5	0.77	20	12	4	0.67
6	5	18	10	5	0.80	9	3	3	2.00

Table 4.13: LCIP results for the network of 300 nodes and $p = 0.1$ (2,3 products).

		2 products				3 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	27	9	5	2.00	15	5	4	2.00
3	3	26	11	4	1.36	40	16	4	1.50
3	4	2	1	2	1.00	14	7	4	1.00
3	5	15	7	4	1.14	16	8	4	1.00
4	2	29	13	4	1.23	23	7	4	2.29
4	3	20	8	4	1.50	26	12	4	1.17
4	4	2	1	2	1.00	24	10	5	1.40
4	5	20	10	4	1.00	14	8	3	0.75
5	2	43	19	4	1.26	42	20	5	1.10
5	3	33	17	4	0.94	35	17	4	1.06
5	4	15	6	5	1.50	31	13	7	1.38
5	5	10	5	3	1.00	19	8	3	1.38
6	2	46*	22	6	1.09	73*	32	4	1.28
6	3	28	16	5	0.75	37	16	7	1.31
6	4	29	15	3	0.93	37	24	4	0.54
6	5	13	6	4	1.17	15	7	3	1.14

Table 4.14: LCIP results for the network of 100 nodes and $p = 0.2$ (2,3 products).

		2 products				3 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	19	5	4	2.80	11	1	5	10.00
3	3	4	2	2	1.00	7	5	4	0.40
3	4	3	2	1	0.50	7	4	3	0.75
3	5	11	4	4	1.75	7	5	3	0.40
4	2	16	5	4	2.20	2	1	2	1.00
4	3	2	1	2	1.00	5	2	2	1.50
4	4	7	3	2	1.33	5	2	3	1.50
4	5	9	4	6	1.25	13	8	5	0.63
5	2	16	6	3	1.67	31	13	7	1.38
5	3	15	6	5	1.50	21	14	6	0.50
5	4	10	7	4	0.43	27	12	6	1.25
5	5	7	4	4	0.75	14	5	6	1.80
6	2	32	15	6	1.13	25	12	4	1.08
6	3	17	9	4	0.89	22	11	3	1.00
6	4	12	6	5	1.00	11	5	5	1.20
6	5	14	6	5	1.33	19	10	4	0.90

Table 4.15: LCIP results for the network of 200 nodes and $p = 0.2$ (2,3 products).

		2 products				3 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	10	4	4	1.50	10	3	4	2.33
3	3	10	4	4	1.50	15	6	3	1.50
3	4	11	5	3	1.20	25	13	6	0.92
3	5	12	6	4	1.00	7	4	3	0.75
4	2	30	10	6	2.00	26	9	6	1.89
4	3	5	3	3	0.67	28	11	6	1.55
4	4	16	11	5	0.45	14	9	6	0.56
4	5	7	4	3	0.75	21	10	4	1.10
5	2	21	12	6	0.75	39	20	7	0.95
5	3	13	7	4	0.86	18	7	5	1.57
5	4	14	6	4	1.33	17	8	4	1.13
5	5	14	8	3	0.75	7	4	3	0.75
6	2	30	15	5	1.00	45	20	6	1.25
6	3	18	9	5	1.00	25	14	5	0.79
6	4	24	15	4	0.60	35	15	5	1.33
6	5	17	10	5	0.70	15	7	5	1.14

Table 4.16: LCIP results for the network of 300 nodes and $p = 0.2$ (2,3 products).

		2 products				3 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	15	5	3	2.00	15	8	3	0.88
3	3	7	1	2	6.00	15	2	3	6.50
3	4	6	3	3	1.00	18	9	4	1.00
3	5	9	3	3	2.00	14	4	3	2.50
4	2	36	14	5	1.57	36	11	6	2.27
4	3	24	11	4	1.18	35	15	6	1.33
4	4	11	6	3	0.83	18	8	3	1.25
4	5	12	5	3	1.40	22	11	3	1.00
5	2	22	11	4	1.00	42	19	6	1.21
5	3	26	15	5	0.73	32	16	5	1.00
5	4	17	7	4	1.43	15	8	3	0.88
5	5	66	4	3	15.50	18	9	5	1.00
6	2	39	19	7	1.05	43	19	6	1.26
6	3	28	18	7	0.56	28	13	4	1.15
6	4	18	10	4	0.80	33	20	7	0.65
6	5	12	7	2	0.71	31	16	4	0.94

Table 4.17: LCIP results for the network of 100 nodes and $p = 0.3$ (2,3 products).

