NEW RESULTS FOR OPTIMIZATION IN STOCHASTIC NETWORKS

BY MERVE UNUVAR

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Operations Research

Written under the direction of

András Prékopa

and approved by

____

____

____

____

New Brunswick, New Jersey

October, 2012

ABSTRACT OF THE DISSERTATION

New Results for Optimization in Stochastic Networks

by Merve Unuvar

Dissertation Director: András Prékopa

We are interested in single commodity stochastic network design problems under probabilistic constraint with discrete and continuous random variables. We use a stochastic programming model under probabilistic constraint (also called a chance-constrained model) to study these problems.

The problem addressed in this research is how to find minimum cost optimal capacities at the nodes and/or arcs subject to the constraint that the demands should be met on a prescribed probability level (reliability constraint). In our first problem formulation, we formulate the reliability constraint in terms of the Gale-Hoffman feasibility inequalities. In latter formulations, we allow system to meet the demand at least k-out-of-n and consecutive k-out-of-n periods. The number of reliability constraints, in both cases, increases exponentially with the size of the nodes and therefore we identify the redundant constraints and reduce their number with elimination methods.

Even with the reduced number of inequalities, it is not simple to solve probabilistic constrained stochastic network problems due to the large number of efficient points that satisfy the probabilistic condition. To overcome the size limitation of the problem, we develop a new theorem for efficient point generation in the case when the random variables are discrete, and we use hybrid cutting plane / supporting hyperplane algorithm in the case when the random variables are continuous.

Acknowledgements

Even though writing a dissertation is an individual work, I could never have succeeded in completing this dissertation without the help, support, guidance and efforts of many people.

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. András Prékopa, for his endless patience, guidance, support, encouragement and belief in me. Throughout my graduate studies, he always believed in me and always made me feel that I could do better and better. His way of thinking, solving problems, teaching, and understanding, led me to admire him, not only as a teacher, but also as a role model for a person I would like to become in the future. Professor Prékopa introduced me to depths of mathematics -which I will be grateful forever- that has allowed me to discover the joys of both learning and applying knowledge to hard problems. Not only has Professor Prékopa shaped who I am as a scientist, but he also shaped my teaching, communication and collaboration skills. I have always said this to everyone, but I would like to express it here, one more time that I couldn't have finished this work without him. Thank you very much for everything Professor!

I would like to thank other members of my dissertation committee, Dr. Endre Boros, Dr. Adi Ben–Israel, Dr. Melike Baykal–Gursoy and Dr. Mine Subasi, for providing me support, feedback and encouragement during the improvement of this dissertation.

I especially would like to thank Dr. Endre Boros for his numerious avenues of support which include recommending me to internships, to my upcoming job and to opportunities on other countless occasions. To me, he is not only the director of RUTCOR but instead a supportive role model who is concerned with all of his students' success.

I also owe a special note of gratitude to Lynn Agre, Clare Smietna, Terry Hart and

Katie D'Agosta. They always cheered me up when I was in a bad mood with their big smiles and warm friendship. From the very first day of my arrival to RUTCOR until the last, they have made me feel like I am an important member of the big RUTCOR family.

I am very grateful to all of my friends and family here in the States and in Turkey. In this long journey, miles away from home, people I have met here have made me feel like I am home again. Everytime I visited Turkey, all of my friends there made me feel like I never left. These countless number of friends, who are now all around the world, would take their own long chapter in this dissertation to even name let alone express my feelings about.

My very precious thanks to my sister, roommate, and best friend, Seda. She took this long and difficult journey with me and believed in me in every second of it. Whenever I needed courage, talking to her helped ease everything.

Last and most importantly, Pierce, Schiller, who has given me the biggest support during the preparation of this dissertation deserves the most special thanks. He always listened to my research ideas patiently and gave me useful feedback. There are no words that can express my gratitude to him.

Dedication

To my mom Fehime and my dad Ergun.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Stochastic networks offer efficient solution methodologies for problems that emerge on the borderlines of operations research, applied mathematics, computer science and statistics. This new field uses a combination of techniques from several branches of mathematics including probability theory, stochastic programming, optimization, combinatorics and graph theory to solve challenging problems. These techniques have been applied to a variety of problems, including telecommunications, service operations, and social and financial networks.

The work here is largely influenced by the results of Prékopa and Unuvar (2012), Unuvar et. al. (2012 a) and Unuvar et. al. (2012 b). In Chapter 2, we describe the single commodity network design problem studied in Prékopa and Unuvar (2012). In Chapters 3 and 4, we describe the problems studied in Unuvar et. al. (2012 a) and Unuvar et. al. (2012 b), respectively.

The history of networks can be traced back to as early as second half of the 1700's when Gaspard Monge first originated the mass transportation problem (see Monge (1781)). In the 1800's, network problems became relevant to real world applications with the construction of the first railroad networks built in the United States. The next major advancement for networks came with Ohm's law in 1827 and Kirchhoff's fundamental equations on electric circuits in 1845. For the first time, electric flow could be represented as a network. In Kirchhoff's electric circuit theory, electric current is the flow between the junction points and resistances (See figure 1.1). Kirchhoff's Current Law (1.1)] states that the total current into a node must be the same as a total current

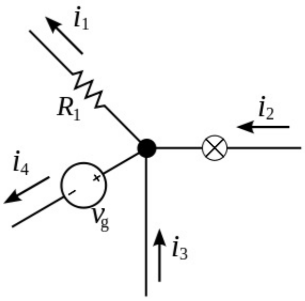


Figure 1.1: Electric circuit network flow given by Gustov Kirchhoff to show the current law: $I_2 + I_3 = I_1 + I_4$

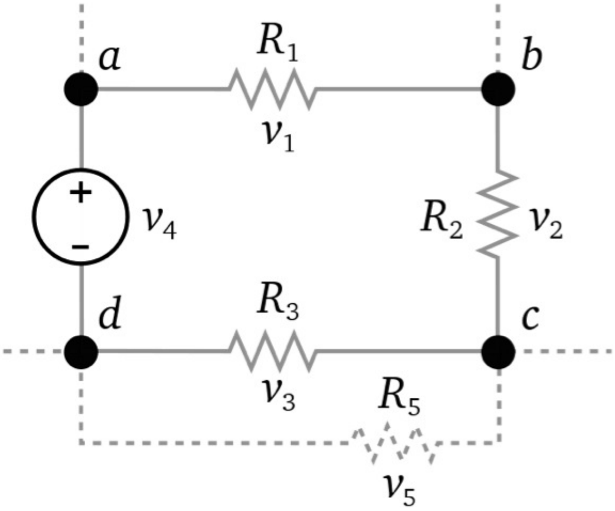


Figure 1.2: Closed circuit network given by Gustov Kirchhoff to show the voltage law: $V_1 + V_2 + V_3 + V_4 = 0$

out of the node. We express this as

$$\sum_{k=1}^{n} I_k = 0, \tag{1.1}$$

where n is the total number of branches with current flowing towards or away from a node. Kirchhoff's Voltage Law (1.2) states that the directed sum of electrical potential differences, the voltages, is zero in any closed network. We express this as

$$\sum_{k=1}^{n} V_k = 0, \tag{1.2}$$

where n is the number of edges in the network. Figure (1.2) gives an example of a closed network to which Kirchoff's Voltage Law can be applied. Later in the 1850's, Samuel Morse constructed the electrical telegraph, a major invention in long distance communication that used an electrical-communication system for the first time. Since then transportation and electric and power networks have been studied and applied all over the world. In the 1900's, A. N. Tolstoĭ, Tjalling Koopmans, Frank Hitchcock, and Lenoid Kantorovich made significant contributions to the theory of network science. Tolstoĭ and Kantorovich were interested in the applications involving transportation in the Soviet rail network. Ford and Fulkerson, influenced by the same problem, published a fundamental book on network flows in 1962. Since then a variety of network flow problems have been studied in computer science, mathematics and operations research.

The stochastic networks, where randomness is included in network parameters, began taking attention in the late 1900's. One of the first stochastic network design problem was modeled by Prékopa (1980) where he discovered that the Gale-Hoffman inequalities for network feasibility provide the necessary and sufficient condition for a feasible flow in power generation networks. Before his work, there were publications in connection with small size (three, four node) networks where the feasibility domain was known but it was an unsolved problem in the power network literature. The majority of the work in this research, Prékopa, Unuvar (2010), is based on the mentioned paper by Prékopa (1980).

In many network applications, such as telecommunication, power generation, and water reservoirs, one entity is being delivered from the source(s) to the sink(s). These are called single commodity networks. In telecommunication networks, phone calls between the specific locations define separate commodities. If the phone calls do not interact with each other in any way, then we model each phone call as a separate single commodity problem. For other problems, where physical entities have their own individual flow constraints, the single commodity network is not appropriate, and instead we use multi-commodity networks. Some examples of commodities which have their own individual flow constraints are vehicles in traffic problems, data in the communication problems, and different commodities in transportation problems. Even though many of the network problems concern multi-commodity networks, there are important applications where the network is a single commodity type. For example, operating power systems, water resources, road traffic, home security (evacuation), finance etc. For our purposes, single commodity networks are the center of interest.

The nodes and arcs together represent the graph structure of the network, or what is commonly called the topology of the network. The topology of a given network is very suitable to describe a number of structural (physical) properties of the network. For the networks arising from modeling power systems, the connectivity of the associated graph tests whether the system will survive if an arc failure occurs. Speed of the water flow in the water reservoir network, resistance or impedance in electrical/power network, and velocity of the vehicles in a traffic network are a few other examples of the physical properties of networks. Attempting to include all physical properties of the underlying commodity in the network model will lead to far too complex of a network to realistically solve. Instead, we model on a macroscopic level. That is to say, the nodes in our networks will represent an aggregate amount in an area (for instance, the total amount of power into a city instead of an an individual house).

There is also extensive work done in literature for the losses and delays on networks. Losses or delays are usually apparent in large scale computer networks (i.e. internet networks) such that data arrival rate to an arc exceeds output arc capacity, and therefore the data is either queued until an available arc capacity occurs or is lost. Another example is loss of active power in electric network systems due to ohmic resistance. The electric energy is converted into heat in these networks and Prékopa (2012) has a model where he takes into account the costs that occur due to transmission losses in power plants. Another representation of loss and delays can be seen in a large scale telecommunication network where the calls are queued until a line becomes available (Kelly, 1991a). Kelly (1991b) names these type of networks as loss networks and explains the details of solution methodologies when delay is taken into account. In our research, the nature of the networks are not designed for delays or losses therefore our type of network definition does not consider those concepts.

Militaries are both some of the earliest adopting and largest users of networks. During World War II, radar communication networks, transportation networks, and evacuation networks held significant interest to the United States. The work begun in this era has continued to present day leading the U.S. Department of Defense to initiate a special research center, Network Science CTA, under the Army Research Laboratory in 2009 for strategic defense reasons. Other governments, including the UK Ministry of Defense, coordinate with the United States to perform research in support of network centric operations addressing the needs that nation.

Besides its uses in the military, we observe networks in many aspect of our daily lives. Electrical and power networks enable us to operate modern machines, telephone and communication networks enable us to communicate across large distances, national highway networks, rail networks, and airline service networks enable us to travel, evacuation networks enable us to move outside a disaster area quickly, distribution networks provide the goods necessary for modern society, and computer networks allow us to share information rapidly. In all of these applications, our goal is to move an entity (electricity, power, information, vehicle etc.) from one point to another through an underlying network efficiently. Most of these examples contain random parameters (demand, supply, capacity, etc.). For instance, the demand in highway networks, which is the number of vehicles on the highway, is not predictable. Therefore, we model these networks stochastically. Stochastic networks are networks that involve randomness in one or more parameters.

Transportation networks are one of the most common applications of stochastic networks. These networks and associated problems often model a homogenous facility, such as a railroad system or highway exchange. From the operations research point of view, the most general form of a transportation problem is defined as follows: a shipper would like to distribute his goods to retail locations from his warehouse in a cost efficient way. In this problem, there is a capacity limitation on these distribution lines and there are demand constraints. When the parameters of this problem are deterministic, the network can be modeled as a deterministic type and solved by existing minimum cost network flow type algorithms. However, when there is randomness involved in one or more than one of the parameters of the problem formulation, the minimum cost network flow solution methodologies are not applicable. When such network contains a probabilistic parameter(s) or needs to satisfy its constraints at a predefined probability level (reliability), the transportation network is stochastic.

The study of transportation problems was initiated independently by Hitchcock, Kantorovitch and Koopmans in the mid 1900s. The earliest and most famous of this body of work was a transportation problem developed by Hitchcock known commonly as the Hitchcock Problem (Hitchcock, 1941). Dantzig (1963) improved on this work, by developing an algorithm to solve a network flow problem with a linear programming algorithm using simplex method which also could be successfully applied to solve the transportation problem. During World War II, Koopmans, who was a member of Combined Shipping Adjustment Board, approached the same problem, by considering economic efficiency point of view. Koopmans and Reiter (1951) further studied the problem and drew the attention to similarities between it and Maxwell-Kirchhoff electrical network. The networks arising from the above models are deterministic. The probabilistic transportation problem was first studied by Doulliez and Jamoulle in 1972, where they introduced a decomposition technique, which was improved by Wallace in 1986. Hassin and Zemel (1988) later studied another variation of the transportation problem from a probabilistic point of view, namely the capacitated transportation problem.

Traffic/road assignment networks frequently come up in connection with transportation networks. Chen et al. (2002) and Arnold et al. (2004) describe the literature of the road transportation systems. Chen et al (2002) also talks about the reliability of the road network in disaster circumstances such as earthquakes, floods etc. by considering the connectivity and travel time reliability. Waller and Ziliiaskopoulos (2006) model the traffic assignment problem with chance constraints as a network and most of the problems in this area are usually solved by a dynamic stochastic programming approach, known as dynamic traffic assignment models.

Stochastic networks are also used to model inventory problems. The goal of inventory research is to develop policies to minimize the related costs while meeting demand (usually probabilistic) and some service constraints. Most of the research in this area is related with distribution structures, and therefore falls under the study of networks. Daskin (1995), Mirchandani and Francis (1990), and Drezner (1995) are just a few among others who mainly focus on the trade-offs between the facility and locations. Further work was done by Daskin et. al (2002).

The networks arising from modeling evacuation problems are also stochastic. The evacuation networks are very important from the point of view of homeland security. Sherali et al. (1991) studied the static network model for shelter location and traffic routing during the evacuation. Similar to traffic assignment applications, most of the problems in this area are modeled as dynamic programming problems. The following example is an evacuation network of Cape May, NJ. The Category 4 Cape May hurricane in 1821 was the last major hurricane to make direct hit in New Jersey. Figure 1.3 shows the Cape May evacuation map which is also studied by Yazici, Ozbay (2007) to model the optimum shelter capacities on a evacuation route. In the network depicted in Figure 1.4, nodes represent the connection points to the highways and the directed arcs represent evacuation roads. People or vehicles are the entities that are carried along the arcs.

In electrical power engineering, the physical networks are also widely used and contain applications in power generation, transformation, control, power transmission and distribution networks. The network may represent an interconnected power system, where the nodes are the areas and the arcs are the transmission system. The power or energy is the entity that is distributed along the transmission lines in these type of networks. An example for an electric transmission system is depicted in Figure 1.5. This

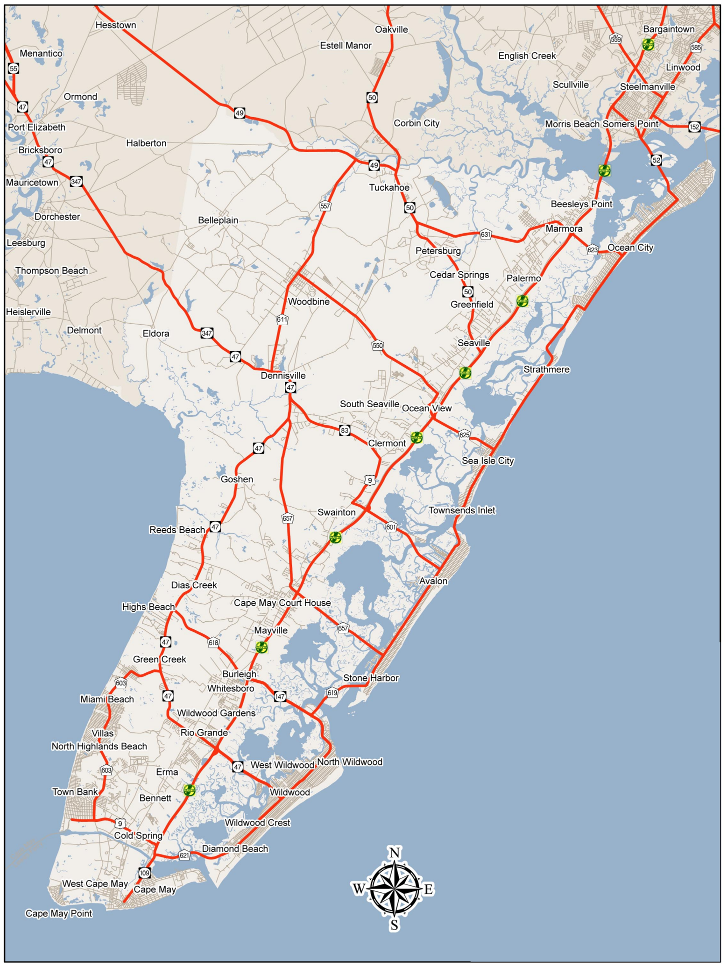


Figure 1.3: Cape May evacuation route, New Jersey Office of Emergency Management

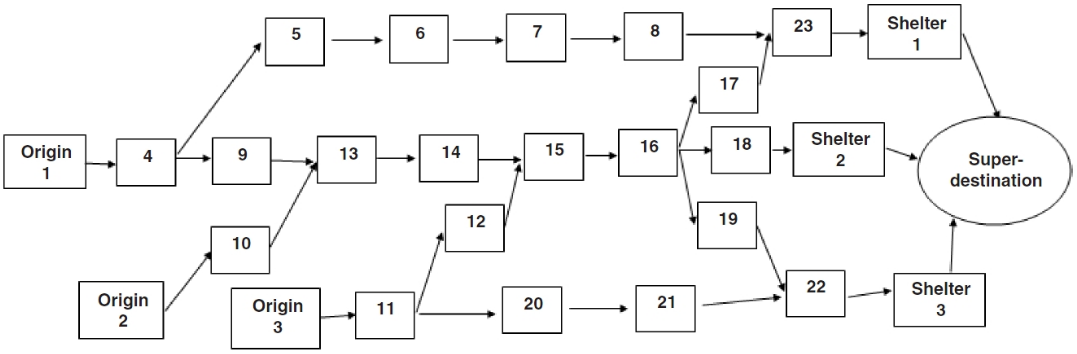
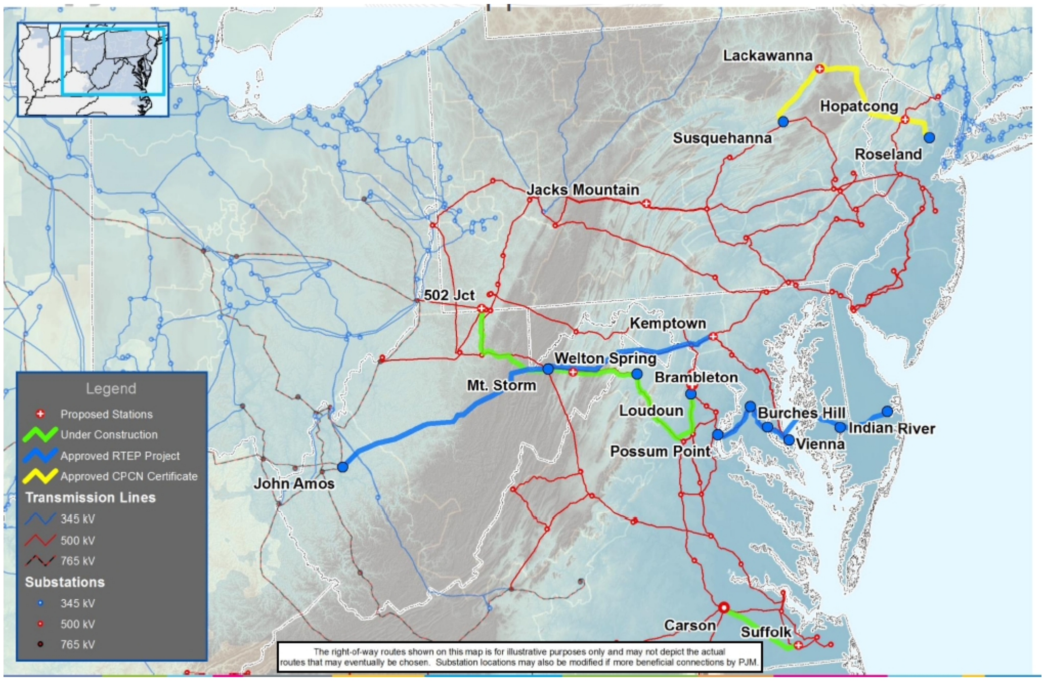


Figure 1.4: Simplified cell representation of Cape May evacuation network, Yazici, Ozbay (2007)



The right-of-way routes shown on this map is for illustrative purposes only and may not depict the actual routes that may eventually be chosen. Substation locations may also be modified if more beneficial connections by PJM.

Figure 1.5: Electric transmission system for North Eastern United States, PJM Interconnection

map represents the movement of wholesale electricity across the electric transmission network in all or parts of Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia and the District of Columbia. PJM Interconnection (www.pjm.com) is a regional transmission organization that coordinates the transmission of the electricity and the map in Figure 1.5 is taken from the company webpage. PJM monitors the flow of power across the transmission lines throughout a large portion of the North Eastern United States, thereby keeping a rich collection of data for the demand and the supply of electricity for the transmission network. It is very important from the practical point of view to keep track of the historical usage to optimize the distribution cost of the power while meeting the demand and supply constraints.

Networks arising from flood control reservoir systems are also stochastic. Mathematically, a natural river system can be represented by a rooted directed tree where the orientations of the edges coincide with the directions of the streamflows. It is possible

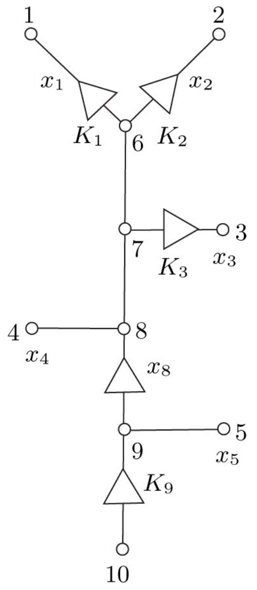


Figure 1.6: Example for river system topology and possible reservoir sites. Streamflow directions are indicated only on edges without reservoir, Prékopa, Szántai (1978)

to build reservoirs on these rivers to retain the flood. Prékopa, Szántai, (1978) has given a nonlinear model to optimize the capacities of such reservoirs by meeting some reliability constraints. Refer to the river system topology given in Figure 1.6 to construct such a model and design a flood control reservoir system for a river as a network flow problem.

Other applications where the process time of the jobs departing the system are larger than process time of arriving jobs can be classified under queueing networks. These type of networks are different than the ones that are mentioned above. Our primary goal in previous models is to satisfy the system demand for each node with the total supply that is available for the network without taking the arrival and departure time concept into consideration. Consider the telegraph network that is referenced in Kelly (1979) Reversibility and Stochastic Networks in Figure 1.7 (a). In this network, nodes correspond to the cities and arcs correspond to directed channels. The commodities are the messages that are generated in a city to be delivered to a destination city. The transmission of the message from a city cannot begin until the entire message has been collected in that city. Each transmission channel (arc) has its own maximum capacity.

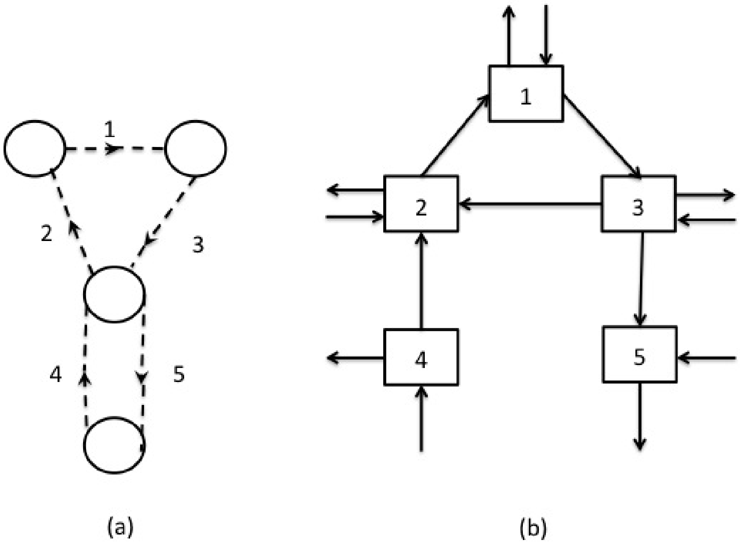


Figure 1.7: (a)A telegraph network system and (b) representation as a queuing network, Kelly (1979).

We can model this problem as a network of queues by considering each message as a customer and each channel as a queue (see Figure 1.7 (b)). In these systems, it is usually assumed that arrival of the messages are coming from outside the system as Poisson process distribution. For the time taken for a message to pass along a channel, we should consider different factors such as length of the message or other random effects that influence the distribution. Often, the time a message takes to pass along the channel is exponentially distributed and independent of the time it takes to pass along other channels along its route. The most likely queueing policy is first come first serve, but there are other networks where they consider the random order. Other applications of queueing networks can be machine interference, timesharing computers, teletraffic models, compartmental models and some other applications.

Network reliability design problems with probabilistic constraints where the goal is to establish a network of links to allow the flow of some commodities while satisfying the system demand is related to our type of network design problem. This type of problems have immediate applications in road networks, communication and supply chain networks. Karger (2001) designed a network that has an application in communication networks where he took into account the probabilistic arc failures as a network reliability. Chen et. al (2002) described solution methods varying from uncertainty analysis to Monte Carlo methods to optimize the design of the reliability of a road network. Asakura, Hato (2003) formulated a similar problem in connection with stochastic network design where the existence of a link is probabilistic. Santoso et al. (2005) applied stochastic programming approach for solving supply chain network design problems with probabilistic problem parameters. More recently, researchers are interested in not only satisfying the reliability constraints but also finding a minimum cost network design while satisfying these reliability constraints. Beraldi et al. (2010) introduced an integer programming problem and gave heuristic algorithms to find local optima or near-optimal solutions for such problems. Even though existing literature is very relevant to our research, we differ from them first, by working on the networks where the topology is already given. Second, by modelling and solving the network design problem as a stochastic programming problem under probabilistic constraints rather than Monte Carlo or heuristic algorithms.

The stochastic network design problems are stochastic programming models which can be subdivided into two types depending on whether the decision is made in a single period or in at least two periods. In the former case, time consists of a single period wherein the decision is made and the observation takes place after. We call this the static model. The decision is made only once after we observe the realized values of some random variables involved; we do not want to wait for the realization of all random variable(s). In the latter case, the system changes its state in time. If the decisions are made in at least two periods such that between any two decisions, observations take place, then we call this decision process the dynamic model. When the observed values in between subsequent decisions are random decision variables, the model is called stochastic static or stochastic dynamic. If the observed values were deterministic and known right at the outset, then in principle we would be able to make only one decision for the entire future of the system under investigation. We call these systems deterministic static or deterministic dynamic models. We will model our network design problem using a stochastic static model.

Our models are of stochastic programming type. Stochastic Programming is an extension of linear and nonlinear programming for decision models where parameters are not known with certainty. Thus, Stochastic Programming offers solutions for problems formulated in connection with stochastic systems. We either study the statistical properties of the random optimum value or we reformulate the problem into a decision type problem by considering the joint probability distribution of the random decision variables. In particular, we use a stochastic programming model under probabilistic constraint, also called chance constrained model. The first paper on programming under probabilistic constraints as a decision model under uncertainty was published by Charnes, Cooper and Symonds (1958). These authors used "chance constraint programming" for this model and variants. In this paper and Charnes and his coauthors' subsequent papers, individual probabilistic constraints imposed on each stochastic constraint. Therefore, this model is rarely legitimate from the point of view of probability theory and statistics theory. Miller and Wagner (1965) have used joint probabilistic constraint, but only in case of independent random variables when the convexity and algorithmic problems are not difficult to handle. Models, theories and algorithms for joint probabilistic constraints, where random variables may also be stochastically dependent, were first introduced and developed by Prékopa in a series of publications (1970, 1971, 1973, 1980, 1995, etc.)

A stochastic network design problem with probabilistic constraints was first introduced in Prékopa (1980). It is a two-stage dynamic type model, but no solution method was proposed. In this research we look at a related static problem and propose an elegant and efficient solution method for it. We will use the method of p-efficient points (see Prékopa 1990 a, Prékopa et al. 1998, Dentcheva et al. 2000) for the discrete random variables and use the supporting hyperplane method (see Kelley 1960, Veinott 1967, Prékopa, Szántai 1978 and Szántai 1988) for the continuous random variables. We will also use preprocessing and reliability results that have been presented by Prékopa and Boros (1991) and Wallace, Wets (1993). See also the presentation of the network reliability calculation and network design model construction in Prékopa (1995). Recently, the method of p-efficient points captured great interest in the civil engineering literature (see Yazici and Ozbay (2007)). We also mention recent papers by Thapalia, Crainic, Kaut, Wallace (2010) and Thapalia, Kaut, Wallace, Crainic (2010), where the reader can find ideas in connection with single commodity stochastic network design, even though their main interest of these authors is different from ours.

This paper is organized as follows.

In the second chapter, we describe a static model, under the assumption that the system is influenced by randomness and we have to decide in the face of uncertainty. Thus, we are dealing with stochastic networks. Our main concern is the handling of reliability, by the use of probabilistic constraints that we include in the model. We will distinguish between local demands and system demands. The local demands are the ζ_i, where i designates a node. For example, in case of an interconnected power system, ζ_i may represent the local demand for power in area i and x_i the local generating capacity. However, x_i may be reduced and the available generating capacity is $x_i - \eta_i$. The system demand is defined as $\eta_i + \zeta_i - x_i \doteq \xi_i - x_i$. Thus, ξ_i represents the local demand plus the deficiency in the generation capacity. In what follows we will simply call ξ_i the local demand. Deficiency may exist in the arc capacities as we will discuss later.

In all our models, we use joint probabilistic constraints, meaning that the constraints are satisfied at some prescribed level of probability for the entire time horizon. The use of simple individual probabilistic constraints, which are much simpler from the algorithmic point of view that satisfies the probabilistic constraints at each step separately, fails to represent the real life applications. Most of the water resource management models use simple individual probabilistic constraints, even though it has been shown that individual constraint models are not the right models for this application (Ackooij et all, 2010). In our models, we use joint probabilistic constraints. As an example, consider an interconnected power system consisting of two areas and a transmission line, described in Prékopa (1995), page 452.

The organization of the Chapter 2 is as follows. We first give definitions that we use for networks, network flows and demands. Then we present the Gale-Hoffman Theorem and its refinements by Prékopa, Boros (1991) and Wallace, Wets (1993) for feasibility of a demand function. Then we describe how we can eliminate redundant inequalities from the feasibility condition. The notion of a p-efficient point is recalled and an important theorem is proved that makes possible to write up the joint probabilistic constraint for the demand feasibility inequalities, using the p-efficient points of a much smaller system of inequalities. Later we formulate and solve a static stochastic network design problem. Section 2.3 is devoted to the case where the demands corresponding to the nodes are independent. In this case the p-efficient points are generated by solutions of multiple choice knapsack problems. We summarize the solution algorithm for the static problem and present two numerical examples for an 8-node network.

In Chapter 3, we formulate and solve probabilistic constrained stochastic programming problems, where we prescribe lower bounds for k-out-of-n and consecutive-k-out-of-n reliabilities in the form of probabilistic constraints. In probabilistic constrained stochastic programming problems we look at the joint probability of a finite number of stochastic constraints and impose a lower bound on it, chosen by ourselves. This ensures that the system we are looking at has a prescribed level of probability. The joint occurrence of constraints, or events, depending on a decision vector is, however, only one type of Boolean function of events among those that appear in reliability theory.

The problems mentioned above have various real life applications. As an example for k-out-of-n reliability constraints, one could determine the optimal safety cash reserve for a bank or one could determine the optimal safety stock for an inventory control system, where the probability constraints are satisfied at least k periods out of n periods. For the second type of reliability constraint given as the consecutive k-out-of-n, one could also find applications in the fields of banking or supply chain management. For example, banks often have legal regulations preventing them from failing to meet their customers' demand for consecutive k periods out of n periods and also in inventory control systems, there are generally policies which incur penalty costs associated with failing to meet demand more than k consecutive periods. The solution approach presented in this paper can be used for the applications where there is a supply and demand or similar type reliability constraints. Supplying goods to sparsely populated areas or commercial supply problems are just a few examples among many other supply/demand problem applications. Our methodology is novel, from the point of model construction, as it serves to enrich the collection of these stochastic programming models that have immediate and wide applications.

