Efficient real-time taxi dispatching by solving the assignment problem

Michal Maciejewski

Division of Transport Systems, Poznan University of Technology
email: michal.maciejewski@put.poznan.pl

Department of Transport Systems Planning and Transport Telematics
TU Berlin
email: maciejewski@vsp.tu-berlin.de

The paper proposes an efficient real-time taxi dispatching strategy that solves the linear assignment problem to find an optimal taxi-to-request assignment at each decision epoch. The strategy performance was measured by means of microscopic traffic simulation in MATSim, and then compared to that of strategies investigated in earlier research.

1 Online taxi dispatching model

Let $N = \{1, \ldots, n\}$ be the set of taxi requests (customers). The following sequence of events is related to serving each request $i \in N$ and is illustrated in Figure 1. Taxi customer i calls a taxi (event E_i^{call}, time τ_i^{call}) specifying the pickup location, p_i. Since only immediate requests are considered, the customer's desired departure time is $\tau_i^{\text{dep}} = \tau_i^{\text{call}}$. A selected taxi is dispatched towards p_i at time τ_i^{disp} (event E_i^{disp}), and immediately after arrival, the pickup starts (event E_i^{pick0}, time τ_i^{pick0}). Once the passenger is picked up (event E_i^{pick1}, time τ_i^{pick1}), he or she specifies the destination, d_i, and the taxi sets out immediately. After reaching d_i, the dropoff begins (event E_i^{drop0}, time τ_i^{drop0}). Once the passenger gets out (E_i^{drop1}, time τ_i^{drop1}), the taxi is ready to serve another request. Due to the stochasticity of taxi dispatching, times $\tau_i^{\text{call}}, \tau_i^{\text{dep}}, \tau_i^{\text{disp}}, \tau_i^{\text{ready}}, \tau_i^{\text{pick0}}, \tau_i^{\text{pick1}}, \tau_i^{\text{drop0}}$ and τ_i^{drop1} are estimated until the respective events take place, and are therefore subject to change.

Request $i \in N$ is *open* if it either has not been planned yet, τ_i^{disp} is not set, or is planned to be served in the future, $\tau_i^{\text{disp}} > \tau^{\text{curr}}$, where τ^{curr} denotes the current time. Let L be the list of all open requests ordered by τ_i^{dep}. Each request i is inserted into L on submission, E_i^{call}, and removed from L on taxi dispatch, E_i^{disp}.

Let $M = \{1, \ldots, m\}$ be the set of vehicles. Each vehicle $k \in M$ is available at location o_k within the time window $[a_k, b_k)$. It is assumed that vehicles do not cruise and remain at the dropoff location of the last served customer. When no request has been assigned to k, o_k is k's initial location and a_k is the time the taxi starts operating. Otherwise, o_k is the dropoff location, d_i, and a_k is the time the dropoff ends, τ_i^{drop1}, of the last request assigned to k, i. Since d_i remains unknown till τ_i^{pick1}, both o_k and a_k are unknown temporarily as well, with the restriction that $a_k > \tau^{\text{curr}}$. Let $M^{\text{A}} \subseteq M$ be the set of all available vehicles,

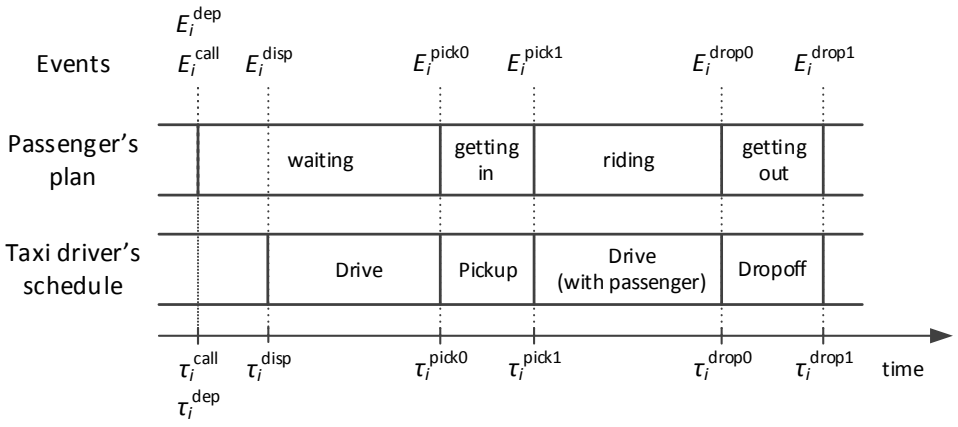

Figure 1: A taxi driver's schedule and a passenger's plan

i.e. vehicles $k \in M$ such that o_k and a_k are known and $b_k > \tau^{\text{curr}}$. Let $M^{\text{I}} \subseteq M^{\text{A}}$ be the set of all idle vehicles; available vehicle $k \in M^{\text{A}}$ is idle if $a_k \leq \tau^{\text{curr}}$.