		2 products				3 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	14	5	4	1.80	6	2	3	2.00
3	3	7	4	3	0.75	9	4	6	1.25
3	4	0	-	-	-	4	2	2	1.00
3	5	11	6	4	0.83	16	6	5	1.67
4	2	4	2	2	1.00	5	1	3	4.00
4	3	12	4	3	2.00	4	2	2	1.00
4	4	7	4	5	0.75	6	4	3	0.50
4	5	4	2	2	1.00	2	1	2	1.00
5	2	15	7	3	1.14	15	7	4	1.14
5	3	13	6	4	1.17	23	9	7	1.56
5	4	10	5	4	1.00	17	10	4	0.70
5	5	17	11	4	0.55	10	5	3	1.00
6	2	31	12	5	1.58	29	16	7	0.81
6	3	18	7	4	1.57	5	2	3	1.50
6	4	13	6	3	1.17	17	10	5	0.70
6	5	7	3	5	1.33	17	8	4	1.13

Table 4.18: LCIP results for the network of 200 nodes and $p = 0.3$ (2,3 products).

		2 products				3 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	4	1	2	3.00	20	9	4	1.22
3	3	11	4	3	1.75	24	12	4	1.00
3	4	6	3	3	1.00	12	5	2	1.40
3	5	8	2	4	3.00	25	10	4	1.50
4	2	21	9	5	1.33	22	4	6	4.50
4	3	14	7	5	1.00	16	9	4	0.78
4	4	2	1	2	1.00	22	12	4	0.83
4	5	7	4	4	0.75	18	9	4	1.00
5	2	31*	17	6	0.82	37	17	5	1.18
5	3	16	10	5	0.60	24	13	6	0.85
5	4	7	3	3	1.33	23	12	4	0.92
5	5	12	6	4	1.00	13	7	4	0.86
6	2	38	15	9	1.53	23	15	6	0.53
6	3	6	4	3	0.50	19	11	3	0.73
6	4	8	5	4	0.60	12	4	4	2.00
6	5	10	6	4	0.67	12	7	3	0.71

Table 4.19: LCIP results for the network of 300 nodes and $p = 0.3$ (2,3 products).

		2 products				3 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	18	8	3	1.25	27	10	5	1.70
3	3	17	6	3	1.83	35	13	8	1.69
3	4	12	5	3	1.40	20	10	4	1.00
3	5	13	8	3	0.63	32	12	5	1.67
4	2	27	8	4	2.38	27	13	4	1.08
4	3	19	10	5	0.90	23	8	4	1.88
4	4	20	9	5	1.22	23	14	4	0.64
4	5	22	12	4	0.83	17	8	4	1.13
5	2	32	13	5	1.46	76*	30	4	1.53
5	3	29	13	5	1.23	40	16	6	1.50
5	4	20	11	5	0.82	15	8	3	0.88
5	5	15	8	4	0.88	16	7	6	1.29
6	2	38	22	5	0.73	54*	24	6	1.25
6	3	26	17	5	0.53	34	17	4	1.00
6	4	22	12	3	0.83	23	12	5	0.92
6	5	23	11	4	1.09	30	13	4	1.31

Table 4.20: LCIP results for the network of 100 nodes and $p = 0.2$ (4,5 products).

		4 products				5 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	8	1	2	7.00	0	-	-	-
3	3	7	4	4	0.75	7	2	3	2.50
3	4	11	6	3	0.83	3	2	1	0.50
3	5	11	4	4	1.75	11	4	3	1.75
4	2	6	4	3	0.50	4	2	2	1.00
4	3	4	1	2	3.00	4	2	2	1.00
4	4	5	2	2	1.50	18	9	5	1.00
4	5	13	7	6	0.86	13	7	2	0.86
5	2	34	13	7	1.62	43	13	6	2.31
5	3	30*	16	9	0.88	22	12	6	0.83
5	4	10	6	5	0.67	19	7	5	1.71
5	5	14	10	7	0.40	16	6	4	1.67
6	2	32	11	8	1.91	16	9	4	0.78
6	3	17	8	5	1.13	23	10	5	1.30
6	4	15	8	4	0.88	19	8	6	1.38
6	5	17	9	5	0.89	19	9	5	1.11

Table 4.21: LCIP results for the network of 200 nodes and $p = 0.2$ (4,5 products).

