© https://fti-tn.net/publications Future Technologies and Innovations (FTI) Proceedings: 4th international conference on computer applications and information security (iccais'2021) / March 19 / 2021/ Tunisia: https://fti-tn.net/iccais-2021-list-of-papers

Algorithmic Expedients for the Network Loading Problem

Imen Mejri, Safa Bhar Layeb
LR-OASIS, National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia
imen.mejri@enit.utm.tn, safa.layeb@enit.utm.tn

Abstract—**The Network Design Problems (NDPs) represent a core combinatorial optimization problem. In this work, we investigate a NDP variant that has several significant applications mainly in telecommunication area, namely the Network Loading Problem (NLP). Precisely, the NLP requires installing capacities in a network that allow simultaneous routing of a set of traffic demands while minimizing fixed design costs. Thus, solving this challenging NP-hard problem is crucial for the economic competitiveness of network companies. For this problem, we address deterministic and stochastic models using an arc-path based formulation instead of the commonly used arc-node formulation. One of the most important challenges for network researchers is to ensure that their networks, as they are designed, remain efficient despite the variability of input data. Thus, to cope with real life situation, we consider demand uncertainties, and to deal with this uncertainty, we use the Sample Average Approximation (SAA) approach. We evaluate the gap between the solution derived by the SAA, and the optimal solution provided by the deterministic model with the estimated values of traffic demand. An experimental study was conducted on benchmark instances, in order to assess the proposed approach in the stochastic context.**

Keywords- Network Loading Problem; Arc-Path Formulation; Stochastic Programming; Sample Average Approximation.

I. INTRODUCTION

In terms of graph theory, let's consider a connected undirected graph and a number of distinct point-to-point demand commodities. Each commodity is assigned to a known demand flow value. In general, the goal of a Network Design Problem (NDP) is to design a network at the lowest possible cost thus allowing the total or partial circulation of traffic demands [1]. NDPs have been the subject of numerous scientific investigations. These investigations have covered realistic applications such as the deployment of optical fiber access networks (e.g. [2]-[4]). In addition to the telecommunications area, NDPs have various applications in the fields of logistics, transportation, localization, location and production, to quote just a few (e.g. [5]-[6]). Therefore, many variants of NDPs can be identified according to specific features such as the nature of capacities that could be installed on the edges and/or the costs of these installations. Moreover, several cases of flow strategies may also be considered. In particular, it is a question of routing a simple flow where there is only one demand to be routed between a single source and a single destination, this could be the case of data transmission between a terminal and a central processing node in centralized computer networks [7]. On the other hand, there is the case of multicommodity flow variants where demands are to be routed between several sources and destinations, as the facility location problem between distribution centers and customers (e.g. [2], [8]). Thus, for each commodity the flow is required to be routed along a single path, namely non-bifurcated routing. The case where bifurcations are allowed is referred to as the bifurcated network routing problem [9].

In this paper, we investigate a particular variant of NDPs, namely the Network Loading Problem (NLP), in which capacities can be multiples of integers. Mirchandani [10] has proved that this strategic and challenging NDP is strongly *NP-hard*. The input of the NLP is defined by a connected undirected graph $G = (V, E)$, where V is a set of n nodes, and E is a set of m edges. On each edge e, $e=\{i,j\} \in E$, $i \in V$, $j \in V$, $i \neq j$, there is a link-cost f_e of installing a capacity unit. We also have a set of different point-to-point demand commodities K. A commodity k, $k=1...K$, is given by a definite source node s_k and a definite sink node t_k. Moreover, for each commodity k, $k=1...K$, we have a function $d_k(.)$ that corresponds to the probabilistic density function of the demand k to be partially circulated among different paths. Hence, we have $\tilde{d} = (\tilde{d}_k)_{k=1,..K}$ that represents the random vector of the stochastic demand amount and we suppose that the probability distribution of demand k is known. We also consider a penalty unit cost γ_k that is estimated for each undelivered unit of demand k, $k=1...K$. The NLP requires an installation of optical fibers cables with sufficient integer capacities to enable routing partially simultaneous traffic demands between network users while minimizing the total installation costs and the estimated penalty for undelivered demands.

