A search acceleration method for optimization problems with transport simulation constraints

Gunnar Flötteröd*

Department of Transport Science, KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

This work contributes to the rapid approximation of solutions to optimization problems that are constrained by iteratively solved transport simulations. Given an objective function, a set of candidate decision variables and a black-box transport simulation that is solved by iteratively attaining a (deterministic or stochastic) equilibrium, the proposed method approximates the best decision variable out of the candidate set without having to run the transport simulation to convergence for every single candidate decision variable. This method can be inserted into a broad class of optimization algorithms or search heuristics that implement the following logic: (i) Create variations of a given, currently best decision variable, (ii) identify one out of these variations as the new currently best decision variable, and (iii) iterate steps (i) and (ii) until no further improvement can be attained. A probabilistic and an asymptotic performance bound are established and exploited in the formulation of an efficient heuristic that is tailored towards tight computational budgets. The efficiency of the method is substantiated through a comprehensive simulation study with a non-trivial road pricing problem. The method is compatible with a broad range of simulators and requires minimal parametrization.

© 2016 The Author. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license. (http://creativecommons.org/licenses/by-nc-nd/4.0/)

1. Introduction

1.1. Motivation and problem statement

Many transport optimization problems are constrained by computationally demanding transport models that are solved through simulation, meaning that they are iterated towards a (deterministic or stochastic) equilibrium. Examples of such problems are signal optimization, origin/destination matrix estimation, network design, and road pricing. This work proposes a new method to accelerate the approximate solution of such problems. The remainder of this section characterizes the considered class of transport simulators, followed by the class of optimization problems and solution techniques with which the proposed acceleration method is compatible.

1.1.1. Considered class of transport simulators

The transport model's simulation process is represented by a state space model

\[ \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}) + \mathbf{e}_k \] (1)

* Corresponding author.
E-mail address: gunnar.floetteroed@abe.kth.se

http://dx.doi.org/10.1016/j.trb.2016.12.009
0191-2615/© 2016 The Author. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license. (http://creativecommons.org/licenses/by-nc-nd/4.0/)
where $k$ is the simulation iteration index, $x_k$ is the state (memory) of the simulation in iteration $k$, $u$ is the decision variable, the function $f$ represents the systematic simulation logic, and $e_k$ represents the possible simulation stochasticity arising in iteration $k$. For readability, no notational difference is made between a random variable and its realization; where this is necessary it is made explicit. Subsequently, all bold-font variables refer to vectors. The elements of a vector are written as simple italic variables, for instance $u = (u_1)$. When a subscript is attached to a bold-font variable, then this variable refers to one complete vector, for instance out of a set $\{u_1, u_2, \ldots \}$.

It is assumed that the transport simulation repeatedly evaluates (1) until a deterministic fixed point (in the absence of simulation stochasticity) or a stationary state distribution (in the stochastic case) is reached that represents the long-term (equilibrium) behavior of the system. The terms “equilibrium” and “stationarity” will subsequently be used exchangeably; a deterministic equilibrium point can be interpreted as the only possible realization of a stationary singleton distribution. This formalism covers (but is not constrained to) a broad range of traffic assignment models (e.g. Watling and Hazelton, 2003, see also the literature review further below):

- The model may be deterministic or stochastic.
- The model may be macroscopic or microscopic. A macroscopic model assigns real-valued commodity flows; a microscopic model assigns particles that could represent trip-makers or simulated persons (“agents”).
- The model may be within-day static or dynamic; a within-day time dimension requires that the simulator state contains within-day dynamic information (for instance per within-day time slice).

The iterative simulation scheme (1) then implements, typically, the following logic:

1. A travel demand (an origin/destination matrix or trip list or synthetic traveler population) is exogenously given.
2. The following is repeated until the corresponding process (1) reaches stationarity:
   (a) Each commodity (origin/destination flow, trip-maker, simulated person) selects a travel alternative (route, departure time, trip sequence, ...). In the microscopic case, this is a single realization; in the macroscopic case, a real-valued commodity may be split up and distributed across multiple alternatives.
   (b) All commodities are loaded onto the network (for instance by solving a continuum flow model or simulating the movement of individual vehicles).
   (c) The resulting network performance measures (in particular, travel times) are used to (re-)assess the travel alternatives available to all commodities.

This sketch of a simulation algorithm is not meant to be exclusive; any iterative simulation scheme that can be mapped on the formalism (1) and complies with the subsequently formulated assumptions is compatible with the proposed method. Vice versa, the proposed method is not applicable to models that are not solved by iterating a process of the form (1) to equilibrium.