Another application exists in agricultural water resource problems. It is a widely accepted view, supported by practical experience, that a plant can survive a given number of dry days, which depends on the plant. If n is the total number of days until harvest and the maximum number of dry days the plant can survive is $k-1$, then we want to ensure the possibility of irrigation in any k consecutive days which means we have a consecutive k-out-of-n reliability. The problem is not to calculate the aforementioned reliability, but rather to optimize with respect to a decision variable subject to the constraint that the k-out-of-n reliability holds true on a prescribed probability level, near 1 in practice. We also look at probabilistic constraint problems, where the reliability is of a weaker type: of k-out-of-n type instead of consecutive k-out-of-n type.

The practical problem, to illustrate our solution methodologies, is mentioned in a paper by Prékopa, Szántai, Zsuffa (2010), where four optimization problems are formulated in connection with water resource problem. However, solutions are offered for three of them and the fourth one, which is the starting point of our research, was left unsolved. The problem is to determine the optimal capacity of a water release, or pump station, to satisfy the demand for irrigation, i.e., a reliability constraint where the reliability is one of the mentioned type.

Formulas are available to compute probability of various Boolean function of events. For the probability of at least k-out-of-n ($P_{(k)}$) and exactly k-out-of-n ($P_{[k]}$) we have, respectively,

$$P_{(k)} = \sum_{i=k}^{n} (-1)^{i-k} \binom{i-1}{k-1} S_k$$

$$P_{[k]} = \sum_{i=k}^{n} (-1)^{i-k} \binom{i}{k} S_k,$$

where $S_1, \ldots, S_n$ are the binomial moments of the random variable equal to the number of events that occur. However, in practice we cannot compute all $S_1, \ldots, S_n$. If n is large then we apply binomial procedures to approximate the probabilities.

In order to create lower and upper bounds for Boolean functions of events arranged in a finite sequence, a simple and frequently efficient method is the one provided by the discrete binomial moment problems. These are linear programs (LP's) where the numbers in the right-hand side are some of the binomial moments $S_1, S_2, \ldots$. Since S_k is the sum of joint probabilities of k-tuples of events, these LP's are called aggregated problems. Better bounds can be obtained if we use the individual probabilities in the sums of all S_k binomial moments that turn up in the aggregated problem. However, the LP's based on these, called the disaggregated problems, have huge sizes in general, and may be unsolvable (see Prékopa, Vizvári, Regös, 1998). Bounding probabilities of Boolean functions of events are studied extensively in literature. The first upper and lower bounds are given by Bonferroni (1937) and Boole (1854), respectively. However, they are weak and rarely useful in practice. Sharp S_1, S_2, S_3 lower bounds were proposed by Dawson and Sankoff (1967) and S_1, S_2, S_3 lower and upper bounds by Kwerel (1975a, 1975b). Prékopa (1988, 1989, 1990, 1995) generalized these results and gave formulas as well as dual type algorithms to obtain the bounds. See also Boros and Prékopa (1989) for a collection of formulas. We will use this in our research but we also use bounds where most sums of probabilities (as in $S_1, \ldots, S_n$) are calculated by the use of individual probabilities. Hunter (1976) gives a solution for an upper bound which is going to be used for the solution of the k-out-of-n type of problem. In Bukszár and Prékopa (2001), a third order upper bound by using graphs called cherry trees is presented. These are graphs that are recursively generated by connecting the new vertex into two already existing vertices. Cherry tree bounds also correspond to a dual feasible bases however they are always as good as or better than Hunter's upper bound. In Chapter 3, S_1, S_2, S_3 sharp lower bounds, Hunter and Cherry tree upper bounds are taken into consideration. Bi-section method is then applied to the model for obtaining the optimal capacity level while satisfying the reliability constraints.

The problem to be solved in Chapter 4 will be to find optimal reservoir capacities, that are serially linked to each other, such that at least k consecutive periods, there will be sufficient supply to meet the demand with a probability which is greater than or equal to a prescribed (in practice, high) probability thus consecutive k-out-of-n type reliability is again present. In this Chapter, different than Chapter 3, more than one reservoir case will be considered. For the sake of simplicity, we will be referring to an example from water resources application to demonstrate the problem.

Reservoir network systems can be represented by graphs where the nodes correspond to reservoirs, direct input (irrigation) or connection points depending on the value of the system demand function. Let $d(i) = \xi_i - x_i$ be the system demand function where ξ_i denotes the inflow and x_i represents the random local demand. At each node, $\xi_i = 0$ or $x_i = 0$ or both of them can be 0. If $x_i = 0$, the corresponding node represents a reservoir (K_1, K_2 etc in Figure 1.6), if $\xi_i = 0$, corresponding node represents an inflow $(3, 4$ etc in Figure 1.6), if both of them are 0, then corresponding node represents the junction or connection points (6, 7, 8 etc in Figure 1.6). The links that connect the nodes in a reservoir network refer to rivers or canals. In most of the models, the inputs to the nodes are water released from upstream reservoirs and probabilistic inflows from external resources such as rivers and rain. The outputs from the nodes are the amount of water that is going to be released during a given period such as day, week or a month.

We assume that released water leaves the system; thus we are considering reservoirs used for irrigation, municipal purposes, etc., and we exclude, for example, hydroelectric power generation applications. The objective function to be minimized is the sum of total building costs of each reservoir per its capacity therefore is to find the optimal water reservoir capacity building problem. The reliability constraints ensures the sufficient supply of water for the demand for at least consecutive k periods. We further assume that, for the numerical applications, the inflow and demand values will be normally distributed for each period. Also, in our models, we assume dependency between the inflow and demand, as well as dependency of their values between each period individually.

There is an extensive history of work that is done for water resources applications. Rivers are one of the most suitable representation forms for serially linked reservoirs placement therefore water engineering paid special attention in these type of problems.

The optimal operation of water reservoir networks usually modeled in connection with dynamic stochastic programming problems. Gal (1979), Yakowitz (1982), Archibadl et al. (1997) are just a few of the many publications where the solution to optimal water reservoir system is given by the techniques from the dynamic programming. We will be looking at the same problem but from the reliability point of view. We will introduce a stochastic programming problem in which chance constraint reliabilities are going to be met in a predefined probability level. The term "reliability programming" was first introduced in the water resources literature by Colorni and Fronza (1976). Since ReVelle et al. (1969) many authors have investigated the use of chance constrained reservoir network design problems. Kirby et al. (1970), Joeres et al. (1971), ReVelle and Gundelach (1975), Houck (1979) and Houck and Datta (1981) are just a few of them. Loucks (1970), Loucks and Dorfman (1975) and Stedinger (1983) studied a similar problem to ReVelle (1969) where the chance constrained reliabilities are solved with extra decision variables. Later, Colorni and Fronza (1976) modeled the same problem in connection with explicit decision variables unlike Loucks (1970). Simonovic (1979) developed a model for the long-term reservoir operating policies under chance-constrained reliability. Simonovic and Marino (1980, 1981), Simonovic and Orlob (1984), Marino and Mohammadi (1983) have also modeled different variations of reservoir system design and operating policies under probabilistic constraints.

The organization of the last chapter is as follows. In Section 4.2, we formulate and describe the nature of the problem and discuss the mathematical properties of the water reservoir model. In Section 4.3, we present the solution methodology and a hybrid algorithm to solve the problem and in section 4.4 a numerical example is given. Finally, in the appendix we list the distribution of random variables that are used in the numerical example, as well as the covariance matrices.

Chapter 2

Single Commodity Stochastic Network Design under Probabilistic Constraint with Discrete Random Variables

2.1 Introduction

2.1.1 Basic Definitions on Networks

Below we give the definitions for the network and network flow, suitable for our problem.

Definition A network $[N, y]$ is a pair of a finite set of nodes N and a capacity function $y(i, j)$ on the arcs $(i, j) \in N \times N$ assumed to have nonnegative values or $+\infty$.

While we speak about network flows, we typically have source, terminal and intermediate nodes. Sometimes we know exactly which node is of which type but sometimes the nodes randomly become sources or terminals and we cannot categorize them in advance. This is the case, for example, in interconnected power systems, where some of the nodes in the network represent areas that may have surplus generating capacities or may need assistance from other nodes to meet the demand, and it happens randomly. Here comes the first definition for a network flow:

Definition A flow or feasible flow is a function $f(i, j), (i, j) \in N \times N$ such that

$$\begin{gathered}
\textstyle\sum_{(i,j)\in N\times N} f(i,j) - \sum_{(j,i)\in N\times N} f(j,i) = d(i) = 0,\ \ \forall i \in I \\
\textstyle\sum_{(s,j)\in N\times N} f(i,t) \le d(S),\ \ s \in S \\
\textstyle\sum_{(i,t)\in N\times N} f(i,t) \ge d(t),\ \ t \in T \\
f(i,j) \le y(i,j),\ \ (i,j) \in N \times N.
\end{gathered} \tag{2.1}$$

where $d(i), i \in N$ is a given function that we call system demand function or briefly demand.

If we know in advance the types of the nodes, then we can write up equations/inequalities for each type to set up the constraints that the flow values have to satisfy. If, however, we do not know the types in advance, then it is more convenient to define the flow differently, e.g., the way Gale (1957) and Ford, Fulkerson (1962) have done it. This definition is presented below.

Definition A flow or feasible flow in the network is a function $f(i,j), (i,j) \in N \times N$ such that

$$\begin{aligned} f(i,j) + f(j,i) &= 0 \\ f(i,j) \leq y(i,j) &\text{ for all } (i,j) \in N \times N. \end{aligned} \tag{2.2}$$

The first line in condition (2.2) expresses a general convention that the flow from j to i is the negative of the flow from i to j. The first part of equalities in (2.2) also implies the flow balance constraints since it guarantees that the amount of flow that is leaving the node is equal to the amount of flow that is coming into the node. The second part of the inequalities in (2.2) named as capacity constraint since the amount of the flow cannot exceed the given capacity $y(i,j)$.

2.1.2 Feasible Demands

If the network flow is defined as in Definition 2.1.1, then we have to introduce additionally, the notion of a system demand and a feasible system demand.

In what follows, we use the following notations:

$$f(A,B) = \textstyle\sum_{i \in A,\ j \in B} f(i,j)$$
$$y(A,B) = \textstyle\sum_{i \in A,\ j \in B} y(i,j) \quad \text{for } A, B \subset N, \ \ AB = \emptyset$$

Definition A system demand is a function $d(j),\ j \in N$. In what follows we use the notation $d(A) = \sum_{i \in A} d(i)$ for $A \subset N$. A system demand is said to be feasible if there exists a flow f satisfying Definition 2.1.1 and the relations:

$$f(N,j) \geq d(j), \ \ j \in N. \tag{2.3}$$

Relations (2.2), (2.3) can be thought of as a system of homogenous linear inequalities, if we take $f(i,j)$, $y(i,j)$, $d(j)$, $j \in N$, $(i,j) \in N \times N$ as variables. An important question is: what are the conditions on $d(j)$, $y(i,j)$, $j \in N$, $(i,j) \in N \times N$ in order to ensure the existence of a feasible flow. The question was answered by Gale (1957) and Hoffman (1960). Gele (1957) gave a system of linear inequalities which are necessary and sufficient for feasibility of a set of supplies and demands in a network. Hoffman (1960) extended this work to feasible circulations. The inequalities are generated by considering all bipartitions of the vertices of the network thus the problem of determining the probability of a feasible flow in a stochastic network can be solved by referring to these inequalities. The following theorem summarizes the Gale and Hoffman's results:

Theorem 2.1.1 *The system demand $d(i)$, $i \in N$ is feasible iff the following inequalities hold:*

$$d(S) \leq y(\bar{S}, S), \;\; S \subset N, \quad \textit{where } \bar{S} = N \setminus S. \tag{2.4}$$

In what follows we call relations (2.4) Gale–Hoffman inequalities.

Let $|N| = n$. The number of Gale–Hoffman inequalities is $2^n - 1$, the case where $S = \emptyset$ being trivial. Since it is a large number, also if n is relatively small, we are looking for reduction of the Gale–Hoffman inequalities by eliminating redundant ones to obtain the minimum number of the Gale–Hoffman inequalities. An equality is redundant if and only if the induced subgraph on at least one of the partitions is not connected. A theorem, first proved by Prékopa and Boros (1991) and later by Wallace and Wets (1993), states the following:

Theorem 2.1.2 *The inequality* (2.4) *is redundant among the Gale–Hoffman inequalities if and only if at least one of the graphs $G(S), G(\bar{S})$ is not connected. In that case the inequality $d(S) \leq y(\bar{S}, S)$ is the sum of other Gale–Hoffman inequalities.*

Remark Prékopa and Boros (1991) stated only the sufficiency part of the theorem but their proof contains also the proof of the necessary part.

Prékopa and Boros (1991) worked out a variety of elimination procedures in connection with the Gale–Hoffman inequalities so that at the end only the non-redundant ones remain. In their paper, however, only the system demands are variables while the arc capacities are assumed to be constant. Since we look at the arc capacities also as variables, we have to slightly rework the elimination procedure. Below, we present the new version.

2.1.3 Preprocessing the Problem- Elimination of Redundant Inequalities

The elimination methods described below are studied in detail by Prékopa and Boros (1991).

Elimination by Network Topology

Based on Theorem 2.1.2, in this procedure we eliminate those inequalities in (2.4) which are sums of others. We subsequently enumerate the sets $S \subset N$ according to their cardinalities, and look for sets $S, \bar{S}$ such that at least one of $G(S), G(\bar{S})$ is not connected. Then, eliminate the corresponding inequality among those in (2.4). The steps of the elimination algorithm that is described in Prékopa, Boros (1991) is as follows:

- *Step 0.* Let $b(H) = 1$ and $e(H) = 0$ for all $H \subseteq N$, $H \neq \emptyset$
- *Step 1.* Choose a nonempty subset of $H \subseteq N$ such that $b(H) = 1$ and $e(H) = 0$, if no such subset exists, then STOP
- *Step 2.* Let $T \subseteq N \backslash H$ be a maximal subset such that there is no arc between T and H
- *Step 3.* Let $b(V) = 0$ for all $V \subseteq H \cup T$ where $V \cap T \neq \emptyset$ and $V \cap H \neq \emptyset$
- *Step 4.* Let $e(H) = 1$ and go back to step 1.

Example: To illustrate the elimination by network topology, we consider a new example which is a 5-node network in Figure (2.1). This same example will be used throughout this chapter to demonstrate further theorems and applications of our solution method.

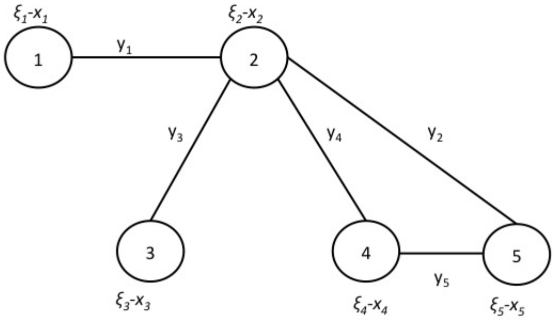


Figure 2.1: Five-node network with demand values $\xi_1 - x_1, \xi_2 - x_2, \xi_3 - x_3, \xi_4 - x_4, \xi_5 - x_5$ and interconnection capacities y_1, y_2, y_3, y_4, y_5.

Here are the Gale-Hoffman inequalities (2.5) for the 5-node network:

$$\begin{array}{ccc}
(1)\xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & 0 \\
(2)\xi_1 - x_1 & \leq & y_1 \\
(3)\xi_2 - x_2 & \leq & y_1 + y_2 + y_3 + y_4 \\
(4)\xi_3 - x_3 & \leq & y_3 \\
(5)\xi_4 - x_4 & \leq & y_4 + y_5 \\
(6)\xi_5 - x_5 & \leq & y_2 + y_5 \\
(7)\xi_1 - x_1 + \xi_2 - x_2 & \leq & y_2 + y_3 + y_4 \\
(8)\xi_1 - x_1 + \xi_3 - x_3 & \leq & y_1 + y_3 \\
(9)\xi_1 - x_1 + \xi_4 - x_4 & \leq & y_1 + y_4 + y_5 \\
(16)\xi_4 - x_4 + \xi_5 - x_5 & \leq & y_2 + y_4 \\
(17)\xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 & \leq & y_2 + y_4 \\
(18)\xi_1 - x_1 + \xi_2 - x_2 + \xi_4 - x_4 & \leq & y_2 + y_3 + y_5 \\
(19)\xi_1 - x_1 + \xi_2 - x_2 + \xi_5 - x_5 & \leq & y_3 + y_4 + y_5
\end{array}$$

$$\begin{array}{rcc}
(20)\xi_1 - x_1 + \xi_3 - x_3 + \xi_4 - x_4 & \leq & y_1 + y_3 + y_4 + y_5 \\
(21)\xi_1 - x_1 + \xi_3 - x_3 + \xi_5 - x_5 & \leq & y_1 + y_2 + y_3 + y_5 \\
(22)\xi_1 - x_1 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_1 + y_2 + y_4 \\
(23)\xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 & \leq & y_1 + y_2 + y_5 \\
(24)\xi_2 - x_2 + \xi_3 - x_3 + \xi_5 - x_5 & \leq & y_1 + y_4 + y_5 \\
(25)\xi_2 - x_2 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_1 + y_3 \\
(26)\xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_2 + y_3 + y_4 \\
(27)\xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 & \leq & y_2 + y_5 \\
(28)\xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 + \xi_5 - x_5 & \leq & y_4 + y_5 \\
(29)\xi_1 - x_1 + \xi_2 - x_2 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_3 \\
(30)\xi_1 - x_1 + \xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_1 + y_2 + y_3 + y_4 \\
(31)\xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_1
\end{array} \tag{2.5}$$

We observe $2^n - 1 = 31$ Gale–Hoffman inequalities, where $n = 5$. Which means that there is an inequality for each combination of the $n = 5$ nodes in the network with the exception of the case where $S = \emptyset$ (when S is empty, $d(S) = 0$ and it is trivial).

We will start the algorithm with $H = \{1\}$ and proceed the inequalities in numerical order.

- **Iteration 1**

  Step 1: $H = \{1\}$

  Step 2: $T = \{3, 4, 5\}$

  Step 3: $V = \{1,3\}, \{1,4\}, \{1,5\}, \{1,3,4\}, \{1,3,5\}$ and $b(V) = 0$

  Step 4: $e(\{1\}) = 1$.

Before continuing, let us describe the algorithm in more detail. Since $e(H)$ keeps record of whether the subset $H \subseteq N$ has ever processed or not during the Step 1 of the algorithm thus all possible subsets will be processed at most once during the algorithm. Furthermore, the algorithm only changes $e(H)$ value once, therefore through out the execution of the algorithm not all the subsets of $H \subseteq N$ need to be processed. One further remark should be that there might be occurrences of T being an empty set and in those instances, algorithm will only execute the change of $e(V)$ step. The binary variable $b(V)$ is an indicator whether an inequality is redundant or not. All subsets $H \subseteq N$ with $b(V) = 0$ can be eliminated at the end of the algorithm.

After the first iteration of the algorithm, we concluded that below inequalities can be eliminated since they are the sum of other inequalities. Let us demonstrate the redundancy of the inequality $V = \{1, 3\}$ explicitly:

$$\begin{array}{llllllll}
(2) & \xi_1 - x_1 & & & \leq & y_1 & & \\
(4) & & & \xi_3 - x_3 & \leq & & y_3 & \\
+ & & & & & & & \\
\hline
 & & & & & & & \\
(8) & \xi_1 - x_1 & + & \xi_3 - x_3 & \leq & y_1 & +y_3 & (\mathit{redundant})
\end{array}$$

Furthermore, $V = \{1, 4\}$ is equation (9) and sum of (2) and (5), $V = \{1, 5\}$ is equation (10) and sum of (2) and (6) , $V = \{1, 3, 4\}$ is equation (20) and sum of (2) , (4) and (5) $V = \{1, 3, 5\}$ is equation (21) and sum of (2), (4) and (6).

After the detailed description of the first iteration we can proceed in an abbreviated manner to complete the remaining iterations of the algorithm for our five-node example.

- **Iteration 2**

  Step 1: $H = \{2\}$

  Step 2: $T = \emptyset$

  Step 3: $V = \emptyset$

  Step 4: $e(\{2\}) = 1.$

- **Iteration 3**

  Step 1: $H = \{3\}$

  Step 2: $T = \{1, 4, 5\}$

  Step 3: $V = \{1, 3\}, \{3, 4\}, \{3, 5\}, \{1, 3, 4\}, \{1, 3, 5\}$ and $b(V) = 0$

  Step 4: $e(\{3\}) = 1.$

- **Iteration 4**

  Step 1: $H = \{4\}$

  Step 2: $T = \{1, 3\}$

  Step 3: $V = \{1, 4\}, \{3, 4\}$ all subsets of V has been considered before.

  Step 4: $e(\{4\}) = 1$.

- **Iteration 5**

  Step 1: $H = \{5\}$

  Step 2: $T = \{1, 3\}$

  Step 3: $V = \{1, 5\}, \{3, 5\}$ all subsets of V has been considered before.

  Step 4: $e(\{5\}) = 1$.

- **Iteration 6**

  Step 1: $H = \{1, 2\}$

  Step 2: $T = \emptyset$

  Step 3: $V = \emptyset$

  Step 4: $e(\{1, 2\}) = 1$.

The next subsets to be considered are $H = \{1, 3\}, \{1, 4\}$ and $\{1, 5\}$, which already eliminated in the previous iterations therefore we can proceed with the $H = \{2, 3\}$. However, $H = \{2, 3\}.\{2, 4\}, \{2, 5\}$ are adjacent to all other nodes therefore the subset V will always be an emptyset and there is no need to illustrate iteration 7, 8 and 9, explicitly.

- **Iteration 10**

  Step 1: $H = \{3, 4\}$

  Step 2: $T = \{1\}$

  Step 3: $V = \{1, 3\}, \{1, 4\}, \{1, 3, 4\}$ all subsets of V has been considered before.

  Step 4: $e(\{3, 4\}) = 1$.

- **Iteration 11**

  Step 1: $H = \{3, 5\}$

Step 2: $T = \{1\}$

Step 3: $V = \{1,3\}, \{1,5\}, \{1,3,5\}$all subsets of V has been considered before.

Step 4: $e(\{3,5\}) = 1$.

- **Iteration 12**

  Step 1: $H = \{4,5\}$

  Step 2: $T = \{1\}$

  Step 3: $V = \{1,4\}, \{1,5\}, \{1,4,5\}$all subsets of V has been considered before.

  Step 4: $e(\{4,5\}) = 1$.

Again, $H = \{1,2,3\}, \{1,2,4\}, \{1,2,5\}, \{2,3,4\}, \{2,4,5\}, \{3,4,5\}$ are adjacent to all other nodes and their subset combinations will always yield a nonempty set therefore there is no need to illustrate iteration 13, 14, 15, 16, 17, 18 explicitly. With the same reason, combination of all 4 and 5 subsets of the nodes will be adjacent to all other nodes therefore we can terminate the algorithm here. The algorithm will eliminate 8 out of 31 inequalities. The remaining non-eliminated inequalities are:

$$\begin{aligned}
(1)\xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 &\leq 0 \\
(2)\xi_1 - x_1 &\leq y_1 \\
(3)\xi_2 - x_2 &\leq y_1 + y_2 + y_3 + y_4 \\
(4)\xi_3 - x_3 &\leq y_3 \\
(5)\xi_4 - x_4 &\leq y_4 + y_5 \\
(6)\xi_5 - x_5 &\leq y_2 + y_5 \\
(7)\xi_1 - x_1 + \xi_2 - x_2 &\leq y_2 + y_3 + y_4 \\
(8)\xi_2 - x_2 + \xi_3 - x_3 &\leq y_1 + y_2 + y_4 \\
(9)\xi_2 - x_2 + \xi_4 - x_4 &\leq y_1 + y_2 + y_3 + y_5 \\
(10)\xi_2 - x_2 + \xi_5 - x_5 &\leq y_1 + y_3 + y_4 + y_5 \\
(11)\xi_4 - x_4 + \xi_5 - x_5 &\leq y_2 + y_4 \\
(12)\xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 &\leq y_2 + y_4 \\
(13)\xi_1 - x_1 + \xi_2 - x_2 + \xi_4 - x_4 &\leq y_2 + y_3 + y_5 \\
(14)\xi_1 - x_1 + \xi_2 - x_2 + \xi_5 - x_5 &\leq y_3 + y_4 + y_5 \\
(15)\xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 &\leq y_1 + y_2 + y_5 \\
(16)\xi_2 - x_2 + \xi_3 - x_3 + \xi_5 - x_5 &\leq y_1 + y_4 + y_5 \\
(17)\xi_2 - x_2 + \xi_4 - x_4 + \xi_5 - x_5 &\leq y_1 + y_3 \\
(18)\xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 &\leq y_2 + y_3 + y_4 \\
(19)\xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 &\leq y_2 + y_5 \\
(20)\xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 + \xi_5 - x_5 &\leq y_4 + y_5 \\
(21)\xi_1 - x_1 + \xi_2 - x_2 + \xi_4 - x_4 + \xi_5 - x_5 &\leq y_3 \\
(22)\xi_1 - x_1 + \xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 &\leq y_1 + y_2 + y_3 + y_4 \\
(23)\xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 &\leq y_1
\end{aligned} \tag{2.6}$$

We further assume that there are also known lower and upper bounds on the variables $d(i), y(i,j)$:

$$\begin{aligned} l(i) \le d(i) \le u(i), &\qquad i \in N \\ l(i,j) \le y(i,j) \le u(i,j), &\qquad (i,j) \in N \times N, \end{aligned}$$

where

$$l(i), l(i,j) \in R \cup \{-\infty\}, u(i), u(i,j) \in R \cup \{+\infty\}.$$

We define $l(A)$, $u(A)$, $A \subset N$, $l(A,B)$, $u(A,B)$, where, $A, B \subset N$, $AB = \emptyset$, in a similar way as we have defined $d(A)$, $f(A,B)$, $y(A,B)$.

Elimination by Upper Bounds

If for an S we have the inequality $u(S) \le l(S,\bar{S})$, then clearly the Gale–Hoffman inequality $d(S) \le y(S,\bar{S})$ is redundant.

Elimination by Lower Bounds

If $S \subset N$ and we have the inequality $y(S,\bar{S}) \ge d(S)$, further, $T \subset S$ and we have the inequality,

$$l(T,\bar{T}) - l(T) \ge u(S) - l(S,\bar{S}), \tag{2.7}$$

then the inequality,

$$y(T,\bar{T}) \ge d(T)$$

is redundant. In fact, if we subtract $l(S)$ on both sides of the inequality $y(S,\bar{S}) \ge d(S)$, we obtain:

$$y(S,\bar{S}) - l(S) \ge d(S) - l(S). \tag{2.8}$$

On the other hand, relation $T \subset S$ implies that:

$$d(S) - l(S) \ge d(T) - l(T). \tag{2.9}$$

Using (2.7), (2.8), and (2.9), we derive

$$\begin{gathered} y(T,\bar{T}) - l(T) \ge l(T,\bar{T}) - l(T) \\ \ge u(S,\bar{S}) - l(S) \ge y(S,\bar{S}) - l(S) \\ \ge d(S) - l(S) \ge d(T) - l(T) \end{gathered}$$

which implies $y(T, \bar{T}) \geq d(T)$.

The elimination works in such a way that we start by $S = N$, eliminate all inequalities corresponding to $T \subset S$ for which (2.7) is satisfied, then decrease the cardinality of S, etc.

Elimination by Linear Programming

Consider the inequalities that have not been eliminated. Let S_0 be one of them and $S_1, \ldots, S_m$ be the remaining ones. Then we formulate the LP:

$$\max \left\{ d(S_0) - y(\bar{S_0}, S) \right\}$$

$$\text{subject to}$$

$$d(S_i) - y(\bar{S_i}, S) \leq 0, \;\; i = 1, \ldots, m \tag{2.10}$$

$$l(i) \leq d(i) \leq u(i), \;\; i \in N$$

$$l(i,j) \leq y(i.j) \leq u(i,j), \;\; (i,j) \in N \times N.$$

The inequality $d(S_0) - y(\bar{S_0}, S) \leq 0$ is redundant if and only if the optimum value of problem (2.10) is nonpositive. Problem (2.10) takes a more convenient form if we subtract the lower bound from each variable. Let

$$\begin{aligned} x(i) &= d(i) - l(i), \;\; i \in N \\ x(S_i) &= d(S_i) - l(S_i), \;\; i = 1, \ldots, m \\ x(i,j) &= y(i,j) - l(i,j), \;\; (i,j) \in N \times N \\ x(\bar{S_i}, S_i) &= y(\bar{S_i}, S_i) - l(\bar{S_i}, S_i), \;\; i = 1, \ldots, m. \end{aligned}$$

Then problem (2.10) takes the form:

$$\max \left\{ x(S_0) - x(\bar{S_0}, S) + l(S_0) - l(\bar{S_0}, S) \right\}$$

$$\text{subject to}$$

$$x(S_i) - x(\bar{S_i}, S_i) \leq l(\bar{S_i}, S_i) - l(S_i), \;\; i = 1, \ldots, S \tag{2.11}$$

$$0 \leq x(i) \leq u(i) - l(i), \;\; i \in N$$

$$0 \leq x(i.j) \leq u(i,j) - l(i,j), \;\; (i,j) \in N \times N.$$

If we remove the constant term $l(S_0) - l(\bar{S_0}, S_0)$ from the objective function, then we can state that the inequality $d(S_0) - y(\bar{S_0}, S_0) \leq 0$ is redundant if the optimum value of problem (2.11) is smaller than or equal to $l(S_0) - l(\bar{S}_0, S_0)$.

In problem (2.11) we may have too many constraints; therefore it may be more convenient to work with the dual. Let $z(S_i)$, $w(i)$, $w(i,j)$ be the dual variables corresponding to the constraints involving $x(S_i)$, $x(i)$, $x(i,j)$, respectively. Then the dual of problem (2.11) can be written as follows:

$$\min\left\{\sum_{i=1}^{m}(l(\bar{S}_i, S_i) - l(S_i))z(S_i) + \sum_{i\in N}(u(i) - l(i))w(i) + \sum_{(i,j)\in NxN}(u(i,j) - l(i,j))w(i,j)\right\}$$

$$\text{subject to}$$

$$w(i) + \sum_{j:S_j\in i} z(S_j) \geq \; 1, i \in S_0 \tag{2.12}$$

$$w(i,k) + \sum_{\bar{S}_j\in i, S_j\in k} z(S_j) \geq -1, \;\; i \in \bar{S}_0, k \in S_0$$

$$z(S_i) \geq 0, \;\; i = 1, \ldots, m$$

$$w(i) \geq 0, \;\; i \in N$$

$$w(i,k) \geq 0, \;\; (i,k) \in N \times N.$$

If the optimum value is smaller than or equal to $l(S_0) - l(\bar{S}_0, S_0)$, then the inequality $d(S_0) - y(\bar{S}_0, S_0) \leq 0$ is redundant.

Note that we do not need to solve problem (2.12) optimally. In fact, if in the course of the optimization procedure we find that the current objective function value is less than or equal to $l(S_0) - l(\bar{S}_0, S_0)$, then we may stop and declare that $d(S_0) - y(\bar{S}_0, S_0) \leq 0$ is a redundant inequality. We may also simply try to find feasible solution to the constraints of problem (2.12) supplemented by the additional constraint that the objective function is less than or equal to $l(\bar{S}_0, S_0), l(S_0)$. In what follows the local demands $\xi_1, \ldots, \xi_n$ will be assumed to be random variables and the system demand will be the function $d(i) = \xi_i - x_i, \;\; i \in N$.

Example for Elimination Procedure

In this section we present an example to demonstrate the efficiency of elimination methods on Gale-Hoffman inequalities. Rather than 5-node example, we use a bigger size, 15-node network to show the impact of all elimination methods. The topology of this new network is taken from Prékopa and Boros (1991) where we have in mind power networks. We use the notation x_i for node capacities and y_{ij} for arc capacities.

Example We look at the 15-node network in the mentioned paper, where the network topology is depicted in Figure 2.2. It may represent an interconnected power system, where the nodes are the areas and the arcs the transmission system. At the nodes we have x_i generating capacities and on the arcs $y_{ij} = y_{ji}$ transmission capacities. At the nodes there are ξ_i random local demands and, by the use of them we define the system demand function $d(i) = \xi_i - x_i$, $1 \leq i \leq 15$. At each node we have both power generation capacity and demand. There may also be random deficiencies in the generation capacities but we have assumed that they are already combined with the demands. Flows on the arcs can take place in both directions. The arc capacities are the same in both directions, on each arc. In the paper by Prékopa and Boros (1991), the arc capacities are assumed to be constant. In this illustration we keep this assumption and indicate the numerical values of the arc capacities in Figure 2.2.

