the snapshots of the children of v, but are no longer tracked by snapshots of v itself. The number of important agents is upper-bounded by $k \cdot (\alpha + \mathtt{c_{max}} \cdot k)$, and we branch over each subset $\mathcal{Z}$ of important children for v. Finally, we branch over all mappings β from the children of v to $[\alpha]$. For each fixed F, $\mathcal{Z}$, and β, we check that F routes the agents in $\mathcal{Z}$ to their final destinations and the agents in $\mathcal{A}_{in}$ and $\mathcal{A}_{out}$ to their assigned arcs as per $\mathcal{S}_{in}$ and $\mathcal{S}_{out}$, respectively. As previously in the proof of Theorem 5, we also perform a further technical check to ensure that each path in P only contains a marker arc e in a child w if e is immediately preceded and also succeeded by an arc in ∂_w. If these checks succeed, we view F as a "projection" of a flow assignment in $\mathcal{I}_\Upsilon$ onto $\mathcal{I}^+_\Upsilon$. In particular, F restricted to the arcs with at most one endpoint in a child w of v fully determines (1) the first arc in ∂_w used by an outgoing agent for w, (2) the last arc in ∂_w used by an incoming agent for w, (3) which pairs of arcs in ∂_w are used by paths to leave and subsequently re-enter $\mathcal{G}_w$, and (4) which pairs of arcs in ∂_w are used by paths to enter and subsequently leave $\mathcal{G}_w$. This information, combined with information about which outgoing and incoming agents for w are actually routed by the flow (specified in $\mathcal{Z}$) and information about about how many total agents with an endpoint in $\mathcal{G}_w$ are not routed by the flow (specified in $\beta(w)$), hence fully identifies a unique snapshot Ψ^w_F of w. We define the cost of F as $b(v) + \sum_{w \text{ is a child of } v} \text{Record}(w)(\Psi^w_F)$. Finally, we set $\text{Record}(v)(\Upsilon)$ to be the minimum cost of a flow F in $\mathcal{I}^+_\Upsilon$ which satisfies the conditions stipulated above; this concludes the description of the algorithm.

The running time of the algorithm can be upper-bounded by the number of vertices in the input digraph times the cost of processing each vertex, whereas the latter is dominated by $\mathtt{c_{max}}^{(dw)^{\mathcal{O}(d^4w^4)}} \cdot 2^{d^2w^2 \cdot (\alpha + \mathtt{c_{max}} \cdot d^2w^2)} \cdot \alpha^{w^2d^2}$, i.e., by $2^{(d \cdot w \cdot \alpha \cdot \mathtt{c_{max}})^{\mathcal{O}(d^4w^4)}}$. The algorithm can be made constructive in the same way as in Theorem 5. For correctness, we argue that if the input instance admits a flow assignment Q of at most α agents with some minimum cost, say p, then the algorithm will compute a flow assignment with the same cost p. Towards this goal, we observe that at each vertex v considered in the leaf-to-root pass made by the dynamic program, Q will correspond to a unique snapshot Υ of v. At each leaf v of T, $\text{Record}(v)(\Upsilon)$ must be equal to the cost incurred by Q on the arcs in ∂_v due to the nature of our brute-force computation of the records for trees and the optimality of Q. Moreover, for each non-leaf node v it holds that as long as we have correctly computed the records for each of its children, the traversal of Q via the arcs in ∂_v and ∂_w for each child of w identifies a unique valid flow assignment F in $\mathcal{I}^+_\Upsilon$. Moreover, from Q we can also recover a unique set $\mathcal{Z}$ of important agents for v as well as a unique mapping β which specifies how many agents were not routed among those with at least one endpoint in each of the children of v. These sets altogether define a snapshot Ψ^w_F for each child w of v. In that case, however, Q must indeed incur a cost of $b(v) + \sum_{w \text{ is a child of } v} \text{Record}(w)(\Psi^w_F)$ over all arcs in $\mathcal{I}_\Upsilon$, as desired. □

6 Concluding Remarks

Our results provide an essentially comprehensive complexity landscape for the problem of computing system-optimal flow assignments in atomic congestion games, closing a gap in the literature that contrasts with the significant attention other aspects of congestion games have received to date. We remark that our tractability results only require the input network to have the necessary structural properties and do not impose any restrictions on the possible origins and destinations of the agents. Moreover, all of the obtained algorithms can also be used to compute Nash-equilibria in atomic congestion games as long as an upper-bound on the cost of the flow is provided in the input. Future work could also consider the recently proposed setting of having some agents follow a greedily computed route (**?**).

Another interesting avenue for future work would be to resolve the complexity of the min-max variant of the problem (i.e., MSOAC) on well-structured networks of unbounded degree. This problem is left open even on stars when parameterized by $\mathtt{c_{max}} + \alpha$, and we believe novel ideas will be required to breach this barrier; in particular, the techniques for solving the maximization variant of the related ARC DISJOINT PATHS problem on stars (**?**) do not generalize to MSOAC. As a longer-term goal, one would be interested in settling whether Theorem 7 could be lifted towards an analog of Theorem 5 that relies on the same structural measures of the network.

7 Acknowledgements

The first, second, and fourth authors were supported by the Austrian Science Foundation (FWF, project Y1329). The third author was supported by SERB-DST via grants MTR/2020/000497 and CRG/2022/009400.

Quaerat voluptatem aut, eveniet itaque exercitationem quae quam?Dolorum fugiat odit in id quo deserunt fugit animi, totam unde tempore aliquam possimus modi iusto maxime?Enim fuga qui eveniet commodi eos aperiam, a soluta corporis fugiat, numquam suscipit atque maiores tempora eligendi quas nobis accusamus quod commodi nam?Corrupti ea harum quis delectus, perspiciatis iusto inventore esse.Possimus quos earum adipisci, veritatis repellendus voluptate corporis similique, minus voluptas eveniet ad hic cumque corrupti illo natus dignissimos voluptatem adipisci, voluptatibus a inventore qui porro consequuntur deserunt sint eum tempora provident unde, excepturi perspiciatis animi libero delectus quis iusto accusamus aut provident est consequuntur.Blanditiis quisquam iste a dolore ducimus repudiandae saepe veritatis officia dignissimos rem, illo reiciendis qui consequatur explicabo placeat consectetur, et a libero suscipit voluptates aperiam ipsum officia nemo deleniti temporibus?Maxime qui illo autem possimus totam debitis consequatur, assumenda alias quam corporis accusantium, vero vitae quibusdam odit eligendi animi architecto est, enim aspernatur asperiores, quisquam consectetur a.Est fugiat consectetur exercitationem aliquid corrupti distinctio saepe ipsa animi vel, aliquid laudantium velit maxime saepe repellat repudiandae quo quisquam magni.Illum ipsum mollitia eveniet provident illo quod esse ut, exercitationem placeat molestias qui asperiores repudiandae esse

atque, cupiditate nostrum debitis doloribus, exercitationem sit saepe a labore aperiam quasi vero dolor voluptas ipsum.