1.1.2. Considered class of optimization problems and solution techniques

The considered Main Problem is to select a decision variable $u$, consisting possibly of both real-valued and discrete (integer, binary) entries, that minimizes a real-valued objective function $Q$ of the stationary simulator states:

$$
\text{Main Problem} \begin{cases}
\min_{u \in \Omega} & Q(u) \\
\text{s.t.} & \tilde{Q}(u) = E\{Q(f(x_k, u)) \} \forall k \geq k_{\text{stat}}
\end{cases}
$$

where $E\{\}$ is the expectation operator and stationarity of the simulation process is postulated as of simulation iteration $k_{\text{stat}}$. The set of feasible decision variables is denoted by $\Omega$. It is exogenous to the simulation process in the sense that membership in $\Omega$ can be tested without having to run the simulator. This excludes constraints where the feasibility of a decision variable depends on the stationary distribution of simulator states it leads to; such constraints may be accounted for by adding corresponding penalty terms to the objective function.

The computational difficulty of the Main Problem stems from the fact that one needs to iterate the simulator all the way to stationarity whenever one wishes to evaluate a single objective function value. A traffic assignment (micro-)simulation of a large metropolitan study region can require more than a day of computing time to approximate a single equilibrium solution (Balmer et al., 2009). An operational optimization approach hence needs to function subject to a tight computational budget where the evaluation of more than a few dozen simulator responses is, at least on a normal computer, practically out of the question. Finding or at least approximating a solution to the Main Problem is further complicated by the often not easily understood black-box specification of the simulator, which in many cases is represented procedurally (as a computer program) but hardly comes with identifiable mathematical properties (such as having the form of a system of equations with given derivatives).

Applying heuristics to tackle the Main Problem hence appears pragmatically justified; so does the deployment of optimization techniques with rigorously proven asymptotic properties subject to tight computational budgets (where their asymptotic convergence proofs do not hold). The acceleration method proposed in the present work is also heuristic. It aims at compatibility with a broad class of Embedding Solution Techniques (optimization algorithms and search heuristics) where
Algorithm 1 Sketch of an Embedding Solution Technique.

1. Initialization:
   (a) Set the Improvement Step counter $R$ to zero.
   (b) Select an initial (and currently best) decision variable $\mathbf{u}^{(0)}$.
   (c) Optional: Set the initial algorithm control parameters $\mathbf{β}^{(0)}$.

2. Repeat for $R = 1, 2, \ldots, R_{\text{max}}$ the following Improvement Step:
   (a) Create $M$ feasible (i.e. elements of $\Omega$) variations $\{\mathbf{u}_1^{(R)}, \ldots, \mathbf{u}_M^{(R)}\}$ of $\mathbf{u}^{(R-1)}$, possibly using $\mathbf{β}^{(R-1)}$.
   (b) Selection Problem: Identify the (approximately) best decision variable $\hat{\mathbf{u}}^{(R)} \in \{\mathbf{u}_1^{(R)}, \ldots, \mathbf{u}_M^{(R)}\}$, ideally such that
      $$ Q(\hat{\mathbf{u}}^{(R)}) = \min\{Q(\mathbf{u}_1^{(R)}), \ldots, Q(\mathbf{u}_M^{(R)})\}, $$
      possibly using $\mathbf{β}^{(R-1)}$.
   (c) Update the currently best solution $\mathbf{u}^{(R)}$ based on $\hat{\mathbf{u}}^{(R)}$, possibly using $\mathbf{β}^{(R-1)}$.
   (d) Adaptation Problem: Compute new parameters $\mathbf{β}^{(R)}$ based on the most recent Improvement Step(s).

A currently best solution to the Main Problem is repeatedly improved through an Improvement Step. A blueprint of this approach is shown in Algorithm 1. Kolda et al. (2003) and Hong et al. (2015) review concrete instances of Embedding Solution Techniques that match this blueprint; more examples are given in the literature review below. The subsequently stressed indifference of the proposed acceleration method with respect to particular properties of the Embedding Solution Technique refers to its technical compatibility; the resulting computational performance is likely to differ among configurations.

The algorithm is initialized in Step 1 with a first decision variable guess and initial algorithm control parameters. It then runs through $R_{\text{max}}$ Improvement Steps, with $R_{\text{max}}$ being either a priori defined or being dependent on some criterion that is evaluated after each Improvement Step. The proposed acceleration method is indifferent to how the Embedding Solution Technique selects $R_{\text{max}}$. The way in which new proposal solutions are created in Step 2a is problem-specific. If, for instance, the decision variable is a real-valued vector, then a Gaussian perturbation of each of its entries could be an option, with the algorithm control parameters comprising the variances of these perturbations. The proposed acceleration method is indifferent to how proposal solutions are generated by the Embedding Solution Technique. The solution of the Selection Problem in Step 2b is where the present work contributes by effectively approximating the best decision variable out of a possibly large candidate set. The most straightforward approach to improving the currently best solution in Step 2c is to set $\mathbf{u}^{(R)}$ to $\hat{\mathbf{u}}^{(R)}$, even though more elaborate procedures are conceivable (for instance, a line search along the direction $\hat{\mathbf{u}}^{(R)} - \mathbf{u}^{(R-1)}$ for real-valued decision variables). The proposed acceleration method is indifferent to how the Embedding Solution Technique updates the currently best solution. Finally, the Adaptation Problem in Step 2d aims at improving the algorithm’s performance by adjusting its control parameters to the problem at hand. One example, continuing the illustration of a real-valued decision vector, would be to adjust the variance of its perturbations in Step 2a, for instance as a function of $R$ or depending on the success of the last Improvement Step. The proposed acceleration method is indifferent to if and how the Embedding Solution Technique adjusts its algorithm control parameters.

