Genetic Algorithms for the Network Slice Design Problem Under Uncertainty

Thomas Bauschert
Technische Universität Chemnitz
Chair of Communication Networks
09126 Chemnitz, Germany
thomas.bauschert@etit.tu-chemnitz.de

Varun S. Reddy
Technische Universität Chemnitz
Chair of Communication Networks
09126 Chemnitz, Germany
srva@etit.tu-chemnitz.de

ABSTRACT

Robust optimisation, in recent years, has surfaced as an essential technique to handle data uncertainty in mathematical programming models. However, the resulting robust counterparts are often hard to solve even for modern state-of-the-art Mixed Integer Programming solvers, underlining the need for approximate algorithms. Based on the works of Gonçalves and Resende [3], we propose genetic algorithms for the network slice design problem (NSDP) under uncertainty. We investigate the performance of the proposed algorithms using realistic datasets from SNDlib [4].

CCS CONCEPTS

• **Computing methodologies → Genetic algorithms**; *Planning under uncertainty*; • **Theory of computation → *Optimization with randomized search heuristics***; Network optimization.

KEYWORDS

Combinatorial optimisation, Genetic algorithms, Network optimisation, Network slicing, Robust optimisation

ACM Reference Format:
Thomas Bauschert and Varun S. Reddy. 2019. Genetic Algorithms for the Network Slice Design Problem Under Uncertainty. In *Genetic and Evolutionary Computation Conference Companion (GECCO'19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322056

1 INTRODUCTION

Network slicing concerns optimal partitioning and allocation of logical network resources among the network slice requests so as to ensure independent control over the resources allocated to the respective slices. We revisit the problem of designing a large-scale logical network slice under traffic uncertainty [1] using the notion of layered graphs [5]. We employ a fine-grained approach to model the uncertainty in the traffic demands by adopting a generalisation of the Bertsimas and Sim model, namely the "multi-band" uncertainty set [2]. By capturing the distribution of the uncertain coefficients using a histogram-like model, the multi-band uncertainty set greatly improves the modelling power of the uncertain coefficients thereby resulting in robust allocation of physical substrate network resources with reduced conservativeness. Notwithstanding the advantages, the resulting robust counterparts can prove very challenging even for state-of-the-art Mixed Integer Programming solvers. To this end, we propose a modified version of the biased random-key genetic algorithm [3] to solve the robust network slice design problems. We first present the optimisation model for the robust network slice design problem followed by the description of the algorithm and some experimental results to conclude the paper.

2 ROBUST NETWORK SLICE DESIGN

We now present the formulation of the robust network slice design problem that operates on the layered graph $G_L^k = (V_L^k, A_L^k)$. Binary decision variables $x_a^k, \forall k \in K, a \in A_L^k$ determine whether the demand k is routed through arc a. Integer variables $y_{vf}, \forall f \in F, v \in V(f)$ specify the number of virtual modules assigned to the network functions. Integer variables $y_v, \forall v \in V$ and $y_e, \forall e \in E$ indicate the number of capacity modules installed on the nodes and edges, respectively.

$$\begin{aligned}
\min \quad & \sum_{v \in V} \gamma_v y_v + \sum_{f \in F} \sum_{v \in V(f)} \gamma_v^0 c_f y_{vf} + \sum_{e \in E} \gamma_e y_e + \\
& + \sum_{e \in E} \sum_{k \in K} \sum_{a \in A_L^{k,E}(e)} \gamma_e^0 \bar{d}^k x_a^k + \sum_{e \in E} \gamma_e^0 \mathrm{DEV}_e^\Gamma(x, \hat{d}) && \text{(1a)} \\
\text{s.t.} \quad & \sum_{a \in \delta_v^+} x_a^k - \sum_{a \in \delta_v^-} x_a^k = b^k && \forall v \in V_L^k, k \in K \quad \text{(1b)} \\
& \sum_{k \in K: f \in F^k} \sum_{a \in A_L^{k,V}(f,v)} \bar{d}^k x_a^k + \mathrm{DEV}_{vf}^\Gamma(x, \hat{d}) \le c_f y_{vf} \\
& && \forall f \in F, v \in V(f) \quad \text{(1c)} \\
& \sum_{f \in F: v \in V(f)} c_f y_{vf} \le c_v^0 + c_v y_v && \forall v \in V \quad \text{(1d)} \\
& \sum_{k \in K} \sum_{a \in A_L^{k,E}(e)} \bar{d}^k x_a^k + \mathrm{DEV}_e^\Gamma(x, \hat{d}) \le c_e^0 + c_e y_e && \forall e \in E \quad \text{(1e)} \\
& x_a^k \in \{0,1\}, y_{vf}, y_v, y_e \in \mathbb{Z}_{\ge 0} && \text{(1f)}
\end{aligned}$$

Objective function (1a) minimises the cumulative costs of substrate resource utilisation and potential capacity installations required to host a network slice. Constraints (1b) represent the flow conservation at each vertex, where $b^k = 1$ if $v = s^k$, $b^k = -1$ if $v = t^k$, else 0. Constraints (1c), (1d), and (1e) denote the capacity requirements at the network functions, nodes and edges, respectively.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322056

3 BIASED RANDOM-KEY GENETIC ALGORITHMS

Built on the principle of natural selection, biased random-key genetic algorithms aim to find near-optimal solutions to a wide array of combinatorial optimisation problems [3]. As an alternative to solving the robust network slice design problem using commercial Mixed Integer Programming solvers, we propose a modified version of the biased random-key genetic algorithm.