For sake of clearness, we present an illustrative example depicted in Fig. 1. This example represents a telecommunication network, where each node may be considered as a user that can communicate data flows with other users, and each edge could be a fiber optic cable connecting two users. Thus, we have 5 nodes, 8 edges and 5 traffic demands to deliver without exceeding any of the installed capacities. The cost of installing a capacity unit for each edge *e*, $e \in E$, is detailed in Fig. 1.

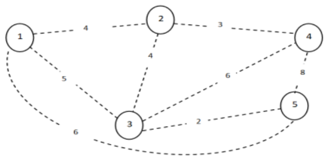

Figure 1. Graph of the illustrative example

Table I depicts the demands characteristics for each commodity *k*, *k=1...K*.

TABLE I. DEMAND CHARACTERISTICS OF THE ILLUSTRATIVE EXAMPLE

Commodity N° k	***(s_k , t_k)***	***Demand value d_k***
1	(1,4)	7
2	(3,4)	1
3	(3,5)	2
4	(4,5)	3
5	(5,2)	2

The optimal solution of the illustrative example of NLP problem is presented in Fig. 2. This solution is obtained when the exact approach of Mejri et al. [11] (discussed further in Section II) is applied on our illustrative example. Herein, we consider a high penalty value to be able to route all demands and we recall that the link capacity variable can take integer values which can be more than 1.

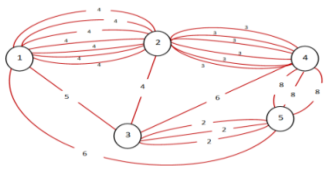

Figure 2. The optimal solution of the illustrative example

As detailed in Fig. 2, the optimal solution value is equal to 103 covering thus the total installation costs. The installed capacities are drawn in Fig. 2, too. For example, 7 units of capacities are installed on the edge {1,2} and only 1 capacity unit is installed on the edge {1,5}. At this stage, it worthy to mention that one difficulty of the NLP is determining the routing demand and assigning link capacities simultaneously, in addition to the multicommodity flow aspect. Not surprisingly, the complexity of NLP increases when the demand amounts are stochastic. To solve this challenging problem, a Sample Average Approximation (SAA) approach is developed, implemented, and assessed on Benchmark instances from the literature.

The rest of the paper is structured as follows. Section II reviews the existing literature. Section III presents the proposed mathematical model. The SAA method is introduced in Section IV. Main results of computational experimentation are reported in Section V. Finally, conclusions and avenues for future works are drawn in Section VI.

II. RELATED WORK

Different flows routing policies have been investigated for the NLP. Particularly, the bifurcated NLP has been explored in [12] and [13]. The authors investigated a large scale problems for which they did not find optimal solutions, and despite the development of several fast heuristics, they were not able to identify the gap deviation from optimality. On the other hand, the case of non-bifurcated routing has the supplementary complexity by considering integer variables to model multicommodity flow paths [14].

When it comes to modeling the NLP, the most proposed and investigated models in the literature are flow-based linear programming formulations ([13], [15]-[16]). This formulation requires, for each edge, a continuous flow variable and an integer design variable. Other formulations for the NLP have been explored, particularly the capacity-based formulation that include only integer link capacity variables. Hence, from the commonly used flow-based formulation the capacity-based formulation is derived by projecting out all continuous flow variables [17]-[18].