1.2. Literature review

1.2.1. Process-based traffic assignment

The (stochastic) process perspective adopted in (1) was pioneered by Cascetta (1989) for a within-day static route swapping model and further developed by Cascetta and Cantarella (1991) for within-day dynamics. The subsequent body of research related to this model class is reviewed by Watling and Hazelton (2003) and Watling and Cantarella (2013). Much of this work relies on concrete mathematical model features that may be difficult if not impossible to obtain from a real transport simulation package (such as the transition function Jacobian, its characteristic polynomial and eigenvalues, as used in Watling, 1999). Cantarella et al. (2015) extend this more mathematical perspective to a class of integrated strategic transport models and give an example where the traffic assignment is inserted into a four-stage travel demand model. The complementary approach of considering a general transport simulation package and then performing a more limited mathematical analysis is pursued, for instance, by Nagel et al. (1998) and Flöteröd et al. (2011).

1.2.2. Optimization methods that do not exploit the process-based assignment framework

Here, optimization algorithms and heuristics are reviewed that aim at minimizing an objective function subject to an equilibrium constraint but do not account for the (simulation or computation) process according to which this constraint is evaluated. One faces a continuum of approaches, ranging from rigorous theoretical work to the deployment of general-purpose heuristics.

Starting out from the mathematical end of the spectrum, the Main Problem can be related to the class of Stochastic Mathematical Problems with Equilibrium Constraints (SMPECs; Patriksson, 2008), with the major differences being that the Main Problem considered here relies on a process-based assignment model and not a stochastically parametrized instantaneous model, and that it allows for the use of a black-box simulator. The latter property prohibits the rigorous solution of the Main Problem with SMPEC techniques. Osorio and Bierlaire (2013) present a simulation-based optimization framework for real-valued decision variables that combines simulated level-of-detail and mathematical tractability by iterating between (i)
fitting an analytical approximation model against stochastic simulator responses and (ii) improving the decision variables by solving a tractable optimization problem that only takes the analytical model as a constraint. Osorio and Chong (2015) and Osorio and Nanduri (2015) are examples of concrete transport applications of this technique. An alternative approach to the same problem class is the Stochastic Approximation Simultaneous Perturbation (SPSA) method of Spall (1992). It has become popular in origin-destination matrix estimation, where a number of problem-specific adaptations of the original algorithm have been proposed (e.g. Lu et al., 2015; Tympakianaki et al., 2015).

Beyond this, an abundance of applied work relies on, for instance, general-purpose local search techniques (e.g. Kolda et al., 2003), evolutionary algorithms (rooted in the work of Holland, 1975), simulated annealing (going back to Kirkpatrick et al., 1983). These techniques may be tailored to a particular problem through suitable mechanisms for trial solution generation, referred to as “search patterns” in local search, by defining “recombination” and “mutation” in evolutionary computing or a “proposal distribution” in simulated annealing, but can otherwise be safely considered as truly black-box search techniques.

1.2.3. (Optimization) methods that exploit the process-based assignment framework

Here, techniques addressing a variety of problems are reviewed that explicitly account for the process-based assignment framework, in that they do not only consider the model’s equilibrium solution but also the simulation process (1) according to which this solution is attained. The subsequently given references address specific problem instances and provide ample references to the much larger body of research in their respective field.

Parry and Hazelton (2013) calibrate stochastic process assignment models from traffic counts in a Markov chain Monte Carlo framework; Hazelton and Parry (2015) perform Bayesian model selection within the same framework. Flöterød et al. (2011) present a generalized path flow estimator that adjusts simulated travel behavior from traffic counts jointly with a stochastic micro-simulation assignment process. Ye et al. (2015) propose a trial-and-error-based marginal cost pricing approach where tolls are repeatedly adjusted based on non-equilibrated flow observations. They assume that the real day-to-day traffic dynamics follow a continuous-time deterministic process, and they do not require the underlying process model to be known. Rampba and Boyles (2016) express the dynamic road pricing problem as an infinite horizon Markov decision process assuming a stochastic day-to-day route assignment process. Smith (2015) analyzes the joint convergence of $P_0$ signaling (a local control strategy introduced by Smith, 1980) and mutually consistent equilibrium route flows and bottleneck delays in an abstract deterministic process assignment setting. Bifulco et al. (2016) study the effect of advanced traveler information systems on network equilibria in a deterministic process framework.

