Comput Manag Sci (2012) 9:139–160
DOI 10.1007/s10287-010-0129-0

ORIGINAL PAPER

Single source single-commodity stochastic network design

Biju K. Thapalia · Stein W. Wallace · Michal Kaut · Teodor Gabriel Crainic

Received: 30 September 2009 / Accepted: 15 December 2010 / Published online: 4 January 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

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

B. K. Thapalia
Molde University College, Molde, Norway

S. W. Wallace
Lancaster University Management School, Lancaster, England
e-mail: stein.w.wallace@lancaster.ac.uk

M. Kaut (✉)
Norwegian University of Science and Technology, Trondheim, Norway
e-mail: michal.kaut@iot.ntnu.no

T. G. Crainic
Université du Québec à Montréal (UQAM), Montreal, Canada
e-mail: TeodorGabriel.Crainic@cirrelt.ca

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.

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 (2010) 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 et al. (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 (2010). 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 et al. (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 cancellation 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 systems (Sherali and Smith 1997), oil pipeline design (Hochbaum and Segev 1989; Rothfarb et al. 1970), sewer network design (Liang et al. 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 et al. 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 $i \in \mathscr{D}$, unsatisfied/lost demand in node i in scenario $s \in \mathscr{S}$;

for $i \in \mathscr{C}$, 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 edge capacity; used for linking capacities and open edges 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:

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

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

$$u_k \leq My_k \qquad \forall k \in E \tag{4}$$

$$0 \leq e_i^s \leq |d_i^s| \qquad \forall i \in \mathscr{D} \bigcup \mathscr{C},\ s \in \mathscr{S} \tag{5}$$

$$x_k^s, z_k^s, u_k \geq 0 \quad \text{and} \quad y_k \in \{0, 1\} \qquad \forall k \in E,\ s \in \mathscr{S} \tag{6}$$

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 used 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) show the bound for the unsatisfied demand and unused supply. 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 h 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 library (Orlowski et al. 2010), available from

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

http://sndlib.zib.de/, with some modification to suit our problem's needs. The fifth instance was generated by us and named Molde and the sixth (Montreal) comes from Crainic et al. (2000). The names of the instances as such do not mean anything particular in this computational setup.

The Montreal test instance does not include node coordinates, so we used Graphviz (Gansner and North 2000), available from http://graphviz.org/, to draw the graph using fixed setup cost as a distance measure. This resulted 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 three nodes (two 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.

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 Lium et al. (2007) that correlations are important in shaping the structure of the network. Hence, we further create three cases from each problem instance: one with uncorrelated demands, one with positively correlated demands and one with mixed correlated demands. In the positively correlated demand cases all correlations are set to 0.7 and in the mixed correlated demand cases: 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 SNDlib and the 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 Sect. 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 Sect. 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 et al. (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 Sect. 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 Sect. 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.

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

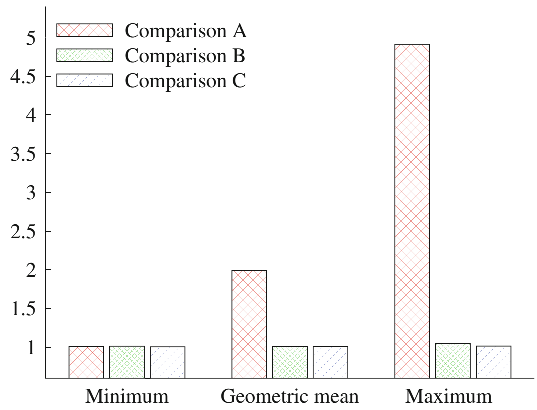

Fig. 1 Ratios of the expected value of the deterministic solutions (25 base test instances) imported into the stochastic setting divided by the optimal objective function value for the stochastic program, for all three comparisons

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 Sect. 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 Fig. 2. We find that the deterministic solution is, as before, rather bad in