It is worthy to note that the existing literature on the NLP often supposes that traffic demands are deterministic in nature and known in advance, whereas in a large number of applications, these demands are stochastic and have considerable variability. Unfortunately, it is very difficult in practice to deal with these uncertainties, and solving a deterministic model to optimality in this context remains significantly hard to handle. In the last decade, many researchers have focused on developing deterministic approaches; however the literature considering uncertainty in NLP's variant remains very scant. A review of the available literature on stochastic NLP's allows identifying two different routing schemes. We can find the static routing procedure where the routing flow must be constant for all realizations of the demands. Adopting this routing policy, Altin et al. [19] focused on polyhedral aspects of the NLP considering hose demand uncertainty for splittable flows. Later, Koster et al. [20] proposed a similar investigations assuming two budgeted uncertainty. On the other hand, we can consider dynamic routing policy that allows adapting the flows to different demand realizations. By considering both bifurcated flows and dynamic routing under demand uncertainty, Mattia

[21] developed the first exact branch-and-cut scheme related to bi-level optimization using the capacity-based formulation for solving the Robust NLP. Later on, an extension of this work was presented in [22] by exploring both static and dynamic routing, with splittable and unsplittable flows.

To the best of our knowledge, no path-based Mixed Integer Linear Programming (MILP) formulation has been investigated so far for solving the stochastic NLP. This could be explained by the exponential number of its path-flow variables in addition to the complexity resulting from the uncertainty of problem parameters. Thus, their resolution poses significant algorithmic challenges. Besides, *the Sample Average Approximation (SAA) approach has not been applied for the stochastic NLP*, which is one of the contributions of this work.

III. Path-Based Formulation

To formulate the stochastic NLP where the demand amounts are considered uncertain, we denote by P_k the list of all available paths coming out from the source nodes s_k to the sink nodes t_k. We associate a binary constant α_{erk} that takes value 1 if edge e, e ϵ E, is in the path r, r ϵ P_k , of commodity k k=1,...,K.

In what follows, we define y_e, $e \in E$, the integer link-capacity variable on each edge and z^k_r, $k=1,...,K$, $r \in P_k$, a continuous nonnegative variable that corresponds to the value of flow of commodity *k* delivered through the path *r*. We also introduce a continuous nonnegative variable ρ_k which expresses the quantity of undelivered demand of the commodity *k*, *k=1,...,K.* Then, the proposed path-based formulation for the stochastic NLP reads as:

$$(PF): Minimize \sum_{e \in E} f_e\, y_e + \sum_{k=1}^{K} \gamma_k \rho_k \qquad (1)$$

Subject to:

$$\sum_{r=1}^{|P^k|} z_r^{\,k} + \rho_k \geq \tilde{d}_k, \quad k = 1,..., K, \qquad (2)$$

$$\sum_{k=1}^{K}\sum_{r=1}^{|P^k|} \alpha_{erk}\; z_r^{\,k} \leq y_e \quad \forall e \in E, \qquad (3)$$

$$z_r^{\,k} \geq 0 \quad , \quad \forall r \in P^k, \quad k = 1,..., K, \qquad (4)$$

$$\rho_k \geq 0 \quad , \quad \forall, k = 1,..., K, \qquad (5)$$

$$y_e \in \mathrm{N}, \quad \forall e \in E. \qquad (6)$$

The objective (1) aims at minimizing the total installation costs and the estimated penalty for undelivered demands. Constraints (2) enable routing partially simultaneous realizations of the demand vector $\tilde{d} = (\tilde{d}_k)_{k=1,..,K}$. Constraints (3) reflect the capacity restrictions for each edge. Constraints (4)-(5) impose the non-negativity of decision variables *z* and ρ. Constraints (6) limit variables *y* to be integers.

