Stochastic Single-Commodity Network Design

Biju K. Thapalia

A dissertation submitted to Molde University College
for the degree of Philosophiae Doctor

PhD thesis in Logistics 2010:2

Molde University College

Molde, Norway 2010

Biju K. Thapalia
Stochastic Single-Commodity Network Design

PhD thesis in Logistics 2010:2

Molde University College
P. O. Box 2110
NO–6402 MOLDE, Norway
www.himolde.no

This dissertation can be ordered from Molde University College Library
biblioteket@himolde.no

Printing: EKH Trykk AS

ISBN: 978-82-7962-123-2
ISSN: 0809-9588

Dedication

To my mother
for letting me pursue my dreams
for so long
so far away from home

&

To my loving wife
for giving me
new dreams to pursue

To my late father
who encouraged me
to think positive
. . . always

Preface

Network design arising in many of the systems which surround us is an interesting area to explore, more so when we realize that many of the deciding factors are uncertain at the time of design. This thesis begins with a case which takes us to the work of understanding what stochastic network design are. We do understand that deterministic consideration in designing a network which ought to operate in uncertain condition will not be good. But this thesis explores in what way deterministic consideration is bad or good for a stochastic situation and ends with understanding the characteristic of an optimal stochastic network. The four papers and Introduction chapter included for this thesis studies to understand network structure under uncertainty.

The thesis has been prepared at Molde University College, Norway as part of the Ph.D degree in Logistics. The work was carried out during the period from August 2005 to May 2010, with Professor Stein W. Wallace as the main thesis advisor and Dr. Michal Kaut as co-supervisor. Professor Teodor G. Crainic have also contributed in advising me in my work. Most of the work is done in Molde University College, along with time spent on filed work, at Nepal Oil Corporation, Kathmandu, Nepal. In connection to my thesis work I had been to The Centre for Research on Transportation (C.R.T) in Montreal, Canada in December 2006.

The work has been evaluated by a committee of Associate Professor Pavel Popela from Brno University of Technology, Brno, Czech Republic, Dr. Nina Detlefsen, Operations Researcher, from Energinet.dk, Fredericia, Denmark, and Associate Professor Johan Oppen from Molde University College, Molde, Norway.

The work embodied in this thesis is born from inspiration, philosophy, love and commitment.

Acknowledgments

The final shape of this work comes from seen or not seen contribution of many people over the years.

First of all I would like to thank my main supervisor Professor Stein W. Wallace, who have been a driving force behind what I have accomplished today. His ideas have inspired me all throughout my PhD work. I had the privilege to work under him and learn a lot more than what have been translated in this thesis. I equally thank my co-supervisor Dr. Michal Kaut, who has shown how important it is to be sharp and focused. His help was always there whenever I wished for. I cannot enough express my gratitude to both my supervisors.

I also thank Professor Teodor G. Crainic, for advising me in my work which has helped to improve my papers as they stand now. Discussions with him were always fruitful and helped to focus my work.

I thank Professor Øyvind Halskau, Associate Professor Berit Helgheim and Professor Irina Gribkovskaia for showing early encouragement in my studies without which I might not have initiated toward my PhD work. I also thank the administration office, the IT-department and the library for excellent service and help. I am equally thankful to my colleagues at Molde University College for help and support given to me.

I also thank officials at Nepal Oil Corporation, Nepal for helping me to provide with relevant data and proper insight to their work. I am grateful to the Centre for Research on Transportation, Université de Montréal, Canada for inviting me for academic work.

I also owe a big thanks to Norwegian Quota program and Molde University College for providing scholarship to carry out my academic work.

I especially appreciate my time with my friends Uttam, Ganesh, Bharat, Aliaksandr, Krystsina, Fubin, Jianyong, Omar and Øyvind who has helped me to overcome my anxiety, stress and hard times and participated in my joy and success. I also thank my colleagues at Management Campus, Purbanchal University, Nepal for making it possible for me to stay here for my thesis work.

I express my gratitude to my friend Jarle Sanden, who has been a constant support in my ups and downs in this foreign land. He made me feel at home by providing warmth like a family member. Kåre and Anne, Sadhona, Ole Kjell and Ragnhild, Dagfinn and Eli and many others from Molde, made my social staying here a pleasurable one. I will always remain thankful for this.

Finally, I would like to thank my family for their unconditional support on my endeavors. I am sure my late father would be happy for this day in my life. I owe a lot to him. I thank my mother who taught me the values which makes me what I am. I will always remember her silent sacrifice, for us to succeed. My wife, Moon, has always introduced cheer to my mood and her love and care has kept my spirit high. I thank her for true support. My sister Meena and brother-in-law Dhurba Thapa's help to me in this period is always I will be indebted with. I finally thank my brother Ajay, sister-in-law Lila, father-in-law Professor Ram B. Thapa, and all other family members who have prayed for my success.

Molde, Norway — Biju K. Thapalia
May, 2010

Contents

Introduction

"Creativity can be described as letting go of certainties."

Gail Sheehy
Author and writer

Networks and network structures arise naturally in many of the systems which surround us in our day to day life, such as communication, transportation, and manufacturing and distribution networks. They are subject to (re)design and analysis. Some of them are as old as our civilization, like water distribution systems, water channels networks, underground sewage networks, or road networks. These networks, in various forms, are built with huge investments, last for many years, and the investments have direct affects on the operational performances in years to come.

So, general network design problems arise in many application contexts. But not only that, some classical combinatorial optimization problems such as the traveling salesman problem, the minimum spanning tree problem, and the Steiner tree problem are special cases of the general network design problem (Magnanti and Wong, 1986). The study of these networks has been the very core of management science and operation researchers for a long time, and the goals have been to find theoretical, algorithmic, and practical insight into the problems (Magnanti and Mirchandani, 1993).

Designing networks in a deterministic setting is not sufficient as many of the parameters which shape a good network are not fixed or cannot be predicted at the time of the design. Strategic decisions, like which links to open between the source and demand points, or link features such as capacity, are *here-and-now* decisions, and they are mostly investment decisions (but they can also simply be large irreversible decisions such as the published schedule of an airline for the following six months). These networks will typically be used on a daily / regular basis for a long period to come, requiring decisions related to supply, transportation, demand fulfillment and storage, all associated with costs, revenues and service levels. The adequate design of a network requires the anticipation of these future activity levels and ignoring the uncertainty on these lead to inferior designs.

Standard textbooks on stochastic programming, such as Kall and Wallace (1994) and Ruszczyński and Shapiro (2003), highlight how ignoring uncertainties can lead to incorrect or poor decisions and suggest stochastic models to address problems with uncertainties in the parameters. Most of the literature dealing with the problems discussed above

focus on efficiency in solving large size problems. But still there is almost no understanding of the basic differences between the network structures that arise from stochastic modeling of the problems as compared to deterministic approaches to the same. We believe that the understanding of what brings flexibility to a network that is designed considering uncertainties in the parameter values will be very helpful to practitioners who have to deal with network planning or operations. It may also be useful in constructing heuristics.

This research is motivated by these observations and the many design problems we encounter in the single-commodity case (which by reformulation also covers some multi-commodity situations). Thus this research focuses on understanding how different or similar the stochastic network designs for the single-commodity cases are as compared to the deterministic counterparts and we hope that it might contribute to better understanding of more general design problems.

The idea to work in this topic arose from my work in the paper on distributing oil in Nepal under uncertain network conditions. It forced us to describe what is a good network structure in a given uncertain environment and we found that we lacked this understanding. Thus my focus is on understanding optimal network structures in stochastic environments.

Scientific contribution

Network planners and designers almost always face the situation that they have to plan for an uncertain future in terms of demand / supply, and / or in terms of link failures or unstable capacities in the links. Despite many publications on heuristics, it still is very difficult to find a good solution in a short time. This thesis gives basic ideas to the planners and designers as to what brings flexibility (and better expected performance) into a network structure. This qualitative understanding regarding optimal network structures under uncertainty is useful when trying to understand how a given network will perform or how to build or update existing networks. Also this thesis highlights the importance of correlations among the random variables, as we find that they have considerable affects on the designs. Another general contribution is to inform planners about how uncertainty in different parameters in the network have different effects on what constitute good designs. What copes well with one type of uncertainty might be rather useless for another.

To the best of our knowledge, our use of what we name Comparisons B and C to test the quality of deterministic solutions and their relationships to their stochastic counterparts is new. This is despite the fact that these tools are very straightforward. Comparison B takes information about which edges to open (but not their capacities) from the deterministic designs, and fixes the corresponding zero-one variables in the stochastic mixed-integer design model. The model then turns into a stochastic linear model, which in turn is used to set the capacities. Hence, we test the *structure* (which we denote the *skeleton*) of the deterministic designs by allowing capacities to be adjusted to the stochastic environment. Comparison C checks to what extent a full deterministic solution (skeleton plus capacities) can be updated to become a good (if not optimal) solution to the stochastic problem. This shows us if the deterministic design forms a backbone of a good stochastic design or not.

Each paper of the thesis has its own contributions that form building blocks in a final

and more complete understanding of the effects of uncertainty. The core of the thesis is the work on network design, which are the last three papers.

Paper 1 **(Thapalia, Wallace, and Kaut, 2009b)** studies a distribution network operating in an uncertain environment. It defines the uncertainties faced when distributing oil in Nepal (a problem not previously modeled). A unique modeling approach is used where a time-space version of the distribution network is looped back on itself. By manipulating the time line this creates a three stage model of an infinite horizon stochastic problem. The looping back method is not a new way to handle end-of-horizon problems, but the way we use it to determine optimal inventory levels under steady-state uncertainty in the distribution network is unique. Our modeling of a *fairness policy factor* in a system where there is always under-supply is also new. We study the relationships between fairness factors and inventory levels and thereby give new insight to the policy makers. The paper provides valuable insight into the steady-state inventories to be maintained during normal periods in anticipation of random events. Also with this work it is possible to see the interaction of randomness of different parameters on the decision variables.

Paper 1 is the inspiration for the thesis' next three papers – which all shed light on the understanding of optimal stochastic network structures.

Paper 2 **(Thapalia, Crainic, Kaut, and Wallace, 2010b)** is on single source single commodity stochastic network design. The major scientific contribution here is that we present a problem class where, in fact, deterministic models produce useful information for the stochastic case. The deterministic skeleton (a tree rooted at the source node) is valuable for the stochastic problem (where the solution must *contain* a tree), as this tree works well in the stochastic situation if capacities are adjusted, more so for designs with high fixed setup costs. Paper 2 also points out that when there are large negative correlations, moderate fixed setup costs compared to capacity cost, and large variation in demand – the stochastic skeleton is rather different from the deterministic one. In all other situations the deterministic skeleton is a good starting point for stochastic settings.

Paper 2 contributes to the understanding of what constitutes a robust design for a single-source single-commodity stochastic network design problem.

Knowing that the stochastic structures are different from deterministic counterparts in certain ways, we wanted to study the multi-source case. In **Paper** 3 **(Thapalia, Crainic, Kaut, and Wallace, 2009a)**, we therefore investigate to what degree the observations from the single-source case carry over to the case of multiple sources and/or sinks. This paper contributes in understanding that the deterministic structure (skeleton) is no longer good, but taking the deterministic solution to built upon by adding extra edges and capacities is a good option for stochastic settings. As the number of source nodes increases in the network the performance of the deterministic structure, which is a forest of many small trees, becomes less suitable for stochastic settings due to less possibility of sharing capacities. This paper contributes in stating that in a stochastic network structure, when demands are negatively correlated, they share paths among them from one or more source nodes. And from a source node point of view, it is best to connect with demand nodes which have as small correlations (preferably negative) in demand as possible. The paper

also highlights that consolidation of flow in the network takes place when source nodes and clusters of demand nodes are far from each other, more so when setup costs are high. Paper 3 presents the reason for loop formation and its usefulness in providing flexibility to the network structure.

Paper 4 **(Thapalia, Crainic, Kaut, and Wallace, 2010a)** concentrates on the study of stochasticity of edge capacities. In other words, we study the capacity as a random variable, not just the on/off situation. The contribution of this paper is in finding that flexibility in network designs under stochastic edge capacities comes from the presence of higher number of alternative routes combines with higher installed capacities. This paper also points out how differently flexibility comes about in the cases of random demand and random edge capacities.

With lower fixed setup costs compared to variable setup costs, the deterministic skeletons have more edges open. This is better in stochastic settings as it generally provides more paths and connections in the networks. As a consequence, if deterministic models are run using expected capacities rather than maximal (installed) capacities, they produce better skeletons, since, from a deterministic perspective, using expected capacities effectively increases the variable setup costs relative to the fixed setup costs.

The paper points out that correlations again have important effects on the structure of the designs. With uncorrelated edge failures, flexibility comes from more edges in the design, while for positively correlated edge failures, the flexibility is to a larger extent based on higher installed capacities. The thesis also finds that loops are predominant in the designs as they provide two connections between any pair of nodes in the loop, and provide consolidation effect as in the deterministic situation. The paper contributes by saying that deterministic skeletons do better for stochastic edge capacities than for stochastic demand as flexibility comes from more connections rather then consolidation.

Suggestions for further research

The thesis has given us a good understanding of what happens to network structures when we see randomness in demand or in capacities on the edges independently. So the natural next question to answer is what will be the network structure when both demand and capacities on edges are random at the same time? The interaction of these two will certainly have affect on the way flexibility is achieved in the network as compared to what we have found. This should definitely be the next step in the future work.

We started with the goal that our results should help to find better heuristics for solving difficult stochastic network design problem. This can be another direction of research, to put into real use what we have found in this thesis.

In the Papers 2 and 3 we assumed that networks had to be *designed* to handle demand uncertainty. But there is another approach to manage these uncertainties – by managing demand using price mechanisms. So another line of progress in this research will be to see how a network behaves in the stochastic demand situation by introducing the management of demand by price mechanisms. Will the way flexibility comes about be considerably different as compared to what we have observed?

Paper 1 introduces a new concept of handling infinite horizon stochastic problems. It would be interesting to see if this approach can be used also for other problem classes.

The papers

This section lists the four papers constituting the thesis. For each paper, the contributions of the different authors are stated along with a description of how the paper evolved. We also give information on the publication of the papers and list conferences and workshops where materials from the papers have been presented.

Paper 1 **– Using inventory to handle risks in the supply of oil to Nepal**

This paper is co-authored with my supervisor Stein W. Wallace and co-supervisor Michal Kaut. This work first focused on vehicle routing problems under uncertainty faced by Nepal Oil Corporation (NOC). However, with a change in government policy, the problem of vehicle routing no longer remained a part of NOC, and this brought us to a meeting with the authorities of NOC. Top managers from NOC, Stein W. Wallace and I then discussed the problem presented in this paper. The conceptualization of the problem and the building of the stochastic model was done by me under guidance from my main supervisor. Michal Kaut later provided input to this model. The approach of looping back the network comes from my supervisor, whereas I am solely responsible for describing the scenarios and collecting the relevant data in the field. Since this was the first time the loop back method was used to represent the infinite horizon problem under steady-state uncertainty, the model was gradually improved and adjusted to represent the problem in hand. This was the most tedious and time consuming process which was fully supported in the form of continuous discussion by my co-supervisor Michal Kaut. All implementation, testing and analysis was done by me with some valuable guidance from Michal Kaut, especially in graphical presentation. The major part of the writing was done by me, but my supervisor polished the language.

This paper is published in the *International Journal of Business Process and Supply Chain Modelling* in 2009. A previous version of the paper is published in the Conference Proceedings of the 12th HKSTS International Conference, Hong Kong, 2007. The paper with focus on the problem description was presented in the 2nd Nordic Optimization Symposium, Oslo, 2007 and with focus on computational methods and results was presented in The 12th HKSTS International Conference, Hong Kong, 2007.

Paper 2 **– Single source single-commodity stochastic network design**

This paper is co-authored with my supervisor Stein W. Wallace, and co-supervisor Michal Kaut and Teodor G. Crainic. The motivation for this paper was from my previous paper. The development of the next three papers started with a conceptual discussion with Stein W. Wallace and Teodor G. Crainic during my visit to the Centre for Research on Transportation, Université de Montréal, Montreal, Canada. The development and setting up of the model was done by me in consultation with Stein W. Wallace. He has contributed in conceptual discussions on the methodology. The selection of test instances and adjustment of parameter values are done by me in continuous discussion with my co-authors. The test instance 'Molde' is generated by me. The implementation, testing,

and analysis are done by me in consultation with my co-supervisor Michal Kaut, who also has been responsible for scenario generation. Scenario generation, critical for this thesis, has throughout been based on earlier work by my supervisors. The observations, which constitute the main contributions of the work, are all mine. With guidance from Teodor G. Crainic on the structure of the presentation, most of the writing is done by me and improved by Stein W. Wallace. Two anonymous referees have also helped to achieve more clarity in the paper.

The paper has been conditionally accepted in *Computational Management Science*, a special issue on Optimal Decision Making under Uncertainty. The paper with a slightly different focus was presented by Stein W. Wallace at 20th International Symposium on Mathematical Programming Chicago, 2009.

Paper 3 **– Single-commodity stochastic network design with multiple sources and sinks**

The paper is co-authored with Teodor G. Crainic, Michal Kaut and Stein W. Wallace. The development and setting up of the model is done by me with some assistance from Michal Kaut. I had continuous discussion with all my co-authors regarding the focus of the paper. Updates to the methodology used in the previous paper were done by me. The implementation, testing and the analysis are also done by me. The observations which constitute the contributions of the paper, are all mine. Most of the writing is done by me and improved by Stein W. Wallace. The paper has improved much as a results of critical comments by Teodor G. Crainic.

The paper is selected for presentation in Seventh Triennial Symposium on Transport Analysis, Tromsø, Norway in June, 2010. The paper with a different focus was presented by Stein W. Wallace at 20th International Symposium on Mathematical Programming, Chicago, 2009. It is submitted to a peer-reviewed international journal.

Paper 4 **– Single-commodity network design with random edge capacities**

The paper is co-authored with Teodor G. Crainic, Michal Kaut and Stein W. Wallace. I had continuous discussions with Stein W. Wallace and Teodor G. Crainic regarding how to define the problem as that had a major effect on the focus of the paper. There was a natural tendency to follow the telecommunications formulation of k-connected networks, but we decided to include a broader range of problems by seeing edge capacities as random variables and looking at expected costs. The development and setting up of the model was done by me. The selection of test instances and adjustment of parameter values are done by me in continuous discussion with my supervisors. The implementation, testing, and the analysis in the paper are also done by me. The development in methodology, as compared to previous papers was my contribution. Michal Kaut again provided valuable input on generating the scenarios used in the numerical analysis. Most of the writing is done by me with Stein W. Wallace polishing the final manuscript.

Bibliography

Peter Kall and Stein W. Wallace. *Stochastic Programming*. John Wiley & Sons, Chichester, N.Y., 1994.

Thomas L. Magnanti and Prakash Mirchandani. Shortest paths, single origin-destination network design, and associated polyhedra. *Networks*, 23(2):103–121, 1993.

T.L. Magnanti and R.T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1986.

A. Ruszczyński and A. Shapiro, editors. *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*. Elsevier Science B.V., Amsterdam, 2003.

Biju K. Thapalia, Teodor G. Crainic, Michal Kaut, and Stein W. Wallace. Single-commodity stochastic network design with multiple sources and sinks. November 2009a.

Biju K. Thapalia, Stein W. Wallace, and Michal Kaut. Using inventory to handle risks in the supply of oil to Nepal. *International Journal of Business Performance and Supply Chain Modelling*, 1(1):41–60, 2009b. ISSN 1758-941X (online), 1758-9401 (print).

Biju K. Thapalia, Teodor G. Crainic, Michal Kaut, and Stein W. Wallace. Single-commodity network design with random edge capacities. Feb 2010a.

Biju K. Thapalia, Teodor G. Crainic, Michal Kaut, and Stein W. Wallace. Single source single-commodity stochastic network design. *Computational Management Science*, conditionally accepted for special issue on 'Optimal Decision Making under Uncertainty', 2010b.

Paper 1

Using Inventory to Handle Risk in Supply of Oil to Nepal

Using inventory to handle risks in the supply of oil to Nepal

Biju Kr. Thapalia* Stein W. Wallace† Michal Kaut‡

Molde University College, P.O. Box 2110, NO-6402 Molde, Norway

30 November 2008

Abstract

Nepal's unique geographical features, frequent political disturbances, strikes, a limited and complicated road network, frequent road breakdowns, government interventions as well as a volatile international oil market make the supply chain of Nepal Oil Corporation (NOC) rather unique. We analyze different risks in the NOC supply chain and discuss what can be done to find inventory strategies that handle these risks without a particular focused on one specific scenario.

Keywords: Supply Chain, Stochastic programming, Uncertainty, Linear programming, Inventory

1 Introduction

Nepal Oil Corporation (NOC), established by the Nepal government in 1970, is a public enterprise to import, store and distribute petroleum products in the country. NOC maintains monopoly in this market. Nepal being a landlocked country, the entire oil procurement is done through India.

Earlier, the supply of oil to Nepal was handled, in a rather random fashion, by a few private companies. To regulate these unplanned purchases as well as the resulting distribution, the Nepalese government established NOC. This monopoly state of NOC was not created to make profit but to have effective distribution at reasonable prices for the customers. We can see this from the losses it has suffered when oil prices increased in the last decade. NOC has over the last five years accumulated a loss of 12 billion rupees (approx 300 million US Dollars). The government has decided to extend loans of 1.7 billion rupees on its guarantee to NOC to pay dues to the Indian supplier. As of June 2007, `http://ekantipur.com` estimated that NOC owes 4.5 billion rupees to the Indian Oil Corporation (IOC) . Long queues in front of petrol pumps have become

*`Biju.K.Thapalia@hiMolde.no`
†`Stein.W.Wallace@hiMolde.no`
‡`Michal.Kaut@hiMolde.no`

a common phenomenon in Kathmandu and in the rest of the country, which is of more concern to the government and to the general customers, than NOC's losses.

Oil being a sensitive product, NOC finds it difficult to operate efficiently due to government interventions. Pricing and procurement are political and bilateral issues, which are dictated by the government and the bilateral agreement between Nepal and India. The major job in NOC is that of operational activities which includes storing and distributing oil products. NOC sees possibilities of improving its performance by managing the operational activities of its supply chain in a better way.

The country's unique geographical features, frequent political disturbances, strikes, limited and complicated road network, frequent road breakdowns, government interventions and a volatile international oil market make this supply chain a very complicated one. A study of NOC's distribution network motivated us to better understand the risks and uncertainties associated with the supply chain and how these can be managed. We shall focus on how long-term inventories can be used to manage the risks. This way of management, being rather standard in general, is particularly involved for a company consistently short on foreign currencies.

The paper is organized as follows. The next section explains the supply chain of NOC and the major players in this network; the third section deals with risks in a supply chain. The fourth section dwells on a linear programming formulation of NOC's distribution problem and its possible extensions. The fifth deals with a stochastic version of the model and the scenarios are discussed. The sixth section explains the result of the optimization problem while, finally, the seventh section is the conclusion of the paper.

2 NOC's supply chain and its distribution network

The supply chain for oil distribution in Nepal is unique in the sense that there are not many players and levels in this network. The focal company in this analysis does not have full control of the complete supply chain, as it is heavily dependent on the agreement between India and Nepal at government level and many other external parties. NOC, our focus company, can only increase its efficiency by taking what is given and optimize the network from there. The paper focuses on the distribution network between the refineries and depots of the IOC and the depots of NOC.

2.1 The players in NOC's supply chain

The supply chain starts with the International market where crude oil is bought. The oil market is among the most volatile commodity markets and is extremely sensitive to international events. Over the last decade, the international prices have increased from about 18 dollars to around 140 dollars per barrel, and then dropped to the present level of about 50.

The *Indian Oil Corporation* buys crude oil and processes it into products. NOC imports oil from refineries/depots of IOC situated at different places in India. IOC determines prices for the oil products that include costs of crude oil at Kolkata port, the transportation cost to the refineries and the refining charges. In particular the refining charges are negotiable. *NOC* has 12 depots in Nepal with total storage capacity of around 70300

kilolitre. Presently NOC fills only 15 to 18 percent of the full capacity due to its financial constraints. This covers just a few days of consumption. Nearly 1200 trucks, owned by *independent transport companies*, are on contract to distribute the oil products. Most of the trucks are under trucks associations, which are mostly at regional level, and they do not like to operate outside their own regions. There are approximately 1900 independently owned distribution points throughout the nation. These points cater to the customers' demand of all oil products except aviation fuel. Aviation fuel is distributed directly by NOC. This network of *distributors* makes it possible to reach the end *customers* of NOC. A population of around 27 million, all dependent on oil in some form, directly or indirectly, puts a lot of pressure on this supply chain. Present total demand of oil products per year is around 0.8 million kilolitre. Sixty-five percent of the total demand originates from the Kathmandu valley. Overall there is an annual growth rate of around 10 percent in the demand for petroleum products. The network structure of the supply chain is illustrated in Figure 1.

In Figure 1, refineries are owned by IOC and are situated at different places in India. Presently NOC uses five refineries or depots of IOC to buy petroleum product. The transportation of petroleum product from the refineries to different depots of NOC is done using trucks. The depots in Nepal are served directly or via some of the other depots. The eight most important depots will be in the focus of this study. Some of the depots here act as transshipment points from which smaller sized tankers are used to the mountainous regions. There are several road links between the refineries/depots and depots situated in Nepal, but only few of these links are actually used. For an idea of the actual road links between the depots of NOC, see `http://www.lirung.com/map/map_road/Nepal_Road_Map_e.html`.

3 Risks in the distribution network

Supply chain thinking has become critical in today's business environment. The distribution network, an element of the supply chain, creates links between two tiers of supply chain members. Most of the time, these links are crucial for the success of a business. Due to the dynamic nature of the business environment, there are lots of uncertainties present in the network. Managers need to identify and manage the risks created by these uncertainties that hamper the performance. The network is mostly analyzed from the buyers' point of view with ideas from marketing and network design, in terms of cost and reach. One of the major issues in a network is the risk of failure. The risk to a network is multi-faceted and cannot be captured with a single number, see March and Shapira (1987).

There are three different types of uncertainty in supply chains: demand uncertainty, supply uncertainty and technology uncertainty, see Davis (1993). These uncertainties bring risk to the supply chain. The risk concept has been extensively studied in business contexts and in all studies supply risk is regarded as one of the major risks—see Zsidisin (2003). By studying the supply risk we capture most of the risk arising from the tangible and intangible features of the supply chain.

Meulbrook (2000) defines supply risk as any type of risk that 'adversely affects inward flow of any type of resource to enable operations to take place' whereas Zsidisin,

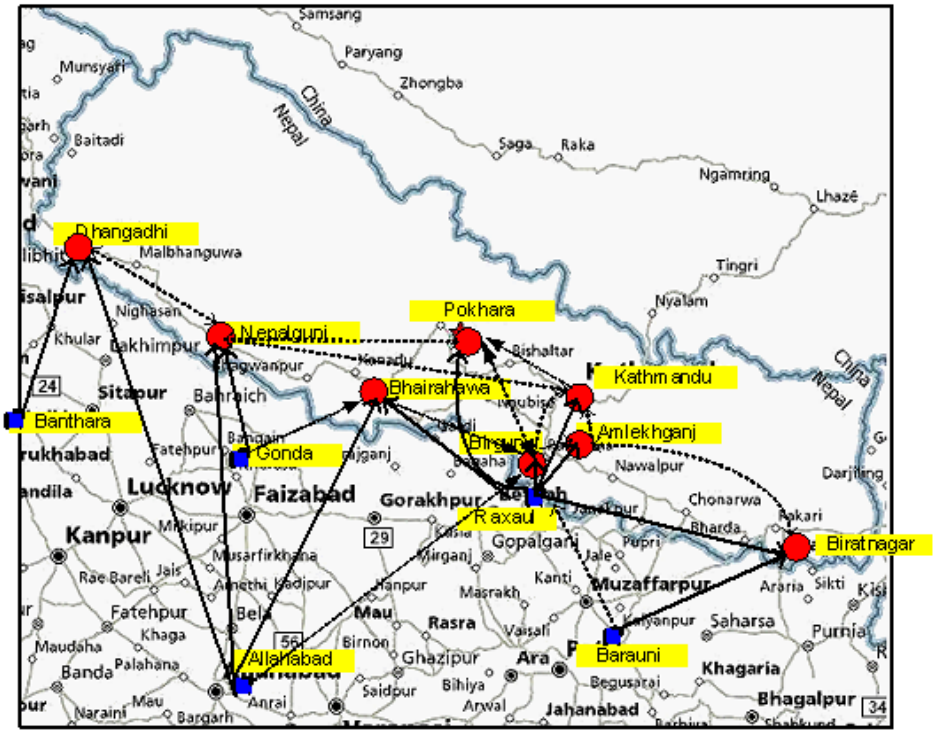

Figure 1: Network structure. The circles denote the NOC depots in Nepal, while the squares denote the IOC refineries. Note that the only low-lands in Nepal are along the Indian border, the rest of the country consists of mountains and the highly elevated Kathmandu Valley.

Panelli, and Upton (2000) define supply risk as 'the transpiration of significant and/or disappointing failures with inbound goods and service'. Zsidisin (2003), from case studies with purchasing organizations involved in supply risk management, concludes that supply risk can be defined as the probability of an incident associated with inbound supply from individual suppliers or the supply market occurring, in which the incident results in the inability of the purchasing firm to meet customer demand or causes threats to customer life and safety.

Most research work on optimization of networks has focused on uncertain demand, but very few have analyzed uncertainty from the supply side. This paper analysis risks due to such as random supply and arc failures.

The network facing NOC has both intangible and tangible features. The intangible features are the political relation between the governments of India and Nepal, the international political scenario, the bargaining power of Nepal with India, and Indian foreign policy towards Nepal. The intangible features of a network are difficult to examine and influence, see Harland, Brencheley, and Walker (2003), but still have profound effects on design decisions and performance of the network. We see that changes in the government and/or managers, change the relation between the management and the government which again affects the decision making style. The tangible features are examinable, but they are also to some extent influenced by the intangible features of the network. Thus identifying, defining, and assessing these risks properly for the given network will be a major task.

The breakdown of a source point due to technical fault can be classified as technology uncertainty. In the given case the breakdown in refineries (source point) will result in disruption of supply so this may also be viewed as a source of supply uncertainty. In the present case the breakdown of any part of the roads due to any climatic condition or man-made situation can also be regarded as supply uncertainty. Also political or organizational fallouts may result in blocking or restriction of supply from seller to buyer, which also may be regarded as supply uncertainty.

NOC does not use any linear programming model to plan the supply chain but uses their experience, basic ideas, and simple mathematics to calculate the figures required for ordering, storing and transportation. This model, however, is built in cooperation with NOC, and reflects their way of thinking. Data was also collected from NOC.

4 Modelling philosophy

There are of course many ways to handle the risks in NOC's supply chain. Reducing those risks that are fully or partly man-made will always have a major focus. Our thinking in this paper is somewhat different. We shall focus on long-term inventory, and study how it changes as the different risks (and other model parameters) change for whatever reason. Inventory as a means of handling risks is particularly involved for a company like NOC which is consistently short on foreign currencies, needed to purchase oil from India.

Obviously, the problem of inventory control while purchasing under a limited period-by-period budget and random disturbances is an infinite horizon problem. The disturbances occur (mostly) independently of each other, and any disturbance can occur at any point in time. As it stands, this problem is not very well suited for stochastic programming. However, we must remember that our main issue is long-term inventory control. That is, we would like to understand how inventory should be controlled when everything runs smoothly in expectation of some disturbance. The expectation is that as the probability of disturbances increases, optimal inventory increases, increasing capital costs. So we expect to see the standard trade-off between inventory costs and shortage costs. But the setting is rather different from a simple inventory model.

We shall first make the somewhat common assumption that only one disturbance can occur at any point in time. Or in other words, that the probability of two disturbances occurring at the same time is so low that we can disregard it. In our case, this assumption is critical.

The present model is not an operational model but rather represents a way to explain the effects of the different random disturbances on long-term inventory. Figure 2 illustrates our modelling. The idea in the model is that we introduce all random events at a particular point in time (arbitrarily denoted 0). At that point inventories and whatever parts of the supply chain that are still working, are used to supply customers. When the events are over, inventory levels must be brought back to their long-term levels (which are our main variables) within time t_1. Thereafter follow t_2 periods of no disturbances. The t_2 periods represent where the model will carry the inventory, anticipating the random events. This is the core of the model – carrying inventory in expectation of some disturbance. The longer t_2, the less likely is an occurrence. As there is also a scenario representing "no disturbance" we have two ways of representing the intensity of distur-

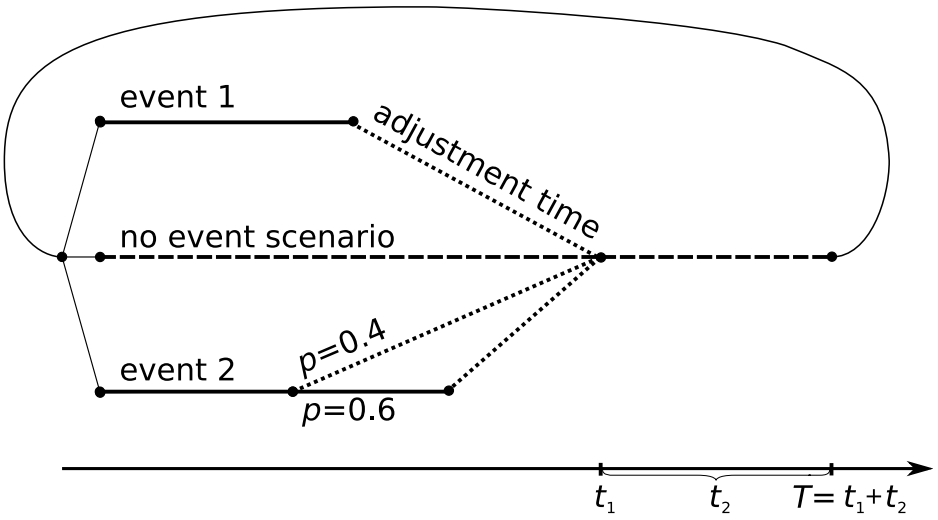

Figure 2: Illustration of events with known and random durations (solid lines), the scenario of no events (dashed line), the period of inventory adjustments (dotted lines), the period of carrying inventory in anticipation of an event (dashed line), and finally the loop.
Event 1 has a known duration, while the duration of Event 2 is stochastic: it has a probability of 40% to finish earlier and 60% of lasting longer then Event 1.

bances: the choice of t_2 and the probability of the no disturbance scenario. After the model reaches time $T = t_1 + t_2$, we loop the model back to the start. This way of looping back can be useful in stochastic models, see for example Lium, Crainic, and Wallace (2007).

