Algorithms for Non-Separability Problems: The Departure Time Choice

Verstraete Jeroen*[1], Tampère M.J. Chris[1]

[1]Ku Leuven, Belgium

SHORT SUMMARY

Many algorithms for choice predictions for dynamic traffic assignment (DTA), e.g. route choice or departure time choice, shift travelers from an expensive alternative to a cheaper alternative in order to find the equilibrium. These algorithms work well if the cost for the alternative is mainly dependent on the alternative itself, meaning that when adding/removing travelers to the alternative, the cost increases/decreases respectively. When this is not the case, these algorithms do not converge smoothly (Dafermos, 1980). In this abstract, some examples are given where these non-separability problems occur in DTA. A generic algorithm on how to solve this kind of problem more efficiently is sketched. We demonstrate the convergence of our approach on a simple bottleneck with departure time choice.

Keywords: Algorithms, Convergence, Departure Time Choice, Dynamic Traffic Assignment, Route Choice

1. INTRODUCTION

Many algorithms for choice predictions for dynamic traffic assignment (DTA), e.g. route choice or departure time choice, shift travelers from an expensive alternative to a cheaper alternative to find the equilibrium. These algorithms work well if the cost for the alternative is mainly dependent on the alternative itself, meaning that when adding/removing travelers to the alternative, the cost increases/decreases respectively. When this is not the case, these algorithms do not converge smoothly (Dafermos, 1980). Route choice is a typical problem where the cost depends mainly on the alternative itself, exceptions exist as will be discussed later. Departure time choice is different. The cost of departing at a certain time mainly depends on the departures of earlier times, especially if there are queues. These are non-separable problems as different alternatives cannot be seen independent from each other.

Mathematically, for non-separability problems, the diagonal elements of the Jacobian matrix are non-dominant. According to Dafermos (Dafermos, 1980), problems with a symmetrical Jacobian or an asymmetrical Jacobian with dominant diagonal elements converge faster than an asymmetrical Jacobian with non-dominant diagonal elements. Most algorithms take only the value of the current cost into account. If an alternative has a cost that is too high, the probability of choosing the alternative is reduced, without knowing if this alternative has indeed a high influence on its own cost, or testing that the proposed swap yields an improvement.

2. NON-SEPARABILITY EXAMPLES

Intersections

An example of non-separability problems are intersections.

A first example that demonstrates this non-separability is an intersection where there is a main road, which has priority, and some minor road. The delay the travelers on the minor road experience is dependent on its own flow, as the higher the flow, the higher the delay to reach the intersection. But it is also dependent on the flow of the main road. If the flow of the main road is at capacity, there is almost no free space for the minor road, making the delay for them large. Therefore, to increase the delay on the minor road, one can either increase the flow on the minor road itself or on the main road. For route choice algorithms, that try to find the equilibrium costs, this can have an effect.

A second example illustrates the non-separability over different classes. Imagine a signalized intersection, in many real-world cities the signals are dependent on the arrival of public transport. If an approach to the intersection is used by busses and cars, its green time can change real-time such that the delay for busses is minimized. This means that the delay for cars is dependent on the class for busses. Neglecting this when computing the swaps will have an influence on the convergence speed.

Departure Time Choice

A more intuitive example of a non-separability problem is the departure time choice. Travelers have a preferred arrival time at their destination and experience some cost for being too early or too late, called the schedule delay cost. The departure time choice module is responsible for predicting when travelers will depart, given the travel and schedule delay cost. This is first described for a single bottleneck in the well-known Vickrey Model (Vickrey, 1969). Many algorithms which solve this problem discretize time in time steps (Small, 2015), we will do the same in the next example. To illustrate the non-separability of the Jacobian, let us consider a typical peak with one congestion period around the desired arrival time (time step 7 in this example). This situation is sketched in Figure 1. α, β, γ and Q_{max} are the parameters of the Vickrey Model representing the weights for travelling, being too early and being too late, and the bottleneck capacity, respectively.