The two-simulation SPSA algorithm of Bhatnagar et al. (2013, Chapter 5.6) is a generalization of SPSA 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 (Spall, 1992). Two-simulation SPSA appears to have not yet been used in the transport domain.

The (transport-independent) class of ranking & selection (RS) techniques aims at identifying the best out of a finite set of alternative simulation configurations (e.g. Swisher et al., 2003; Kim and Nelson, 2006b). RS has been integrated in optimization routines matching Algorithm 1 (Hong et al., 2015), and there exist dedicated RS techniques for stationary stochastic processes of the form (1) (Goldsman et al., 2002). The subsequently developed method could indeed be labeled as an RS heuristic for computationally heavy simulators; a discussion of its relation to and difference from the existing RS literature is postponed to Section 2.3.3 because this requires to first develop the proposed method in detail.

2. Method

The proposed method is derived based on a set of assumptions that lead to certain performance guarantees. This reveals problem structure that is ultimately exploited in the construction of an efficient heuristic that may be deployed even if the underlying assumptions can no longer be validated, i.e. for a black-box simulation process.

**Assumption 1.** The zero-mean simulation noise vector $\mathbf{e}$ in (1) is uncorrelated across simulation iterations: $E[\mathbf{e}_k \mathbf{e}_l^T] = \mathbf{0}$ for simulation iterations $k \neq l$, with superscript $T$ denoting the transpose and $\mathbf{0}$ being an all-zero matrix of suitable dimension. The variance of each element in $\mathbf{e} = (\mathbf{e}_j)$ is bounded from above by a finite number: $\text{VAR}[\mathbf{e}_j] \leq \sigma_j^2$.

Uncorrelated simulation noise means that the transition function $\mathbf{f}$ in (1) captures all (stochastic) dependence between subsequent simulator states. This requires that all simulation process components that are dependent across iterations are included in the simulator’s state space. Bounded noise variance is a mild assumption that follows, for instance, from the simulator state space being bounded.

The proposed approach to tackling Algorithm 1’s Selection Problem will make heavy use of interpolations (more specifically, convex combinations), both of simulator states and of objective function values. This is justified and becomes mathematically tractable if one assumes both the simulator dynamics and the objective function to be linear for a given Selection Problem.

**Assumption 2.** For a given Selection Problem, the simulator dynamics are linear:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = A\mathbf{x} + B\mathbf{u}$$

(2)
with \( A \) and \( B \) matrices of suitable dimension that are specific to the given Selection Problem.

**Assumption 3.** For a given Selection Problem, the objective function is linear:

\[
Q(x) = c^T x
\]

where the function argument \( x \) is an arbitrary point in simulator state space and \( c \) is a vector of suitable dimension that is specific to the given Selection Problem.

Both linearity assumptions are made only for the sake of the subsequent mathematical reasoning; an actual linearization (computation of derivatives) is not needed. To make this explicit, \( Q(x) \) and \( f(x, u) \) are written instead of \( c^T x \) and \( Ax + Bu \) wherever possible. Means to treat non-linear simulators and/or objective functions are presented at the end of this section.

It has been stated before that the decision variables may be discrete-valued, and the possibility of discrete-valued simulator states has neither been excluded. Linear simulator dynamics, however, call for a real-valued simulator state space. If there are discrete-valued elements in the simulator state space, then assuming linearity can be interpreted as allowing for a (fictitious, since never explicitly computed) interpolation between these discrete states. Discrete-valued decision variables are fully compatible with the linearity assumption because nothing prohibits the evaluation of (2) in the presence of integer or binary decision variable components.

**Assumption 4.** The deterministic process \( x_{k+1} = f(x_k, u) \), i.e. (1) without simulation noise, is for all possible decision variables \( u \), a contraction mapping with a Lipschitz constant \( L \) strictly smaller than one, meaning that it has exactly one fixed point (e.g. Berinde, 2007, Chapter 2). In combination with Assumptions 1 and 2, this implies that the expected simulator state follows the deterministic dynamics \( E[x_{k+1}] = AE[x_k] + Bu \) and converges to \( \bar{x}(u) = (I - A)^{-1} Bu \), with \( I \) an identity matrix of suitable dimension.

Intuitively, this means that a simulation trajectory moves on average only closer to its expected stationary state. Combining Assumptions 2, 3 and 4 leads to the following linear version of the Main Problem:

**Linear Main Problem**

\[
\begin{align*}
\min_{u \in \Omega} \quad & Q(\bar{x}) = c^T \bar{x} \\
\text{s.t.} \quad & \bar{x} = f(\bar{x}, u) = Ax + Bu.
\end{align*}
\]

The assumptions leading to the Linear Main Problem are made only within each Improvement Step of Algorithm 1, meaning that they correspond to what would be assumed by any other optimization technique that solves the Main Problem through repeated linearizations. The general function symbols \( Q \) and \( f \) are still used when referring to the Linear Main Problem because an actual linearization (computation of derivatives) is never required.