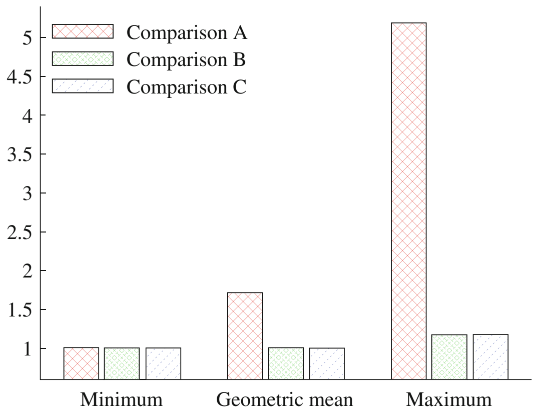

Fig. 2 Ratios of the expected value of the deterministic solutions imported into the stochastic setting divided by the optimal objective function value for the stochastic program, for all three comparisons, taken over all ways to distribute costs between fixed setup costs and capacity costs

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.

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 Sect. 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.

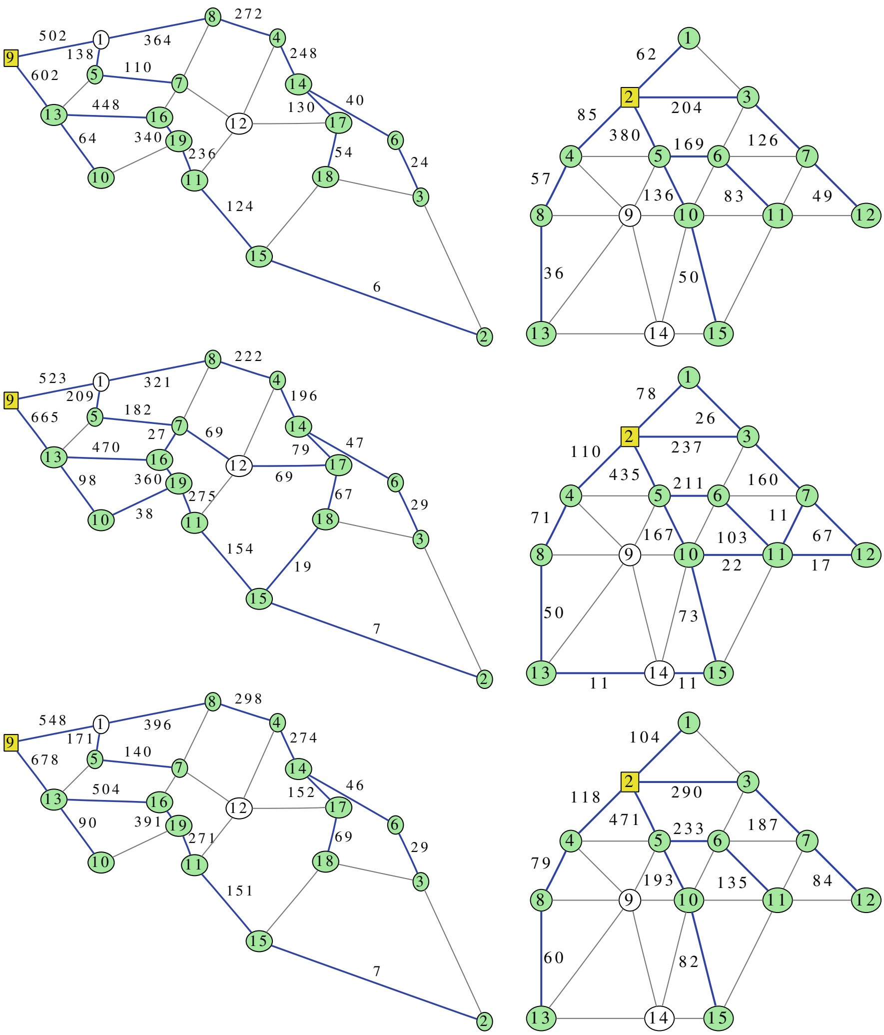

Fig. 3 Deterministic, stochastic, and comparison B solutions of Nobel_EU (*left*) and Molde (*right*) test cases showing presence of deterministic structures in the stochastic solutions

3.4.1 Similarity

Let us first look at Fig. 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 Fig. 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 one because we have only one supply node) is different from the tree of the deterministic solution.