So we shall end up with a three-stage stochastic programming model. First stage variables shall be inventory levels in periods of no disturbance, second and third stage variables are all others. Third stage variables are associated with events with random durations (like Event 2 in Figure 2). We could have let also transportation and purchasing decisions be first stage in the same periods, but have chosen not to: there might be problems of stability of those variables around the end points of the stable periods, and our focus is in any case on the long-term (first-stage) inventories.

It is important to note that the second and third stage variables do not in principle represent operational decisions that can be implemented. The model's sole focus is on inventories in stable periods. And even more obviously: The elapse of time in the model does not represent the actual flow of time. We believe this formulation of a three stage model for an infinite horizon problem to be a major contribution of this paper. Of course, the approach works as well for pure two or multiple stages depending on the branching structure of the events.

So we conclude this section by repeating the relationship between our problem understanding and our model. In reality, events occur at random points in time, but with known frequencies (data is available for many of the events). Whenever one occurs, NOC does its best to supply the country with oil using inventory and those parts of the supply chain still functioning. As soon as the event is over, NOC will try to recover the chosen inventory levels. Those inventories will then be carried until the next event starts. So we

see that the less likely the events, the more costly the inventories (in the sense that they are more rarely used). In the model, the quiet period of length t_2 (as well as the no-event scenario) represents when NOC waits for the next event. In the model it is known when the events occur. But the model is not allowed to take that into account since we force inventories to be carried at stable levels throughout the quiet period. Once an event occurs, the model will try to supply oil by using inventories and whatever of the supply chain is working. Then, after the event is over, the model forces a rebuild of the inventories, so that they reach the inventories of the stable period by time t_1. Hence, the major variables – the inventories – function the same way in the model as in reality. They are kept in anticipation of events when all is quiet, and used whenever an event occurs. The assumption of no two events taking place at the same time is needed for this modelling to work. In this case this is a reasonable assumption.

5 LP formulation

We formulate the present distribution philosophy of NOC as an LP. This formulation looks into the distribution of oil products from up to five refineries of IOC in India to a number of depots of NOC in Nepal. The objective function minimizes the overall cost of distribution, inventory and penalties for non-delivery. Following present practice, we always source a depot from the nearest refinery. Since not all refineries deliver all products, a given depot might be sourced from several refineries, but there will never be two refineries sourcing the same product to a given depot. Purchasing of products takes place in USD. Since the Rupee is not convertible, NOC cannot transfer any amount into USD. So instead of having a total budget for all activities, or at least minimizing total costs, we have chosen to minimize costs that occur in Rupees under two major constraints: The availability of USD for purchasing and a requirement that the budget is fully used for purchasing.

We start with a basic LP formulation of the problem when everything is normal. It is a kind of deterministic multi-period multi-commodity production and transportation problem with inventory. To address the end-of-horizon problem, we present the model in a circular fashion. This can be done in a simple way by letting the period following period t be $(t+1) \mod (T+1)$ and the previous period $(t+T) \mod (T+1)$, where $0,\ldots,T$ are the time periods in the model.[1] Since we develop the model in a circular fashion we need not provide initial inventories. The model will itself find the appropriate inventory levels, in fact, that is the purpose of the model. This way we avoid ending up analyzing the build-up (transient) stage of the operation, which is quite different from the steady-state, which is our focus.

Since NOC is short on foreign currencies, there will be rejection of demand. We represent that by piece-wise linear penalties. It is crucial not to use simple linear penalties, as that will result in unreasonable rejections, such as all demand in a given depot on a given day rejected rather than rejections spread over depots and time.

The term "node" might need a brief explanation. In the deterministic version of the problem a node represents all decisions associated with a specific time period (day in

[1] There seems to be a disagreement about the interpretation of $a \mod b$ if a is negative. Hence, we have chosen to let the period preceding period t be $(t+T) \mod (T+1)$ and not $(t-1) \mod (T+1)$

our case). Hence, if we start counting both nodes and time at zero, the time associated with node n, denoted $\mathrm{t}(n)$ is always given by $\mathrm{t}(n) = n$. For the stochastic model that will change. We use n and $\mathrm{t}(n)$ also in the deterministic formulation to make the transition to the stochastic model as easy as possible.

Sets

- $\mathscr{N}$ Nodes
- $\mathscr{D}_1$ Depots that can be reached from their nearest refinery in one day.
- $\mathscr{D}_2$ Depots that can be reached from their nearest refinery in two days.
- $\mathscr{D}$ Set of all depots; $\mathscr{D} = \mathscr{D}_1 \cup \mathscr{D}_2$.
- $\mathscr{I}$ Set of intervals for the piece-wise linear penalty for unsatisfied demand.

Input parameters

- H_k Inventory holding cost per unit of product k per unit of time. As this is mainly capital costs, it does not depend on $j \in \mathscr{D}$.
- C^n_{ik} Unit cost of transporting product k to depot i (from its nearest refinery).
- c^n_{ij} Transportation costs between depots i and j inside Nepal.
- P_k Price of product k, measured in USD.
- g^n_{ik} Fairness factor (percentage of demand fulfilled) for each product at each depot.
- d^t_{ik} Demand for product k at depot i at day t.
- M_{ik} Maximum holding capacity of product k at depot i.
- B Budget in USD for buying products over the time horizon $0, \ldots, T$.
- U Total capacity of trucks available each day.
- b_l Lengths of intervals for the piece-wise linear penalty; $l \in \mathscr{I}$
- $N_{ik,l}$ Coefficients for the piece-wise linear penalty for product i at depot k; $l \in \mathscr{I}$.
- $\mathrm{t}(n)$ The time of node n.
- $\mathrm{pa}(n)$ Parent of node n, i.e. the node preceding it. To make the network circular, we set the parent of the first node to be equal to the last node. This will guarantee that $\mathrm{t}\big(\mathrm{pa}(n)\big) = \big(\mathrm{t}(n) + T\big) \mod (T+1)$.

Decision variables (all depend on n, something that will not be repeated throughout)

- x^n_{jk} Amount of product k transported to depot j. We use preprocessing to determine which refinery will be used. Hence there is no "from" index on x.
- w^n_{ijk} Amount of product k sent from depot i to depot j (both within Nepal)
- z^n_{ik} End-of-day inventory of product k at depot i.
- q^n_{ik} Sales of product k at depot i.
- $r^n_{ik,l}$ Rejected demand of product k at depot i, corresponding to the l-th part of the piece-wise linear penalty; $l \in \mathscr{I}$.

Then solve

$$\min \sum_{k,n} \Big(\sum_j C^n_{jk} x^n_{jk} + \sum_j H_k z^n_{jk} + \sum_{ij} c^n_{ij} w^n_{ijk} + \sum_{l=1}^{3} N_{jk,l} \sum_j r^n_{jk,l} \Big) \tag{1}$$

Subject to

$$\mathbf{1}_{\mathscr{D}_1}(j)\, x^n_{jk} + \mathbf{1}_{\mathscr{D}_2}(j)\, x^{\mathrm{pa}(n)}_{jk} + \sum_{i \in \mathscr{D}} w^n_{ijk} + z^{\mathrm{pa}(n)}_{jk} = q^n_{jk} + \sum_{i \in \mathscr{D}} w^n_{jik} + z^n_{jk} \quad \forall j,k,n \tag{2}$$

$$\sum_{j \in \mathscr{D}} \sum_k \Big(x^n_{jk} + \sum_i w^n_{ijk} \Big) + \sum_{j \in \mathscr{D}_2} \sum_k x^{\mathrm{pa}(n)}_{jk} \le U \quad \forall n \tag{3}$$

$$q^n_{jk} + \sum_i w^n_{jik} \le z^{\mathrm{pa}(n)}_{jk} \quad \forall j,k,n \tag{4}$$

$$q^n_{jk} + \sum_{l=1}^{3} r^n_{jk,l} = d^{\mathrm{t}(n)}_{jk} \quad \forall j,k,n \tag{5}$$

$$\frac{q^n_{jk}}{d^{\mathrm{t}(n)}_{jk}} \ge g^n_{jk} \frac{\sum_{m \in \mathscr{N}} p^m \sum_{i \in \mathscr{D}} q^m_{ik}}{\sum_{m \in \mathscr{N}} p^m \sum_{i \in \mathscr{D}} d^{\mathrm{t}(m)}_{ik}} \quad \forall j,k,n : d^{\mathrm{t}(n)}_{jk} > 0 \tag{6}$$

$$\sum_{n,j,k} P_k x^n_{jk} = B \tag{7}$$

$$z^n_{jk} \le M_{jk} \quad \forall n,j,k \tag{8}$$

$$x^n_{jk}, w^n_{ijk}, q^n_{jk}, z^n_{jk} \ge 0 \quad \forall n,i,j,k \tag{9}$$

$$0 \le r^n_{jk,l} \le b_l\, d^{\mathrm{t}(n)}_{jk} \quad \forall n,j,k,\ l \in \mathscr{I} \tag{10}$$

Here the objective function (1) is the financial representation of the operational activities; the first component is the cost of transporting oil products to the different depots, the second component is holding costs at different depots (mostly capital costs); the third component shows the inter-depot transportation costs and the last term describes the cost associated with rejecting demand.

Constraints (2) maintain the flow of products in the depots. Here, $\mathbf{1}_A(x)$ denotes the *indicator function* of set A, i.e. it is equal to one if $x \in A$ and zero otherwise.

Constraints (3) enforce truck capacity. Here the first part is the truck capacity utilized for all trucks staring out that day, be that from a refinery or a depot. For depots two days away from the refineries we have a second part representing the previous day's purchases which are on the way.

Constraints (4) say that oil sent to other depots plus oil sold to customers must come from inventory. In other words, the incoming volumes x^n_{jk} and w^n_{ijk} cannot be used the same day they arrive—for example because they arrive in the evening.

Constraints (5) make sure that sales plus rejections add up to demand. Since the rejections are non-negative, it also ensures that the sales do not exceed the demand.

Constraints (6) make sure that we overall treat all depots fairly. Note that the numerator of the fraction in the right-hand side is equal to the *total expected sale* of product k, while the denominator is equal to the *total demand* for the product. Hence, the constraint

says that if we satisfy on average $p\%$ of the total demand for product k, then we must satisfy at least $g^n_{jk}\, p\%$ of the demand in each depot j and node n.

Constraint (7) shows the limited budget available to NOC to purchase oil from India. This budget is available in foreign currencies. We use equality as we know that the budget is always too low to satisfy all demand. As this is a cost minimization model, this is our way of expressing that all sales are worthwhile. These constraints, combined with conservation of flow (2), make sure that all we buy is sold, nothing ends up permanently in inventory. In reality prices vary a bit over refineries. However, had we included that in the model, we would have ended up with a difficulty. Since the NOC budget is not big enough to cover all demand, the model would buy as much oil as possible in order to avoid penalties for lost demand. And that would mean buying from the cheapest refinery before re-distributing it all over Nepal. This would take place as long as the extra transportation and inventory costs did not outweigh the saved penalties, which they would not in the base case. To avoid this, which certainly does not describe reality, we use prices P_k which do not vary over refineries, and instead include the differences between depot prices as a part of the transportation cost C^n_{jk}. This eliminates the problem.

Finally, constraints (8) guarantee that the depots will not exceed their holding capacities, constraints (9) insure non-negative decision variables and constraints (10) limit the size of the rejections.

6 Stochastic LP formulation

The model we develop in this section is a stochastic multi-period model representing the steady-state of the operation. We use scenarios as outlined in Figure 3, which is a specific case of the more generic Figure 2. Note that because of the looping-back, it is technically not a scenario tree. We will, however, still use some of the scenario-tree terminology and call the nodes in the period before the converging nodes *leaves*, even if technically they are not.

At this point we might want to properly define our first-stage variables. Focus then on node 260: whatever is the inventory level in node 260 is also enforced during the quiet period (nodes 261 through 269, as well as in the no-disturbance scenario: nodes 0 and 114 to 133). Further, the same inventory is enforced in all leaf nodes. The idea is that after an event is over, the scenario is given some time to recover from the event, bringing inventories back to "normal" (i.e. that of node 260).

However, a problem occurs here as a result of the chosen modelling approach. Within a scenario, but after the event is over, the model "knows" there will be no disturbances. Hence, it brings inventory down to a minimal level for a while, and then builds it up in time for its leaf node. This does not reflect reality. Hence, we have added another set of nodes that must have the same inventory levels as node 260. These are found, to some extent by trial and failure, by making sure that each scenario has enough time (but not more than that) to rebuild inventory. Our first stage variables are the inventories in all nodes where inventory is forced to be equal to that of node 260.

Event durations and starts of the fixed-inventory periods for each scenario are given in Table 1, both in terms of time and node numbers. From the table we can, for example, read that the event in the third scenario starts in node 41 in period 1 and ends in node 44

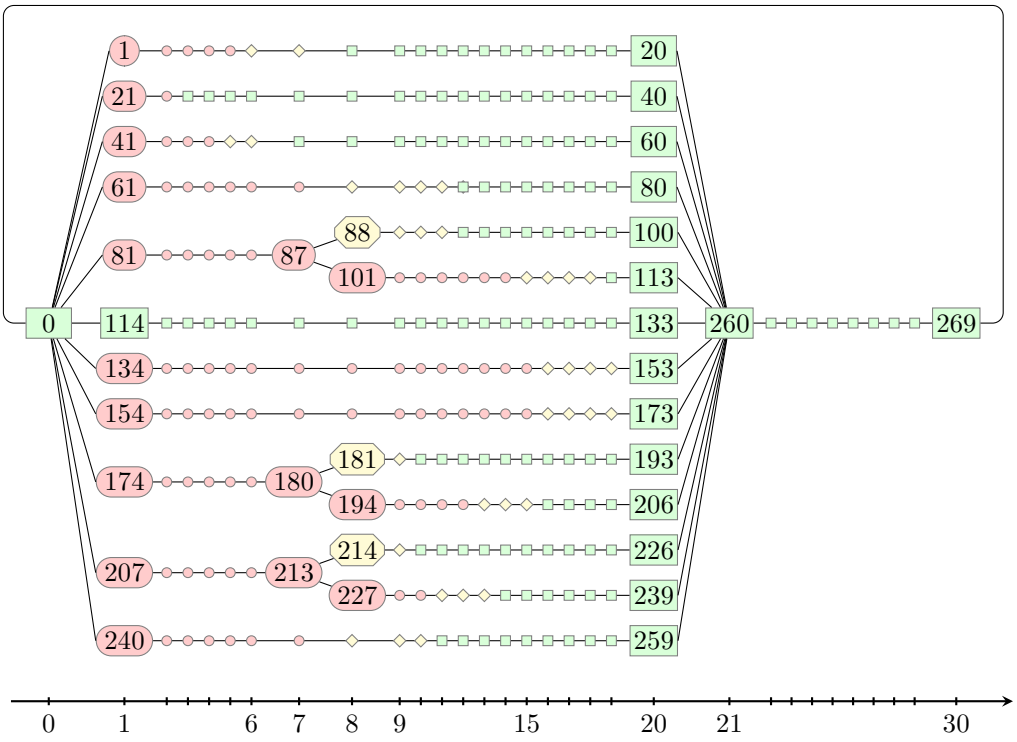

Figure 3: Scenario tree used in the numerical tests. Nodes without displayed numbers are numbered consecutively from the previous given number. Circles/ellipses represent nodes with an ongoing event, octagons and diamonds the recovery nodes and finally rectangles represent the normal nodes, i.e. nodes where the inventory is fixed to the steady-state level.

in period 4. After that, the scenario is given 3 periods to recover from the event, before it is forced to have the steady-state inventory from node 47 at time 7. If we combine information from the table with Figure 3, we can see that scenarios 10 and 11 start as one, in node 174. The difference is that in scenario 10, the event ends in node 180 in period 7, while in scenario 11 it continues up to period 12. This means that the two scenarios differ from period 8 onwards, as we can see in Figure 3.

At this point, we are ready to present the stochastic model. Since most of the notation is the same as in the deterministic case, we only present the extra things needed for the stochastic case:

Table 1: Information about the scenario events. For each scenario, the table shows information about the start and end of the event, plus the start of the enforced normal (steady-state) state. The information is presented in terms of time and nodes—node that we do not show the time of the start of the event, as they all start in the first period ($t = 1$).

Scen.		Time		Nodes		
No.	Code	Ev. End	Nor. Start	Ev. Start	Ev. End	Nor. Start
1	R2	5	8	1	5	8
2	R4	2	3	21	22	23
3	R3	4	7	41	44	47
4	R1	7	12	61	67	72
5	R5a	7	12	81	87	92
6	R5b	14	19	81	107	112
7	N	.	0	114	.	0
8	R6	15	20	134	148	153
9	S1	15	20	154	168	173
10	P1a	7	10	174	180	183
11	P1b	12	16	174	198	202
12	P2a	7	10	207	213	216
13	P2b	10	14	207	229	233
14	P3	7	11	240	246	250

- τ Root node, i.e. the node just before the start of the event. This is node 0 in Figure 3.
- α Converging node, i.e. the node where all scenarios converge. This is node 260 in Figure 3.
- $\mathscr{L}$ Set of leaf nodes, i.e. the nodes just before the converging node; $\mathscr{L} \subset \mathscr{N}$. In our case, these are the nodes at $t = 20$.
- $\mathscr{C}$ Set of control nodes, i.e. the nodes where inventories must be the same as in node α; $\mathscr{C} \subset \mathscr{N}$. Note that these steady-state inventories are our first stage variables, as just outlined.
- p^n Probability of node n.
- $\mathscr{S}$ Set of scenarios, i.e. paths from the root to the final node (node 269). The probability of a scenario is given by the probability of its leaf node.
- $\mathscr{N}_s$ Set of nodes belonging to scenario $s \in \mathscr{S}$. Note that a node can belong to more than one scenario; in particular, $\{\tau, \alpha\} \subset \mathscr{N}_s$ for all $s \in \mathscr{S}$.

The objective is then to solve

$$\min \sum_n p^n \sum_k \Big(\sum_j C^n_{jk} x^n_{jk} + \sum_j H_k z^n_{jk} + \sum_{ij} c^n_{ij} w^n_{ijk} + \sum_{l=1}^{3} N_{jk,l} \sum_j r^n_{jk,l} \Big) \tag{11}$$

Subject to

$$\mathbf{1}_{\mathscr{D}_1}(j)\,x^n_{jk}+\mathbf{1}_{\mathscr{D}_2}(j)\,x^{\mathrm{pa}(n)}_{jk}+\sum_{i\in\mathscr{D}} w^n_{ijk}+z^{\mathrm{pa}(n)}_{jk}=q^n_{jk}+\sum_{i\in\mathscr{D}} w^n_{jik}+z^n_{jk}\quad \forall j,k,\ \forall n\in\mathscr{N}\setminus\{\alpha\} \tag{12}$$

$$\mathbf{1}_{\mathscr{D}_1}(j)\,x^\alpha_{jk}+\mathbf{1}_{\mathscr{D}_2}(j)\,x^n_{jk}+\sum_{i\in\mathscr{D}} w^\alpha_{ijk}+z^n_{jk}=q^\alpha_{jk}+\sum_{i\in\mathscr{D}} w^\alpha_{jik}+z^\alpha_{jk}\quad \forall j,k,\ \forall n\in\mathscr{L} \tag{12c}$$

$$\sum_{j\in\mathscr{D}}\sum_k\Big(x^n_{jk}+\sum_i w^n_{ijk}\Big)+\sum_{j\in\mathscr{D}_2}\sum_k x^{\mathrm{pa}(n)}_{jk}\le U\quad \forall n\in\mathscr{N}\setminus\{\alpha\} \tag{13}$$

$$\sum_{j\in\mathscr{D}}\sum_k\Big(x^\alpha_{jk}+\sum_i w^\alpha_{ijk}\Big)+\sum_{j\in\mathscr{D}_2}\sum_k x^n_{jk}\le U\quad \forall n\in\mathscr{L} \tag{13c}$$

$$q^n_{jk}+\sum_i w^n_{jik}\le z^{\mathrm{pa}(n)}_{jk}\quad \forall j,k,\ \forall n\in\mathscr{N}\setminus\{\alpha\} \tag{14}$$

$$q^\alpha_{jk}+\sum_i w^\alpha_{jik}\le z^n_{jk}\quad \forall j,k,\ \forall n\in\mathscr{L} \tag{14c}$$

$$q^n_{jk}+\sum_{l=1}^{3} r^n_{jk,l}=d^{\mathrm{t}(n)}_{jk}\quad \forall n,j,k \tag{15}$$

$$\frac{q^n_{jk}}{d^{\mathrm{t}(n)}_{jk}}\ge g^n_{jk}\frac{\sum_{m\in\mathscr{N}} p^m\sum_{i\in\mathscr{D}} q^m_{ik}}{\sum_{m\in\mathscr{N}} p^m\sum_{i\in\mathscr{D}} d^{\mathrm{t}(m)}_{ik}}\quad \forall j,k,\ \forall n: d^{\mathrm{t}(n)}_{jk}>0 \tag{16}$$

$$\sum_{j,k} P_k\sum_{n\in\mathscr{N}_s} x^n_{jk}=B\quad \forall s\in\mathscr{S} \tag{17}$$

$$z^n_{jk}\le M_{jk}\quad \forall n,j,k \tag{18}$$

$$z^n_{jk}=z^\alpha_{jk}\quad \forall j,k,\ \forall n\in\mathscr{C} \tag{19}$$

$$x^\tau_{jk}=\frac{\sum_{n\in\mathscr{C}} p^n x^n_{jk}}{\sum_{n\in\mathscr{C}} p^n}\quad \forall j\in\mathscr{D}_2,\ \forall k \tag{20}$$

$$x^n_{jk},w^n_{ijk},q^n_{jk},z^n_{jk}\ge 0\quad \forall n,i,j,k \tag{21}$$

$$0\le r^n_{jk,l}\le b_l\,d^{\mathrm{t}(n)}_{jk}\quad \forall n,j,k,\ l\in\mathscr{I} \tag{22}$$

Here the objective function (11) is the financial representation of the operational activities; it is the same as in the deterministic model except for the obvious addition of probabilities. Constraints (15), (16), (18), (21) and (22) are exactly the same as their deterministic counterparts (5), (6), (8), (9) and (10).

For constraints that point one period back, we have to treat the converging node separately, since it does not have one given parent, but a whole set of parent nodes—the set of leaves. In the model, these pairs of constraints share the same equation number, with the converging-node variant having 'c' added to the number. Apart from this, these constraints remain unchanged from the deterministic case. In particular, constraints (12)–(14) are the same as respectively (2)–(4).

The budget constraint (17) differs from its deterministic counterpart (7) in the sense that we require it to hold for every scenario.

The first constraints without any deterministic counterpart are constraints (19), which take care of our first stage variables by forcing inventory in the *control nodes* to be equal to the steady-state inventory. The set of control nodes includes all the nodes in the quiet periods, the converging node, the leaf nodes, and those nodes where inventory would otherwise dip to minimal levels as explained earlier.

Finally, constraints (20) make sure that purchases in the root node for depots which are two days away from their refineries equal the average of the purchase in the steady-state nodes of the model. This is included to make sure that extra-ordinary purchases are not made on the day preceding the event. Such purchases would contradict the logic of the model.

6.1 Random events

In this section, we discuss the different possible random events that may occur and then present the scenarios used in our numerical tests. Note that the scenario names are the same as in Table 1.

Among many possible uncertainties in the supply chain of NOC, we discussed a few which occur with a reasonable frequency. Landslides often block road links between depots. Another common event is strikes called by truck owners or drivers. This can be isolated to a region or affect the whole system. Accidents may also block parts of the network. Political disturbances have been a major reason for blocked road in recent years. Again this can be local or affect the whole network. Breakdown of refineries is also a source of uncertainties that NOC faces. We consider a few scenarios of these random events for our analysis.

6.1.1 Scenarios due to road problems.

Scenario R1. The road link between Kathmandu and Amlekhganj breaks down due to landslide for a week. Here the inter-depot transport to Kathmandu is affected from Amlekhganj and Biratnagar. These are now routed through Birgunj. Also because of this breakdown the supply to Kathmandu from Raxaul is via Birgunj.

Scenario R2. The road to Kathmandu closes down due to an accident for 5 day and makes Kathmandu isolated from the rest of the country. In this scenario supply to and all inter-depot movement from and to Kathmandu are stopped.

Scenario R3. The road link between Kathmandu and Amlekhganj breaks down due to strike of truck drivers for 3 days. In this scenario the inter-depot link from Amlekhganj and Biratnagar is affected and now will be routed through Birgunj. Also the supply to Kathmandu from Raxaul will be through Birgunj.

Scenario R4. The road link between Raxaul refinery and Amlekhganj depot breaks down due to accident for 2 days. Here inter-depot movement is not affected and only the supply to Amlekhganj is routed through Birgunj. In doing so the supply to reach Amlekhganj takes two days instead of one day. To be able to model this, we have to

let the sets $\mathscr{D}_1, \mathscr{D}_2$ depend on node n and change the appropriate part of constraints (12) and (12c) to

$$\mathbf{1}_{\mathscr{D}_1^n}(j)\, x_{jk}^n + \mathbf{1}_{\mathscr{D}_2^{\mathrm{pa}(n)}}(j)\, x_{jk}^{\mathrm{pa}(n)}\,.$$

Note the $\mathrm{pa}(n)$ index on the $\mathscr{D}_2$ set. This way, we can properly model the fact that there is no delivery on the first day and two deliveries on the third day: the delayed delivery from the second day of the break-down, and the one-day delivery ordered on the third day.

Scenario R5. The road link between Biratnagar depot and Barauni refinery breaks down due to flood for one or two weeks with equal probability. Since the Barauni refinery supplies all products except ATF to Biratnagar, the new source for Biratnagar for those products will be Raxaul, as this is the nearest refinery available.

Scenario R6. The link between Allahabad refinery and Bhairahawa depot breaks down due to damage of a bridge for 15 days. Since Allahabad is the source for ATF (air fuel), the breakdown affects the flow of ATF to Bhairahawa. Here the nearest refinery or depot of IOC after Allahabad will be Raxaul and the route will be via Birgunj.

6.1.2 Scenario due to refinery breakdown.

Scenario S1. Barauni refinery is down for 15 days. As Barauni is down, the next nearest source point for Biratnagar is Raxaul.

6.1.3 Scenarios due to political disturbance.

Scenario P1. The link to the Kathmandu depot from Amlekhganj is down for one week with a probability of 60 percent or down for 12 days with probability of 40 percent.

Scenario P2. The link to the Kathmandu depot from Amlekhganj is down for one week or for 10 days with probability of 60 and 40 percent, respectively.

Scenarios P1 and P2 are structurally the same (only durations vary). This reflects the importance of this link in connection with political disturbances.

Scenario P3. All the links to Biratnagar, Kathmandu and Pokhara depots are down for one week. In this scenario these depots are not reachable from any other depots or refineries.

Here we can see that when the scenarios R1, R2, R3, R4, R6, S1, and P3 occur we know their durations and hence these scenarios can be represented in a fan structure. But scenarios R5, P1 and P2 have random durations, and hence can be represented in a tree structure. In total, we end up with the scenario structure presented in Figure 3.

7 Computational Results

The model has been implemented using AMPL modelling language , and solved using CPLEX. 9.0.0 on a 3 GHz PC with 1 GB of RAM. Solution times were mostly less than 2 minutes (the base case takes 10 seconds), with one case using 15 minutes, all with cold starts, while warm starts would mostly take just a couple of seconds.

The cases we are about to present are realistic. However, our goal is not to provide specific advice in a specific case, but rather to illustrate how optimal inventories depend on different model parameters. We have used real data, collected on the ground in Nepal, for transportation costs, inventory costs (mostly capital costs), truck capacities, network structure, and demand. Our base case for the budget covers about 78% of demand (which is reasonable), and we have used all fairness factor g^n_{jk} the same and equal to 0.7.

For penalties, we have used three intervals with $b = \{0.12, 0.20, 0.68\}$. In other words, the break points are at 12% and 32% of unsatisfied demand, which corresponds to the average level of unsatisfied demand (100% - 78% = 22%) plus/minus 10%. Base case penalties are such that penalties do not drive the solution, but enough to make sure non-deliveries are evenly distributed (the non-deliveries which are not governed by the fairness). This is the hardest parameter to set. We observe, though, that within reasonable (and large) intervals, the sensitivity to penalties is low for the other parameters at their base values.

Let us start by showing how total inventory develops over time (averaged over the events) for our base case. It is shown in Figure 4. We have distinguished between inventory (to be named *technical inventory*) that would be there even if there were no events at all (caused by (4)), and what is added due to the events (hereafter called *event inventory*), i.e. the inventory of interest for this paper. As can be seen, inventory is kept in expectation of events, then when events start, inventory drops, and is then rebuilt. Although not shown in the figure, for each inventory, there is at least one event (scenario) for which the event inventory goes to zero. If that had not been the case, we would have been keeping inventory (at a cost) that was never needed, and that could not be optimal.

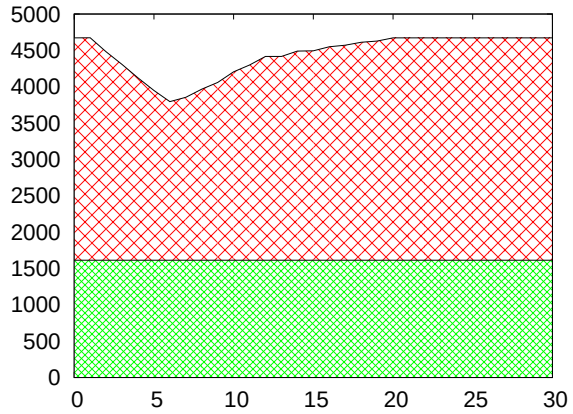

Figure 4: Optimal inventory over time for the base case. The bottom part shows technical inventory caused by (4) in the deterministic case, the top part event inventory that we keep for the random events.

7.1 Budget

Inventory is kept in anticipation of events and deliveries. If the budget is increased, we can deliver more, and this will of course increase technical inventory as deliveries have to come from inbound inventory. But also the event inventory will increase. This is simply because the amounts to be handled during events have also increased. There are two driving factors here, the penalties and the fairness. If deliveries in some important depot drops substantially during an event, the piece-wise linearity of the penalties will force inventory to prevent that from happening. But g also have some interesting effects. Some of these effects can be found in Figure 5. In the left part $g = 0.25$, in the right $g = 0.95$.

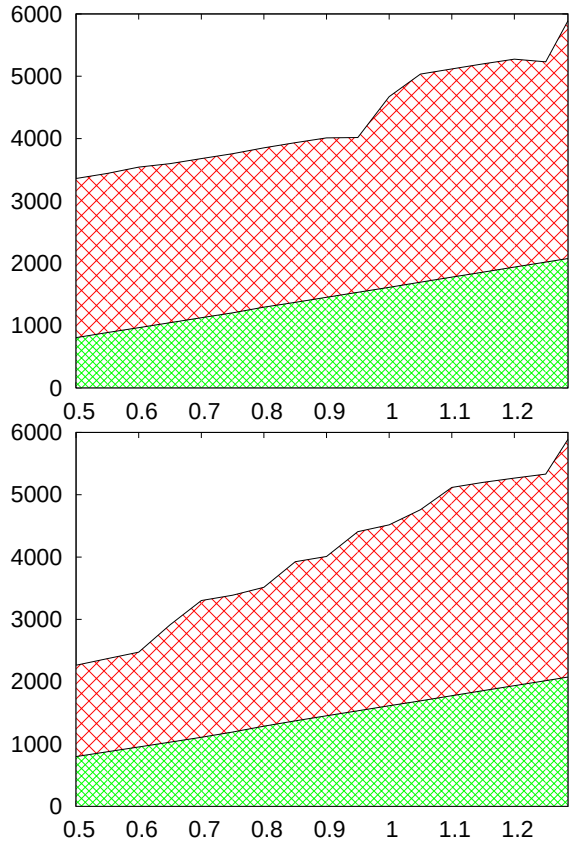

Figure 5: Steady-state inventory as function of budget for low and high g. Budget is measured relative to our base case (corresponding to 1 on the horizontal axis). The bottom part shows technical inventory caused by (4) in the deterministic case, the top part event inventory that we keep for the random events.

In the figure, we can see that for all but rather high budgets a low g yields higher inventories than a high g. How can that happen? With a low g, deliveries are driven mainly by costs and penalties, so some depots can be treated very badly. The effect is that we supply mainly Kathmandu, as it has large demands and the highest penalties (except for the first interval, where the penalties $N_{jk,l}$ are the same for a given product k over all depots j).

This reflects that Kathmandu has more critical infrastructure than the rest of the country. As g increases, we are forced to supply also other depots, so the sales in Kathmandu

go down. Now, because of its location and its status as the capital of Nepal, Kathmandu is exposed both to the physical and the political disturbances—most of the scenarios harm Kathmandu and/or its neighbours. As a result, Kathmandu needs the largest event inventory relative to sales. Hence, as g forces more and more sales out of Kathmandu, the overall inventory goes down.

These results are stable over penalty levels. The reason is that in all runs Kathmandu has higher penalties than the other depots, whether the penalties are high or low, so the qualitative arguments above hold. The objective function is of course dependent on the level of the penalties, but the solutions are not. This was the intended effect of penalties.

The effects of g are reasonable. If fairness is given low importance, the pressures from the major depots, caused by their importance, will guide the deliveries. Fairness is therefore needed to reflect that NOC is not mainly there to make a profit, but to efficiently distribute oil in a socially acceptable way.

7.2 Political unrest

The relationship between the level of political unrest and steady-state inventory illustrates how the model prepares for the random events. This is illustrated in Figure 6.

We ran the model by increasing the probabilities for the political disturbances proportionally and reducing equally that of the non-event scenario, keeping the probability of all other types of risks fixed. We expected inventory levels to be monotonously increasing with the level of political unrest. This is also what we observe, but the effect is very moderate. Why is the curve so flat for all but the lowest probabilities? The reason is that most of the event inventory is driven by fairness (which acts as a constraint) and penalties. The fairness aspect does not depend on probabilities: the nodes are treated equally irrespective of their probabilities (except when zero). Penalties are of course more serious when probabilities are high, but the effect is marginal as the relative sizes of penalties are not changed, only the level. The higher is g, the more this effect is true: as soon as events exist and must be protected against, we act even if probabilities are low.

We are careful about using precise numbers here, as we are generally looking for qualitative understanding. But even so, it is worth noting that already at 0.25% probability of political unrest, the event inventory increases by 58% (from 1477 to 2336), compared to the case without any political risk. And for 16% risk, the event inventory is up 107% (from 1477 to 3060).

Conclusion

We have formulated, solved, and analyzed the problem of distribution and inventory management in an infinite horizon problem with random disturbances in the flow network. A unique modelling approach is used to find the optimal inventory positions. It is based on looping the network back on itself and changing the time line so as to create a three stage model out of what is, in reality, an infinite horizon problem.