The above linearity assumptions are idealizations. The following measures may help to minimize the degree to which they are violated in a practical setting. (i) Create the candidate decision variables in Step 2a of Algorithm 1 as relatively small variations of the currently best decision variable. The assumption that the simulator reacts linearly to a decision variable within a given Selection Problem then only needs to hold for decision variables that are relatively similar to each other. (ii) Set the common initial state of all simulations for all candidate decision variables to the stationary state of the currently best decision variable. The assumption of linear simulator responses within a given Selection Problem then only needs to hold at all points in state space that can be reached from this common initial state.

### 2.1. Approximate solution of the Selection Problem

The Improvement Step counter \( R \) of Algorithm 1 is omitted in this section. The candidate decision variable set \( \{u_1, \ldots, u_M\} \) is, without loss of generality, replaced here by an ordered multi-set \( \{u_1, \ldots, u_N\} \) with \( N \geq M \); this implies that (i) the same distinct decision variable may appear multiple times in that set and (ii) an element of that set can be uniquely identified by its index \( i \in \{1, \ldots, N\} \).

The Selection Problem to be solved in Step 2b of Algorithm 1 is written as follows:

**Exact Selection Problem**

\[
\begin{align*}
\min_{\alpha \in \Omega_{\text{exact}}} \quad & Q_{\text{exact}}(\alpha) = \sum_{i=1}^{N} \alpha_i Q(\bar{x}_i) \\
\text{s.t.} \quad & \Omega_{\text{exact}} = \left\{ \alpha = (\alpha_i) : \sum_{i=1}^{N} \alpha_i = 1 \text{ and } \alpha_i \in \{0, 1\}, i = 1 \ldots N \right\} \\
& \bar{x}_i = f(\bar{x}_i, u_i), \quad i = 1 \ldots N
\end{align*}
\]

with \( \bar{x}_i \) representing the expected stationary simulator state given that decision variable \( u_i \) is used. The Exact Selection Problem is to identify a binary vector \( \alpha = (\alpha_i) \in \{0, 1\}^N \) with the property that if \( \alpha_i = 1 \) then \( u_i \) solves the Selection Problem. If some \( \alpha \) with \( \alpha_i = 1 \) solves the Exact Selection Problem then both this \( \alpha \) and the corresponding decision variable \( u_i \) are referred to as solutions of the Exact Selection Problem. It further is written \( Q_{\text{exact}} = \min_{\alpha \in \Omega_{\text{exact}}} Q_{\text{exact}}(\alpha) \).
The Relaxed Selection Problem differs from the Exact Selection Problem in that it replaces the integrality constraints on $\alpha$ by non-negativity constraints, meaning that $\alpha$ is now permitted to be a real-valued vector:

\[
\begin{align*}
\text{Relaxed Selection Problem} & \quad: \min_{\alpha \in \Omega_{\text{relaxed}}} Q_{\text{exact}}(\alpha) \\
\text{s.t.} & \quad \Omega_{\text{relaxed}} = \left\{ \alpha = (\alpha_i) : \sum_{i=1}^{N} \alpha_i = 1 \text{ and } \alpha_i \geq 0, \ i = 1 \ldots N \right\} \\
& \quad \hat{x}_i = f(\hat{x}_i, u_i), \ i = 1 \ldots N.
\end{align*}
\]

Note that the Exact and Relaxed Selection Problem use the same objective function $Q_{\text{exact}}$. Let $Q_{\text{relaxed}}^{\ast} = \min_{\alpha \in \Omega_{\text{relaxed}}} Q_{\text{exact}}(\alpha)$. The Exact Selection Problem and the Relaxed Selection Problem are related as follows.

**Proposition 1.** The minimal objective function value of the Exact Selection Problem is equal to the minimal objective function value of the Relaxed Selection Problem: $Q_{\text{exact}}^{\ast} = Q_{\text{relaxed}}^{\ast}$. Further, a strictly positive $\alpha_i$ in the optimal solution of the Relaxed Selection Problem implies that $\alpha_i = 1$ is optimal for the Exact Selection Problem, which in turn means that the corresponding decision variable $u_i$ is optimal.