		4 products				5 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	0	-	-	-	0	-	-	-
3	3	12	6	3	1.00	16	10	4	0.60
3	4	16	7	5	1.29	35	15	5	1.33
3	5	20	9	4	1.22	16	6	4	1.67
4	2	32	13	7	1.46	16	7	4	1.29
4	3	33	12	7	1.75	23	11	3	1.09
4	4	21	14	6	0.50	27	14	4	0.93
4	5	16	8	3	1.00	28	13	5	1.15
5	2	34*	18	6	0.89	27	13	7	1.08
5	3	25	13	4	0.92	25	12	5	1.08
5	4	17	8	4	1.13	11	5	3	1.20
5	5	22	11	5	1.00	21	9	6	1.33
6	2	25	11	7	1.27	58	23	5	1.52
6	3	24	9	5	1.67	49*	23	6	1.13
6	4	32	13	6	1.46	17	9	5	0.89
6	5	24	10	3	1.40	28	13	5	1.15

Table 4.22: LCIP results for the network of 300 nodes and $p = 0.2$ (4,5 products).

		4 products				5 products			
Num. of attributes	Num. of levels	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return	Num. of new buyers	Num. of people with incentives	Num. of Iterations	Return
3	2	11	4	3	1.75	2	1	2	1.00
3	3	14	5	3	1.80	34	14	5	1.43
3	4	21	8	4	1.63	35	15	4	1.33
3	5	21	8	5	1.63	4	2	2	1.00
4	2	30	12	6	1.50	44	13	5	2.38
4	3	43*	19	5	1.26	34	20	6	0.70
4	4	28	16	5	0.75	22	12	6	0.83
4	5	28	15	4	0.87	24	13	4	0.85
5	2	66*	32	4	1.06	65*	23	5	1.83
5	3	32	15	5	1.13	41	19	5	1.16
5	4	40	15	5	1.67	42	20	5	1.10
5	5	35*	15	6	1.33	32	14	5	1.29
6	2	61*	26	5	1.35	53	26	5	1.04
6	3	32	14	5	1.29	39	17	6	1.29
6	4	24	11	5	1.18	52*	28	5	0.86
6	5	39	16	5	1.44	50*	21	6	1.38

4.6 Concluding Remarks

In this chapter, we looked at a product line design problem taking into account peer influence effects. We model these effects in two dimensions as first and second order effects, and extend our genetic algorithm for the single product model to solve this computationally challenging problem. The least cost influence problem in the single product model is also encountered in the multiple product model, and we propose simple methods to adapt the existing model to the product line problem, and to solve it in a fast and efficient method.

After extensive computational studies, we see that the market share increases as the size of a product line increases. This is expected since each additional product will be designed to satisfy the hurdles of the nodes which are not buyers. So increasing the size of the product line always results in an increase in the market share. The number of products where the marginal contribution of the next added product to the market share will be zero would be the point right after the whole market share is reached. However the costs of introducing a new product are overlooked in our model. An important trade-off of interest is whether a company can be better off by providing higher amount of incentives to the people with the current number of products, or by incrementing the size of the product line.

The model we proposed in this chapter allows product line design problem to take into account peer influence effects which previously was not possible. We see that an optimal product profile for a single product model may not stay optimal as the number of products are increased in the product line. We did not encounter any evidence suggesting a decrease in the amount of incentives required to reach the solution of the GA as the number of products increased.

The running times of the GA increase rapidly, and the main reason behind this is that the fitness is evaluated by the integer program SOCNEPL. We believe a stronger IP model can help to decrease the search time.

Chapter 5

Extensions

In this chapter, we first discuss how the model for the least cost influence problem can be applied in an epidemiological setting and then introduce two extensions to the SOCSNE model. We first present a model which takes into account budgetary concerns. Next, we present a comprehensive model in which we solve the product design problem while minimizing the incentives distributed to the market, i.e., it models the product design and diffusion problems simultaneously.

5.1 Application of the LCIP in Epidemiology and a Slight Generalization

Although the LCIP has been framed in a product diffusion setting, it can also be equivalently viewed in an epidemiological setting. Suppose that e_{js} denotes the risk factors or influence of untreated node j on node s (e.g., if δ_{js} denotes the probability of node s getting infected by untreated node j, then $e_{js} = -\log(1 - \delta_{js})$). Let f_{js} denote the reduction of influence of node j on node s if node j is treated so that its risk level is less than or equal to a threshold risk level r_j. In other words, if node j is treated so that its risk level is below r_j, its influence on node s is $e_{js} - f_{js}$. We would like to ensure that the sum of all $e_{js} - f_{js}$'s for node s minus the intervention or treatment strategy z_s reduces the overall risk of node s below the threshold risk level r_s. This may be equivalently set in the marketing setting with $b_s = -r_s + \sum_{j \in V'} e_{js}$ and $d_{js} = f_{js}$, with a discrete set of intervention or treatment strategy choices at each node (with associated costs). The LCIP in the epidemiological setting is then the problem of finding a least cost treatment plan to ensure that a given population has its risk levels for a particular epidemic reduced to below a target threshold level for each member of the population.