The present approach is unique in the sense that it can incorporate any type and intensity of disturbances in the flow network and can show their affect on the steady-state inventory positions.

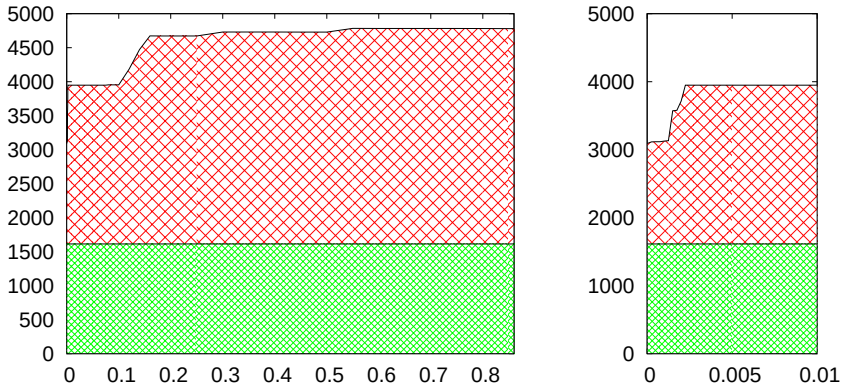

Figure 6: Steady-state inventory as a function of the probability of political unrest. The bottom part shows technical inventory caused by (4) in the deterministic case, the top part event inventory that we keep for the random events. The smaller figure to the right presents a detail view of probabilities close to zero. We can see that already at 0.25% probability, the inventory reaches a level that is sufficient for probabilities up to 10%.

Our numerical test results from Nepal show that management can get useful insights into the steady-state inventories to be maintained during normal periods in anticipation of random events. Also this approach may be used for analyzing the effects of any new plans, like adding a pipeline or a new road link, to the existing flow network.

Acknowledgements

This project is supported by The Research Council of Norway under grant 171007/V30.

References

Govt. to lend rs. 1.70b to noc, jul 2007. URL `http://www.ekantipur.com`.

T. Davis. Effective supply chain management. *Sloan Management Review*, 34(4):35–46, 1993.

R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, second edition, 2002.

C. Harland, H. Brencheley, and H. Walker. Risk in supply network. *Journal of Purchasing and Supply Management*, 9(2):51–62, 2003.

A.-G. Lium, T. G. Crainic, and S. W. Wallace. Correlations in stochastic programming: A case from stochastic service network design. *Asia-Pacific Journal of Operational Research*, 24(2):161–179, 2007.

J. G. March and Z. Shapira. Managerial perspective on risk and risk taking. *Management Science*, 33(11):1404–1423, 1987.

L. Meulbrook. Total strategies for company-wide risk control. Financial Times, May 9 2000.

G. A. Zsidisin. A grounded definition of supply risk. *Journal of Purchasing & Supply Management*, 9(5–6):217–224, 2003.

G. A. Zsidisin, A. Panelli, and R. Upton. Purchasing organization involvement in risk assessments, contingency plans, and risk management: an exploratory study. *Supply Chain Management*, 5(4), 2000.

Paper 2

Single Source Single-Commodity Stochastic Network Design

Single Source Single-Commodity Stochastic Network Design

Biju Kumar Thapalia* Michal Kaut† Stein W. Wallace‡

Teodor Gabriel Crainic§

30 September 2009

Abstract

Stochastics affects the optimal design of a network. This paper examines the single-source single-commodity stochastic network design problem. We characterize the optimal designs under demand uncertainty and compare with the deterministic counterparts to outline the basic structural differences. We do this partly as a basis for developing better algorithms than are available today, partly to simply understand what constitutes robust network designs.

Keywords: Single-commodity network design, Stochastic, Correlation, Robustness

1 Introduction

There are many real-life problems that can be described as network flow problems and for most (if not all) of them there is an underlying design problem. The purpose of this paper is to study the relationship between the stochastic and the deterministic single commodity network design problem, all the time under the assumption of a single source.

The traditional approach to network design is to formulate deterministic models. The demand is usually set to its expected value or sometimes some other, somewhat higher, value, to cater for "normal variation". In almost all cases, it is understood that the demand is actually stochastic, but the handling of stochasticity is deferred to the operational planning level. The reasons for doing so can be many: computational complexity even of the deterministic network design model; a view that modeling wise, we know too little about demand while still being at the network design level of the planning; or simply that it is appropriate to postpone such details of the plan. After all, the goal is to set up the network, not decide how to route the flow.

*Molde University College, Norway – biju.k.thapalia@himolde.no

†Norwegian University of Science and Technology (NTNU), Trondheim, Norway

‡Lancaster University Management School, England

§Université du Québec à Montréal (UQAM), Canada

The question we ask here is: For the single-source single-commodity stochastic network design (SSSND) problem, how much do we lose by not taking stochasticity in demand into account already at the design level? Could it be that the design coming from a model which is explicitly told that the future demand is uncertain is substantially better than a design not based on this knowledge? *Given* the distributional information used in the stochastic formulation, the design coming from the stochastic model will by definition be better (measured by the objective function) than the design from any deterministic model. Of course, if the distributional information is substantially incorrect, a deterministic design might (by chance) behave better in the real world. That, however, is not the focus of this paper. Rather, what we are interested in is, given distributional information, how much better is the stochastic design, and even more importantly: In what way does the stochastic design differ from its deterministic counterpart, that is, *what* is it that makes one design better than the other? We know it is related to investment in flexibility, see Wallace (2009) for a discussion in the framework of option theory, but we would like to know rather precisely what this investment in flexibility consists of. And conversely, we are also interested to see if some structures from the deterministic design actually carry over to the stochastic counterpart.

We thus study the structural difference between the deterministic and stochastic formulation to better understand the phenomenon of investing in flexibility. We also hope to use the results to develop algorithms to solve the problem approximately (for large cases) or potentially to optimality (for moderate cases).

Our work is related to that of Lium, Crainic, and Wallace (2009). They study the multi-commodity problem (and hence have several sources and several sinks for the flow). They identify two major structural differences: In the stochastic solution it is valuable to have several paths for each commodity and each of these paths should be shared with other commodities. Sharing is particularly useful in the case of negative correlations between demands. Without enforcing consolidation, their networks end up as consolidation networks, often hub-and-spoke. Contrary to conventional deterministic design, consolidation is a hedging device, not a volume related undertaking. Hence, they identify structures that can be seen as investments in flexibility, that is, options, along what is discussed in Wallace (2009). Deterministic models would not produce such results.

We are studying the single commodity case. And we are limiting ourselves to a single supply node (or alternatively a single demand node). We chose to look at the single supply node case in order to have a simpler (structurally speaking) problem, so that it is easier to see what structures emerge in the solutions.

From a linear programming perspective, single commodity flow problems are simpler to solve than multi-commodity flow problems since classical network flow algorithms can be applied directly, see for example Ahuja, Magnanti, and Orlin (1993). However, this simplicity of the single commodity case in terms of flow problems does not carry over to the problem we are studying: The structure of the designs. In fact, we believe that single commodity design is structurally more complicated to *interpret* than the multi-commodity counterpart. In the multi-commodity case, the commodities share edge capacities, while in the single commodity case with a single source node, the different demand nodes (which is the closest we can get to something that corresponds to a commodity in the multi-commodity case), certainly share edge capacities, but also experience cancel-

lation of flow. That is, if two commodities in the multi-commodity case need to use the same edge, but in opposite directions, we must cater explicitly for both, while if the same occurs for two demand nodes in the single-commodity case, flow cancellation occurs and we must cater only for the difference. It is our experience that this cancellation of flow increases the complexity of interpretation, and this is why we in this paper start with the simpler single source case.

The use of network optimization occurs in many different fields. Production-distribution systems, economic planning, energy systems, communication systems, material handling systems, water distribution, traffic systems, railway systems, evacuation systems, and many others use network optimization models. Aronson (1989) surveys applications of network design problems in different fields. Most existing works are focused on the multi-commodity case, whereas much less attention is given to the single-commodity setting. This is, at least partly, caused by the assumption that single-commodity network problems are not very rich in applications. However, when we look into the single-commodity network design problems, we see that this is not totally true. The single-commodity network design problem is encountered in various applications, like the design of water distribution system (Sherali and Smith, 1997), oil pipeline design (Hochbaum and Segev, 1989, Rothfarb, Frank, Rosenbaum, Steiglitz, and Kleitman, 1970), sewer network design (Liang, Thompson, and Young, 2004), one-terminal telpak problems (Rothfarb and Goldstein, 1971), local access design problems in telecommunications networks (Hochbaum and Segev, 1989), and feeder-bus network design problems (Kuah and Perl, 1989, Kuan, Ong, and Ng, 2006), to name a few. The richness of the problem class also increases with the transformation of certain multi-commodity network problems to a single-commodity setting as discussed in Evans (1978).

The remainder of the paper is organized as follows. Section 2 explains the problem in detail with its mathematical formulation. Section 3 explains the set-up of our experiments and lists the computational results with discussions. Section 4 concludes the paper.

2 Problem description

Given a set of potential undirected edges connecting a set of nodes, one of which is the supply node and the rest are demand and transshipment nodes, determine which edges to open (including their capacities), such that the edges can carry flow from the source node to fulfill the demand at the demand nodes. The design is based on minimizing the sum of the fixed costs of selecting edges connecting the nodes, linear costs to open capacities on the edges, per unit costs of flows on the edges, and per unit penalty costs for not satisfying demand. Lack of satisfaction of demand could amount to using another transportation mode, using the same mode, but delayed, using a competitor, or simply rejecting the demand.

It is important to include the possibility of rejecting flow in the model. The main reason is that reality dictates, except in extremely particular situations, that it is prohibitively costly to build a network that can meet any possible demand—however unlikely it might be. Deterministic models, operating on expected demand, may reasonably operate under the assumption that (average) demand *must* be met. But even there, there will normally be an understanding that some demand may end up being turned down in reality. When

working with stochastic demand, there is also the problem that requiring demand to be met turns the model into a worst-case model, where the worst-case in most cases is not even well understood. So, in total, we find it crucial to include the possibility of not satisfying all the demand. We use the same formulation also in the deterministic models, to make the results comparable.

The stochastics in the problem arises in the form of demand uncertainties. It is rare that demand is fully known when the design is determined, be it a distribution network or a pipeline network.

This problem is formulated as a two-stage stochastic programing model where the first-stage decisions are which edges to open, and which capacities to install. The second-stage decisions are the flow decisions in the given network. The recourse actions, which are performed in the second stage, are described by a penalty cost incurred for unsatisfied demand.

In the deterministic case, the demand in each node is fixed at the mean demand for the stochastic case. We do not discuss edge failures here, but leave that for a later paper.

2.1 Mathematical formulation

Let $G = (\mathscr{N}, E)$ be a network defined by a set $\mathscr{N}$ of n nodes, where one of them is a source node and rest are demand nodes and transshipment nodes, and a set E of m undirected edges, where

$$E \subset \{k = (i, j) : i \in N, j \in N \text{ and } i < j\}.$$

Each edge is indexed either by i, j or by k. We assume that supply equals demand in all scenarios. The notation for the sets, parameters, and variables associated with this problem is as follows:

Sets:

$\mathscr{D}$	set of all nodes with non-zero demand;
$\mathscr{T}$	set of all nodes with zero demand (transshipment nodes);
$\mathscr{C}$	singleton set containing the supply node so $\mathscr{N} = \mathscr{D} \bigcup \mathscr{T} \bigcup \mathscr{C}$;
$\mathscr{S}$	set of all scenarios s .

Variables:

$x_k^s = x_{ij}^s$	flow on edge $k = (i, j) \in E$ going in direction $i \to j$, in scenario $s \in \mathscr{S}$;
$z_k^s = z_{ij}^s$	flow on edge $k = (i, j) \in E$ going in direction $j \to i$, in scenario $s \in \mathscr{S}$;
u_k	new capacity that is developed on edge $k \in E$;
e_i^s	for $i \in \mathscr{D}$, this is the unsatisfied/lost demand in node i in scenario $s \in \mathscr{S}$;
	for $i \in \mathscr{C}$, this is the unused capacity of source node i in scenario $s \in \mathscr{S}$;
y_k	1 if edge $k \in E$ is developed, 0 otherwise.

Parameters:

M	maximal arc capacity; used for linking capacities and open arcs in (4);
R	unit cost of unsatisfied demand;
p^s	probability of scenario $s \in \mathscr{S}$;

c_k flow cost on edge $k \in E$;
g_k fixed setup cost for edge $k \in E$;
h_k variable setup cost; the cost for adding one unit of capacity to edge $k \in E$;
v_k initial/existing capacity on edge $k \in E$;
d_i^s demand ($d_i^s < 0$) or supply ($d_i^s > 0$) in node $i \in \mathscr{N}$ in scenario $s \in \mathscr{S}$.

Our overall problem is hence:

$$\min \sum_k g_k y_k + \sum_k h_k u_k + \sum_s p^s \left\{ \sum_k c_k \left(x_k^s + z_k^s\right) + R \sum_{i \in \mathscr{D}} e_i^s \right\} \tag{1}$$

Subject to:

$$\sum_{j:(ij)\in E} \left(x_{ij}^s - z_{ij}^s\right) - \sum_{j:(ji)\in E} \left(x_{ji}^s - z_{ji}^s\right) = \begin{cases} 0 & \forall i \in \mathscr{T}, \forall s \in \mathscr{S} \\ d_i^s - e_i^s & i \in \mathscr{C}, \forall s \in \mathscr{S} \\ d_i^s + e_i^s & \forall i \in \mathscr{D}, \forall s \in \mathscr{S} \end{cases} \tag{2}$$

$$x_k^s + z_k^s \leq u_k + v_k \quad \forall k \in E \quad \forall s \in \mathscr{S} \tag{3}$$

$$u_k \leq M y_k \qquad \forall k \tag{4}$$

$$d_i^s \geq e_i^s \geq 0 \qquad \forall i \in \mathscr{D}; \forall s \tag{5}$$

$$x_k^s, z_k^s, u_k \geq 0 \text{ and } y_k \in \{0,1\} \qquad \forall k; \forall i; \forall s \tag{6}$$

The objective function (1) minimizes the total expected cost of the network. The first part is the costs of constructing the new edges, the second part, the costs of building the new capacities, the third part, the flow costs through all the edges, and the fourth part is the penalty costs of not fulfilling demand.

Constraints (2) model conservation of flow at nodes. The left-hand side is the net outflow from node i, which must be zero for all the transshipment nodes $i \in \mathscr{T}$ and is equal to the unused capacity for the single source node $i \in \mathscr{C}$. For the demand nodes, the net outflow must be equal to the satisfied demand; since d_i^s is negative in this case, the right-hand side is the difference between the scenario demand d_i^s and the (positive) unsatisfied demand e_i^s.

Notice that in an optimal solution, we will never have flow in both directions of an edge, consequently Constraints (3) represent the flow limit on each edge. The left hand side of the equation is the net flow on edge k which should be less then or equal to the total capacity of the edge. Constraints (4) show that new capacity u_k can be developed only if edge k is built. Constraints (5) shows the bound for the rejection and, finally, (6) ensure that all variables are non-negative and the edge construction variables are binary.

We model the problem in AMPL and solve it to optimality using CPLEX 9.0. The solution times varied from few seconds to 18 hours depending on the case, on an Intel® Core™ Duo running at 2.2 GHz with 3.5 GB of RAM.

3 Experimentation and Computational Results

In this section, we begin by describing how we generate our random test instances, and then present our computational results.

3.1 Test instance generation

We used six different type of network instances taken from two different libraries. The first four instances, namely Atlanta, France, Nobel-EU, and Pdh are telecommunication examples from the SNDlib[1] library, with some modification to suit our problem's needs. The fifth instance was generated by us and named Molde and the sixth (Montreal) was obtained from CIRRELT[2], Montreal. The names of the instances as such do not mean anything particular in this computational setup.

The Montreal test instance does not have node coordinates, so we used Graphviz[3] to draw the graph using fixed setup cost as distance measure. This resulted it in a non-planar graph. The graphs of the test instances Atlanta, Nobel-EU and Molde are all planar. From each of them, we created non-planar instances by randomly adding a few extra edges. This gave us a total of nine problems. For each of the nine problems, we picked 3 nodes (2 in the cases of Nobel-EU_nonplannar and Pdh) as possible source nodes, thus creating in total 25 base test instances. These 25 different versions of the test cases are presented in Table 1. Given the difficulty of solving the stochastic network design problem to optimality, we kept n (the number of nodes) below 20 and m (the number of edges) below 40.

Table 1: The list of different versions of the test cases used for computation.

Problem name	alt. sources	# nodes	# edges	# demand nodes
Atlanta	1, 2, 11	15	22	12
Atlanta_nonplanar	1, 2, 11	15	29	12
France	12, 13, 16	16	29	13
Nobel-EU	1, 9, 12	19	28	16
Nobel-EU_nonplanar	1, 9	19	32	16
Pdh	1, 2	11	30	9
Molde	2, 9, 14	15	30	12
Molde_nonplanar	2, 9, 14	15	38	12
Montreal	5, 8, 10	10	29	7

Out of the three potential supply nodes (or two nodes for some problems), when one of them is the source node, the other two (or one) are transshipment nodes. The possible source nodes are listed in the second column of Table 1. We know from the work of

[1] SNDlib is a library of test instances for **S**urvivable fixed telecommunication **N**etwork **De**sign, available from `http://sndlib.zib.de/`.

[2] CIRRELT is an *Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation* in Montréal, Canada, see `http://cirrelt.ca`.

[3] Graphviz is an open-source graph-visualization tool, see `http://graphviz.org`.

Lium, Crainic, and Wallace (2007) that correlations are important in shaping the structure of the network. Hence, we further create 3 cases from each problem instance: one with uncorrelated demands, one with positively correlated demands and one with mixed correlated demands. In positively correlated demand case all correlations are set to 0.7 and in mixed correlated demand case: the demand nodes are divided into two groups such that each group contains about half of the nodes. All correlations *within* a group are set to 0.7, while *between* groups we use -0.7. Thus we have in total 75 test cases. It is worth noting that all the cases from the SNDlib and Montreal test instances are multi-commodity network design problems, so not all parameters can be used directly by us. We only kept the coordinates (where available) for the nodes and the fixed setup cost g_k for the edges. The values for the other parameters – variable setup costs h_k and flow costs c_k – are all chosen proportional to the Euclidean distance between the node pairs. The cost of unfulfilled demand R is selected with trial and error until we felt that it did not drive the solution in an unreasonable way. We chose R so that we do not get more than 5% of total demand rejected. Mostly we saw rejections around 2–3 percent. The results in the first part of Section 3.3 are based on test cases with these cost structures. In the second part of the section, we changed the fixed costs (as discussed in the Section 3.3) to understand their relative importance.

In the absence of reference to a particular distribution representing the random demand we chose to use normal distributions with mean equal to the deterministic demand (from the underlying cases) and standard deviation equal to 25% of the mean to represent its stochasticity. As stochastic programs need discrete distributions to represent the stochastics, we discretized the chosen distributions by creating scenarios each having equal probabilities to occur. This process of creating scenarios to discretize the distribution representing the stochastics is known as scenario generation. We generated scenarios using the moment-matching method from Høyland, Kaut, and Wallace (2003). This method generates scenarios with a given correlation matrix and marginal distributions specified their first four moments (mean, variance, skewness and kurtosis). Since we use standard deviation equal to 25% of the mean, negative values happen with probability 0.000032, or 1 in 31 574. Thus when we generate scenarios, the possibility of getting extreme values are very low. But we have seen that when the mean value is a very small positive number we might observe demands with the wrong sign, and if so, we manually replace them with zero. Thus, in practice, we use truncated normal distributions to represent the stochasticity of demand.

The decision on the number of scenarios used to represent the stochastics is critical as we want to be sure we study the effects of randomness on our model, and not some random effect of the scenario generating procedure. There is a trade-off between the quality of scenarios representing the underlying distribution reasonably well and the time needed to solve the stochastic program to optimality. As we increase the number of scenarios, we increase the quality of the representation of the distribution, but also decrease the chance to solve the model to optimality within a manageable time. In our case, we generated 100 scenarios to represent the distributions as this gives us in-sample stability (solving the same problem on a large number of 100-scenario trees gives only a 2% difference between the highest and the lowest optimal objective-function values, except the case 'Molde' where it was 3.5%) and manageable solution times. For more discussion on this

subject we refer to Kaut and Wallace (2007).

3.2 Comparison Tests

It is well known that, in general, the solutions from the deterministic versions of a problem can behave rather badly in a stochastic environment. The reasons are outlined in some detail in Wallace (2000) and Higle and Wallace (2003). If the scenario tree used when solving the stochastic version of the problem is considered the "truth", then, by definition, the stochastic solution is always better than the deterministic one. But even though the objective function values corresponding to deterministic solutions at times were extremely bad, we seemed to observe that the structure of the deterministic solutions were retained in the stochastic solutions (see Section 3.3), an observation that is *not* common. We have thus devised three different tests to better understand the relationship between the deterministic and stochastic solutions.

A The classical test where the deterministic solution is evaluated using the scenario tree from the stochastic version of the problem. This amounts to solving the stochastic program with all first-stage variables fixed.

B Only edge information is imported from the deterministic case, the fixed setup costs g_k being set to zero for the edges opened in the deterministic case, while all other fixed setup costs are set to infinity (i.e. we do not allow these edges to be opened). The stochastic model is then solved.

C The deterministic solution (both edges and capacities) is taken as an input to the stochastic program. Then, for the stochastic program, these edges with corresponding capacities are "free". Both fixed setup costs g_k and variable setup costs h_k are paid for the installed capacities. The stochastic program can then add new edges (paying both fixed and variable setup costs) and new capacities on already opened edges (paying only variable setup costs). No premium is given for not using capacities opened in the deterministic case.

In all cases, for each of the comparisons, all costs are added up, both those inherited from the deterministic solution and those incurred via the stochastic program. All costs are therefore comparable.

The purpose of Comparisons B and C is to check whether the structure from the deterministic solution really is good for the stochastic case. By making edges from the deterministic case "free" in two different ways, the stochastic program is guided toward the deterministic solution. If this is not a good idea, the result will be a solution with a behavior which is much worse than that of the stochastic program which has no deterministic input. Note that, while Comparison B also represents an alternative solution procedure (use a deterministic model to determine which edges to open and then a continuous two-stage stochastic program to set the capacities), Comparison C does in itself imply the solution of a problem of the same type as the original problem, albeit with some discrete variables fixed.

3.3 Inheritance from the deterministic solutions

We focus on the major findings, while details of the results are given in the appendix. Our first need is to understand the relationship between the stochastic and deterministic designs. We therefore perform Comparisons A, B, and C from Section 3.2. That is, for all 25 deterministic cases, we solve the corresponding network design problem. Also, we solve all 75 stochastic cases, representing the stochastic versions of the deterministic cases (each with three different correlation structures).

Then, each of the 25 deterministic solutions are imported into its three stochastic counterparts (three different correlation matrices). This is done for all three comparisons. Figure 1 shows the results.

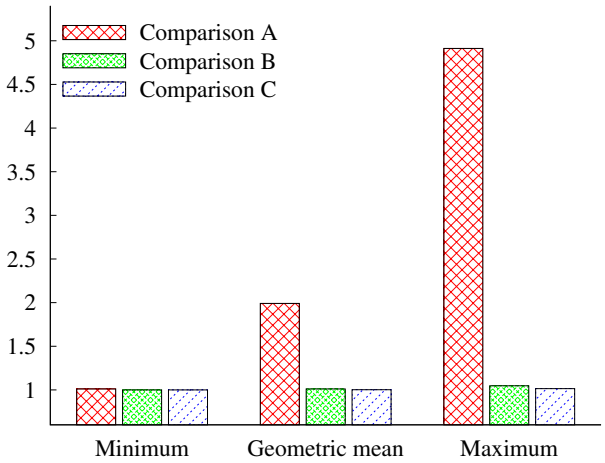

Figure 1: Ratios of the expected value of the deterministic solutions (25 base test instances) imported into the stochastic setting divided by the optimal objective function value for the stochastic program, for all three comparisons.

The deterministic solution is rather bad in the stochastic environment, while inheriting the structure seems to be rather good. For Comparison A, the deterministic solutions have expected objective function values which are from one percent to almost 400 percent higher than that of the stochastic counterpart. But Comparisons B and C show that inheriting the structure of the deterministic solution is surprisingly good. Errors ranges from 0% to 5%, and larger values are observed for the mixed and zero correlation cases. This is not unreasonable since these are the cases where we see, to the largest extent, the different demand nodes interact. We shall see more of that in Section 3.4.

We believe that a major reason for the somewhat surprising result that forcing the deterministic solution structures upon the stochastic program has so little effect is the fact that we are studying the single source case. In that case, all demand nodes are supplied from the same single node, which therefore becomes the root of a tree in the deterministic case. Since the whole point of network design is the fixed charges (particularly the fixed setup costs, but also the capacity costs) discouraging the opening of edges, this tree is also useful for the stochastic case, although it might have preferred a slightly different one (the stochastic solution must contain a tree rooted in the supply node). Hence, forcing it upon the solution of the stochastic program is not too serious. Note that the deterministic solution itself is at times very bad. It is only after capacities have been adjusted (Comparison

B) or new edges and capacities have been added (Comparison C) that the deterministic structure is good in most cases.

Let us turn next to testing these results when we vary the way setup costs are distributed between the fixed setup cost g_k and the variable setup cost h_t. Let M be the maximal capacity of an edge, as defined in our parameter list. For each of the 25 test cases, we calculate for each edge $C_k = g_k + Mh_k$. Then we redistribute C_k in following five different ways:

a Fixed setup cost 0.1% of C_k and variable setup cost $0.999C_k/M$

b Fixed setup cost 5% of C_k and variable setup cost $0.95C_k/M$

c Fixed setup cost 25% of C_k and variable setup cost $0.75C_k/M$

d Fixed setup cost 50% of C_k and variable setup cost $0.5C_k/M$

e Fixed setup cost 99.9% of C_k, and variable setup cost $0.001C_k/M$.

All tests described earlier were performed for each of these five cases and results are shown in Figure 2. We find that the deterministic solution is, as before, rather bad in the stochastic environment (with errors up to over 400%), while inheriting the structure is no longer as good as it was, in particular when the fixed setup cost is low and the capacity cost is high. This is natural, since in that case, the cost of opening one edge with a given capacity costs basically the same as opening two edges with the same total capacity. Hence, the structures enforced upon the solutions in Comparisons B and C become costly and make robustness more expensive to achieve. We see errors up to 18%, which in most real terms is rather high, but not in comparison to the behavior of the deterministic solution. The results seem little dependent on correlations.

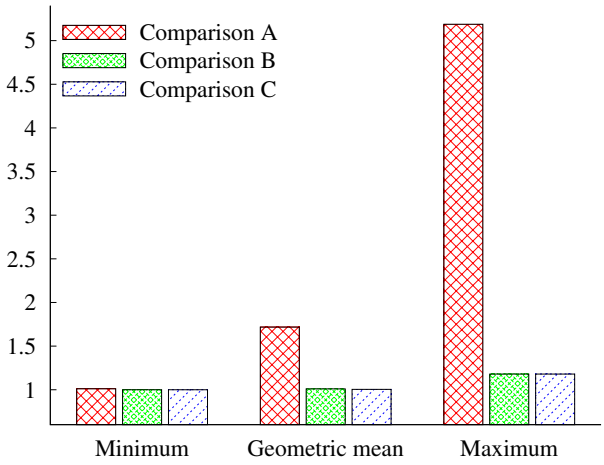

Figure 2: Ratios of the expected value of the deterministic solutions imported into the stochastic setting divided by the optimal objective function value for the stochastic program, for all three comparisons, taken over all ways to distribute costs between fixed setup costs and capacity costs

3.4 Structural differences

In Lium et al. (2009), some structures are observed for a multi-commodity case. These structures are very general and applicable across many different parameter settings. We were not able to find equally simple and general rules in the SSSND case and believe the reason to be the structural complexity of flow cancellation. But we do observe differences in design between the deterministic and stochastic formulations, which gives insights into what constitutes a robust design. Hence, what we shall do in this section is to provide a number of specific examples, all taken from the collection of problems in Section 3.1, and show in detail how the stochastic solutions differ from their deterministic counterparts. By doing so, we provide examples of how to think about flexibility in routing flow, and hence robustness in design. That, after all, is the goal of this paper. This structural knowledge can be used to develop heuristics as well as, and maybe more importantly, help researchers and practitioners alike to understand how to look at a given design and check its quality, even when no quantitative tool is involved. We want to develop a qualitative understanding of a robust design for SSSND. Some of the results are "obvious". Rather than this being a problem, we view it as a very desirable property. It means that robust designs are, at least structurally, not so difficult to understand.

Similarity

Let us first look at Figure 3 which covers two cases with uncorrelated demand where the fixed setup costs g_k are rather high. (Thick lines denote installed edges, with capacities given by labels, while thin lines denote edges that are not installed. The supply node is marked by a filled rectangle and the filled rounded nodes denote the demand nodes. The rest, i.e. the unfilled nodes, are the transshipment nodes. This color scheme is followed in all the subsequent figures.) The deterministic solutions are, of course, trees; we do not necessarily have spanning trees due to the presence of transshipment nodes. These trees are contained in the stochastic designs, but with different capacities. Generally, the capacities are higher in the stochastic case to cater for the high-demand scenarios. To what extent this happens (rather than high demands not being fully served) of course depends on the relationships among the different cost elements. The reason we get the same trees in these cases is that they represent the cheapest way of connecting all demand nodes with the source node (at least for the expected demand case). This becomes such a forceful property of the network that even in the stochastic case this structure is kept as long as the fixed setup costs are reasonably high. However, consider the bottom two graphs in Figure 3. They show the solutions corresponding to Comparison B, that is, when only edges opened in the deterministic case are allowed, but capacities can be set freely. What we see is that capacities generally are much higher than in the deterministic case. That is natural as one wishes to cater for high demand scenarios. But capacities are also generally higher than for the stochastic case, showing that even though the deterministic tree is present in the stochastic design, it is certainly valuable to add cross-over edges rather than just increasing capacities on the deterministic tree.

The advantage of test cases like this one, where the structure is similar, is that it is easier to see how the stochastics updates the tree and thereby what constitutes a robust design. Later we will see cases where the tree within the stochastic solution (there must be

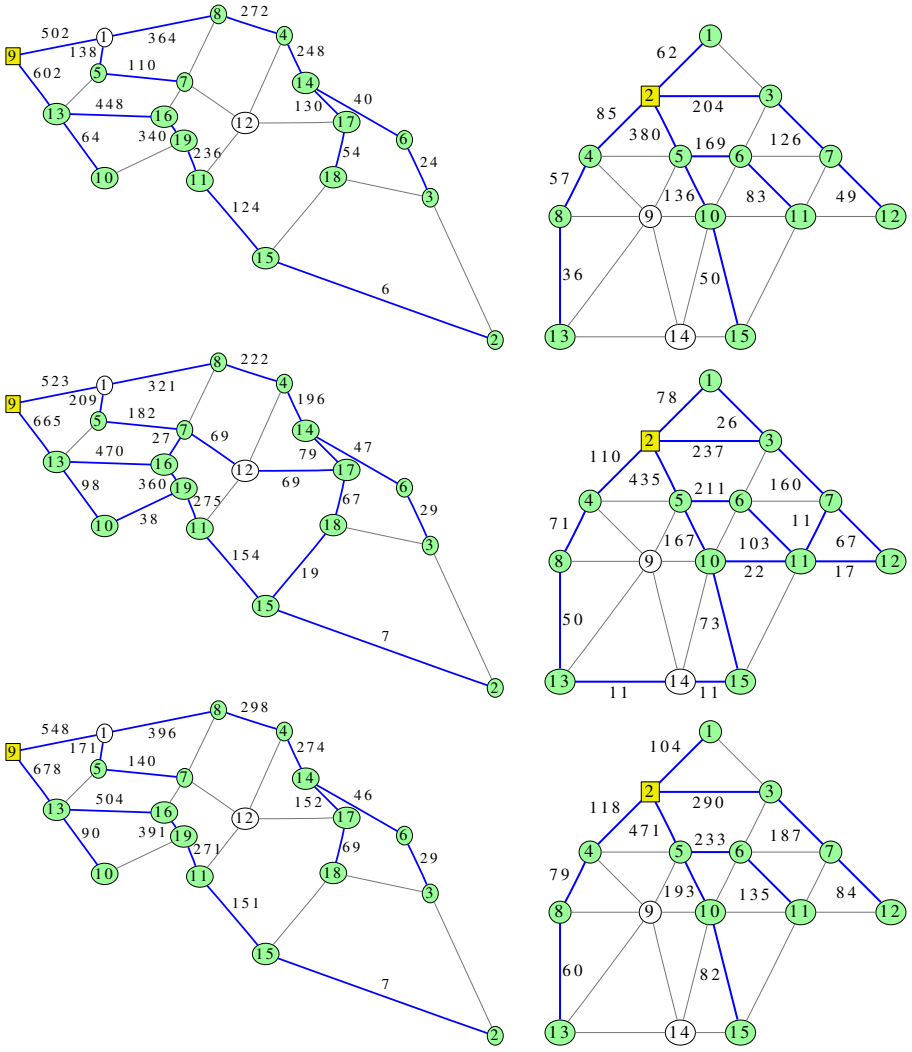

Figure 3: Deterministic, stochastic, and Comparison B solutions of Nobel_EU (left) and Molde (right) test cases showing presence of deterministic structures in the stochastic solutions.

one because we have only one supply node) is different from the tree of the deterministic solution.

Connecting branches

Sometimes branches in the deterministic design are connected in the stochastic design. Consider the Molde test case (right) in Figure 3. Note how an edge has been added between nodes 1 and 3, serving two purposes: If node 1 has a particularly large demand, it can be supplied via node 3. On the other hand, if demand in node 1 is small, the path via node 1 can be used to supply all nodes downstream from node 3. Also note how nodes 7, 10, 11 and 12 have been connected downstream from the supply node. This way less capacity needs to be added close to the source node on several branches since the branches can share capacities using these new edges. Also this makes evident that in stochastic design the tree actually splits farther from the source node as compared to the

deterministic design and this is to benefit from available installed capacities.

We can see the importance of these cross-over edges by comparing the stochastic solutions with the Comparison B solutions. As we just noted, the Comparison B solutions, which do not have cross-over edges, have higher capacities than the stochastic solution, as we see in Figure 3.