**Proof.** From $\Omega_{\text{exact}} \subset \Omega_{\text{relaxed}}$ follows $Q_{\text{relaxed}}^{\ast} \leq Q_{\text{exact}}^{\ast}$. Assumption 3 turns the Relaxed Selection Problem into a linear program. (Here, it does not matter if the simulator dynamics leading to the stationary states that parametrize this program are linear or non-linear.) Assumption 4 ensures that this program is feasible. There hence exists a vertex in its constraint polyhedron that optimally solves the program, i.e. has the objective function value $Q_{\text{relaxed}}^{\ast}$ (e.g. Bierlaire, 2015, Theorem 16.2). This vertex is (like all vertices of the constraint polyhedron) an element of $\Omega_{\text{exact}}$, implying $\alpha \in \Omega_{\text{exact}} : Q(\alpha) = Q^{\ast}_{\text{relaxed}}$. Combining this with the previous observation that $Q^{\ast}_{\text{relaxed}} \leq Q^{\ast}_{\text{exact}}$ yields $Q^{\ast}_{\text{relaxed}} = Q^{\ast}_{\text{exact}}$. Further, any non-binary optimal solution of the Relaxed Selection Problem can be expressed as a convex combination of the (objective function values of the) vertices of its constraint polyhedron, i.e. of the elements of $\Omega_{\text{exact}}$, implying that all elements of $\Omega_{\text{exact}}$ contributing to that combination are also optimal.

This does not claim that every optimal solution to the relaxation of a binary integer program is an optimal solution to the binary integer program itself. **Proposition 1** is only stated for the Exact/Relaxed Selection Problem, in which the decision variables take the form of weights in a convex combination.

All subsequent references to the objective function $Q_{\text{exact}}(\alpha)$ come with the assumption that this objective function evaluates expected stationary simulator states, with the difference between the Exact and the Relaxed Selection Problem being that the former is bound to binary $\alpha$ values.

The following developments aim at also making use of transient and possibly stochastic simulator states. Assign to each distinct $u \in \{u_1, \ldots, u_N\}$ a corresponding simulation trajectory. Let this trajectory be the result of setting the simulator to some given initial state $x_0$ and then evaluating one or more subsequent simulation transitions, i.e. moves in state space according to (1), using in each transition the decision variable $u_i$. Let $\{x_1, \ldots, x_N\}$ and $\{\Delta x_1, \ldots, \Delta x_N\}$ be the sets of all states and all state transitions contained in the trajectories for all $u \in \{u_1, \ldots, u_N\}$. To relate a concrete state or state transition to its corresponding decision variable, the indexing in the multi-sets $\{u_1, \ldots, u_N\}$, $\{x_1, \ldots, x_N\}$ and $\{\Delta x_1, \ldots, \Delta x_N\}$ is such that

\[
\begin{align*}
x_i = f(x_i - \Delta x_i, u_i) + e_i, \ i = 1 \ldots N.
\end{align*}
\]

In words: Given that the simulator is in state $x_i - \Delta x_i$ and that decision variable $u_i$ is implemented, one (stochastic) iteration of the simulator leads to the state transition $\Delta x_i$ and reaches the new state $x_i$.

The **Approximate Selection Problem**, which is central to the following developments, can now be stated.

\[
\begin{align*}
\text{Approximate Selection Problem} & \quad: \min_{\alpha \in \Omega_{\text{relaxed}}} \tilde{Q}(\alpha) = \tilde{Q}(\alpha) + \nu \cdot \text{eg}(\alpha) + \omega \cdot \text{ug}(\alpha) \\
\text{s.t.} & \quad \tilde{Q}(\alpha) = \sum_{i=1}^{N} \alpha_i Q(x_i) \quad (\text{Transient Performance}) \\
& \quad \text{eg}(\alpha) = \left\| \sum_{i=1}^{N} \alpha_i \Delta x_i \right\| \quad (\text{Equilibrium Gap}) \\
& \quad \text{ug}(\alpha) = \sum_{i=1}^{N} \alpha_i^2 \quad (\text{Uniformity Gap}) \\
& \quad x_i = f(x_i - \Delta x_i, u_i) + e_i, \ i = 1 \ldots N \quad (\text{Simulator Transitions})
\end{align*}
\]

where $\nu, \omega \geq 0$ are real-valued constants and $\| \cdot \|$ refers to the Euclidean norm throughout. The Approximate Selection Problem is constructed around a set of Simulator Transitions $\{(u_i, x_i, \Delta x_i)\}_{i=1}^{N}$ which satisfy (4). This means that the Approximate Selection Problem can be solved even if only transient and stochastic simulator output realizations are available, i.e. without
having to run the simulation to stationarity. The minimum objective function value of the Approximate Selection Problem is $\hat{Q}^* = \min_{\alpha \in \Omega_{\text{relaxed}}} \hat{Q}(\alpha)$.