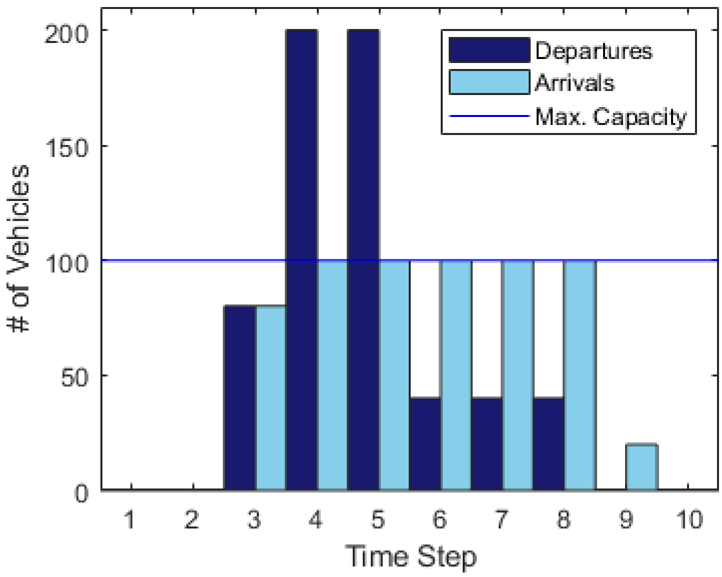

Figure 1 Departures and Arrivals per Time Step of a Simple Rus

Table 1 Sensitivity Matrix of a Simple Example

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	$\frac{\alpha-\beta}{Q_{max}}$	$\frac{\alpha+\gamma}{Q_{max}}$	$\frac{\alpha+\gamma}{Q_{max}}$	$\frac{\alpha+\gamma}{Q_{max}}$	$\frac{\alpha+\gamma}{Q_{max}}$	0	0
5	0	0	0	0	$\frac{\alpha+\gamma}{Q_{max}}$	$\frac{\alpha+\gamma}{Q_{max}}$	$\frac{\alpha+\gamma}{Q_{max}}$	$\frac{\alpha+\gamma}{Q_{max}}$	0	0
6	0	0	0	0	0	$\frac{\alpha+\gamma}{Q_{max}}$	$\frac{\alpha+\gamma}{Q_{max}}$	$\frac{\alpha+\gamma}{Q_{max}}$	0	0
7	0	0	0	0	0	0	$\frac{\alpha+\gamma}{Q_{max}}$	$\frac{\alpha+\gamma}{Q_{max}}$	0	0
8	0	0	0	0	0	0	0	$\frac{\alpha+\gamma}{Q_{max}}$	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

Let us now investigate the structure of the Jacobian, which for this example is given in Table 1. $J_{a,b}$ denotes the extra cost for time step b if an additional traveler would depart in time step a. Note that the cost of the last traveler in a time step is taken as the cost of the time step. The Jacobian is an upper triangular matrix, meaning that a time step has no influence on the cost of earlier time steps. There is a difference between the rows where the corresponding time step has a queue and where it doesn't have a queue. If there is no queue (rows 1,2,3,9,10), the partial derivatives are zero. Note that this assumes the travel time to only be a function of the queue length, otherwise only the partial derivative of the travel time function would be found. If there is a queue (rows 4 to 8), there will be an influence over all later time steps that still have the same queue. In this example, the influence goes to time step 8 as in time step 9 the queue is gone. The size of the influence is depending on the desired arrival time. $J_{4,4}$ and $J_{4,5}$ have different values because currently time step 4 arrives too early while time step 5 arrives too late. If there would be an extra traveler in time step 4, the travelers in time step 4 will now experience more delay but arrive closer to their desired time, while travelers in time step 5 will now be even later on top of the extra delay. The whole sensitivity matrix contains only four different values: (i) extra delay for travelers who were and still are too early, (ii) extra delay for travelers who were too early but now are too late, (iii) extra delay for travelers who were and still are too late and (iv) no extra delay as there is no influence. Due to discretization, in this simple case (ii) does not appear. In the continuous case, some departing time will experience (ii).

As can be seen in Table 1, the diagonal elements are not dominant if a queue is present. Increasing the departure rate of time step 4 increases the costs of time steps 4-8. This illustrates that the departure time choice indeed is a non-separability problem. In the next section, we will suggest a new strategy to solve this problem, with the departure time choice as an example.

3. SOLVING NON-SEPARABILITY PROBLEMS

This new class of algorithms considers all sensitivities when calculating the desired swaps. Most state-of-the-art algorithms swap between the higher cost alternatives to the lower cost alternatives. In this process, no sensitivity information is considered. This new class of algorithms can be divided in two steps. First, it determines the sensitivity for each alternative to each alternative (class dependent). Second, a more desired swap is calculated.

The sensitivity matrix (S) can be calculated with numerical derivation. For each alternative, the cost to all other alternatives is calculated when an extra traveler is added to the current alternative. This requires many calculations and will not be scalable to complex networks. However, this first naïve approach shows the idea behind the new class of algorithms. In later research, the method to calculate the sensitivity matrix can be optimized. For instance, while doing equilibration assignments in DTA, some information about the sensitivity matrix can already be found. For now, assume the naïve approach.

Once the sensitivity matrix is known, the optimal swap vector can be calculated as:

$$S\,X = b = \tilde{c} - c \tag{1}$$

With X the swap vector, b the target vector and $\tilde{c}$ the target cost. Note that Equation (1) does not necessarily have an exact solution, X is considered the vector that results in the least square solution of the equation. During equilibration, there can be imperfect knowledge about the sensitivity matrix. At the same time the equilibrated target vector cannot be computed as the equilibrium costs are unknown. Because of this, the whole framework will need to be run iteratively.