2 Dispatching strategy based on the linear assignment problem

A taxi dispatch decision is based on solving the linear assignment problem, where variable x_{ki} represents the assignment of available vehicle $k \in M^{\text{A}}$ to open request $i \in L$. The cost of serving request i by vehicle k, c_{ki}, is defined as the passenger wait time from τ^{curr} on, i.e.

$$c_{ki} = \max\left(a_k, \tau^{\text{curr}}\right) + t^{\text{O}}_{ki}\left(\max\left(a_k, \tau^{\text{curr}}\right)\right), \tag{1}$$

where $t^{\text{O}}_{ki}(t)$ is the travel time from o_k to p_i, given the departure time, t.

If $\left|M^{\text{A}}\right|$ is less/greater than $|L|$, either M^{A} must be extended with dummy vehicles, or L with dummy requests, respectively, so that both sets are of equal size. The cost of serving a dummy request by vehicle k (i.e. vehicle k remains available at o_k) or dispatching a dummy taxi to request i (request i remains unplanned) is 0.

In the basic setting, the strategy reacts to events E^{call}_i and E^{call}_i by re-running the Hungarian method in order to solve the assignment problem. Due to the stochasticity of the travel time estimates, the method may be re-run each time the predictions for a_k change.

3 Reference strategies

The performance of the assignment-based dispatching was compared to that of selected four strategies studied earlier [1, 2]:

- *nearest-idle-taxi* – always dispatches the nearest idle taxi to the longest waiting request
- *nearest-idle-taxi/nearest-open-request* (also called *demand/supply balancing*) – extension of the first heuristic; when there is more then one open requests (*undersupply*), each idle taxi is dispatched to the nearest request; otherwise (*oversupply*) the nearest idle taxi is sent to the only open request
- *nearest-taxi* – extension of the first heuristic; takes into account all available vehicles (M^{A}) instead of idle ones only (M^{I})

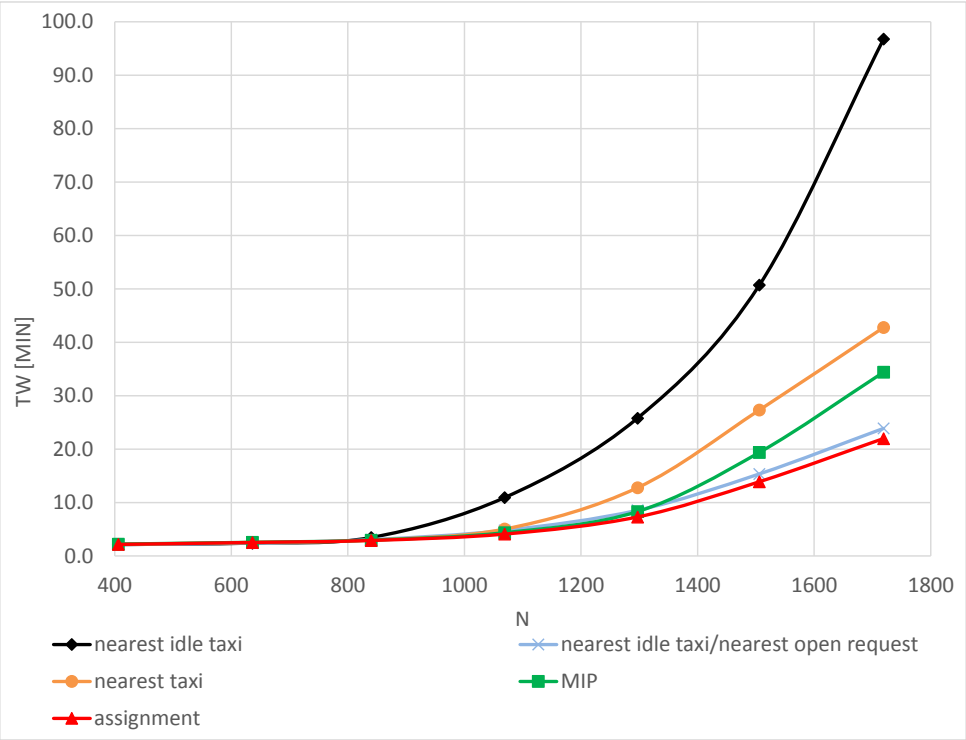

Figure 2: Average wait time, T_{W}, as a function of n

- *MIP* – transforms the online taxi dispatching problem into a mixed integer programming problem (with time-independent travel times) and solves it for a given rolling horizon, h, and computation time limit, t; the best settings obtained in experiments was $h = |M|$ and $t = 1$ minute when used with Gurobi Optimizer on a 6-core Intel Core i7-3930K processor