IV. Sample Average Approximation Approach

Herein, we are interested in estimating the real-valued network solution by (stochastic) simulation. Let *S* be the set of the possible demand scenarios. We notice that a scenario *s*, *s* ϵ *S*, corresponds to a vector $\tilde{d}^{\,s} = (\tilde{d}_1^{\,s},..., \tilde{d}_K^{\,s})$ of potential random demand realizations $\tilde{d}_k$, *k=1,...,K.* This leads to the following extended stochastic optimization formulation:

$$Minimize \sum_{e \in E} f_e\, y_e + \frac{1}{|S|}\sum_{s=1}^{|S|}\sum_{k=1}^{K} \gamma_k \rho_k^s \qquad (7)$$

Subject to: (6),

$$\sum_{r=1}^{|P^k|} z_{rk}^{\;s} + \rho_k^s \geq \tilde{d}_k^{\,s}, \quad k = 1,..., K, \quad s = 1,...|S|, \qquad (8)$$

$$\sum_{k=1}^{K}\sum_{r=1}^{|P^k|} \alpha_{erk}\; z_{rk}^{\;s} \leq y_e \quad \forall e \in E, \quad s = 1,...|S|, \qquad (9)$$

$$z_{rk}^{\;s} \geq 0 \quad , \quad \forall r \in P^k, \quad k = 1,..., K, \quad s = 1,...|S|, \qquad (10)$$

$$\rho_k^s \geq 0 \quad , \quad \forall, k = 1,..., K, \quad s = 1,...|S|. \qquad (11)$$

Due to the large number of possible demand realizations, it is usually impossible to find the exact optimal solution of stochastic programming. Thus, we conducted studies using one standard approach, namely Sample Average Approximation (SAA) method. SAA is a Monte Carlo simulation technique involving sampling and optimization methods for deterministic problems. The main idea of the SAA procedure consists in generating a finite number of scenarios to approximate the expected value of the objective function [23]. Hence, in the following, we consider a sampling procedure that was introduced by Mak, et al. [24], and later on used in [25].

To efficiently solve the resulting approximated model derived from (7)-(11), we propose a SAA based-procedure as follows:

- First, we generate *N* independent sample subsets of scenarios $S^1,...,S^N$. For each subset S^ξ , $\xi=1,...,N$, we have *M* different combinations of scenarios $s^\xi_1,..., s^\xi_M$.
- For each possible demand realization $((\tilde{d}_1^{\,p,\xi},..., \tilde{d}_k^{\,p,\xi})$, $p=1,...,M$, $\xi=1,...N)$, associated to each scenario s^ξ_p, we consider the following problem :

$$z(S^\xi): Minimize \sum_{e \in E} f_e\, y_e + \frac{1}{|M|}\sum_{P=1}^{|M|}\sum_{k=1}^{K} \gamma_k \rho_k^P \qquad (12)$$

© https://fti-tn.net/publications Future Technologies and Innovations (FTI) Proceedings: 4th international conference on computer applications and information security (iccais'2021) / March 19 / 2021/ Tunisia: https://fti-tn.net/iccais-2021-list-of-papers

Subject to: (6),

$$\sum_{r=1}^{|P^k|} z_{rk}{}^{p} + \rho_k^p \geq \tilde{d}_k^{p,\xi}, \quad k=1,...,K, \quad p=1,...|M|, \qquad (13)$$

$$\sum_{k=1}^{K}\sum_{r=1}^{|P^k|} \alpha_{erk} z_{rk}{}^{P} \leq y_e \quad \forall e \in E, \quad p=1,...|M|, \qquad (14)$$

$$z_{rk}{}^{P} \geq 0 \quad , \quad \forall r \in P^k, \quad k=1,...,K, \quad p=1,...|M|, \qquad (15)$$

$$\rho_k^p \geq 0 \quad , \quad \forall, k=1,...,K, \quad p=1,...|M|. \qquad (16)$$

Then, the above problem should be solved to find $\tilde{z}(S^{\xi})$ the value of the objective function for each sample subset of scenarios S^{ξ}, $\xi=1,...N$. To achieve this, we applied the exact approach proposed in [11] based on a tailored Benders decomposition procedure, coupled with a column generation method and an exact cut generation model [26].

