Paths, Proofs, and Perfection: Developing a Human-Interpretable Proof System for Constrained Shortest Paths

Konstantin Sidorov[1] [0009-0009-0655-4200], Gonçalo Homem de Almeida Correia[2] [0000-0002-9785-3135], Mathijs de Weerdt[1] [0000-0002-0470-6241], and Emir Demirović[1] [0000-0003-1587-5582]

[1] Algorithmics Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology
{k.sidorov, m.m.deweerdt, e.demirovic}@tudelft.nl
[2] Faculty of Civil Engineering and Geosciences, Delft University of Technology
g.correia@tudelft.nl

Abstract. People want to rely on optimization algorithms for complex decisions but verifying the optimality of the solutions can then become a valid concern, particularly for critical decisions taken by non-experts in optimization. We propose a proof system for shortest-path problems with external constraints, which gives a set of logical rules to derive new facts about feasible solutions. The key trait of the proposed proof system is that it specifically includes high-level graph concepts within its reasoning steps (such as connectivity or path structure), in contrast to using linear combinations of model constraints. Using our proof system, we can provide a step-by-step, human-auditable explanation showing that the path given by an external solver cannot be improved. We also propose a proof search procedure that specifically aims to find small proofs of this form that proceeds similarly to A* search. We evaluate our proof system on constrained shortest path instances generated from real-world road networks and experimentally show that we may indeed derive more interpretable proofs compared to an integer programming approach, in some cases leading to much smaller proofs.

Keywords: Explainability · Shortest path problems · Proof systems

This is an encore abstract of the homonymous paper [9] from AAAI-24.

Motivation. Combinatorial optimization has achieved remarkable success in various practical domains; despite most of such problems being NP-complete, many approaches can efficiently solve them in practice [7]. The downside of this success is that the software for solving these problems is typically very sophisticated, and testing such software is a non-trivial task [5].

One way to address this issue is to augment the software output with a *certificate* – a logical derivation supporting the optimality/infeasibility claim. One of the most celebrated examples of this idea is proof logging in Boolean SAT solving, [1] and related approaches are known for mixed-integer programming [3] and pseudo-Boolean solving [6]. Proof systems of this type are generic enough to support various application domains, however, the downside is that the produced proofs are impenetrable for a human auditor.

Contribution. In this work, we address this concern for *constrained shortest path problems* in which the path must satisfy some arbitrary set of conditions. We design a proof system for the constrained shortest path problems that exploit knowledge specific to the graph-theory domain, such as the removal of too-distant vertices. This results in proofs that are easier to understand for humans, which opens a new opportunity of validating the verdict of a solver by a human observer. We also propose an algorithm for proof search and evaluate it against the state-of-the-art implementation for a proof system [3] for integer programming, with favorable results on instances with mandatory vertex constraints.

Results. We compare the proofs produced by our approach with the state-of-the-art certified MIP approach [3] with respect to the proof width *width*, the number of leaves in the proof tree, which translates to comparing the number of cases introduced to support the optimality conclusion.

For the evaluation, we used constrained shortest path problems with two different types of side constraints: (a) *mandatory vertices* that require visiting some sets of vertices at least once as defined in [4], and *resource constraints* which impose a knapsack capacity constraint on chosen edges as defined in [8]. We generate the underlying graphs by using the OpenStreetMap data [2]. Our approach has successfully constructed proofs within a 60-second timeout on 96% of mandatory-vertex instances and 92% of resource-constrained instances, on par with the baseline approach. Figure 1 summarizes the comparison of proof widths for the baseline and for our approach; the major highlight is that our approach produces shorter proofs for feasible mandatory-vertex instances in 72.8% of the runs.

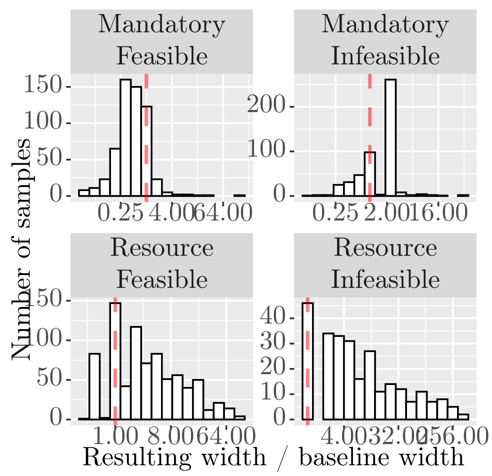

Fig. 1. Distribution of the ratio of produced proof depth and baseline proof depth.

Resource constraint does not exhibit improvement in the proof size, which is caused by the fact that the baseline solves most (> 95%) of such instances with at most one level of branching. Nonetheless, the proofs found by our approach are reasonably sized and could be used for human inspection.

Acknowledgments. Konstantin Sidorov is supported by the TU Delft AI Labs program as part of the XAIT lab.

References

1. Biere, A., Heule, M., van Maaren, H. (eds.): Handbook of Satisfiability. No. volume 336 in Frontiers in Artificial Intelligence and Applications, IOS Press, Amsterdam ; Washington, DC, second edition edn. (2021)
2. Boeing, G.: OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. Computers, Environment and Urban Systems **65**, 126–139 (Sep 2017). https://doi.org/10.1016/j.compenvurbsys.2017.05.004
3. Cheung, K.K.H., Gleixner, A., Steffy, D.E.: Verifying Integer Programming Results. In: Eisenbrand, F., Koenemann, J. (eds.) Integer Programming and Combinatorial Optimization. vol. 10328, pp. 148–160. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-59250-3_13
4. de Uña, D., Gange, G., Schachte, P., Stuckey, P.J.: A Bounded Path Propagator on Directed Graphs. In: Rueher, M. (ed.) Principles and Practice of Constraint Programming, vol. 9892, pp. 189–206. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-44953-1_13
5. Gillard, X., Schaus, P., Deville, Y.: SolverCheck: Declarative Testing of Constraints. In: Schiex, T., de Givry, S. (eds.) Principles and Practice of Constraint Programming. vol. 11802, pp. 565–582. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-30048-7_33
6. Gocht, S., McCreesh, C., Nordström, J.: VeriPB: The easy way to make your combinatorial search algorithm trustworthy. In: Workshop From Constraint Programming to Trustworthy AI at the 26th International Conference on Principles and Practice of Constraint Programming (CP'20). Paper Available at http://www. Cs. Ucc. Ie/Bg6/Cptai/2020/Papers/CPTAI_2020_paper_2. Pdf (2020)
7. Korte, B., Vygen, J.: Combinatorial Optimization, Algorithms and Combinatorics, vol. 21. Springer Berlin Heidelberg, Berlin, Heidelberg (2000). https://doi.org/10.1007/978-3-662-21708-5
8. Lozano, L., Medaglia, A.L.: On an exact method for the constrained shortest path problem. Computers & Operations Research **40**(1), 378–384 (Jan 2013). https://doi.org/10.1016/j.cor.2012.07.008
9. Sidorov, K., Correia, G.H.d.A., De Weerdt, M., Demirović, E.: Paths, proofs, and perfection: Developing a human-interpretable proof system for constrained shortest paths. AAAI **38**(18), 20794–20802 (Mar 2024)