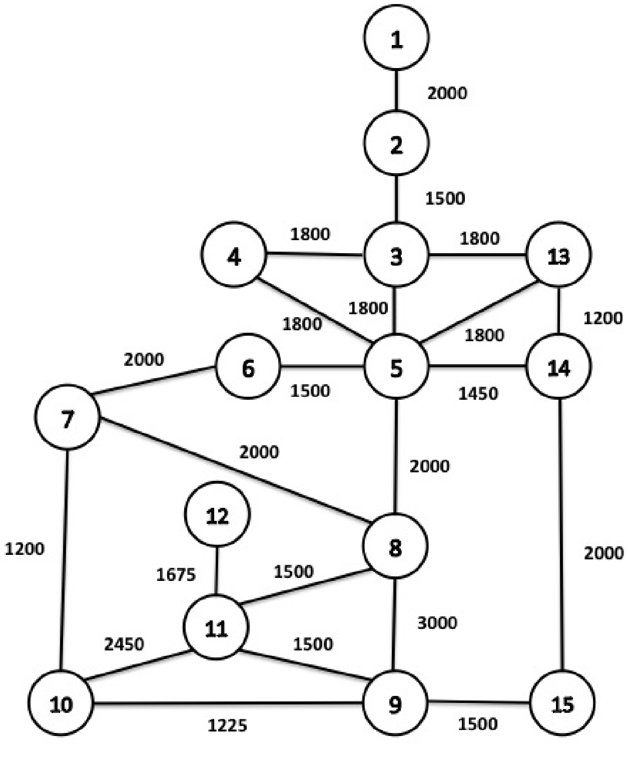


Figure 2.2: Fifteen-node network with arc capacities.

After the elimination procedure, we observed a minor error in Prékopa and Boros (1991) results, we were able to eliminate one more inequality and reduce the total number of inequalities to 12 from 13. Table 2.1 shows the number of eliminated inequalities by each elimination method. Table 2.2 contains the characteristics vectors of noneliminated inequalities.

Table 2.1: Number of eliminated inequalities by each elimination method

Number of original Gale–Hoffman inequalities	**32,767**
By network topology	28,655
By upper bounds	2,588
By lower bounds	1,288
By linear programming	224
Remaining inequalities	**12**

Table 2.2: Characteristics vectors of noneliminated inequalities for 15-node network

Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
3	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1
4	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1
5	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1
8	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0
9	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0
10	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

2.2 *p*-level Efficient Points

The concept of a p-level efficient point or briefly p-efficient point was introduced in Prékopa (1990). It was further studied and used to solve probabilistic constrained stochastic programming problem with discrete random variables by Prékopa, Vizvári, Badics (1998). The new results in that paper include an algorithmic enumeration method of the p-efficient points. Another algorithm is proposed by Boros, Elbassioni, Gurvich, Khachiyan and Makino (2003). Dentcheva, Prékopa and Ruszczyński (2000) gave another solution method for the same problem that generates the p-efficient points simultaneously with the solution algorithm if the random variables are independent.

The notion of Pareto efficient points usually comes in mind in connection with the definition of p-efficient points. Pareto efficient points are the minimal points of compact sets in linear spaces with convex orderings. It is a concept in economics with applications in engineering and other sciences. In economics, Pareto efficiency is interpreted as a minimal notion of efficiency and does not necessarily result in overall optimization of the society. In optimization literature, great deal of attention has been paid to Pareto efficient points. Stadler (1974) provides a good survey until 1960's and Penot (1978) up to 1978. The work by Borwein (1983) provides the below definition for Pareto efficient points.

Definition We define x to be Pareto efficient (minimal) for a set C in X, with respect to a convex cone K, if $x \in C$ and

$$c \in C, \;\; c \leq_K x \Rightarrow c \sim_K x, \tag{2.13}$$

and write $x \in E(C; K)$.

Before describing the differences between Pareto and p-efficient points, first, for the reader's convenience, we recall the definition of a p-efficient point.

Let $\xi = (\xi_1, \ldots, \xi_n)$ be a discrete random vector, where the supports of the random variables $\xi_1, \ldots, \xi_n$ are the finite sets $Z_1, \ldots, Z_n$, respectively. Introduce the notation:

$$Z = Z_1 \times Z_2 \times \cdots \times Z_n. \tag{2.14}$$

Let $Z_i = \{z_{i1}, \dots, z_{ik_i}\}$, where $z_{i1} \leq \cdots \leq z_{ik_i}$, $i = 1, \dots, n$.

Definition A point $z \in Z$ is a p-efficient point $(0 \leq p \leq 1)$ of the probability distribution function F of ξ if $F(z) \geq p$ and there is no $y < z$ such that $F(y) \geq p$. ($y < z$ means $y \leq z$, $y \neq z$).

Obviously, there is a connection between the definition of a Pareto efficient point and a p-efficient point but not very strong. Given the set $\{z|F(z) \geq p\}$ is C in Equation (2.13), then its Pareto efficient points are our p-efficient points. However, the theory of the two applications are already different. Pareto efficient points belong to sets while p-efficient points belong to probability distributions. Given a probability distribution function, still there is a variety of p-efficient points because p can be chosen infinitely many ways. We want equivalent formulations of probabilistic constrained problems, want to determine the p-efficient points algorithmically and generate them simultaneously in the course of a procedure. In addition, we want to investigate their convexity properties. These are not present in Pareto efficient point theory. Moreover, the distribution can be given in many ways. Our principle notion is the distribution, not simply a set. A lot of hard and novel problems come up in connection with p-efficient points. Prékopa (2010) gives a detailed use of p-efficient points that is applied in finance, water, and power engineering, with multivariate probability distribution where Pareto efficient points cannot be used to represent the set of feasible points.

In this chapter, our optimization problem is of probabilistic constrained type. If, for example, the arc capacities in a network are constants but the demands are random, then the $d(S)$ symbols on the left hand sides in (2.3) are random variables while the right hand sides are constants and a probabilistic constraint in an optimization problem may take the form:

$$P\left(d(S) \leq y(S, \bar{S}), S \subset N\right) \geq p. \tag{2.15}$$

The inequalities in (2.3), however, include a number of redundant ones that first we eliminate and it is sufficient to impose probabilistic constraint on those inequalities that are not deleted in the course of the eliminations. Still, in many cases quite a few inequalities remain after the elimination (as it can be seen in example 1 of section 3; see Appendix 2.9), hence it is reasonable to look for further simplifications in the enumeration of the set of p-efficient points. Fortunately, the random variables $d(S)$ in (2.3) allow for such simplification. We formulate it in more general terms.

The next theorem, Prékopa and Unuvar (2010), tells us that if we know the p-efficient points of a random vector ξ, then, under some conditions, we can at once obtain the p-efficient points of a random vector consisting of all components of ξ and some others that are linear combinations of ξ with nonnegative coefficients.

Theorem 2.2.1 *Let $\xi \in Z$ be a random vector and $B \geq 0$ a matrix with n columns and an arbitrary number of rows such that in each row there is at least one positive element. Suppose that the p-efficient points of ξ are $z^{(1)}, \ldots, z^{(M)}$ and the following condition holds for every $i = 1, \ldots, M$: $P\left(\{z \in Z \mid z \leq z^{(i)}\} \backslash \{z^{(i)}\}\right) < p$. Then the p-efficient points of the random vector $\left(\begin{smallmatrix} \xi \\ B\xi \end{smallmatrix}\right)$ are:*

$$\left(\begin{array}{c} z^{(1)} \\ \\ Bz^{(1)} \end{array}\right), \ldots, \left(\begin{array}{c} z^{(M)} \\ \\ Bz^{(M)} \end{array}\right). \tag{2.16}$$

Note that $P\left(\{z \mid z \leq z^{(i)}\}\right) \geq p$ for every $i = 1, \ldots, M$, hence the condition in Theorem 2.2.1 implies that every p-efficient vector has positive probability.

Proof For every i, $1 \leq i \leq M$, the inequality $B\xi \leq Bz^{(i)}$ is a consequence of the inequality $\xi \leq z^{(i)}$. It follows that

$$P\left(\left(\begin{array}{c} \xi \\ \\ B\xi \end{array}\right) \leq \left(\begin{array}{c} z^{(i)} \\ \\ Bz^{(i)} \end{array}\right)\right) \geq p, \;\; i = 1, \ldots, M. \tag{2.17}$$

We have to show that if we decrease the value of at least one of the components of $\left(\begin{smallmatrix} z^{(i)} \\ Bz^{(i)} \end{smallmatrix}\right)$ within the support of $\left(\begin{smallmatrix} \xi \\ B\xi \end{smallmatrix}\right)$ then the inequality (2.17) is no longer valid for given i. Obviously, if the decrease happens among the first n components, $z^{(i)}$ decreases to $w^{(i)}$, then $P(\xi \leq w^{(i)}) < p$ and also $P(\xi \leq w^{(i)},\ B\xi \leq z^{(i)}) < p$. If, on the other hand, one component of $Bz^{(i)}$ decreases to $Bw^{(i)}$, then, because B has at least one positive entry in each row, the point $z^{(i)}$ is excluded. In view of our assumption the probability decreases to $P(\xi \leq z^{(i)}, \xi \neq z^{(i)}, B\xi \leq Bw^{(i)}) < p$. □

Remark While in practice most frequently we have $P(\xi = z^{(i)}) > 0$, $i = 1, \ldots, M$, the condition that $P(\xi \leq z^{(i)},\ \xi \neq z^{(i)}) < p$ may not hold. Still, we advise to use the set of vectors (2.16) as an approximation of the set of p-efficient points of $\left(\begin{smallmatrix} \xi \\ B\xi \end{smallmatrix}\right)$. The reason is that the probability distribution of ξ can slightly be perturbed (at least in most practical problems) in such a way that the p-efficient points of the perturbed distribution are those in (2.16). In fact, if we add to each $p_{jk} = P(\xi = z_{jk})$, $z_{jk} \in \bigcup_{i=1}^{M} (z^{(i)} + R_-^n) \cap Z$ a value $\varepsilon_{jk} \geq 0$ and subtract $\varepsilon_{lt} \geq 0$ from each $p_{lt} = P(\xi = z_{lt})$, $z_{lt} \notin \bigcup_{i=1}^{M} (z^{(i)} + R_-^n) \cap Z$, keeping the probabilities nonnegative and their sum equal to 1, then under suitable choices of the ε's we may accomplish the task.

An important special case of Theorem 2.2.1 is the following. Let $I_1, \ldots, I_l$ be nonempty subsets of the set $\{1, \ldots, l\}$ and consider the random vector

$$\left(\xi_1, \ldots, \xi_n, \sum_{i \in I_1} \xi_i, \ldots, \sum_{i \in I_l} \xi_i \right)^T . \tag{2.18}$$

Theorem 2.2.2 *If the p-efficient points of ξ are $\{z^{(1)}, \ldots, z^{(M)}\}$ and $P(\xi \leq z^{(k)},\ \xi \neq z^{(k)}) < p$, $k = 1, \ldots, M$, then the p-efficient points of the random vector* (2.18) *are:*

$$\left(z_1^{(k)}, \ldots, z_n^{(k)}, \sum_{i \in I_1} z_i^{(k)}, \ldots, \sum_{i \in I_l} z_i^{(k)} \right)^T, \quad k = 1, \ldots, M. \tag{2.19}$$

Remark The condition in Theorem 2.2.2 holds true if $P(\xi \leq z^{(k)}) = p$ and $P(\xi = z^{(k)}) > 0,\ \ k = 1, \ldots, M$.

Example Let $\xi = (\xi_1,\ \xi_2,\ \xi_3,\ \xi_4)^T$ and consider the random vector:

$$(\xi_1,\ \ \xi_2,\ \ \xi_3,\ \ \xi_4,\ \ \xi_1 + \xi_2,\ \ \xi_1 + \xi_3 + \xi_4,\ \ \xi_2 + \xi_4)^T. \tag{2.20}$$

If the condition mentioned in Theorem 2.2.2 holds true, then the p-efficient points of the random vector (2.20) are:

$$\begin{pmatrix} z_1^{(k)} \\ z_2^{(k)} \\ z_3^{(k)} \\ z_4^{(k)} \\ z_1^{(k)} + z_2^{(k)} \\ z_1^{(k)} + z_3^{(k)} + z_4^{(k)} \\ z_2^{(k)} + z_4^{(k)} \end{pmatrix}, \quad k = 1, \ldots, M.$$

If each random variable is uniformly distributed in the same support set $\{1, 2, 3, 4, 5\}$ and $p = 0.8$ then the p-efficient points of ξ are:

$$\begin{pmatrix} 4 \\ 5 \\ 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 4 \\ 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \\ 4 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \\ 5 \\ 4 \end{pmatrix}.$$

The value of the joint c.d.f. is equal to 0.8 at each of these points, hence the condition mentioned in Theorem 2.2.2 is satisfied. It follows that the p-efficient points of the random vector (2.20) are:

$$\begin{pmatrix} 4 \\ 5 \\ 5 \\ 5 \\ 9 \\ 14 \\ 10 \end{pmatrix}, \begin{pmatrix} 5 \\ 4 \\ 5 \\ 5 \\ 9 \\ 15 \\ 9 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \\ 4 \\ 5 \\ 10 \\ 14 \\ 10 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \\ 5 \\ 4 \\ 10 \\ 14 \\ 9 \end{pmatrix}.$$

Example We show that the statement of Theorem 2.2.1, may not be valid without the assumption. Consider the random vector $\xi = (\xi_1, \xi_2)^T$ and suppose that ξ has the following probability distribution: $P(\xi = (0, 0)) = 0.4$, $P(\xi = (0, 1)) = P(\xi = (1, 0)) = 0.2$, $P(\xi = (0, 2)) = P(\xi = (2, 0)) = 0.1$.

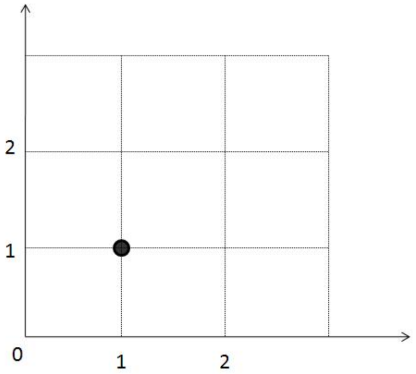


Figure 2.3: The condition in Theorem 2.2.2 is not satisfied for the marked point.

The marked point $(1, 1)$ in Figure 2.3 is the only 0.8-efficient point of ξ and it has 0 probability. If we consider the random vector $\xi = (\xi_1,\ \xi_2,\ \xi_1 + \xi_2)^T$, then we can see that its only 0.8-efficient point is (1, 1, 1) and not (1, 1, 2) as it would be the case under the condition of Theorem 2.2.1.

For the sake of completeness we present an algorithm that generates all p-efficient points of the random vector (2.18). The p-efficient points of (2.18) will be called network p-efficient points.

If for some k the condition of Theorem 2.2.1 is satisfied, i.e., $P(\xi \le z^{(k)},\ \xi \ne z^{(k)}) < p$, then the vector (2.19) is a network p-efficient point. If this is not the case, then we define the set

$$K_h^{(k)} = \left\{ z \in Z \mid z \le z^{(k)},\ \sum_{i \in I_j} (z_i^{(k)} - z_i) \ge h_j,\ j = 1, \dots, t \right\}$$
$$h_j \text{ integer},\ 0 \le h_j \le |I_j|, \quad j = 1, \dots, t; \quad k = 1, \dots, M. \tag{2.21}$$

For a given k, for which $P(\xi \le z^{(k)},\ \xi \ne z^{(k)}) < p$, we want to find all $h^* = (h_1^*, \dots, h_t^*)$ vectors that satisfy the condition in the second line of (2.21) and

$$F(z^{(k)}) - P(K_h^{(k)}) \ge p$$
$$F(z^{(k)}) - P(K_{h^*}^{(k)}) < p \tag{2.22}$$

for $h \leq h^*$. If we rewrite (2.22) as follows:

$$\begin{aligned} P(K_h^{(k)}) &\leq F(z^{(k)}) - p \\ P(K_{h^*}^{(k)}) &> F(z^{(k)}) - p \end{aligned} \tag{2.23}$$

then the problem is to find all $F(z^{(k)}) - p$-efficient points in the integer lattice of the cube $\{h \mid h_j \leq |I_j|,\ j = 1, \ldots, t\}$. The efficiency is now defined in the sense of (2.23). To find all efficient h^*, in the sense of (2.23), any of the existing algorithms, to find all p-efficient points, can be used with obvious modification.

If $M = 1$, i.e., there is only one p-efficient point of ξ and the condition of Theorem 2.2.1 is satisfied, then we are done, the corresponding vector in (2.20) is the only network p-efficient point. If $M = 1$ and the condition of Theorem 2.2.1 is not satisfied, then we generate all h^* and the obtained $F(z^{(1)}) - p$-efficient points simultaneously provide us with the set of all network p-efficient points. If $M > 1$ and there is at least one k $(1 \leq k \leq M)$ for which the condition of Theorem 2.2.1 is not satisfied, then we use an algorithm to generate the set of network p-efficient points. Note that it is not enough to find the $F(z^{(k)}) - p$-efficient points for every k for which $P(\xi \leq z^{(k)},\ \xi \neq z^{(k)}) < p$, because some of them may be dominated by others, corresponding to different k values and therefore an elimination procedure has to be included.

2.2.1 Algorithm to Find all Network p-efficient Points

- *Step 1.* Find all p-efficient points of ξ and designate them by $z^{(k)}$, $k = 1, \ldots, M$.
- *Step 2.* Initialize $J = 0$ and let $H^{(J)}$ be the current set of network p-efficient points.
- *Step 3.* Set $J = J + 1$. If $J > M + 1$, then go to Step 6. Otherwise go to Step 4.
- *Step 4.* If for $z^{(J)}$ we have $P(\xi \leq z^{(J)},\ \xi \neq z^{(J)}) < p$, then include $z^{(J)}$ into $H^{(J)}$ and go to Step 6. Otherwise go to Step 5.
- *Step 5.* Using $z^{(J)}$, generate all $F(z^J) - p$-efficient points and form the new (2.20)-type vectors. Eliminate those which are dominated by vectors in $H^{(J)}$. Include into $H^{(J)}$ the remaining ones. Go to Step 3.

- *Step 6.* Stop, $H^{(M)}$ is the set of all network p-efficient points.

A numerical example at the end of this chapter contains the results of above algorithm.

2.3 Static Stochastic Network Design Problem Using Probabilistic Constraint

Our stochastic network design problem can be formulated in following way.

Corresponding to each node i in the network a capacity x_i and a random demand ξ_i are associated. Bearing in mind the application to the interconnected power systems, we call x_i as generating capacity and the capacity y_{ij}, corresponding to arc (i, j) as transmission capacity. If the system demand $\xi_i - x_i$ at node i is positive, then the local generating capacity is not enough to meet the local demand ξ_i and assistance is needed from other nodes. If, however, $\xi_i - x_i < 0$, then there is surplus generating capacity at node i and the node can assist others.

The unknown decision variables in our optimization problem are the node capacities $x_i, \ i \in N$ and the arc capacities $y_{ij}, \ (i, j) \in N \times N$. The static formulation of the problem is the following:

$$\begin{gathered}
\min \left\{ \sum_{i \in N} c_i(x_i) + \sum_{(i,j) \in N \times N} c_{ij}(y_{ij}) \right\} \\
\text{subject to} \\
P(d(S) \leq y((S, \bar{S}), S \subset N)) \geq p \qquad (2.24) \\
A_1 x + A_2 y \geq b \\
x \geq 0, y \geq 0.
\end{gathered}$$

The constraint $A_1x+A_2y \geq b$ may simply mean lower and upper bounds for the decision variables $x_i, \ y_{ij}$. In that case, we write them up as follows:

$$\begin{gathered}
l_i \leq x_i \leq u_i, \ \ i \in N \\
l_{ij} \leq y_{ij} \leq u_{ij}, \ \ (i, j) \in N \times N \quad . \qquad (2.25)
\end{gathered}$$

Illustration of model 2.24 on 5 node example in Figure 2.1

To demonstrate how to design a network problem as a probabilistic constrained stochastic programming problem, we worked on the 5-node example in Figure 2.1. Below we constructed the model with all the feasibility constraints (before elimination) and some deterministic constraints:

$$\min \left\{ \textstyle\sum_{i=1}^{5} c_i(x_i) + \sum_{j=1}^{5} K_j(y_j) \right\}$$

$$\text{subject to}$$

$$P \left(\begin{array}{ccc}
\xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & 0 \\
\xi_1 - x_1 & \leq & y_1 \\
\xi_2 - x_2 & \leq & y_1 + y_2 + y_3 + y_4 \\
\xi_3 - x_3 & \leq & y_3 \\
\xi_4 - x_4 & \leq & y_4 + y_5 \\
\xi_5 - x_5 & \leq & y_2 + y_5 \\
\xi_1 - x_1 + \xi_2 - x_2 & \leq & y_2 + y_3 + y_4 \\
\xi_1 - x_1 + \xi_3 - x_3 & \leq & y_1 + y_3 \\
\xi_1 - x_1 + \xi_4 - x_4 & \leq & y_1 + y_4 + y_5 \\
\xi_1 - x_1 + \xi_5 - x_5 & \leq & y_1 + y_2 + y_5 \\
\xi_2 - x_2 + \xi_3 - x_3 & \leq & y_1 + y_2 + y_4 \\
\xi_2 - x_2 + \xi_4 - x_4 & \leq & y_1 + y_2 + y_3 + y_5 \\
\xi_2 - x_2 + \xi_5 - x_5 & \leq & y_1 + y_3 + y_4 + y_5 \\
\xi_3 - x_3 + \xi_4 - x_4 & \leq & y_3 + y_4 + y_5 \\
\xi_3 - x_3 + \xi_5 - x_5 & \leq & y_2 + y_3 + y_5 \\
\xi_4 - x_4 + \xi_5 - x_5 & \leq & y_2 + y_4 \\
\xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 & \leq & y_2 + y_4 \\
\xi_1 - x_1 + \xi_2 - x_2 + \xi_4 - x_4 & \leq & y_2 + y_3 + y_5 \\
\xi_1 - x_1 + \xi_2 - x_2 + \xi_5 - x_5 & \leq & y_3 + y_4 + y_5 \\
\xi_1 - x_1 + \xi_3 - x_3 + \xi_4 - x_4 & \leq & y_1 + y_3 + y_4 + y_5 \\
\xi_1 - x_1 + \xi_3 - x_3 + \xi_5 - x_5 & \leq & y_1 + y_2 + y_3 + y_5 \\
\xi_1 - x_1 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_1 + y_2 + y_4
\end{array} \right) \geq p$$

cont'

$$P\left(\begin{array}{ccc} \xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 & \leq & y_1 + y_2 + y_5 \\ \xi_2 - x_2 + \xi_3 - x_3 + \xi_5 - x_5 & \leq & y_1 + y_4 + y_5 \\ \xi_2 - x_2 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_1 + y_3 \\ \xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_2 + y_3 + y_4 \\ \xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 & \leq & y_2 + y_5 \\ \xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 + \xi_5 - x_5 & \leq & y_4 + y_5 \\ \xi_1 - x_1 + \xi_2 - x_2 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_3 \\ \xi_1 - x_1 + \xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_1 + y_2 + y_3 + y_4 \\ \xi_2 - x_2 + \xi_3 - x_3 + \xi_4 - x_4 + \xi_5 - x_5 & \leq & y_1 \end{array}\right) \geq p$$

$$A_1x + A_2y \geq b$$

$$x \geq 0, y \geq 0 \quad . \tag{2.26}$$

where c_i is the cost function for network capacities and K_j is the cost function for the arc transmission capacities in Figure 2.1. Moreover, let the $A_1x + A_2y \geq b$ constraint represent an upper bound on the capacities such that:

$$l_i \leq x_i \leq u_i, \ \ i = 1, 2, 3, 4, 5$$

$$l_j \leq y_j \leq u_j, \ \ j = 1, 2, 3, 4, 5$$

Through the rest of the chapter, we will be illustrating the models and methods by using this 5 node example. Suppose the deterministic constraints are of the form $A_1x + A_2y \geq b$ for compatibility with the closed-form of the model.

The static stochastic programming problems can be of probabilistic constrained, recourse (penalty) or hybrid type. Instead of problem (2.24) we may easily construct a hybrid type model, where the expectation of the measure of violation of the stochastic constraints is incorporated into the objective function. Since we have discrete random variables, the inclusion of penalty terms into the objective function does not change the type of the problem. The objective function is extended by linear terms and new linear constraints are incorporated (see Prékopa, 1995, Chapter 9). However, our main concern is the handling of the probabilistic constraint, therefore we disregard the formulation of a hybrid model. In the probabilistic constraint of problem (2.24) we have the Gale–Hoffman inequalities: $d(S) \leq y(S, \bar{S}),\ S \subset N$. If we apply the elimination procedure described in the introduction, then we can significantly reduce the number of them. After the elimination the problem takes the form:

$$\min \left\{ \sum_{i \in N} c_i(x_i) + \sum_{(i,j) \in NxN} c_{ij}(y_{ij}) \right\}$$

$$\text{subject to}$$

$$P \left(\begin{array}{c} \xi_k \leq x_k + \sum_{(j,k) \in N \times N} y_{jk}, \;\; k = 1, \ldots, n \\ \sum_{k \in I_j} \xi_k \leq \sum_{k \in I_j} x_k + \sum_{k \in I_j} \sum_{(j,k) \in N \times N} y_{jk}, \;\; j = 1, \ldots, t \end{array} \right) \geq p \quad (2.27)$$

$$A_1 x + A_2 y \geq b$$

$$x \geq 0, y \geq 0,$$

where $n = |N|$. Inside the parentheses in the probabilistic constraint the non-eliminated feasibility inequalities are listed. It is essential, from our point of view, that all individual stochastic constraints, i.e., those that contain a single component of the random vector $\xi = (\xi_1, \ldots, \xi_n)^T$ appear among the stochastic constraints. We need them in order to be able to apply the methodology of Theorem 2.2.1. The requirement that the individual stochastic constraints should not be eliminated is not a restriction, however, needed from the practical point of view.

Illustration of model 2.27 on 5 node example in Figure 2.1

After the network topology elimination, our 5-node example becomes;

$$\min \left\{ \textstyle\sum_{i=1}^{5} c_i(x_i) + \sum_{j=1}^{5} K_j(y_j) \right\}$$

$$\text{subject to}$$

$$P \left(\begin{array}{ccc}
\xi_1 & \leq & x_1 + y_1 \\
\xi_2 & \leq & x_2 + y_1 + y_2 + y_3 + y_4 \\
\xi_3 & \leq & x_3 + y_3 \\
\xi_4 & \leq & x_4 + y_4 + y_5 \\
\xi_5 & \leq & x_5 + y_2 + y_5 \\
\xi_1 + \xi_2 & \leq & x_1 + x_2 + y_2 + y_3 + y_4 \\
\xi_2 + \xi_3 & \leq & x_1 + x_3 + y_1 + y_2 + y_4 \\
\xi_2 + \xi_4 & \leq & x_2 + x_4 + y_1 + y_2 + y_3 + y_5 \\
\xi_2 + \xi_5 & \leq & x_2 + x_5 + y_1 + y_3 + y_4 + y_5 \\
\xi_4 + \xi_5 & \leq & x_4 + x_5 + y_2 + y_4 \\
\xi_1 + \xi_2 + \xi_3 & \leq & x_1 + x_2 + x_3 + y_2 + y_4 \\
\xi_1 + \xi_2 + \xi_4 & \leq & x_1 + x_2 + x_4 + y_2 + y_3 + y_5 \\
\xi_1 + \xi_2 + \xi_5 & \leq & x_1 + x_2 + x_5 + y_3 + y_4 + y_5 \\
\xi_2 + \xi_3 + \xi_4 & \leq & x_2 + x_3 + x_4 + y_1 + y_2 + y_5 \\
\xi_2 + \xi_3 + \xi_5 & \leq & x_2 + x_3 + x_5 + y_1 + y_4 + y_5 \\
\xi_2 + \xi_4 + \xi_5 & \leq & x_2 + x_4 + x_5 + y_1 + y_3 \\
\xi_3 + \xi_4 + \xi_5 & \leq & x_3 + x_4 + x_5 + y_2 + y_3 + y_4 \\
\xi_1 + \xi_2 + \xi_3 + \xi_4 & \leq & x_1 + x_2 + x_3 + x_4 + y_2 + y_5 \\
\xi_1 + \xi_2 + \xi_3 + \xi_5 & \leq & x_1 + x_2 + x_3 + x_5 + y_4 + y_5 \\
\xi_1 + \xi_2 + \xi_4 + \xi_5 & \leq & x_1 + x_2 + x_4 + x_5 + y_3 \\
\xi_1 + \xi_3 + \xi_4 + \xi_5 & \leq & x_1 + x_3 + x_4 + x_5 + y_1 + y_2 + y_3 + y_4 \\
\xi_2 + \xi_3 + \xi_4 + \xi_5 & \leq & x_1 + x_3 + x_4 + x_5 + y_1 \\
\xi_1 + \xi_2 + \xi_3 + \xi_4 + \xi_5 & \leq & x_1 + x_2 + x_3 + x_4 + x_5
\end{array} \right) \geq p$$

$$A_1 x + A_2 y \geq b$$

$$x \geq 0, y \geq 0 \quad . \tag{2.28}$$

In what follows we will be looking at the random vector:

$$\begin{pmatrix} \xi \\ \sum_{k\in I_j} \xi_k \\ j=1,\ldots,t \end{pmatrix} \tag{2.29}$$

Illustration of random vector 2.29 on 5 node example in Figure 2.1

$$\begin{pmatrix}
\xi_1 & & & & \\
& \xi_2 & & & \\
& & \xi_3 & & \\
& & & \xi_4 & \\
& & & & \xi_5 \\
\xi_1 & +\xi_2 & & & \\
& \xi_2 & +\xi_3 & & \\
& \xi_2 & & +\xi_4 & \\
& \xi_2 & & & +\xi_5 \\
& & & \xi_4 & +\xi_5 \\
\xi_1 & +\xi_2 & +\xi_3 & & \\
\xi_1 & +\xi_2 & & +\xi_4 & \\
\xi_1 & +\xi_2 & & & +\xi_5 \\
& \xi_2 & +\xi_3 & +\xi_4 & \\
& \xi_2 & +\xi_3 & & +\xi_5 \\
& \xi_2 & & +\xi_4 & +\xi_5 \\
& & \xi_3 & +\xi_4 & +\xi_5 \\
\xi_1 & +\xi_2 & +\xi_3 & +\xi_4 & \\
\xi_1 & +\xi_2 & +\xi_3 & & +\xi_5 \\
\xi_1 & +\xi_2 & & +\xi_4 & +\xi_5 \\
\xi_1 & & +\xi_3 & +\xi_4 & +\xi_5 \\
& \xi_2 & +\xi_3 & +\xi_4 & +\xi_5 \\
\xi_1 & +\xi_2 & +\xi_3 & +\xi_4 & +\xi_5
\end{pmatrix} \tag{2.30}$$

We assume that $\{z^{(1)},\dots,z^{(M)}\}$ is the set of p-efficient points of ξ. Then, we use the vectors

$$\begin{pmatrix} z^{(i)} \\ \sum_{k\in I_j} z_k^{(i)} \\ j=1,\dots,t \end{pmatrix},\ \ i=1,\dots,M \tag{2.31}$$

as the correct or approximate set of network p-efficient points. With these vectors our network design problem can be formulated in the following way:

$$\min\left\{\sum_{i\in N} c_i(x_i)+\sum_{(i,j)\in N\times N} c_{ij}(y_{ij})\right\}$$

subject to the constraints that for at least one $i=1,\dots,M$ we have

$$x_k+\sum_{(j,k)\in N\times N} y_{jk}\geq z_k^{(i)},\ \ k=1,\dots,n$$

$$\sum_{k\in I_l} x_k+\sum_{k\in I_l}\sum_{(j,k)\in N\times N} y_{jk}\geq \sum_{k\in I_l} z_k^{(i)},0\ \ l=1,\dots,t \tag{2.32}$$

and

$$A_1x+A_2y\geq b$$

$$x\geq 0,\ \ y\geq 0.$$

Problem (2.32) is a disjunctive optimization problem that we relax by a standard convexification procedure: we take the convex combination of the upper $M(n+t)$ inequalities. The new problem is:

$$\min\left\{\sum_{i\in N} c_i(x_i)+\sum_{(i,j)\in N\times N} c_{ij}(y_{ij})\right\}$$

subject to

$$\begin{pmatrix} x_k+\sum_{(j,k)\in N\times N} y_{jk}\geq z_k^{(i)},\ \ k=1,\dots,n \\ \sum_{k\in I_l} x_k+\sum_{k\in I_l}\sum_{(j,k)\in N\times N} y_{jk}\geq\sum_{k\in I_l} z_k^{(i)},\ \ l=1,\dots,t \end{pmatrix}\geq\sum_{i=1}^{M}\lambda_i\begin{pmatrix} z_k^{(i)},\ \ k=1,\dots,n \\ \sum_{k\in I_l} z_k^{(i)},\ \ l=1,\dots,t \end{pmatrix}$$

$$A_1x+A_2y\geq b \tag{2.33}$$

$$\sum_{i=1}^{M}\lambda_i=1$$

$$x\geq 0,\ \ y\geq 0,\ \ \lambda\geq 0.$$

In what follows, we assume that the cost functions $c_k(x_k),\ \ c_{jk}(y_{jk})$ are linear. If these functions are nonlinear but convex, then we approximate them by piecewise linear functions and again the problem is an LP.

