Optimization subject to transport micro-simulation constraints

Gunnar Flötteröd

April 10, 2015

Problem statement and related work

The purpose of this work is to solve optimization problems that are constrained by a dynamic traffic assignment (DTA) simulation. Examples are signal optimization (where the DTA simulation evaluates the system-wide performance of a given signal plan), origin/destination matrix estimation (where the DTA simulation predicts the network flow patterns resulting from an origin/destination matrix, which then can be compared to real data), and network design problems (where the DTA predicts the usage pattern of new/modified infrastructure elements).

DTA simulations for strategic planning implement, essentially, the following iterative scheme:

1. Create a synthetic population of individual travelers ("agents").
2. In every iteration (loosely interpreted: simulated day):
   (a) Every traveler chooses a travel plan.
   (b) All travelers execute their plans (i.e. they travel).
   (c) All travelers observe the resulting network conditions.

This scheme can be conveniently represented by a state space model

$$\mathbf{x}^{k+1} = \mathbf{f}[\mathbf{x}^k, \mathbf{u}] + \varepsilon^k \quad (1)$$

where k is the iteration (simulated day) index, $\mathbf{u}$ are decision variables (to be optimally selected), $\mathbf{x}^k$ is the state (memory) of the simulation, and ε^k is the zero-mean simulation stochasticity. Cascetta (1989); Cascetta and Cantarella (1991); Hazelton (2002); Nagel et al. (1998); Watling and Hazelton (2003) refer to similar formalisms, even though these approaches are mostly constrained to trip (and not full-day travel plan) DTA problems. A strategic planning simulation runs this process until a (stochastic) fixed point is reached.

The state of a DTA simulation that assigns all-day travel plans can be defined as a real-valued vector that contains the utilities of all possible travel plans of all agents, as "learned" by the individual agents. This real-valued state space is extremely large. To give an example, Bowman and Ben-Akiva (1998) estimate the number of single-day travel plans (comprising all-day route, mode, time choice) for a single individual to be in the order of 10^5; an urban population of size 10^6 then leads to a simulator state space dimension of 10^{11} – and this is in light of the combinatorial size of the universal route choice set a rather conservative estimate. It then becomes a rather striking observation that strategic simulators tend to find good approximations to this, say 10^{11}-dimensional, fixed point within the order of 10^3 iterations. One may conclude that the effective dimension of the state space through which the simulator moves is much smaller than the number of degrees of freedom in the underlying model system.

The objective of this work is to exploit this observation in the design of an efficient simulation-based optimization procedure. Consider the problem of selecting a decision variable $\mathbf{u}$, consisting possibly of both real-valued and integer entries, that minizing a real-valued objective function Q of the simulation constraints:

$$\min_{\mathbf{u}} \quad Q(\mathbf{x}) \quad (2)$$

$$\text{s.t.} \quad \mathbf{x} = \mathbf{f}[\mathbf{x}, \mathbf{u}]. \quad (3)$$

The equilibrium constraint (3) means that the expected state of the simulator has reached a deterministic fixed point; other formulations are conceivable. This is a computationally challenging problem because one needs to iterate the simulator all the way to convergence whenever one wishes to evaluate a single objective function value. The applicability of alternative approaches to incorporating the constraints less explicitly, for instance by introducing Lagrangian multipliers, is limited by the process-based simulation logic that is not easily amenable to a mathematical reformulation.

The approach pursued in this work is based on making $\mathbf{u}$-improvement steps while the simulator converges, meaning that one optimization iteration (improvement step) coincides with one simulation iteration (evaluating (1) once). A key ingredience of the method is to identify the effective state space of the simulator already while the simulator converges. A number of related approaches deserve attention in this context.