3.4.2 Connecting branches

Sometimes branches in the deterministic design are connected in the stochastic design. Consider the Molde test case (right) in Fig. 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 a 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 Fig. 3.

In the deterministic solution of Fig. 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

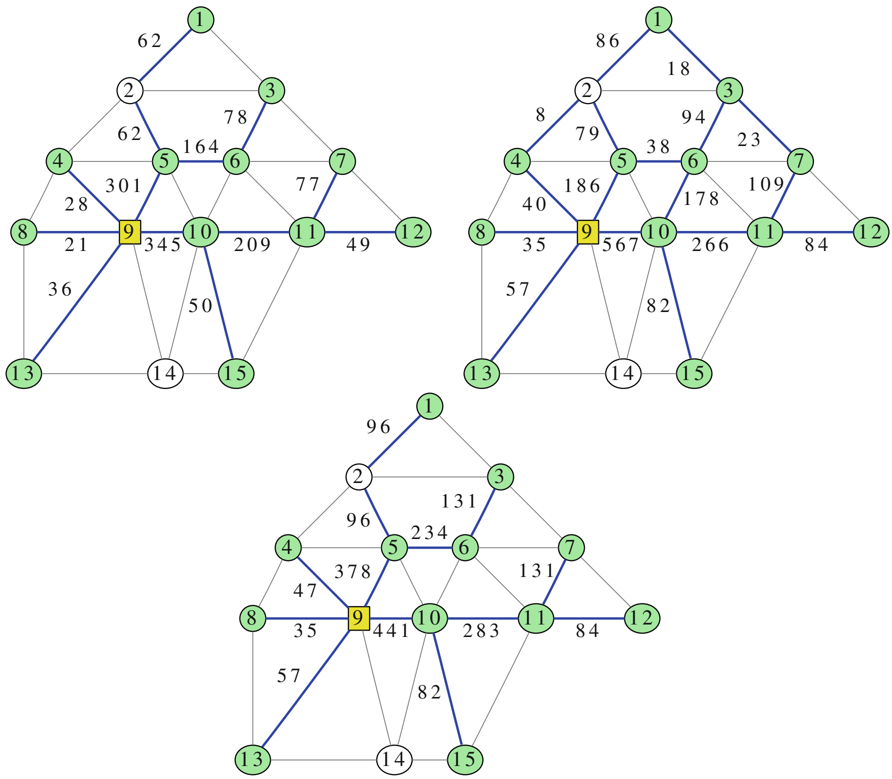

Fig. 4 Deterministic (*top left*), stochastic (*top right*), and comparison B (bottom) solutions of the Molde test instance showing connections between leaves of the deterministic solution tree in stochastic solution

stochastic case it increases by 16.5% to 753. The capacities installed on these branches in comparison B, shown in Fig. 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 among 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.

3.4.3 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 Fig. 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