Set of p-efficient points of the 5 node example in Figure 2.1

Let the cumulative probability distribution and associated values for the demand values be in Table 2.3 and let the corresponding set of all p-efficient points for the individual random variables be: $z^{(1)} = \begin{pmatrix} 10 \\ 9 \\ 12 \\ 9 \\ 15 \end{pmatrix}$, $z^{(2)} = \begin{pmatrix} 15 \\ 9 \\ 8 \\ 9 \\ 15 \end{pmatrix}$ $z^{(3)} = \begin{pmatrix} 15 \\ 9 \\ 12 \\ 6 \\ 15 \end{pmatrix}$, $z^{(4)} = \begin{pmatrix} 15 \\ 9 \\ 12 \\ 9 \\ 10 \end{pmatrix}$, $z^{(5)} = \begin{pmatrix} 15 \\ 9 \\ 12 \\ 6 \\ 10 \end{pmatrix}$.

Table 2.3: Cumulative Probability Distribution of ξ_i for 5 node example in Figure 2.1

ξ_1		ξ_2		ξ_3		ξ_4		ξ_5	
Value	P(Value)	Value	P(Value)	Value	P(Value)	Value	P(Value)	Value	P(Value)
5	0.2	3	0.3	4	0.2	3	0.3	5	0.1
10	0.8	6	0.7	8	0.8	6	0.9	10	0.9
15	1	9	1	12	1	9	1	15	1

Let us demonstrate how to construct the rest of the p-efficient points for $z^{(1)}$ of ξ_i by referring to vector 2.30 and vector 2.31.

$$z_{all}^{(1)} = \left(\begin{array}{ccccc}
\xi_1 & & & & \\
 & \xi_2 & & & \\
 & & \xi_3 & & \\
 & & & \xi_4 & \\
 & & & & \xi_5 \\
\xi_1 & +\xi_2 & & & \\
 & \xi_2 & +\xi_3 & & \\
 & \xi_2 & & +\xi_4 & \\
 & \xi_2 & & & +\xi_5 \\
 & & & \xi_4 & +\xi_5 \\
\xi_1 & +\xi_2 & +\xi_3 & & \\
\xi_1 & +\xi_2 & & +\xi_4 & \\
\xi_1 & +\xi_2 & & & +\xi_5 \\
 & \xi_2 & +\xi_3 & +\xi_4 & \\
 & \xi_2 & +\xi_3 & & +\xi_5 \\
 & \xi_2 & & +\xi_4 & +\xi_5 \\
 & & \xi_3 & +\xi_4 & +\xi_5 \\
\xi_1 & +\xi_2 & +\xi_3 & +\xi_4 & \\
\xi_1 & +\xi_2 & +\xi_3 & & +\xi_5 \\
\xi_1 & +\xi_2 & & +\xi_4 & +\xi_5 \\
\xi_1 & & +\xi_3 & +\xi_4 & +\xi_5 \\
 & \xi_2 & +\xi_3 & +\xi_4 & +\xi_5 \\
\xi_1 & +\xi_2 & +\xi_3 & +\xi_4 & +\xi_5
\end{array}\right) = \begin{pmatrix} 10 \\ 9 \\ 12 \\ 9 \\ 15 \\ 10+9 \\ 9+12 \\ 9+9 \\ 9+15 \\ 9+15 \\ 10+9+12 \\ 10+9+9 \\ 10+9+15 \\ 9+12+9 \\ 9+12+15 \\ 9+9+15 \\ 12+9+15 \\ 10+9+12+9 \\ 10+9+12+15 \\ 10+9+9+15 \\ 10+12+9+15 \\ 9+12+9+15 \\ 10+9+12+9+15 \end{pmatrix} = \begin{pmatrix} 10 \\ 9 \\ 12 \\ 9 \\ 15 \\ 19 \\ 21 \\ 18 \\ 24 \\ 24 \\ 31 \\ 28 \\ 34 \\ 30 \\ 36 \\ 33 \\ 36 \\ 40 \\ 46 \\ 43 \\ 46 \\ 45 \\ 55 \end{pmatrix}$$

Illustration of model 2.32 on 5 node example in Figure 2.1

Let us demonstrate how to create model 2.32 with the first p-efficient point below:

$$\min \left\{ \sum_{i=1}^{5} c_i(x_i) + \sum_{j=1}^{5} K_j(y_j) \right\}$$

subject to

$$\begin{array}{ccc}
\xi_1 & \leq & x_1 + y_1 \\
\xi_2 & \leq & x_2 + y_1 + y_2 + y_3 + y_4 \\
\xi_3 & \leq & x_3 + y_3 \\
\xi_4 & \leq & x_4 + y_4 + y_5 \\
\xi_5 & \leq & x_5 + y_2 + y_5 \\
\xi_1 + \xi_2 & \leq & x_1 + x_2 + y_2 + y_3 + y_4 \\
\xi_2 + \xi_3 & \leq & x_1 + x_3 + y_1 + y_2 + y_4 \\
\xi_2 + \xi_4 & \leq & x_2 + x_4 + y_1 + y_2 + y_3 + y_5 \\
\xi_2 + \xi_5 & \leq & x_2 + x_5 + y_1 + y_3 + y_4 + y_5 \\
\xi_4 + \xi_5 & \leq & x_4 + x_5 + y_2 + y_4 \\
\xi_1 + \xi_2 + \xi_3 & \leq & x_1 + x_2 + x_3 + y_2 + y_4 \\
\xi_1 + \xi_2 + \xi_4 & \leq & x_1 + x_2 + x_4 + y_2 + y_3 + y_5 \\
\xi_1 + \xi_2 + \xi_5 & \leq & x_1 + x_2 + x_5 + y_3 + y_4 + y_5 \\
\xi_2 + \xi_3 + \xi_4 & \leq & x_2 + x_3 + x_4 + y_1 + y_2 + y_5 \\
\xi_2 + \xi_3 + \xi_5 & \leq & x_2 + x_3 + x_5 + y_1 + y_4 + y_5 \\
\xi_2 + \xi_4 + \xi_5 & \leq & x_2 + x_4 + x_5 + y_1 + y_3 \\
\xi_3 + \xi_4 + \xi_5 & \leq & x_3 + x_4 + x_5 + y_2 + y_3 + y_4 \\
\xi_1 + \xi_2 + \xi_3 + \xi_4 & \leq & x_1 + x_2 + x_3 + x_4 + y_2 + y_5 \\
\xi_1 + \xi_2 + \xi_3 + \xi_5 & \leq & x_1 + x_2 + x_3 + x_5 + y_4 + y_5 \\
\xi_1 + \xi_2 + \xi_4 + \xi_5 & \leq & x_1 + x_2 + x_4 + x_5 + y_3 \\
\xi_1 + \xi_3 + \xi_4 + \xi_5 & \leq & x_1 + x_3 + x_4 + x_5 + y_1 + y_2 + y_3 + y_4 \\
\xi_2 + \xi_3 + \xi_4 + \xi_5 & \leq & x_1 + x_3 + x_4 + x_5 + y_1 \\
\xi_1 + \xi_2 + \xi_3 + \xi_4 + \xi_5 & \leq & x_1 + x_2 + x_3 + x_4 + x_5
\end{array} \tag{2.34}$$

$$A_1 x + A_2 y \geq b$$

$$x \geq 0, y \geq 0 \qquad .$$

Illustration of model 2.33 on 5 node example in Figure 2.1

The relaxed problem takes the form;

$$\min\left\{\sum_{i=1}^{5} c_i(x_i)+\sum_{j=1}^{5} K_j(y_j)\right\}$$

subject to

$$\begin{array}{rcc}
x_1+y_1 & \geq & 10\lambda_1+15\lambda_2+15\lambda_3+15\lambda_4+15\lambda_5 \\
x_2+y_1+y_2+y_3+y_4 & \geq & 9\lambda_1+9\lambda_2+9\lambda_3+9\lambda_4+9\lambda_5 \\
x_3+y_3 & \geq & 12\lambda_1+8\lambda_2+12\lambda_3+12\lambda_4+12\lambda_5 \\
x_4+y_4+y_5 & \geq & 9\lambda_1+9\lambda_2+6\lambda_3+9\lambda_4+6\lambda_5 \\
x_5+y_2+y_5 & \geq & 15\lambda_1+15\lambda_2+15\lambda_3+10\lambda_4+10\lambda_5 \\
x_1+x_2+y_2+y_3+y_4 & \geq & 19\lambda_1+24\lambda_2+24\lambda_3+24\lambda_4+24\lambda_5 \\
x_1+x_3+y_1+y_2+y_4 & \geq & 21\lambda_1+17\lambda_2+21\lambda_3+21\lambda_4+21\lambda_5 \\
x_2+x_4+y_1+y_2+y_3+y_5 & \geq & 18\lambda_1+18\lambda_2+15\lambda_3+18\lambda_4+15\lambda_5 \\
x_2+x_5+y_1+y_3+y_4+y_5 & \geq & 24\lambda_1+24\lambda_2+24\lambda_3+19\lambda_4+19\lambda_5 \\
x_4+x_5+y_2+y_4 & \geq & 24\lambda_1+24\lambda_2+21\lambda_3+19\lambda_4+16\lambda_5 \\
x_1+x_2+x_3+y_2+y_4 & \geq & 31\lambda_1+32\lambda_2+36\lambda_3+36\lambda_4+36\lambda_5 \\
x_1+x_2+x_4+y_2+y_3+y_5 & \geq & 28\lambda_1+33\lambda_2+30\lambda_3+33\lambda_4+30\lambda_5 \\
x_1+x_2+x_5+y_3+y_4+y_5 & \geq & 34\lambda_1+39\lambda_2+39\lambda_3+34\lambda_4+34\lambda_5 \\
x_2+x_3+x_4+y_1+y_2+y_5 & \geq & 30\lambda_1+26\lambda_2+27\lambda_3+30\lambda_4+27\lambda_5 \\
x_2+x_3+x_5+y_1+y_4+y_5 & \geq & 36\lambda_1+32\lambda_2+36\lambda_3+31\lambda_4+31\lambda_5 \\
x_2+x_4+x_5+y_1+y_3 & \geq & 40\lambda_1+33\lambda_2+30\lambda_3+28\lambda_4+25\lambda_5 \\
x_3+x_4+x_5+y_2+y_3+y_4 & \geq & 46\lambda_1+32\lambda_2+33\lambda_3+31\lambda_4+28\lambda_5 \\
x_1+x_2+x_3+x_4+y_2+y_5 & \geq & 43\lambda_1+41\lambda_2+42\lambda_3+45\lambda_4+42\lambda_5 \\
x_1+x_2+x_3+x_5+y_4+y_5 & \geq & 46\lambda_1+47\lambda_2+51\lambda_3+46\lambda_4+46\lambda_5 \\
x_1+x_2+x_4+x_5+y_3 & \geq & 43\lambda_1+48\lambda_2+45\lambda_3+43\lambda_4+40\lambda_5 \\
x_1+x_3+x_4+x_5+y_1+y_2+y_3+y_4 & \geq & 46\lambda_1+47\lambda_2+48\lambda_3+46\lambda_4+43\lambda_5 \\
x_2+x_3+x_4+x_5+y_1 & \geq & 45\lambda_1+41\lambda_2+42\lambda_3+40\lambda_4+37\lambda_5 \\
x_1+x_2+x_3+x_4+x_5 & \geq & 55\lambda_1+56\lambda_2+57\lambda_3+55\lambda_4+52\lambda_5
\end{array}$$

$$\begin{aligned} &\textstyle\sum_{i=1}^{5} \lambda_i = 1 && (2.35)\\ &A_1 x + A_2 y \geq b \\ &x \geq 0, y \geq 0 \quad . \end{aligned}$$

2.4 Solution of the Problem Presented in Section 5

For simplicity we assume that the objective function is linear, but it can be of a very large size, hence a special algorithm may be more efficient than the use of a general purpose LP package. There are two algorithms available that offer solutions for our problem:

I. The Prékopa–Vizvári-Badics (PVB) algorithm (1998), where the p-efficient points are first enumerated or they are all known from another source.

II. The Dentcheva–Prékopa–Ruszczyński (DPR) algorithm (2000) that generates the p-efficient points simultaneously with the solution algorithm.

The PVB algorithm is described in a somewhat more complete way in Prékopa (2006). We will comment on it in the next section.

We propose the use of the DPR algorithm with an important improvement regarding the calculation of the new p-efficient points in the course of the iteration. We also use ideas from Vizvári (2002), where the DPR algorithm is presented in a slightly different way. First we rewrite problem (2.33) in the following form, where $J = \{1, \ldots, M\}$:

$$\begin{gathered} \min\left\{c_1^T x + c_2^T y\right\} \\ \text{subject to} \\ T_1 x + T_2 y \geq \textstyle\sum_{j \in J} \lambda_j v^{(j)} \\ A_1 x + A_2 y \geq b \qquad (2.36) \\ \textstyle\sum_{i=1}^{M} \lambda_i = 1 \\ x \geq 0,\ y \geq 0,\ \lambda \geq 0. \end{gathered}$$

where $v^{(1)}, \ldots, v^{(M)}$ are the network p-efficient points 2.31.

If we introduce slack variables u,w in the inequality constraints, then the problem can be written as:

$$\min\left\{c_1^T x + c_2^T y + 0^T u + 0^T w + 0^T \lambda\right\}$$

$$\text{subject to}$$

$$\text{(P)} \qquad \begin{pmatrix} T_1 & T_2 & -E & 0 & -V \\ A_1 & A_2 & 0 & -E & 0 \\ 0^T & 0^T & 0^T & 0^T & e^T \end{pmatrix} \begin{pmatrix} x \\ y \\ u \\ w \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ b \\ 1 \end{pmatrix} \tag{2.37}$$

$$x \geq 0, \;\; y \geq 0, \;\; u \geq 0, \;\; w \geq 0, \;\; \lambda \geq 0,$$

where $V = \left(v^{(j)},\ j \in J\right)$, is the $(n+t) \times M$ matrix and $e^T = (1, \ldots, 1)$. We subsequently generate the columns of V. Let J_h designate the subscript set of the available p-efficient points and $V_h = \left(v^{(j)},\ j \in J_h\right)$. In iteration h we have a problem that differs from (P) in such a way that we replace V_h for V. Let (P_h) designate that LP.

Solve (P_h) by a method that produces an optimal basis satisfying the optimality condition and let α be the optimal dual vector. Partition α into α_1, α_2, α_3, consistent with the partitioning of the rows of the matrix in problem (P_h).

If α is an optimal dual vector in problem (P) too, then we are done, the current problem (P_h) provides us with the optimal solution of problem (P). Otherwise there exists a column in the matrix of problem (P) that has a scalar product with α greater than the corresponding objective function coefficient. This may happen to a column that belongs to the last block of the matrix because the other columns are the same as those in (P_h). The column in the last block we are referring to is unknown but we know that its transpose has the form: $(-v^T, 0^T, 1)$, where v is a column in V but not in V_h. Writing up the scalar product we obtain:

$$\left(-v^T, 0^T, 1\right)\alpha = \alpha_3 - v^T \alpha_1 > 0. \tag{2.38}$$

On the other hand, if we look at the columns in problem (P_h), then we observe that at least one component of λ must be basic. If the corresponding column in the problem is $\left(-v^{(h)}, 0^T, 1\right)$, then we have the equation:

$$\left(-\left(v^{(h)}\right)^T, 0^T, 1\right)\alpha = 0$$

which implies that

$$\alpha_3 - \left(v^{(h)}\right)^T \alpha_1 = 0. \tag{2.39}$$

Relations (2.38) and (2.39) tell us that a new column (and variable) can enter the basis in problem (P_h) iff

$$\min_{i \in J_h} \alpha_1^T v^{(i)} > \min_{i \in J} \alpha_1^T v^{(i)}. \tag{2.40}$$

The new column and variable will be supplied by the solution of the problem:

$$\min_{i \in J} \alpha_1^T v^{(i)} \tag{2.41}$$

If the two values in (2.40) are equal, then the procedure terminates. Note that if we take the scalar product with the columns in (P_h) that belong to the second to the last block, then we obtain the inequality $\alpha_1 \geq 0$. On the other hand, if the optimum value of problem (2.37) is different from 0, then (2.39) implies that $\alpha_1 > 0$. In fact, if $\alpha_1 = 0$, then $\alpha_3 = 0$ and the optimum value of the dual of problem (2.37) would be 0, contrary to the assumption.

In the next section we need the stronger inequality: $\alpha_1 \gg 0$. To ensure it, we need same condition in connection with the p-efficient points.

Regularity condition. Let $v^{(i)}$, $i \in L$ be a collection of linearly independent p-efficient points. Then there exists a $v^{(j)}$, $j \in L$ such that the intersection of the linear subspace spanned by $v^{(i)} - v^{(j)}$, $i \in L$, $i \neq j$ and the nonnegative orthant R_+^n has the 0 vector in common.

Theorem 2.4.1 *If the regularity condition holds true, then at any iteration of the solution algorithm of Section 6 the dual vector can be chosen in such a way that $\alpha_1 \gg 0$.*

Proof Proof of Theorem 2.4.1. Let L be the subscript set of the basic vectors from the last block in problem (2.37). The regularity condition tells us that there is a $j \in L$

such that there are real values y_i, $i \in L \setminus \{j\}$ satisfying

$$\sum_{i \in L \setminus \{i\}} y_i (v^{(i)} - v^{(j)}) > 0. \tag{2.42}$$

Then, by the theorem of Stiemke ($Ax = 0$ has a solution $x \gg 0$ iff there is no y such that $y^T A > 0$) we have that

$$(v^{(i)} - v^{(j)})^T \alpha_1 = 0, \ \ i \in L \setminus \{j\} \tag{2.43}$$

has a solution $\alpha_1 \gg 0$. Since $\alpha_3 = (v^{(j)})^T \alpha_1$, (2.43) implies that

$$\alpha_3 - (v^{(i)})^T \alpha_1 = 0, \ \ i \in L.$$

□

Illustration of model 2.37 on 5 node example in Figure 2.1

Explicit version of model 2.37 for the 5 node example is available in Appendix A. After we solve problem 2.59 (see Appendix A) with a method that produces dual optimal vector, α we obtain the below optimal solutions for primal and dual vector:

$$Primal = \begin{pmatrix} 15 \\ 10 \\ 10 \\ 10 \\ 10 \\ 10 \\ 10 \\ 10 \\ 10 \\ 10 \\ 15 \\ 41 \\ 8 \\ 21 \\ 15 \\ 36 \\ 34 \\ 42 \\ 36 \\ 16 \\ 24 \\ 37 \\ 31 \\ 30 \\ 14 \\ 10 \\ 14 \\ 22 \\ 19 \\ 12 \\ 39 \\ 5 \\ 4.20E{-}12 \\ 5 \\ 5.54E{-}13 \\ 1.96E{-}12 \\ 3.76E{-}13 \\ 3.25E{-}13 \\ 1.43E{-}12 \\ 4.99E{-}13 \\ 4.08E{-}13 \\ 1.62E{-}12 \\ 6.20E{-}13 \\ 1 \end{pmatrix} \quad Dual = \alpha = \begin{pmatrix} 1.03245E{-}13 \\ 3.67671E{-}14 \\ 1.89368E{-}13 \\ 7.19208E{-}14 \\ 1.00581E{-}13 \\ 4.2023E{-}14 \\ 4.49084E{-}14 \\ 3.58616E{-}14 \\ 4.19383E{-}14 \\ 9.47355E{-}14 \\ 6.39031E{-}14 \\ 4.08078E{-}14 \\ 4.89421E{-}14 \\ 5.02673E{-}14 \\ 1.10418E{-}13 \\ 1.56502E{-}13 \\ 1.13039E{-}13 \\ 6.97396E{-}14 \\ 8.1942E{-}14 \\ 1.31502E{-}13 \\ 3.89224E{-}14 \\ 5.73884E{-}13 \\ 1 \\ 3.3134E{-}13 \\ 4 \\ 1 \\ 6 \\ 7 \\ 2 \\ 5 \\ 6 \\ 2 \\ 4 \\ 55 \end{pmatrix}$$

After representing the dual vector α, we can partition it into $(\alpha_1, \alpha_2, \alpha_3)$ and verify the condition (2.39) and show that the regularity condition holds true for the α_1 vector.

Condition (2.39) verification:

$$\alpha_3 - \left(v^{(h)}\right)^T \alpha_1 = 55 - \begin{pmatrix} 10 \\ 9 \\ 12 \\ 9 \\ 15 \\ 19 \\ 21 \\ 18 \\ 24 \\ 24 \\ 31 \\ 28 \\ 34 \\ 30 \\ 36 \\ 40 \\ 46 \\ 43 \\ 46 \\ 43 \\ 46 \\ 45 \\ 55 \end{pmatrix} \times \begin{pmatrix} 1.03245E{-}13 \\ 3.67671E{-}14 \\ 1.89368E{-}13 \\ 7.19208E{-}14 \\ 1.00581E{-}13 \\ 4.2023E{-}14 \\ 4.49084E{-}14 \\ 3.58616E{-}14 \\ 4.19383E{-}14 \\ 9.47355E{-}14 \\ 6.39031E{-}14 \\ 4.08078E{-}14 \\ 4.89421E{-}14 \\ 5.02673E{-}14 \\ 1.10418E{-}13 \\ 1.56502E{-}13 \\ 1.13039E{-}13 \\ 6.97396E{-}14 \\ 8.1942E{-}14 \\ 1.31502E{-}13 \\ 3.89224E{-}14 \\ 5.73884E{-}13 \\ 1 \end{pmatrix}^T = 0$$

Regularity condition verification:

Since the regularity condition holds, we can show that $\alpha_1 \gg 0$:

$$\alpha_1 = \begin{pmatrix} 1.03245E{-}13 \\ 3.67671E{-}14 \\ 1.89368E{-}13 \\ 7.19208E{-}14 \\ 1.00581E{-}13 \\ 4.2023E{-}14 \\ 4.49084E{-}14 \\ 3.58616E{-}14 \\ 4.19383E{-}14 \\ 9.47355E{-}14 \\ 6.39031E{-}14 \\ 4.08078E{-}14 \\ 4.89421E{-}14 \\ 5.02673E{-}14 \\ 1.10418E{-}13 \\ 1.56502E{-}13 \\ 1.13039E{-}13 \\ 6.97396E{-}14 \\ 8.1942E{-}14 \\ 1.31502E{-}13 \\ 3.89224E{-}14 \\ 5.73884E{-}13 \\ 1 \end{pmatrix} \gg 0$$

Since V_h vector doesn't yield an optimal solution to the original problem, we need to find a new p-efficient point to enter as a new column to V_h.

2.5 Finding new p-efficient Point

2.5.1 The General Case

This section is largely influenced by Prékopa, Unuvar (2010). The solution of the problem (2.41) can be carried out by solving another problem, where the unknown vector is of much smaller size. Let $F(z)$ be the c.d.f. of ξ.

If we take into account the p-efficient point $v^{(i)}$, $i = 1, \ldots, M$ are those in (2.31), then we can derive an expression for $\alpha_1^T v^{(i)}$ by the use of $z^{(i)}$ which is an efficient point of ξ. In fact,

$$\begin{aligned} \alpha_1^T v^{(i)} &= \sum_{j=1}^{n} \alpha_{1j} z_j^{(i)} + \sum_{h=n+1}^{r} \alpha_{1h} \sum_{j \in I_h} z_j^{(i)} \\ &= \sum_{j=1}^{n} \alpha_{1j} z_j^{(i)} + \sum_{j=1}^{n} z_j^{(i)} \sum_{I_h \ni j} \alpha_{1h} \\ &= \sum_{j=1}^{n} \left(\alpha_{1j} + \sum_{I_h \ni j} \alpha_{1h} \right) z_j^{(i)}, \;\; i = 1, \ldots, n, \end{aligned} \tag{2.44}$$

where $r = n + t$. Introducing the notation:

$$\begin{aligned} \gamma_j &= \alpha_{1j} + \sum_{I_h \ni j} \alpha_{1h} \\ \gamma &= (\gamma_1, \ldots, \gamma_n)^T, \end{aligned}$$

equation (2.44) can be written in the form:

$$\alpha_1^T v^{(i)} = \gamma^T z^{(i)}, i = 1, \ldots, n. \tag{2.45}$$

There is a one-to-one correspondence between the p-efficient points $z^{(i)}$, $i = 1, \ldots, M$, such that $v^{(i)} \leftrightarrow z^{(i)}$ and (2.45) holds true. Since $\alpha_1 \gg 0$, this implies that problem (2.41) can be solved in such a way that we solve the smaller size problem:

$$\begin{aligned} &\min \gamma^T z \\ &\text{subject to} \\ &F(z) \geq p \\ &z \in Z. \end{aligned} \tag{2.46}$$

In most cases we know lower and upper bounds on the components of ξ, which, in turn, can be prescribed for the components of z. At this point we just supplement the constraint $z \in D$ to problem (2.46) with the remark that it may mean the mentioned lower and upper bounds on z. The solution of problem (2.46), in that general form, may still be computationally intensive because the number of p-efficient points of F may be very large. There is no need, however, to solve problem (2.46) optimally. It is enough to enumerate the p-efficient points (e.g., by the use of the PVB algorithm), until a $z \in J$, $z \notin J_h$ is found for which

$$\min_{i \in J_h} \alpha_1^T v^{(i)} > \gamma_1^T z.$$

Let z be the new p-efficient point.

Problem (2.46) can be reformulated as a discrete optimization problem as follows:

$$\begin{gathered} \min \textstyle\sum_i \gamma^T u_i \varepsilon_i \\ \text{subject to} \\ \textstyle\sum_i F(u_i)\varepsilon_i \geq p \\ \textstyle\sum_i \varepsilon_i = 1 \\ u_i \in Z, u_i \in D, \varepsilon_i \in \{0,1\}, \text{all } i. \end{gathered} \tag{2.47}$$

Another simple reformulation is possible if the probability function values, rather than the c.d.f. values of ξ are available.

Let $p_i = p(\xi = u_i)$, for $u_i \in Z$, $u_i \in D$. Then the problem is:

$$\begin{gathered} \min \textstyle\sum_i \gamma^T u_i \varepsilon_i \\ \text{subject to} \\ \textstyle\sum_i p_i \varepsilon_i \geq p \\ \varepsilon_i \leq \varepsilon_j \text{ for } u_j \leq u_i \\ u_i \in Z, u_i \in D, \varepsilon_i \in \{0,1\}, \text{all } i. \end{gathered} \tag{2.48}$$

2.5.2 Specialization and Relaxation of Problem (2.47)

In this section we present our version of the greedy algorithm (see Pisinger, 1995) for the solution of the knapsack problem and its application to solve problem (2.46) in case of independent random variables with strictly logconcave univariate marginal c.d.f.'s are also logconcave.

Assume that $\xi_1, \ldots, \xi_n$ are independent, integer valued and let F_i be the c.d.f. of ξ_i, $i = 1, \ldots, n$. Assume, further, that $\xi_i \in [l_i, u_i]$ and F_i is strictly logconcave in $[l_i, u_i]$, $i = 1, \ldots, n$. Then problem (2.46) an be written in the form:

$$\begin{gathered}
\min \textstyle\sum_{i=1}^{n} \sum_{k=l_i}^{u_i} \gamma_i k \delta_{ik} \\
\text{subject to} \\
\textstyle\sum_{i=1}^{n} \sum_{k=l_i}^{u_i} a_{ik} \delta_{ik} \leq d \\
z \in D \\
\textstyle\sum_{k=l_i}^{u_i} \delta_{ik} = 1, \;\; i = 1, \ldots, n \\
\delta_{ik} \in \{0, 1\}, \;\; \text{all } i, k,
\end{gathered} \tag{2.49}$$

where $a_{ik} = -\log F_i(k)$ and $d = -\log p$. The problem is a special case of the Multiple Choice Knapsack Problem (MCKP). In the general case we have h_{ik} instead of $k\alpha_{2i}$.

Illustration of model 2.47 on 5 node example in Figure 2.1

Let us assume that ξ_i's in model 2.26 are independent of each other and we further assume that there are lower and upper bounds for each ξ_i such that $[l_i, u_i]$ where $i = 1, \ldots, 5$. We can model our 5 node example similarly in (2.47) to obtain the new p-efficient point.

$$
\begin{gathered}
\min \textstyle\sum_{i=1}^{5} \sum_{k=l_i}^{u_i} \gamma_i k \delta_{ik} \\
\text{subject to} \\
-\log(0.2)\delta_{1,5} - \log(0.8)\delta_{1,10} - \log(1)\delta_{1,15} \\
-\log(0.3)\delta_{2,3} - \log(0.7)\delta_{2,6} - \log(1)\delta_{2,9} \\
-\log(0.2)\delta_{3,4} - \log(0.8)\delta_{3,8} - \log(1)\delta_{3,12} \\
-\log(0.1)\delta_{4,3} - \log(0.9)\delta_{4,6} - \log(1)\delta_{4,9} \\
-\log(0.1)\delta_{5,5} - \log(0.9)\delta_{5,10} - \log(1)\delta_{5,15} \quad \leq \log(0.8) \\
z \in D \\
\textstyle\sum_{k=l_i}^{u_i} \delta_{ik} = 1, \;\; i = 1, \ldots, 5 \\
\delta_{ik} \in \{0, 1\}, \;\; \text{all } i, k,
\end{gathered}
\tag{2.50}
$$

For the solution of problem (2.49) we use a greedy method in Pisinger (1995), where the first step is the solution of a relaxed LP called Linear Multiple Choice Knapsack Problem (LMCKP). We relax problem (2.49) in such a way that we allow the δ_{ik} variables to move freely in the interval $[0, 1]$:

$$
\begin{gathered}
\min \textstyle\sum_{i=1}^{n} \sum_{k=l_i}^{u_i} \gamma_i k \delta_{ik} \\
\text{subject to} \\
\textstyle\sum_{i=1}^{n} \sum_{k=l_i}^{u_i} a_{ik} \delta_{ik} \leq d \\
z \in D \\
\textstyle\sum_{k=l_i}^{n_i} \delta_{ik} = 1, \;\; i = 1, \ldots, n \\
\delta_{ik} \geq 0, \;\; \text{all } i, k.
\end{gathered}
\tag{2.51}
$$

To solve problem (2.51) we use a special algorithm. We introduce slack variable u in the inequality constraint in problem (2.51), then split the sum into n terms, each term corresponds to a component of ξ. It will be more convenient in the new problem to use slightly different notations. We change the range of the second subscripts so that the summation should go from 1 to m_i and designate the coefficient of δ_{ik} in the objective function by h_{ik}. Note that for every i, the discrete function h_{ik} is linear in k

with coefficient $\delta_i > 0$. Then the new problem is:

$$\min\left\{0u + 0u_1 + \cdots + 0u_n + h_{11}\delta_{11} + \cdots + h_{1m_1}\delta_{1m_1} + \cdots + h_{n1}\delta_{n1} + \cdots + h_{nm_n}\delta_{nm_n}\right\}$$

subject to

$$\begin{array}{cccccccc}
u & + & u_1 & + \cdots + u_n & & & = & d \\
 & & -u_1 & & +a_{11}\delta_{11} + \cdots + a_{1m_1}\delta_{1m_1} & & = & 0 \\
 & & & \ddots & & & & \vdots \\
 & & & -u_n & & +a_{n1}\delta_{n1} + \cdots + a_{nm_n}\delta_{nm_n} & = & 0 \\
 & & & & \delta_{11} + \cdots + \delta_{1m_1} & & = & 1 \\
 & & & & \ddots & & & \vdots \\
 & & & & & \delta_{n1} + \cdots + \delta_{nm_n} & = & 1
\end{array} \tag{2.52}$$

$$u \geq 0, \;\; u_i \geq 0, \;\; i = 1, \ldots, n, \;\; \delta_{ik} \geq 0, \quad \text{all } i, k.$$

Illustration of model 2.52 on 5 node example in Figure 2.1

Let us relax the problem 2.50 and then rewrite in the form of 2.52:

$$\min\left\{0u + 0u_1 + \cdots + 0u_5 + h_{11}\delta_{11} + \cdots + h_{13}\delta_{13} + \cdots + h_{51}\delta_{51} + \cdots + h_{53}\delta_{53}\right\}$$

subject to

$$\begin{array}{cccccccc}
u & + & u_1 & + \cdots + u_5 & & & = & -\log(0.8) \\
 & & -u_1 & & -\log(0.2)\delta_{11} - \cdots - \log(1)\delta_{13} & & = & 0 \\
 & & & \ddots & & & & \vdots \\
 & & & -u_5 & & -\log(0.1)\delta_{51} - \cdots - \log(1)\delta_{53} & = & 0 \\
 & & & & \delta_{11} + \quad \cdots \quad + \delta_{13} & & = & 1 \\
 & & & & \ddots & & & \vdots \\
 & & & & & \delta_{51} + \quad \cdots \quad + \delta_{53} & = & 1
\end{array} \tag{2.53}$$

$$u \geq 0, \;\; u_i \geq 0, \;\; i = 1, \ldots, 5, \;\; \delta_{ik} \geq 0, \quad \text{all } i, k.$$

Problem (2.52) is related to the simple recourse problem in stochastic programming, when we apply the λ-representations for the piecewise linear separable functions in the objective, for the case of discrete random variables (see Prékopa, 1990 a, 1995, Chapter 9). The matrix of the equality constraints, together with the coefficient sequences in the objective function, can be partitioned into $n+1$ blocks and labeled by $0, 1, \ldots, n$ respectively. The matrices taken from blocks $1, \ldots, n$,

$$\begin{pmatrix} a_{i1} & \ldots & a_{im_i} \\ 1 & \ldots & 1 \end{pmatrix}, \quad \begin{pmatrix} h_{i1} & \ldots & h_{im_i} \\ a_{i1} & \ldots & a_{im_i} \\ 1 & \ldots & 1 \end{pmatrix} \tag{2.54}$$

have a property enjoyed by the corresponding matrices in the simple recourse problem as formulated by Prékopa.

