Solving the Shortest Path Labeling Problem with Reactive GRASP for In-Network Intrusion Detection

Edoardo Scalzo[1], Floriano De Rango[2,3], Francesca Guerriero[1], Antonio Iera[2,3], Mattia Giovanni Spina[2,3], and Giovanni Terremoto[4]

[1]Department of Mechanical, Energy and Management Engineering, University of Calabria, Rende (CS) 87036, Italy, ✉ {edoardo.scalzo, francesca.guerriero}@unical.it
[2]Department of Computer Engineering, Modeling, Electronics and Systems, University of Calabria, Rende (CS) 87036, Italy, ✉ {f.derango, antonio.iera, mattiagiovanni.spina}@dimes.unical.it
[3]CNIT - National Inter-University Consortium for Telecommunications, Parma, (PR) 43124, Italy
[4]Department of Mathematics and Computer Science, University of Calabria, Rende (CS) 87036, Italy, ✉ {giovanni.terremoto}@unical.it

Data-plane programmability and the adoption of 5G/6G architectures are fostering the development of *in-network* services, where part of the processing is performed directly by network devices. In *distributed intrusion detection* scenarios, a Machine Learning model can be decomposed into multiple lightweight components (functions) and distributed across programmable nodes, which cooperate to classify flows ([3], [5]). This yields a natural operational constraint: for some source–destination pair in the network, the routed path must traverse at least one instance of each required function, while minimizing overall latency. From an operations research standpoint, these constraints fall within the broader literature on optimization for network design and routing problems, where exact and (meta)heuristic techniques have been widely investigated (see, among others, the contributions [4], [7], [6], [1], and [2]).

Motivated by this use case, we define the *Shortest Path Labeling Problem* (SPLP) on a graph to be labeled $G_L = (G, L)$, where $G = (V, E, w)$ is an undirected, connected, weighted graph with $w : E \to \mathbb{R}_{\geq 0}$ and L is a set of labels. Given a set of pairs $A \subseteq V \times V$, a label cost $c : L \to \mathbb{N}_0$, and a coverage requirement $k : L \to \mathbb{N}_0$, SPLP asks to determine a labeling function $\mathfrak{f} : V \to L$ and, for each pair $(s, t) \in A$, a simple path P_{st} from s to t such that at least k_l required nodes are traversed for each label l.

The objective is to minimize a two-term cost function that captures the trade-off between *deployment* and *routing*. The first term accounts for the labeling (deployment) cost, i.e., the total cost incurred by assigning labels to nodes, computed as the sum of the costs $c(\mathfrak{f}(v))$ over all $v \in V$. The second term accounts for the routing cost induced by the labeling, i.e., the sum of the edge weights along the selected feasible paths P_{st} for all query pairs $(s, t) \in A$.

The core of the work is the development of a *Reactive GRASP* ([9], [10], and [8]) with a restart strategy to tackle large size instances. At each GRASP iteration, a semi-greedy constructive phase builds a feasible solution by repeatedly selecting a path from a Restricted Candidate List (RCL) and adding labels where coverage is missing. The RCL is driven by a priority score that targets the "most critical" paths (i.e., those that most severely violate label coverage), so that early iterations focus on eliminating infeasibilities with minimal additional cost. After construction, a local search phase applies a sequence of neighborhood moves (relabeling/delabeling/swap) to reduce the objective value while preserving feasibility; when feasibility is lost, a dedicated *repair* step restores coverage.

The *reactive* component adapts, during the search, the parameter α controlling the size of the RCL and hence the greediness/randomness balance. Specifically, we maintain a discrete set of candidate α values and update their selection probabilities based on the quality of the solutions obtained with each value; better-performing α values are sampled more frequently, while still preserving diversification. Finally, a restart mechanism is triggered after a prescribed number of iterations without improvement to mitigate stagnation. Furthermore, we considered two boolean options that affect the construction phase: (i) *RCLWeights* controls how a path is evaluated to prioritize selection in the RCL. Specifically, the RCL score uses either the weighted length (sum of edge weights) or only the number of edges (#hop), ignoring weights; (ii) *HighestWeightedDegree* controls the choice of the node to label along the selected path. In particular, one may favor the node maximizing the metric given by the average weighted degree of its incident edges, or select the node with maximum degree. In summary, *RCLWeights* impacts path selection (RCL), whereas *HighestWeightedDegree* impacts the selection of nodes to be labeled.