- Finally, we compute the following valid lower bound for the stochastic NLP:

$$Z^{sAA} = \frac{1}{N}\sum_{\xi=1}^{N} \tilde{z}(S^{\xi}). \quad (17)$$

As discussed in [24], the SAA is an effective approach when the number of possible scenarios is sufficiently high to estimate the optimal solution of stochastic programming. Thus, the quality of SAA approximated solution monotonically improves by increasing the number of parameters *M* (number of scenarios) and *N* (subsets cardinality). Unfortunately, it is very hard in practice to solve large-size instances considering a high number of scenarios. This leads to an extremely long computing time. A way to deal with these restrictions is to identify whether an obvious point of compromise could be made between the performance of the approximate solutions and the required time consuming. Herein, in this work, we focus on making this trade-off while deciding on values of *M* and *N*.

V. Computational Experimentation

The proposed SAA approach has been coded using C# language in concert with the commercial solver CPLEX (version 12.6). All computations were carried out on a computer equipped with an Intel(R) CORE (TM) i7-9750H CPU@ 2.6GHz processor and 24 GB of RAM.

The considered test-bed consists of 6 realistic network instances inspired from live traffic data histories. There is one instance denoted by ABILENE and defined by the U.S. Internet2 Network [27] and one instance denoted by GERMANY17 and provided by the NOBEL project [28]. The physical topology of ABILENE and GERMANY17 networks appear in Fig. 3 and Fig. 4, respectively. The other remaining real-life instances are studied in [20] and [22]. It is worthy to mention that these instances are also accessible from the Survivable Network Design Library (SNDlib) [29].

Figure 3. ABILENE network physical topology [29]

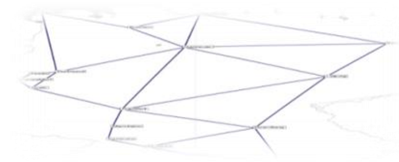

Figure 4. GERMANY17 network physical topology [29]

A description of the realistic instances is shown in Table II, where the column (***Inst.***) refers to the instance designation. The numbers of nodes (***n***) and edges (***m***) range from 11-17, and 15-42, respectively. The number of commodities (***K***) varies from 22 to 121.

TABLE II. The Instances Characteristics [29]

Inst.	***n***	***m***	***K***
ABILENE	12	15	66
GERMANY17	17	26	121
DI-YUAN	11	42	22
PDH	11	34	24
POLSKA	12	18	66
NOBEL-US	14	21	91

To deal with demand uncertainty, we transform the commonly-used deterministic demand d_k to a probabilistic demand $\tilde{d}_k$ such that $E[\tilde{d}_k] = d_k$. For the numerical experiments, we set the coefficient of variation at 20% (i.e. $Var[\tilde{d}_k] = (0.2 \times \mathrm{E}[\tilde{d}_k])^2$), defining a 'low' variance scenario. Then, we suppose that $\tilde{d}_k$ will follow a LogNormal distribution with $E[\tilde{d}_k] = d_k$. The LogNormal distribution has been adopted because this distribution should be preferred than the Normal distribution in the case of positive demands [30]. Then, we choose the values of location parameter μ_k, and scale parameter σ_k, corresponding to each lognormally distributed demand $\tilde{d}_k$ $\sim$*Lognormal* (μ_k, σ_k), according to the method of moments introduced by Juan et al. [30] and used in designing stochastic networks such as [6] and [25]. More precisely, we apply the following equations (18) and (19):

$$\mu_k = \ln\left(\mathrm{E}[\tilde{d}_k]\right) - \frac{1}{2}\ln\left(1 + \frac{Var[\tilde{d}_k]}{\left(\mathrm{E}[\tilde{d}_k]\right)^2}\right), \qquad (18)$$

$$\sigma_k = \left| \sqrt{\left(1+\ln\left(\frac{Var[\tilde{d}_k]}{(E[\tilde{d}_k])^2}\right)\right)} \right|. \quad (19)$$

Setting the appropriate parameter configurations is often complex to tune as it is a crucial and error prone task. In this context, in order to find an appropriate tuning parameter of our approach, we have considered in our experiments different combinations, as previously explained in Section IV, for defining the number of scenarios M and subsets cardinality N characterizing the SAA approach. A compromise between the solutions performance and required CPU times has been taken into account. Thus, a large empirical experimentation was conducted to eventually set the values of the parameters M and N as follows: $M=100$, $N=10$.

