



# Comments on: Perspectives on integer programming for time-dependent models

Tom Van Woensel<sup>1</sup>

Published online: 10 May 2019  
© The Author(s) 2019

## 1 Introduction

With great interest, I read the paper *Perspectives on Integer Programming for Time-Dependent Models* (Boland and Savelsbergh 2019). Evidenced by the growing stream of literature (as referenced in their paper), *dynamic discretization discovery* (DDD) is, and will be, a very rich future path for time-dependent models. I very much enjoyed reading the paper and I would like to congratulate Natasha and Martin for presenting this challenging material in a clear and comprehensive manner.

Time-dependent (TD) problems are everywhere and are a more realistic representation of real life than their time-independent (TID) counterpart. Additionally, solutions based on a TID model formulation are usually suboptimal or more likely to be infeasible in a TD world. In this comment, I elaborate on the importance of the DDD and present some additional thoughts. In the vehicle routing research domain, I did some work on the incorporation of time-dependent travel times (Dabia et al. 2013; Franceschetti et al. 2017; Sever et al. 2018; Sun et al. 2018a, b; Van Woensel et al. 2008). Interested readers are also referred to the excellent review paper by Gendreau et al. (2015).

## 2 Importance of dynamic discretization discovery

Time-dependent models are difficult. Using continuous variables leads to weak relaxations. Using binary variables leads to an explosion of the number of variables (Boland and Savelsbergh 2019). In the latter case, the original number of variables (of the time-independent model) needs to be multiplied with the number of time slices to have the needed variables. The variable definition now also identifies on what time unit  $t$  some-

---

This comment refers to the invited paper available at <https://doi.org/10.1007/s11750-019-00514-4>.

---

✉ Tom Van Woensel  
t.v.woensel@tue.nl

<sup>1</sup> Eindhoven University of Technology, Eindhoven, The Netherlands

thing will happen. In a routing context, the vehicle goes from customer  $i$  to customer  $j$  at time unit  $t$ . As noted by Boland and Savelsbergh (2019), the time bucketing is very important (Van Woensel et al. 2008). It has a direct effect on the needed granularity of information changing over time (e.g., speeds or traffic flows), on the amount of decision variables one needs to consider in the programming model, and, consequently on the solution times for the algorithms (both exact and heuristics).

In my opinion, it is for these cases that DDD has a key role: being able to use the finest granularity without all the above-mentioned problems. As such, this indeed allows for solving “a TD model based on a fine discretization without ever fully constructing it” (Boland and Savelsbergh 2019). Good to note is that often the availability of the information sometimes drives or limits the possible granularity that can be used in the modeling.

The price of adding time in the model formulations and the related solution methods is clear. On the other hand, adding time also leads to interesting new explicit decisions related to departure times or waiting at nodes. This gave rise to some literature related to departure time and speed optimization (see, e.g., Franceschetti et al. 2018; Wen et al. 2017), but also waiting at the customer nodes. Interestingly, only very few publications allow for idle waiting after service completion as a strategy to cope with time-dependent conditions, e.g., avoiding congestion which varies over time (Franceschetti et al. 2017). This issue is also pointed out by the authors as a challenging setting for the DDD.

### 3 Heuristics and dynamic discretization discovery

The authors already discussed, in their *Research Opportunities* section, many extensions that can be considered. In this second part of my comment, I would like to focus on the possible role of DDD in relation to heuristics. Scaling up the proposed algorithms seems promising, but so far only relatively small instances are handled, up to 100 locations for the TD-TSPTW (Boland and Savelsbergh 2019). Combining the DDD into heuristics, either metaheuristics or matheuristics, might overcome this scaling challenge, with a possible price of losing optimality.

In their paper, Visser and Spliet (2017) introduce some new methods for efficiently evaluating moves in a neighborhood search for a time-dependent routing problem. Evaluating several neighborhoods is computationally expensive in a TD setting. The authors develop a new tree-based data structure to improve the complexity of computations and memory usage. On 1000 customer instances, the authors show significant speedups in the algorithm. The expensive feasibility and cost calculation of a route can be done in  $\mathcal{O}(np \log n)$  time rather than in  $\mathcal{O}(n^2 p)$  time, where  $n$  is the number of customers on the route and  $p$  is the number of breakpoints coming from the time-dependent travel time functions.

The latter parameter  $p$  is actually a representation of the granularity of the time dependency considered (i.e., buckets). An interesting question would be how DDD could be used to determine the  $p$  dynamically, potentially decreasing the evaluation times further. Note that a similar question arises when solving the TD-VRPTW in an exact manner using traditional branch-and-price methodologies (Dabia et al. 2013).