This illustrates that the LCIP is an extremely useful model in a social network setting where the behavior of one's immediate neighbors influences ones own, and it is of interest to understand how "information" (loosely defined) spreads through a network. In particular, it is of interest to understand the power of different nodes in a network in terms of their relative "influence" in helping spread (or stop the spread) the "information" over a network.

As explained in Section 3.3, in a setting where it is desired that a fraction of the population (as opposed to the entire population) be influenced, the LCIP model can be easily modified.

5.2 Budget constraint for SOCSNE

In many articles that study conjoint analysis, cost has been either explicitly included in the price or has been included in the objective function negatively while maximizing total profit (Dobson and Kalish, 1993). In many cases however, because of the financial market conditions, the budget is not always available to develop a product that will return the highest market share. Consequently, it is important to take product design decisions in the context of budgetary constraints at the manufacturing organization. This is especially true if the costs are in terms of man-power or time, or the focus is a pre-stated design objective (e.g., design a laptop with a manufacturing cost of less than \$300 while maximizing market share). Consequently, rather than finding the optimal design that will produce the highest market share, the question is to find the optimal design that will give the largest market share *under a limited design budget*. To address these concerns, we propose adding constraint (5.1) to the SOCSNE model. The design cost associated with level l of attribute k is given by c_{kl} and the total budget is B. In this way, the total cost of the selected levels for the attributes will not be allowed to exceed a given design budget.

$$\sum_{k=1}^{K} \sum_{l=1}^{L_k} c_{kl} x_{kl} \leq B. \tag{5.1}$$

In order to take into account the budget constraint, the evaluation phase of the genetic algorithm needs to be modified. Remember that the fitness of a product profile is calculated optimally using the integer program, SOCSNE. Now, if the evaluation is made after constraint 5.1 is added to the SOCSNE, we can obtain product profiles which will not violate the budget constraint.

5.3 A Comprehensive Model

An interesting question is whether the two problems (share-of-choice problem with network effects and the least cost influence problem) can be solved (both for product design and the total amount of incentives) in conjunction. We provide a slightly more complex integer programming model where the two problems are integrated. This model allows one to find the optimal product profile by maximizing the market share and minimizing the amount of incentives at the same time.

SOCSNLCI:

$$\text{Maximize} \quad (\sum_{s=1}^{n} y_{s|T|}) - \gamma \sum_{s=1}^{n} z_s \tag{5.2}$$

$$\text{subject to} \quad \sum_{k=1}^{K}\sum_{l=1}^{L_k} u_{kl}^{s} x_{kl} \geq h_s^H y_{s0} \qquad s = 1, 2, ..., n, \tag{5.3}$$

$$\sum_{k=1}^{K}\sum_{l=1}^{L_k} u_{kl}^{s} x_{kl} \geq h_s^H y_{st} - z_s - \Delta_s \sum_{j\in V} a_{sj} y_{jt-1} \qquad s = 1, 2, ..., n, \forall t \geq 1, \tag{5.4}$$

$$\sum_{l=1}^{L_k} x_{kl} = 1 \qquad k = 1, 2, ..., K, \tag{5.5}$$

$$y_{s0} \leq y_{s1} \leq y_{s2} \leq \ldots \leq y_{s|T|} \qquad s = 1, 2, ..., n. \tag{5.6}$$

$$x_{kl}, y_{st} \in \{0, 1\}, \ z_s \geq 0 \tag{5.7}$$

$$k = 1, 2, ..., K, \ \ l = 1, 2, ..., L_k, \ \ s = 1, 2, ..., n, \ \ \forall t \geq 0.$$

The objective function is to maximize the market share, after the amount paid as incentives are subtracted from it. The same variables and parameters are used as the previous models (SOCSNE and LCIP) with an additional parameter $\gamma \geq 0$ that is introduced. Since z_s represents cost and the rest of the objective function represents market share, the comparison of objectives should be done accordingly. γ serves as a scaling parameter for the marketing/product promotion effects and represents the relative importance of the amount spent with respect to the market share. If $\gamma = 0$, the amount of incentives paid are not important at all and the objective becomes the same as the original problem (SOCSNE). If person s buys the product in time period t, $y_{st'}$ equals 1, $\forall t' \geq t$. So the market share equals to the number of people who buy in the last period. The first constraint represents the first period, similar to constraint (2.8) in the LCIP model. In constraint (5.4), hurdles and utilities are compared to ensure that only the people whose updated hurdles are less than or equal to their utilities buy the product. It is the same as constraint (2.9) in the LCIP model. Constraint (5.5) allows one level for each attribute. With constraint (5.6), purchasing decisions of customers are carried to the successor periods. Note that in this model, the $y_{s|T|}$ variables corresponding to the purchasing decisions in the last period are not necessarily 1. This is because the optimal product profile is not an input in this model, so at the end there may be customers who decide not to buy the product at all.