- Bierlaire and Crittin (2006) present an efficient techniques for solving large noisy nonlinear systems of equations. Their approach is based on fitting a regression model against the problem; it may be conjectured that the efficiency of their approach is a consequence of the relatively low effective dimension of their problem. Their approach does, however, not aim at the solution of simulation-based optimization problems.
- The two-simulation SPSA algorithm of Bhatnagar et al. (2013) is a generalization of SPSA (Spall, 1992) that requires to run two simulations in parallel, performs symmetric decision variable variations in each iteration of both simulations, and then computes improvement steps based on the usual SPSA finite difference scheme. The intuitive reason why the algorithm of Bhatnagar et al. (2013) converges is that the simulation responses are additionally smoothed over the iterations at a rate that is higher than the rate at which the search step size goes to zero, meaning that the simulator responses eventually appear stationary (converged) from the search algorithm's perspective. Interestingly, this particular method appears to have never been tried out in the transport optimization community, despite of its extensive use of SPSA.
- Rested multi-arm bandits (e.g. Tekin and Liu, 2012) provide another interesting perspective on the problem at hand. A multi-arm bandit is a gambling machine with two or more arms where playing a particular arm yields a pay-off that is drawn from a fixed pay-off distribution assigned to that arm. A rested multi-arm bandit attaches to each arm a discrete-time Markov process conditional on which the payoff is computed and that advances by one step whenever the arm is probed. Payoff-maximizing strategies for rested multi-arm-bandits are conceivable approaches to tackle discrete decision problems subject to simulation constraints that apparently have not yet received attention in the transportation community.

Methodology

The proposed approach looks as follows. Consider the simulation transition

$$\Delta\mathbf{x}^k \quad = \quad (\mathbf{f}[\mathbf{x}^k, \mathbf{u}] + \varepsilon^k) - \mathbf{x}^k \tag{4}$$

that describes the simulation's movement vector in state space in iteration k. Note that convergence of the simulation in the above sense is equivalent to

$$E\{\Delta\mathbf{x}\} \quad = \quad \mathbf{0}. \tag{5}$$

Now allow the decision variables $\mathbf{u}^k$ to change in every iteration. In iteration k, consider the last M transitions

$$\left(\mathbf{x}^{k-i}, \mathbf{u}^{k-i}, \Delta\mathbf{x}^{k-i}\right), \quad i = 0 \ldots M-1. \tag{6}$$

Based on this information, solve the problem

$$\min_{\boldsymbol{\alpha}} \quad \left\| \sum_{i=0}^{M-1} \alpha_i \Delta\mathbf{x}^{k-i} \right\|_2^2 + r \sum_{i=0}^{M-1} \alpha_i^2 \tag{7}$$

$$\text{s.t.} \quad \sum_{i=0}^{M-1} \alpha_i \; = \; 1; \quad \forall i : \alpha_i \geq 0, \tag{8}$$

which, essentially, aims at identifying a minimum-norm convex combination of the last M state transitions. (The second term, which is necessary to guarantee both a unique solution to the problem and certain asymptotic properties of the overall algorithm, may for now be ignored by letting $r = 0$.)

The usefulness of (7), (8) becomes clear when pretending the simulator was linear and, for the sake of presentational simplicity, deterministic:

$$\mathbf{x}^{k+1} \quad = \quad A\mathbf{x}^k + B\mathbf{u}^k \tag{9}$$

with A and B being matrices of suitable dimension. Finding an $\boldsymbol{\alpha}$ that exactly solves

$$\sum_{i=0}^{M-1} \alpha_i \Delta\mathbf{x}^{k-i} \quad = \quad \mathbf{0} \tag{10}$$

then allows to write

$$\tilde{\mathbf{x}}^k = A\tilde{\mathbf{x}}^k + B\left(\sum_{i=0}^{M-1} \alpha_i \mathbf{u}^{k-i}\right) \tag{11}$$

with

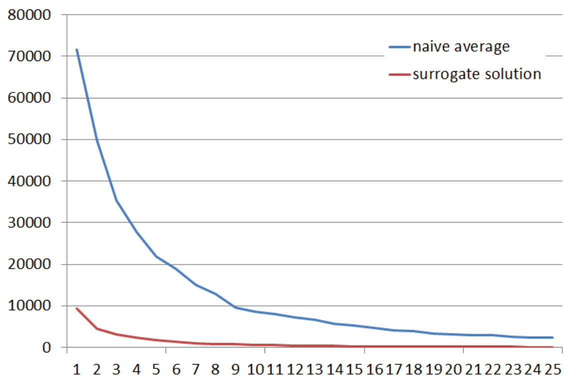
$$\tilde{\mathbf{x}}^k = \sum_{i=0}^{M-1} \alpha_i \mathbf{x}^{k-i}. \tag{12}$$

Thich results from (i) pre-multiplying (9) with α_i, (ii) summing over all $i = 0 \dots M-1$, and (iii) inserting (10). Noting that the convex combination $\sum_{i=0}^{M-1} \alpha_i \mathbf{u}^{k-i}$ is the expected value of a bootstrap distribution $\boldsymbol{\alpha}$ of the M last decision variables, one hence can conclude that re-sampling $\mathbf{u}^{k-i}$ with probability α_i in future iterations will make the simulation converge to a stationary state distribution with known expected value $\tilde{\mathbf{x}}^k$. That is, one can predict a surrogate simulator solution without having to run the simulator to convergence.