4 Comparison

The computational experiments were carried out in MATSim using a taxi scenario created for the city of Mielec, Poland, where non-uniformly distributed (in terms of time and space) taxi requests, in the amount of $n = 406$ to 1719, were served by 25 taxis [1]. The following two measures were calculated: the average passenger wait time, T_{W} (Fig. 2), and average pickup trip time, T_{P} (Fig. 3).

Both figures show that the proposed strategy outperformed the other strategies. It gave similar results as MIP in the low and medium demand cases. This resulted from the MIP strategy being more and more often terminated without finding an optimal solution as n grew; thus, giving results similar to the *nearest taxi* strategy, which was used as the initialization for the MIP simplex algorithm.

The second of the reference strategies (aiming at balancing demand and supply) proved to be almost as good as the assignment-based strategy in terms of minimizing T_{W}. The simplest *nearest idle taxi* approach performed poorest in all respects.

Interestingly, in T_{P} may drop under high demand: when the number of open requests grows, the average distance between vehicles and their nearest requests drops. Both the balancing strategy (explicitly) and the assignment-based strategy (implicitly) take advantage of this phenomenon.

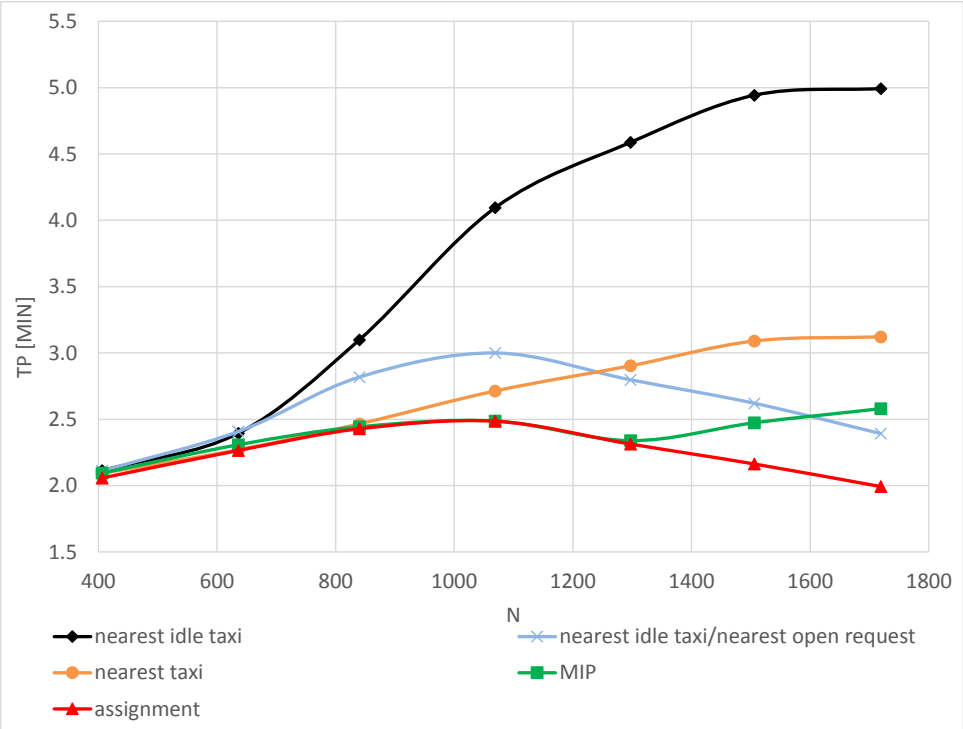

Figure 3: Average pickup trip time, T_{P}, as a function of n

5 Conclusions

The assignment-based strategy proved to be very efficient, both in terms of quality of solutions and computation time. Further study directions will include testing this strategy against big scenarios (e.g. Berlin [2]) and incorporating the notion of location attractiveness (i.e. vehicles staying in less attractive location will be more likely to be dispatched to new requests).

References

[1] Michal Maciejewski. Online taxi dispatching via exact offline optimization. *Logistyka*, 3:2133–2142, 2014.

[2] Michal Maciejewski and Joschka Bischoff. Large-scale microscopic simulation of taxi services. *accepted for ANT 2015*, 2015.