Theorem 2.5.1 *All* 2×2 *minors of the first and all* 3×3 *minors of the second matrices in (2.54) are nonnegative.*

Proof Proof of Theorem 2.5.1. The sequence $-a_{i1}, \dots, -a_{im_i}$ is non-decreasing, hence any 2×2 minor of the first matrix is nonnegative. As regards the second matrix, if we pick three columns from it, corresponding to $j < k < l$, then its determinant is γ_i times the second order divided difference of $-a_{ij}, -a_{ik}, -a_{il}$, where $\gamma_i > 0$. Here we took into account the convexity of the sequence $-a_{i1}, \dots, -a_{im_i}$ which is a consequence of the strict logconcavity of the c.d.f. F_i and the strict convexity of the sequences $-a_{i1}, \dots, -a_{im_i}$. It follows that, any dual feasible basis of problem (2.52) has two consecutive columns from each block $1, \dots, n$. (see Prékopa 1990 a, 1995).

An LP: $\min(\max) c^T x$ subject to $Ax = b,\ x \geq 0$ is called totally positive (in Prékopa, 1009 b) if all $m \times n$ minors of A and all $(m+1) \times n$ minors of $\left(\begin{smallmatrix} c^T \\ A \end{smallmatrix}\right)$ are positive, where A is a $m \times n$ matrix, $n \geq m + 1$. In the above mentioned paper Prékopa proved the following

Theorem 2.5.2 *The dual feasible basis of a totally positive LP have the following structure, presented in terms of the basis subscripts:*

	m even	*m odd*
min problem	$i, i+1, \dots, j, j+1$	$i, i+1, \dots, j, j+1, n$
max problem	$1, i, i+1, \dots, j, j+1, n$	$1, i, i+1, \dots, j, j+1,$

where the subscripts are arranged in increasing order.

If we specialize this theorem for the LP:

$$\begin{gathered} \min\ \textstyle\sum_{j=1}^{n_i} h_{ij} \delta_{ij} \\ \text{subject to} \\ \textstyle\sum_{j=1}^{n_i} a_{ij} \delta_{ij} = u_i \\ \textstyle\sum_{j=1}^{u_i} \delta_{ij} = 1 \\ \delta_{ij} = 1, \ \ j = 1, \dots, n_i, \end{gathered} \tag{2.55}$$

then we derive the consequence that all dual feasible basis of the problem are consecutive pairs of columns of the matrix of the equality constraints.

2.5.3 Solution of Problem (2.52)

An efficient dual type algorithm for the solution of the simple recourse problem is presented in Prékopa (1990 a, 1995) and further developed by Fábián, Prékopa, Ruff-Fiedler (1995). The same method solves efficiently problem (2.52) too. Here we present only the construction of the initial dual feasible basis, because it is particularly simple in this case and mention how we can obtain fast and very good bounds for the optimum value.

Finding initial dual feasible basis

Pick arbitrary two consecutive columns from each of the blocks $1, \ldots, n$ in problem (2.52), as part of a dual feasible basis of the entire problem. Let v_i, w_i be the dual variables corresponding to problem (2.55), $i = 1, \ldots, n$. Since the rows of blocks $1, \ldots, n$ are disjoint, the $v_1, \ldots, v_n,\ w_1, \ldots, w_n$ can be regarded as dual variables corresponding to problem (2.55), where, however, one column and one dual variable is further to be chosen. Let y designate the last dual variable. This and the final column of the dual feasible basis can be found by the solution of the LP:

$$\begin{gathered}\min \textstyle\sum_{i=1}^{n} (-v_i) u_i \\ \text{subject to} \\ \textstyle\sum_{i=1}^{n} u_i = d \\ u \geq 0.\end{gathered} \tag{2.56}$$

The optimal solution is $u_i = d$, $u_i = 0$, for $i \neq j$, where $j = \text{argmin}(v_i)$. The column of u_i in block 0 is the final one to form a dual feasible basis B_0 with the already chosen consecutive pairs from blocks $1, \ldots, n$. The final consecutive of the corresponding dual vector is $y = v_j$.

Illustration of finding initial dual feasible basis on 5 node example in Figure 2.1

Let us demonstrate how to initiate the dual feasible basis for the 5 node example through the model (2.53). Let us first write down the dual problem to help visualizing the dual variables easily.

$$
\begin{gathered}
\max dy + 0v_1 + 0v_2 + 0_v3 + 0v_4 + 0v_5 + w_1 + w_2 + w_3 + w_4 + w_5 \\
\text{subject to} \\
y \le 0 \\
y + v_1 \le 0 \\
y + v_2 \le 0 \\
y + v_3 \le 0 \\
y + v_4 \le 0 \\
y + v_5 \le 0 \\
-\log(0.2)v_1 + w_1 \le h_{11} \\
-\log(0.8)v_1 + w_1 \le h_{12} \\
-\log(1)v_1 + w_1 \le h_{13} \\
\vdots \\
-\log(0.1)v_5 + w_5 \le h_{51} \\
-\log(0.9)v_5 + w_5 \le h_{52} \\
-\log(1)v_5 + w_5 \le h_{53} \\
y, v_i, w_i \text{ are free where } i = 1, \dots, 5
\end{gathered}
\tag{2.57}
$$

In the primal problem, there are total 6 blocks. We call the block which covers u_i variables as 0^{th} block and then there are 5 more blocks that covers the δ_{ij} variables. Below we represent the matrices that are taken from the last 5 blocks including the cost coefficients of the problem (2.53).

$$
\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ -\log(0.2) & -\log(0.8) & -\log(1) \\ 1 & 1 & 1 \end{pmatrix} \cdots \begin{pmatrix} h_{51} & h_{52} & h_{53} \\ -\log(0.1) & -\log(0.9) & -\log(1) \\ 1 & 1 & 1 \end{pmatrix}
$$

As Theorem 2.5.2 states, all dual feasible basis of simple recourse problem are consecutive pairs of columns of the above mentioned blocks. One can pick two arbitrary columns from each of the 5-blocks in problem (2.53) to define a dual feasible vector for the v_i and w_i variables where $i = 1, \dots, 5$. However, for the dual variable y, the minimum of $v_1, \dots, v_5$ needs to be picked, in other words, the solution of the problem (2.56) will yield a feasible value for the dual vector y. Below we present the bases encountered by the procedure. Initially we choose first two vectors

Subscripts of the basic δ_{ij} variables,					Subscripts of the basic u_j variables
$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	
1,2	1,2	1,2	1,2	1,2	1,2,5

from each blocks 1, 2, 3, 4, 5.

where

$$\begin{aligned}
\delta_{11} &= 0.98678937, & \delta_{12} &= 0.01321063\\
\delta_{21} &= 0.8745342, & \delta_{22} &= 0.1254658\\
\delta_{31} &= 0.86374993, & \delta_{32} &= 0.74364\\
\delta_{41} &= 0.873545, & \delta_{42} &= 0.126455\\
\delta_{51} &= 0.78345255, & \delta_{52} &= 0.21654745
\end{aligned}$$
$$u_1 = 0.691016413,\ \ u_2 = 0.623432066,\ \ u_5 = 0.568595448$$

The above solution is not primal feasible since sum of u_i variables is exceeding $\log(d)$ yielding a negative value for u. In the next section, we will describe and show how one can create a primal feasible basis and calculate an upper bound for problem (2.53).

Fast bounds for the optimal value

Having dual feasible basis for the minimization problem we also have a lower bound for the optimum value. The basic solution, corresponding to the dual feasible basis is not necessarily primal feasible, but we can easily create a primal feasible basis in the following way. Keep the vector that has been obtained as the optimal solution of problem (2.55) and $j_1, \dots, j_n$ in such a way that the solution for $\delta_{ij}, \delta_{ij_{i+1}}$ of the equation:

$$\begin{aligned}
a_{ij_i}\delta_{ij_i} + a_{ij_{i+1}}\delta_{ij_{i+1}} &= u_i\\
\delta_{ij_i} + \delta_{ij_{i+1}} &= 1, \quad i = 1, \dots, n
\end{aligned}$$

be nonnegative. Then the new basis B_1, consisting of the columns subscripted by j, from block 0, and j_i, j_{i+1}, from block i, $i = a, \dots, n$, is primal feasible and provides us with an upper bound for the optimum value of problem (2.52).

The bounding procedure can be continued. Keeping the consecutive pairs from blocks $1, \dots, n$ we can construct a further dual feasible basis B_2 in the same way as we have constructed B_0 etc. The lower bounds may not be increasing and the upper bounds may not be decreasing. In addition, the bounding procedure may not provide us with the exact optimum value but we choose the best bounds after a finite number of steps. Having a close bound, corresponding to a primal feasible basis we may pass to a feasible solution where any one of the $\delta_{ij_i}, \delta_{ij_{i+1}}$ is positive, for every $i = 1, \dots, n$, in a cost efficient way. If the obtained p-efficient point is not good enough, then we solve problem (2.52) optimally and only then pass to a p-efficient point in a cost efficient way.

An efficient algorithm for the solution of a problem of which (2.52) is a special case, is presented in Prékopa (1990, 1995) and further developed by Fábián, Prékopa, Ruff-Friedler (1995). The application of it to problem (2.52) is straightforward and will not be detailed. We note, however, that the specialized algorithm is very simple because of the simplicity of that part of the matrix which constitutes block 0. In the optimal solution, we need exactly one argument z_i of each F_i so that $z = (z_1, \ldots, z_n)$ is an optimal solution to the problem (2.52). However, at the end of the algorithm there may be blocks, among these labeled by $1, \ldots, n$, which have two columns in the optimal basis. The final step is to remove one out of each consecutive pairs in a cost efficient way. The obtained z solves problem (2.52).

Once we have the new p-efficient point for the distribution of ξ, we create the new p-efficient point for the random vector (2.29) and enter it into problem (P_h) to obtain (P_{h+1}).

Illustration of finding lower and upper bounds for the optimum value on 5 node example in Figure 2.1

Since the solution we found with the first basis in problem (2.53) is not primal feasible, let us create a primal feasible basis with a method described above. For the primal feasible basis, we will force each u_i variable to be equal to $a_{ij_i}\delta_{ij_i} + a_{ij_{i+1}}\delta_{ij_{i+1}}$ while $\delta_{ij_i} + \delta_{ij_{i+1}} = 1$ is satisfied with nonnegativity. The primal feasible basis that is obtained through this solution will yield an upper bound for the problem. Solution of below system of equations yield a non negative vector therefore gives a primal feasible basis with an optimum value of 2.322768. This optimum value constructs an upper bound for the primal problem.

$$\begin{gathered} -\log(0.2)\delta_{11} - \log(0.8)\delta_{12} = u_1 \\ -\log(0.3)\delta_{21} - \log(0.7)\delta_{22} = u_2 \\ -\log(0.8)\delta_{32} - \log(1)\delta_{33} = u_3 \\ -\log(0.1)\delta_{41} - \log(0.9)\delta_{42} = u_4 \\ -\log(0.9)\delta_{52} - \log(1)\delta_{53} = u_5 \\ \delta_{11} + \delta_{12} = 1 \\ \delta_{21} + \delta_{22} = 1 \\ \delta_{32} + \delta_{33} = 1 \\ \delta_{41} + \delta_{42} = 1 \\ \delta_{52} + \delta_{53} = 1 \end{gathered} \tag{2.58}$$

For the 5-node example, since the distribution function of the random variables can only take 3 different values,number of combinations for picking the consecutive columns to calculate the dual feasible basis is limited. However, for the systems where the distribution function takes much bigger number of values, it is better to select the consecutive columns in a cost-efficient way. In the below table we present the bases we encountered by the procedure where we pick them in a cost efficient way:

Subscripts of the basic δ_{ij} variables,					Subscripts of the basic u_j variables
$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	
1,2	1,2	1,2	1,2	1,2	1,2,5
1,2	2,3	1,2	1,2	1,2	1,2
1,2	2,3	2,3	1,2	1,2	1,2,5
1,2	2,3	2,3	1,2	1,2	1,2,5
2,3	2,3	3	1,2	1,2	1,3,4,5
2,3	2,3	3	1,2	1,2	1,3,4,5
2,3	2,3	3	2,3	1,2	1,3,5
2,3	2,3	3	2,3	2,3	1,4,5

with the optimum vector:

$$\begin{array}{rclcrcl} \delta_{12} & = & 0.00000123, & & \delta_{23} & = & 0.99999877 \\ \delta_{22} & = & 0.0000124, & & \delta_{23} & = & 0.9999876 \\ \delta_{32} & = & 0, & & \delta_{33} & = & 1 \\ \delta_{42} & = & 0.00000021, & & \delta_{43} & = & 0.9999979 \\ \delta_{51} & = & 0.0000233, & & \delta_{53} & = & 0.9999767 \end{array}$$

$$u_1 = 0.0000000191, \;\; u_4 = 0.045757691, \;\; u_5 = 0.0000001066.$$

2.6 Summary of the Solution Algorithm

In this section we summarize the solution algorithm of the stochastic network design problem (2.36).

It consists of the following steps.

Step 1. Rewrite problem (2.36) in the form of (2.37).

Step 2. Generate a few p-efficient points for ξ and create the corresponding p-efficient points of the random vector (2.29). Initialize J_0 as the subscript set of these p-efficient points.

Step 3. Set up and solve problem (P_h) by a method that produces primal-dual feasible (optimal) basis. Let α designate the optimal dual vector.

Step 4. Solve problem (2.46) to check if an entering variable to (P_h) exists, i.e. (2.40) holds. If it is not the case, then go to Step 5. If (2.40) holds then we find a new p-efficient point, form the union of j_h and the new p-efficient point, to obtain j_{h+1} and define (P_{h+1}). Go to Step 3.

Step 5. Stop, the optimal solution of problem (P_h) is the optimal solution of problem (P).

Finding a new p-efficient point means the solution of problem (2.47), if the components of ξ, are stochastically dependent. If the components of ξ are independent, then to find new p-efficient point for ξ in problem (2.29) is a multiple choice knapsack problem that we solve by the algorithm in Section 2.5.3.

2.7 Numerical Examples

Example This example is an 8 node network in Figure 2.4 with only 2 random demand nodes (Node 2 and Node 5) which are both binomially distributed on arithmetic sequences. The demands are assumed to be independent. Among 161 non-eliminated inequalities (see Appendix B) only 136 include at least one of the two demands. The 136 inequalities are stochastic and the remaining 25 are deterministic constraints. Table 2.4 provides us with the possible values of the random demands at Nodes 2 and 5. The associated probability distributions are presented in Table 2.5.

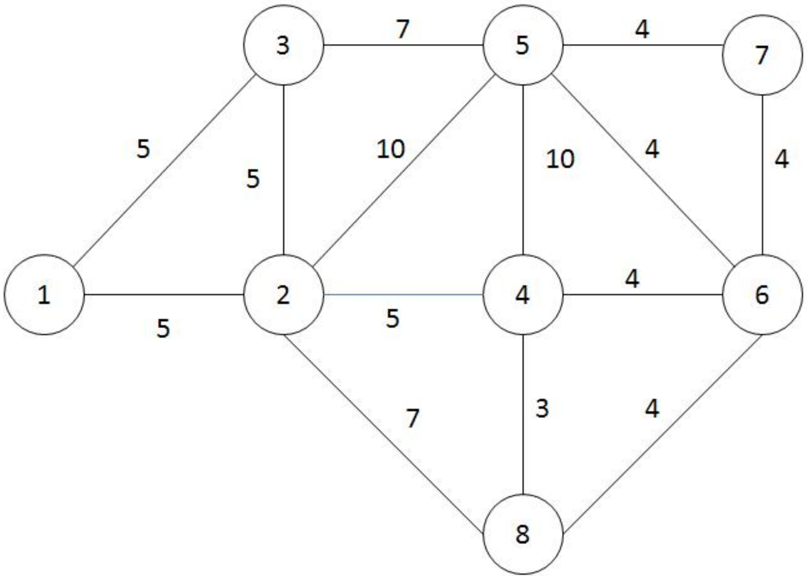


Figure 2.4: Eight-node network with arc capacities.

Table 2.4: Possible values of the random demands

ξ_2	33	38	43	48	53	58	63	68	73	78
ξ_5	15	20	25	30	35	40	45	50	55	60

Table 2.5: Associated probabilities (Binomial), c.d.f.

F_2 $(n = 9,\ p = 0.47)$	0.004	0.038	0.149	0.361	0.621	0.83	0.950	0. 990	0.999	1
F_5 $(n = 9,\ p = 0.47)$	0.003	0.029	0.123	0.316	0.573	0.80	0.936	0. 987	0.998	1

Since the binomial probability function is logconcave, both F_2 and F_5 are logconcave discrete function. The Stochastic Programming Problem to be solved is: $((S)$ means the feasibility inequality corresponding to $S \subset N$):

$$\min c^T x$$

$$\text{subject to}$$

$$P(y(S, \bar{S}) \geq d(S),\ \ S \subset N_1,\ \ (S)\ \text{ non-eliminated}) \geq 0.95,$$

$$\text{where } N_1 \text{ is the collection of the nodes with random demand,}$$

$$y(S, \bar{S}) \geq d(S),\ \ S \subset N_2,\ \ N_2 = N \setminus N_1$$

$$d(S) = \textstyle\sum_{i \in S}(\xi_i - x_i)$$

$$l_i \leq x_i \leq u_i \text{ for } i = 1, \dots, 8,$$

$$\text{where } l_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},\ \ u_i = \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \end{bmatrix},\ \ c = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 2 \\ 1 \\ 1 \\ 7 \\ 4 \end{bmatrix}$$

and $x = (x_1,\ x_2,\ x_3,\ x_4, x_5,\ x_6,\ x_7,\ x_8)^T$ is the decision vector. The solution steps are the following.

Step 1. Rewrite problem in the form of (2.37).

Step 2. $V_1 = (z^{(1)}) = \binom{78}{45}$.

Step 3. Set up and solve problem (P_1) by a method that produces primal-dual feasible (optimal) basis. The optimal solution is:

$$x = [89\ 100\ 58\ 100\ 100\ 87\ 34\ 47]^T$$

with optimum value= 1523. Let α designate the optimal dual vector.

Step 4.

Iteration 1: Solve problem (2.44). The optimal solution is $z^{(2)} = \binom{73}{55}$. Since (2.40) holds, include the new p-efficient point (2.31) that we obtain for the random vector (2.29), into (P_1), define (P_2) with $V_2 = (V_1, \binom{73}{55})$.

Iteration 2: Solve problem (2.44). The optimal solution is $z^{(3)} = \binom{68}{60}$. Since (2.40) holds, include the new p-efficient point into (P_1), define (P_2) with (P_2) with $V_3 = (V_2, \binom{68}{60})$.

Iteration 3: Solve problem (2.44). The optimal solution is $z^{(4)} = \binom{63}{60}$. Since (2.40) holds, include the new p-efficient point into (P_1), define (P_2) with (P_2) with $V_4 = (V_3, \binom{63}{60})$.

Iteration 4: Solve problem (2.44). The optimal solution is $z^{(5)} = \binom{63}{45}$. Since (2.40) holds, include the new p-efficient point into (P_1), define (P_2) with (P_2) with $V_5 = (V_4, \binom{63}{45})$.

Iteration 5: Solve problem (2.44). The optimal solution is $z^{(6)} = \binom{68}{55}$. Since (2.40) holds, include the new p-efficient point into (P_1), define (P_2) with (P_2) with $V_6 = (V_5, \binom{68}{55})$.

Iteration 6: Solve problem (2.44). The optimal solution is $z^{(7)} = \binom{78}{50}$. Since (2.40) holds, include the new p-efficient point into (P_1), define (P_2) with (P_2) with $V_7 = (V_6, \binom{78}{50})$.

Iteration 7: Solve problem (2.44). The optimal solution is $z^{(8)} = \binom{68}{45}$. Since (2.40) holds, include the new p-efficient point into (P_1), define (P_2) with (P_2) with $V_8 = (V_7, \binom{68}{45})$.

Iteration 8: Solve problem (2.44). The optimal solution is $z^{(9)} = \binom{73}{60}$. Since (2.40) holds, include the new p-efficient point into (P_1), define (P_2) with (P_2) with $V_9 = (V_8, \binom{73}{60})$.

Iteration 9: Solve problem (2.44). Solve problem (2.44). The optimal solution is $z^{(10)} = \binom{58}{60}$. Since (2.40) holds, include the new p-efficient point into (P_1), define (P_2) with (P_2) with $V_10 = (V_9, \binom{58}{60})$.

Step 5. Equation (2.40) has equal values on both sides of the inequality therefore algorithm terminates. Optimal solution is obtained;
$x = \begin{bmatrix} 89 & 100 & 44.785 & 100 & 100 & 87 & 34 & 45.215 \end{bmatrix}^T$
with optimum value= 1463.

If we solve the same problem by using the existing multiple choice knapsack solution algorithms, we obtain the same optimal solution with following p-efficient points:
$\binom{78}{45}, \binom{63}{50}, \binom{68}{45}, \binom{78}{40}, \binom{63}{45}, \binom{63}{60}, \binom{58}{50}, \binom{58}{60}$.

Example This example is the same 8 node network in Figure 2.4, where all nodes have random demands with binomial distribution on arithmetic sequences. The number of non-eliminated 161 inequalities is (see Appendix B), those are the stochastic constraints. Table 2.6 provides us with the possible values of the random demands of Node 1 thorough Node 8. The associated probability distributions can be found in Table 2.7.

All eight distributions are binomial hence all discrete functions F_i, $i = 1, 2, 3, 4, 5, 6, 7, 8$ are

Table 2.6: Possible values of the random demands

ξ_1	34	39	44	49	54	59	64	69	74	79
ξ_2	33	38	43	48	53	58	63	68	73	78
ξ_3	17	22	27	32	37	42	47	52	57	62
ξ_4	33	38	43	48	53	58	63	68	73	78
ξ_5	15	20	25	30	35	40	45	50	55	60
ξ_6	10	15	20	25	30	35	40	45	50	55
ξ_7	15	20	25	30	35	40	45	50	55	60
ξ_8	25	30	35	40	45	50	55	60	65	70

Table 2.7: Associated probabilities (Binomial) c.d.f.

$p(1)$ $(n=9,\ p=0.4)$	0.01	0.0704	0.2316	0.4824	0.7332	0.9004	0.9747	0.9959	0.9994	1
$p(2)$ $(n=9,\ p=0.45)$	0.0046	0.0385	0.1494	0.3612	0.6212	0.8339	0.9499	0.9905	0.9988	1
$p(3)$ $(n=9,\ p=0.5)$	0.0019	0.0194	0.0897	0.2537	0.4997	0.7457	0.9097	0.98	0.9975	1
$p(4)$ $(n=9,\ p=0.6)$	0.0002	0.0037	0.0249	0.0992	0.2664	0.5172	0.768	0.9292	0.9896	1
$p(5)$ $(n=9,\ p=0.48)$	0.0027	0.0257	0.1109	0.2945	0.5488	0.7835	0.9279	0.985	0.9981	1
$p(6)$ $(n=9,\ p=0.35)$	0.0207	0.121	0.3371	0.6087	0.828	0.9461	0.9885	0.9982	0.9995	1
$p(7)$ $(n=9,\ p=0.42)$	0.0074	0.0558	0.196	0.4329	0.6902	0.8765	0.9664	0.9943	0.9993	1
$p(8)$ $(n=9,\ p=0.38)$	0.0135	0.0881	0.2711	0.5329	0.7735	0.921	0.9812	0.997	0.9994	1

logconcave in the supports of ξ_i, $i = 1, 2, 3, 4, 5, 6, 7, 8$, respectively. The Stochastic Programming Problem to be solved is:

$$\min c^T x$$

$$\text{subject to}$$

$$P(y(S, \bar{S})) \geq d(S),\ \ (S)\ \text{non-eliminated}) \geq 0.95,\ \ S \subset N,$$

$$\text{where } N \text{ is the collection of the nodes with random demand,}$$

$$\text{and } d(S) = \textstyle\sum_{i \in S}(\xi_i - x_i),\ \ \text{for } S \subset N.$$

$$l_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},\ u_i = \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \\ 100 \end{bmatrix},\ c = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 2 \\ 1 \\ 1 \\ 7 \\ 4 \end{bmatrix}$$

and $x = (x_1,\ x_2,\ x_3,\ x_4, x_5,\ x_6,\ x_7,\ x_8)^T$ is the decision vector. The solution steps are the following.

Step 1. Rewrite problem in the form of (6.2).

Step 2. $V_1 = (z^{(1)}) = \begin{pmatrix} 64 \\ 68 \\ 57 \\ 73 \\ 55 \\ 45 \\ 55 \\ 65 \end{pmatrix}$.

Step 3. Set up and solve problem (P_0) by a method that produces primal-dual feasible (optimal) basis. Optimal basis$x = \begin{bmatrix} 74 & 60 & 40 & 67 & 86.1799 & 56.8201 & 47 & 51 \end{bmatrix}^T$ with optimal value= 1298.

Let α designate the optimal dual vector.

Step 4.

Iteration 1:Solve problem (2.44). The optimal solution is $z^{(2)} = \begin{pmatrix} 64 \\ 73 \\ 62 \\ 78 \\ 50 \\ 45 \\ 50 \\ 65 \end{pmatrix}$. Since (2.40) holds, include the new p-efficient point (2.31) that we obtain for the random vector (2.29), into (P_1), define (P_2) with $V_2 = \left(V_1, \begin{pmatrix} 64 \\ 73 \\ 62 \\ 78 \\ 50 \\ 45 \\ 50 \\ 65 \end{pmatrix} \right)$.

Iteration 2: Solve problem (2.44). The optimal solution is $z^{(3)} = \begin{pmatrix} 69 \\ 68 \\ 57 \\ 78 \\ 50 \\ 40 \\ 55 \\ 60 \end{pmatrix}$. Since (2.40) holds, include the new p-efficient point (2.31) that we obtain for the random vector (2.29), into (P_1), define (P_2) with $V_3 = \left(V_2, \begin{pmatrix} 69 \\ 68 \\ 57 \\ 78 \\ 50 \\ 40 \\ 55 \\ 60 \end{pmatrix} \right)$.

Iteration 3: Solve problem (2.44). The optimal solution is $z^{(4)} = \begin{pmatrix} 59 \\ 73 \\ 62 \\ 73 \\ 55 \\ 40 \\ 55 \\ 55 \end{pmatrix}$. Since (2.40) holds, include the new p-efficient point (2.31) that we obtain for the random vector (2.29), into (P_1), define (P_2) with $V_4 = \left(V_3, \begin{pmatrix} 59 \\ 73 \\ 62 \\ 73 \\ 55 \\ 40 \\ 55 \\ 55 \end{pmatrix} \right)$.

Iteration 4:Solve problem (2.44). The optimal solution is $z^{(5)} = \begin{pmatrix} 74 \\ 73 \\ 62 \\ 73 \\ 55 \\ 40 \\ 55 \\ 55 \end{pmatrix}$. Since (2.40) holds, include the new p-efficient point (2.31) that we obtain for the random vector (2.29), into (P_1), define (P_2) with $V_5 = \left(V_4, \begin{pmatrix} 74 \\ 73 \\ 62 \\ 73 \\ 55 \\ 40 \\ 55 \\ 55 \end{pmatrix} \right)$.

Iteration 5: :Solve problem (2.44). The optimal solution is $z^{(6)} = \begin{pmatrix} 79 \\ 73 \\ 52 \\ 73 \\ 60 \\ 40 \\ 50 \\ 60 \end{pmatrix}$. Since (2.40) holds, include the new p-efficient point (2.31) that we obtain for the random vector (2.29), into

(P_1), define (P_2) with $V_6 = \left(V_5, \begin{pmatrix} 79 \\ 73 \\ 52 \\ 73 \\ 60 \\ 40 \\ 50 \\ 60 \end{pmatrix} \right)$.

Iteration 6: :Solve problem (2.44). The optimal solution is $z^{(7)} = \begin{pmatrix} 74 \\ 68 \\ 57 \\ 73 \\ 50 \\ 50 \\ 50 \\ 60 \end{pmatrix}$. Since (2.40) holds, include the new p-efficient point (2.31) that we obtain for the random vector (2.29), into (P_1), define (P_2) with $V_7 = \left(V_6, \begin{pmatrix} 74 \\ 68 \\ 57 \\ 73 \\ 50 \\ 50 \\ 50 \\ 60 \end{pmatrix} \right)$.

Step 5.Equation (2.40) has equal values on both sides of the inequality therefore algorithm terminates. Optimal solution is obtained;

$x = \begin{bmatrix} 69 & 60 & 40 & 67 & 81.0192 & 61.9808 & 42 & 46 \end{bmatrix}^T$

with optimal value= 1233.

If we solve the same problem by using the existing multiple choice knapsack solution algorithms, we obtain the same optimal solution with following p-efficient points:

$\begin{pmatrix} 64 \\ 68 \\ 57 \\ 73 \\ 55 \\ 45 \\ 55 \\ 65 \end{pmatrix}, \begin{pmatrix} 69 \\ 68 \\ 57 \\ 78 \\ 50 \\ 40 \\ 55 \\ 60 \end{pmatrix}, \begin{pmatrix} 79 \\ 78 \\ 52 \\ 73 \\ 60 \\ 40 \\ 50 \\ 60 \end{pmatrix}, \begin{pmatrix} 74 \\ 73 \\ 62 \\ 73 \\ 55 \\ 40 \\ 55 \\ 55 \end{pmatrix}, \begin{pmatrix} 69 \\ 68 \\ 62 \\ 78 \\ 60 \\ 40 \\ 50 \\ 55 \end{pmatrix}, \begin{pmatrix} 74 \\ 68 \\ 57 \\ 73 \\ 50 \\ 50 \\ 50 \\ 60 \end{pmatrix}.$

Remark The condition in Theorem 2.2.1 holds true for the p-efficient points encountered in the solution algorithm in examples 1 and 2.

Remark All the p-efficient points for the second example can be found in Appendix C.