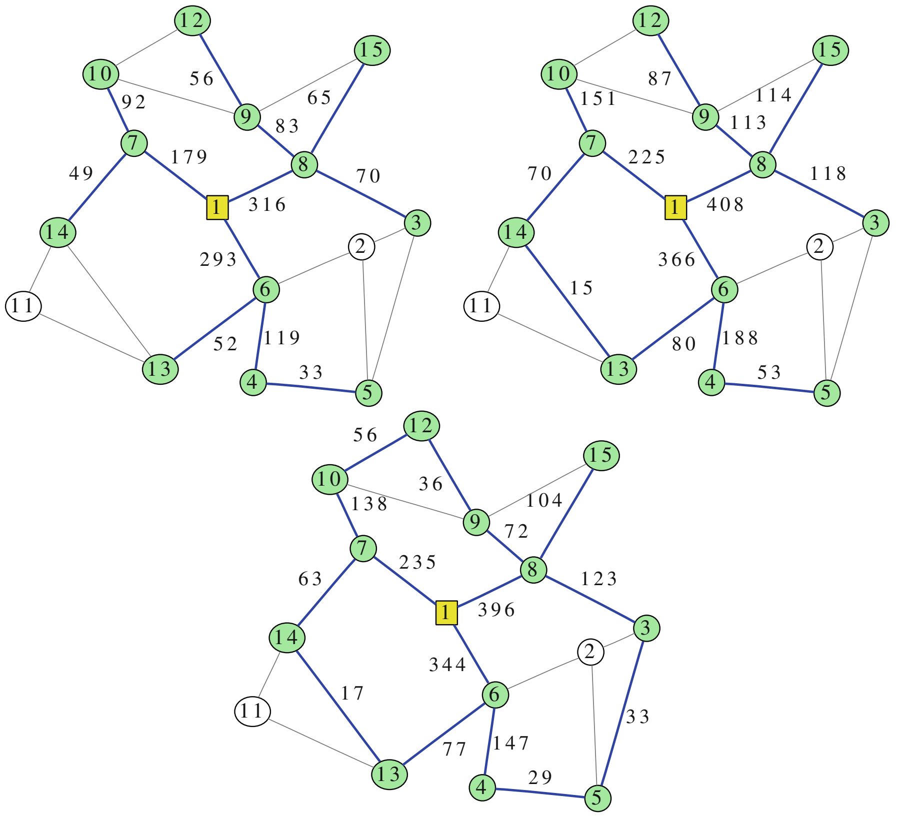

Fig. 5 Deterministic (*top left*) and stochastic (*top right*) solutions of Atlanta. The *bottom figure* is the stochastic solution when demands at nodes 10 and 12 are changed to be of the same magnitude. The installed capacities are divided by one hundred for better readability

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 Fig. 3.

3.4.4 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 Fig. 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

9,239 and 5,582 for nodes 10 and 12. In the bottom graph we have changed the mean demands to the fairly similar values of 6,639 and 5,582. 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.

3.4.5 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 of approximately the same size. 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 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.

As an extreme case of the hedging phenomena observed for the uncorellated cases, where some of the deterministic edges become weaker, for mixed correlations some edges disappear completely. This happens especially when the demand nodes can connect more beneficially to other nearby node(s) with negatively correlated demand. In other words, some nodes switch from one subpath to another. This can be seen in Fig. 6. In the upper part of the figure, the demand of node 4 is positively correlated to that of node 6 and negatively correlated to that of node 3. Hence, in the stochastic solution, it switches from being connected to node 6 to being connected to node 3. A simililar phenomenon can be observed for node 5 in the bottom part of the figure.

With negative correlations present, most leaf nodes in the deterministic tree get connected 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.

3.4.6 Positive correlations

While it is most beneficial to connect nodes with negatively correlated demands, positively correlated nodes may be connected as well. 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) is connected to negatively correlated nodes, creating a pool of nodes that can share capacity.