Alternatively, a coefficient equal to the revenue associated with person s could be assigned to $\sum_{s=1}^{n} y_{s|T|}$ while $\gamma = 1$ for a model that maximizes revenue.

Chapter 6

Summary and Concluding Remarks

In this thesis, we have focused on integrating social network effects in product design and diffusion problems. We started with a single product model and extended it to a product line design problem. The integer program proposed for the share-of-choice problem required the introduction of a complementary problem to reach the forecasted market share of the integer program. In the third problem of the thesis, we extended our study in the first and second problems for the product line design problem with a two dimensional influence model. We introduced a genetic algorithm for the first and the third problems and an optimal solution method for the second problem for small size networks.

This thesis lies in the intersection of the operations, marketing and complex systems research and is motivated by (i) the increased connectivity of the people around the world and (ii) our ability to observe the interaction/effects of these connections. Communication has become a lot easier, and it has helped strengthen information sharing among large social networks with faster and cheaper internet technologies improving day by day. In this thesis, we consider this social interaction and model influence among people when they are making purchasing decisions. We propose that if these network effects are taken into account when the product is designed, we can design better products that will reach a larger market share. The aim is to manage the introduction of a new product operationally (through product design) and strategically (through product diffusion) taking into account the spread of the adoption behavior over a social network with on-site (well advised) marketing interventions.

The share-of-choice problem is a product design problem that aims to find the best combination of levels of attributes of a product in order to create a product with the largest market share. We introduce network effects and formulate an integer programming model for the SOCSNE. However, since this formulation is computationally challenging to solve, we describe a genetic algorithm (GA) to generate high-quality heuristics for the SOCSNE. The GA alleviates the computational time significantly (capable of solving problems which could not be solved using commercial IP solvers) while finding feasible solutions which are on average 98.76% of the optimal solution.

In this setting, we introduce a secondary problem, the Least Cost Influential Problem (LCIP), of determining the least expensive way of influencing individuals in a social network. The necessary expenses to strengthen diffusion over the social network are similar to marketing costs of advertising in terms of free samples or discount coupons. The integer programming model for this problem is difficult to solve. On a tree network when influence from neighbors on an individual are equal to each other we show that a dynamic programming algorithm (which reduces to a greedy algorithm) polynomially solves the problem. When the influence depends on the neighbor (i.e., not the same for each neighbor), the problem is NP-Hard on tree networks. This secondary problem, LCIP, is of independent interest, as it addresses contagion models and the issue of determining influential nodes in a social network, which are of significant interest in marketing and epidemiological settings.

In the last part of the thesis, we solve the product line design problem with network effects. Here the problem is more complicated both because the network influence is more complex and because the size of the problem is larger. We distinguish our study from the literature with the introduction of network effects and by including the selection of the product with the highest utility. Inclusion of these properties prevents us from using previous solution methods. We propose a genetic algorithm which uses an integer program to evaluate fitness of an individual in the product population. We also extend our solution method for the LCIP for the product line design problem.

To our knowledge, this is the first study to integrate social network effects in the share-of-choice problem and coalesce optimization methods with marketing applications. We show that integrating social network effects in the product design problem leads to better products that will provide a larger utility to the user and a larger market share for the business. We propose smart and fast algorithms to solve problems arising in this context. Further, we believe there are many opportunities for operations research in terms of the variety of combinatorial optimization problems that appear, not only in a marketing setting but in general, when diffusion of information takes place. This thesis represents a big first step in developing a strong theoretical model and solution methods for product design that accounts for network effects. Going forward (as future research) it is important to gather real data and conduct studies using actual data to apply this theory to practice.

Bibliography

Adar, E., L.A. Adamic. 2005. Tracking information epidemics in blogspace. *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*. IEEE, 207–214.

Alexouda, G., K. Paparrizos. 2001. A genetic algorithm approach to the product line design problem using the seller's return criterion: An extensive comparative computational study. *European Journal of Operational Research* **134**(1) 165–178.

Aral, S., D. Walker. 2011. Creating social contagion through viral product design: A randomized trial of peer influence in networks. *Management Science* **57**(9) 1623.