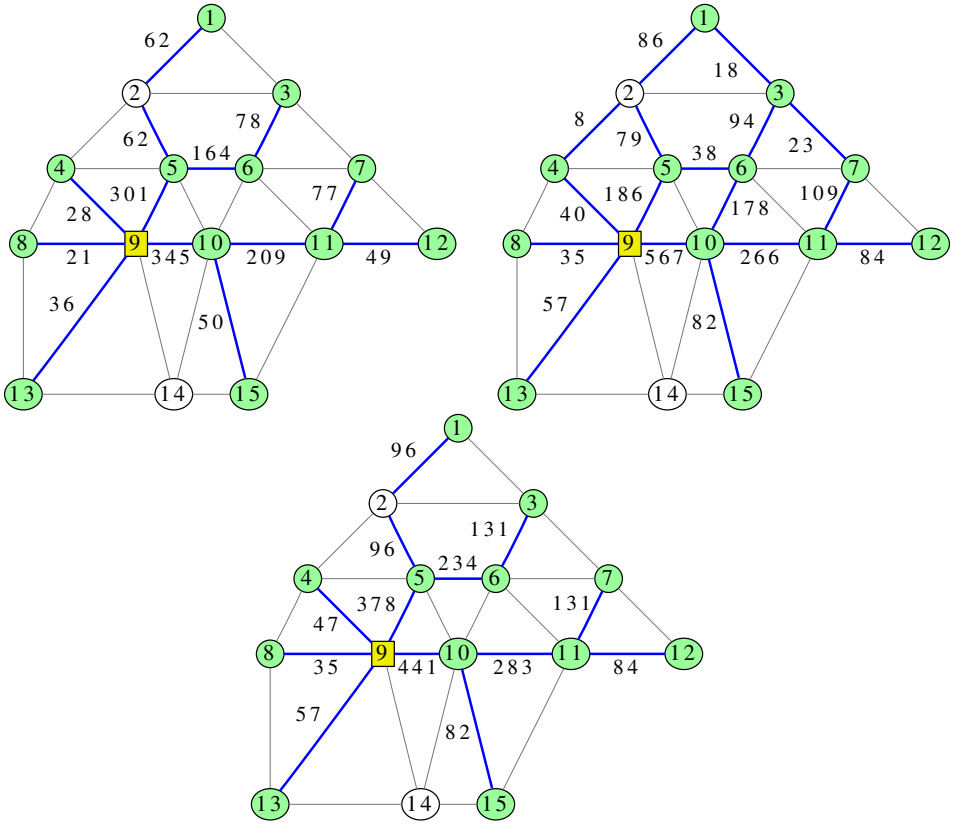

Figure 4: Deterministic (top left), stochastic (top right), and Comparison B (bottom) solutions of the Molde test instance showing connections between leaves of the deterministic solution tree in stochastic solution

In the deterministic solution of Figure 4, we see two major branches connecting the demand nodes with the source node. One of them (named branch-1) connects demand nodes 1, 3, 5 and 6 with source node 9, while the other (branch-2) connects 7, 10, 11, 12, and 15. In the stochastic solution we see that these two branches are connected by edges 6–10 and 3–7 (while 1–3 connects branches within branch-1). Branch-1 and branch-2 have 301 and 345 units of demands respectively in the deterministic case. In the stochastic case, branch-1 and branch-2 have maximal demands of 378 and 441 units, respectively. But if we compare the capacities of the edges coming out of the source node, we observe that in the deterministic case it is 646 in total, while for the stochastic case it increases by 16.5% to 753. The capacities installed on these branches in Comparison B, shown in Figure 4, is 819 units in total. This is an uncorrelated stochastic case, and there is no scenario with such a total demand (as it is very unlikely to happen given the distributional assumptions). So we can see that in the stochastic case, instead of installing a total of 819 units, the demand is managed by installing less (only 753 units in total plus some

assistance via node 4) and this is possible due to the edges 6–10 and 3–7 which help sharing installed capacity between the nodes of the two branches. The fact that the maximal branch demands cannot occur at the same time cannot be utilized in Comparison B.

Connecting leaves

In some cases the cross-over edges occur at the leaves of the deterministic tree, usually with moderate capacities. This typically happens when the nodes have comparable variation in demand. (In our test cases, where standard deviation is set at 25% of mean demand, it means we see nodes with similar mean demands). Two examples can be found in Figure 4 with the edges between nodes 1, 3 and 7 helping out all three leaves of the tree. The reason is that in these cases all three nodes can handle most of their high demand scenarios (which in this example occur independently of each other) using these moderate cross-over edges. If the variation in demand is very different between two leaf nodes, then we do not see this cross-over (discussed in detail in the next heading). However, generally, if the leaves are far from the source node the chances of edge formation is highest as then it becomes cheaper to open a new edge between them and share the capacity of the cross-over edge rather than increasing the flow capacity all along the two paths from the source node. This can be seen in the stochastic structure (link between demand nodes 13 and 15 via transshipment node 14) of Molde in Figure 3.

Balanced variation

We mentioned earlier that leaves of the tree are typically connected if the variation in demand are of comparable size, since the edge can then help out both the connected nodes. This is illustrated in Figure 5.

In the top right graph, we see a connection between leaf nodes 13 and 14, while there is none between 10 and 12. The mean demands are 9239 and 5582 for nodes 10 and 12. In the bottom graph we have changed the mean demands to the fairly similar values of 6639 and 5582. Then the cross-over edge appears. This is due to the fact that when the fixed setup costs are high (as they are here), these linking edges are economically useful only when used to resolve demand variation in both ends. If the variation in demand is very different between two leaf nodes, only the smaller of the nodes can fully utilize the cross-over edge (the other one needs more help), and then we typically do not see the cross-over edges. Remember that in our tests standard deviation is 25% of mean demand for all demand nodes.

Mixed correlations

Let us next turn to some cases where we face mixed correlations, that is, some correlations are positive, some negative, still with fixed setup costs relatively high. We have set up the correlation structure as follows: The demand nodes have been put into two groups such that each group have almost equal number of nodes. Within each group all demands are strongly positively correlated, while all correlations between pairs of node in different groups are strongly negatively correlated. So the assumption is that there is some underlying phenomenon that causes low demands in one group to typically match high demands

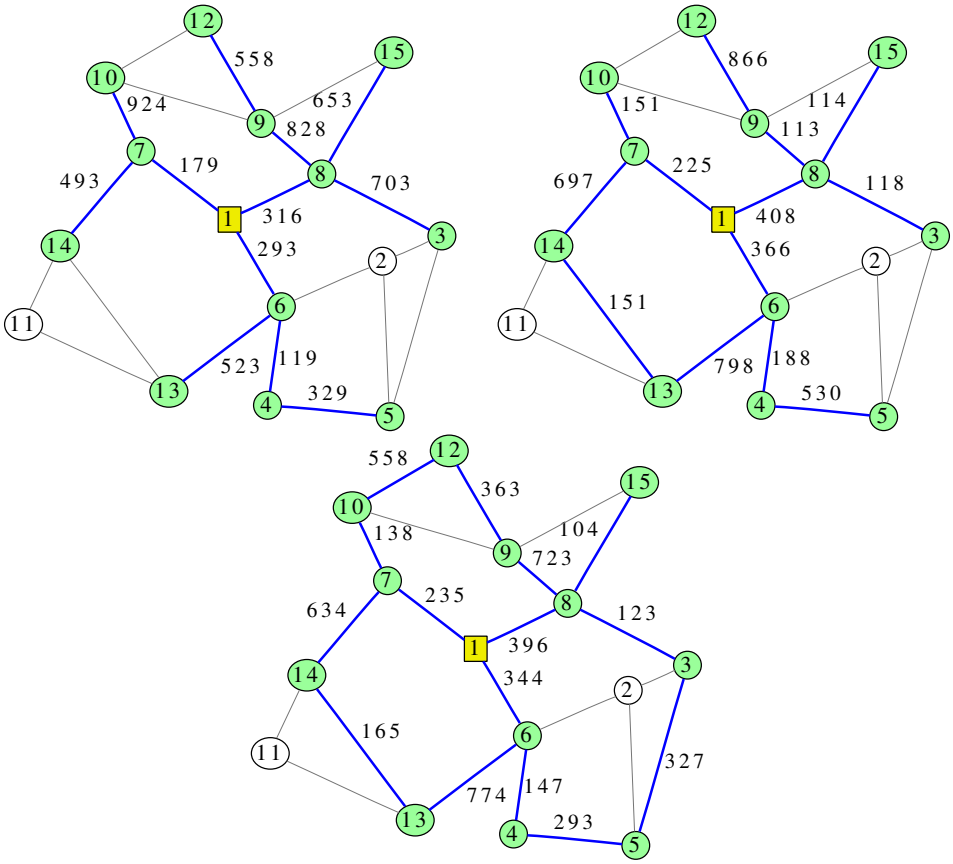

Figure 5: Deterministic (top left) and stochastic (top right) solutions of Atlanta. The bottom figure is the stochastic solution when demands at nodes 10 and 12 are changed to be of the same magnitude. The installed capacities are divided by one hundred for better readability.

in another group. For example, water demand tends to be high in residential areas on warm, dry days, while colder, wetter days cause high demand in indoor sports facilities.

The structure of the deterministic solution is apparent in the stochastic solution in most cases. But some of the edges become weaker in terms of capacities and some even disappear. The latter happens when the demand nodes can connect with other nearby negatively correlated demand node(s) more beneficially.

The new edges help demand nodes with negatively correlated demands fulfill their needs while opening lower capacity overall. This we see in the upper right graph of Figure 6 where edge 3–4 in the stochastic solution is between negatively correlated demand nodes. Similarly, edges 5–6 and 5–10 in the stochastic solution in the lower right graph are between negatively correlated demand nodes. The more negative the correlations, the more often we see this effect. When negatively correlated nodes are connected this way, alternative edges from the deterministic tree may disappear. This we can see in the lower left graph of the Figure 6 where edge 5–9 in the deterministic solution does not exist in the corresponding stochastic solution (lower right), due to new edges connecting negatively correlated nodes nearby.

With negative correlations present, most leaf nodes in the deterministic tree get con-

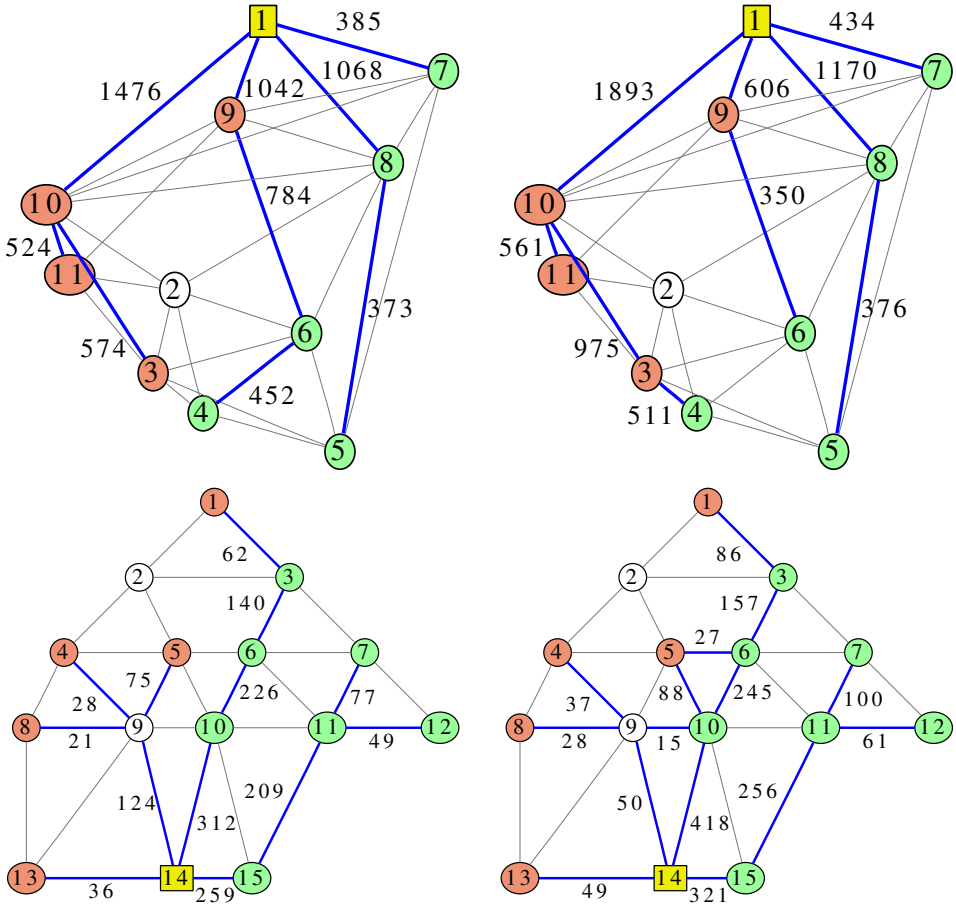

Figure 6: Deterministic (left) and stochastic (right) solutions of Pdh and Molde test instances showing connections between negatively correlated demand nodes. The two different shades denote the two groups of demand nodes.

nected to some other leaf nodes, where connections are guided by negative correlations. Further we observe that leaves from different branches are linked in the stochastic solution if they have similar variation in demand or if they are far from the source node as in the previous cases.

Positive correlations

Note that even positively correlated nodes may be connected. Typically, the connection is of moderate size, while the variations in demand are reasonably large (but of same size) for both nodes, so that even variation consistent with positive correlations can use a new edge in a balanced way. In addition, positively correlated demand nodes may be connected, because one of them (or both) are connected to negatively correlated nodes, creating a pool of nodes that can share capacity.

But generally, we observe that the solutions are very similar to the deterministic case. The reason is that the stronger are the correlations, the less likely it is that one demand node has low demand when another has it high, which removes incentives for capacity sharing. Because of the variable demand, the installed capacities are higher than in the

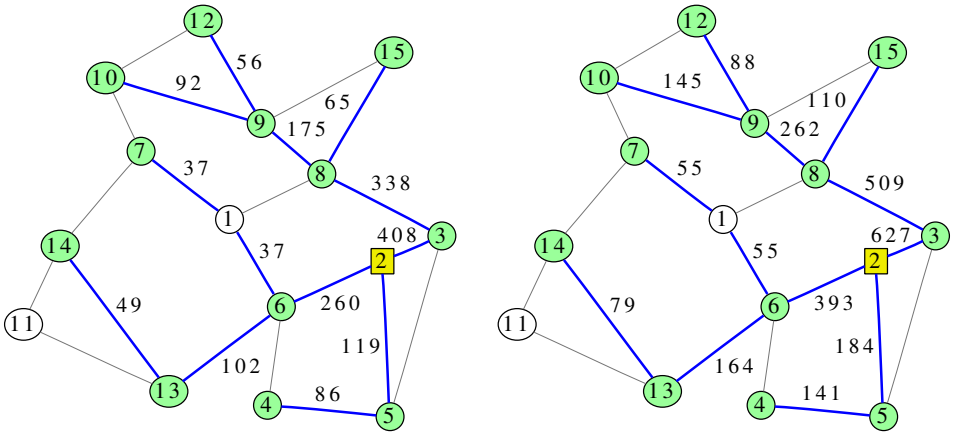

Figure 7: Deterministic (left) and stochastic (right) solutions of Atlanta test instance with positive correlations showing that structure is same, only with higher capacities. Note that here capacities are one hundredth of the actual values.

deterministic case (Figure 7), with the difference depending on the rejection costs. With weak positive correlations the solutions are close to the ones from the uncorrelated case which we already discussed.

No similarity

We have seen that in most cases the deterministic tree is kept in the stochastic solution (but with different capacities). We have also seen cases where a few edges disappear. Test case "Montreal" in Figure 8 illustrates a case where the stochastic solution has a sub-structure which differs from its deterministic counterpart. It concerns how node 3 is connected to the source (node 5). In the stochastic solution, robustness is achieved by letting nodes 3 and 7 share the edge leaving the source. For that sub-structure, Comparisons B and C end up with the same solution, which is the deterministic structure with added capacities. So, with the deterministic structure as a starting point, we do not get the robust structure of the stochastic solution. In terms of cost, we saw deviations up to 4 percent from the optimal solution. The stochastic solution is by definition the best one, so we see that by starting from the deterministic case, we miss the optimal structure, but for these problems with only one source node, these errors are low.

4 Conclusion

The purpose of this paper has been to better understand what constitutes a robust design for a single-source single-commodity stochastic network design problem. The single commodity case is, structurally speaking, more complex to understand than the multi-commodity case due to the phenomenon of flow cancellation. For that reason, we chose to start our investigation of the single commodity case with just a single source, to increase the chance of capturing the structural properties of robust designs.

Not very surprisingly, we find that the deterministic solution can be very bad indeed in terms of expected behavior. However, we also find that in most cases, the structure from

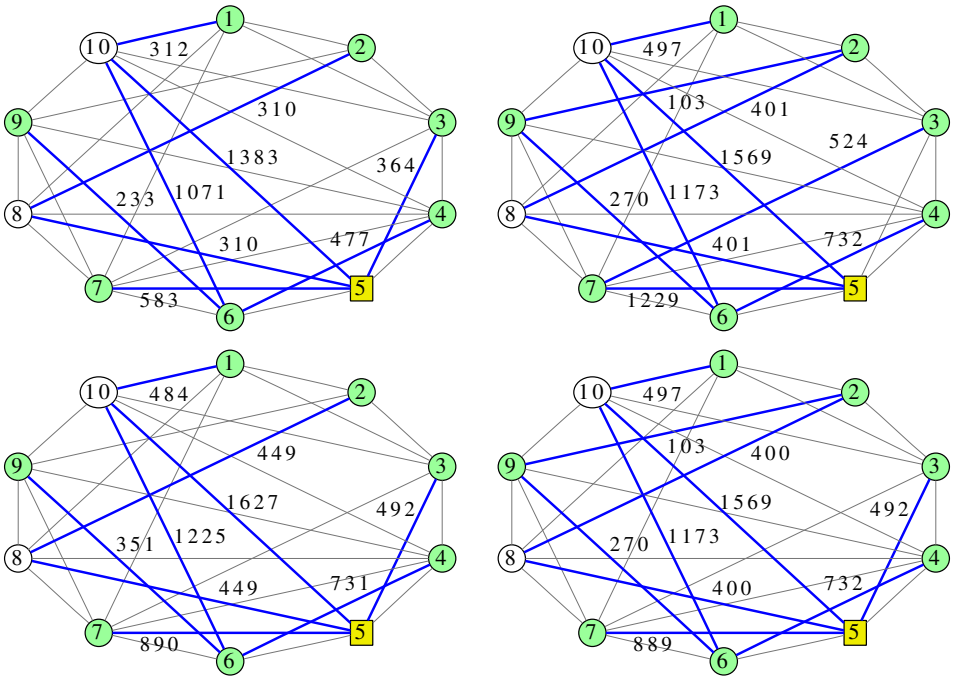

Figure 8: Deterministic (top left), stochastic (top right), Comparison B (bottom left) and Comparison C (bottom right) solutions of Montreal test instance where the stochastic solution structure is different from the deterministic one.

the deterministic solution tends to be good also in the stochastic setting, if we can adjust the capacities and/or add new edges.

Apart from this, it is impossible to make totally general conclusions. The results will always depend on the chosen numbers. However, there are certainly lessons to be learned. First, note that the deterministic solution will always be a tree as long as any amount of capacity can be opened on an edge. Secondly, if the fixed setup costs are large enough relative to the capacity and flow costs, also the stochastic solution will be a tree, as creating loops will simply be too expensive. It is not clear that we shall get the same tree, but since the deterministic tree carries the expected flow at minimal cost, it is also likely to carry the stochastic flow most cheaply. As the variance in demand increases, particularly when there are negative correlations, the stochastic tree will tend toward one where negatively correlated demand nodes sit on the same branches of the tree. If this was not the case in the deterministic solution (which is a fully random phenomenon), the trees will tend to be different. With high fixed setup costs, this will only happen if several trees have about the same fixed setup cost.

As the fixed setup costs decrease, we still get a tree in the deterministic case. And the very fact that this tree (where now capacity and flow costs count relatively more than before) carries the expected flow at minimum cost still carries weight in the stochastic case. Therefore, the deterministic tree tends to remain in the stochastic solution. However, in this case, where capacity costs are relatively more important than the fixed setup costs, it is much less costly to add new edges, creating circuits in the solution. We also occasionally see the structure change totally, and edges from the deterministic tree disappear. Important phenomena, which makes the stochastic structure different from determinis-

tic structure, are the size of the variation (the variance) representing how stochastic the problem really is, and the correlation structure.

A case where all correlations are positive and large is similar to a deterministic case with demands higher than the expected demands. We typically get a tree with higher capacities than in the deterministic case to facilitate the high demand scenarios. As before, if fixed setup costs are high, we tend to get the same tree, but if capacity costs dominate, the tree might be different. Positive small correlations are similar to the uncorrelated case.

With uncorrelated demands and moderate fixed setup costs, a number of important phenomena occur. If variance in demand is moderate, the deterministic tree is still a good candidate for carrying a major portion of the flow. However, in addition we observe cross-over edges between branches in this tree in the stochastic solution. The placement of these cross-overs will depend on the following factors: Firstly, nodes with similar variation in demand, if far from the source node, will tend to be connected with edges of moderate capacity, taking care of much of the (uncorrelated) variation between them. We may also see a group of nodes, lying far from the source node, connected that way. Secondly, somewhat downstream from where two branches split, we tend to see high capacity cross-over edges, used to make the branches help each other when demand varies. However, connecting two large branches (in terms of the number of demand nodes) too close to the node where the branches split is not useful, as each branch will tend to have a rather fixed demand due to the law of large numbers. Also, too close to the split it might be better to add capacity to both branches to avoid the fixed setup cost of the cross-over. So cross-over edges must be placed such that there is genuine (and preferably comparable in size) variation in demand downstream or upstream (or both) from the cross-over. In addition it is worth noting that in the stochastic case, there is a tendency for branches to split later than in the deterministic case, as that will tend to utilize the installed capacity better.

With negative correlations present, we see the phenomena from the uncorrelated case strengthened. In particular, cross-over arcs connect, when possible, nodes or clusters of nodes, with negatively correlated demand upstream or downstream (or both). The more negative, the better. A typical setting is a collection of leaf nodes, with some pairs having negatively correlated demands, being connected with moderately large edges, basically facilitating the demand variation in the whole collection.

So, what brings us furthest away from the deterministic tree is a case with large negative correlations, moderate fixed setup costs, and large variation (large variance) in demand.

We plan to follow up this work by studying the case of multiple source, as well as random edge failures.

Acknowledgments

This project was supported in part by grant 171007/V30 from The Research Council of Norway. While working on this project, T.G. Crainic was the NSERC Industrial Research Chair on Logistics Management, ESG UQAM, and Adjunct Professor with the Department of Computer Science and Operations Research, Université de Montréal, and the Department of Economics and Business Administration, Molde University College, Norway. Partial funding for this project has been provided by the Natural Sciences and

Engineering Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grants programs.

References

R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows – Theory, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs, NJ., 1993.

Jay E Aronson. A survey of dynamic network flows. *Annals of operations research*, 20: 1–66, 1989.

James R. Evans. A single-commodity transformation for certain multi-commodity networks. *Operations Research*, 26(4):673–680, July-August 1978.

J. L. Higle and S. W. Wallace. Sensitivity analysis and uncertainty in linear programming. *Interfaces*, 33:53–60, 2003.

D. S. Hochbaum and A. Segev. Analysis of a flow problem with fixed charge. *Networks*, 19:291–312, 1989.

K. Høyland, M. Kaut, and S.W. Wallace. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24(2–3):169–185, 2003.

Michal Kaut and Stein W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257–271, 2007.

Geok Koon Kuah and Jossef Perl. The feeder-bus network-design problem. *The Journal of the Operational Research Society*, 40(8):751–767, August 1989.

S. N. Kuan, H. L. Ong, and K. M. Ng. Solving the feeder bus network design problem by genetic algorithms and ant colony optimization. *Advances in Engineering Software*, 37 (6):351–359, June 2006.

Lou Y. Liang, Russell G. Thompson, and David M. Young. Optimising the design of sewer networks using genetic algorithms and tabu search. *Engineering, Construction and Architectural Management*, 11(2):101, 2004.

A.-G. Lium, T. G. Crainic, and S. W. Wallace. Correlations in stochastic programming: A case from stochastic service network design. *Asia-Pacific Journal of Operational Research*, 24(2):161–179, 2007.

A.-G. Lium, T. G. Crainic, and S. W. Wallace. A study of demand stochasticity in stochastic network design. *Transportation Science*, 43(2):144–157, 2009.

B. Rothfarb and M. Goldstein. The one-terminal telepak problem. *Operations Research*, 19:156–169, 1971.

B. Rothfarb, H. Frank, D. M. Rosenbaum, K. Steiglitz, and D. J. Kleitman. Optimal design of offshore natural-gas pipeline system. *Operations Research*, 18:992–1020, 1970.

Hanif D. Sherali and Ernest P. Smith. A global optimization approach to a water distribution network design problem. *Journal of Global Optimization*, 11(2):107–132, September 1997.

Stein W. Wallace. Decision making under uncertainty: Is sensitivity analysis of any use? *Operations Research*, 48(1):20–25, 2000.

Stein W. Wallace. Stochastic programming and the option of doing it differently. *Annals of Operations Research*, page to appear, 2009.

Appendix

Results of the numerical tests

This appendix provides detailed results from the tests in Section 3. We provide the numbers used to generate Figures 1 and 2 in Section 3.3. Also the discussions in Section 3.4 are based on these computations, but the individual cases cannot be reproduced from these tables.

Table 2: The numbers corresponding to Figures 1 and 2

	Figure 1			Figure 2		
	A	B	C	A	B	C
Minimum value	1.011	1.000	1.000	1.011	1.000	1.000
Geometric mean	1.991	1.011	1.002	1.718	1.010	1.004
Maximum value	4.912	1.047	1.015	5.188	1.180	1.180

Since the results, particularly for Figure 2, depend on correlations, we also show the ratios in Table 2 split by correlation structures – see Tables 3 and 4. Finally, the full computational results for Figure 1 are presented in Table 5.

Table 3: The numbers corresponding to Figure 1 split by correlation structure.

	Zero correlations			Mixed correlations			Positive correlations		
	A	B	C	A	B	C	A	B	C
Minimum Value	1.017	1.004	1.000	1.022	1.000	1.000	1.011	1.000	1.000
Geometric Mean	1.973	1.016	1.003	1.994	1.017	1.003	2.005	1.001	1.001
Maximum value	4.721	1.047	1.011	4.910	1.040	1.015	4.912	1.007	1.007

Table 4: The numbers corresponding to Figure 2 split by correlation structure.

	Zero correlations			Mixed correlations			Positive correlations		
	A	B	C	A	B	C	A	B	C
Minimum Value	1.013	1.000	1.000	1.019	1.000	1.000	1.011	1.000	1.000
Geometric Mean	1.705	1.011	1.002	1.737	1.011	1.002	1.712	1.009	1.008
Maximum value	5.013	1.075	1.048	5.188	1.081	1.018	5.059	1.180	1.180

Table 5: The numbers corresponding to Figure 1 split by correlation structure.

		Zero correlations				Mixed correlations				Positive correlations			
Test name	Det. opt.	stoch. sol.	comp. A	comp. B	comp. C	stoch. sol.	comp. A	comp. B	comp. C	stoch. sol.	comp. A	comp. B	comp. C
Atlanta_S01	9087238	11953640	40578824	12045757	11953600	12212052	41959064	12373942	12212100	13053547	45623641	13053796	13053500
Atlanta_S02	11276107	14515466	40580320	14715094	14580000	14274385	38818806	14444797	14371800	16231748	47478400	16231936	16231700
Atlanta_S11	14719239	18547523	42502387	19168231	18586200	18664619	43860743	19200906	18664600	20891150	50660785	20908154	20909400
Atlanta_np_S01	7651846	10424209	42281086	10608271	10542400	10701375	44580111	10952420	10753100	11119443	45041234	11119638	11119500
Atlanta_np_S02	8896342	11855965	42095193	12059936	11936300	11949849	43977023	12422449	12004200	12835045	46295293	12862006	12861900
Atlanta_np_S11	9376441	12352942	43430017	12485820	12353100	12586100	46175315	12738631	12734800	13459920	46531703	13460053	13457900
France_S12	1296060	1605774	3083243	1617190	1616120	1618355	3033761	1623340	1623340	1734378	3403562	1742550	1742550
France_S13	920793	1186527	2937197	1198880	1198560	1253813	3106080	1263260	1263260	1279881	3126648	1288440	1288440
France_S16	818906	1073065	2836188	1077570	1077370	1144028	3005087	1144030	1144030	1156005	3025673	1156010	1156010
Nobel_EU_S01	812311	910231	976753	918145	910317	914078	995130	934043	915393	958505	1024510	958505	958505
Nobel_EU_S09	882938	982927	1040790	991573	982993	982288	1058930	996235	984074	1031535	1090050	1031780	1031780
Nobel_EU_S12	524481	608254	711571	611000	608254	622640	744766	626142	623027	636951	754379	636951	636951
Nobel_EU_np_S01	806992	908173	979108	918197	908252	913626	997367	937951	914387	954287	1022489	954286	954288
Nobel_EU_np_S09	882938	982977	1040787	991616	983022	981534	1058929	996264	983634	1031598	1261370	1031825	1031840
Pdh_S01	10975138	12128934	12331328	12236609	12236600	12144531	12408971	12339476	12250800	12334799	12474156	12383528	12383500
Pdh_S02	7466965	8684380	9037947	8731629	8731520	8657235	9137165	8806293	8787060	8806447	9128912	8815660	8815590
Molde_S02	6246	7616	34329	7977	7616	8088	37417	8339	8088	8640	40380	8640	8640
Molde_S09	5848	7249	34220	7483	7249	7542	37032	7730	7542	8148	40026	8148	8148
Molde_S14	7919	9516	35376	9869	9524	9776	37282	10014	9789	10995	42058	11011	11007
Molde_np_S02	6186	7345	8952	7465	7345	7577	9138	7613	7577	7712	9414	7712	7712
Molde_np_S09	5722	6875	8553	6963	6877	7006	8553	7046	7006	7182	8983	7182	7182
Molde_np_S14	7810	9108	10585	9332	9113	9198	10585	9417	9199	9602	11025	9602	9602
Montreal_S05	459011	501498	505828	497250	662183	671818	651666	503461	505947	503597	501713	505817	497742
Montreal_S08	382511	423140	433463	415731	578759	600968	561071	427026	436598	417414	423138	433598	415727
Montreal_S10	476996	532313	538141	528131	679966	692610	678876	542017	542934	544220	533672	538876	529772

Paper 3

Single-Commodity Stochastic Network Design with Multiple Sources and Sinks

Single-Commodity Stochastic Network Design with Multiple Sources and Sinks

Biju K. Thapalia* Teodor Gabriel Crainic† Michal Kaut‡

Stein W. Wallace§

30 November 2009

Abstract

This paper examines the single-commodity stochastic network design problem with multiple sources and sinks. We characterize the structures of the optimal designs and compare with the deterministic counterparts. We do this primarily to understand what constitutes good robust network designs, but hope that the understanding can also be used to develop better heuristic algorithms than those available today.

Keywords: single-commodity network design, multiple sources and sinks, stochastic, correlations, robustness

1 Introduction

Many operations-research (OR) applications, as well as problems in computer science, applied mathematics, and many fields of engineering and management are based on network formulations with an underlying design problem, see for example Ahuja, Magnanti, Orlin, and Reddy (1995). Today's complex supply chains require goods and information to be distributed in many layers and in integrated ways. Increased competition force decision-makers to study the whole supply chain, all the way from suppliers to end consumers, trying to achieve overall optimality.

Network design has been a major area of research for the last four or five decades and shows great diversity in methodology, see for example Scheibe and Ragsdale (2009). But still we know very little about the structural characteristics of the optimal designs. We are interested in revealing structural properties of the designs that can be used to understand / evaluate designs even without solving the corresponding design problems.

It is evident that in most cases, at the time when a network is designed (or expanded), the demand or supply that it will later face is uncertain. Traditionally this is not taken into account during the design phase, but rather, the handling of uncertainty is postponed

*Molde University College, biju.k.thapalia@himolde.no

†University of Quebec at Montreal

‡Norwegian University of Science and Technology

§Lancaster University Management School

to the operational phase of the problem at hand. While it is true that the actual handling of uncertainty – meaning the reaction to revealed information – by definition, must take place when it occurs, it is equally clear that different designs offer different opportunities for *how* the uncertainty is handled, in particular, how costly the handling might be. This principle is well explained in for example Yen and Birge (2006). A discussion may also be found in Ball, Barnhart, Nemhauser, and Odini (2007). So apart from understanding designs in general we are particularly interested in understanding how designs stemming from assuming deterministic demands differ from designs where uncertainty is included already in the design phase of a project. We ask: Does it matter? Are there recognizable differences between the two designs? Technically speaking, we shall compare, in different ways, designs coming from two-stage stochastic programs (where the design is stage 1 and the commodity flows stage 2) and their deterministic counterparts (where random demand is replaced by expected demand).

A common way to handle this situation is to perform single- or multi-parameter sensitivity analysis in order to understand how the optimal solution changes as a function of demand. This approach might seem appropriate, but in fact it is not. This is outlined in detail in Wallace (2000) and Higle and Wallace (2003). Logically, when performing sensitivity analysis, one is assuming that the design can be postponed until after demand has become known. So, whether sensitivity analysis is performed or not, we end up with a solution not created to handle uncertainty, and hence, we may have to face difficult operational decisions when demand is revealed.

It is old news that a deterministic solution might perform very badly in a stochastic environment. The reason is simply that it is not made to handle variation in parameters such as price or demand in a good way. This argument often follows the logic of ”The value of the stochastic solution”, see Birge (1982). In the network design case, good designs stem from flexibility in the commodity flows, i.e., the ability to utilize installed capacity across very different demand realizations. We have illustrated this in Thapalia, Crainic, Kaut, and Wallace (2009): the deterministic solution is itself badly suited to handle stochastic demand for the single commodity, single source, multiple sink network design problem. However, in the same paper we also observed that the structure (i.e., which edges to open) might be similar in the deterministic and stochastic cases, albeit with rather different capacities installed. Even this kind of similarity is unusual.

Lium, Crainic, and Wallace (2009) found consolidation to be a way to hedge against uncertain demand in their multi-commodity stochastic service network design model. This cannot (of course) be observed in the solution to the corresponding deterministic model, as the model has no reason to hedge against uncertainty. The deterministic design might contain volume-related consolidation, but that is not enough to cater properly for uncertainty. So in that case, not only is the expected behavior of the deterministic design bad, but also the structure (i.e. information about which edges to open) is of limited value. A question for this paper is therefore: As we pass to the case of multiple sources for the single-commodity case, shall we observe that the structure of the deterministic solution is good (as we observed in the single-source single-commodity case) or bad as in the multi-commodity case?