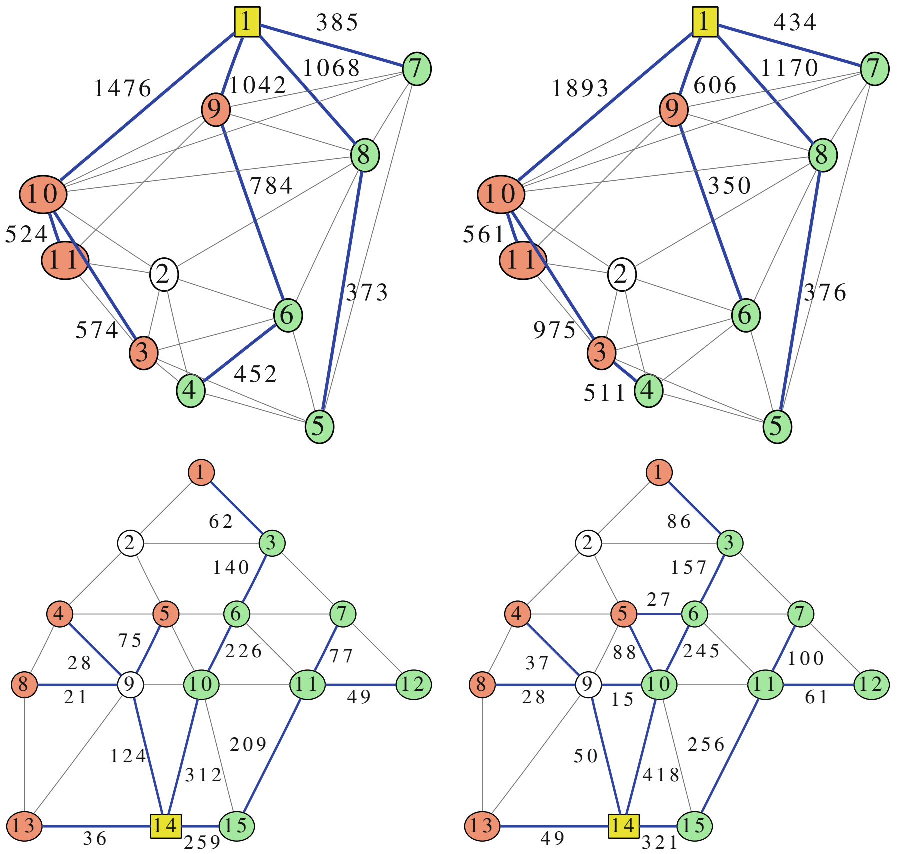

Fig. 6 Deterministic (*left*) and stochastic (*right*) solutions of Pdh and Molde test instances showing connections between negatively correlated demand nodes. The two different shades denote the two groups of demand nodes

Generally speaking, if all the demand is positively correlated, we observe that the stochastic solutions are very similar to the deterministic ones. The reason is that the stronger are the correlations, the less likely it is that one demand node has low demand when another has high, which removes incentives for capacity sharing. Because of the variable demand, the installed capacities are higher than in the deterministic case (Fig. 7), with the difference depending on the rejection costs. With weak positive correlations the solutions are close to those from the uncorrelated case which we already discussed.

4 Conclusion

The purpose of this paper has been to better understand what constitutes a robust design for the SSSND 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

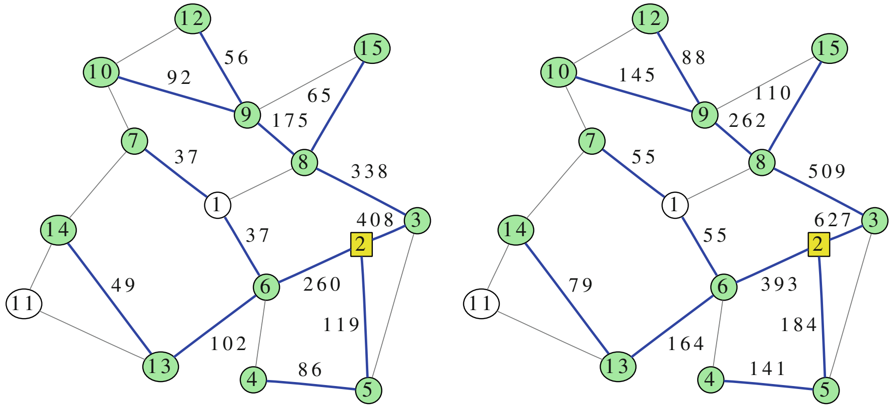

Fig. 7 Deterministic (*left*) and stochastic (*right*) solutions of Atlanta test instance with positive correlations showing that structure is same, only with higher capacities. Note that here capacities are one hundredth of the actual values