A. *Performance of the Sample Average Approximation approach*

We first evaluate empirically the performance of the probabilistic lower bound for stochastic NLP provided by SAA approach. Then, we compare the deterministic solution where traffic demand is fixed over time to the stochastic one where we have random demand realizations. Specifically, we define the corresponding GAP, which is the difference in objective values between the stochastic lower bound solution and the optimal deterministic solution. We notice that, at this stage, the penalty value is fixed at 5.

The obtained results are reported in Table III. Let $\boldsymbol{Z^{SAA}}$, $\boldsymbol{Time^{SAA}}$, $\boldsymbol{Z^{*}}$, $\boldsymbol{Time^{*}}$, and ***GAP*** denote respectively, the value of the proposed SAA approach solution, the total CPU time to calculate Z^{SAA} in seconds, the value of the optimal deterministic solution and its total CPU time in seconds as computed in [11], and the gap in percentage computed as $GAP=100\times((Z^{*}-Z^{SAA})/Z^{*})$.

TABLE III. PERFORMANCE OF THE PROPOSED SAA APPROACH

Inst.	Z^{SAA}	$Time^{SAA}$	Z^{*}	$Time^{*}$	*GAP*
ABILENE	32817830396	359.44	32822031845	7.33	0.01%
GERMAN Y17	28361	321.58	29480	66.74	3.80%
DI-YUAN	1617135	82.30	1619886	15.87	0.17%
PDH	502238977	186.19	502572465	58.36	0.07%
POLSKA	5477976	264.06	5639987	3216.87	2.87%
NOBEL-US	191787	274.30	209840	2607.15	8.60%

Globally, the results show that instances with up to 17 nodes and 121 commodities are solved in a reasonable average time, which reflects the effectiveness of the proposed stochastic procedure. In addition, we see from table III that stochastic programming provides more effective results when compared to the deterministic approach and with respect to our objective function. However, the SAA proposed approach requires not surprisingly more computing times. Notice also that the average gap between the obtained lower bound solutions and the available deterministic results is about 2.5% which is not really important; this is due to the 'low' variance scenario. We should, nevertheless, recall that this comparative analysis is not absolutely rigorous because we compare different approaches for different types of demand realizations (fixed scenarios versus random scenarios). A better comparison is performed if upper bounds of the stochastic solutions are derived with a simulation-based optimization such as in [25]. This gives an additional perspective by indicating the performance of the stochastic model with respect to the demand variability.

B. *Relationship between the total installation costs and the expected penalty costs*

An interesting aspect to investigate is the relationship between the costs of installing capacities and the estimated penalties of undelivered demands. The computational results are reported in Table IV. For each network instance, let $\boldsymbol{F^{SAA}}$ be the corresponding fixed installation costs, and $\boldsymbol{P^{SAA}}$ be the corresponding estimated penalties resulting from the unrouted flows. Obviously, $\boldsymbol{Z^{SAA}} = \boldsymbol{F^{SAA}} + \boldsymbol{P^{SAA}}$. Table V shows that the estimated penalties constitute an average of 0.437% of the total installation costs. This shows that the installed capacities in the network are not sufficient to satisfy the simultaneous routing of all traffic demands. We also mention that these results reflect the real-life situations, such as in telecommunications where the fixed installation costs are significantly high compared to the penalty costs.