Hence, while we focus on comparing stochastic and deterministic designs, it is not primarily to (once again) show the weaknesses of the deterministic solution, but to really

understand in what ways (if any) the deterministic solution is good and in what ways it is bad. We also hope that this can be used not only to obtain a deeper understanding of the effects of uncertainty on design, but also to develop heuristics for the stochastic case.

2 Problem description

Given a set of nodes (divided into source nodes, demand nodes, and transshipment nodes) and a set of potential edges connecting these nodes, the single-commodity stochastic network design problem with multiple sources and sinks (MSSND) is the problem of determining a subset of the edges to open (including the edges' capacities), so as to fulfill the demand at the demand nodes at minimal cost, taking into account capacities of the source nodes.

In general, the stochastics in this problem arises in the form of demand uncertainties at the demand nodes, supply uncertainties at the source nodes, and failure of connections (or failure of certain proportion of capacities in the edges) between the nodes. Demand uncertainties and edge failures are observed in most real life problems, as it is rare that demand is fully known when the design is determined or that edges never fail. In this paper, we discuss only random demand. The design is based on minimizing the sum of the fixed costs of selecting edges connecting the nodes; linear costs to open capacities in the edges; per unit flow costs of flows on the edges; and per unit penalty costs for not satisfying demand. Not satisfying demand can have many interpretations, such as sending the flow at a later point in time, with another mode, or a straightforward rejection. In any case, in the model, it takes the form of a penalty cost per unit of unsatisfied demand.

It is important to include the possibility of flow being rejected in the model. The main reason is that in real life, except in extremely particular situations, it is prohibitively costly to build a network that can meet any possible demand—however unlikely it might be. Deterministic models, operating on expected demand, may reasonably operate under the assumption that (average) demand *must* be met. But even there, there will normally be an understanding that some demand may end up being turned down in reality. When working with stochastic demand, there is also the problem that requiring demand to be met turns the model into a worst-case model, where the worst-case in most cases is not even well understood. So, in total, we find it crucial to include the possibility of not satisfying all the demand. We use the same formulation also in the deterministic models, to make the results comparable.

We shall let all source nodes have the same capacity so that our focus will be on the random demand. Hence, our assumption is that a set of demand nodes will have their random demands satisfied from a set of equally-sized source nodes.

In the deterministic case, the demand in each node is fixed at the expected demand from the stochastic case. This corresponds to the classical case of a single-commodity multiple source network design problem. The first stage decisions in this problem are to decide which edges to open and what capacities to install. The second stage decisions are the flow decisions in the given network. The recourse action here is described by a penalty cost incurred for not satisfying demand.

2.1 Mathematical formulation

Let $G = (\mathscr{N}, \mathscr{E})$ be a network defined by a set $\mathscr{N}$ of n nodes and set $\mathscr{E}$ of m edges (undirected arcs), where

$$\mathscr{E} \subset \{k = (i, j) : i \in \mathscr{N}, j \in \mathscr{N} \text{ and } i < j\}.$$

Each edge is indexed either by i, j or by k.

The random demand is described by a set of scenarios $\mathscr{S}$, where each individual scenario $s \in \mathscr{S}$ has one demand realization for each demand node. We shall discuss in Section 3.1 how the scenarios were generated. The notations for the sets, parameters, and variables associated with this problem are as follows:

Sets:

$\mathscr{C}$	set of all source nodes;
$\mathscr{D}$	set of all demand nodes;
$\mathscr{T}$	set of all nodes with zero demand (transshipment nodes); $\mathscr{T} = \mathscr{N} \setminus (\mathscr{C} \bigcup \mathscr{D})$;
$\mathscr{S}$	set of all scenarios s.

Parameters:

M	maximal arc capacity; used for linking capacities and open arcs in (5);
R	unit cost of unsatisfied demand;
P^s	probability of scenario $s \in \mathscr{S}$;
C_k	flow cost on edge $k \in \mathscr{E}$;
G_k	fixed setup cost for edge $k \in \mathscr{E}$;
H_k	variable setup cost; the cost for adding one unit of capacity to edge $k \in \mathscr{E}$;
V_k	initial/ existing capacity on edge $k \in \mathscr{E}$, if any;
D_i^s	demand ($D_i^s < 0$) in node $i \in \mathscr{D}$ in scenario $s \in \mathscr{S}$;
D	supply in each source node, $D > 0$.

Variables:

$x_k^s = x_{ij}^s$	flow on edge $k = (i, j) \in \mathscr{E}$ going in direction $i \to j$, in scenario $s \in \mathscr{S}$;
$z_k^s = z_{ij}^s$	flow on edge $k = (i, j) \in \mathscr{E}$ going in direction $j \to i$, in scenario $s \in \mathscr{S}$;
u_k	new capacity that is developed on edge $k \in \mathscr{E}$;
e_i^s	for $i \in \mathscr{D}$, this is the unsatisfied/lost demand in node i in scenario $s \in \mathscr{S}$;
	for $i \in \mathscr{C}$, this is the unused capacity of source node i in scenario $s \in \mathscr{S}$;
y_k	1 if edge $k \in \mathscr{E}$ is developed, 0 otherwise.

We assume that total supply from equally-sized source nodes equals maximal demand in the network, so that

$$D = \max_s \{ \sum_{j \in \mathscr{D}} \{D_j^s\} \} / |\mathscr{C}| \tag{1}$$

where $|\mathscr{C}|$ is the number of source nodes.

Our overall problem is hence:

$$\min \sum_k G_k y_k + \sum_k H_k u_k + \sum_s P^s \left\{ \sum_k C_k \left(x_k^s + z_k^s\right) + R \sum_{i \in \mathscr{D}} e_i^s \right\} \tag{2}$$

Subject to:

$$\sum_{j:(ij)\in\mathscr{E}} \left(x_{ij}^s - z_{ij}^s\right) - \sum_{j:(ji)\in\mathscr{E}} \left(x_{ji}^s - z_{ji}^s\right) = \begin{cases} 0 & \forall i \in \mathscr{T}, \forall s \in \mathscr{S} \\ D - e_i^s & \forall i \in \mathscr{C}, \forall s \in \mathscr{S} \\ D_i^s + e_i^s & \forall i \in \mathscr{D}, \forall s \in \mathscr{S} \end{cases} \tag{3}$$

$$x_k^s + z_k^s \leq u_k + V_k \;\; \forall k \in \mathscr{E} \;\; \forall s \in \mathscr{S} \tag{4}$$

$$u_k \leq M y_k \qquad \forall k \tag{5}$$

$$0 \leq e_i^s \leq -D_i^s \qquad \forall i \in \mathscr{D}; \forall s \tag{6}$$

$$x_k^s, z_k^s, u_k, e_i^s \geq 0 \text{ and } y_k \in \{0,1\} \qquad \forall k; \forall i; \forall s \tag{7}$$

The objective function (2) minimizes the total costs of the network. The first part is the costs of constructing all new edges, the second part the costs of building all the new capacities, the third part the expected flow costs through all the edges and the fourth part is the expected penalty costs of not fulfilling demand. Constraints (3) model conservation of flow at nodes. The left-hand side is the net outflow from node i, which must be zero for all transshipment nodes $i \in \mathscr{T}$ and is equal to the unused capacity for source node $i \in \mathscr{C}$. For the demand nodes, the net outflow must be equal to the satisfied demand; since D_i^s is negative in this case, the right-hand side is the a difference between the scenario demand D_i^s and the (positive) unsatisfied demand e_i^s.

Constraints (4) represent the flow limit in each edge. The left hand side of the equation is the net flow on edge k which should be less then or equal to the total capacity of the edge. Since we do not start with any initial/existing capacity in our test cases, we always have $V_k = 0$. Note that in an optimal solution, an edge will never have flow in both directions. Constraints (5) show that new capacity u_k can be developed only if edge k is built. Constraints (6) give bounds for the rejection amount and finally, (7) insure that all variables are non-negative and the edge constructions binary.

For the deterministic counterpart we replace the stochastic demand by its expectation.

We model the problem in AMPL and solve it to optimality using CPLEX 9.0. The solution time varies from few seconds to 5 hours depending on the case, on a PC with 3 GHz Intel® CPU and 8 GB of RAM.

3 Experimentation and Computational Results

The tests have two related goals. First we primarily focus on the quality of the deterministic designs by trying to understand how they differ from their stochastic counterparts,

and to what extent solving a deterministic problem will guide us toward a good design in a stochastic environment. Does the deterministic design contain useful information, or is it totally misleading if the real setting is that of stochastic demand? In the second part we try more directly to characterize good designs, assuming that the stochastic design model is the appropriate one. Our goal is to make qualitative statements about what characterizes of a good design in light of random demand. These two questions are of course related, but we find it useful to have these two focuses.

In order to answer these problems we have constructed a number of test cases. These are now described together with our scenario generation approach.

3.1 Test instance generation

We have used seven different network instances. The first four instances, namely Germany, Nobel-EU, NY, and US are telecommunication examples from the SNDlib library (Orlowski, Pióro, Tomaszewski, and Wessäly, 2009), with some modification to suit our problem's needs. The fifth case was generated by us and named Molde and the last two, Montreal_r06.1 and Montreal_r10.1 were obtained from CIRRELT (Interuniversity Research Center on Enterprise Networks, Logistics and Transportation), Montreal. The names of the instances do not mean anything particular in our computational setup.

It is worth noting that in all cases from SNDlib and Montreal, the test instances are multi-commodity network design problems, so not all parameters can be used directly by us. We only kept the coordinates (where available) for the nodes and the fixed setup cost G_k for the edges. The values for the other parameters – variable setup costs H_k and flow costs C_k – are all chosen proportional to the Euclidean distance between the node pairs. The cost of unfulfilled demand R is derived for each test case using some multiple of the highest value of the fixed plus variable setup cost for an edge in the network. We made sure that R is not driving the solution. The results in the first part of Section 3.3 are based on test cases with these cost structures. In the second part of the section, we changed the fixed costs to understand their relative importance.

The Montreal test instance does not have node coordinates, so we used Graphviz (Gansner and North, 2000) to draw the graph using fixed setup cost as distance measure. The graphs of the test instances Nobel-EU, US and Molde are planar whereas the graphs of the test instances Germany, NY, Montreal_r06.1 and Montreal_r10.1 are non-planar.

For each of the seven problems, we picked 3 sets of nodes (2 in the case of Montreal_r06.1) as possible source node sets, thus creating in total 20 base test instances. These 20 different versions of the problem instances are presented in Table 1. A set of source nodes contains three or four nodes depending upon the test instances; the number of source nodes for each test instance is listed in the fifth column of Table 1.

Given the difficulty of solving the stochastic network design problem to optimality we kept n (the number of nodes) below 30 and m (the number of edges) below 50 for the first six cases, while for the Montreal_r10.1 cases we have up to 87 edges.

We know from the work of Lium, Crainic, and Wallace (2007) that correlations might be important in shaping the structure of the network. Hence, we further create 3 cases for each problem instance: one with uncorrelated demands, one with positively correlated demands (all correlations are set to 0.7), and one with mixed correlated demands: the

Table 1: The different test cases. Test case Molde is generated by us. For the others, the names have been kept, even though the cases are adjusted to our needs.

Problem name	# nodes	# edges	# demand nodes	# sources	# source sets
Germany	29	48	9	3	3
Nobel-EU	28	41	8	4	3
NY	16	41	7	4	3
US	26	42	9	3	3
Molde	22	45	8	4	3
Montreal_r06.1	10	37	5	3	2
Montreal_r10.1	20	87	6	4	3

demand nodes are put into groups such that each group contains about half of the total number of nodes. All correlations *within* a group are set to 0.7, while *between* groups we use -0.7. All of this leads to positive definite correlation matrices. Thus we have in total 60 test cases.

As stochastic programs need discrete distributions to represent the stochastics, we discretized the chosen distributions (discussed below) by creating scenarios each having equal probabilities to occur using the moment-matching method from Høyland, Kaut, and Wallace (2003). In the absence of reference to a particular distribution representing the random demand we chose to use truncated normal distributions with mean equal to the deterministic demand (from the underlying cases) and standard deviation equal to 25% of the mean to represent its stochasticity.

The decision on the number of scenarios used to represent the stochastics is critical as we want to be sure we study the effects of randomness on our model, and not some random effect of the scenario generating procedure. There is a trade-off between the quality of scenarios representing the underlying distribution reasonably well and the time needed to solve the stochastic program to optimality. As we increase the number of scenarios, we increase the quality of the representation of the distribution, but also decrease the chance to solve the model to optimality within a manageable time. In our case, we generated 100 scenarios to represent the distributions as this gives us in-sample stability and manageable solution times. The in-sample stability is checked by solving the same problem repeatedly with different 100-scenario trees. This lead to a coefficient of variation (the standard deviation divided by the mean) of less than 1%, except for the cases of 'NY', 'Molde' and 'Nobel-EU' where it was 1.2%, 3.9%, and 4.8% respectively. The cases of 'Molde' and 'Nobel-EU' have very high rejection costs, so even a minor change in rejection volume results in a large change in objective function value.

With these values we are satisfied that we have in-sample stability for the problems at hand. Since this is a necessary, but not sufficient, property of a satisfactory scenario generation procedure, we also check out-of-sample stability. Out-of-sample stability is checked by creating scenario trees with 1000 scenarios using sampling and then evaluating the solutions over those scenarios. The evaluation is performed by re-optimizing the flow in the network with given designs i.e., fixing the first stage variables of problem (2) to (7), representing the network design under evaluation. The procedure is repeated 10

times for a given network design and the objective values are compared by again calculating the coefficient of variation. For our problem, out-of-sample stability was achieved as the coefficient of variation for all the test instances was less then 0.2%, except for the cases 'NY', 'Molde' and 'Nobel-EU' where it was 2.9%, 1.4%, and 2.6% respectively. For more discussion on this subject we refer to Kaut and Wallace (2007).

Finally, we have to reconsider the definition of the supply D given in (1): since it depends on the actual realizations of the stochastic demands D_j^s, it would be different for the three versions (with different correlations) we generate for each test case, making it difficult to compare the results. To avoid this problem, we calculated D for each of the three correlation versions of a given case and used the median of the three D's as our demand size in all three versions.

3.2 Comparison Tests

As outlined in the Introduction, the deterministic solution, by its nature, has a worse expected behavior than its stochastic counterpart. However, we would like to understand more about why this is the case, and in what sense it is worse.

In order to check the quality of the deterministic designs, as well as comparing them to the stochastic ones, we have set up three tests, named *comparisons*. Whenever a comparison is performed, we take the deterministic and stochastic designs – or parts thereof – (i.e. the first-stage solutions) and evaluate them using reference trees – in our case trees with 1000 scenarios, to make sure we have good approximations of the true distributions. The costs from the design and evaluation phases are added up, making the reported costs comparable across all tests.

A word of warning might be worthwhile here. If a stochastic programming problem, as well as its deterministic counterpart, use hard constraints in the formulation, the deterministic solution will normally be infeasible in the stochastic formulation (caused by capacity problems when demand is high), and hence, its expected cost will be infinitely large. On the other hand, if soft constraints are used, the deterministic solution will normally be feasible in the stochastic model, but its expected performance can be made arbitrarily bad by choosing large penalties on the soft constraints. This way, it is always possible to make the deterministic solution look bad. We shall, however, set the penalties at reasonable levels, and our goal is to *understand* how the deterministic solutions relate to their stochastic counterparts. So, we shall certainly present numbers, and we do believe the numbers are informative. But there will never be really objective numerical results in this setting.

The three comparisons are:

A The classical test where the whole first-stage solution is evaluated out-of-sample. This amounts to solving a 1000-scenario stochastic program with all first-stage variables (designs and capacities) fixed, so in fact this equals the solution of 1000 independent second-stage problems. Since the second stage does not involve any integer variables, this is very fast.

B Only edge information is imported from the first stage. So, in a 1000-scenario stochastic program, all discrete variables y describing opened and closed edges—

we call it a *skeleton*—are fixed and the stochastic program is run. So the model is allowed to install any capacity on the opened edges (also lower than in the deterministic case), but not to open new ones.

C The whole design (both the skeleton and its capacities) is taken as input to the 1000-scenario stochastic program. The stochastic program can then add new capacities on already opened edges (paying only variable setup costs) and new edges (paying both fixed and variable setup costs). Hence, all capacities opened in the deterministic case add cost to the objective function, even if these are not needed in the final design.

The purpose of Comparisons B and C is to check if the design from the deterministic solution really is good for the stochastic case, and if it bad, in what way it is bad. By making edges from the deterministic case "free" in two different ways, the stochastic programs (as defined in the comparisons) are guided toward the deterministic solution. This way we compare if stochastic programs solved with input from the deterministic solutions behave much worse than stochastic programs which have no deterministic input (they will never behave better).

So, Comparison A is the classical test of the quality of the deterministic solution. Comparison B, on the other hands, checks if we can use a deterministic method to determine the skeleton and then solve a stochastic *linear* program to set the capacities. If Comparison B comes out with good results for the deterministic solution, it points to an alternative solution procedure that avoids solving a stochastic mixed integer program: First use a deterministic method to find the skeleton, then a stochastic linear program to set capacities. This represents a severe saving in computation (if it works well, of course).

Comparison C can be seen as testing what happens if we first solve the deterministic design problem and implement the solution, but then discover that it is not very good, and wish to update it. If Comparison C comes out well for the deterministic solution, a deterministic design can be corrected and become almost optimal for the stochastic case provided setup costs must not be paid again. If Comparison C comes out badly, the costs of updating a deterministic design in light of uncertainty in demand will be high. Note that Comparison C is itself a stochastic mixed integer program, so in most cases it does not represent an alternative solution approach. In our tests, though, Comparison-C with 1000 scenarios is faster than the original stochastic problem with 100 scenarios. But for large problems, both are unsolvable.

In what follows of this section, we discuss the major findings, details are given in the Appendix. Our first need is to understand the relationship between the stochastic and deterministic solutions. We therefore perform Comparisons A, B and C as discussed in Section 3.2. That is, for all our 20 deterministic cases, we solve the corresponding network design problem. Also, we solve all 60 stochastic cases, representing the stochastic versions of the deterministic cases (each with three different correlation structures).

Then each of the 20 deterministic designs are imported into its three stochastic counterparts (the three different correlation matrices). This is done for all three comparisons. In all cases, as outlined earlier, the evaluations are done out-of-sample using 1000 scenarios. Figure 1 shows the results, where also the stochastic designs are evaluated out-of-sample.

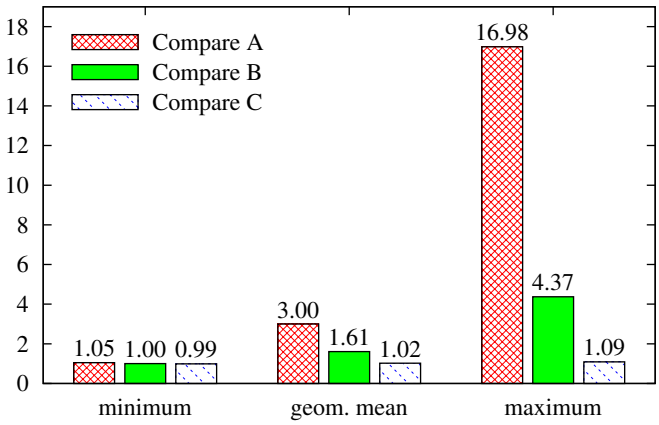

Figure 1: Results of the Comparison Tests. Ratios between the expected objective values, obtained from importing the deterministic solution into the stochastic setting, and the expected objective function value of the stochastic solution.

3.3 Inheritance from the deterministic solutions

The deterministic solution is bad in the stochastic environment, and inheriting the structure (the skeleton) is also not good, see Figure 1. For Comparison A, the deterministic solutions have expected objective function values which are from five up to almost 1600% higher than that of their stochastic counterparts with mean value of 200%. For Comparison B errors are from 0% up to little over 300% with mean value of 61%. On the other hand, for Comparison C the errors are very low, from 0% up to just 9% with mean value of 2%. This shows that when we allow to add new edges and open new capacities, the deterministic design can be updated to become almost as good as the stochastic design. (Note that in these tests we might observe that the deterministic design is better than the one coming from a 100-scenario stochastic model, since both designs are evaluated out-of-sample with 1000 scenarios. We have observed one single case which can be seen in Figure 1.)

Now, if we look at Figure 2, which compares the cases where the costs of the first stage decisions (the fixed and/or variable setup costs) are highest (the test instances of Molde, Nobel-EU, and NY), then we see that the *best* design for Comparison A is 4.17 times more costly than its stochastic counterpart. And on average, the deterministic design produces expected costs that are 10.22 times what a stochastic model would produce. If we further look only at certain components of the overall costs for these cases, which are presented in Figure 3, we find that on average, the costs for opening edges are just 74%, and capacity built is just 62%, of that in the stochastic designs. This indicates that the solutions are far from the optimal structure both in terms of edges opened and capacities built.

So what do we see? Not very surprisingly we observe that the deterministic solution behaves rather badly in a stochastic environment, implying that the common practice of creating a design based on expected values and then handling randomness operationally is not a very good idea—the costs can be astronomical. One reason for this is simply that the deterministic design does not install enough capacity – even in our case where the penalty is set at a very reasonable level. So what if only the skeleton – the edges to be opened –

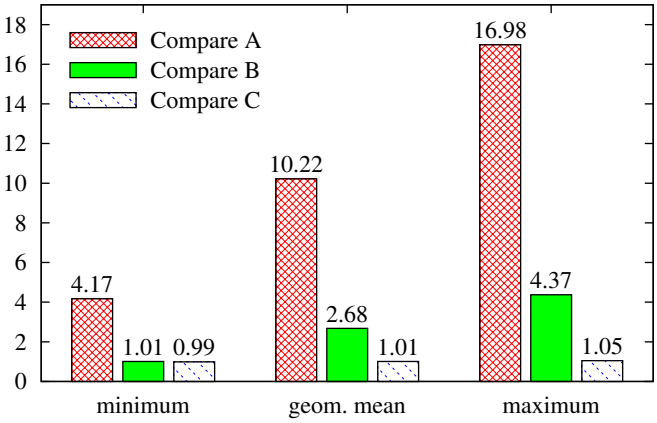

Figure 2: Results for cases with higher cost of first stage decision. Ratios between the expected objective value, obtained from importing the deterministic solution for the test cases with high fixed and variable setup cost into the stochastic setting, and the expected objective function value for the stochastic solution.

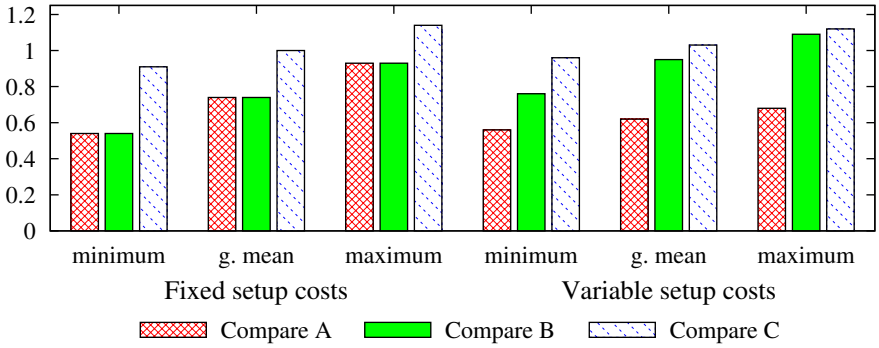

Figure 3: Components of result. Ratios of the fixed and variable setup costs obtained from importing the deterministic solution, for the test cases with high fixed and variable setup cost, into the stochastic setting divided by the fixed and variable setup costs for the stochastic solution.

is imported from the deterministic design, and the capacities are set as in Comparison B? This is computationally effective, as solving the stochastic program of Comparison B is very simple even for huge problems (it has no integer variables). It helps, of course, but we can still be several hundred percent off, which in most cases is not acceptable. There are situations where Comparison B does well, but it isn't easy to know upfront if a given case is of that type.

The results of Comparison C are worth an extra comment as they do not represent a very common situation: if the deterministic design is taken as a starting point, a very good (even if not optimal) design can be found by adding extra edges and capacities on top of the deterministic one. In our test cases, we never lost more than 9% that way. This is still a large number in many cases, but given the uncertainty in the model (which of course is always there) this is not bad. Computationally, we must then solve a deterministic design problem first, and then a stochastic one (Comparison C). This stochastic program has substantially fewer integer variables than the original one, but can still be expected to

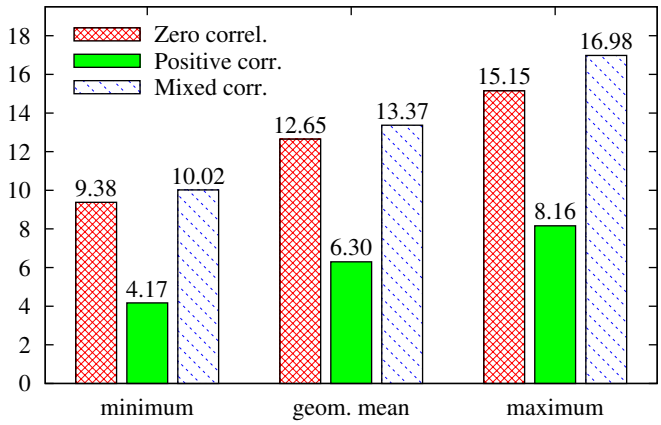

Figure 4: Results for Comparison A for the cases with higher cost of first stage decision. Ratios between the expected objective value, obtained from importing the deterministic solution into the stochastic setting, and the expected objective function value for the stochastic solution.

be as unsolvable as the original one for practical problems. This is not the main point, though. The main observation is that the deterministic solution can be updated to become very good even in a stochastic environment.

It is worth mentioning that the results in Comparison C are not in line with the general observation that a stochastic solution is normally *not* the deterministic one "plus something"—see Wallace (2009) for reasons why. In our case that is exactly what we observe (genuinely or as a good approximation): the stochastic design equals the deterministic design plus "something".

For the single-source case, as described in Thapalia et al. (2009), Comparison B came out rather well contrary to what we observe here. This difference can mainly be attributed to the fact that we now have many source nodes with limited supply capacity. As we minimize costs, the deterministic skeletons have many short paths, we term them *arms*, connecting individual source nodes to nearby demand nodes. If not generally, this typically gives us a forest of small trees. Just adjusting the capacities of these trees is not enough to find good designs. When there is only one supply node, the deterministic skeleton is a tree, and hence, all nodes are connected, even if the connections are not optimal. When the skeleton is a forest, there are simply too few connections.

We know from Lium et al. (2007) that correlations may play important roles in shaping the solution structure of the stochastic problem in terms of sharing capacity and taking benefit from variation in demand. By importing the deterministic solution into the stochastic problem, the deterministic solution structure with short arms from each source node cannot benefit from this variation in demand. Hence we see poor performance in Comparison A. This is more evident in the cases where we have higher setup costs (fixed, variable, or both) as these result in deterministic designs with particularly short arms. Consider Figure 4, where we observe that it is worse in the uncorrelated and mixed correlated cases as compared to the positively correlated cases. This is natural since with strong positive correlations there is less to be gained from joint use of edges in any case.

Relation between the number of source nodes and inheritance

The extreme case—one source node—was covered in Thapalia et al. (2009). In that case both Comparisons B and C were rather good. We have already seen that for three or four source nodes - as in this paper - Comparison B is no longer very good, while Comparison C remains very strong. With high setup costs this is even more evident, as that will cause the skeleton to be as minimalistic as possible in terms of the number of edges. We wonder if also Comparison C will become weak as we get more source nodes.

Some of this we already understand: With one source node, the deterministic skeleton is a tree (not necessarily spanning because of the transshipment nodes), while as the number of source nodes increases, we tend to get several trees, in the extreme case, one for each source node, and the ability to share capacity when randomness hits becomes steadily lower. Comparisons A and B, limited by the deterministic skeleton, suffer from this lack of connectedness – as it prevents sharing of supply capacity – and hence they do not do very well.

In order to better understand the effect of the number of source nodes, we have increased the number of source nodes for a few cases. What we observe is that Comparison C gets steadily worse as the number of source nodes increases. However, be aware that it is not easy to define what "these two cases are the same except that one has more source nodes than the other" means. The reason is that as the number of source nodes increases, the whole network design problem changes, and comparisons become unclear. In this paper we are limited to cases we can solve to optimality. That prevents checking the fate of Comparison C for really large cases. Within what we could check, we found that Comparison C got worse as the number of source nodes increased, but remained very good throughout, the deterministic solution never being more than 10% worse than the stochastic one.

So, it seems, taking the deterministic design and adding edges and capacities produces good solutions. Note again, however, that since Comparison C is also a stochastic integer program, it it likely to be as unsolvable as the original stochastic program for large realistic cases.

Fixed costs

We want to make sure that the observations of how the stochastic solutions differ from the deterministic ones do not depend on the cost structures we have used. So we turn to testing these results when we vary the way setup costs are distributed between fixed setup cost G_k and variable setup cost H_k. Let L be some large positive number, selected conveniently. Here we take it to be 25% of M. For each of the 20 test cases, we calculate for each edge $C_k = G_k + LH_k$. Then we redistribute C_k in five different ways:

a Fixed setup cost 0.1% of C_k and variable setup cost 99.9% of C_k/L

b Fixed setup cost 5% of C_k and variable setup cost 95% of C_k/L

c Fixed setup cost 25% of C_k and variable setup cost 75% of C_k/L

d Fixed setup cost 50% of C_k and variable setup cost 50% of C_k/L

e Fixed setup cost 99.9% of C_k, and variable setup cost 0.1% of C_k/L.

All tests described earlier are now performed for each of these five cases and results are shown in Figure 5. The tests are performed with 100 scenarios. Some test instances which CPLEX could not solve within 15 days are ignored.

We find that the deterministic design is, as before, rather bad in the stochastic environment (with errors up to nearly 2700%), while Comparisons B and C are getting somewhat worse, in particular when the fixed setup cost is low and the capacity cost is high. This is natural since in that case the cost of opening one edge with a given capacity costs basically the same as opening two edges with the same total capacity. Hence, the structures enforced upon the solutions in Comparisons B and C become more costly. We see errors up to nearly 700% in the case of Comparison B, and up to 11% in the case of Comparison C—still quite good. The results seems little dependent on correlations. The details of the results are presented in the Appendix.

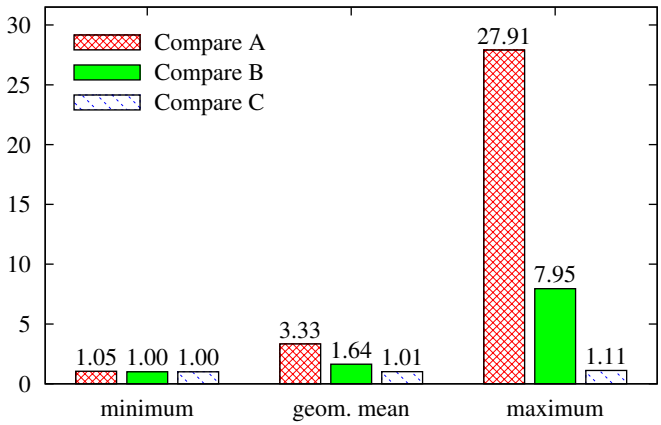

Figure 5: Results of the Comparison tests for different distributions of setup costs. Ratios between the expected objective value, obtained from importing the deterministic solution into the stochastic settings, and the expected objective function value for the stochastic solution.

3.4 Structural differences

So far we have discussed different ways to bring the deterministic design into the stochastic environment to understand to what extent the deterministic design is useful in creating good solutions. Now we shall pass to a more direct comparison of the the structures of the stochastic and deterministic designs, instead of just looking at their expected costs.

The optimal structures in single-commodity network design seem to be more complicated to *understand* than those of the corresponding multi-commodity cases. The main reason is that in the multi-commodity case the commodities only share edge capacities whereas in the single-commodity case there is also the phenomenon of flow cancellation. So while it is easier to *solve* single-commodity flow problems (as standard network flow theory can be applied directly), the optimal design is harder to characterize.

As already pointed out, the stochastic designs tend to have more capacity and more edges than the deterministic counterparts. For the cases of uncorrelated and mixed correlated demands, the extra edges and capacities are mainly there to cater for capacity sharing. This results in loop formations, leave connections, and connections between different clusters of nodes. The trees with few short arms, typical of the deterministic skeletons, will generally not allow sharing based on some demands being large when others are small simply because there are few demand nodes in each tree, and there is no particular reason why demand nodes with negatively correlated demand end up in the same tree.

In the cases of positive correlations, the edges and capacities are mostly there to cater for the high-demand scenarios. Two phenomena occur: The high demand scenarios (which now have high probabilities attached to them) need much more capacity than the deterministic (expected value) case, and the limited capacity of the individual source nodes makes it necessary to connect them so that all supply is used well. These connections are simply to few (if at all) in the deterministic designs. In other words, even though the demands are positively correlated, there is some variation, and connections are needed to utilize overall supply.

Let us now turn to a more direct study of the stochastic designs rather than primarily *comparing* stochastic and deterministic designs to understand the qualities of the deterministic ones. We know that good designs stem from flexibility in the routing of flow, and the goal is then to understand how this is achieved. We shall do this by studying the problems from Section 3.1. Some of the results are "obvious" meaning that good robust designs are, at least structurally, not so difficult to understand. We consider this a strength.

Similarity in designs

The deterministic skeletons are trees (not spanning trees) with arms emerging from the different source nodes. If the supply capacities are tight, the trees might be connected to each other so that the supply nodes can help each other satisfy demand in "their" demand nodes. These skeletons are contained within the stochastic designs under certain conditions. Remember that the trees from the deterministic designs represent the cheapest way to connect to the demand nodes in the average case. Hence, if the capacities of the source nodes are high enough to handle high demand scenarios, then we often see that the deterministic skeletons are part of (or even the same as) the stochastic skeletons. Figure 6 compares the deterministic design with two stochastic designs, one with high demand variation and one with low. (Solid edges (blue) are installed with the given capacities, light (grey) edges are not installed. The dark (yellow) square nodes are the supply nodes, the shaded (green) circular nodes are demand nodes, and the white circular ones transshipment nodes. This color scheme is followed in all subsequent figures.) The low variation case has the same skeleton as the deterministic design. This is because with lower variation the source nodes still have sufficient capacities to fulfill the demand in most scenarios (there is some unsatisfied demand). The higher variation case results in high demand scenarios, where some source nodes may have insufficient capacity to fulfill the demands of "their" demand nodes and hence we see a re-alignment of the distribution patterns to fulfill more demand than would be possible from the deterministic skeleton.