2.8 Appendix A

$$\min\left\{c_1^T x + c_2^T y + 0^T u + 0^T w + 0^T \lambda\right\}$$

subject to

$$\left(\begin{array}{ccccccccccccccccccccccccccccccccccccccccccccc}
1&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-10&-15\\
0&1&0&0&0&1&1&1&1&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-9&-9\\
0&0&1&0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-12&-8\\
0&0&0&1&0&0&0&0&1&1&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-9&-9\\
0&0&0&0&1&0&1&0&0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-15&-15\\
1&1&0&0&0&0&1&1&1&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-19&-24\\
1&0&1&0&0&1&1&0&1&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-21&-17\\
0&1&0&1&0&1&1&1&0&1&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-18&-18\\
0&1&0&0&1&1&0&1&1&1&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-24&-24\\
0&0&0&1&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-24&-24\\
1&1&1&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-31&-32\\
1&1&0&1&0&0&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-28&-33\\
1&1&0&0&1&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-34&-39\\
0&1&1&1&0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-30&-26\\
0&1&1&0&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-36&-32\\
0&1&0&1&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-40&-33\\
0&0&1&1&1&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-46&-32\\
1&1&1&1&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-43&-41\\
1&1&1&0&1&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-46&-47\\
1&1&0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&-43&-48\\
1&0&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&-46&-47\\
0&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&-45&-41\\
1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&-55&-56\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1
\end{array}\right)
\begin{pmatrix} x \\ y \\ u \\ w \\ \lambda \end{pmatrix}
=
\begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\10\\10\\10\\10\\10\\10\\10\\10\\10\\10\\1 \end{pmatrix}
\tag{2.59}$$

$$x \geq 0,\ \ y \geq 0,\ \ u \geq 0,\ \ w \geq 0,\ \ \lambda \geq 0,$$

2.9 Appendix B

The number of eliminated inequalities by network topology: 94. Their numbers are: 11, 12, 13, 14, 15, 19, 20, 22, 24, 25, 26, 29, 33, 36, 40, 41, 43, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 60, 61, 65, 70, 72, 73, 74, 75, 76, 79, 80, 81, 83, 91, 98, 99, 102, 103, 104, 112, 118, 120, 125, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 144, 145, 146, 148, 150, 155, 159, 160, 165, 178, 184, 185, 186, 187, 188, 189, 190, 193, 194, 195, 203, 209, 211, 216, 226, 228, 241.

Table 2.8: Gale–Hoffman inequalities for the 8-node network ($2^8 - 1 = 255$ inequalities)

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
1	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0
4	0	0	0	1	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	1
9	1	1	0	0	0	0	0	0
10	1	0	1	0	0	0	0	0
11	1	0	0	1	0	0	0	0
12	1	0	0	0	1	0	0	0
13	1	0	0	0	0	1	0	0
14	1	0	0	0	0	0	1	0
15	1	0	0	0	0	0	0	1
16	0	1	1	0	0	0	0	0
17	0	1	0	1	0	0	0	0
18	0	1	0	0	1	0	0	0
19	0	1	0	0	0	1	0	0
20	0	1	0	0	0	0	1	0
21	0	1	0	0	0	0	0	1

Continued on next page

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
22	0	0	1	1	0	0	0	0
23	0	0	1	0	1	0	0	0
24	0	0	1	0	0	1	0	0
25	0	0	1	0	0	0	1	0
26	0	0	1	0	0	0	0	1
27	0	0	0	1	1	0	0	0
28	0	0	0	1	0	1	0	0
29	0	0	0	1	0	0	1	0
30	0	0	0	1	0	0	0	1
31	0	0	0	0	1	1	0	0
32	0	0	0	0	1	0	1	0
33	0	0	0	0	1	0	0	1
34	0	0	0	0	0	1	1	0
35	0	0	0	0	0	1	0	1
36	0	0	0	0	0	0	1	1
37	1	1	1	0	0	0	0	0
38	1	1	0	1	0	0	0	0
39	1	1	0	0	1	0	0	0
40	1	1	0	0	0	1	0	0
41	1	1	0	0	0	0	1	0
42	1	1	0	0	0	0	0	1
43	1	0	1	1	0	0	0	0
44	1	0	1	0	1	0	0	0

Continued on next page

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
45	1	0	1	0	0	1	0	0
46	1	0	1	0	0	0	1	0
47	1	0	1	0	0	0	0	1
48	1	0	0	1	1	0	0	0
49	1	0	0	1	0	1	0	0
50	1	0	0	1	0	0	1	0
51	1	0	0	1	0	0	0	1
52	1	0	0	0	1	1	0	0
53	1	0	0	0	1	0	1	0
54	1	0	0	0	1	0	0	1
55	1	0	0	0	0	1	1	0
56	1	0	0	0	0	1	0	1
57	1	0	0	0	0	0	1	1
58	0	1	1	1	0	0	0	0
59	0	1	1	0	1	0	0	0
60	0	1	1	0	0	1	0	0
61	0	1	1	0	0	0	1	0
62	0	1	1	0	0	0	0	1
63	0	1	0	1	1	0	0	0
64	0	1	0	1	0	1	0	0
65	0	1	0	1	0	0	1	0
66	0	1	0	1	0	0	0	1
67	0	1	0	0	1	1	0	0

Continued on next page

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
68	0	1	0	0	1	0	1	0
69	0	1	0	0	1	0	0	1
70	0	1	0	0	0	1	1	0
71	0	1	0	0	0	1	0	1
72	0	1	0	0	0	0	1	1
73	0	0	1	0	0	0	1	1
74	0	0	1	0	0	1	0	1
75	0	0	1	0	0	1	1	0
76	0	0	1	0	1	0	0	1
77	0	0	1	0	1	0	1	0
78	0	0	1	0	1	1	0	0
79	0	0	1	1	0	0	0	1
80	0	0	1	1	0	0	1	0
81	0	0	1	1	0	1	0	0
82	0	0	1	1	1	0	0	0
83	0	0	0	1	0	0	1	1
84	0	0	0	1	0	1	0	1
85	0	0	0	1	0	1	1	0
86	0	0	0	1	1	0	0	1
87	0	0	0	1	1	0	1	0
88	0	0	0	1	1	1	0	0
89	0	0	0	0	1	1	1	0
90	0	0	0	0	1	1	0	1

Continued on next page

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
91	0	0	0	0	1	0	1	1
92	0	0	0	0	0	1	1	1
93	0	0	0	0	1	1	1	1
94	0	0	0	1	0	1	1	1
95	0	0	0	1	1	0	1	1
96	0	0	0	1	1	1	0	1
97	0	0	0	1	1	1	1	0
98	0	0	1	0	0	1	1	1
99	0	0	1	0	1	0	1	1
100	0	0	1	0	1	1	0	1
101	0	0	1	0	1	1	1	0
102	0	0	1	1	0	0	1	1
103	0	0	1	1	0	1	0	1
104	0	0	1	1	0	1	1	0
105	0	0	1	1	1	0	0	1
106	0	0	1	1	1	0	1	0
107	0	0	1	1	1	1	0	0
108	0	1	0	0	0	1	1	1
109	0	1	0	0	1	0	1	1
110	0	1	0	0	1	1	0	1
111	0	1	0	0	1	1	1	0
112	0	1	0	1	0	0	1	1
113	0	1	0	1	0	1	0	1

Continued on next page

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
114	0	1	0	1	0	1	1	0
115	0	1	0	1	1	0	0	1
116	0	1	0	1	1	0	1	0
117	0	1	0	1	1	1	0	0
118	0	1	1	0	0	0	1	1
119	0	1	1	0	0	1	0	1
120	0	1	1	0	0	1	1	0
121	0	1	1	0	1	0	0	1
122	0	1	1	0	1	0	1	0
123	0	1	1	0	1	1	0	0
124	0	1	1	1	0	0	0	1
125	0	1	1	1	0	0	1	0
126	0	1	1	1	0	1	0	0
127	0	1	1	1	1	0	0	0
128	1	0	0	0	0	1	1	1
129	1	0	0	0	1	0	1	1
130	1	0	0	0	1	1	0	1
131	1	0	0	0	1	1	1	0
132	1	0	0	1	0	0	1	1
133	1	0	0	1	0	1	0	1
134	1	0	0	1	0	1	1	0
135	1	0	0	1	1	0	0	1
136	1	0	0	1	1	0	1	0

Continued on next page

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
137	1	0	0	1	1	1	0	0
138	1	0	1	0	0	0	1	1
139	1	0	1	0	0	1	0	1
140	1	0	1	0	0	1	1	0
141	1	0	1	0	1	0	0	1
142	1	0	1	0	1	0	1	0
143	1	0	1	0	1	1	0	0
144	1	0	1	1	0	0	0	1
145	1	0	1	1	0	0	1	0
146	1	0	1	1	0	1	0	0
147	1	0	1	1	1	0	0	0
148	1	1	0	0	0	0	1	1
149	1	1	0	0	0	1	0	1
150	1	1	0	0	0	1	1	0
151	1	1	0	0	1	0	0	1
152	1	1	0	0	1	0	1	0
153	1	1	0	0	1	1	0	0
154	1	1	0	1	0	0	0	1
155	1	1	0	1	0	0	1	0
156	1	1	0	1	0	1	0	0
157	1	1	0	1	1	0	0	0
158	1	1	1	0	0	0	0	1
159	1	1	1	0	0	0	1	0

Continued on next page

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
160	1	1	1	0	0	1	0	0
161	1	1	1	0	1	0	0	0
162	1	1	1	1	0	0	0	0
163	0	0	0	1	1	1	1	1
164	0	0	1	0	1	1	1	1
165	0	0	1	1	0	1	1	1
166	0	0	1	1	1	0	1	1
167	0	0	1	1	1	1	0	1
168	0	0	1	1	1	1	1	0
169	0	1	0	0	1	1	1	1
170	0	1	0	1	0	1	1	1
171	0	1	0	1	1	0	1	1
172	0	1	0	1	1	1	0	1
173	0	1	0	1	1	1	1	0
174	0	1	1	0	0	1	1	1
175	0	1	1	0	1	0	1	1
176	0	1	1	0	1	1	0	1
177	0	1	1	0	1	1	1	0
178	0	1	1	1	0	0	1	1
179	0	1	1	1	0	1	0	1
180	0	1	1	1	0	1	1	0
181	0	1	1	1	1	0	0	1
182	0	1	1	1	1	0	1	0

Continued on next page

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
183	0	1	1	1	1	1	0	0
184	1	0	0	0	1	1	1	1
185	1	0	0	1	0	1	1	1
186	1	0	0	1	1	0	1	1
187	1	0	0	1	1	1	0	1
188	1	0	0	1	1	1	1	0
189	1	0	1	0	0	1	1	1
190	1	0	1	0	1	0	1	1
191	1	0	1	0	1	1	0	1
192	1	0	1	0	1	1	1	0
193	1	0	1	1	0	0	1	1
194	1	0	1	1	0	1	0	1
195	1	0	1	1	0	1	1	0
196	1	0	1	1	1	0	0	1
197	1	0	1	1	1	0	1	0
198	1	0	1	1	1	1	0	0
199	1	1	0	0	0	1	1	1
200	1	1	0	0	1	0	1	1
201	1	1	0	0	1	1	0	1
202	1	1	0	0	1	1	1	0
203	1	1	0	1	0	0	1	1
204	1	1	0	1	0	1	0	1
205	1	1	0	1	0	1	1	0

Continued on next page

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
206	1	1	0	1	1	0	0	1
207	1	1	0	1	1	0	1	0
208	1	1	0	1	1	1	0	0
209	1	1	1	0	0	0	1	1
210	1	1	1	0	0	1	0	1
211	1	1	1	0	0	1	1	0
212	1	1	1	0	1	0	0	1
213	1	1	1	0	1	0	1	0
214	1	1	1	0	1	1	0	0
215	1	1	1	1	0	0	0	1
216	1	1	1	1	0	0	1	0
217	1	1	1	1	0	1	0	0
218	1	1	1	1	1	0	0	0
219	0	0	1	1	1	1	1	1
220	0	1	0	1	1	1	1	1
221	0	1	1	0	1	1	1	1
222	0	1	1	1	0	1	1	1
223	0	1	1	1	1	0	1	1
224	0	1	1	1	1	1	0	1
225	0	1	1	1	1	1	1	0
226	1	0	0	1	1	1	1	1
227	1	0	1	0	1	1	1	1
228	1	0	1	1	0	1	1	1

Continued on next page

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
229	1	0	1	1	1	0	1	1
230	1	0	1	1	1	1	0	1
231	1	0	1	1	1	1	1	0
232	1	1	0	0	1	1	1	1
233	1	1	0	1	0	1	1	1
234	1	1	0	1	1	0	1	1
235	1	1	0	1	1	1	0	1
236	1	1	0	1	1	1	1	0
237	1	1	1	0	0	1	1	1
238	1	1	1	0	1	0	1	1
239	1	1	1	0	1	1	0	1
240	1	1	1	0	1	1	1	0
241	1	1	1	1	0	0	1	1
242	1	1	1	1	0	1	0	1
243	1	1	1	1	0	1	1	0
244	1	1	1	1	1	0	0	1
245	1	1	1	1	1	0	1	0
246	1	1	1	1	1	1	0	0
247	1	1	1	1	1	1	1	0
248	1	1	1	1	1	1	0	1
249	1	1	1	1	1	0	1	1
250	1	1	1	1	0	1	1	1
251	1	1	1	0	1	1	1	1

Continued on next page

Number	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
252	1	1	0	1	1	1	1	1
253	1	0	1	1	1	1	1	1
254	0	1	1	1	1	1	1	1
255	1	1	1	1	1	1	1	1

2.10 Appendix C

Table 2.10: All the p-level efficient points for the 8-node example

Plep 1	Plep 2	Plep 3	Plep 4	Plep 5	Plep 6	Plep 7	Plep 8	Plep 9	Plep 10	Plep 11	Plep 12
64	64	69	69	69	69	69	59	59	79	79	79
68	73	68	68	73	78	78	68	73	73	78	78
57	62	57	62	57	52	57	57	62	52	52	62
73	78	78	78	73	78	73	73	73	73	73	78
55	50	50	60	55	55	50	50	55	55	60	60
45	45	40	40	40	55	40	50	40	40	40	55
55	50	55	50	55	50	50	50	55	50	50	60
65	65	60	55	55	55	70	60	55	60	60	70
132	137	137	137	142	147	147	127	132	152	157	157
121	126	126	131	126	121	126	116	121	131	131	141
125	135	125	130	130	130	135	125	135	125	130	140
141	151	146	146	146	156	151	141	146	146	151	156
123	123	118	128	128	133	128	118	128	128	138	138
133	138	128	123	128	133	148	128	128	133	138	148
112	112	107	122	112	107	107	107	117	107	112	122
128	128	128	138	128	133	123	123	128	128	133	138
118	123	118	118	113	133	113	123	113	113	113	133
138	143	138	133	128	133	143	133	128	133	133	148
100	95	90	100	95	110	90	100	95	95	100	115

Continued on next page

Plep 1	Plep 2	Plep 3	Plep 4	Plep 5	Plep 6	Plep 7	Plep 8	Plep 9	Plep 10	Plep 11	Plep 12
110	100	105	110	110	105	100	100	110	105	110	120
100	95	95	90	95	105	90	100	95	90	90	115
110	110	100	95	95	110	110	110	95	100	100	125
189	199	194	199	199	199	204	184	194	204	209	219
205	215	215	215	215	225	220	200	205	225	230	235
187	187	187	197	197	202	197	177	187	207	217	217
197	202	197	192	197	202	217	187	187	212	217	227
176	176	176	191	181	176	176	166	176	186	191	201
198	213	203	208	203	208	208	198	208	198	203	218
180	185	175	190	185	185	185	175	190	180	190	200
190	200	185	185	185	185	205	185	190	185	190	210
196	201	196	206	201	211	201	191	201	201	211	216
186	196	186	186	186	211	191	191	186	186	191	211
206	216	206	201	201	211	221	201	201	206	211	226
168	168	158	168	168	188	168	168	168	168	178	193
178	173	173	178	183	183	178	168	183	178	188	198
188	188	178	183	183	188	198	178	183	188	198	208
178	183	168	163	168	188	188	178	168	173	178	203
185	190	185	200	185	185	180	180	190	180	185	200
157	157	147	162	152	162	147	157	157	147	152	177
167	162	162	172	167	157	157	157	172	157	162	182
173	173	168	178	168	188	163	173	168	168	173	193

Continued on next page

Plep 1	Plep 2	Plep 3	Plep 4	Plep 5	Plep 6	Plep 7	Plep 8	Plep 9	Plep 10	Plep 11	Plep 12
183	178	183	188	183	183	173	173	183	178	183	198
193	193	188	193	183	188	193	183	183	188	193	208
173	173	173	168	168	183	163	173	168	163	163	193
183	188	178	173	168	188	183	183	168	173	173	203
155	145	145	150	150	160	140	150	150	145	150	175
165	160	150	155	150	165	160	160	150	155	160	185
165	160	155	145	150	160	160	160	150	150	150	185
262	277	272	277	272	277	277	257	267	277	282	297
244	249	244	259	254	254	254	234	249	259	269	279
254	264	254	254	254	254	274	244	249	264	269	289
260	265	265	275	270	280	270	250	260	280	290	295
250	260	255	255	255	280	260	250	245	265	270	290
270	280	275	270	270	280	290	260	260	285	290	305
232	232	227	237	237	257	237	227	227	247	257	272
242	237	242	247	252	252	247	227	242	257	267	277
252	252	247	252	252	257	267	237	242	267	277	287
242	247	237	232	237	257	257	237	227	252	257	282
249	254	254	269	254	254	249	239	249	259	264	279
221	221	216	231	221	231	216	216	216	226	231	256
231	226	231	241	236	226	226	216	231	236	241	261
253	263	253	268	258	263	258	248	263	253	263	278
243	258	243	248	243	263	248	248	248	238	243	273

Continued on next page

Plep 1	Plep 2	Plep 3	Plep 4	Plep 5	Plep 6	Plep 7	Plep 8	Plep 9	Plep 10	Plep 11	Plep 12
263	278	263	263	258	263	278	258	263	258	263	288
225	230	215	230	225	240	225	225	230	220	230	255
235	235	230	240	240	235	235	225	245	230	240	260
245	250	235	245	240	240	255	235	245	240	250	270
235	245	225	225	225	240	245	235	230	225	230	265
241	246	236	246	241	266	241	241	241	241	251	271
251	251	251	256	256	261	251	241	256	251	261	276
261	266	256	261	256	266	271	251	256	261	271	286
241	246	241	236	241	261	241	241	241	236	241	271
251	261	246	241	241	266	261	251	241	246	251	281
223	218	213	218	223	238	218	218	223	218	228	253
233	233	218	223	223	243	238	228	223	228	238	263
243	238	233	233	238	238	248	228	238	238	248	268
233	233	223	213	223	238	238	228	223	223	228	263
230	235	225	240	225	240	220	230	230	220	225	255
240	240	240	250	240	235	230	230	245	230	235	260
250	255	245	255	240	240	250	240	245	240	245	270
212	207	202	212	207	212	197	207	212	197	202	237
222	222	207	217	207	217	217	217	212	207	212	247
228	223	223	228	223	238	213	223	223	218	223	253
238	238	228	233	223	243	233	233	223	228	233	263
248	243	243	243	238	238	243	233	238	238	243	268

Continued on next page

Plep 1	Plep 2	Plep 3	Plep 4	Plep 5	Plep 6	Plep 7	Plep 8	Plep 9	Plep 10	Plep 11	Plep 12
238	238	233	223	223	238	233	233	223	223	223	263
220	210	205	205	205	215	210	210	205	205	210	245
317	327	322	337	327	332	327	307	322	332	342	357
307	322	312	317	312	332	317	307	307	317	322	352
327	342	332	332	327	332	347	317	322	337	342	367
289	294	284	299	294	309	294	284	289	299	309	334
299	299	299	309	309	304	304	284	304	309	319	339
309	314	304	314	309	309	324	294	304	319	329	349
299	309	294	294	294	309	314	294	289	304	309	344
305	310	305	315	310	335	310	300	300	320	330	350
315	315	320	325	325	330	320	300	315	330	340	355
325	330	325	330	325	335	340	310	315	340	350	365
305	310	310	305	310	330	310	300	300	315	320	350
315	325	315	310	310	335	330	310	300	325	330	360
287	282	282	287	292	307	287	277	282	297	307	332
297	297	287	292	292	312	307	287	282	307	317	342
307	302	302	302	307	307	317	287	297	317	327	347
297	297	292	282	292	307	307	287	282	302	307	342
294	299	294	309	294	309	289	289	289	299	304	334
304	304	309	319	309	304	299	289	304	309	314	339
314	319	314	324	309	309	319	299	304	319	324	349
276	271	271	281	276	281	266	266	271	276	281	316

Continued on next page

Plep 1	Plep 2	Plep 3	Plep 4	Plep 5	Plep 6	Plep 7	Plep 8	Plep 9	Plep 10	Plep 11	Plep 12
286	286	276	286	276	286	286	276	271	286	291	326
298	308	293	308	298	318	298	298	303	293	303	333
308	313	308	318	313	313	308	298	318	303	313	338
318	328	313	323	313	318	328	308	318	313	323	348
298	308	298	298	298	313	298	298	303	288	293	333
308	323	303	303	298	318	318	308	303	298	303	343
280	280	270	280	280	290	275	275	285	270	280	315
290	295	275	285	280	295	295	285	285	280	290	325
300	300	290	295	295	290	305	285	300	290	300	330
290	295	280	275	280	290	295	285	285	275	280	325
296	296	291	296	296	316	291	291	296	291	301	331
306	311	296	301	296	321	311	301	296	301	311	341
316	316	311	311	311	316	321	301	311	311	321	346
306	311	301	291	296	316	311	301	296	296	301	341
288	283	273	273	278	293	288	278	278	278	288	323
285	285	280	290	280	290	270	280	285	270	275	315
295	300	285	295	280	295	290	290	285	280	285	325
305	305	300	305	295	290	300	290	300	290	295	330
277	272	262	267	262	267	267	267	267	257	262	307
293	288	283	283	278	293	283	283	278	278	283	323
362	372	362	377	367	387	367	357	362	372	382	412
372	377	377	387	382	382	377	357	377	382	392	417

Continued on next page

Plep 1	Plep 2	Plep 3	Plep 4	Plep 5	Plep 6	Plep 7	Plep 8	Plep 9	Plep 10	Plep 11	Plep 12
382	392	382	392	382	387	397	367	377	392	402	427
362	372	367	367	367	382	367	357	362	367	372	412
372	387	372	372	367	387	387	367	362	377	382	422
344	344	339	349	349	359	344	334	344	349	359	394
354	359	344	354	349	364	364	344	344	359	369	404
364	364	359	364	364	359	374	344	359	369	379	409
354	359	349	344	349	359	364	344	344	354	359	404
360	360	360	365	365	385	360	350	355	370	380	410
370	375	365	370	365	390	380	360	355	380	390	420
380	380	380	380	380	385	390	360	370	390	400	425
370	375	370	360	365	385	380	360	355	375	380	420
352	347	342	342	347	362	357	337	337	357	367	402
349	349	349	359	349	359	339	339	344	349	354	394
359	364	354	364	349	364	359	349	344	359	364	404
369	369	369	374	364	359	369	349	359	369	374	409
341	336	331	336	331	336	336	326	326	336	341	386
353	358	348	358	353	368	348	348	358	343	353	393
363	373	353	363	353	373	368	358	358	353	363	403
373	378	368	373	368	368	378	358	373	363	373	408
363	373	358	353	353	368	368	358	358	348	353	403
345	345	330	335	335	345	345	335	340	330	340	385
361	361	351	351	351	371	361	351	351	351	361	401

Continued on next page

Plep 1	Plep 2	Plep 3	Plep 4	Plep 5	Plep 6	Plep 7	Plep 8	Plep 9	Plep 10	Plep 11	Plep 12
350	350	340	345	335	345	340	340	340	330	335	385
417	422	417	427	422	437	417	407	417	422	432	472
427	437	422	432	422	442	437	417	417	432	442	482
437	442	437	442	437	437	447	417	432	442	452	487
427	437	427	422	422	437	437	417	417	427	432	482
409	409	399	404	404	414	414	394	399	409	419	464
425	425	420	420	420	440	430	410	410	430	440	480
414	414	409	414	404	414	409	399	399	409	414	464
418	423	408	413	408	423	418	408	413	403	413	463
482	487	477	482	477	492	487	467	472	482	492	542

Chapter 3

Optimal Capacity Design under k-out-of-n and Consecutive k-out-of-n type Probabilistic Constraints

3.1 Introduction

In this chapter, we formulate and solve probabilistic constrained stochastic programming problems, where we prescribe lower and upper bounds for k-out-of-n and consecutive-k-out-of-n reliabilities in the form of probabilistic constraints. The problem is to determine the optimal capacity of a water release, or a reserve of a bank, to satisfy irrigation demand or demand of financial transactions, i.e., a reliability constraint where the reliability is one of the abovementioned type. For the non-consecutive type reliability problem, normal and gamma distributions are used for supply and demand values, respectively. By using the properties of the standard gamma distribution, the reliability constraint is written as an equation, which can then be solved by simulation. For the k-consecutive case, different probability bounds are used in order to solve the reliability equation. To create lower and upper bounds for the reliability constraint, discrete binomial problems are used, which are constructed as linear programming (LP) problems. S_1, S_2, S_3 sharp lower bounds, Hunter's upper bound and Cherry tree upper bound are calculated to obtain the desired probability level for the reliability constraint. A bi-section algorithm is later applied to find the optimal capacity level.

3.2 Formulation of the Problem

There will be two types of problems modeled in this section. In the first model, system is allowed to fail a demand for at most any k periods out of n periods while solving for the optimal capacity. In the second model, same objective function is solved while allowing at most consecutive k periods out of n periods without satisfying the demand. Following notation is used for both problems:

ξ_i	supply in the i^{th} period, random variable
γ_i	demand in the i^{th}period, random variable
δ_i	additional supply (i.e. rain in water reservoir design) in the i^{th} period, random variable
x_i	if demand is satisfied in the i^{th} period, 1 o.w., 0, boolean decision variable
m	capacity, decision variable
m_{opt}	value of the optimum capacity
M	upper limit for the capacity, predefined number
k	number of permitted days of not meeting the demand, predefined number
p	probability level of reliability, predefined number

First Model: Failing to meet the demand is allowed for at most k periods

Problem to be solved:

$$\begin{gathered}
\min m \\
\text{subject to} \\
P\{(\min(\xi_i, m) + \delta_i) \geq x_i \gamma_i, \\
i = 1, \dots, n, \;\; x_1 + \dots + x_n \geq n - k\} \geq p \\
x_i \in \{0, 1\} \;\; i = 1, \dots, n \\
0 \leq m \leq M
\end{gathered} \tag{3.1}$$

Second Model: Failing to meet the demand is allowed for at most k consecutive periods

Problem to be solved:

$$\begin{gathered}
\min m \\
\text{subject to} \\
P\{(\min(\xi_i, m) + \delta_i) \geq x_i \gamma_i \\
i = 1, \ldots, n, \;\; x_i + \cdots + x_{i+k-1} \geq 1, \;\; i = 1, \ldots, n-k+1\} \geq p \\
x_i \in \{0, 1\} \;\; i = 1, \ldots, n \qquad (3.2) \\
0 \leq m \leq M
\end{gathered}$$

3.3 Mathematical Properties of the Reservoir System Design Model

In this section, we prove a convexity theorem for problems (3.1) and (3.2) where there are no discrete variables. The convexity statement is based on the theory of multivariate logconcave measures and functions. In order to make the paper self contained, we recall some facts from logconcavity. First we present two definitions.

Definition A function $f(x) \geq 0$, $x \in R^n$ is logconcave if for every $x, y \in R^n$ and $0 < \lambda < 1$ we have

$$f(\lambda x + (1-\lambda)y) \geq (f(x))^{\lambda} (f(y))^{1-\lambda} \qquad (3.3)$$

Definition A probability measure P is the Borel subsets of R^n is logconcave (Prékopa 1971, 1973a) if for every convex subsets A, B of R^n and $0 < \lambda < 1$ we have

$$P(\lambda A + (1-\lambda)B) \geq (P(A))^{\lambda}(P(B))^{1-\lambda}.$$

A simple consequence of the second definition is that the c.d.f., corresponding to a logconcave probability measure, is logconcave (as a point function). The basic theorem of logconcave measure is the following:

Theorem 3.3.1 *(Prékopa, 1971, 1973a). If the probability measure P is generated by a logconcave p.d.f., then P is a logconcave measure.*

Another theorem that we use in connection with problem (3.1) is the following:

Theorem 3.3.2 *(Prékopa, 1972). If $g_1(x, y), \ldots, g_r(x, y)$ are concave functions in R^{n+q}, where $x \in R^n$, $y \in R^q$ and $\xi \in R^n$ is a random variable that has logconcave distribution, then the function*

$$P(g_i(\xi, y) \geq 0, \;\; i = 1, \ldots, r)$$

is a logconcave function of $y \in R^n$.

A consequence of the above theorem is

Theorem 3.3.3 *If the joint p.d.f. of the random variables $\xi_i, \delta_i, \gamma_i$, $i = 1, \ldots, n$ is logconcave, then for every fixed x,*

$$P(\min(\xi_i, m) + \delta_i \geq x_i\gamma_i, \;\; i = 1, \ldots, n)$$

is logconcave function of ξ_i, δ_i, and γ_i.

Proof Theorem 3.3.1 ensures the logconcavity of the joint distribution of the random variables $\xi_i, \delta_i, \gamma_i, \;\; i = 1, \ldots, n$. On the other hand, if we consider $\xi_i, \delta_i, \gamma_i, \;\; i = 1, \ldots, n$ as deterministic variables, then we can see that the functions

$$\min(\xi_i, m) + \delta_i - x_i\gamma_i, \;\; i = 1, \ldots, n$$

are concave in all these variables and n. By Theorem 3.3.2, the assertion follows.

3.4 Solution of the Problem

3.4.1 Solution of the Model (3.1)

First we present a method to find an upper bound for the optimal solution of model where there are no discrete variables. For the case of I.I.D. $\gamma_1, \ldots, \gamma_n$, where each has gamma distribution with p.d.f.:

$$\frac{\lambda^{\vartheta} z^{\vartheta-1} e^{-\lambda z}}{\tau(\vartheta)}, z > 0 \tag{3.4}$$

we can obtain an upper bound for m_{opt}. For simplicity, we assume that $(\gamma_1, \ldots, \gamma_n)$ is independent of $(\xi_1, \ldots, \xi_n, \delta_1, \ldots, \delta_n)$.

First, we mention that the random variables $\lambda\gamma_1, \ldots, \lambda\gamma_n$ follow the standard gamma distribution, i.e. the distribution with a p.d.f. (3.4), where $\lambda = 1$. The second observation is that the following relations hold:

$$\begin{aligned} & P(\min(\xi_i, m) + \delta_i \geq x_i\gamma_i, \;\; i = 1, \ldots, n) \\ \leq \;\; & P(\sum_{i=1}^{n} [\min(\xi_i, m) + \delta_i] \geq \sum_{i=1}^{n} x_i\gamma_i) \\ = \;\; & P(\sum_{i=1}^{n} \lambda[\min(\xi_i, m) + \delta_i] \geq \sum_{i=1}^{n} x_i\lambda\gamma_i) \end{aligned} \tag{3.5}$$

The distribution of $\sum_{i=1}^{n} x_i \lambda \gamma_i$ is the same as the distribution of $\lambda\gamma_1 \sum_{i=1}^{n} x_i$. In fact, the sum of independent standard gamma random variables is also a standard gamma random variable and the (ϑ) parameter for the sum is also the sum of individual (ϑ) parameters. Thus, we can replace $\sum_{i=1}^{n} x_i \lambda \gamma_i$ by $\lambda\gamma_1 \sum_{i=1}^{n} x_i$ for the last line in (3.5). On the other hand, it is prescribed that $\sum_{i=1}^{n} x_i \geq n-k$, hence we obtain the inequality

$$\begin{aligned} & P \quad (\min(\xi_i, m) + \delta_i \geq x_i \gamma_i, \ \ i = 1, \ldots, n) \\ \leq \ & P \quad \left(\sum_{i=1}^{n} \lambda \left[\min(\xi_i, m) + \delta_i\right] \geq (n-k)\lambda\gamma_1 \right). \end{aligned} \tag{3.6}$$

Inequality (3.6) implies that the optimum value of the problem

$$\begin{gathered} \min m \\ \text{subject to} \\ P\left(\sum_{i=1}^{n} \lambda \left[\min(\xi_i, m) + \delta_i\right] \geq (n-k)\lambda\gamma_1\right) \geq p \\ 0 \leq m \leq M \end{gathered} \tag{3.7}$$

is an upper bound for the optimum value of problem (3.1). On the other hand, if there exists a feasible m in problem (3.7), then, due to the monocity of the constraining function in the first constraint, the optimal solution of problem (3.7) can simply be obtained by solving the equation:

$$P\left(\sum_{i=1}^{n} \lambda \left[\min(\xi_i, m) + \delta_i\right] \geq (n-k)\lambda\gamma_1 \right) = p. \tag{3.8}$$

3.4.2 Solution of the problem (3.2)

The problem (3.2) can be solved in multiple ways; however in this paper we will use bounding techniques. For the sake of computational easiness, we will ignore δ_i. In order to apply bounding methodology, the reliability constraint in model (3.2) will be re-written as follows:

$$\begin{gathered} P\{\min\left((\xi_i, m) \geq \gamma_i x_i\right), \ \ i = 1, \ldots, n, \\ x_i + \cdots + x_{i+k-1} \geq 1, \ \ i = 1, \ldots, n-k+1\} \geq p \\ x_i \in \{0, 1\} \ \ i = 1, \ldots, n \end{gathered}$$

This inequality can also be expressed as:

$$P\{(\xi_i \geq \gamma_i x_i, m \geq \gamma_i x_i),\ \ i = 1, \ldots, n,$$

$$x_i + \cdots + x_{i+k-1} \geq 1,\ \ i = 1, \ldots, n-k+1\} \geq p$$

$$x_i \in \{0, 1\}\ \ i = 1, \ldots, n$$

which claims that, the minimum of the supply and the capacity should be greater than or equal to the demand with a probability p so that the condition for failing to meet the demand for k-consecutive periods can be satisfied. In order to solve this inequality, we will consider k periods starting from the first period, (first period + $k-1$ periods), second period (second period + $k-1$ periods) etc. individually and then we will consider the intersection of these $n-k+1$ events. The intersection of these events will ensure that at least one period from the first period until the k^{th} period and from the second period until the $(k+1)^{th}$ period will have enough supply to meet the demand. When we consider total number of events l, the supply will never fail to meet the demand for the k-consecutive periods out of n total periods.