Here, dominance rules are extended, which consider the predefined time discretizations (via the ready time functions). Comparing two partial paths, potentially having a different time discretization (originating from DDD), is not straightforward.

In most (if not all) TD papers, it is assumed that the time discretization is given a priori, leading to the above discussed problems. Alternatively, using DDD in (meta-) heuristics as an additional neighborhood function, where the time discretization could be refined as part of the evaluation, seems interesting. Open questions remain on how to do this exactly, but also how to coordinate discretizations over the iterations and/or neighborhoods. In this case, it could be foreseeable that every neighborhood has possibly its own time refinement, necessitating some overall structure, or aligning time somehow.

Finally, coming up with a more general DDD policy could be an interesting research venue as well, but might only be feasible for specific problem families. Similar to Ulmer et al. (2019), who, for a single-vehicle routing problem with stochastic service requests, incrementally improved the granularity of their lookup table depending upon the received rewards, some learning aspects could be incorporated, leading to a policy (Powell 2019). Specifically, reinforcement learning techniques (Sutton and Barto 2018), like actor–critic approaches, could help to develop a good policy for DDD. The critic would then measure how good the discretization is (e.g., based on the obtained bounds) and the actor would suggest a change in the discretization (i.e., the discovery of new time points in Fig. 5 of Boland and Savelsbergh 2019). Rewards could also be related to the number of repairs needed. A disadvantage could be the training time needed with this type of methodology. But once the policy is there, it would help to instantly identify a good discretization. Some first steps in terms of combining reinforcement learning in a time-dependent environment (with given time discretizations) can be found in Mao and Shen (2018).

## 4 Concluding remarks

All in all, I very much enjoyed reading the paper and its presented material. It has the potential to be a large research area applying the proposed methodology to a variety of time-dependent problems. This paper gives a good and clear summary of the interested reader aiming to start working in this field. Moreover, it provides valuable pointers for future research.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

Boland NL, Savelsbergh WP (2019) Perspectives on integer programming for time-dependent models. TOP. <https://doi.org/10.1007/s11750-019-00514-4>

- Dabia S, Ropke S, Van Woensel T, De Kok AG (2013) Branch and price for the time-dependent vehicle routing problem with time windows. *Transp Sci* 47(3):380–396
- Franceschetti A, Demir E, Honhon D, Van Woensel T, Laporte G, Stobbe M (2017) A metaheuristic for the time-dependent pollution-routing problem. *Eur J Oper Res* 259(3):972–991
- Franceschetti A, Honhon D, Laporte G, Van Woensel T (2018) A shortest-path algorithm for the departure time and speed optimization problem. *Transp Sci* 52(4):756–768
- Gendreau M, Ghiani G, Guerriero E (2015) Time-dependent routing problems: a review. *Comput Oper Res* 64:189–197
- Mao C, Shen Z (2018) A reinforcement learning framework for the adaptive routing problem in stochastic time-dependent network. *Transp Res Part C Emerg Technol* 93:179–197
- Powell WB (2019) A unified framework for stochastic optimization. *Eur J Oper Res* 275(3):795–821
- Sever D, Zhao L, Dellaert N, Demir E, Van Woensel T, De Kok AG (2018) The dynamic shortest path problem with time-dependent stochastic disruptions. *Transp Res Part C Emerg Technol* 92:42–57
- Sun P, Veelenturf LP, Dabia S, Van Woensel T (2018) The time-dependent capacitated profitable tour problem with time windows and precedence constraints. *Eur J Oper Res* 264(3):1058–1073
- Sun P, Veelenturf LP, Hewitt M, Van Woensel T (2018) The time-dependent pickup and delivery problem with time windows. *Transp Res Part B Methodol* 116:1–24
- Sutton RS, Barto AG (2018) Reinforcement learning, an introduction. MIT Press, Cambridge
- Ulmer MW, Goodson JC, Mattfeld DC, Hennig M (2019) Offline online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transp Sci* 53:185–202
- Van Woensel T, Kerbache L, Peremans H, Vandaele N (2008) Vehicle routing with dynamic travel times: a queueing approach. *Eur J Oper Res* 186(3):990–1007
- Visser TR, Spliet R (2017) Efficient move evaluations for time-dependent vehicle routing problems (EI2017-23). *Econometric Institute Research Papers*. <https://repub.eur.nl/pub/100852/>
- Wen M, Pacino D, Kontovas CA, Psaraftis HN (2017) A multiple ship routing and speed optimization problem under time, cost and environmental objectives. *Transp Res Part D Transp Environ* 52:303–321

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.