Also when setup costs are substantially lower than flow costs, the deterministic skeletons are contained in the stochastic skeletons. This is because the deterministic design

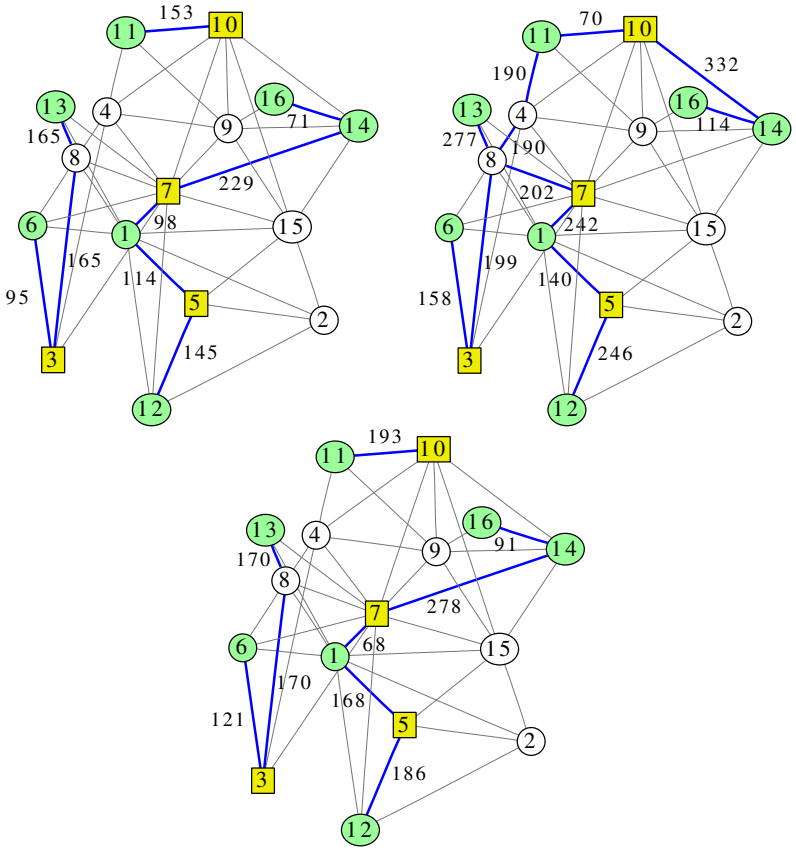

Figure 6: Comparison between deterministic and stochastic structure. Deterministic (top left), stochastic with high variance (top right) and low variance (bottom) solutions of the NY_02 test case showing that the stochastic solution structures are similar to the deterministic ones in the low variation demand case, but rather different in the case of high variation.

represents the cheapest way to transport the bulk of the demand. Longer routes will result in substantially higher flow costs. On top of the deterministic solution, new edges needed to facilitate high demand scenarios and coordination of source nodes are cheap to add. In Figure 7 we see that both the mixed correlated and uncorrelated cases contain the deterministic skeleton, and add a few extra edges to meet higher demands. For example, the demand node 15 connects to source node 2 (for uncorrelated case) and node 15 & 3 get connected to source node 12 (for mixed correlated case) in the stochastic solution. This helps satisfying the higher demands. But the deterministic skeleton is fully used in the stochastic designs.

When source nodes are far from the demand nodes, we also find the deterministic skeletons, more or less fully, within the stochastic ones. Consider Figure 8. In the first column we see a case where the deterministic skeleton is almost kept. This is because source nodes 2 and 12 are in the corners of the graph and both are far from demand nodes 22 and 20. Hence, the paths needed to reach those demand nodes are long. In that case it is usually better to keep these cheapest connections even in the stochastic case. But even so, the mixed correlated case is different. This is caused by another feature of the stochastic solution which we shall discuss later in the section on negative correlations.

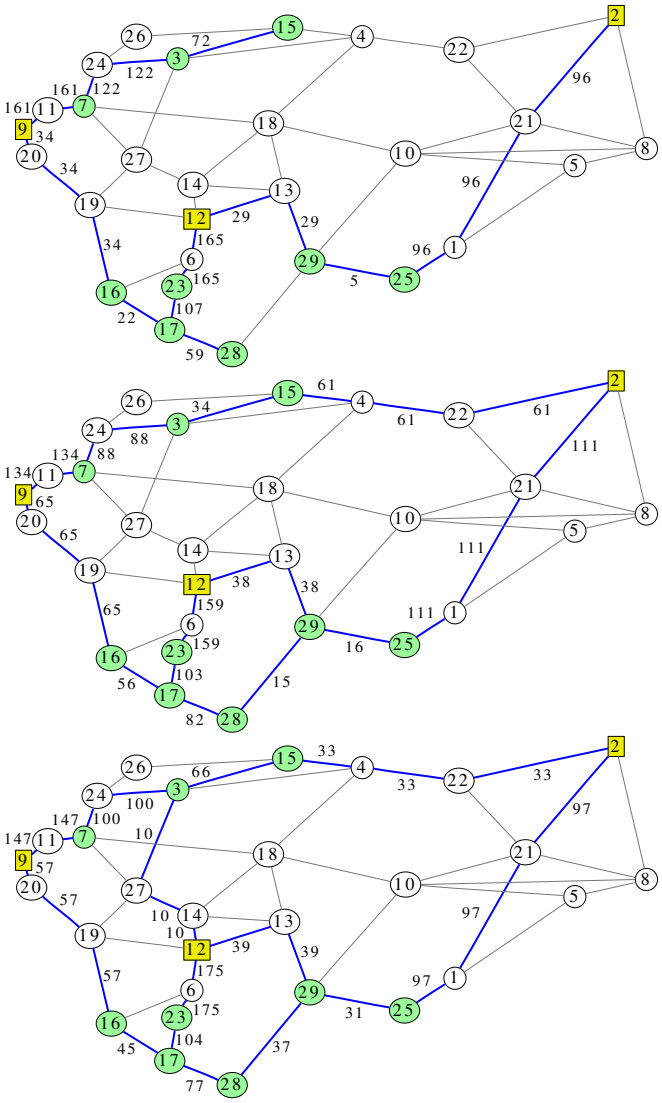

Figure 7: Similar deterministic and stochastic structure. Deterministic demand (top), uncorrelated stochastic demand (middle) and mixed correlated stochastic demand (bottom) solutions of Germany_01 test cases showing that the stochastic solution structures are similar to the deterministic ones when setup costs are low.

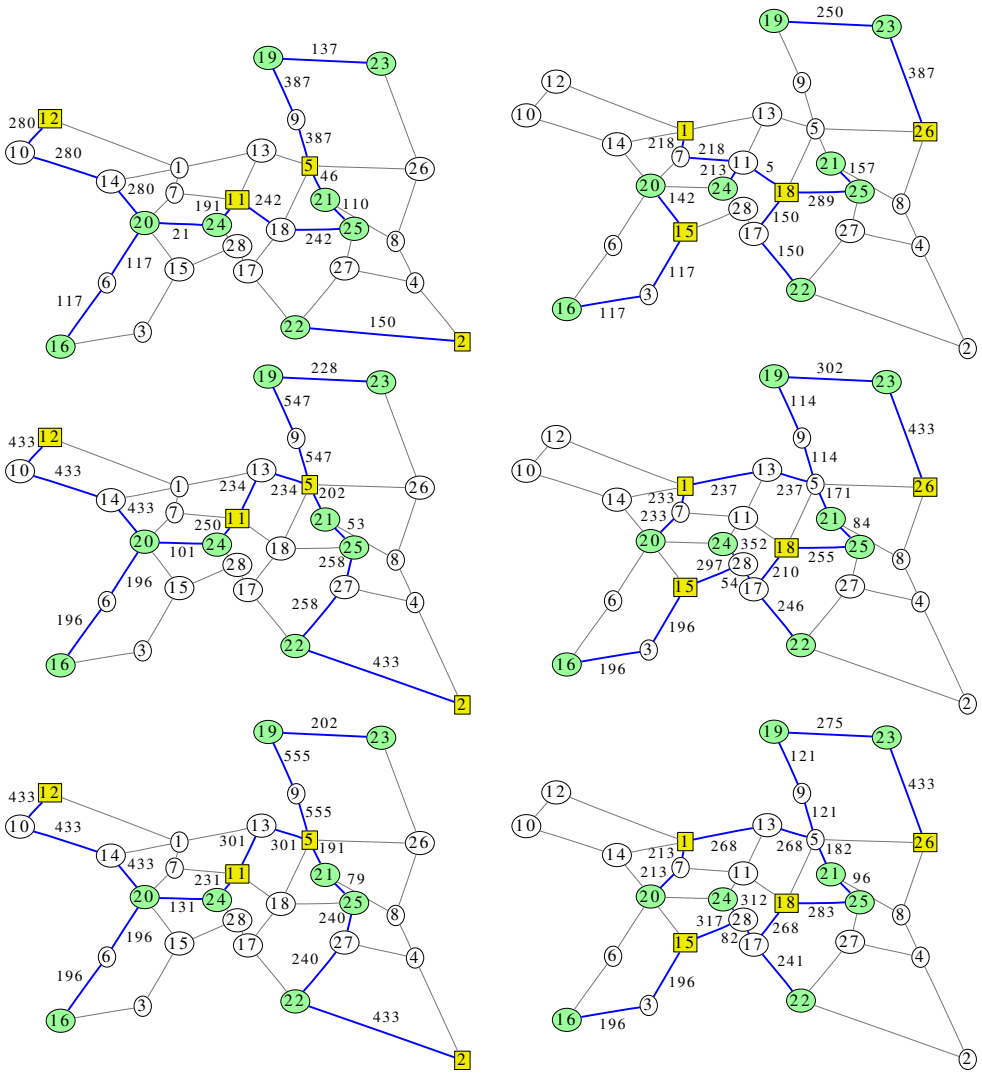

Figure 8: Deterministic skeletons in stochastic solution. Deterministic demand (top), uncorrelated stochastic demand (middle) and positively correlated stochastic demand (bottom) solutions of Nobel-EU_01 (left) and Nobel-EU_02 (right) test cases showing that the stochastic solution contain the deterministic skeletons when source nodes are far from demand nodes and not otherwise.

For the case in the second column, the skeleton is not retained, as here the source nodes are very near to the demand nodes and hence have the possibilities to utilize variation of demand and re-align the design.

Consolidated paths

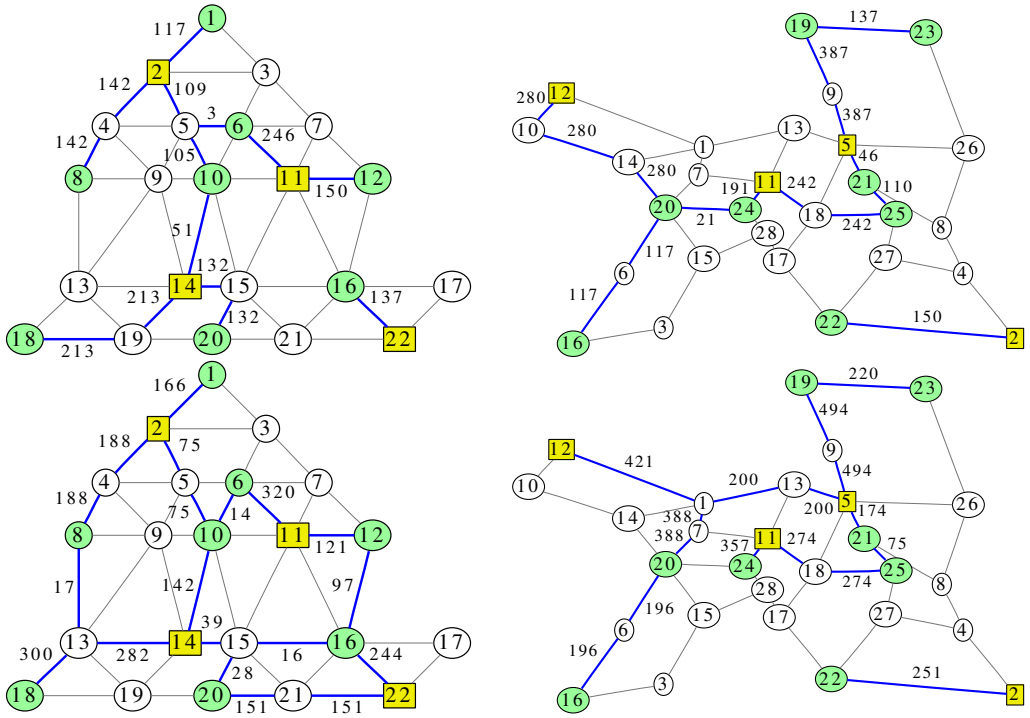

Figure 9: Consolidated paths. Deterministic (top) and stochastic (bottom) solutions of Molde and Nobel_EU test cases, showing that the stochastic solution structures have consolidated paths which are not in the deterministic solutions.

In the stochastic design we sometimes observe that there are paths which act like highways carrying supply and demand for many nodes. This is more evident in the cases where the setup costs are very high, especially when variable setup costs are proportionally very high. This can be explained by the fact that as the setup costs are high, it becomes beneficial to consolidate demand of different nodes in order to reduce the total installed capacity. In Figure 9, we show two cases where variable setup costs are comparatively much higher than other costs. The first set of figures shows a consolidated path 22–21–20–15–14–13 efficiently connecting nodes on the path and one edge away. The path is also part of a loop (see later). This is better according to the out-of-sample evaluation, even though most individual demand nodes now have longer paths to the source nodes. Here it is possible, for example, to use free capacity available on the path to reach demand nodes 16, 18 and 8 from source node 22.

In the second set of figures we see that demand nodes 20 and 16 are connected with source node 12 via transshipment node 1 instead of the shorter (in terms of fixed and variable setup costs) path via node 10, as in the deterministic design. If we look at the stochastic design in more detail, we observe that edge 1–13 has 200 units of capacity

installed, while 1–7 and 7–20 have 388 units of capacity. These add to 588 units of capacity usable for source node 12. But only 421 units of capacity are installed on edge 12–1. This makes sense since by consolidating the flow from source node 12 to nodes 20 and 16 with the flow to node 13 (and onward), the capacity into node 1 can be set 167 units lower than the outgoing capacity (this statement makes sense even though the edges are not directed). Thus, consolidating flow saves costs in total even though the paths used may be longer and costlier than in the deterministic case.

But this feature may not be attractive when the setup costs becomes low i.e., when flow costs matter much for the optimal design. Then sharing does not create sufficient savings as flow costs more than offset the savings in setup costs.

Loops

The stochastic design sometimes has loops. That will never happen in the deterministic case as long as there are no effective upper bounds on edge capacities. Two types of loops are seen, one where the source node(s) are part of them and the other is where the loops are formed with nodes excluding source nodes. In Figure 10 we see both types of loops. In the second chart of the figure, we see that loops 3–5–4–2–3, 11–12–7–9–11, and 11–16–14–17–7–9–11 are formed having a source node in them. The third chart shows a loop 16–13–15–18–14–16 without a source node in it.

The first kind of loop takes advantage of free capacity available in one of the arms of the deterministic skeleton to fulfill demand of some demand node lying on a different arm. An extra edge, connecting the two arms, makes a loop and helps satisfy demand. For the second chart of the figure, demand node 4 has a maximal demand of 1803 in one of the scenarios, whereas the path serving it (edges connecting 3, 2, and 4) has 1421 units of capacity. Here, the free capacity available on the path 3–5–12 is utilized by building an extra edge 5–4 to form a loop to serve most of this higher demands of node 4. In the third chart of the figure, we see a loop created by adding an extra edge 16–14 to the deterministic solution. Here the loop is formed to fulfill higher demand scenarios of node 14. This loop provides more supply to demand node 14 than the capacity we see in path 11–16–14, as extra capacity of the path 16–13–15 is utilized to serve node 14.

And if we look back at the second network in Figure 7, we see that node 28, a leaf in the deterministic design, gets connected to the path 12–13–29–25 by an edge 28–29 to form a loop. This is because it is cheaper to add the extra edge 28–29 with some capacity than increase capacities all the way on the paths 12–6–23–17–28 and 12–13–29 to satisfy demand of nodes 28 and 29 respectively. This edge is valuable as it can be use in both directions, to supplement demand needs of nodes 28 and 29.

Negative correlations

A very basic hedging principle is seen when source and sink nodes are linked. This principle does not show in the deterministic solutions. A source node is typically linked with demand nodes with negatively correlated demands, so that variation in demand can be utilized. In Figure 11, we see that in the mixed correlation case demand node 16 is connected to source node 10 which is also supplying demand node 11. This is because demand nodes 11 and 16 have negatively correlated demands. In the other cases, demand

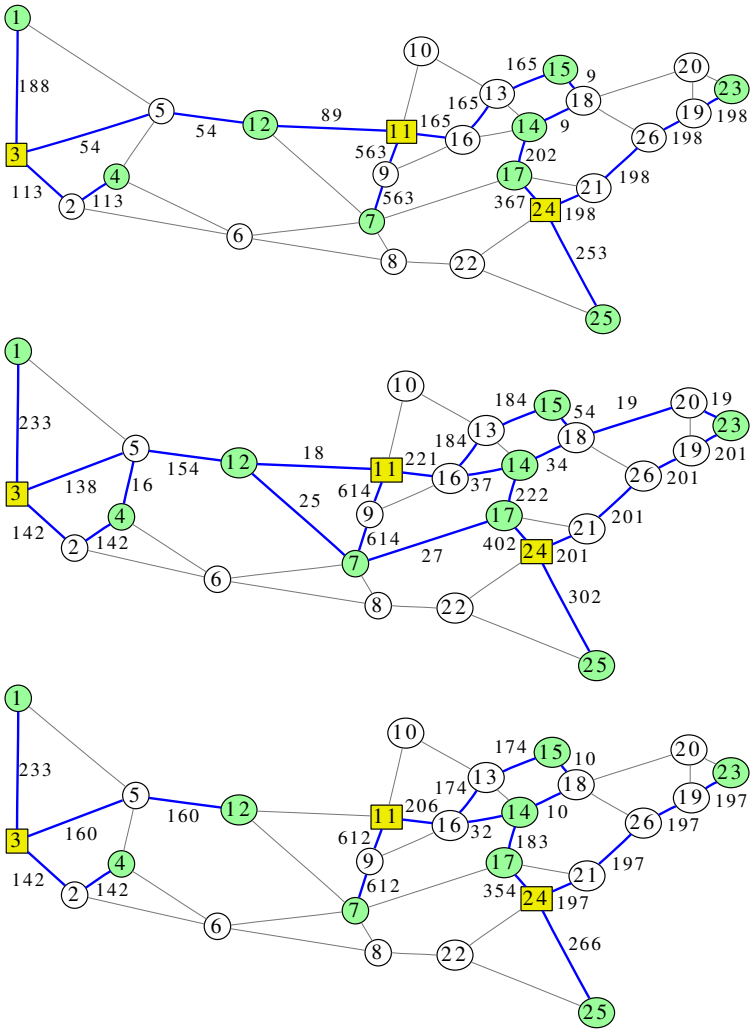

Figure 10: Loop formation in stochastic solution. Deterministic (top), uncorrelated (middle) and positively correlated (bottom) stochastic solutions of US_02 showing that the stochastic solution structures have loops which are not in the deterministic solutions. Note that the values in the figures are one-tenth of actual.

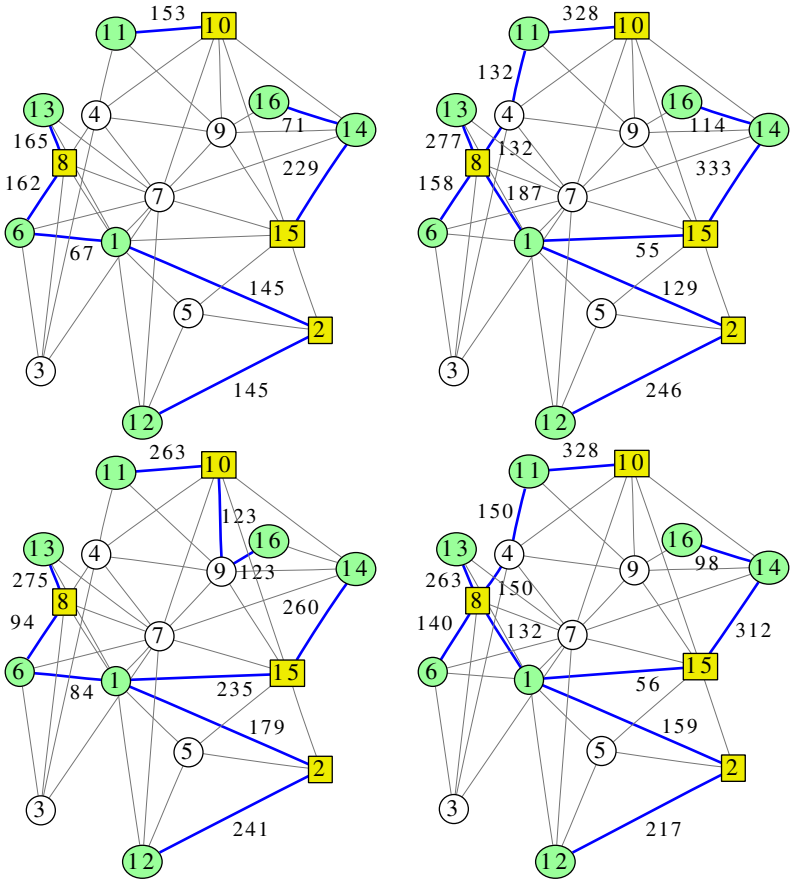

Figure 11: Pairing of negatively correlated demand nodes. Deterministic (top left) and three correlated cases of stochastic solutions–zero (top right), mixed (bottom left) and positive (bottom right)–of NY_04 test case showing that in the stochastic solution negatively correlated demand nodes 11 and 16 get connected to same source node.

node 16 is served by source node 15. Similarly, we observe that positively correlated demand nodes are disconnected. Refer to the bottom-right chart of Figure 9 of Nobel_01 mixed correlated case where positively correlated demand nodes 20 and 24 are no longer connected the way they were in the uncorrelated case.

4 Conclusion

The purpose of this paper has been to understand what constitutes a good robust design for a single-commodity stochastic network design problem with multiple sources and sinks. This paper discusses only randomness in demand.

We observe that the deterministic solution can be very bad with respect to expected behavior. But still we see certain structural patterns re-emerging in the stochastic solutions. First we observe that the deterministic solution behaves worse in the stochastic environment as the number of source nodes increases. With many source node, the deterministic solution, which is a forest, typically of many trees, decreases the ability to

share capacity in a stochastic environment. Also, as correlations are of no concern in a deterministic setting, the assignment of demand nodes to source nodes may be rather far off what is optimal.

If the variation in demand is moderate or low, the deterministic skeleton (being a forest) can be used to carry a major portion of the flow, needing very few additional edges in the stochastic environment. As the variation increases, the source nodes will have insufficient capacity to fulfill demand using the deterministic skeleton (irrespective of installed capacities), and hence will need more edges. Therefore, a re-alignment of distribution patterns will emerge. However, when the fixed and variable setup costs are low compared to the flow costs, the deterministic skeleton is contained in the stochastic one, with only few extra edges added, even in cases of higher variation in demand.

For uncorrelated and positively correlated demands existing far from the source nodes, we keep the portion of the deterministic skeleton that contains paths leading up to clusters of demand nodes. However, within a cluster of demand nodes we observe changes from that of the deterministic solution, in order to benefit from demand variations.

With high variable setup cost, we see consolidation of capacities in paths reaching downstream demand nodes. These paths will emerge more so between demand nodes which are negatively correlated. However, with increasing proportion of flow cost in deciding the optimal solution, this consolidation will be weaker.

Networks with all types of possible correlations among demands show loops in the stochastic solution. The loop formation gets stronger with increasing variable setup costs.

Source nodes choose demands to serve according to the possibility for hedging among them. Hence, everything else being equal, negatively correlated demand nodes are most likely to be served from the same source node. This also results in the breaking or weakening of links between positively correlated demand nodes relative to the deterministic solution, which has no such concerns.

In total, the main observations of optimal designs are therefore as follows. Note that the observations are connected, and to some extent see the same phenomena from different perspectives.

- Especially with high setup costs, the deterministic design tends to be a forest of small trees. This is particularly bad in a stochastic environment. The good robust designs will contain many more connections than the deterministic counterpart, and the design will contain loops.

- From a demand node perspective: When negative correlations in demand are present, the design should be such that nodes with negatively correlated demands share paths to one or more source nodes. If there are no negative correlations, it is still important to utilize variation in demand to reduce investments by looking for as small (albeit positive) correlations as possible.

- From a source node perspective: If supply capacity is limited, a source node needs to be connected to several demand nodes, preferably nodes with as small correlations (negative if feasible) in demand as possible, so as to be able to use its supply capacity well in all scenarios.

- Especially with high setups costs and source and demand nodes concentrated in different areas, it is good to build a high capacity path – a highway – from the area of the source nodes to the area of the demand nodes. The source and demand nodes are then connected to the highway using the principles of the previous two items.

- A single loop may be seen as two paths plus a crossover edge (or path). The crossover must be placed such that demand along either the two upstream or the two downstream sub-paths are negatively correlated. If no negative correlations are possible, the same correlations should be as small as possible. This will reduce the overall investments. The same logic applies if the crossover edge (path) connects two disjoint paths.

Future work. We plan to follow up this work by studying the case of random arc capacities.

Acknowledgments

This project was supported in part by grant 171007/V30 from The Research Council of Norway. While working on this project, T.G. Crainic was the NSERC Industrial Research Chair on Logistics Management, ESG UQAM, and Adjunct Professor with the Department of Computer Science and Operations Research, Université de Montréal, and the Department of Economics and Business Administration, Molde University College, Norway. Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grants programs.

References

R.K. Ahuja, T.L. Magnanti, J.B. Orlin, and M.R. Reddy. Applications of networks optimization. In M. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, pages 1–83. North-Holland, Amsterdam, 1995.

M. Ball, C. Barnhart, G. Nemhauser, and A. Odini. Air transportation: Irregular operations and control. In G. Laporte and C. Barnhart, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*. Elsevier Publishing, 2007.

J. R. Birge. The value of the stochastic solution in stochastic linear programming with fixed recourse. *Mathematical Programming*, 24:314–325, 1982.

Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software: Practice and Experience*, 30(11):1203–1233, 2000. doi: 10.1002/1097-024X(200009)30:11⟨1203::AID-SPE338⟩3.0.CO;2-N.

J. L. Higle and S. W. Wallace. Sensitivity analysis and uncertainty in linear programming. *Interfaces*, 33:53–60, 2003.

K. Høyland, M. Kaut, and S.W. Wallace. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24(2–3):169–185, 2003.

ILOG CPLEX 9.0 User's Manual. ILOG, S. A., 2003.

Michal Kaut and Stein W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257–271, 2007.

A.-G. Lium, T. G. Crainic, and S. W. Wallace. Correlations in stochastic programming: A case from stochastic service network design. *Asia-Pacific Journal of Operational Research*, 24(2):161–179, 2007.

A.-G. Lium, T. G. Crainic, and S. W. Wallace. A study of demand stochasticity in stochastic network design. *Transportation Science*, 43(2):144–157, 2009.

S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0 – Survivable Network Design library. *Networks*, Early view, 2009. doi: 10.1002/net.20371.

Kevin P. Scheibe and Cliff T. Ragsdale. A model for the capacitated, hop-constrained, per-packet wireless mesh network design problem. *European Journal of Operational Research*, 197(2):773–784, 2009. doi: 10.1016/j.ejor.2008.07.020.

Biju K. Thapalia, Teodor G. Crainic, Michal Kaut, and Stein W. Wallace. Single source single-commodity stochastic network design. Feb 2009.

Stein W. Wallace. Decision making under uncertainty: Is sensitivity analysis of any use? *Operations Research*, 48(1):20–25, 2000.

Stein W. Wallace. Stochastic programming and the option of doing it differently. *Annals of Operations Research*, page to appear, 2009.

Joyce W. Yen and John R. Birge. A stochastic programming approach to the airline crew scheduling problem. *Transportation Science*, 40(1):3–14, 2006.

A Results of the numerical tests

This appendix provides detailed results from the tests in Section 3. Table 2 provides the numbers used to generate Figure 3, in Section 3.3. The analysis in Section 3.4 is also based on these computations, but the individual cases cannot be reproduced from these tables. Tables 3 to 5 present the full computational results for Figure 1, while those of Figure 5 are found in Tables 6 to 8.

Table 2: The numbers corresponding to Figure 3

	Fixed setup cost			Variable setup cost		
	A	B	C	A	B	C
Minimum value	0.54	0.54	0.91	0.56	0.76	0.96
Geometric mean	0.74	0.74	1.00	0.62	0.95	1.03
Maximum value	0.93	0.93	1.14	0.68	1.09	1.12

Table 3: Results of Comparison A corresponding to Figure 1, split by correlation structure.

	Deterministic solution			**Stochastic solution**		
Test Name	$\rho = \mathbf{0}$	$\rho > \mathbf{0}$	$\rho \gtrless \mathbf{0}$	$\rho = \mathbf{0}$	$\rho > \mathbf{0}$	$\rho \gtrless \mathbf{0}$
Germany_01	17165	18525	17504	13410	14838	16038
Germany_02	18077	18749	18047	13774	15728	17210
Germany_04	17372	18048	17711	12182	13569	14822
Molde_01	89318	89814	89844	6968	21110	6794
Molde_02	90920	90791	90831	7470	21767	7032
Molde_04	90746	90618	90657	6987	21248	6769
Montreal_r06.1_01	157972	157974	157969	112619	111764	113638
Montreal_r06.1_02	147886	147892	147884	102292	104391	102577
Montreal_r10.1_01	116165	117029	115614	103571	105251	102774
Montreal_r10.1_02	80913	81512	81325	59662	61636	59078
Montreal_r10.1_03	92081	92666	92496	71747	73474	71545
Nobel-EU_01	1209410	1332870	1237110	128951	192455	123499
Nobel-EU_02	1269300	1339410	1243730	110903	172271	108688
Nobel-EU_03	1262480	1332590	1236910	100219	163244	97482
NY_01	24860500	26023300	26012800	1851100	3366850	1786500
NY_02	25761800	26256300	26239100	1726640	3423220	1545050
NY_04	24693000	25856600	25846100	1629720	3268480	1553470
US_01	414391	414559	414378	371013	388984	360429
US_02	348460	353104	345384	314188	333334	303797
US_04	394493	400384	390647	354348	373575	345213

Table 4: Results of Comparison B corresponding to Figure 1, split by correlation structure.

	Deterministic solution			**Stochastic solution**		
Test Name	$\rho = \mathbf{0}$	$\rho > \mathbf{0}$	$\rho \gtrless \mathbf{0}$	$\rho = \mathbf{0}$	$\rho > \mathbf{0}$	$\rho \gtrless \mathbf{0}$
Germany_01	13772	16130	13552	13388	14812	13077
Germany_02	14246	15922	13565	13732	15710	13190
Germany_04	12448	13566	12645	12139	13556	12213
Molde_01	28210	63165	17824	6449	20898	6156
Molde_02	28727	63623	18177	7046	21340	6574
Molde_04	28643	63527	18102	6595	21088	6278
Montreal_r06.1_01	118847	121655	116083	101285	105398	95922
Montreal_r06.1_02	108009	110727	105391	87818	93341	85182
Montreal_r10.1_01	104465	108639	103366	103536	105220	102748
Montreal_r10.1_02	62477	66216	60969	59605	61577	59002
Montreal_r10.1_03	74936	78331	73921	71676	73418	71505
Nobel-EU_01	163710	517543	117573	122131	189166	116968
Nobel-EU_02	204476	305544	130275	105638	168372	103345
Nobel-EU_03	300252	639796	143485	96720	160822	94153
NY_01	5813210	10931900	5899210	1675610	3281980	1700720
NY_02	3850040	8315310	2995700	1629340	3225060	1501700
NY_04	5586750	10731400	5691810	1516610	3128730	1483130
US_01	379657	389969	373204	369994	388887	360025
US_02	318592	334225	308236	313412	333110	303356
US_04	358255	373970	349697	353351	373159	344944

Table 5: Results of Comparison C corresponding to Figure 1, split by correlation structure.

	Deterministic solution			**Stochastic solution**		
Test Name	$\rho = 0$	$\rho > 0$	$\rho \gtrless 0$	$\rho = 0$	$\rho > 0$	$\rho \gtrless 0$
Germany_01	13588	15807	13180	13406	14836	13101
Germany_02	13752	16053	13172	13751	15717	13186
Germany_04	12114	13684	11968	12170	13563	11992
Molde_01	6582	21104	6175	6520	20911	6216
Molde_02	7168	21616	6667	7034	21352	6607
Molde_04	6908	21491	6380	6606	21076	6321
Montreal_r06.1_01	97268	102469	94633	97280	102753	94745
Montreal_r06.1_02	86172	91202	84293	86232	91237	84299
Montreal_r10.1_01	107376	109207	106848	103546	105228	102765
Montreal_r10.1_02	64556	66634	64130	59659	61593	59018
Montreal_r10.1_03	71833	73624	71604	71718	73431	71534
Nobel-EU_01	124432	191719	117573	125304	190475	116983
Nobel-EU_02	106971	170755	103847	105794	169633	103365
Nobel-EU_03	101393	165388	98930	97405	162022	94310
NY_01	1738870	3326650	1761180	1676470	3286030	1705290
NY_02	1632510	3246760	1534730	1629740	3225230	1501700
NY_04	1524290	3143840	1472440	1521280	3128920	1483130
US_01	377252	401183	366298	370814	388883	360311
US_02	317048	341646	306368	313987	333100	303691
US_04	353590	373386	345223	353987	373186	345195

Table 6: The ratios corresponding to Figure 5 split by correlation structure.

		Comparison A			**Comparison B**			**Comparison C**		
Test Name	$\mathbf{g_k/C_k}$	$\rho = 0$	$\rho > 0$	$\rho \gtrless 0$	$\rho = 0$	$\rho > 0$	$\rho \gtrless 0$	$\rho = 0$	$\rho > 0$	$\rho \gtrless 0$
Germany_01	0.001	1.32	1.29	1.36	1.04	1.01	1.08	1.01	1.06	1.01
	0.05	1.27	1.27	1.33	1.02	1.09	1.03	1.01	1.07	1.01
	0.25	1.30	1.25	1.33	1.05	1.06	1.12	1.02	1.03	1.01
	0.5	1.33	1.29	1.35	1.07	1.07	1.15	1.00	1.00	1.00
	0.999	1.40	1.35	1.39	1.03	1.01	1.00	1.00	1.00	1.00
Germany_02	0.001	1.37	1.23	1.42	1.07	1.02	1.09	1.00	1.03	1.00
	0.05	1.32	1.20	1.36	1.03	1.01	1.03	1.00	1.03	1.00
	0.25	1.31	1.24	1.38	1.01	1.04	1.05	1.00	1.01	1.00
	0.5	1.31	1.26	1.34	1.03	1.06	1.09	1.01	1.03	1.01
	0.999	1.37	1.33	1.37	1.03	1.01	1.00	1.00	1.00	1.00
Germany_04	0.001	1.47	1.36	1.49	1.05	1.01	1.07	1.00	1.01	1.00
	0.05	1.39	1.34	1.45	1.05	1.11	1.11	1.00	1.02	1.00
	0.25	1.39	1.34	1.43	1.04	1.10	1.07	1.01	1.03	1.01
	0.5	1.40	1.35	1.44	1.02	1.09	1.05	1.02	1.04	1.03
	0.999	1.45	1.38	1.45	1.03	1.01	1.00	1.00	1.00	1.00

Table 7: Ratios corresponding to Figure 5 split by correlation structure, cont. The asterisk (*) denotes cases where the solver did not finished within 15 days.