The three addends constituting the objective function $\hat{Q}(\alpha)$ of the Approximate Selection Problem can be informally interpreted as follows. The Transient Performance $\hat{Q}(\alpha)$ evaluates an $\alpha$-weighted convex combination of the objective function values of the transient simulator states $\{x_1, \ldots, x_N\}$. It should be interpreted as an inexact approximation of the corresponding $\alpha$-weighted convex combination of expected stationary simulator states. The second and third component of $\hat{Q}(\alpha)$ reflect this approximation error (in a sense that is made concrete by the two propositions following immediately below). The Equilibrium Gap $\varepsilon g(\alpha)$ functions as a penalty representing the error made by evaluating out-of-equilibrium conditions: The $\Delta x$ vectors are movements in state space, the norm of which is on average the smaller the closer one is to an equilibrium. The Uniformity Gap $\varepsilon u(\alpha)$ constitutes a penalty reflecting the possible presence of stochasticity in the evaluated transient simulator states: The variance of $\hat{Q}(\alpha)$ and $\varepsilon g(\alpha)$ is minimized by $\alpha_i = 1/N$ for all $i = 1, \ldots, N$, which is also the minimizer of $\varepsilon u(\alpha)$.

The Approximate Selection Problem is related to the Relaxed Selection Problem through the following two properties.

**Proposition 2.** Let $d$ be the dimension of the simulator state space, let $\sigma^2_j$ be the upper bound on the simulation noise variance of state vector element $j = 1 \ldots d$ referred to in Assumption 1, let $c$ be the gradient of the linear objective function referred to in Assumption 3, let $L < 1$ be the Lipschitz constant of the simulation mapping referred to in Assumption 4, and let $\delta$ be an arbitrarily selected non-negative real-valued constant. Given the concrete parameter values

$$v = \|c\| \frac{L}{(1 - L)}$$

$$w = (\|c\| + v)\delta \sum_{j=1}^{d} \sigma^2_j$$

in the objective function of the Approximate Selection Problem, the following inequality holds:

$$\Pr(Q_{\text{exact}}(\alpha) \geq \hat{Q}(\alpha)) \leq 1/\delta^2 \quad \text{for all } \alpha \in \Omega_{\text{relaxed}}.$$  

(5)  

Here, the stochastic Simulator Transitions evaluated in $\hat{Q}(\alpha)$ are treated as random variables, meaning that this inequality holds prior to their computation. $Q_{\text{exact}}(\alpha)$, on the other hand, uses expected stationary simulator states.

**Proof.** See Appendix A.  

An informal interpretation of Proposition 2 reads as follows. Inequality (7) bounds the probability that $\hat{Q}(\alpha)$ fails to be an upper bound of $Q_{\text{exact}}(\alpha)$. By selecting a large $\delta$ (and through (6) a correspondingly large $w$ coefficient), this probability can be made small, and one hence can state that $\hat{Q}(\alpha)$ (using only stochastic and transient simulator states) is likely to bound $Q_{\text{exact}}(\alpha)$ (evaluating expected stationary simulator states) from above. For the interpretation of (5) and (6), recall that $v$ and $w$ are multiplied into the Equilibrium Gap and the Uniformity Gap of the Approximate Selection Problem, respectively. $v$ grows with $L$, meaning that the slower the simulator converges the more the Equilibrium Gap contributes to increasing $\hat{Q}(\alpha)$. $w$ is proportional to the magnitude of the simulator noise, meaning that the impact of the Uniformity Gap on $\hat{Q}(\alpha)$ increases with the amount of stochasticity in the simulation.

One may not be able to compute concrete numbers from (5) and (6) because these equations rely on in general unknown problem properties (simulator noise variances, Lipschitz constant). In such a setting, the added value of Proposition 2 is that it guarantees the existence of $v$ and $w$ values for which $\hat{Q}(\alpha)$ is likely to bound $Q_{\text{exact}}(\alpha)$ from above. This then justifies to set $v$ and $w$ based on alternative and more operational criteria, such as the one presented in Section 2.2.

**Proposition 3.** Let the set of Simulator Transitions in the Approximate Selection Problem be constructed by iterating the simulator $K$ times forward for each individual candidate decision variable. As $K$ goes to infinity, the minimal objective function value $Q^\ast$ of the Approximate Selection Problem converges in mean square to a value that is not larger than the minimal objective function value $Q^\ast_{\text{relaxed}}$ of the Relaxed Selection Problem for the same candidate decision variables.

**Proof.** See Appendix B.  

This means, informally, that adding transitions to the simulation trajectories of all decision variables asymptotically leads to $Q^\ast \leq Q^\ast_{\text{relaxed}}$.