TABLE IV. RELATIONSHIP BETWEEN THE COSTS OF INSTALLING CAPACITIES AND PENALTIES OF UNDELIVERED DEMANDS

Inst.	Z^{SAA}	F^{SAA}	P^{SAA}	P^{SAA}/F^{SAA}
ABILENE	32817830396	32476688141	1524363	0.00005
GERMANY17	28361	28303	58	0.00205
DI-YUAN	1617135	1616727	408	0.00025
PDH	502238977	490806903	11432074	0.02329
POLSKA	5477976	5477590	386	0.00007
NOBEL-US	191787	191684	103	0.00054
Average				0.00437

Next, we investigate the effect of the per-unit penalty cost on the obtained SAA solutions. Test cases with different per-unit penalty cost values are examined, namely 5, 50, 100, and 150. Since no significant difference in results was detected while testing these values on all realistic instances, so to lighten the analysis, only the results of ABILENE and DI-YUAN Networks are reported in Table V.

TABLE V. COMPARISON OF THE OBTAINED SAA SOLUTIONS FOR DIFFERENT PENALTY VALUES

Inst.	*Penalty*	Z^{SAA}	$Time^{SAA}$	F^{SAA}	P^{SAA}

ABILENE	5	32817830396	359.44	32476688141	1524363
	50	32835360579	364.90	32816306033	19054546
	100	32854415125	382.40	32816306033	38109092
	150	32872856845	314.90	32816306033	56550812
DI-YUAN	5	1617135	82.31	1616727	408
	50	1618276	21.15	1617460	816
	100	1618684	20.01	1617460	1224
	150	1621000	17.50	1617460	3540

Table V shows that the proposed SAA approach has a tendency to design networks with higher link installation costs, when the penalty costs increase. Then, the network will be completely saturated (corresponds to a penalty of 50 in Table V), which explains why the fixed installation costs remain constant despite the continuing increase of the penalty values.

VI. Conclusion and Perspectives

This paper investigates the Network Loading Problem with stochastic demands. It is a challenging NP-hard network design problem that is drawing the interest of practitioners as well as researchers for its relevant applications, particularly in the field of telecommunications. To tackle this problem, we begun by addressing deterministic and stochastic models using an arc-path based MILP formulations instead of the commonly used arc-node formulation. To deal with the demand uncertainty, we tailored a Sample Average Approximation based procedure. Computational experimentation was conducted on realistic benchmark instances from the literature. Promising results have been reported regarding the performance of the proposed approach. Studies are underway to explore the avenue of developing effective approach that couples simulation as well as optimization, to derive approximate stochastic solutions for the NLP.

References

[1] Minoux, M. (1989). Networks synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, *19*(3), 313-360.

[2] Mejri, I., Layeb, S. B., & Zeghal Mansour, F. (2019). Formulations and Benders Decomposition based Procedures for the Discrete Cost Multicommodity Network Design Problem. *International Journal of Computing and Digital Systems*, *8*(6), 659-668.

[3] Mejri, I., Layeb, S. B., & Mansour, Z., F. (2018, November). Solving the Discrete Cost Multicommodity Network Design Problem to Optimality. In *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)* (pp. 1-5). IEEE.

[4] Layeb, S. B., Heni, R., & Balma, A. (2017, May). Compact MILP models for the discrete cost multicommodity network design problem. In *2017 International Conference on Engineering & MIS (ICEMIS)* (pp. 1-7). IEEE.

[5] Balakrishnan, A., Magnanti, T. L., & Mirchandani, P. (1997). Network design. *Annotated bibliographies in combinatorial optimization*, 311-334.

[6] Layeb, S. B., Jaoua, A., Jbira, A., & Makhlouf, Y. (2018). A simulation-optimization approach for scheduling in stochastic freight transportation. *Computers & Industrial Engineering*, *126*, 99-110.

[7] Randazzo, C. D., & Luna, H. P. L. (2001). A comparison of optimal methods for local access uncapacitated network design. *Annals of Operations Research*, *106*(1-4), 263-286.

[8] Geoffrion, A. M., & Graves, G. W. (1974). Multicommodity distribution system design by Benders decomposition. *Management science*, *20*(5), 822-844.

[9] Barahona, F. (1996). Network design using cut inequalities. *SIAM Journal on optimization*, *6*(3), 823-837.