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

The message of the paper could thus be summarised as follows: there are important very large stochastic network design problems in the real world; the corresponding models cannot in any way be solved, not even approximately. On the other hand, the decision problems remain, and solutions are found, somehow. Distribution networks are planned and built even though traditional OR models cannot really help with respect to the stochastics. So we put on our OR hats and say: anything that sheds light on these problems is useful as it (at least potentially) brings us closer to understanding what we are looking for: good network designs when stochastics is considered. It is in this light the paper is written—with the very modest hope of adding some understanding.

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.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

Appendix

Results of the numerical tests

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

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

Table 2 The numbers corresponding to Figs. 1, 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

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

Test name	Det. opt.	Zero correlations				Mixed correlations				Positive correlations			
		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

References

Ahuja RK, Magnanti TL, Orlin JB (1993) Network flows—theory, algorithms, and applications. Prentice-Hall, Englewood Cliffs

Aronson JE (1989) A survey of dynamic network flows. Ann oper res 20:1–66

Crainic TG, Gendreau M, Farvolden JM (2000) A simplex-based tabu search method for capacitated network design. Inf J Comput 12(3): 223–236. doi:10.1287/ijoc.12.3.223.12638

Evans JR (1978) A single-commodity transformation for certain multi-commodity networks. Oper Res 26(4):673–680

Gansner ER, North SC (2000) An open graph visualization system and its applications to software engineering. Softw Pract Experience 30(11): 1203–1233. doi:10.1002/1097-024X(200009)30:11<1203::AID-SPE338>3.0.CO;2-N

Higle JL, Wallace SW (2003) Sensitivity analysis and uncertainty in linear programming. Interfaces 33: 53–60

Hochbaum DS, Segev A (1989) Analysis of a flow problem with fixed charge. Networks 19:291–312

Høyland K, Kaut M, Wallace SW (2003) A heuristic for moment-matching scenario generation. Comp Optim Appl 24(2–3):169–185

Kaut M, Wallace SW (2007) Evaluation of scenario-generation methods for stochastic programming. Pacific J Optim 3(2):257–271

Kuah GK, Perl J (1989) The feeder-bus network-design problem. J Oper Res Soc 40(8):751–767

Kuan SN, Ong HL, Ng KM (2006) Solving the feeder bus network design problem by genetic algorithms and ant colony optimization. Adv Eng Softw 37(6):351–359

Liang LY, Thompson RG, Young DM (2004) Optimising the design of sewer networks using genetic algorithms and tabu search. Eng, Constr Archit Manag 11(2):101

Lium A-G, Crainic TG, Wallace SW (2007) Correlations in stochastic programming: a case from stochastic service network design. Asia-Pacific J Oper Res 24((2):161–179

Lium A-G, Crainic TG, Wallace SW (2009) A study of demand stochasticity in stochastic network design. Transp Sci 43(2):144–157

Orlowski S, Pióro M, Tomaszewski A, Wessäly R (2010) SNDlib 1.0—Survivable network design library. Networks 55(3):276–286. doi:10.1002/net.20371

Rothfarb B, Goldstein M (1971) The one-terminal telepak problem. Oper Res 19:156–169

Rothfarb B, Frank H, Rosenbaum DM, Steiglitz K, Kleitman DJ (1970) Optimal design of offshore natural-gas pipeline system. Oper Res 18:992–1020

Sherali HD, Smith EP (1997) A global optimization approach to a water distribution network design problem. J Glob Optim 11(2):107–132

Wallace SW (2000) Decision making under uncertainty: Is sensitivity analysis of any use. Oper Res 48(1):20–25

Wallace SW (2010) Stochastic programming and the option of doing it differently. Ann Oper Res 177(1): 3–8