The target cost itself has a big influence on the convergence of the overall method. If the sensitivity matrix is known and all resulting swaps are feasible (see later), the best the method can do is reach the target cost for each alternative. If this target cost is far from equilibrium, so will be the new state after the proposed swaps. In the full paper, we go in more detail about some good and bad approaches for the target cost.

Once the target cost is known, the system of equations needs to be solved. This can be done in multiple ways. There are some feasibility constraints, depending on the concrete problem, when solving Equation (1). Either these constraints are added before solving the system of equations, resulting in a more complex problem but where the resulting swap vector is always feasible. Or, one can solve the system of equations directly and project that solution back to the feasible space. Note that a combination of the two can also be made, where only some constraints are added. By working with swaps, the conservation of travelers is always guaranteed. When a projection to the feasible space is needed, there are two options. First, one can project the swap vector. Second, because the swap vector is an internal variable, one could also project the resulting distribution over the alternatives.

4. NUMERICAL EXAMPLE

A forthcoming paper will discuss the exact algorithm and method used to solve the departure time choice model. In this section we want to illustrate how considering the sensitivities can improve the overall convergence.

In this numerical example, a simple case study is taken. It resembles a typical morning rush, all relevant parameters are given in Table 2. The costs found in equilibrium is plotted in Figure 2, while Figure 3 and Figure 4 plot the departures and arrivals per time step and their cumulatives, respectively.

Table 2 Parameter Values for Literature Example

α	$\$10/h$
β	$\$5/h$
γ	$\$15/h$
t^*	$2h$ or Time Step 40
N	$5100\ veh$
Q_{max}	$3533.6\ veh/h$
Δt	$0.05h$