[10] Mirchandani, P. (1989). Polyhedral Structure ofa CapacitatedNetwork Design Problem with an Application to the Telecommunication Industry. *Ph. D. Dissertation*, MIT Sloan School of Management, Cambridge, MA.

[11] Mejri, I., Layeb, S. B., & Mansour, F. Z. (2019, April). Enhanced Exact Approach for the Network Loading Problem. In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 970-975). IEEE.

[12] Bienstock, D., & Günlük, O. (1996). Capacitated network design—polyhedral structure and computation. *Informs journal on Computing*, *8*(3), 243-259.

[13] Magnanti, T. L., Mirchandani, P., & Vachani, R. (1995). Modeling and solving the two-facility capacitated network loading problem. *Operations research*, *43*(1), 142-157.

[14] Gavish, B., & Altinkemer, K. (1990). Backbone network design tools with economic tradeoffs. *ORSA Journal on Computing*, *2*(3), 236-252.

[15] Magnanti, T. L., Mirchandani, P., & Vachani, R. (1993). The convex hull of two core capacitated network design problems. *Mathematical programming*, *60*(1-3), 233-250.

[16] Ljubić, I., Putz, P., & Salazar‐ González, J. J. (2012). Exact approaches to the single‐ source network loading problem. *Networks*, *59*(1), 89-106.

[17] Avella, P., Mattia, S., & Sassano, A. (2004, June). Metric inequalities and the network loading problem. In *International Conference on Integer Programming and Combinatorial Optimization* (pp. 16-32). Springer, Berlin, Heidelberg.

[18] Mattia, S. (2012). Separating tight metric inequalities by bilevel programming. *Operations research letters*, *40*(6), 568-572.

[19] Altın, A., Yaman, H., & Pınar, M. Ç. (2011). The robust network loading problem under hose demand uncertainty: formulation, polyhedral analysis, and computations. *INFORMS Journal on Computing*, *23*(1), 75-89.

[20] Koster, A. M., Kutschka, M., & Raack, C. (2013). Robust network design: Formulations, valid inequalities, and computations. *Networks*, *61*(2), 128-149.

[21] Mattia, S. (2013). The robust network loading problem with dynamic routing. *Computational Optimization and Applications*, *54*(3), 619-643.

[22] Mattia, S., & Poss, M. (2018). A comparison of different routing schemes for the robust network loading problem: polyhedral results and computation. *Computational Optimization and Applications*, *69*(3), 753-800.

[23] Kim, S., Pasupathy, R., & Henderson, S. G. (2015). A guide to sample average approximation. In *Handbook of simulation optimization* (pp. 207-243). Springer, New York, NY.

[24] Mak, W. K., Morton, D. P., & Wood, R. K. (1999). Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations research letters*, *24*(1-2), 47-56.

[25] Mejri, I., Layeb, S. B., Haouari, M., & Mansour, F. Z. (2019). A simulation-optimization approach for the stochastic discrete cost multicommodity flow problem. *Engineering Optimization*, *52*(3), 507-526.

[26] Mejri, I., Haouari, M., Layeb, S. B., & Mansour, F. Z. (2019). An exact approach for the multicommodity network optimization problem with a step cost function. *RAIRO-Operations Research*, *53*(4), 1279-1295.

[27] ABILENE. Abilene Internet2 Network. Available at www.internet2.edu/network.

[28] GERMANY17. NOBEL – Next generation Optical networks for Broadband European Leadership , 2004– 2007. Available at http://www.ist-nobel.org.

[29] Orlowski, S., Wessäly, R., Pióro, M., & Tomaszewski, A. (2010). SNDlib 1.0—Survivable network design library. *Networks: An International Journal*, *55*(3), 276-286.

[30] Juan, A. A., Grasman, S. E., Caceres-Cruz, J., & Bektaş, T. (2014). A simheuristic algorithm for the single-period stochastic inventory-routing

problem with stock-outs. *Simulation Modelling Practice and Theory*, *46*, 40-52.