		Comparison A			**Comparison B**			**Comparison C**		
Test Name	$\mathbf{g_k/C_k}$	$\rho = \mathbf{0}$	$\rho > \mathbf{0}$	$\rho \gtrless \mathbf{0}$	$\rho = \mathbf{0}$	$\rho > \mathbf{0}$	$\rho \gtrless \mathbf{0}$	$\rho = \mathbf{0}$	$\rho > \mathbf{0}$	$\rho \gtrless \mathbf{0}$
Molde_01	0.001	16.10	4.57	16.11	2.19	2.01	3.71	1.04	1.01	1.01
	0.05	13.69	4.41	14.38	2.53	1.11	1.79	1.02	1.01	1.01
	0.25	12.04	4.25	12.06	4.16	1.40	3.00	1.01	1.00	1.00
	0.5	10.45	4.04	10.55	3.66	1.35	2.67	1.01	1.00	1.00
	0.999	8.52	3.72	9.14	3.05	1.27	2.37	1.00	1.00	1.02
Molde_02	0.001	14.96	4.50	15.10	3.16	3.18	4.77	1.05	1.02	1.02
	0.05	13.67	4.40	14.03	2.84	3.10	4.44	1.03	1.02	1.02
	0.25	11.49	4.19	11.90	2.45	2.96	3.82	1.03	1.02	1.03
	0.5	9.22	3.91	9.64	1.93	1.33	1.86	1.02	1.00	1.04
	0.999	8.02	*	*	3.02	*	*	1.00	*	*
Molde_04	0.001	15.92	4.55	15.87	3.27	3.20	5.01	1.08	1.03	1.02
	0.05	14.60	4.46	14.57	3.03	3.14	4.62	1.07	1.03	1.03
	0.25	11.84	4.20	11.93	4.13	1.39	3.01	1.02	1.00	1.04
	0.5	10.15	3.99	10.44	3.59	1.34	2.68	1.02	1.00	1.05
	0.999	8.17	3.68	8.05	2.30	1.33	2.41	1.03	1.01	1.04
Montreal_r06.1_01	0.001	1.40	1.36	1.43	1.05	1.05	1.02	1.00	1.00	1.00
	0.05	1.39	1.36	1.43	1.04	1.04	1.01	1.00	1.00	1.00
	0.25	1.39	1.35	1.42	1.03	1.03	1.01	1.00	1.00	1.00
	0.5	1.38	1.34	1.42	1.02	1.03	1.00	1.00	1.00	1.00
	0.999	1.38	1.34	1.44	1.01	1.02	1.00	1.00	1.00	1.00
Montreal_r06.1_02	0.001	1.50	1.46	1.53	1.06	1.06	1.02	1.00	1.00	1.00
	0.05	1.50	1.45	1.53	1.05	1.06	1.02	1.00	1.00	1.00
	0.25	1.49	1.45	1.52	1.04	1.05	1.01	1.00	1.00	1.00
	0.5	1.48	1.44	1.52	1.03	1.04	1.00	1.00	1.00	1.00
	0.999	1.48	1.43	1.53	1.02	1.03	1.00	1.00	1.00	1.00
Montreal_r10.1_01	0.001	1.14	1.13	1.14	1.03	1.05	1.02	1.00	1.00	1.00
	0.05	1.13	1.12	1.13	1.02	1.03	1.01	1.00	1.00	1.00
	0.25	1.13	1.12	1.13	1.02	1.02	1.00	1.00	1.00	1.00
	0.5	1.13	1.12	1.14	1.02	1.02	1.00	1.00	1.01	1.00
	0.999	1.15	1.14	1.15	1.01	1.01	1.00	1.00	1.00	1.00
Montreal_r10.1_02	0.001	1.35	1.32	1.36	1.08	1.10	1.06	1.00	1.00	1.00
	0.05	1.34	1.31	1.36	1.03	1.03	1.02	1.00	1.00	1.00
	0.25	1.33	1.30	1.35	1.02	1.02	1.01	1.00	1.00	1.00
	0.5	1.33	1.30	1.35	1.01	1.02	1.00	1.00	1.00	1.00
	0.999	1.34	1.31	1.36	1.00	1.01	1.00	1.00	1.00	1.00
Montreal_r10.1_03	0.001	1.33	1.30	1.33	1.02	1.02	1.01	1.00	1.00	1.00
	0.05	1.33	1.30	1.33	1.03	1.02	1.03	1.00	1.00	1.00
	0.25	1.32	1.29	1.32	1.03	1.03	1.03	1.00	1.00	1.00
	0.5	1.31	1.29	1.31	1.02	1.03	1.02	1.00	1.00	1.00
	0.999	1.31	1.29	1.31	1.02	1.02	1.01	1.00	1.00	1.00
Nobel-EU_01	0.001	12.50	9.02	14.59	1.21	1.51	1.36	1.02	1.01	1.01
	0.05	11.24	8.58	13.58	1.16	1.26	1.11	1.02	1.01	1.00
	0.25	9.89	7.88	11.04	1.35	3.22	1.00	1.02	1.02	1.00
	0.5	8.58	7.43	10.12	4.28	6.81	3.00	1.03	1.02	1.07
	0.999	7.61	6.81	9.11	3.80	6.23	2.74	1.01	1.00	1.11

Table 8: The remaining ratios corresponding to Figure 5 split by correlation structure

		Comparison A			**Comparison B**			**Comparison C**		
Test Name	$\mathbf{g_k/C_k}$	$\rho = \mathbf{0}$	$\rho > \mathbf{0}$	$\rho \gtrless \mathbf{0}$	$\rho = \mathbf{0}$	$\rho > \mathbf{0}$	$\rho \gtrless \mathbf{0}$	$\rho = \mathbf{0}$	$\rho > \mathbf{0}$	$\rho \gtrless \mathbf{0}$
Nobel-EU_02	0.001	16.17	10.45	16.01	3.70	4.95	4.99	1.02	1.01	1.01
	0.05	14.74	10.03	14.85	3.06	4.20	1.50	1.01	1.00	1.00
	0.25	12.52	9.13	12.87	2.55	2.40	1.39	1.02	1.01	1.00
	0.5	10.83	8.33	11.58	2.24	2.21	1.30	1.00	1.00	1.00
	0.999	9.34	7.51	9.89	1.98	2.03	1.18	1.00	1.00	1.00
Nobel-EU_03	0.001	16.68	10.78	16.95	4.41	5.59	6.19	1.01	1.01	1.01
	0.05	15.68	10.41	15.96	3.19	4.32	1.56	1.00	1.00	1.02
	0.25	13.35	9.52	13.90	2.66	2.46	1.43	1.04	1.02	1.02
	0.5	11.65	8.74	12.28	2.35	2.28	1.31	1.03	1.02	1.01
	0.999	9.82	7.72	10.27	2.04	2.06	1.17	1.00	1.01	1.01
NY_01	0.001	22.52	8.73	23.11	5.17	3.65	5.24	1.03	1.01	1.01
	0.05	20.55	8.45	21.39	4.74	3.54	4.86	1.03	1.01	1.01
	0.25	17.71	7.97	17.91	4.13	3.36	4.13	1.03	1.02	1.03
	0.5	14.97	7.46	16.11	6.02	3.70	7.95	1.00	1.00	1.01
	0.999	12.58	6.77	14.13	5.22	3.41	7.08	1.02	1.01	1.02
NY_02	0.001	25.80	9.14	25.09	4.73	4.46	1.67	1.05	1.02	1.00
	0.05	23.63	8.88	23.11	6.96	4.55	2.98	1.06	1.02	1.01
	0.25	18.79	8.21	20.10	4.81	2.70	3.04	1.03	1.01	1.01
	0.5	16.44	7.66	17.71	2.63	2.66	2.19	1.01	1.01	1.03
	0.999	14.12	7.02	15.10	5.53	3.48	7.47	1.01	1.00	1.02
NY_04	0.001	27.91	9.34	26.74	5.95	3.81	5.66	1.02	1.01	1.03
	0.05	25.38	9.07	24.78	5.43	3.71	5.26	1.01	1.00	1.02
	0.25	20.58	8.38	20.73	4.55	3.50	4.69	1.03	1.01	1.05
	0.5	17.75	7.82	18.24	3.98	3.29	4.18	1.00	1.00	1.03
	0.999	14.73	7.10	15.61	3.55	3.04	3.69	1.00	1.00	1.00
US_01	0.001	1.12	1.06	1.15	1.03	1.00	1.03	1.02	1.03	1.02
	0.05	1.09	1.06	1.13	1.01	1.00	1.02	1.00	1.00	1.00
	0.25	1.08	1.06	1.11	1.00	1.00	1.00	1.00	1.00	1.00
	0.5	1.09	1.07	1.12	1.00	1.00	1.00	1.00	1.00	1.00
	0.999	1.15	1.11	1.16	1.03	1.02	1.07	1.03	1.02	1.07
US_02	0.001	1.11	1.05	1.14	1.02	1.00	1.01	1.01	1.02	1.01
	0.05	1.11	1.07	1.13	1.02	1.01	1.01	1.01	1.03	1.01
	0.25	1.10	1.07	1.13	1.00	1.00	1.00	1.00	1.00	1.00
	0.5	1.11	1.07	1.13	1.00	1.00	1.00	1.00	1.00	1.00
	0.999	1.14	1.10	1.17	1.00	1.00	1.00	1.00	1.00	1.00
US_04	0.001	1.11	1.07	1.13	1.02	1.00	1.01	1.00	1.00	1.00
	0.05	1.08	1.07	1.11	1.00	1.00	1.00	1.00	1.00	1.00
	0.25	1.09	1.07	1.12	1.00	1.00	1.00	1.00	1.00	1.00
	0.5	1.12	1.08	1.15	1.02	1.00	1.04	1.02	1.00	1.02
	0.999	1.16	1.13	1.18	1.00	1.00	1.00	1.00	1.00	1.00

Paper 4

Single-Commodity Network Design with random edge capacities

Single-Commodity Network Design with random edge capacities

Biju K. Thapalia* Teodor Gabriel Crainic† Michal Kaut‡

Stein W. Wallace§

01 February 2010

Abstract

This paper examines the single-commodity network design problem with stochastic edge capacities. We characterize the structures of the optimal designs and compare with the deterministic counterparts. We do this primarily to understand what constitutes robust network designs, but hope that the results can be used also to develop better heuristics than those available today.

Keywords: Single-commodity network design, Survivable networks, Edge failure, Correlations, Robustness

1 Introduction

Many important physical networks, such as distribution networks for water, oil and gas pipelines, road system, or distribution channels are integral parts of our lives. These networks are made to last for a long time and are often subjected to daily routine operational decisions. If any parts of these networks are down, major portion of society will be affected. Focus on cost savings tends to make these networks sparser, and hence also more vulnerable to any kind of disruption, failure, maintenance, congestion, etc that may occur (see discussions in Ball, Colbourn, and Provan (1995) and Balakrishnan, Magnanti, and Mirchandani (1998)). The owners of these networks must therefore design and maintain them, often under strict budgetary regimes, so that they work well even in the case of reduced capacities or broken links in the network. Hence, in our view, there is an increased need to understand what constitutes a good design in light of random capacities. In particular, we wish to see if it is important to use models explicitly expressing the randomness in capacities when designing the networks. And if the answer is yes, we would like to understand in what ways the designs from deterministic design models fall short of designs from models explicitly considering the uncertainty.

*Molde University College, biju.k.thapalia@himolde.no

†University of Quebec at Montreal

‡Molde University College

§Lancaster University Management School

Two terms which are used frequently in case of network failures are reliability and survivability. Reliability is the probability that a network functions according to a specification. Survivability is the ability of a network to perform according to a specification after it has been damaged. Reliability is a measure of the service provided by the network and survivability is a measure of the network itself. Hence, higher level of reliability depends upon higher degree of survivability.

Good robust designs trade off deterministic initial costs versus expected future costs in a good way. Most often, by increasing initial costs, the network is provided with more operational flexibility, and expected future costs decrease. Optimality is achieved when additional initial investments cost more than what is gained operationally. (Note though, that at times uncertainty induces lower initial investments. This may for example take the form of postponements of decisions.) The best way to have a high degree of survivability in a network is to have many links between the nodes of the network, i.e., a dense network, but this increases the cost to built the network. So it is expensive to built a network with a high level of survivability, but of course, the operational costs in light of disturbances will decrease. More important to this paper, though, is that many networks, having the same initial investment costs, may react very differently to disturbances. So even though it is clear that many links will increase the quality of a network, we would like to understand what characterizes a good way to increase the number of links. And, again, we would like to understand if deterministic models will guide us well in designing the networks, or if they will lead us astray.

Stochastics in a network may arise both in terms of supply/demand and edge capacities. The latter, which is what we discuss in this paper, may be in the form of edges being on/off or in terms of capacities being a random fraction of the maximal capacity, representing such as broken cables and damaged roads. The on/off situation may cover both capacitated and uncapacitated edges. Clearly, for capacitated edges, the on/off situation is a special case of the random capacity case where there are only two possibilities, no capacity or full capacity. The latter is sometimes referred to as a binary state system, while with many possible capacities it is referred to as a multi-state system. For more information about the multi-state system refer to the book by Lisnianski and Levitin (2003). We consider the case of capacitated edges with random edge capacities, and hence includes also the on/off capacitated case.

There are many application areas where network design with random edge capacities are important. Possibly the most famous one is the design of survivable networks in telecommunications. Much of this work originates with Suurballe and Tarjan (1984). Refer to Balakrishnan et al. (1998), Clarke and Anandalingam (1995) and Myung, Hyunjoon., and Dong-wan. (1999) for a more detailed understanding and many variations and extensions. Water, oil, and gas distribution system are other central cases. If we drop the pressure constraints, these pipeline design problems simplify and can be expressed as single commodity design problems with underlying linear single commodity flow problems (Brimberg, Hansen, Lih, Maldenović, and Breton, 2003).

A water pipeline network spans different consumers, of which some are very sensitive to disruption in supply, such as hospitals and certain industries. Disruption in the supply arise due to pipeline ruptures, leakages or blockage, which affect the overall flow in the network. Many studies are done on designing optimal and reliable water distribu-

tion systems (Chung, Lansey, and Bayraksan, 2009, di Pierro, Khu, Savić, and Berardi, 2009, Montalvo, Izquíerdo, Pérez, and Tung, 2008). A seminal paper in gas distribution is by Rothfarb, Frank, Rosenbaum, Steiglitz, and Kleitman (1970). The papers by Wolf and Smeers (1996), Martin, Möller, and Moritz (2006) and Brimberg et al. (2003) show the design issues of oil\gas distribution networks. The work by Midthun, Bjørndal, and Tomasgard (2009) shows how the network structure and the physical properties effects the operation and development of natural gas transportation networks. Logistics network design with robustness consideration (Meepetchdee and Shah, 2007) is another application area. The paper by Chen, Yang, Lo, and Tang (1999) discuss the reliability of transport networks with random link capacity. Many applications are also seen in network interdiction problems (Cormican, Morton, and Wood, 1998, Ramirez-Marquez and Rocco S., 2009, Smith, Lim, and Sudargho, 2007), as well as the protection of network systems from natural disasters. Examples of the latter are retrofit of highway bridges for increased robustness of the transportation system (Liu, Fan, and Ordonez, 2009), and evacuation planning for emergencies (Andreas and Smith, 2009).

In their work on fleet management (see for example Cheung and Powell (1996)) Powell and his co-authors use random edge capacities to represent random demand. Bounds on the recourse problem in this situation is discussed in Wallace (1987).

It is evident that a network with random edge capacities must function reasonably well in many situations with partial or full breakdown of capacities. A common way to investigate this situation is to perform single- or multi-parameter sensitivity analysis in order to understand how the optimal solution changes as a function of these breakdowns. This approach might seem appropriate, but in fact it is not. This is outlined in detail in Wallace (2000) and Higle and Wallace (2003). Logically, when performing sensitivity analysis one is assuming that the design can be postponed until after breakdowns have become known. This is hardly ever pointed out, though. So, whether sensitivity analysis is performed or not, one ends up with a solution not created for robustness, and hence, may have to face difficult operational decisions when breakdowns occur.

Much of the literature in the field of survivable networks discuss different heuristics but do not address the resulting network structure. We know that a deterministic solution might perform very badly when used in a stochastic environment, that is, when subjected to the uncertainties that were suppressed when the deterministic model was solved, see for example Thapalia, Crainic, Kaut, and Wallace (2009a). The reason is simply that it is not made to handle uncertainties in a good way. This paper studies the structural differences between stochastic and deterministic designs, in order to understand what flexibility means in the optimal network structure for a single commodity flow problem with single or multiple sources and sinks. We also hope that this can be used to develop heuristics for the stochastic case.

The remainder of the paper is organized as follows. Section 2 explains the problem in detail with the mathematical formulation. Section 3 explains the experimentation set-ups and scenario generation. Section 4 lists the computational results with discussions and finally Section 5 will conclude the paper.

2 Problem description and Modeling issues

Given a set of nodes (divided into source nodes, demand nodes, and transshipment nodes) and a set of potential edges connecting these nodes, the single-commodity network design problem with random edge capacities is the problem of determining a subset of the edges to open (including the edges' capacities), so as to fulfill the demand at the demand nodes at minimal cost, taking into account capacities of the source nodes and the potential failures of the edges.

The design is based on minimizing the sum of the fixed costs of selecting edges connecting the nodes; linear costs to open capacities in the edges; per unit flow costs on the edges; and per unit penalty costs for not satisfying demand. Not satisfying demand can have many interpretations, such as sending the flow at a later point in time, with another mode, or a straightforward rejection. In any case, in the model, it takes the form of a penalty cost per unit of unsatisfied demand. We find it crucial to include the possibility of not satisfying all the demand, as it is unlikely to have a network which satisfies all demand in all situations (see Thapalia et al. (2009a) for more discussion). The same formulation is used in both the stochastic and deterministic models, to make the results comparable. We view supply as a capacity, and hence, do not consider unused supply as a problem.

When we wish to compare a stochastic network design model with its deterministic counterpart, we need to be careful about how we define the deterministic model. For random demand, this is not so difficult. If historical data is available, for example, demand will usually be the average observed demand (or possibly some other forecasted demand based on the history). Hence, it is not unreasonable to compare the stochastic model with a deterministic model where all demands are replaced with their mean values.

It is not quite as easy for the case of random edge capacities. If the starting point is the stochastic model, and we ask "What is the natural deterministic counterpart?", the answer is most likely a model where edge capacities are replaced by their means. But if the starting point is that of setting up a deterministic network design model (possibly realizing that edges might fail, but not wanting to model it), it is rather likely that edges will be treated with capacities equal to their capacities when they are fully operational, that is, their maximal capacities from a stochastic perspective. One will argue: This edge costs a\$ and has a capacity of b. One will not use expected capacity taking possible failures into account. So in what follows, for each stochastic case, we shall consider two deterministic cases: average capacity and maximal (design) capacity.

We shall let all source nodes (in the case of multiple sources) have the same capacity. Hence, our assumption is that a set of demand nodes will have their demands satisfied from a set of equally-sized source nodes through edges with random capacities. We have chosen this approach to prevent our optimal designs from being affected by variations (across nodes) in a parameter which is not the primary focus of the paper. So, the first stage decisions in this problem are to decide which edges to open and what capacities to install. The second stage decisions are the flow decisions in the given network. The recourse action here is described by a penalty cost incurred for not satisfying demand.

A word of warning might be in place here. If a stochastic optimization problem, as well as its deterministic counterpart (where all random variables are replaced by their means or some other related values), use hard constraints in the formulation (for example

requiring that demand *must* be met), the optimal design from the deterministic program will normally be infeasible in the stochastic program, and hence, its expected cost be infinitely large. In our context that will be caused by the network not having enough capacity to satisfy all demand in scenarios with many edges at low capacity. On the other hand, if soft constraints are used (allowing demand to be rejected at a cost), the deterministic solution will normally be feasible in the stochastic model, but its expected performance can be made arbitrarily bad by choosing large penalties on the soft constraints (unsatisfied demand). Hence, if the *goal* is to make the deterministic solution look bad, that is easy to achieve. However, that is not our goal. So we set the penalties at reasonable levels, and our goal is not to (again) show how bad the deterministic solution is, but to *understand* its relationship to its stochastic counterpart. So, we shall certainly present numbers, and we do believe the numbers are informative. However, there will never be really objective results in this setting. There will always be a subjective element.

2.1 Mathematical formulation

Let $G = (\mathscr{N}, \mathscr{E})$ be a network defined by a set $\mathscr{N}$ of n nodes and set $\mathscr{E}$ of m edges (undirected arcs), where

$$\mathscr{E} \subset \{k = (i, j) : i \in \mathscr{N}, j \in \mathscr{N} \text{ and } i < j\}.$$

Each edge is indexed either by i, j or by k.

The randomness in the edges is described by a set of scenarios $\mathscr{S}$, where each individual scenario $s \in \mathscr{S}$ has one capacity realization for each edge. We shall discuss in Section 3.2 how the scenarios were generated. The notations for the sets, parameters, and variables associated with this problem are as follows:

Sets:

$\mathscr{C}$	set of all source nodes;
$\mathscr{D}$	set of all demand nodes;
$\mathscr{T}$	set of all nodes with zero demand (transshipment nodes); $\mathscr{T} = \mathscr{N} \setminus (\mathscr{C} \bigcup \mathscr{D})$;
$\mathscr{S}$	set of all scenarios s.

Parameters:

M	"big M";
R	unit cost of unsatisfied demand;
P^s	probability of scenario $s \in \mathscr{S}$;
C_k	flow cost on edge $k \in \mathscr{E}$;
G_k	fixed setup cost for edge $k \in \mathscr{E}$;
H_k	variable setup cost; the cost for adding one unit of capacity to edge $k \in \mathscr{E}$;
V_k	initial/ existing capacity on edge $k \in \mathscr{E}$, if any;
D_i	demand ($D_i < 0$) in node $i \in \mathscr{D}$;
D	supply in each source node, $D > 0$;
Δ_k^s	the portion of capacity on edge $k \in \mathscr{E}$ that works in scenario s.

Variables:

$x_k^s = x_{ij}^s$	flow on edge $k = (i,j) \in \mathscr{E}$ going in direction $i \to j$, in scenario $s \in \mathscr{S}$;
$z_k^s = z_{ij}^s$	flow on edge $k = (i,j) \in \mathscr{E}$ going in direction $j \to i$, in scenario $s \in \mathscr{S}$;
u_k	new capacity that is developed on edge $k \in \mathscr{E}$;
e_i^s	for $i \in \mathscr{D}$, this is the unsatisfied/lost demand in node i in scenario $s \in \mathscr{S}$;
	for $i \in \mathscr{C}$, this is the unused capacity of source node i in scenario $s \in \mathscr{S}$;
y_k	1 if edge $k \in \mathscr{E}$ is developed, 0 otherwise.

We assume that total supply, coming from equally-sized source nodes equals maximal demand in the network, so that

$$D = \{-\sum_{j \in \mathscr{D}} \{D_j\}\}/|\mathscr{C}| \tag{1}$$

where $|\mathscr{C}|$ is the number of source nodes.

Our overall problem is hence:

$$\min \sum_k G_k y_k + \sum_k H_k u_k + \sum_s P^s \left\{ \sum_k C_k \left(x_k^s + z_k^s\right) + R \sum_{i \in \mathscr{D}} e_i^s \right\} \tag{2}$$

Subject to:

$$\sum_{j:(ij) \in \mathscr{E}} \left(x_{ij}^s - z_{ij}^s\right) - \sum_{j:(ji) \in \mathscr{E}} \left(x_{ji}^s - z_{ji}^s\right) = \begin{cases} 0 & \forall i \in \mathscr{T}, \forall s \in \mathscr{S} \\ D - e_i^s & \forall i \in \mathscr{C}, \forall s \in \mathscr{S} \\ D_i + e_i^s & \forall i \in \mathscr{D}, \forall s \in \mathscr{S} \end{cases} \tag{3}$$

$$x_k^s + z_k^s \leq \Delta_k^s \left(u_k + V_k\right) \quad \forall k \in \mathscr{E} \quad \forall s \in \mathscr{S} \tag{4}$$

$$u_k \leq M y_k \qquad \forall k \tag{5}$$

$$0 \leq e_i^s \leq -D_i \qquad \forall i \in \mathscr{D}; \forall s \tag{6}$$

$$x_k^s, z_k^s, u_k, e_i^s \geq 0 \text{ and } y_k \in \{0,1\} \qquad \forall k; \forall i; \forall s \tag{7}$$

The objective function (2) minimizes the total costs of the network. The first part is the costs of constructing all new edges, the second part the costs of building all the new capacities, the third part the expected flow costs through all the edges and the fourth part is the expected penalty costs of not fulfilling demand. Constraints (3) model conservation of flow at nodes. The left-hand side is the net outflow from node i, which must be zero for all the transshipment nodes $i \in \mathscr{T}$ and is equal to the unused capacity for source node $i \in \mathscr{C}$. For the demand nodes, the net outflow must be equal to the satisfied demand; since D_i is negative in this case, the right-hand side is the a difference between the scenario demand D_i and the (positive) unsatisfied demand e_i^s.

Constraints (4) represent the flow limit in each edge. The left hand side of the equation is the net flow on the edge k which should be less then or equal to the total capacity of the edge. Since we do not start with any initial/existing capacity, we always have $V_k = 0$. Note that in an optimal solution, an edge will never have flow in both directions. Constraints (5) show that new capacity u_k can be developed only if edge k is built. Constraints (6) give bounds for the rejection amount and finally, (7) insure that all variables are non-negative and the edge constructions binary.

For the deterministic counterpart (as mentioned in Section 2) we replace the stochastic edge capacities by their expectations and their maximal values, resulting in two separate deterministic cases.

We model the problem in AMPL and solve it to optimality using CPLEX 9.0. The solution time varies from few seconds to 5 hours depending on the case, on a PC with 3 GHz Intel® CPU and 8 GB of RAM.

3 Experimentation and Scenario generation

In this section we first discuss the test cases and their sources before turning to scenario generation and the question of stability relative to the chosen scenarios. Our tests are designed to achieve two goals: Firstly, we wish to understand how deterministic designs perform in stochastic environments and to what extent information from deterministic designs are useful for the stochastic problem. Secondly, our goal with these tests is to be able to characterize the stochastic designs, so we can qualitatively describe good designs and use this knowledge to evaluate a given design without making any serious calculations.

3.1 Test instance generation

We took five different networks used in Thapalia et al. (2009a) and Thapalia, Crainic, Kaut, and Wallace (2009b). The networks named Germany, Nobel-EU, France, and Pdh are telecommunication examples from the SNDlib library (Orlowski, Pióro, Tomaszewski, and Wessäly, 2009), and Montreal_r06 is obtained from CIRRELT (Interuniversity Research Centre on Enterprise Networks, Logistics and Transport), Montreal. The names, as such, of the test networks do not mean anything in our computational setup.

In total 76 test instances are constructed using the above five networks. These test instances are created in the following way: for each of the five networks, we created single-source and multi-source test cases by selecting one or multiple source nodes. This way we created 38 test instances of which 20 are single-source and 18 are multi-source. Since we know that correlations may play important roles in the design of a network, we created positive correlated and uncorrelated cases for each instance. In the case of positive correlation, adjacent edges are given correlations of 0.5, while edges which are separated by one edge are given correlations of 0.2. Edges which are separated by two edges have a correlation of 0.1. This is a natural setting for natural calamities. Whenever one edge is hit hard, there is a chance that also nearby edges are hit, see for example Che, Yang, Lo, and Tang (2002). In this way we get 76 test instances, of which 40 are for the single-source case and 36 are for the multi-source case.

It is worth noting that all these networks are multi-commodity network design problems, so to suit our problem only some parameter values are used. We only kept the coordinates (where available) for the nodes and the fixed setup cost G_k for the edges. The values for the other parameters – variable setup costs H_k and flow costs C_k – are all chosen proportional to the Euclidean distance between the node pairs. The cost of unfulfilled demand R is derived for each test cases using some multiple of the highest value of the fixed setup cost, variable setup cost, and unit flow cost together for an edge in the network. We made sure that R is not driving the solution. The results in the first part of section 4.1 are based on test cases with these cost structures.

The Montreal test instance does not have node coordinates, so we used Graphviz (Gansner and North, 2000) to draw the graph using fixed setup cost as distance measure. The graphs of the test instances coming from Nobel-EU are planar whereas the other graphs are non-planar. A closer description of the test instances are found in Table 1.

Table 1: The different test cases. The basic names are kept as in the source, even though the cases are adjusted to our needs.

Problem name	# nodes	# edges	# demand nodes	# sources	# test instances
Germany_SS	29	48	10	1	4
Germany_MS	29	48	10	3	4
France_SS	16	30	10	1	4
France_MS	16	30	10	3	3
Montreal_r06.1_SS	10	38	5	1	4
Montreal_r06.1_MS	10	38	7	3	3
Pdh_SS	11	30	7	1	4
Pdh_MS	11	30	7	4	4
Nobel-EU_SS	28	38	8	1	4
Nobel-EU_MS	28	38	8	4	4

In the case of single-source networks, we selected four potential source nodes. When one of them is source node, the rest are transshipment nodes. And for the cases of multi-source networks we selected four (or three for Montreal_r06 and France) sets of source nodes to make four (or three for Montreal_r06 and France) test instances from each network. The number of source nodes for each case is listed in the fifth column of Table 1. In the pictures that will follow, the distance between two nodes reflects not only the actual distance, but also the levels of the variable setup costs and flow costs. If an edge is twice as long as another, it is also twice as costly with respect to these two costs.

Given the difficulty of solving the stochastic network design problem to optimality we kept n (the number of nodes) below 30 and m (the number of edges) below 50 for the cases.

3.2 Scenario generation and Stability test

Stochastic programs need discrete probability distributions. A scenario is a vector of length m containing a possible capacity for each edge. We have created scenarios with

equal probabilities of occurring, using a variant of the moment-matching method from Høyland, Kaut, and Wallace (2003).

Lacking specific knowledge, we have chosen a triangular distribution on the $[0, 1]$ interval, with mode at one, which gives an expected value of 0.67. Note that a mode below one would imply that the edge (in the continuous case) has an extremely low probability of being close to fully operational. We feel the chosen distribution is a reasonable description of edge failures.

The decision on the number of scenarios used to represent the stochastics is critical as we want to be sure we study the effects of randomness on our model, and not some random side-effect of the scenario generating procedure. For a given scenario generation procedure, there is normally a trade-off between the number of scenarios representing the underlying distribution and the time needed to solve the stochastic program to optimality. As we increase the number of scenarios, we increase the quality of the representation of the distribution, but also decrease the chance to solve the model to optimality within a manageable time. The task is thus to find the smallest number of scenarios that still gives solutions that are both in- and out-of-sample stable, in the sense described in Kaut and Wallace (2007).

We ran our in-sample stability test with different numbers of scenarios and ended up with 200, considering the solution time and stability. The deviation (measured by standard deviation of the objective values of all runs divided by the mean of the objective values) in all cases are less then 1% except for the case of Montreal_r06, where it is 1.5% for single-source cases and 2% for multi-source cases. Out-of-sample stability tests, using a reference tree with 2000 scenarios, are all within 1%. With these values, we are satisfied that we have stability.

3.3 Comparison Tests

As outlined in the Introduction, the deterministic solution, by construction, has a worse expected behavior than its stochastic counterpart. However, we would like to understand more about why this is the case, and in what sense it is worse. This is partly motivated by what we found in Thapalia et al. (2009a) and Thapalia et al. (2009b) where we discussed random demand: The edges (if not their capacities) from the deterministic solution provided a good starting point for the stochastic case. This is unusual.

In order to check the quality of the deterministic designs, as well as comparing them to the stochastic ones, we have set up two tests, named *comparisons*. Whenever a comparison is performed, we take the deterministic and stochastic designs – or parts thereof – (i.e. the first-stage solutions) and evaluate them using reference trees – in our case trees with 2000 scenarios, to make sure we have good approximations of the true distributions. The costs from the design and evaluation phases are added up, making the reported costs comparable across all tests.

A The classical test where the whole first-stage solution is evaluated out-of-sample. This amounts to solving a 2000-scenario stochastic program with all first-stage variables (designs and capacities) fixed, so in fact this equals the solution of 2000 independent second-stage problems. Since the second stage does not involve any integer variables, this is very fast.

B Only information on which edges should be opened is imported from the first stage. So, in a 2000-scenario stochastic program, all discrete variables y describing opened and closed edges—we call it a *skeleton*—are fixed and the stochastic program is run. So the model is allowed to install any capacity on the opened edges (also lower than in the deterministic case), but not to open new ones.

Applied on the deterministic solution, Comparison A is the classical test of the quality of the deterministic solution. The purpose of Comparisons B is to check if the deterministic solution possibly has a good structure, but badly chosen capacities (typically too low). If this is the case, the skeleton can be found using a deterministic model, and then capacities set in a stochastic *linear* model. Algorithmically, this is much simpler than solving a stochastic mixed-integer model. Of course, if the skeleton is good, it also provides information about the relationship between the stochastic and deterministic models of the same problem.

In what follows, we discuss the major findings, details are given in the Appendix. Our first need is to understand the relationship between the stochastic and deterministic solutions.

For each of the 38 test instances, we solve two deterministic problems, one with full capacity available and the other with the mean (in our case 67% of full) capacity on each edge. These choices were motivated in Section 2. In addition, we solve two stochastic versions of each instance, one with uncorrelated and one with positively correlated edge failures. For each stochastic versions, we take the solution of the stochastic model and the two deterministic solutions and evaluate them out-of-sample on the reference tree, i.e. a tree with 2000 scenarios and the same correlations as those used to solve the stochastic programs.

Our measure of the quality of a solution (or a partial solution like a skeleton) will be the ratio between the expected costs using the deterministic solution and the expected costs using the stochastic solution. As the expected costs are never close to zero in our problems, there is no danger of running into problems amounting to a division by zero. Note that since both the stochastic and deterministic solutions are evaluated out-of-sample, the ratio might become slightly smaller than 1.