Bakshy, E., J.M. Hofman, W.A. Mason, D.J. Watts. 2011. Identifying influencers on twitter. *Fourth ACM International Conference on Web Seach and Data Mining (WSDM)*.

Balakrishnan, PV, R. Gupta, V.S. Jacob. 2004. Development of hybrid genetic algorithms for product line designs. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **34**(1) 468–483.

Balakrishnan, P.V., V.S. Jacob. 1996. Genetic algorithms for product design. *Management Science* **42**(8) 1105–1117.

Barabasi, A.L., R. Albert. 1999. Emergence of scaling in random networks. *Science* **286**(5439) 509.

Bass, F.M. 1969. A new product growth for model consumer durables: The Bass model. *Management Science* **15**(5) 215–227.

Bearden, W.O., M.J. Etzel. 1982. Reference group influence on product and brand purchase decisions. *Journal of Consumer Research* **9**(2) 183–194.

Bell, D.R., S. Song. 2007. Neighborhood effects and trial on the internet: Evidence from online grocery retailing. *Quantitative Marketing and Economics* **5**(4) 361–400.

Belloni, A., R. Freund, M. Selove, D. Simester. 2008. Optimizing Product Line Designs: Efficient Methods and Comparisons. *Management Science* **54**(9) 1544–1552.

Berger, J., K. Milkman. 2012. What makes online content viral? *Forthcoming, Journal of Marketing Research* .

Berning, C.A.K., J. Jacoby. 1974. Patterns of information acquisition in new product purchases. *Journal of Consumer Research* **1**(2) 18–22.

Blume, L.E. 1993. The statistical mechanics of strategic interaction. *Games and economic behavior* **4** 387–424.

Bourne, F.S. 1957. Group influence in marketing and public relations. *Some Applications of Behavioral Research, Basil, Switzerland: UNESCO* 207–225.

Burnkrant, R.E., A. Cousineau. 1975. Informational and normative social influence in buyer behavior. *Journal of Consumer Research* **2**(3) 206–215.

Büther, M., D. Briskorn. 2012. Reducing the 0-1 knapsack problem with a single continuous variable to the standard 0-1 knapsack problem. *International Journal of Operations Research and Information Systems (IJORIS)* **3**(1) 1–12.

Camm, J.D., J.J. Cochran, D.J. Curry, S. Kannan. 2006. Conjoint optimization: An exact branch-and-bound algorithm for the share-of-choice problem. *Management Science* **52**(3) 435–447.

Chen, K.D., W.H. Hausman. 2000. Mathematical properties of the optimal product line selection problem using choice-based conjoint analysis. *Management Science* **46**(2) 327–332.

Dawande, M., V. Mookerjee, C. Sriskandarajah, Y. Zhu. 2011. Structural search and optimization in social networks. *INFORMS Journal on Computing* .

Dobson, G., S. Kalish. 1993. Heuristics for pricing and positioning a product-line using conjoint and cost data. *Management Science* **39**(2) 160–175.

Domingos, P. 2005. Mining social networks for viral marketing. *IEEE Intelligent Systems* **20**(1) 80–82.

Domingos, Pedro, Matt Richardson. 2001. Mining the network value of customers. *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data mining*. 57–66.

Easley, D., J. Kleinberg. 2010. *Networks, crowds, and markets*. Cambridge Univ Press.

Erdös, P., A. Rényi. 1959. On random graphs. *Publicationes mathematicae* **6**(290-297) 53–54.

Escalas, J.E., J.R. Bettman. 2003. You are what they eat: The influence of reference groups on consumers' connections to brands. *Journal of Consumer Psychology* **13**(3) 339–348.

Fruchter, GE, A. Fligler, R.S. Winer. 2006. Optimal product line design: Genetic algorithm approach to mitigate cannibalization. *Journal of optimization theory and applications* **131**(2) 227–244.

Garcia, R. 2005. Uses of agent-based modeling in innovation/new product development research. *Journal of Product Innovation Management* **22**(5) 380–398.

Goldberg, D.E. 1989. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA.

Goldenberg, J., S. Han, D. Lehmann, J. Hong. 2009. The role of hubs in the adoption process. *Journal of Marketing* **73**(2) 1–13.

Goldenberg, J., B. Libai, E. Muller. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters* **12**(3) 211–223.

Granovetter, M. 1978. Threshold models of collective behavior. *American Journal of Sociology* **83**(6) 1420–1443.

Green, P.E., A.M. Krieger. 1989. Recent contributions to optimal product positioning and buyer segmentation. *European Journal of Operational Research* **41**(2) 127–141.