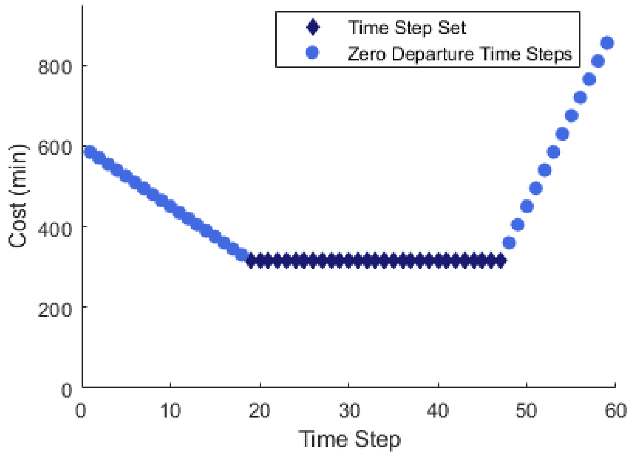

Figure 2 Equilibrium Costs of the Literature Example

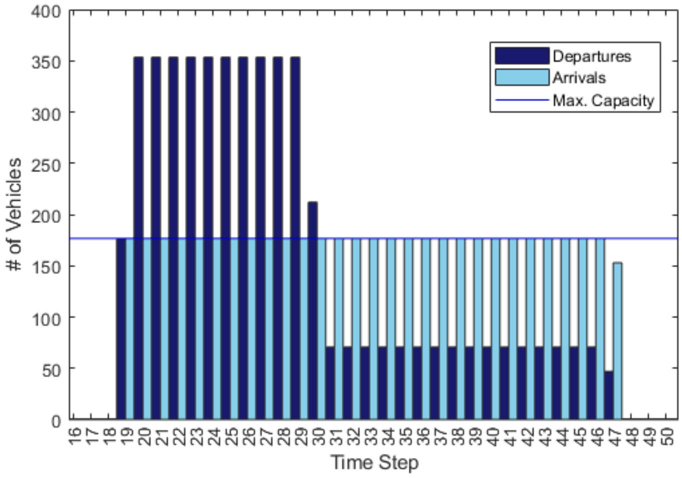

Figure 3 Departures and Arrivals for the Literature Example

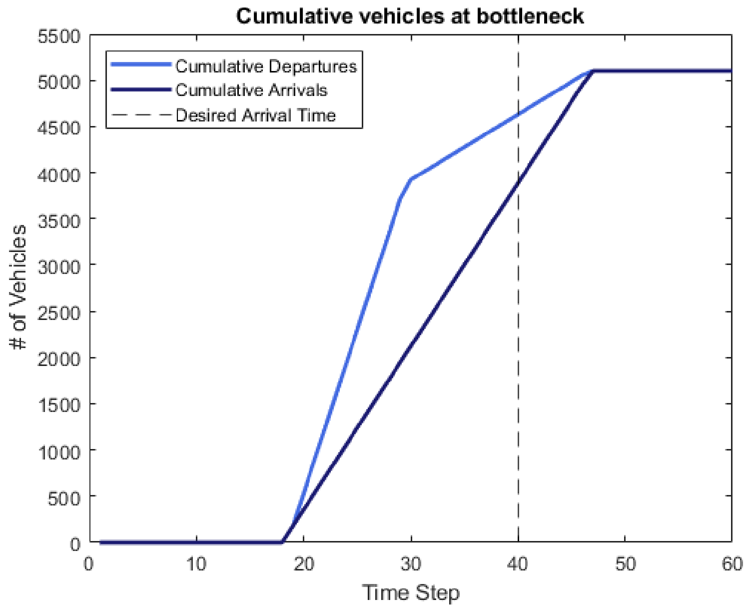

Figure 4 Cumulative Departures and Arrivals for the Literature Example

The best state-of-the-art models need ~1800 iterations for reaching a root mean square error (RMS) lower then 10^{-10}. The general framework, that takes the sensitivities into account, requires 39 iterations. Figure 5 plots the convergence of both discussed gap measurements.

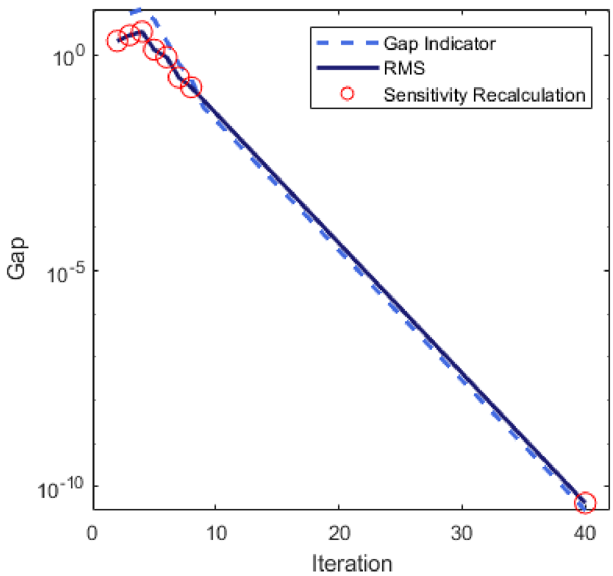

Figure 5 Convergence for the Literature Example

One iteration is much more expensive due to the sensitivity calculation. This example is to illustrate the benefit for using the sensitivities. More research should go to reduce the computation cost for one iteration. In the full paper a first step is taken. In that paper we will suggest a method that does not require to recalculate the sensitivity matrix in each iteration. In the full paper, we will also discuss other ways to reduce the cost for one iteration.

5. CONCLUSIONS

Many state-of-the-art algorithms try to mimic the perspective of the traveler. If the traveler experiences a high cost, he wants to change to an alternative with lower costs, without considering how the costs will change due to his and other travelers' actions. The general class proposed in this paper calculates a swap considering all sensitivities on how the cost of all alternatives will change. Our intuition says that this will improve convergence, and this has been shown for an example of the departure time choice problem. The proposed swaps are a numerical procedure to find the equilibrium, it does not mimic how people will change their behavior. This algorithmic class is no part of the day-to-day dynamics, as most other algorithms are. Travelers in real life will not swap according to the given swap vector as travelers are not aware of the full sensitivities and of course are not aware how other travelers will behave. This algorithmic class should only be used for evaluating the equilibrium itself. Natural questions that arise is how travelers do change their behavior and how they end up in the equilibrium. However, this is out of scope for the current research.

To be practical on real networks, an important addition to the model will be to cluster alternatives which are dependent on each other. The proposed method is not scalable for large networks. However, in many cases, the cost of an alternative is mainly dependent on its own. Finding for which alternatives this holds and finding clusters of alternatives that are dependent on each other, can further improve overall convergence.

The main contribution of this abstract is to demonstrate that using sensitivities can help convergence, especially for some special cases where alternatives are dependent on each other. A first method is proposed on how to incorporate the sensitivities in an algorithm, and it has been shown on a simple example how it can help the convergence in terms of iterations.

ACKNOWLEDGMENT

This project was supported by a PhD grant from the FWO-Vlaanderen (Fonds wetenschappelijk onderzoek, Strategisch Basisonderzoek). [1S60818N]

REFERENCES

Dafermos, S. (1980). Traffic Equilibrium and Variational Inequalities. *Transportation Science*, *14*(1). https://doi.org/10.1287/trsc.14.1.42

Small, K. A. (2015). The bottleneck model: An assessment and interpretation. *Economics of Transportation*, *4*(1–2), 110–117. https://doi.org/10.1016/j.ecotra.2015.01.001

Vickrey, W. (1969). Congestion Theory and Transport Investment. *The American Economic Review*, *59*(2), 251–260.