Path evaluation and updates are performed via a label-constrained variant of Dijkstra based on binary masks. Given the labeling function $\mathfrak{f}$, the algorithm works on an expanded state space in which each state is a pair (v, mask), where $v \in V$ is the current node and $\mathsf{mask} \in \{0, \ldots, 2^{|L|-1} - 1\}$ encodes (in binary) which labels have already been visited along the partial path. Let $requiredLabelMask = (2^{|L|-1} - 1)_2$ denote the mask with all required labels set to 1, excluding the special "unlabeled" value, if present. The distance label $d(v, \mathsf{mask})$ stores the minimum cost to reach v from s while having visited exactly the set of labels represented by mask. A priority queue maintains tuples (v, d, mask); when relaxing an edge (v, u), the algorithm updates $\mathsf{mask}' = \mathsf{mask}\,\mathrm{OR}\,\mathfrak{f}(u)$ and pushes the improved state if $d(u, \mathsf{mask}')$ decreases. The target t is accepted only when reached with $\mathsf{mask} = requiredLabelMask$, ensuring that feasibility, label coverage, is enforced during the shortest-path search rather than checked a posteriori.

In addition to GRASP, we report an experimental comparison with (i) new results obtained via a BRKGA and (ii) the exact model, both developed in a previous work ([11]), as quality baselines and to analyze time–optimality trade-offs. Moreover, we conduct a sensitivity analysis on GRASP parameters, including the boolean choices *RCLWeights* and *HighestWeightedDegree*, the number of local search moves, and the parameters determining the RCL size, in order to quantify their impact on solution quality and running time. The analysis also evaluates both computational cost and achieved objective value as key network features vary, including graph density, number of source–target pairs, number of labels, and number of nodes.

References

[1] Bernardetta Addis, Dallal Belabed, Mathieu Bouet, and Stefano Secci. Virtual network functions placement and routing optimization. In *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, pages 171–177. IEEE, 2015.

[2] Bernardetta Addis, Giuliana Carello, and Meihui Gao. On a virtual network functions placement and routing problem: Some properties and a comparison of two formulations. *Networks*, 75(2):158–182, 2020.

[3] Houssem Jmal, Firas Ben Hmida, Nardine Basta, Muhammad Ikram, Mohamed Ali Kaafar, and Andy Walker. Spgnn-api: A transferable graph neural network for attack paths identification and autonomous mitigation. *IEEE Transactions on Information Forensics and Security*, 19:1601–1613, 2023.

[4] Ivana Ljubić, Ahlam Mouaci, Nancy Perrot, and Éric Gourdin. Benders decomposition for a node-capacitated virtual network function placement and routing problem. *Computers & Operations Research*, 130:105227, 2021.

[5] Daniele Moro, Giacomo Verticale, and Antonio Capone. A framework for network function decomposition and deployment. In *2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020*, pages 1–6. IEEE, 2020.

[6] Ahlam Mouaci, Éric Gourdin, Ivana LjubiĆ, and Nancy Perrot. Virtual network functions placement and routing problem: Path formulation. In *2020 IFIP Networking Conference (Networking)*, pages 55–63. IEEE, 2020.

[7] Ahlam Mouaci, Éric Gourdin, Ivana Ljubić, and Nancy Perrot. Two extended formulations for the virtual network function placement and routing problem. *Networks*, 82(1):32–51, 2023.

[8] M.G.C. Resende and C.C. Ribeiro. Restart strategies for grasp with path-relinking heuristics. *Optimization Letters*, 5:467–478, 08 2011.

[9] M.G.C. Resende and C.C. Ribeiro. *Optimization by GRASP*. Springer, 2016.

[10] M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures: advances and extensions. pages 169–220, 2019.

[11] Mattia Giovanni Spina, Edoardo Scalzo, Floriano De Rango, Francesca Guerriero, and Antonio Iera. Optimal in-network distribution of learning functions for a secure-by-design programmable data plane of next-generation networks. *arXiv preprint arXiv:2411.18384*, 2024.