Green, P.E., V.R. Rao. 1971. Conjoint measurement for quantifying judgmental data. *Journal of Marketing Research* **8**(3) 355–363.

Gupta, S., R. Kohli. 1990. Designing products and services for consumer welfare: Theoretical and empirical issues. *Marketing Science* **9**(3) 230–246.

Helm, S. 2000. Viral marketing-establishing customer relationships by “word-of-mouse”. *Electronic Markets* **10**(3) 158–161.

Hill, S., F. Provost, C. Volinsky. 2006. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science* 256–276.

Holland, J.H. 1975. *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, MI.

Howard, D.J., C. Gengler. 2001. Emotional contagion effects on product attitudes. *Journal of Consumer Research* **28**(2) 189–201.

Iyengar, R., S. Han, S. Gupta. 2009. Do friends influence purchases in a social network? *Harvard Business School Marketing Unit, Working paper* .

Iyengar, R., C. Van den Bulte, T.W. Valente. 2011. Opinion leadership and social contagion in new product diffusion. *Marketing Science* **30**(2) 195–212.

Jurvetson, S. 2000. What exactly is viral marketing. *Red Herring* **78** 110–112.

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

Kelman, H.C. 1961. Processes of opinion change. *Public Opinion Quarterly* **25**(1) 57.

Kempe, D., J. Kleinberg, É. Tardos. 2003. Maximizing the spread of influence through a social network. *Proceedings of the 9. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 137–146.

Kempf, K., E. Rash. 2011. Product line design and scheduling at intel. *Full written paper will appear in Interfaces September/October 2012 issue* URL `http://bit.ly/oo6QdV`.

Kohli, R., R. Krishnamurti. 1987. A heuristic approach to product design. *Management Science* **33**(12) 1523–1533.

Kohli, R., R. Krishnamurti. 1989. Optimal product design using conjoint analysis: Computational complexity and algorithms. *European Journal of Operational Research* **40**(2) 186–195.

Kwak, H., C. Lee, H. Park, S. Moon. 2010. What is twitter, a social network or a news media? *Proceedings of the 19th international conference on World wide web*. ACM, 591–600.

Leenders, R.T.A.J., J.M.L. van Engelen, J. Kratzer. 2003. Virtuality, communication, and new product team creativity: a social network perspective. *Journal of Engineering and Technology Management* **20**(1-2) 69–92.

Leibenstein, H. 1950. Bandwagon, snob, and Veblen effects in the theory of consumers' demand. *The Quarterly Journal of Economics* **64**(2) 183–207.

Leskovec, J., L.A. Adamic, B.A. Huberman. 2007. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)* **1**(1).

Luo, L. 2011. Product line design for consumer durables: An integrated marketing and engineering approach. *Journal of Marketing Research* **48**(1) 128–139.

Luo, X. 2009. Quantifying the long-term impact of negative word of mouth on cash flows and stock prices. *Marketing Science* **28**(1) 148–165.

Mahajan, V., E. Muller, F.M. Bass. 1990. New product diffusion models in marketing: A review and directions for research. *The Journal of Marketing* **54**(1) 1–26.

Manchanda, P., Y. Xie, N. Youn. 2008. The role of targeted communication and contagion in product adoption. *Marketing Science* **27**(6) 961–976.

Maniezzo, V., T. Stützle, S. Voß. 2009. *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*, vol. 10. Springer Verlag.

Marchand, H., L.A. Wolsey. 1999. The 0-1 knapsack problem with a single continuous variable. *Mathematical Programming* **85**(1) 15–33.

Merton, R.K., Alice Kitt Rossi. 1949. *Contributions to the theory of reference group behavior*. New York, The Free Press.

Michalewicz, Z. 1996. *Genetic algorithms+ data structures= evolution programs*. Springer-Verlag London, UK.

Miller, K.M., R. Hofstetter, H. Krohmer, Z.J. Zhang. 2011. How should consumers' willingness to pay be measured? An empirical comparison of state-of-the-art approaches. *Journal of Marketing Research* **48**(1) 172–184.

Mirza, B.J., B.J. Keller, N. Ramakrishnan. 2003. Studying recommendation algorithms by graph analysis. *Journal of Intelligent Information Systems* **20**(2) 131–160.

Money, R.B., M.C. Gilly, J.L. Graham. 1998. Explorations of national culture and word-of-mouth referral behavior in the purchase of industrial services in the United States and Japan. *The Journal of Marketing* **62**(4) 76–87.