The event A_l means that starting from the l^{th} period, at least one out of these k periods, there will be sufficient supply. On the other hand, the complimentary event $\overline{A}_l$ indicates that there will not be sufficient supply for the demand during any k periods. A_1 is represented as follows when $k = 7$:

$$A_1 = \ \ (\xi_i \geq \gamma_i,\ \ m \geq \gamma_i) \geq 1,\ \ i = 1, \ldots, 7$$

which implies that for at least 1 period out of 7 periods, there will be enough supply. On the other hand, the $\overline{A}_1$ event is defined as follows:

$$\overline{A}_1 = \ \ (\xi_i \geq \gamma_i,\ \ m \geq \gamma_i) < 1,\ \ i = 1, \ldots, 7$$

which indicates that demand will not be met for the whole 7 periods. Similarly, A_2 event is defined as:

$$A_2 = \ \ (\xi_i \geq \gamma_i,\ \ m \geq \gamma_i) \geq 1,\ \ i = 2, \ldots, 8$$

which implies that for at least 1 period out 7 periods (from second period until the eighth period), there will be sufficient supply whereas the $\overline{A}_2$ event is given as:

$$\overline{A}_2 = \ \ (\xi_i \geq \gamma_i,\ \ m \geq \gamma_i) < 1,\ \ i = 2, \ldots, 8$$

which implies that there will not be sufficient supply for any of the mentioned 7 periods.

The methodology to create A_l events includes creating k consecutive periods, where the demand is met, starting with the l^{th} period till $(l+k-1)^{th}$ period. Here, there will be a total of $n-k+1$ number of events where n is the total number of periods that will be taken into consideration. With this information, the last event is represented as follows:

$$A_{n-k+1} = \ (\xi_i \geq \gamma_i, \ \ m \geq \gamma_i) \geq 1, \ \ i = n-k+1, \ldots, n-1$$

whereas the $\overline{A}_{n-k+1}$ event is:

$$\overline{A}_{n-k+1} = \ (\xi_i \geq \gamma_i, \ \ m \geq \gamma_i) < 1, \ \ i = n-k+1, \ldots, n-1$$

Next, the probability of intersection of all these events, that ensures that the demand is allowed not to be met at most for k-consecutive periods can be written as follows:

$$P(A_1 \cap \cdots \cap A_{n-k+1}) \geq p \tag{3.9}$$

The purpose of expressing the reliability constraint as the intersection of the specially defined events is to be able to apply bounding techniques while searching for the optimum value of the capacity m.

3.4.3 Sharp bounds on the probability

Solving Equation (3.9) is not practically easy and manageable therefore well known bounds for the union of events will be used to define a lower and upper bound to the desired probability which then will be used to determine the optimal capacity.

In order to calculate the lower and upper bounds for the reliability constraint, we provide the description of the bounds in the next section where the same notation and definition of Prékopa (1995) are used. Since all most known bounds are given for the union of the events, we will explain how to convert the union of the events into intersection of the events, which is needed to solve the optimal capacity problem.

Lower bounds, S_1, S_2, S_3 given

Sharp lower bound is given in Prékopa (1995) as follows:

$$P(A_1 \cup \cdots \cup A_n) \geq \frac{i+2n-1}{(i+1)n} S_1 - \frac{2(2i+n-2)}{i(i+1)n} S_2 + \frac{6}{i(i+1)} S_3 \tag{3.10}$$

where

$$i = 1 + \left\lfloor \tfrac{-6S_3+2(n-2)S_2}{-2S_2+(n-1)S_1} \right\rfloor$$
$$S_k = \textstyle\sum_{1 \le j_1 < \cdots < j_k \le n} P(A_{j_1} \cap \cdots \cap A_{j_k}), \;\; k = 1, \ldots, n.$$

Hunter's upper bound

Let $A_1, \ldots, A_n$ be arbitrary events in an arbitrary probability space. Hunter (1976) provides an upper bound for $P(A_1 \cup \cdots \cup A_n)$ by the use of S_1 and the individual probabilities $P(A_i \cap A_j), \;\; 1 \le i < j \le n$. Hunter's upper bound; therefore is given as:

$$P(A_1 \cup \cdots \cup A_n) \le S_1 - \sum_{(i,j)\in T} P(A_i \cap A_j). \tag{3.11}$$

The second term on the right hand side in inequality (3.11) is the weight of the spanning tree T. The best bound of this type is obtained when we choose the maximum weight spanning tree T^*. Maximum weight spanning tree can be found using the Kruskal's algorithm (Kruskal, 1956).

Cherry tree upper bound

A third order upper bound on the probability of the union of a finite number of events, is presented by means of graphs called cherry trees. These are graphs that we construct recursively in such a way that every time we pick a new vertex, we connect it with two already existing vertices. If the latters are always adjacent, we call this a t-cherry tree. A cherry tree has a weight that provides us with the upper bound on the union. A cherry tree bound can be identified as a feasible solution to the dual of the Boolean probability bounding problem. Moreover, a t-cherry tree bound can be identified as the objective function value of the dual vector corresponding to a dual feasible basis in the Boolean problem. This enables us to make an improvement on the bound algorithmically, if we use the dual method of linear programming. First we will recall the definition of a cherry tree:

Definition (Bukszár, Prékopa 2001) We define a cherry tree recursively in the following manner:

(i) An adjacent pair of vertices constitutes the only cherry tree that has exactly two vertices.

(ii) From a cherry tree we can obtain another cherry tree by adding a new vertex and two new edges, connecting the new vertex with two already existing vertices. These two edges constitute a cherry.

(iii) If V is the set of vertices, ξ the set of edges and ε the set of cherries obtained that way then we call the triple $\Delta = (V, \xi, \varepsilon)$ a cherry tree.

Theorem 3.4.1 Bukszár and Prékopa (2001) *For any cherry tree* $\Delta = (V, \xi, \varepsilon)$ *with* $V = 1, \ldots, n$ *we have*

$$P(A_1 \cup \cdots \cup A_n) \leq \textstyle\sum_{i=1}^{n} P(A_i) - w(\Delta) = S_1 - w(\Delta)$$
$$\textit{where} \tag{3.12}$$
$$w(\Delta) = \textstyle\sum_{\{i,j\} \in \xi} P(A_i \cap A_j) - \sum_{(i,j,k) \in \xi} P(A_i \cap A_j \cap A_k)$$

Proof of the Theorem (3.4.1) can be found in Bukszár, Prékopa 2001.

3.4.4 Our lower and upper bounds

In Section 4.3, sharp lower and upper bounds that are widely used in the probability theory are defined and formulated. There are two proposed upper bounds for the union of the events; Hunter and Cherry tree upper bounds which are both dual feasible bases for the Boolean bounding problem. However, all of these bounds calculate upper and lower ranges for the probability of the union of the events. Since the main interest in our problem is to find a lower and upper bound for the intersection of the events that is defined in Section 3.2, the following conversion is needed:

$$P(A_1 \cap \cdots \cap A_{n-k+1}) = 1 - P(\overline{A}_1 \cup \cdots \cup \overline{A}_{n-k+1}) \tag{3.13}$$

If the lower bound for the union of the events is defined by equation (3.10), then we will have the following formulation for the lower bound:

$$\frac{i+2n-1}{(i+1)n} S_1 - \frac{2(2i+n-2)}{i(i+1)n} S_2 + \frac{6}{i(i+1)} S_3 = LB_1.$$

Although we calculate both Hunter and Cherry tree upper bounds, our computational experience shows that cherry tree bounds are always better than or equal to Hunter upper bound so that, we will define the upper bound for the union of the events with Equation (3.12) and rename the right hand side as UB_1. Then, we will have the following:

$$S_1 - w(\Delta) = UB_1 \tag{3.14}$$

Therefore, ranges for the union of the events can be rewritten as follows:

$$LB_1 \leq P(\overline{A}_1 \cup \cdots \cup \overline{A}_{n-k+1}) \leq UB_1 \tag{3.15}$$

If we rewrite the union of all the events in terms of intersection of complimentary events, we will obtain the following:

$$LB_1 \leq 1 - P(A_1 \cap \cdots \cap A_{n-k+1}) \leq UB_1 \tag{3.16}$$

After manipulating Equation (3.16), we will have the lower and upper bounds for the intersection of the events as follows:

$$1 - UB_1 \leq P(A_1 \cap \cdots \cap A_{n-k+1}) \leq 1 - LB_1 \tag{3.17}$$

3.4.5 Bisection method

After calculating a lower and an upper bound for the intersection of the events, the probability interval for the reliability constraint will be used to obtain the optimal capacity, m. The m value will be bisected until the calculated p is within one-decimal accuracy of the probability bounds.

To perform this, the well-known bisection algorithm will be used. Bisection method in mathematics, is a root-finding method which repeatedly bisects an interval and then selects a subinterval in which a root must lie for further processing. In our case, the root that we would like to find is the p. (Wood, 1989)

3.4.6 Summary of the steps for the solution of model 3.2

Here, we summarize the solution steps.

Step 0:

Generate ξ and η in Matlab by sampling from a normal distribution. m is a reasonable fixed number which is subject to change during the bisection algorithm.

Step 1:

$\overline{A}_i$,$\overline{A}_{ij}$, $\overline{A}_{ijk}$ are calculated in order to find the S_1, S_2, S_3.

Step 2:

S_1, S_2, S_3 are calculated.

Step 3:

Lower and upper bounds are calculated with Equation (3.10) and Equation (3.11) for the event:

$$P(\overline{A}_1 \cup \cdots \cup \overline{A}_n)$$

Step 4:

Transformation of the intersection of the events from union of the events is performed as follows:

$$LB_1 \leq P(\overline{A}_1 \cup \cdots \cup \overline{A}_{n-k+1}) \leq UB_1$$
$$LB_1 \leq 1 - P(A_1 \cap \cdots \cap A_{n-k+1}) \leq UB_1$$
$$1 - UB_1 \leq P(A_1 \cap \cdots \cap A_{n-k+1}) \leq 1 - LB_1.$$

Step :5

Bisection algorithm is applied.Three possibilities can occur during the bisection algorithm:

- If p is larger than upper bound, then pick larger m
- If p is smaller than lower bound, then pick smaller m
- If p is in between lower and upper bound, try picking smaller/larger m
  - If a large value of m performs better on value of m, keep bisection into the same direction
  - If a large value of m does not perform better on value of m, repeat the bisection in other direction.

Step 6:

If p is in between one decimal digit of lower and upper bound STOP, else go to *Step 0* and change m.

3.5 Illustrative Example

In this section, we present a solution for an optimal water reservoir problem where a total period of eight weeks (56 days) is considered. In the first formulation of the problem, any seven days of dryness (not meeting the demand) is permitted with a probability level of 90%. For the sake of computational easiness, the rain amount which is an additional type of supply is considered with the inflow (supply). Distribution of the inflow is assumed to be normal and the demand is considered to follow gamma distribution in the first formulation. In the second formulation of the problem, we use normal distribution for both inflow and demand distributions. Dryness in consecutive seven days is forbidden with a probability level of 90%.

Model (3.1)

$$\begin{gathered}\min m\\ \text{subject to}\\ P\{(\min(\xi_i, m) \geq x_i\eta_i)\,,\ \ i = 1, \ldots, 56\ \ x_1 + \cdots + x_{56} \geq 7\} \geq 0.90\\ x_i \in \{0, 1\}\ \ i = 1, \ldots, 56\\ 0 \leq m \leq M\end{gathered} \tag{3.18}$$

Solution of Model (3.1) is basically solving the following equation:

$$P\left(\sum_{i=1}^{56} \lambda\left[\min(\xi_i, m)\right] \geq (49)\lambda\gamma_1\right) = 0.9 \tag{3.19}$$

where ξ_i is a normally distributed random variable with parameters $(200, 30)$ and γ_1 belongs to a gamma distribution with parameters $\lambda = 20, \vartheta = 10$. Equation (3.19) is solved by coding a simulator in JAVA and running the equation 1,000 times to get the most observed accurate minimum value for the capacity value m. The most frequently observed m value is 162 therefore we can say that with the given inflow and demand variables, the upper bound of the minimum capacity for the reservoir can be approximated as 162, while maintaining at most 7 days of dry periods with a probability level of 90% (with a precision of 0.01).

Model (3.2)

$$\begin{gathered}\min m\\ \text{subject to}\\ P\{(\min(\xi_i, m) \geq x_i\eta_i)\,,\ \ i = 1, \ldots, 56\ \ x_i + \cdots + x_{i+6} \geq 1\ \ i = 1, \ldots, 50\} \geq 0.90\\ x_i \in \{0, 1\}\ \ i = 1, \ldots, 56\\ 0 \leq m \leq M\end{gathered} \tag{3.20}$$

where the reliability constraint of the problem is equivalent to:

$$P\{(\xi_k \geq \eta_k, m \geq \eta_k),\ \ i = 1, \ldots, 56\ \ x_i + \cdots + x_{i+6} \geq 1\ \ i = 1, \ldots, 50\} \geq 0.90$$

Distributions for the inflow (supply) and demand can be found in the Table 4.3 of Appendix A. We first choose the initial capacity value m as 180. Next, we apply the bi-section algorithm based on the intervals of lower and upper bounds of the probability value. After applying the steps given in Section 3.4, we obtain the results given in Table 1.

Table 1 shows that the interval that contains our desired p is obtained with a capacity value of m that equals 151.4. Results clearly indicate that, with given inflow and demand distributions, the land will not be dry for 7 consecutive days with the probability level of 0.90 when the capacity of the reservoir is 151.4.

Table 3.1: Bisection algorithm results

Steps	Capacity, M	S_1, S_2, S_3 Lower Bound	Hunter Upper Bound	Cherry Tree Bound	p_{value}	Comment
1	180	0.973505	14.45638	12.345356	0.9	p<LB, pick smaller M
2	158.8	0.879283	3.664441	3.5678921	0.9	LB<p<UB, try decreasing M
3	153.4	0.890781	1.708109	1.684544	0.9	LB<p<UB, try decreasing M
4	152.1	0.893874	1.195414	1.194523	0.9	LB<p<UB, try decreasing M
5	151.3	0.895434	0.937376	0.937283	0.9	LB<p<UB, try increasing M
6	151.7	0.894653	1.06553	1.064428	0.9	LB<p<UB, try decreasing M
7	151.5	0.895043	1.00159	1.000967	0.9	LB<p<UB, try decreasing M
8	151.4	0.895541	0.919803	0.91789	0.9	STOP!

3.6 Appendix A

Table 3.2: Inflow ξ and demand η distributions

Inflow (ξ) Distribution		Demand (η) Distribution	
Mean	Standard Deviation	Mean	Standard Deviation
198	17	149	38
186	50	165	49
117	13	204	35
186	2	181	32
162	14	199	42
206	72	178	56
206	17	177	32
163	91	173	59
209	13	227	58
165	108	202	53
186	45	194	48
211	52	189	41
147	63	181	51
201	62	204	39
192	93	204	28
182	12	182	45
225	79	183	41
224	75	179	41
258	31	180	36

Continued on next page

Inflow (ξ) Distribution		Demand (η) Distribution	
Mean	Standard Deviation	Mean	Standard Deviation
198	5	144	32
235	62	157	32
213	54	186	33
218	40	204	47
179	8	225	44
223	98	142	50
226	44	174	41
160	56	175	45
192	40	250	33
219	49	189	28
139	11	187	58
197	44	170	43
175	25	188	47
170	80	190	47
208	60	187	49
195	5	174	35
223	29	209	38
183	10	201	25
203	7	157	43
199	78	191	35
218	34	187	38
206	41	188	30

Continued on next page

Inflow (ξ) Distribution		Demand (η) Distribution	
Mean	Standard Deviation	Mean	Standard Deviation
189	2	181	21
232	17	208	42
238	52	172	62
234	48	178	53
187	102	180	42
166	1	170	32
197	45	198	30
191	52	205	15
142	40	159	45
192	34	165	36
234	62	172	23
185	28	174	36
175	16	214	45
177	101	200	29
177	52	179	26

Chapter 4

A Serially Linked Reservoir Network Design Problem with Consecutive k-out-of-n Type Reliability

4.1 Introduction

In this chapter, we formulate and solve a probabilistic constrained stochastic programming model for a serially linked reservoir system under consecutive k-out-of-n type reliability. The problem is to determine the optimal capacities of water releases in reservoirs or buffer stations in manufacturing system or shelter capacities in an evacuation network while satisfying the reliability constraint. We propose a hybrid algorithm of cutting planes and supporting hyperplanes as a solution methodology. Our solution is novel in the sense that it is fast, efficient, and can be applied in many different areas such as water engineering, manufacturing buffer systems, evacuation networks, and so forth. We formulate our problem as a stochastic programming problem under probabilistic constraint, prove its mathematical properties, and explain the solution methodology. Finally, we present a numerical example for two reservoirs that satisfy the demand for at least 8 consecutive periods out of 24 periods. The demand and inflow values that are used in the example are random, stochastically dependent, and normally distributed.

4.2 Formulation of the Problem

The capacity design of the reservoirs on the rivers are one of the best application areas of serially linked type of networks therefore we refer to an example coming from water engineering to describe the model and the solution method throughout this chapter. The topology of the main river, the sider rivers, and the possible reservoir sites that are serially linked is illustrated in Figure 1.

Time is subdivided into a finite number of periods. Periods can be weeks, decades, etc. in practice. We assume that at the beginning of each period, certain water inputs occur in accordance with the topology of the rivers and reservoirs. If a reservoir becomes full, then the water spills and fills downstream reservoirs. When this is the case, no more water is released from upstream reservoirs to downstream reservoirs. At the end of every period, demands occur which can be assigned to different reservoirs. If possible, every demand is primarily satisfied from the reservoir that they are assigned. If this is not the case, then the following policy happens. First, demands are satisfied from the corresponding reservoir with the amount that is available from that reservoir. Then starting from the reservoir furthest downstream, stop at the first unsatisfied demand reservoir. From that reservoir, aggregate the demand and try to meet this demand from the upstream. If this is not possible, proceed similarly in the upstream direction. If the whole system can meet the demand, then this procedure stops at a certain point and we can conclude that the system can satisfy the downstream demands. This procedure is repeated for the remaining upstream subsystem, and so forth. In our model, we guarantee meeting the demand for at least one period out of k consecutive periods with a prescribed probability.

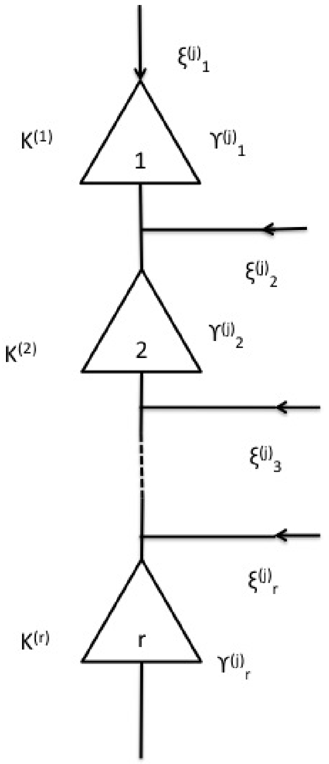


Figure 4.1: Topology of main river, the side rivers, and the possible reservoir sites.

Let us introduce the following notation:

r	number of sites
k	number of consecutive periods
n	number of periods
$K^{(j)}$	unknown capacity of reservoir j
$C^{(j)}$	building cost of reservoir j as a function of its capacity
$V^{(j)}$	prescribed constant, upper bound for $K^{(j)}$
$\zeta_i^{(j)}$	water content in reservoir j at end of i^{th} period
$\xi_i^{(j)}$	direct inflow into reservoir j in the i^{th} period
$\eta_i^{(j)}$	direct demand against reservoir j in the i^{th} period
$x_i^{(j)}$	1 if j^{th} reservoir satisfies the demand in the i^{th} period, 0 otherwise

4.2.1 Demand will be met at least k consecutive periods in total of n periods

The model below is the general case where there are r reservoirs and n periods with the condition of satisfying the demand for at least k consecutive periods out of n periods.

We assume that at the beginning of every period, certain water inputs occur according to the topology of rivers and reservoirs. If reservoir becomes full, then the additional water overflows to downstream reservoirs. No more water is released from upstream reservoirs to downstream reservoirs. At the end of every period, demands occur which can be assigned to separate reservoirs. Every demand is satisfied from the assigned reservoir if it is possible. If not, then our assumed operating policy is as follows. First, demands are satisfied to the extent that water amounts are in the corresponding reservoirs. If some demand is left unsatisfied from the corresponding reservoir, then the extra demand is satisfied from the reservoir upstream to the extent that water is available there. If some demand is still left unsatisfied, we proceed further upstream until all the demand is satisfied. Our model is given by;

$$\text{Min} \sum_{j=1}^{r} C^{(j)} K^{(j)}$$

$$\text{subject to} \tag{4.1}$$

$$P\left(\begin{array}{c} \mathrm{Min}\big\{\zeta_{i-1}^{(j)} + \big[\min(\zeta_{i-1}^{(j-1)} + \xi_i^{(j)}, K^{(j-1)}) - x_i^{(j)}\eta_i^{(j)}]_+ + \cdots + \\ \big[\min(\zeta_{i-1}^{(1)} + \xi_i^{(1)}, K^{(1)}) - x_i\eta_i^{(1)}]_+ + \xi_i^{(2)}, K_i^{(j)} \geq x_i^{(j)}\eta_i^{(r)}\big\} \\ x_1^{(j)} + \cdots + x_k^{(j)} \geq 1 \\ x_2^{(j)} + \cdots + x_{k+1}^{(j)} \geq 1 \\ \vdots \\ x_{n-k+1}^{(j)} + \cdots + x_n^{(j)} \geq 1 \end{array}\right) \geq p$$

$$K_i^{(j)} \leq V^{(j)}$$

$$x_i^{(j)} \in \{0,1\},\ \ \zeta_i^{(j)}, \eta_i^{(j)}, \xi_i^{(j)}, K^{(j)} \geq 0$$

$$\text{where } i = 1, \ldots, n \ \ j = 1 \ldots, r$$

4.3 Mathematical Properties of the Reservoir System Design Model (4.5)

From the point of finding a solution, it is very important to know if the problem is convex. Since we have discrete and continuous variables simultaneously, overall convexity cannot be expected. However, we can prove the convexity of the set of feasible solutions for any fixed values of $0-1$ boolean variables.

The mathematical properties of the probabilistic constraining function will be derived from the following theorems.

Theorem 4.3.1 *Let $g_i(x,y), i = 1, \ldots, r$ be concave function in $\mathbb{R}^{m+n}$ where x is an n-component and y is an m-component vector. Let ξ be an m-component random vector having a logarithmic concave probability distribution. Then the function of the variable x:*

$$P\left(g_i(x,\xi) \geq 0,\ \ i = 1, \ldots, r\right), \tag{4.2}$$

(4.2) is logarithmic concave in the space $\mathbb{R}^n$.

This theorem was proved by Prékopa (1971). For further similar results, one should refer to the paper by Prékopa (1978).

Theorem 4.3.2 *For any fixed $x_i^{(j)}$, $i = 1, \ldots, n$, $j = 1, \ldots, r$, the probability, standing on the left-hand side in the probabilistic constraint (1), is a logconcave function of $\big(K^{(1)}, \ldots, K^{(r)}\big)$.*

Proof Inside the parentheses of the probability in (1) we have inequalities that we imagine reformulated in such a way that as 0 remains on the right-hand sides of the inequalities.

Then, we have n inequalities which contain functions of the variables $\left(K^{(1)}, \ldots, K^{(r)}\right)$ and $\left(\xi_1^{(j)}, \ldots, \xi_n^{(j)}, \gamma_1^{(j)}, \ldots, \gamma_n^{(j)}\right)$, and these functions are concave. The application of Theorem 4.3.1 for this case proves Theorem 4.3.2.

4.4 Solution Methodology of the Problem

4.4.1 Reduction of the feasible set

The probabilistic constraint in (4.5) is formulated as a set of inequalities that need to be satisfied for each component of each vector $\mathbf{x}^{(j)} = (x_1^{(j)}, \ldots, x_n^{(j)})$ where

$$\begin{aligned} & x_1^{(j)} + \ldots + x_k^{(j)} \leq k-1 \\ & x_2^{(j)} + \ldots + x_{k+1}^{(j)} \leq k-1 \\ & \ldots \\ & x_{n-k+1}^{(j)} + \ldots + x_n^{(j)} \leq k-1 \\ & x_i^{(j)} \in \{0,1\},\ i = 1, \ldots, n \end{aligned} \tag{4.3}$$

In this section we show that it suffices to check only a subset of values of $\mathbf{x}^{(j)}$ satisfying (4.3) to ensure that the probabilistic constraint holds for all $\mathbf{x}^{(j)}$ satisfying (4.3).

Observe that the right-hand side of the probabilistic constraint in (4.1) is monotonously increasing in $\mathbf{x}^{(j)}$, and the left-hand side is monotonously decreasing in $\mathbf{x}^{(j)}$ (the larger values of $x_i^{(j)}$ imply the smaller values of $\zeta_j^{(i)}$) Therefore, if the constraints hold for some $\mathbf{x}^{(j)\prime} \geq \mathbf{x}^{(j)\prime\prime}$, then they also hold for $\mathbf{x}^{(j)\prime\prime}$. As a result, the constraints only need to be checked for those values of $\mathbf{x}^{(j)}$ which are maximal elements of the set defined by (4.3).

We now proceed to characterize the maximal elements of (4.3). A vector $\mathbf{x}$ is *not* maximal if for one of its coordinates x_r, we have $x_r^{(j)} = 0$ and the vector $\mathbf{x}' = (x_1^{(j)}, \ldots, x_{r-1}^{(j)}, 1, x_{r+1}^{(j)}, \ldots, x_n^{(j)})$ also satisfies (4.3). This only happens if for all r_0 such that $r - k + 1 \leq r_0 \leq r$, we have $x_{r_0}^{(j)} + \ldots + x_{r_0+k-1}^{(j)} < k-1$. In other words, for each sequence of coordinates $x_{r_0}^{(j)}, \ldots, x_{r_0+k-1}^{(j)}$, there is at least one zero coordinate besides $x_r^{(j)}$. A vector $\mathbf{x}^{(j)}$ is maximal if there is no such coordinate $x_r^{(j)}$, that is

$$\forall r \in \overline{1,n} : x_r^{(j)} = 0 \Rightarrow \exists r_0, r-k+1 \leq r_0 \leq r : x_{r_0}^{(j)} + \ldots + x_{r_0+k-1}^{(j)} = k-1 \tag{4.4}$$

Condition (4.4) can be restated in a simpler form through the lengths of runs of ones in the vector $\mathbf{x}^{(j)}$. If $x_{r_0}^{(j)} = 1, x_{r_0+1}^{(j)} = 1, \ldots, x_{r_0+l-1}^{(j)} = 1$, but either $x_{r_0-1}^{(j)} = 0$ or $r_0 = 1$ and either $x_{r_0+l}^{(j)} = 0$ or $r_0 + l - 1 = n$, then we say that $(x_{r_0}^{(j)}, \ldots, x_{r_0+l-1}^{(j)})$ is a run of ones of length l. If both $x_{r_0-1}^{(j)} = 0$ and $x_{r_0}^{(j)} = 0$ we will also say that they are separated by a run of ones of length

0. Given vector $\mathbf{x}^{(j)}$, let l_1 be the length of run of ones starting at position 1, let l_2 be the length of run of ones starting at position $l_1 + 1$, l_3 be the length of run of ones starting at position $l_1 + l_2 + 2$ and so on. The sequence $l_1, l_2, \ldots, l_s$ uniquely defines the vector $\mathbf{x}^{(j)}$. Moreover, we can now restate the condition (4.4): vector $\mathbf{x}^{(j)}$ is maximal if $\forall i : l_i + l_{i+1} \geq k - 1$. Notice, that vector $\mathbf{x}^{(j)}$ satisfies (4.3) if and only if $\forall i : l_i \leq k - 1$. Thus, all maximal vectors $\mathbf{x}^{(j)}$ can be enumerated by listing all sequences of nonnegative integers $l_1, \ldots, l_s$ satisfying

$$\begin{aligned} & l_i \leq k-1, \quad i = 1, \ldots, s, \\ & l_i + l_{i+1} \geq k-1, \quad i = 1, \ldots, s-1, \\ & \textstyle\sum_{i=1}^{s} l_i + s = n, \end{aligned} \tag{4.5}$$

where the last constraint encodes the fact that the length of vector $\mathbf{x}^{(j)}$ is n.

The sequences $\{l_i\}$ in (4.5) can be enumerated directly with, for example, a depth-first search algorithm.

4.4.2 Hybrid Algorithm: Cutting Plane & Supporting Hyperplane Method

The supporting hyperplane method is applied to probabilistic constrained stochastic programming problem and it is an improvement on the cutting plane algorithm that is developed by Kelley (1960) and Cheney and Goldstein (1959). It is first developed by Veinott (1967) and adapted by Szántai (1988) to solve the below:

$$\begin{gathered} \min c^T x \\ \text{subject to} \\ Ax \geq b \\ P(Tx \geq \xi) \geq p \\ x \geq 0 \qquad . \end{gathered} \tag{4.6}$$

where p $(0 < p < 1)$ is a prescribed probability. We assume that ξ has a continuous distribution with logconcave probability distribution function (pdf). Let $h(x) = P(Tx \geq \xi) - p$ and assume the convex polyhedron $K^0 = \{x | Ax \geq b, x \geq 0\}$ is bounded. Assume further that Slater's condition is satisfied: there exists an $x^0 \in K_0$ such that $h(x^0) > 0$. Throughout the algorithm, we used fixed x^0 satisfying Slater's condition. If we encounter an x such that $h(x) > 0$, we may terminate the algorithm and set $x^0 = x$.

We combined the Prékopa, Vizvári, Badics algorithm for cutting plane and supporting hyperplane method of Veinott (1967), Prékopa, Szántai (1978) and Szántai (1988) to solve the problem (2) with continuously distributed ξ. Let's briefly summarize the steps of this hybrid algorithm:

- Step 0. Find x^0 satisfying $Ax^0 \geq b, x^0 \geq 0, h(x^0) > 0$. Go to Step 1.
- Step 1. Solve the LP:

  $$\text{Min } c^T x$$

  $$\text{subject to}$$

  $$Ax \geq b$$

  $$\bigtriangledown h(x^i)(x - x^0) \geq 0, i = 1, \ldots, k$$

  $$x \geq 0.$$

  Let x^{*k} be an optimal solution. Go to Step 2.

- Step 2. Check for the sign of $h(x^{*k})$. If $h(x^{*k}) \geq 0$, Stop, optimal solution to problem (3) has been found. Otherwise go to Step 3.
- Step 3. Find λ^k such that $0 < \lambda^k < 1$ and $h(x^0 + \lambda^k(x^{*k} - x^0)) = 0$. Define $x^{k+1} = x^0 + \lambda^k(x^k - x^0)$ and go to step 4.
- Step 4. Introduce the cut: $\bigtriangledown h(x^{k+1})(x - x^0) \geq 0$, set $k \leftarrow k + 1$ and go to Step 1.

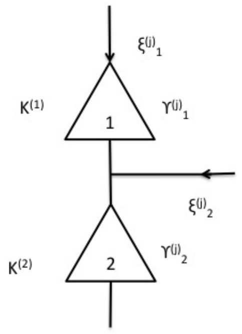


Figure 4.2: Topology of main river, one side river, and 2 reservoir sites.