We also want to explore the relationship between the variable setup cost and the performance of the deterministic *skeleton* in the stochastic environment. The hypothesis is that as the variable setup costs increase, the stochastic skeleton will look increasingly like the deterministic one (which is a tree) due to the cost of opening more capacity than what is absolutely necessary. Similarly, as the variable setup costs increase, the deterministic *skeleton* will tend toward the stochastic one, in terms of number of open edges, as it is (relatively speaking) governed more and more by installed capacity and less and less by the number of opened edges. In other words, we postulate that as variable setup costs decrease, the expected behavior of the deterministic *skeleton* in a stochastic setting becomes increasingly bad, and this is true for both multi- and single-source cases. And as the variable setup costs increase the deterministic *skeleton* will perform better, more so for multi-source case than the single-source case.

For this, we start with the above discussed test cases which we define as base cases. From each base case we make five additional test cases by taking 33%, 66%, 133%, 166% and 200% of the variable setup cost of the base case.

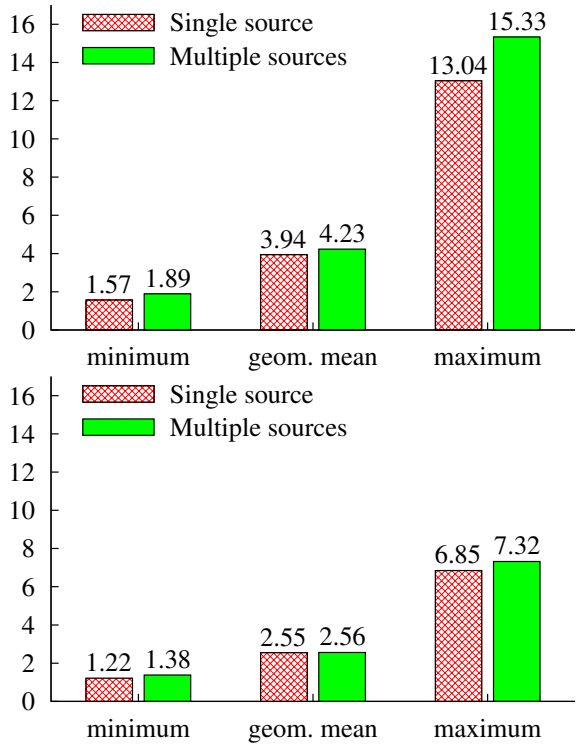

Figure 1: Results of the Comparison A Tests. Quality of the solutions to the deterministic problems with full (*top*) and mean (*bottom*) edge capacities, measured as a ratio of their out-of-sample costs relative to the costs of the corresponding stochastic solutions.

Certainly, the mean-capacity deterministic problem is equivalent to the full-capacity deterministic problem with $1/0.67 = 1.5$ times higher variable setup costs. So this parametric analysis of variable setup costs contains the analysis of the relationship between the two deterministic cases. However, to keep the interpretations apart, we have chosen this approach instead of reading one set of deterministic results from within the results of the other.

4 Computational Results

We present our computational results with discussion.

4.1 Inheritance from the deterministic solutions

The deterministic solution behaves badly in the stochastic environment, while inheriting the deterministic structures, *the skeletons*, is rather good for both the single- and multi-source cases, see Figures 1 and 2. In Figure 1 we see that both deterministic designs are bad but there are some differences. We observe clearly that when mean capacities are used, results are better (around 2.5 times higher on average for both the single- and multi-source cases) than when the full capacities are used (around 4 to 4.25 times higher on average). The reason is simply that when mean capacities are used, the edges seem

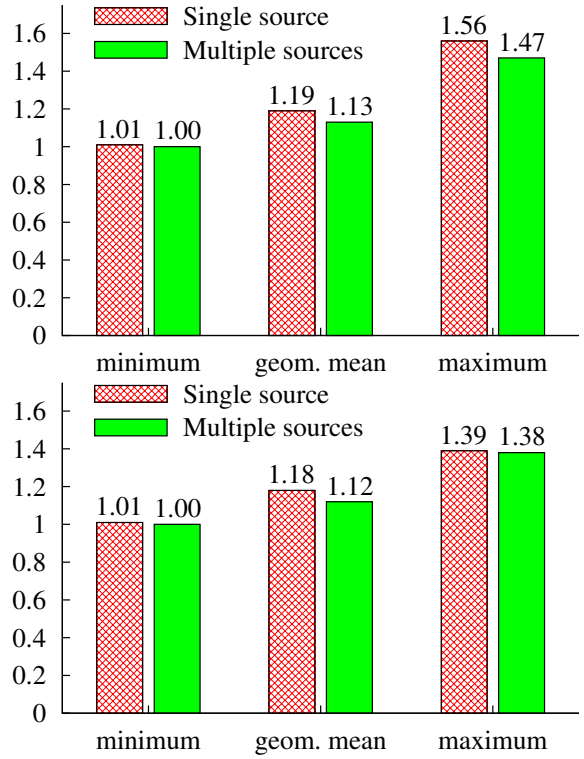

Figure 2: Results of the Comparison B Tests. Quality of the solutions to the deterministic problems with full (*top*) and mean (*bottom*) edge capacities, measured as a ratio of their out-of-sample costs relative to the costs of the corresponding stochastic solutions.

to have less capacity, and hence more is installed. And as we have observed in earlier papers—and that will be confirmed here—deterministic designs do not only suffer in terms of structure (skeletons) but also in terms of too low capacities. So doing what many practitioners do—run deterministic models with a pessimistic view on edge capacities—is indeed a good idea.

From Figure 1, we also observe that the single-source case performs better than the multi-source one for Comparison A for both deterministic versions. This observation is in line with our previous work on random demand (Thapalia et al., 2009b), where we found that as the number of source nodes increases, the deterministic network designs behave steadily worse. The reason is that while the deterministic skeleton is a tree for the single-source case, it is generally a forest with rather shallow trees in the multi-source case. And in a stochastic environment the forest simply does not provide enough connections to satisfy demand in the case of highly variable capacities. The negative effects of a forest are less pronounced than in the case of random demands, though, as the demand within a tree does not change.

Figure 2 shows that when we use the deterministic *skeleton* and apply a stochastic program to set capacities, the results are rather good, implying that the skeletons perform quite well. We can also see that the multi-source cases do better than the single-source cases. This observation holds for most of the tested levels of variable costs, as shown in Figure 3. There, we can also observe that when variable setup costs are only 33% of the

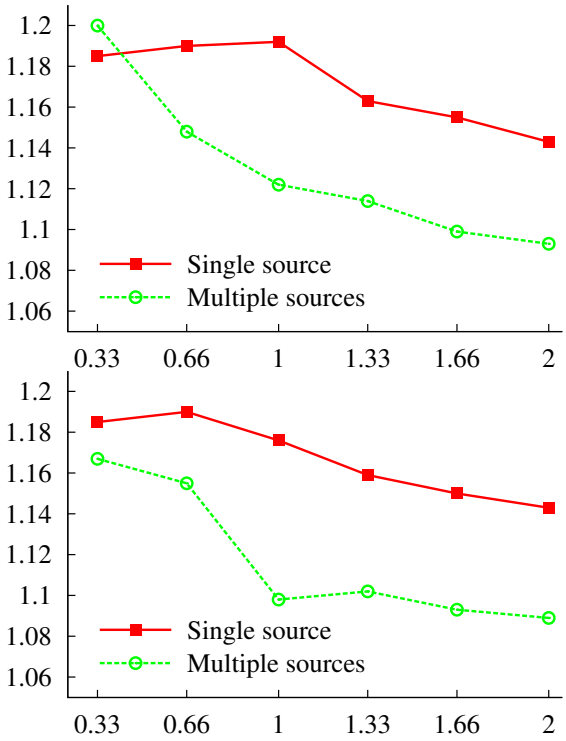

Figure 3: Graph of Comparison B test values for different variable setup cost. The X-axis shows the percentage of variable setup cost as compared to the base case and the Y-axis the quality of the solutions to the deterministic problems with full (*top*) and mean (*bottom*) edge capacities, measured as a ratio of their out-of-sample costs relative to the costs of the corresponding stochastic solutions.

base case costs, the losses are comparable and higher than for higher variable setup costs. As the variable setup costs increase, the loss of using a deterministic skeleton decreases, and more so for the multi-source case. This is true for both maximal and mean value edge capacities. This observation confirms that the quality of a design is a function of both connectivity and capacity. When the variable setup cost is low, the deterministic designs are guided by the shortest routes in terms of fixed costs of opening the edges, and the resulting *skeletons* perform relatively badly in a stochastic environment. But when variable setup costs increase the deterministic designs change. They are no longer primarily guided by the shortest routes in terms of fixed setup costs (mostly implying as few edges as possible), but also edge capacity costs. This results in more edges being opened as total installed capacity, rather than the number of edges with capacity is the primary driver of costs. These *skeletons*, when used in a stochastic environment, perform better as they contain more connections. In the multi-source case this happens more quickly because there are more edges than in the single-source case. Again, since the demand does not vary within each small tree in the forest, contrary to the case with random demand, the fact that the trees effectively cut the design into smaller parts is not a problem. So what is needed here is high density (many paths) and enough installed capacity within each tree to make sure the demand within the tree is satisfied with a high probability so as to achieve low penalty costs.

4.2 Structural Characteristics

This section examines the structures of the deterministic and stochastic network designs from the tests mentioned in Section 3.1 and focuses on a few important observations which shed light on the characteristics of the stochastic designs under random edge capacities.

Deterministic structures with maximal and mean value capacities

Though not very surprising, it is interesting to note that there are some differences between the skeletons based on maximal edge capacity and those based on mean capacity. The designs with mean capacity have higher installed capacities. This is natural since the reduction in actual edge capacities causes higher installed capacities.

As mentioned before, the mean-capacity deterministic problem is equivalent to the full-capacity deterministic problem with $1/0.67 = 1.5$ times higher variable setup costs. Hence, also Figure 3 sheds light on the relationship between the two deterministic solutions.

Network Density

We observe that stochastic designs (multi-source as well as single-source cases) have more edges than their deterministic counterparts and also have higher installed capacities per open edge. In Table 2, the second, third, and fourth columns show that stochastic designs have more edges than their deterministic counterparts. Similarly, columns five, six, and seven show that installed capacity *per open edge* is higher in the stochastic designs ex-

Table 2: Average number of edges and average capacities installed per open edge for all tests in each case. SS and MS denotes single-source and multi-source cases. The parameter ρ refers to correlations in stochastic cases.

	Number of edges			Capacity per edge		
	det.	$\rho \geq 0$	$\rho = 0$	det.	$\rho \geq 0$	$\rho = 0$
France_SS	11.75	16.25	18	624	1096	843
Germany_SS	16.75	28.75	32.75	7	5	4
Montreal_r06_SS	5.5	9.5	10	104	162	111
Nobel-EU_SS	13.62	25.25	29	28	24	18
Pdh_SS	7.37	9.5	10.5	131	186	154
France_MS	13	17	17.67	475	745	646
Germany_MS	19	32.25	36	5	4	3
Montreal_r06_MS	7	9	9.33	53	188	130
Nobel-EU_MS	15.75	23	25.5	13	19	15
Pdh_MS	7.89	8.75	9	87	170	159

cept for the cases of Germany (both for multi- and single-source case) and Nobel-EU_SS. However, in all cases we find a higher total installed capacity in the stochastic designs.

The reasons for this is firstly that with more edges in the network, there are alternative ways to reach demand nodes in the event of reduced edge capacities. And secondly, higher installed capacities insure that there are reasonably high capacities reaching the demand nodes even when there are faults in the edges.

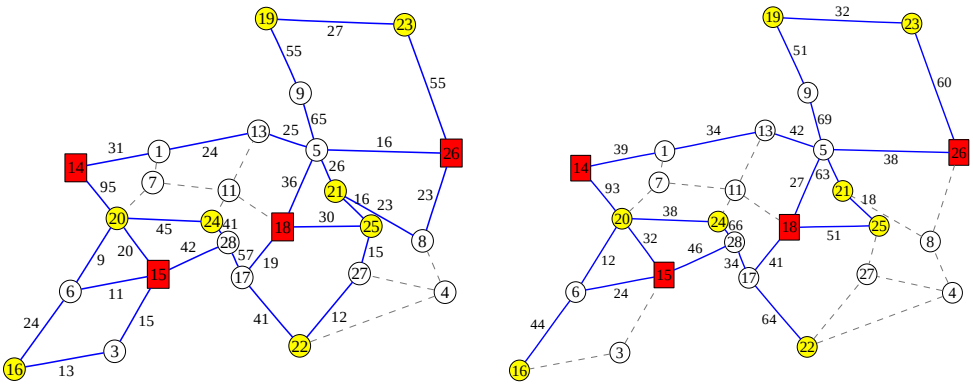

Figure 4: Network Density. Uncorrelated (left) and positively correlated (right) stochastic structure of Nobel-EU showing higher number of edges for the uncorrelated case and generally higher installed capacities on edges for the positively correlated case.

Also we can see from Table 2 (third and fourth columns), that designs for the uncorrelated cases generally have more edges than the corresponding positively correlated cases. This can be explained by the fact that in the uncorrelated cases it is very useful to have alternate paths by setting up extra edges. So when one path leading to a demand node has reduced capacity, another might work well — capacities are uncorrelated. In the positive

correlation case, all edges incident to a node are positively correlated, so even though there may be many paths, they would tend to have difficulties at the same time. This reduces the value of alternative paths. But still some alternate paths are present, which is discussed later. The positive correlation cases compensate by installing more capacities on the edges (see sixth and seventh columns of Table 2). The latter is of course a function of how we defined edge failures – as a percentage of installed capacity. These effects can be seen in Figure 4, where the uncorrelated case has more edges than the positively correlated one and the positively correlated case generally has higher installed capacities on the edges. In the figure, solid (blue) edges are installed with the given capacities, dotted edges are not installed. The dark (red) nodes are the source nodes, the white ones transshipment nodes. The shaded (yellow) nodes are demand nodes. The same color scheme is followed throughout the paper.

Alternative paths

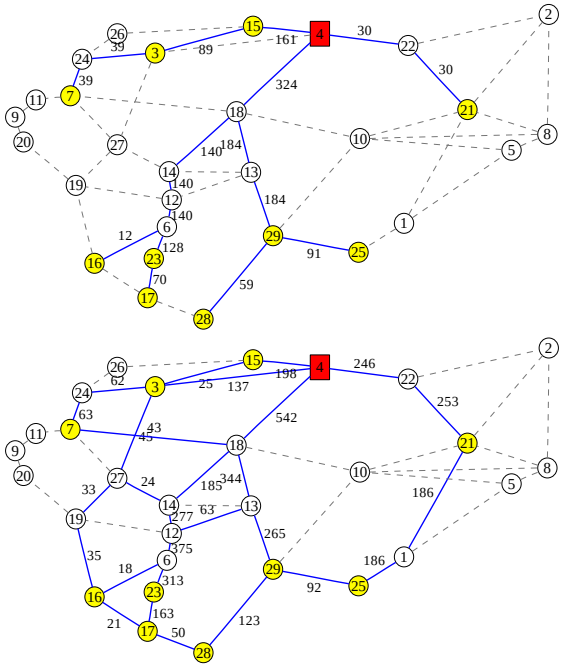

Figure 5: Alternate Paths. Deterministic (top) and stochastic structures(bottom) of Germany_SS_04 showing the alternative paths in the stochastic structure.

In the stochastic structures we observe the creation of alternative paths, even for the case of positively correlated edge failures. This is due to the fact that with alternative paths, the network increases the chance that demand is at least partly satisfied even when one of the paths fails or works at low capacity. We can see this aspect in Figure 5. Demand node 16 in the stochastic structure is served by three paths, one approaching via transshipment node 19, one via transshipment node 6 and finally one via a collection of other demand nodes (through node 17). In this way alternative paths help fulfill demand when one of the paths may be down or have low capacities. Even though this is less useful when failures are positively correlated, the alternative paths still provide some extra

chance of reaching demand node 16.

When studying optimal designs under random demand (see Thapalia et al. (2009b)) we observed that consolidation was the major tool for hedging against uncertainty. By having several demand nodes share paths, one node could use the path when the others didn't need it. With random edge capacities, this need does not emerge since demand is known. Instead hedging comes from having alternative paths, and of course, generally higher installed capacities since edges may fail. Consolidation-like phenomena are hence observed in some cases, but they come from the same phenomena as in deterministic cases: Paths to demand nodes share edges (even if the paths become slightly longer) so that fewer fixed setup costs need to be paid, and, of course, two shortest paths (including both variable and fixed setup costs) may simply happen to share edges.

Loops

The formation of loops is quite visible in networks designed for stochastic edge capacities. Loops are formed among the demand nodes, including or excluding the source nodes, or by joining the leaves of the trees. When we compare this with the networks for stochastic demand (Thapalia et al., 2009b), it is far more prominent here. The main reason for loop formation is the need to provide alternate paths to fulfill demand when some edges are (partly) down. Loops have the advantage that they can be used both ways. So somewhat high capacities (which characterizes the stochastic designs) combined with loops provide alternative paths to demand nodes. We can see this in Figures 5. Here loops are seen in the stochastic network design which is not in the deterministic structures.

Removing edges

It is observed from the test results that in almost all cases, the stochastic skeleton contains the deterministic one. The additional edges are providing flexibility to the network structure. But as we increase the fixed or variable setup costs compared to the base case, the additional edges which were seen in the stochastic skeletons disappear and finally very few are left. We can observe this in Figure 6. When fixed costs are increased, keeping the rest of the cost the same (left column figures), we see that edges or partial paths which were seen only in the stochastic design, but with rather low capacities, like 16-17, 18-10-21, or 3-27-19-16 disappear, making it similar to the deterministic design (top structure of Figure 5). Also we see this effect when we increase the variable setup costs (right column figures). This points toward the suggestion that the deterministic structures are the base for the stochastic network structures. The reason behind this phenomenon is that as the fixed or variable setup costs increase, maintaining the same structure is no longer optimal given the cost of opening the edges and building the capacities compared to the rejection cost. Thus as the fixed or variable setup costs increase, it becomes profitable to remove the edges, which are not core to the basic *backbone*. The edges which are removed are the ones which are not seen in the deterministic structures and\or with lower capacities or the ones which are far from the source nodes.

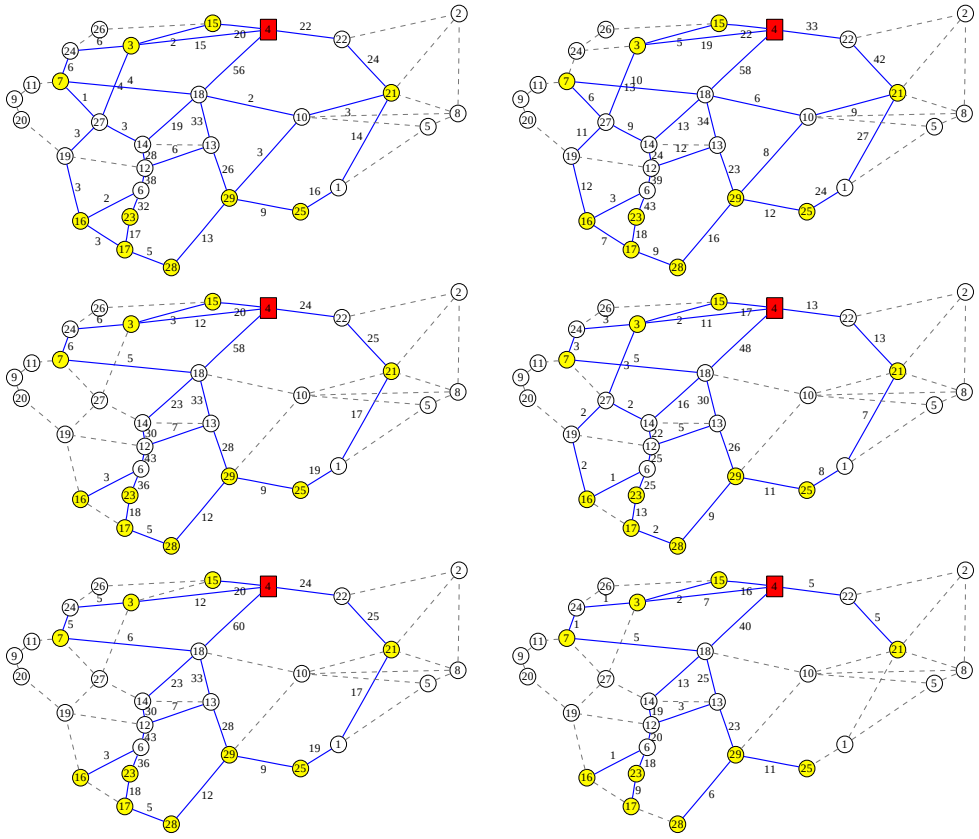

Figure 6: Disappearing Edges. Stochastic structures of Germany_SS with increasing fixed setup costs (left column) and with increasing variable setup costs (right column) showing that stochastic the stochastic designs eliminate edges to emerge as similar to deterministic ones.

5 Conclusion

We have seen that optimal stochastic designs for both the single- and multi-source cases differ from the deterministic ones in both skeletons and capacities. The flexibility, which gives the stochastic designs better expected performance, comes from a higher number of edges and higher installed capacities. With a higher number of edges there exist more paths to demand nodes and hence it also becomes easier to find alternative routes to the demand nodes in the case of edge failures. So while the sharing of paths is the main vehicle for hedging in the case of random demand, here it comes from providing alternative paths from sources to demand nodes.

The deterministic structures as such are not good in the stochastic setting as their expected performance is bad. But borrowing the skeleton from the deterministic structure is rather good. The reason is that the deterministic structure often forms a backbone in the stochastic one. With increased costs for adding edges or for adding capacities in the edges, the stochastic skeletons start to look more and more like the deterministic ones. This happens by shedding installed capacities and edges which are not observed in the deterministic structures. Thus it seems that for cases of this type, using a deterministic method to set the skeleton, and solving a stochastic *linear* program to set capacities is a

very promising approach. Naturally, we cannot test this for larger cases, since we cannot solve the stochastic versions to optimality.

Using the deterministic skeleton is slightly better if based on average edge capacities rather than maximal ones. The reason is somewhat subtle: Using average rather than maximal edge capacities is equivalent to increasing variable setup costs. That reduces the importance of the fixed setup costs, generally leading to more edges being opened, and hence a better starting point for the stochastic linear program.

Correlations have important effects on the structure of the design. With uncorrelated edge failures, the stochastic designs have more edges than when edge failures are positively correlated. With positively correlated edge failures, the networks have higher installed capacities. The reason is simply that with positively correlated failures, all paths to a node tend to have difficulties at the same time, providing less hedging from multiple paths.

Loops are present in the stochastic networks due to a combination of two phenomena. The first is the one we observe for consolidation in the deterministic problem: avoid paying too many fixed setup costs. The second is the characteristics of a ring network. It provides two connections between any pair of nodes in the ring, and the ring can be used in both directions. For these reasons, loops are much more prominent here than with random demand.

So, network designs for stochastic edge capacities are fundamentally different from network designs for stochastic demand. With stochastic edge capacities there are more edges, more loops, and more installed capacities as compared to the design for stochastic demand. A major reason is that there is less consolidation. For stochastic edge capacities we only see consolidation of the type we see in deterministic designs, mostly caused by savings in the fixed setup costs. Instead, alternative connections become more important as a hedge against edges having reduced capacities. Skeletons generally do better here than with random demand as trees in the forests – typical for deterministic designs – no longer have the need to contact each other when randomness strikes, as each tree has enough supply.

Future work. We have now studies random demand and random arc capacities separately. A potential future project is to understand how they interact.

Acknowledgments

This project was supported in part by grant 171007/V30 from The Research Council of Norway. While working on this project, T.G. Crainic was the NSERC Industrial Research Chair on Logistics Management, ESG UQAM, and Adjunct Professor with the Department of Computer Science and Operations Research, Université de Montréal, and the Department of Economics and Business Administration, Molde University College, Norway. Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grants programs.

References

April K. Andreas and J. Cole Smith. Decomposition algorithms for the design of a non-simultaneous capacitated evacuation tree network. *Networks*, 53(2):91–103, 2009.

A. Balakrishnan, T.L. Magnanti, and P. Mirchandani. Designing hierarchical survivable networks. *Operations Research*, 46(1):116–136, 1998.

Michael O. Ball, Charles J. Colbourn, and J. Scott Provan. Network reliability. In M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors, *Network Models*, volume 7 of *Handbooks in Operation Research & Management Science*, chapter 11. North-Holland, Amsterdam, 1995.

Jack Brimberg, Pierre Hansen, Keh-Wei Lih, Nenad Maldenović, and Michéle Breton. An oil pipeline design problem. *Oper. Res.*, 51(2):228–239, March-April 2003.

Anthony Che, Hai Yang, Hong K. Lo, and Wilson H. Tang. Capacity relaibility of a road network: an assessment methodology and numerical results. *Trans. Research. -B.*, 36: 225–252, 2002.

Anthony Chen, Hai Yang, Hong K. Lo, and Wilson H. Tang. A capacity related relaibility for transportation networks. *Journal of Advanced Transportation*, 33(2):183–200, 1999.

R.K.-M. Cheung and W.B. Powell. An algorithm for multistage dynamic networks with random arc capacities, with an application to dynamic fleet management. *Operations Research*, 44(6):951–963, 1996.

G. Chung, K. Lansey, and G. Bayraksan. Reliable water supply system design under uncertainty. *Environmental Modelling & Software*, 24:449–462, 2009.

L. W. Clarke and G. Anandalingam. A bootstrap heuristic for designing minimum cost survivable networks. *Computers Ops Res.*, 22(9):921–934, 1995.

K. Cormican, D.P. Morton, and R.K. Wood. Stochastic network interdiction. *Operations Research*, 46:184–197, 1998.

Francesco di Pierro, Soon-Thaim Khu, Dragan Savić, and Luigi Berardi. Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms. *Environmental Modelling & Software*, 24:202–213, 2009.

Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software: Practice and Experience*, 30(11):1203–1233, 2000. doi: 10.1002/1097-024X(200009)30:11⟨1203::AID-SPE338⟩3.0.CO;2-N.

J. L. Higle and S. W. Wallace. Sensitivity analysis and uncertainty in linear programming. *Interfaces*, 33:53–60, 2003.

K. Høyland, M. Kaut, and S.W. Wallace. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24(2–3):169–185, 2003.

ILOG CPLEX 9.0 User's Manual. ILOG, S. A., 2003.

Michal Kaut and Stein W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257–271, 2007.

Anatoly Lisnianski and Gregory Levitin. *Multi-state system reliability-:assessment, optimization and applications*, volume 6. World Scienetific Publishing Co. Pte. Ltd., 2003.

Changzheng Liu, Yueyue Fan, and Fernando Ordonez. A two-stage sytochastic programming model for transportation network protection. *Computers & Operations Research*, 36:1582–1590, 2009.

Alexander Martin, Markus Möller, and Susanne Moritz. Mixed integer models for the stationary case of gas network optimization. *Math Programming*, 105(2-3):563–582, Feb 2006.

Youngyut Meepetchdee and Nilay Shah. Logistical network design with robustness and complexity considerations. *International Journal of Physical Distribution & Logistics Management*, 37(3):201–222, 2007.

Kjetil T. Midthun, Mette Bjørndal, and Asgeir Tomasgard. Modelling optimal economic dispatch and system effects in natural gas networks. *The Energy Journal*, 30(4):155, 2009.

Idel Montalvo, Joaquin Izquíerdo, Rafael Pérez, and Michael M. Tung. Particle swarm optimization applied to the design of water supply systems. *Computers and Mathematics with Applications*, 56:769–776, 2008.

Young-Soo. Myung, Kim Hyun-joon., and Tch Dong-wan. Design of communication networks with survivability constraints. *Management Science*, 45(2):238–252, 1999.

S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0 – Survivable Network Design library. *Networks*, Early view, 2009. doi: 10.1002/net.20371.

José Emmanuel Ramirez-Marquez and Claudio M. Rocco S. Stochastic network interdiction optimization via capacitated network reliability modeling and probabilistic solution discovery. *Reliability Engineering and System Safety*, 94:913–921, 2009.

B. Rothfarb, H. Frank, D. M. Rosenbaum, K. Steiglitz, and D. J. Kleitman. Optimal design of offshore natural-gas pipeline system. *Operations Research*, 18:992–1020, 1970.

J. Cole Smith, Churlzu Lim, and Fransisca Sudargho. Survivable network design under optimal and heuristic interdiction scenarios. *Journal of Global Optimization*, 38:181–199, 2007.

J.W. Suurballe and Robert E. Tarjan. A quick method for finding shortest pairs of paths. *Networks*, 14:325–336, 1984.

Biju K. Thapalia, Teodor G. Crainic, Michal Kaut, and Stein W. Wallace. Single source single-commodity stochastic network design. Feb 2009a.

Biju K. Thapalia, Teodor G. Crainic, Michal Kaut, and Stein W. Wallace. Single-commodity stochastic network design with multiple sources and sinks. November 2009b.

S. W. Wallace. A piecewise linear upper bound on the network recourse function. *Mathematical Programming*, 38:133–146, 1987.

Stein W. Wallace. Decision making under uncertainty: Is sensitivity analysis of any use? *Operations Research*, 48(1):20–25, 2000.

Daniel De Wolf and Yves Smeers. Optimal dimensioning of pipe networks with application to gas transmission networks. *Oper. Res.*, 44(4):596, 1996.

A Results of the numerical tests

This appendix provides detailed results from the tests in Section 3.1.

Table 3: The ratios for the single source cases corresponding to Figures 1 and 2, split by correlation structure.

	Full capacity				**Mean value capacity**			
	comp. A		**comp. B**		**comp. A**		**comp. B**	
Test Name	$\rho = 0$	$\rho \geq 0$	$\rho = 0$	$\rho \geq 0$	$\rho = 0$	$\rho \geq 0$	$\rho = 0$	$\rho \geq 0$
Germany_SS_04	1.76	1.62	1.20	1.11	1.39	1.26	1.18	1.09
Germany_SS_10	1.71	1.57	1.17	1.08	1.34	1.22	1.17	1.08
Germany_SS_13	2.12	1.92	1.20	1.10	1.57	1.39	1.20	1.10
Germany_SS_27	1.99	1.81	1.21	1.10	1.55	1.37	1.21	1.10
France_SS_06	6.31	5.46	1.27	1.13	3.15	3.23	1.27	1.13
France_SS_10	4.82	4.41	1.56	1.43	2.74	2.66	1.39	1.28
France_SS_13	5.04	4.42	1.31	1.18	3.44	2.74	1.31	1.18
France_SS_16	5.91	5.30	1.22	1.11	3.98	2.91	1.22	1.11
Montreal_r06_SS_01	12.29	11.45	1.24	1.17	6.17	5.71	1.24	1.17
Montreal_r06_SS_03	11.16	9.35	1.30	1.10	5.92	4.89	1.30	1.10
Montreal_r06_SS_04	12.62	11.17	1.33	1.15	6.61	5.69	1.33	1.15
Montreal_r06_SS_08	13.04	10.84	1.32	1.14	6.85	5.65	1.32	1.14
Pdh_01_SS	1.97	1.81	1.10	1.03	1.49	1.33	1.10	1.03
Pdh_02_SS	2.48	2.36	1.07	1.03	1.67	1.56	1.07	1.03
Pdh_04_SS	2.36	2.25	1.04	1.01	1.62	1.51	1.04	1.01
Pdh_08_SS	2.15	2.00	1.13	1.07	1.53	1.40	1.13	1.07
Nobel_EU_SS_04	3.88	3.56	1.42	1.32	2.81	2.54	1.39	1.29
Nobel_EU_SS_05	4.86	4.24	1.34	1.19	3.28	2.79	1.34	1.19
Nobel_EU_SS_15	4.02	3.53	1.33	1.18	2.87	2.45	1.37	1.21
Nobel_EU_SS_18	4.83	4.27	1.31	1.18	3.14	2.74	1.31	1.18

Table 4: The ratios for the multi-source cases corresponding to Figures 1 and 2, split by correlation structure.

	Full capacity				**Mean value capacity**			
	comp. A		**comp. B**		**comp. A**		**comp. B**	
Test Name	$\rho = 0$	$\rho \geq 0$	$\rho = 0$	$\rho \geq 0$	$\rho = 0$	$\rho \geq 0$	$\rho = 0$	$\rho \geq 0$
Germany_1_MS	2.05	1.89	1.14	1.07	1.54	1.40	1.14	1.07
Germany_2_MS	2.23	2.00	1.22	1.12	1.63	1.43	1.22	1.12
Germany_3_MS	2.10	1.95	1.17	1.09	1.52	1.38	1.17	1.09
Germany_4_MS	2.15	1.96	1.12	1.04	1.57	1.41	1.12	1.04
France_MS_1	6.57	5.98	1.17	1.08	3.56	3.20	1.17	1.08
France_MS_2	5.83	5.35	1.38	1.26	3.59	3.30	1.38	1.26
France_MS_3	6.33	5.72	1.47	1.35	3.48	3.10	1.22	1.13
Montreal_r06_MS_1	15.33	14.78	1.09	1.04	6.90	6.76	1.09	1.04
Montreal_r06_MS_2	14.29	13.23	1.20	1.10	6.88	6.36	1.20	1.10
Montreal_r06_MS_3	15.31	14.04	1.14	1.02	7.32	6.69	1.14	1.02
Pdh_1_MS	2.50	2.42	1.02	1.00	1.60	1.52	1.02	1.00
Pdh_2_MS	2.43	2.30	1.03	1.00	1.58	1.48	1.03	1.00
Pdh_3_MS	2.65	2.58	1.02	1.00	1.65	1.59	1.02	1.00
Pdh_4_MS	2.65	2.49	1.08	1.03	1.74	1.60	1.08	1.03
Nobel_EU_MS_plus_01	4.50	4.21	1.28	1.18	2.88	2.70	1.17	1.10
Nobel_EU_MS_plus_02	4.61	4.32	1.20	1.13	2.72	2.54	1.20	1.13
Nobel_EU_MS_plus_03	5.05	4.67	1.26	1.17	2.94	2.70	1.26	1.17
Nobel_EU_MS_plus_04	4.89	4.52	1.23	1.13	3.08	2.77	1.23	1.13

Table 5: The ratios for Comparison B corresponding to Figure 3, for full capacities and mean capacities at different variable setup costs.

	Single source		**Multi-source**	
Var. setup cost	**Full Cap.**	**Mean Cap.**	**Full Cap.**	**Mean Cap.**
0.33	1.18	1.18	1.2	1.17
0.66	1.19	1.19	1.15	1.16
1	1.19	1.18	1.12	1.1
1.33	1.16	1.16	1.11	1.1
1.66	1.16	1.15	1.1	1.09
2	1.14	1.14	1.09	1.09