Nair, S.K., L.S. Thakur, K.W. Wen. 1995. Near optimal solutions for product line design and selection: beam search heuristics. *Management Science* **41**(5) 767–785.

Narayan, V., V.R. Rao, C. Saunders. 2011. How peer influence affects attribute preferences: A Bayesian updating mechanism. *Marketing Science* **30**(2) 368–384.

Park, C.W., V.P. Lessig. 1977. Students and housewives: Differences in susceptibility to reference group influence. *Journal of Consumer Research* **4** 102–110.

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

R Development Core Team. 2010. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

Reichheld, F.F. 2003. The one number you need to grow. *Harvard Business Review* **81**(12) 46–55.

Reingen, P.H., J.B. Kernan. 1986. Analysis of referral networks in marketing: Methods and illustration. *Journal of Marketing Research* **23**(4) 370–378.

Richardson, M., P. Domingos. 2002. Mining knowledge-sharing sites for viral marketing. *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data mining*. 61–70.

Rogers, E.M. 1983 First Ed., 1962. *The Diffusion of innovations (Third Ed.*. New York: Free Press.

Sadikov, E., M. Medina, J. Leskovec, H. Garcia-Molina. 2011. Correcting for missing data in information cascades. *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 55–64.

Sawtooth Software. 2003. Advanced simulation module for product optimization v1.5. Tech. rep., Sawtooth Software, Inc., Sequim, WA.

Shi, L., S. Ólafsson, Q. Chen. 2001. An optimization framework for product design. *Management Science* **47**(12) 1681–1692.

Sosa, M.E., S.D. Eppinger, C.M. Rowles. 2004. The misalignment of product architecture and organizational structure in complex product development. *Management Science* **50**(12) 1674–1689.

Srinivasan, R., G.L. Lilien, A. Rangaswamy. 2004. First in, first out? The effects of network externalities on pioneer survival. *Journal of Marketing* **68**(1) 41–58.

Steiner, Winfried, Harald Hruschka. 2003. Genetic algorithms for product design: How well do they really work? *International Journal of Market Research* **45**(2) 229 – 240.

Stonedahl, F., W. Rand, U. Wilensky. 2010. Evolving viral marketing strategies. *Proceedings of the 12th annual conference on Genetic and Evolutionary Computation (GECCO)*. ACM, 1195–1202.

Story, L., B. Stone. 2007. Facebook retreats on online tracking. `http://tinyurl.com/4358cev`. *The New York Times*.

Strogatz, S.H. 2001. Exploring complex networks. *Nature* **410**(6825) 268–276.

Sun, E., I. Rosenn, C. Marlow, T. Lento. 2009. Gesundheit! modeling contagion through facebook news feed. *Proc. of International AAAI Conference on Weblogs and Social Media*.

Tarasewich, P., P.R. McMullen. 2001. A pruning heuristic for use with multisource product design. *European Journal of Operational Research* **128**(1) 58–73.

Trusov, M., A.V. Bodapati, R.E. Bucklin. 2010. Determining influential users in internet social networks. *Journal of Marketing Research* **47**(4) 643–658.

Tsafarakis, S., Y. Marinakis, N. Matsatsinis. 2011. Particle swarm optimization for optimal product line design. *International Journal of Research in Marketing* **28**(1) 13–22.

Van den Bulte, C., Y.V. Joshi. 2007. New product diffusion with influentials and imitators. *Marketing Science* **26**(3) 400–421.

Van den Bulte, C., S. Stremersch. 2004. Social contagion and income heterogeneity in new product diffusion: A meta-analytic test. *Marketing Science* **23**(4) 530–544.

Van den Bulte, C., S. Wuyts. 2007. *Social networks and marketing*. Marketing Science Institute.

Wang, X.J., J.D. Camm, D.J. Curry. 2009. A Branch-and-Price approach to the share-of-choice product line design problem. *Management Science* **55**(10) 1718–1728.

Watts, D.J., S.H. Strogatz. 1998. Collective dynamics of small-worldnetworks. *Nature* **393**(6684) 440–442.

Wiedemann, D.G. 2007. Exploring the Concept of Mobile Viral Marketing through Case Study Research. *Proceedings of the 2nd Conference on Mobility and Mobile Information Systems*. 49–60.

Wu, L.L., L. Lee. 2008. Online Social Comparison: Implications derived from web 2.0. *PACIS 2008 Proceedings* 197.

Zufryden, F.S. 1977. A conjoint measurement-based approach for optimal new product design and market segmentation. A. D. Shocker, ed., *Analytical Approaches to Product and Market Planning*. Marketing Science Institute, Cambridge, MA, 110–114.