An initial population comprising a set of random-key vectors $\mathcal{P}$ is constructed. A vector p encodes a candidate solution for the robust NSDP using a sequence of random keys $\langle p_k \mid k \in K \rangle$, where a key p_k is drawn uniformly at random from the set $\mathcal{R}_k$, where $\phi\colon P_k \xrightarrow{\sim} \mathcal{R}_k$. At every generation, we evaluate the fitness of the vectors by applying a problem-specific decoder. The vectors are sorted in the ascending order of their fitness values, and the population is then bifurcated into a set of elite $\mathcal{P}_e$ and non-elite $\mathcal{P}_{\bar{e}}$ individuals. As a part of evolution, the elite partition is copied onto the new population of the next generation unchanged. Furthermore, a set of mutants $\mathcal{P}_m$ are added to the population, where each mutant is again a random-key vector representing a candidate solution. The population is completed by adding $|\mathcal{P}| - |\mathcal{P}_e| - |\mathcal{P}_m|$ individuals, where each individual is generated by applying a crossover function. Two individuals, p_M and p_N are picked randomly from the population in the current generation and are combined using parametrised uniform crossover to produce an offspring, with $\varrho > 0.5$ being the probability with which an offspring inherits the key of the individual p_M. The algorithm continues to evolve the population until a restart criterion is not satisfied. Subsequently, a restart strategy is applied which involves re-initialising the non-elite partition $\mathcal{P}_{\bar{e}}$ with a fresh set of random-key vectors after a fixed number of (non-improving) generations in order to escape local optima.

Decoder for the robust NSDP. A feasible solution to the robust NSDP is extracted from a random-key vector by applying the function ϕ^{-1} to each of its random keys p_k to get the corresponding encoded paths. Following this, the arcs comprising the encoded paths are activated to obtain the fixing of the variables x_a^k. The number of capacity modules required to support the network slice are computed by identifying the total volume of traffic in the network slice for the worst-case realisation of the uncertain traffic demands, thereby obtaining a feasible solution to the robust NSDP. Finally, the fitness i.e., the value of the extracted solution is computed using equation (1a).

4 EXPERIMENTAL RESULTS

We validate the performance of the proposed solution methodologies using realistic problem instances from the SNDlib [4]. The biased random-key genetic algorithm is configured as follows: We set the size of the population to 50 individuals, the percentage of elites and mutants to 10% and 5% of the population, respectively. The crossover probability is set to 0.6 and a restart parameter of 250 iterations is used to reset the non-elite partition. Candidate paths $P_k, \forall k \in K$ are computed using the familiar $\mathscr{k}-$ shortest paths algorithm, where $\mathscr{k} = 50$. The mutants, however, are restricted to work on a much smaller candidate pathset ($\mathscr{k} = 3$) in order to accelerate the search. We evaluate two variants of the algorithm that differ from each other in terms of how the individuals are chosen from the population to perform the crossover. In the first variant BRKGA-I, p_M is drawn at random from $\mathcal{P}_e$ whereas p_N is chosen from the entire population $\mathcal{P}$. For the crossover in the second variant BRKGA-II, vectors p_M and p_N are drawn at random from $\mathcal{P}_e$ and $\mathcal{P}_{\bar{e}}$, respectively.

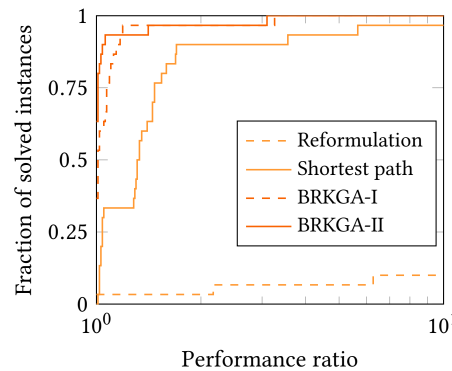

Figure 1: Performance profile of the solution methodologies.

Figure 1 depicts the computational performance of the proposed solution methods by means of their performance profiles. We observe from the profile plot that, for any given problem instance, BRKGA-II outperforms the other methods with a probability of 0.63 whereas the probability with which the compact reformulation performs the best on any given instance is 0.03. Furthermore, we notice that both BRKGA-I and BRKGA-II are still the most efficient of the algorithms despite relaxing the quality of the solution to be within a factor of 4 of the best performing solution method. Meanwhile, the algorithm based on shortest path routings functions reasonably well with a probability of 0.93 to return solutions of similar quality.

5 CONCLUSION

We revisit the robust network slice design problem using the notion of layered graphs. As the resulting problem proved computationally challenging for state-of-the-art Mixed Integer Programming solvers, we devise biased random-key genetic algorithms for the robust network slice design problem. For a vast majority of the considered problem instances, the quality of the solutions yielded by the two variants of the algorithm were exceptionally high in comparison to those obtained from the commercial MIP solver.

REFERENCES

[1] A. Baumgartner, T. Bauschert, A. M. C. A. Koster, and V. S. Reddy. 2017. Optimisation Models for Robust and Survivable Network Slice Design: A Comparative Analysis. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 1–7.

[2] Christina Büsing and Fabio D'Andreagiovanni. 2012. New Results about Multi-band Uncertainty in Robust Optimization. In *Experimental Algorithms*, Ralf Klasing (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 63–74.

[3] José Fernando Gonçalves and Mauricio G. C. Resende. 2011. Biased random-key genetic algorithms forÂăcombinatorial optimization. *Journal of Heuristics* 17, 5 (01 Oct 2011), 487–525.

[4] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski. 2010. SNDlib 1.0-Survivable Network Design Library. *Networks* 55, 3 (2010), 276–286.

[5] Matthias Rost and Stefan Schmid. 2016. Service Chain and Virtual Network Embeddings: Approximations using Randomized Rounding. *CoRR* abs/1604.02180 (2016).