Fast calculation of dynamic traffic assignment by parallelised network loading algorithm

Takamasa Iryo[1], Junji Urata[1], Kazuki Fukuda[1],
Genaro Peque Jr.[1], Lalith Wijerathne[2], and Wasuwat Petprakob[2].

[1] Graduate School of Engineering, Kobe University
[2] Earthquake Research Institute, The University of Tokyo

Considering the time-dependent property of a traffic flow and its congestion in an urban road network, it is natural to employ the dynamic traffic assignment (DTA) technique to evaluate congestion and queues on a road network. A huge number of existing studies have been accumulated, while practical implementations are still not popular, especially for evaluating a large network by applying user-equilibrium criteria such as Wardrop's first principle, called dynamic user equilibrium (DUE). The delay of the practical implementation of DUE would be caused by several reasons; e.g. difficulties in collecting time-dependent demand data, issues on solution properties such as uniqueness (Iryo, 2013), and computational burden caused by time-dependent structure of the problem.

Solving DUE incurs two major calculation tasks, namely: (1) calculations of dynamic flow propagations and (2) shortest path searches. Both must be time-dependent, implying that the calculation must be performed for all the time steps, which apparently demand huge computation time that is hundreds or thousands times greater than that for static user-equilibrium assignment problems. While both of them are important issues for the speed-up of the DUE calculations, this study mostly focuses on the first issue, which is an emerging topic in the area of transport studies (e.g. Himpe et al, 2016).

Using a high performance computer is perhaps the most straightforward approach to resolve the computationally intensive problem, while we need to construct a solution algorithm suitable for parallelism to utilise such a computer. Recent computers gain a power by combining a number of CPUs (central processing units). Even a typical desktop server has multiple-core processors enabling us to run dozens of processes in parallel. On the other extreme end, the distributed parallel super computers combine so many CPUs (e.g. more than 80,000 in K computer), each consisting of multiple CPU cores, through a high speed network to obtain their giant power. Actually, the performance of a single CPU is equivalent to a typical desktop server, hence we have to develop suitable parallel algorithms which split the involved main task into sub-tasks. Moreover, as the speed of the network between CPUs is much slower than the speed of data transmissions in each CPU, we need to carefully consider how to divide the calculation task so as to minimize the need of data exchange via the network. This makes it a challenging task to utilize a modern high performance computer to solve DUE.

A number of studies and commercial software packages implement parallelism for DTA (e.g. Rickert and Nagal, 2001; Wang et al., 2014), while typical implementations are subject to frequent communications between CPU cores. For the calculation of dynamic flow propagations, they basically divide a DTA problem into several sub-networks so that each core can calculate flow propagations of the links in each sub-network in parallel. However, it is apparent that the flow propagations between sub-networks continuously occur during the calculation, forcing the CPU cores to communicate frequently. Communicating data between cores incur long latency time when the transfer is set up. Hence, it is necessary to transfer a larger chunk of data at lesser frequency to communicate in a reasonable speed. Regarding the calculation of dynamic flow propagations, Wang et al. (2014) performed it by a heuristic method using the asynchronous iterative approach.

This study also employs the asynchronous iterative approach to reduce the frequency of communications. A major point of the proposed method is that it utilises the variational theory (Newell, 1993; Daganzo, 2005) to construct a simpler and more robust algorithm that can be implemented much easier. This algorithm also incorporates the shortest-path search calculation so that the vehicles' always select shortest (or near-shortest) routes during the calculation. This process would represent the drivers' day-to-day adjustment process of route choices, which would let the system be in equilibrium.

The following is the outline of the proposed method:

1. Let the network be empty (i.e. no vehicle).
2. Select a certain number of vehicles (say, 5%), which have been loaded or yet been loaded.

3. Find shortest routes for the selected vehicles based on current link travel times.
4. Unload these vehicles from the original routes (if they have been loaded already) and reload them to the new routes.
5. Update travel times of all vehicles loaded to each link. This calculation is independently performed for each link using the continuous (fluid-based) traffic flow model (each vehicle is converted to a continuous traffic flow for a certain time duration). Consistency between links are not considered. Exit times of vehicles on each link is updated using the calculation results.
6. Send information on exit times of all vehicles from a link to the downstream links. Then, each link updates vehicles' entry times according to this information.
7. Repeat steps 5-7 till entry and exit times of each vehicle between any successive pair of links on its route are the same.
8. Calculate backward wave of each link to detect whether queue in the link is overflown or not. If so, let the upstream links know that the incoming capacity of the link is restricted to avoid the overflow of the queue at the next iteration.
9. Go back to Step 2.

Step 5 demands a considerable amount of the calculation time. Because the calculation of Step 5 can be performed independently on each link, this process can be parallelised effectively once the links of the network are properly assigned to each CPU core. A state-of-the art technique is necessary for an effective parallelisation of this process. This will be provided in a different presentation of the symposium (Petprakob et al., 2017).

Repeating this procedure, it is expected that every vehicles selects their (near-) shortest routes and no overflow of the queue exists on any link, i.e. calculated flow propagations are consistent with the traffic flow theory.

In the presentation at the symposium, we will provide theoretical analyses to explain why and how the proposed method works. A case study of the Kanto area network (consisting of approx. 400,000 links) will be also shown.

Acknowledgements
This research is a part of the project developing 'system for integrated simulation of earthquake and tsunami hazard and disaster', which is the priority issue No. 3 of the 'development and research of applications on post K for social and scientific issues' (Ministry of Education, Culture, Sports, Science and Technology). A part of the results of this paper was obtained by the supercomputer K of RIKEN (hp160224 and hp170271).

References
Daganzo, C. F. (2005). A variational formulation of kinematic waves: basic theory and complex boundary conditions. Transportation Research Part B, **39**(2), 187–196.
Himpe, W., Corthout, R., and Tampère. M. J. C. (2016). An efficient iterative link transmission model. Transportation Research Part B, **92**, 170-190.
Iryo, T. (2013). Properties of dynamic user equilibrium solution: existence, uniqueness, stability, and robust solution methodology. Transportmetrica B: Transport Dynamics, **1**(1), 52–67.
Newell, G. F. (1993). A simplified theory of kinematic waves in highway traffic, Part I: General theory. Transportation Research Part B, **27**(4), 281–287.
Rickert, M., & Nagel, K. (2001). Dynamic traffic assignment on parallel computers in TRANSIMS. Future Generation Computer Systems, **17**(5), 637–648.
Petprakob, W., Wijerathne L., Iryo, T., Urata, J., and Fukuda, K. (2017). On the implementation of high performance computing extension for day-to-day traffic assignment, ISTS & IWTDCS 2018.
Wang, Y., Schuppen, J. H. Van, & Vrancken, J. (2014). Simulation Modelling Practice and Theory On-line distributed prediction of traffic flow in a large-scale road network. Simulation Modelling Practice and Theory, 47, 276–303.