A convex combination of intermediate solution points is also at the core of the algorithm of Frank and Wolfe (1956). The purpose, however, is different: Frank and Wolfe (1956) solve a real-valued convex optimization problem, whereas here the convex combination is used to construct a stationary point of a hypothetical stochastic process. A number of further steps are needed in order to exploit this intermediate result in a solution to the optimization problem at hand.

- Means to evaluate and process (an approximation of) the huge-dimensional state vector are needed. A method that appears to function well in preliminary experiments is to use instead the network flows of one or several previous iterations. The subsequently reported result is based on this approximation.

- There needs to be a computational benefit in resorting to an approximate surrogate simulator solution. This is the case. For illustration, consider the following figure, which is based on an optimization problem with two real-valued decision variables subject to a multi-agent simulation of the City of Zurich. (Details are omitted due to space restrictions.) It shows the objective function value (7), which can be interpreted as an equilibrium gap function, over simulation iterations, with "naive average" corresponding to all αs set to $1/M$ and "surrogate solution" referring to the αs resulting from minimizing that function.



- Asymptotic convergence of the surrogate simulator solution to the true simulator solution needs to be established. There are several ways to go about this; one is to gradually increase the r coefficient in (7), another one is to gradually reduce the variability of the trial decision variables.

It remains to combine these elements into a concrete algorithm. A blueprint of one iteration looks as follows:

1. Create a trial decision variable. For instance, drawn from some problem-specific proposal distribution.
2. Advance the simulator by one iteration using this decision variable and compute the new surrogate simulator solution.
3. Accept the trial decision variable if the objective function value of the trial surrogate simulator solution constitutes an improvement, otherwise discard this trial.
4. Ensure that the surrogate simulator solution of the next iteration is closer to the real simulator solution than before.

The practical importance of using a surrogate simulator solution when evaluating a trial decision variable with only a single simulator transition is that the surrogate solution is (approximately) instantaneously equilibrated, whereas the original simulator is not. This (approximately) removes the path-dependence of the original simulation process that otherwise renders the current simulator state not representative for its equilibrated state.

Several concrete instances of this algorithm are currently being explored and tested; results will be presentable by the time of the conference.

References

Bhatnagar, S., Prasad, H. and Prashanth, L. (2013). *Stochastic Recursive Algorithms for Optimization*, Lecture Notes in Control and Information Sciences, Springer, London.

Bierlaire, M. and Crittin, F. (2006). Solving noisy large scale fixed point problems and systems of nonlinear equations, *Transportation Science* **40**(1): 44–63.

Bowman, J. and Ben-Akiva, M. (1998). Activity based travel demand model systems, *in* P. Marcotte and S. Nguyen (eds), *Equilibrium and advanced transportation modelling*, Kluwer, pp. 27–46.

Cascetta, E. (1989). A stochastic process approach to the analysis of temporal dynamics in transportation networks, *Transportation Research Part B* **23**(1): 1–17.

Cascetta, E. and Cantarella, G. (1991). A day-to-day and within-day dynamic stochastic assignment model, *Transportation Research Part A* **25**(5): 277–291.

Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming, *Naval Research Logistics Quarterly* **3**: 95–110.

Hazelton, M. (2002). Day-to-day variation in Markovian traffic assignment models, *Transportation Research Part B* **36**(7): 637–648.

Nagel, K., Rickert, M., Simon, P. and Pieck, M. (1998). The dynamics of iterated transportation simulations, *Proceedings of the 3rd Triennial Symposium on Transportation Analysis*, San Juan, Puerto Rico.

Spall, J. (1992). Multivariate stochastic approximation using a simultaneous pertubation gradient approximation, *IEEE Transactions on Automatic Control* **37**(3): 332–341.

Tekin, C. and Liu, M. (2012). Online learning of rested and restless bandits., *IEEE Transactions on Information Theory* **58**(8): 5588–5611.

Watling, D. and Hazelton, M. (2003). The dynamics and equilibria of day-to-day assignment models, *Networks and Spatial Economics* **3**(3): 349–370.