4.5 Numerical Results

Here we present a 2-serially linked reservoir network problem. (See Figure(4.2).) We consider a total number of 24 days and we force our model to satisfy the demand at least one out of eight consecutive days in these total of 24 days at a probability level of 0.9. The cost function for the reservoirs is convex with coefficients of 1.5×10^4 for $C^{(1)}$ and 1×10^4 for $C^{(2)}$. Below is the formulation of our problem:

$$\begin{gathered}
\text{Min } 15000 \times K^{(1)} + 10000 \times K^{(2)} \\
\text{subject to} \\
P \left(\begin{gathered}
\text{Min}\big\{\zeta_{i-1}^{(j)} + \big[\min(\zeta_{i-1}^{(j-1)} + \xi_i^{(j)}, K^{(j-1)}) - x_i^{(j)}\eta_i^{(j)}\big]_+ + \cdots + \\
\big[\min(\zeta_{i-1}^{(1)} + \xi_i^{(1)}, K^{(1)}) - x_i\eta_i^{(1)}\big]_+ + \xi_i^{(2)}, K_i^{(j)} \geq x_i^{(j)}\eta_i^{(r)}\big\} \\
x_1^{(j)} + \cdots + x_k^{(j)} \geq 1 \\
x_2^{(j)} + \cdots + x_{k+1}^{(j)} \geq 1 \\
\vdots \\
x_{n-k+1}^{(j)} + \cdots + x_n^{(j)} \geq 1
\end{gathered} \right) \geq 0.9 \\
K_i^{(j)} \leq 50 \\
x_i^{(j)} \in \{0,1\},\ \ \zeta_i^{(j)}, \eta_i^{(j)}, \xi_i^{(j)}, K^{(j)} \geq 0 \\
\text{where } i = 1, \ldots, 24\ \ j = 1, 2
\end{gathered} \tag{4.7}$$

We assume the inflow and the demand values are random variables and coming from a normal distribution. We further assume that demand values for different days for each reservoir are dependent on each other. The covariance matrices are shown in Appendix A. Inflow values are also dependent to each other with respect to days and covariance matrices are again shown in Appendix A. The data for the distribution of inflow and demand for each reservoirs that is used for this problem can be reached in Appendix B.

Since there are 24 days, we have 17 constraints for the first reservoir to ensure the availability of water for at least one day out of eight consecutive days. There are also 17 constraints for the second reservoir with the same purpose. If we consider all the possible $x_i^{(j)}$ pairs that satisfies the 34×8 system of inequalities, there is going to be 2^{34} total feasible vectors that need to be used to achieve the optimal solution. With the simple elimination technique described in Section 4.4.1, we are able to reduce the set of feasible vectors to 254. After that, we used the

Table 4.1: Results for hybrid algorithm

	Supporting hyperplane
K_1 Capacity (1000 m^3)	15.06498
K_2 Capacity (1000 m^3)	5.7687
Optimal Cost	28.36617
Number of Iterations	78

hybrid algorithm to obtain the optimum solution while satisfying the reliability constraint with a probability value of 0.9. The results can be found in table 4.1.

4.6 Appendix A

Covariance Matrices Between Days for Demand and Inflow variables for Reservoir 1 and Reservoir 2

$R^{(j)}_{Demand}$	covariance matrix for demand of reservoir j
$R^{(j)}_{Inflow}$	covariance matrix for inflow of reservoir j
	where $j = 1, 2$

$$R^{(1)}_{Demand} = \begin{bmatrix}
1.00 & 0.28 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.28 & 1.00 & 0.20 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.20 & 1.00 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.01 & 1.00 & 0.14 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.14 & 1.00 & 0.21 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.21 & 1.00 & 0.12 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.12 & 1.00 & 0.27 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.27 & 1.00 & 0.31 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.31 & 1.00 & 0.36 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.36 & 1.00 & 0.45 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.45 & 1.00 & 0.45 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.45 & 1.00 & 0.23 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.23 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.02 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.02 & 1.00 & 0.22 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.22 & 1.00 & 0.04 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.04 & 1.00 & 0.45 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.45 & 1.00 & 0.45 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.45 & 1.00 & 0.03 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.03 & 1.00 & 0.02 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.02 & 1.00 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.10 & 1.00 & 0.11 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.11 & 1.00 & 0.69 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.69 & 1.00
\end{bmatrix}$$

$$R^{(2)}_{Demand} = \left[\begin{array}{cccccccccccccccccccccccc}
1.00 & 0.29 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.29 & 1.00 & 0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.25 & 1.00 & 0.14 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.14 & 1.00 & 0.14 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.14 & 1.00 & 0.20 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.20 & 1.00 & 0.15 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.15 & 1.00 & 0.36 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.36 & 1.00 & 0.37 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.37 & 1.00 & 0.12 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.12 & 1.00 & 0.45 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.45 & 1.00 & 0.47 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.47 & 1.00 & 0.09 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.09 & 1.00 & 0.09 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.09 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.09 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.09 & 1.00 & 0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.25 & 1.00 & 0.08 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.08 & 1.00 & 0.44 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.44 & 1.00 & 0.28 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.28 & 1.00 & 0.03 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.03 & 1.00 & 0.07 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.07 & 1.00 & 0.18 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.18 & 1.00 & 0.12 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.12 & 1.00
\end{array} \right]$$

$$R^{(1)}_{Inflow} = \begin{bmatrix}
1.00 & 0.48 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.48 & 1.00 & 0.38 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.38 & 1.00 & 0.32 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.32 & 1.00 & 0.16 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.16 & 1.00 & 0.26 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.26 & 1.00 & 0.23 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.23 & 1.00 & 0.02 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.02 & 1.00 & 0.26 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.26 & 1.00 & 0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.25 & 1.00 & 0.42 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.42 & 1.00 & 0.45 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.45 & 1.00 & 0.07 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.07 & 1.00 & 0.26 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.26 & 1.00 & 0.32 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.32 & 1.00 & 0.12 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.12 & 1.00 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.01 & 1.00 & 0.45 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.45 & 1.00 & 0.45 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.45 & 1.00 & 0.12 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.12 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.07 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.07 & 1.00 & 0.17 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.17 & 1.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00
\end{bmatrix}$$

$$R^{(2)}_{Inflow} = \begin{bmatrix}
1.00 & 0.21 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.21 & 1.00 & 0.09 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.09 & 1.00 & 0.07 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.07 & 1.00 & 0.01 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.01 & 1.00 & 0.06 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.06 & 1.00 & 0.11 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.11 & 1.00 & 0.08 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.08 & 1.00 & 0.16 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.16 & 1.00 & 0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.25 & 1.00 & 0.23 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.23 & 1.00 & 0.23 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.23 & 1.00 & 0.09 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.09 & 1.00 & 0.16 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.16 & 1.00 & 0.19 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.19 & 1.00 & 0.03 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.03 & 1.00 & 0.03 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.03 & 1.00 & 0.02 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.02 & 1.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.18 & 1.00 & 0.11 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.11 & 1.00 & 0.04 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.04 & 1.00 & 0.15 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.15 & 1.00 & 0.18 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.18 & 1.00 & 0.03 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.03 & 1.00
\end{bmatrix}$$

4.7 Appendix B

Distribution of Demand and Inflow Values for Reservoir 1 and Reservoir 2

Table 4.2: Demand η distributions

	RESERVOIR 1		RESERVOIR 2	
	Demand ($\eta_i^{(1)}$) Distribution		Demand ($\eta_i^{(2)}$) Distribution	
	Mean	Standard Deviation	Mean	Standard Deviation
Day (i)	1000 m^3	1000 m^3	1000 m^3	1000 m^3
1	2.544	0.2544	1.564	0.1564
2	4.778	0.4778	3.895	0.3895
3	6.4	0.64	5.89	0.589
4	6.481	0.6481	4.763	0.4763
5	7.607	0.7607	5.92	0.592
6	5.933	0.5933	4.327	0.4327
7	4.993	0.4993	3.123	0.3123
8	7.117	0.7117	5.983	0.5983
9	4.67	0.467	3.012	0.3012
10	7.554	0.7554	2.074	0.2074
11	11.167	1.1167	6.93	0.693
12	5.87	0.587	3.21	0.321
13	4.012	0.4012	3.923	0.3923
14	4.014	0.4014	3.217	0.3217
15	4.205	0.4205	3.214	0.3214
16	5.985	0.5985	3.97	0.397
17	6.345	0.6345	5.936	0.5936
18	12.54	1.254	5.32	0.532
19	3.987	0.3987	1.784	0.1784
20	4.231	0.4231	3.984	0.3984
21	4.094	0.4094	3.758	0.3758
22	4.875	0.4875	4.345	0.4345
23	5.73	0.573	2.937	0.2937
24	2.98	0.298	1.984	0.1984

Table 4.3: Inflow η distributions

	RESERVOIR 1		RESERVOIR 2	
	Inflow ($\eta_i^{(1)}$) Distribution		Inflow ($\eta_i^{(2)}$) Distribution	
	Mean	Standard Deviation	Mean	Standard Deviation
Day (i)	1000 m^3	1000 m^3	1000 m^3	1000 m^3
1	6.023	0.6023	3.785	0.3785
2	9.855	0.9855	5.463	0.5463
3	6.778	0.6778	4.739	0.4739
4	9.342	0.9342	5.263	0.5263
5	10.583	1.0583	5.307	0.5307
6	8.521	0.8521	4.851	0.4851
7	6.721	0.6721	3.987	0.3987
8	6.844	0.6844	4.632	0.4632
9	8.9	0.89	5.933	0.5933
10	6.869	0.6869	3.914	0.3914
11	10.229	1.0229	5.743	0.5743
12	5.498	0.5498	3.894	0.3894
13	6.03	0.603	4.62	0.462
14	8.091	0.8091	5.938	0.5938
15	5.53	0.553	4.422	0.4422
16	6.482	0.6482	4.695	0.4695
17	6.437	0.6437	4.909	0.4909
18	13.22	1.322	5.095	0.5095
19	4.4	0.44	3.695	0.3695
20	5.34	0.534	4.56	0.456
21	5.78	0.578	4.234	0.4234
22	6.321	0.6321	5.442	0.5442
23	4.98	0.498	3.986	0.3986
24	5.01	0.501	3.724	0.3724

References

[1] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, and M. R. Reddy, Applications of Network Optimization, *Handbooks in OR & MS*, **7**, (1995).

[2] R. K. Ahuja, T. L. Magnanti and J. B, Orlin, Network flows: Theory, algorithms and applications, *Prentice Hall* Englewood Cliffs, NJ, (1993).

[3] T. W. Archibald, K.I.M. McKinnon, and L.C. Thomas, An aggregate stochastic dynamic programming model of multireservoir systems, *Water Resources Research,* **33**, (1997), 333–340.

[4] J. E. Aronson, A survey of dynamic network flows, *Annals of Operations Research*, **20 (1)**, (1989), 1–66.

[5] Y. Asakura, and E. Hato, Stochastic network design problem: An optimal link investment model for reliable network, *Network Reliability of Transport. Proceedings of the 1st International Symposium on Transportation Network Reliability (INSTR)* Kyoto, Japan, (2001).

[6] V. Belevitch, L.A. Zadeh, R. M. Foster, and E.A. Guillemin, Circuit Theory. *Proceedings of the IRE*, (1961).

[7] P. Beraldi, M. E. Bruni, and F. Guerriero, Network reliability design via joint porbabilistic constraints, *IMA Journal of Management Mathematics*, **21**, (2009), 213–226.

[8] D. R. Bish, Demand management in evacuation planning models algorithms and applications, (2006).

[9] J. M. Borwein, On the Existence of Pareto Efficient Points, *Mathematics of Operations Research*, **8**, (1983), 64–73.

[10] J. Bukszár, A. Prékopa, Probability Bounds with Cherry Trees, *Mathematics of Operations Research*, **26**, (2001), 174-192.

[11] A. Castelleti, F. Pianosi, and R. Soncini-Sessa, Water Reservoir Control under Economic, Social and Environmental Constraints, Automatica, 44, (2008), 1595–1607.

[12] C. Cervellera, V. C. P. Chen, and A. Wen, Optimization of a large-scale water reservoir network by stochastic dynamic programming with efficient state space discretization, *European Journal of Operations Research*, **171**, (2006), 1139–1151.

[13] A. Charnes, W. W. Cooper, and G. H. Symonds, Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science* **4**, (1958), 235–263.

[14] A. Chen, H, Yang, H. K. Lo, and W. H. Tang, Capacity reliability of a road network: as assessment methodology and numerical results. *Transportation Research* **Part B 36**, (2003), 225–252.

[15] E. W. Cheney, and A. A. Goldstein, Newton's method for convex programming and Tchebycheff approximation, *Numerische Mathematik* **1**, (1959), 253–268.

[16] E. Cohen, and N. Megiddo, New algorithms for generalized network flows, *Mathematical Programming*, **64**, (1994), 325–336.

[17] A. Colorni, and G. Fronza, Reservoir management via reliability programming *Water Resources Research*, **12 (1)**, (1976), 85–88.

[18] G. B. Dantzig, Linear programing and extensions. *Princeton University Press*, (1963), Princeton NJ.

[19] M. S. Daskin, Network and Discrete Location: Models, Algorithms and Applications , John Wiley and Sons, Inc., New York, (1995).

[20] M. S. Daskin, C. Coullard, and Z.-J. M. Shen, An Inventory-location model: formulation, solution algorithm and computational results, *Annals of Operations Research*, **110**, (2002), 83–106.

[21] D. Dentcheva, A. Prékopa, and A. Ruszczyński, Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming* **Ser. A 89:**, (2000), 66–77.

[22] P. Doulliez, and E. Jamoulle, Transportation networks with random arc capacities *R. A. I. R. O* **3**, (1972), 45–60.

[23] Z. Drezner, Spherical approximations for minimax location problems, *Computers and Industrial Engineering*, **28**, (1995), 893–897.

[24] Z. Drezner, Facility Location: A Survey of Applications and Methods, Springer, (1995).

[25] J. R. Evans, and J. J. Jarvis, Network topology and integral multicommodity flow problems, *Networks*, **8, (2)**, (1978), 107–119.

[26] C. Fábián, A. Prékopa, and O. Ruff-Fiedler, On a dual method for a specially structured linear programming problem with application to stochastic programming, *Optimization Methods and Software* **17**,(2002), 445–492.

[27] L.R. Ford and D.R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, N.J, (1962).

[28] E. Foufoula-Georgiou, and P.K. Kitanidis, Gradient dynamic programming for stochastic optimal control of multidimensional water resources systems, *Water Resources Research,* **24**, (1988), 1345-1359.

[29] O. Friedler, A. Prékopa, and C. Fábián, On a dual method for a specially structured linear programming problem with application to stochastic programming, *Optimization Methods and Software* **17**, (2002), 445–492.

[30] D. R. Fulkerson, The Hitchcock transportation problem, *The RAND Corporation*, (1959), **Paper P-890**

[31] S. Gal, Optimal management of a multireservoir water supply system, *Water Resources Research,* **15**, (1979), 737?-749.

[32] D. Gale, A theorem on flows in network. *Pacific J. Math.* **7,** , (1957), 1073–1082

[33] M. K. Gavurin, The Application of mathematical methods in problems of weight flow analysis, *Collection of Problems Concerned with Increasing the Effectiveness of Transports*, Publication of the Akademii Nauk SSSR, Moskow-Leningrad, (1949), 110–138.

[34] F. Glover, D. Klingman, and N. Philips, Netform modeling and applications, *Interfaces*, **20**, (1990), 7–27.

[35] D. Goldfarb, and Z. Jin, A faster combinatorial algorithm for the generalized circulation problem, *Mathematics of Operations Research*, **21**, (1996), 259–539.

[36] G. Y. Handler, and P. B. Mirchandani, Location on Networks: Theory and Algorithms. MIT Press, MA, (1979).

[37] R. Hassin, and E. Zemel, Probabilistic analysis of the capaciated transportation problem, *Mathematics of Operations Research* **13 (1)**, (1988), 80–89.

[38] F. L., Hitchcock, The Distribution of a product from several sources to numerous localities, *Journal of Math and Physics* **20**, (1941), 224–230.

[39] A.J. Hoffman, Some recent applications of the theory of linear inequalities to external combinatorial analysis, Proceedings of Symposia in Applied Mathematics, Vol. X. Combinatorial Analysis *American Mathematical Society* , (1960), 113–127.

[40] M. H. Houck, A chance-constrained optimization model for reservoir design and operation, *Water Resources Research*, **15 (5)**, (1979), 1011-1016.

[41] M. H. Houck, and B . Datta, Performance evaluation of a stochastic optimization model of for reservoir design and management with explicit reliability criteria, *Water Resources Research*, **17 (4)**, (1981), 827–832.

[42] H. G. Huang, and D. P. Loucks, An inexact two-stage stochastic programming model for water resources management under uncertainty, *Civil Engineering and Environmental Systems,* **17 (2)**, (2000), 95–118.

[43] T. P. Hughes, Networks of Power Electrification in Western Society 1880-1930. *The John Hopkins University Press* London, (1983).

[44] D. Hunter, An upper bound for the probability of a union, *Applied Probability Trust*, **13**, (1976) 597–603.

[45] W. S. Jewell, Optimal flow through networks with gains, *Operations Research*, **10**, (1962), 476–499.

[46] E. F. Joeres, ,J .C . Liebman, and C. S. ReVelle, Operating rules for joint operation of raw water resources, *Water Resources Research*, **7 (2)**, (1971), 225– 235.

[47] E. F. Joeres, G. J. Seus, and H. M. Engelmann, The linear decision rule (LDR) reservoir problem with correlated inflows, Model development, *Water Resource Research*, **7 (1)**, (1981), 18–24.

[48] L. Kantorovitch, On the Translocation of masses, *Computational Red. (Doklady) Aced. Science* **37**, (1942), 199-201.

[49] D. R. Karger, A Randomized fully polynomial time approximation scheme for the all-terminal network reliability problem, *SIAM Review*, **43 -3**, (2001), 499-522.

[50] F. P. Kelly, Loss networks, *The Anals of Applied Probability* **1 (3)**, (1991-b), 319–378.

[51] F. P. Kelly, Network routing, *Philosophical Transactions: Physical Sciences and Engineering,* **337- 1647**, (1991-a), 343–367.

[52] J. E. Kelly, The Cutting Plane Method for Solving Convex Programs, *SIAM Journal of Applied Mathematics* **11**, (1960), 703–712.

[53] J. E. Kelly, Critical path planning and scheduling: Mathematical Basis, *Operations Research* **9**, (1961), 296–320.

[54] M. Klein, A primal method for minimal cost flows with applications to the assignment and transportation problems, *Management Science*, **14**, (1967), 205–220.

[55] W. Kirby, C. S. ReVelle, and E. Joeres, Reply, *Water Resources Research*,**6 (4)**, (1970), 1242–1245.

[56] T. C. Koopmans, and S. Reiter, A Model of transportation, *Activity Analysis of Production and Allocation* Cowles Commission Monograph,Wiley **13**, (1951), 222–259.

[57] B.F. Lamond, and M.J. Sobel, Exact and approximate solutions of affine reservoirs models, *Operations Research,* **43** , (1995), 771?-780.

[58] D. P. Loucks, Stochastic methods for analyzing river basin systems, research project technical completion report, OWRR project C-1034, Dep. of Water Resource Eng. Cornell Univ., Ithaca, N.Y., (1969).

[59] D. P. Loucks, Some comments on linear decision rules and chance constraints, *Water Resources Research*, **6 (2)**, (1970), 668–671.

[60] D. P. Loucks, and P. J. Dorfman, An evaluation of some linear decision rules in chance-constrained models for reservoir planning and operation, *Water Resources Research*, **11 (6)**, (1975), 777-782.

[61] J. T. Lovelace, and C. N. Strauser, Protecting society from flood damage, a case study from the 1993 Upper Mississippi River flood, *U.S. Army Corps of Engineers*, (1993).

[62] M. A. Marino, and B. Mohammadi, Reservoir management: A reliability programming approach, *Water Resources Research*, **19 (3)**, (1983), 613–620.

[63] H. Markowitz, Portfolio selection, *The Journal of Finance*, (1952), 77–91.

[64] L. B. Miller, and H. M. Wagner, Chance constrained programming with joint constraints, *Operations Research* **13**, (1965), 930–945.

[65] M. Minoux, Network synthesis and optimum network design problems: Models, solution methods and applications, *Network* **19**, (1989), 313–360.

[66] P. B. Mirchandani, and R. L. Francis, Discrete Location Theory, John Wiley & Sons Inc., New York, (1990).

[67] G. Monge, Mémoire sur la théorie des d'éblais et de remblais,*Histoire de Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*, (1781), 666–704.

[68] P. A. P. Moran, The theory of storage, Methuen, London, (1959).

[69] J. P. Penot, L'Optimisatioan la Pareto:D eux au TroisC hoseq ue Je Sais d'Elle(preprint, Pau), (1978).

[70] A. Prékopa, On Probabilistic Constrained Programming, *Mathematical Programming Study*, **28**, (1970), 113–138.

[71] A. Prékopa, Logarithmic concave measures with applications to stochastic programming, *Acta Sci. Math. (Szeged)* **32**, (1971), 301–316.

[72] A. Prékopa, Contributions to the theory of stochastic programming, *Mathematical Programming* **4**, (1973), 202–221.

[73] A. Prékopa, On logarithmic concave measures and functions, *Acta Sci. Math. (Szeged)* **34**, (1973), 339–343.

[74] A. Prékopa, Logarithmic Concave Measures and Related Topics, *Proceedings of the International Conference on Stochastic Programming, Oxford, 1974*, New York, (1978).

[75] A. Prékopa, T. Rapcsák, and I. Zsuffa, Serially Linked Reservoir System Design Using Stochastic Programming, *National Water Authority of Hungary,* **14, 4**, (1978), 672–678.

[76] A. Prékopa, and T. Szantai, Flood Control Reservoir System Design Using Stochastic Programming, *Mathematical Programming Study*, **9**, (1978), 138–151.

[77] A. Prékopa, and T. Szantai, Optimal Control of Storage Level Using Stochastic Programming, *European Journal of Operations Research*, **3**, (1979), 175-189.

[78] A. Prékopa, Network planning using two-stage programming under uncertainty, *Recent Results in Stochastic Programming. Lecture Notes in Economics and Mathematical Systems 179* (P. Kall, A. Prékopa, eds, Springer, Berlin), (1980), 216–237.

[79] A. Prékopa, Boole-Bonferroni inequalities and linear programming, *Operations Research*, **36**, (1987), 145–162.

[80] A. Prékopa, and E. Boros, Closed form two-sided bonds for probabilities that at Least r and exactly r out of n events occur, *Mathematics of Operations Research*, **14**, (1989), 317-342.

[81] A. Prékopa, Dual method for the solution of a one-stage stochastic programming problem with random RHS obeying a discrete probability distribution, *ZOR- Methods and Models of Operations Research,* 34, (1990), 441–461.

[82] A. Prékopa, and E. Boros, On the existence of a feasible flow in a stochastic transportation network, *Operations Research* **39,** (1991), 119–129.

[83] A. Prékopa, Stochastic programming. *Kluwer Scientific Publishers*, Boston, (1995), 84, 107, 180–190, 210, 315, 363, 364, 368, 369, 375, 387, 452, 504, 507, 510.

[84] A. Prékopa, B. Vizvári, and T. Badics, Programming under Probabilistic Constraint with Discrete Random Variable, New Trends in Mathematical Programming, Kluwer Academic Publishers, Netherlands, (1998), 235–255.

[85] A. Prékopa, B. Vizvári, and G. Regös, A Method of disaggregation for bounding probabilities of boolean functions of events, Rutcor Research Report, 21-97, (1998).

[86] A. Prékopa, The Use of discrete moment bounds in probabilistic constrained stochastic programming models, *Anals of Operations Research*, **85**, (1999), 21–38.

[87] A. Prékopa, On the relationship between probabilistic constrained, disjunctive and multiobjective programming, *RUTCOR Research Report,* 07–2007, (2007).

[88] A. Prékopa, T. Szántai, and I. Zsuffa, Természetes Vizfolyások Ontozoviz Készle]'enek Meghatározását Célzo Matematikai Modellek, *Alkalmazott Mathematika Lapok*, **27**, (2010), 175–188.

[89] A. Prékopa, Multivariate value at risk and related topics, *Annals of Operations Research*, **193 (1)**, (2012) 49–69.

[90] A. Prékopa, J. Mayer, B. Strazicky, I.Deák, J. Hoffer, Á. Németh, and B. Potecz, Short-term Optimal Scheduling of Power Generation with Thermal Power Generation with Thermal Power Plants under Network Constraints, Springer, (2012).

[91] D. Pisinger, A Minimal algorithm for the multiple-choice knapsack problem, *European Journal of Operations Research* **83** , (1995), 394–410.

[92] T. Radwan, A. Habib, R. Alseedy, and A. Elsherbeny, Bounds for increasing multi-state consecutive k-out-of-r-from-n: F system with equal components probabilities, *Applied Mathematical Programming*, **35**, (2011), 2366–2373.

[93] C. S. ReVelle, E. Joeres, and W. Kirby, The linear decision rule in reservoir management and design, Development of stochastic model, *Water Resources Research*,**5 (4)**, (1969), 767–777.

[94] C. S. ReVelle, and J. Gundelach, Linear decision rule in reservoir management and design, A rule that minimizes output variance, *Water Resources Research*, **11 (2)**, (1975), 197–203.

[95] T. Santoso, S. Ahmed, M. Goetschalckx, and A. Shapiro A Stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operations Research*, **167, (1)**, (2005), 96–115.

[96] M. Schwartz, Telecommunication Networks, Protocols, Modeling and Analysis *Addison Wesley Publishing Company*, (1987).

[97] H. D. Sherali, and E. P. Smith, An Optimal replacement-design model for a reliable water distribution network system. I *Integrated Computer Applications in Water Supply*, **1** Research Studies Press Ltd., Somerset, England, (1993), 61–75.

[98] J. Shu, C. Teo, and Z.M. Shen, Stochastic transportation-inventory network design problem, *Operations Research* **53, (1)**, (2005), 48–60.

[99] S. Simonovic, Two-step algorithm for design-stage long-term control of a multi-purpose reservoir, *Advanced Water Resources*, **2**, (1979), 47–49.

[100] S. Simonovic, and M. A. Marino, Reliability programming in reservoir management, Single multipurpose reservoir, *Water Resources Research*,**16 (5)**, (1980), 844–848.

[101] S. Simonovic, and M. A. Marino, The role of unconditioned and conditioned cumulative distribution functions in reservoir reliability programming, *J. Hydrol.*, **52 (3/4)**, (1981), 219–228.

[102] S. Simonovic, and G. T. Orlob, Risk-reliability programming for optimal water quality control, *Water Resources Research*, **20 (6)**, (1984), 639–646.

[103] W. Stadler, A Survey of multicriteria optimization of the vector maximum problem, Part 1: 1776-1960, *J. Optim. Theory Appl.*, **29**, (1979), 1–52.

[104] J. R. Stedinger, B. Sule, and D. Pei, Multiple-reservoir system screening models,*Water Resources Research*, **19 (6)**, (1983), 1383–1393.

[105] J. R. Stedinger, B. F. Sule, and D. P. Loucks, Stochastic dynamic-programming models for reservoir operation optimization, *Water Resources Research,* **20 (11)**, (1984), 1499–1505.

[106] E. Stiemke, Über positive lösungen homogener linearer gleichungen, *Mathematische Annalen* **76,** (1915), 340-342.

[107] B. J. Strycharczyk, and J. R. Stedinger, Evaluation of a "reliability programming" reservoir model, *Water Resources Research,* **23 (2)**, (1987), 225–229.

[108] T. Szántai, A Computer Code for Solution of Probabilistic- Constrained Stochastic Programming Problems, *Numerical Tech- niques for Stochastic Optimization* (Y. Ermoliev, and R. J.-B. Wets, eds.), Springer-Verlag, New York, (1988), 229–235.

[109] B. K. Thapalia, T.G. Crainic, M. Kaut, and S.W. Wallace, Single-commodity stochastic network design with multiple sources and sinks. *Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation* CIRRELT, (2010), 59.

[110] B. K. Thapalia, M. Kaut, S.W. Wallace, and T.G. Crainic. Single source single-commodity stochastic network design, *Computational Management Science*, Springer, Berlin, (2010), 1619.

[111] A. Turgeon, A decomposition method for the long-term scheduling of reservoirs in series, *Water Resources Research,* **17**, (1981), 1565–1570.

[112] A. F. Veinott, The Supporting Hyperplane Method for Uni- modal Programming.*Operations Research* **15**, (1967), 147–152.

[113] B. Vizvári, The Integer programming background of a stochastic integer programming algorithm of Dentcheva-Prèkopa-Ruszczyński, *Optimization Methods and Software* **17,** (2002), 543–559

[114] S. W. Wallace, Decomposing the requirement space of a transportation problem into polyhedral cones, *Mathematical Programming Study* **28**, (1984), 29–47.

[115] S. W. Wallace, and R.J.-B. Wets, The facets of the polyhedral set determined by the Gale- Hoffman inequalities, *Mathematical Programming* **62,** (1993), 215–222.

[116] S. T. Waller,and A. K. Ziliaskopoulos, A chance-constrained based stochastic dynamic traffic assignment model: Analysis, formulation and solution algorithms, *Transportation Research* **Part C 14**, (2006), 418–427.

[117] G. R. Wood, The bisection method in higher dimensions, *Mathematical Programming*, **55**, (1992), 319-337.

[118] S. Yakowitz, Dynamic programming applications in water resources, *Water Resources Research,* **18**, (1982), 673–696.

[119] M. A. Yazici, and K. Ozbay, Impact of probabilistic road capacity constraints on the spatial distribution of hurricane evacuation shelter capacities, *Transportation Research Record* Vol.2022, (2007), 55–62.

[120] T. Zhang, A rail/road intermodal facility location problem with stochastic demand, *Stochastic Programming Course Project*, (2008).

[121] en.wikipedia.org/wiki/Kirchhoff'scircuitlaws

[122] en.wikipedia.org/wiki/GaspardMonge

[123] New Jersey Office of Emergency Management. www.state.nj.us/njoem. Accessed April 2, 2012.

Vita

Merve Unuvar

2012 Ph. D. in Operations Research, Rutgers University

2009 M. Sc. in Operations Research, Rutgers University

2007 B. Sc. in Industrial Engineering, Bilkent University

2007-2012 Teaching assistant, Department of Mathematics, Rutgers University, NJ

2012 Research Intern, IBM T. J. Watson Research Center, Cambridge, MA

2011 Research Intern, IBM T. J. Watson Research Center, Cambridge, MA

2010-2011 Research Intern, Dun and Bradstreet, Shorth Hills, NJ

2012 A. Prékopa, M. Unuvar. Single Commodity Stochastic Network Design under Probabilistic Constraint with Discrete Random Variables, Operations Research, (submitted) Available as RUTCOR Research Report 9-2012.

2012 b M. Unuvar, O. Mynduk and A. Prékopa. Serially Linked Reservoir Network Design Problem with consecutive k-out-of-n type Reliability, working paper.

2012 a M. Unuvar, E.E. Ozguven and A. Prékopa. Solution of an Optimal Reservoir Capacity Problem under Probabilistic Constraints, Computational Management Science, (submitted) Available as RUTCOR Research Report 14-2012.

2011 M. Unuvar, G.T. Lakshmanan, Y. N. Doganata, R. Khalaf. Generating Predictions for Business Processes whose Execution is Driven by Data, IBM Patent, YOR9-2011-0642US1.

2011 M. Unuvar, G.T. Lakshmanan, Y.N. Doganata, A System and Method to Train Decision Support Systems from Business Process Execution Traces That Contain Repeated Tasks, IBM Patent, YOR8-2011-1053US1.

2011 M. Unuvar, G. T. Lakshmanan, Y.N. Doganata. Using Decision Trees to Predict the Behavior in Data-Driven Semi-Structered Business Processes with Loops and Parallelism, Knowledge and Information Systems (submitted).

2011 G. T. Lakshmanan, D. Shamsi, Y.N. Doganata, M. Unuvar et al. A Markov Prediction Model for Data-Driven Semi-Structured Business Processes, to be submitted: Knowledge and Information Systems, (submitted).

2011 A. Prékopa, M. Unuvar. Stochastic Network Design Problem with Probabilistic Constraint, 2011 INFORMS Conference, North Carolina (Stochastic Programming Session Chair)

2010 A. Prékopa, M. Unuvar. Stochastic Network Design with Random Demand and Arc Capacities, 2010 EURO Conference, Lisbon, Portugal.

2009 A. Prékopa, M. Unuvar. Bounding the Reliability of a Stochastic Transportation Network with Random Demands and Arc Capacities, 2009 EURO Conference, Bonn, Germany.

2008 M. Unuvar, O. Cavus and A. Prekopa. Improved Bounds on the Existence of a Feasible Flow in a Stochastic Transportation Network, DIMACS (Center for Discrete Mathematics and Theoretical Computer Science) received DIMACS Summer Project award stipend, available as RUTCOR Research Report 23-2012.