The following puts these propositions to use. It attempts a gradual transition from rigorous reasoning to intuitive arguments because it ultimately aims at motivating a heuristic technique to solve the Selection Problem of Algorithm 1. Phrases of the type “X holds by Proposition Y given that condition Z is satisfied” are in cases where condition Z is stated in Proposition Y simplified into “X holds (Proposition Y)”. Let $\alpha^\ast$ be an optimal solution of the Approximate Selection Problem. By adding Simulator Transitions to that problem, $\hat{Q}(\alpha^\ast) \leq Q^\ast_{\text{relaxed}}$ can be attained (Proposition 3). Proposition 1 states that $Q^\ast_{\text{relaxed}} = Q^\ast_{\text{exact}}$. from which follows that
\( Q^{\ast}_{\text{relaxed}} \leq Q^{\text{exact}}(\alpha^{\ast}) \). It also is possible to establish \( Q^{\text{exact}}(\alpha^{\ast}) \leq \hat{Q}(\alpha^{\ast}) \) with arbitrarily high probability (Proposition 2). Combining these inequalities yields \( \hat{Q}(\alpha^{\ast}) \leq Q^{\ast}_{\text{relaxed}} \leq Q^{\text{exact}}(\alpha^{\ast}) \leq \hat{Q}(\alpha^{\ast}) \) and hence \( Q^{\ast}_{\text{relaxed}} = Q^{\text{exact}}(\alpha^{\ast}) \). meaning that \( \alpha^{\ast} \) also solves the Relaxed Selection Problem. Given this, Proposition 1 establishes that all non-zero entries in \( \alpha^{\ast} \) correspond to decision variables that solve the Exact Selection Problem, meaning that if \( \alpha^{\ast} > 0 \) then \( \alpha_{i} = 1 \) solves the Exact Selection Problem.

Informally: Evaluating many Simulator Transitions for each candidate decision variable and then solving the Approximate Selection Problem using all of these transitions comes with a high chance that the non-zero entries in the resulting \( \alpha \) vector indicate solutions of the Exact Selection Problem. However, this requires to also compute a large number of Simulator Transitions for inferior decision variables. Given a tight computational budget, the proposed approach hence incrementally builds a set of Simulator Transitions by (i) solving an Approximate Selection Problem that uses all so far available Simulator Transitions, and (ii) selecting the decision variable to be evaluated in the next Simulator Transition based on the (non-zero) entries of the \( \alpha \)-solution of this Approximate Selection Problem.

This method is specified in Algorithm 2. It is heuristic because (i) it uses a reasoning that is based on Proposition 2 and may hence hold, for each performed experiment, with a probability below one, and (ii) it relies on the asymptotic result of Proposition 3 in a non-asymptotic context.

The algorithm starts out by evaluating in Step 1 all candidate decision variables once; this enables a first (and possibly very rough) estimate of their performance. It then continues by repeatedly solving the Approximate Selection Problem in Step 2a and using its solution in Step 2b as a probability distribution according to which the decision variable for the next Simulator Transition is selected. (If one wishes a more explorative allocation of the computational budget that gives every decision variable a non-zero probability of being sampled, then one may replace the non-negativity constraint on \( \alpha \) in \( \Omega^{\text{relaxed}} \) by some strictly positive lower bound of at most \( 1/N \).) The simulation trajectory convergence criterion used in Step 2d needs to be chosen in a problem- and simulator-specific manner. The algorithm’s termination criterion defined in Step 2d has the effect that decision variables on which the computational budget is focused have indeed a higher chance of being returned by the algorithm.

2.2. Corresponding Adaptation Problem

Algorithm 2 requires to assign concrete values to the parameters \( v \) and \( w \), which may not be possible based on (5) and (6), as discussed above. Even if these formulas could be evaluated, they may not yield an ideal parametrization of Algorithm 2 because this algorithm relates only heuristically to Proposition 2.

A concrete instance of Algorithm 1’s Adaptation Problem with the algorithm control parameters being \( \beta = (v \, w) \) is hence formulated. It aims at minimizing the objective function value of the decision variable returned by Algorithm 2. Denote this objective function value by \( Q^{\ast} \). It depends through a complicated sequence of Approximate Selection Problems and stochastic simulator transitions on \( v \) and \( w \).

Consider a single iteration of Algorithm 2 and recall that the objective function of the Approximate Selection Problem is

\[
\hat{Q}(\alpha) = \hat{Q}(\alpha) + v \cdot \mathsf{e}(\alpha) + w \cdot \mathsf{u}(\alpha).
\]

Since the choice of \( \alpha \) is decisive for which decision variable is to be simulated next, it is reasonable to postulate a dependency of \( Q^{\ast} \) on \( \alpha \):

\[
Q^{\ast}(\alpha) = Q^{\ast}(\alpha).
\]

Now assume that one could select \( v, w \) such that

\[
|\hat{Q}(\alpha) - Q^{\ast}(\alpha)| \approx v \cdot \mathsf{e}(\alpha) + w \cdot \mathsf{u}(\alpha)
\]

(8)

could be established. Solving the Approximate Selection Problem would then amount to minimizing \( \hat{Q}(\alpha) \approx \hat{Q}(\alpha) + |\hat{Q}(\alpha) - Q^{\ast}(\alpha)| \). This would strike a balance between (i) reducing the Transient Performance \( \hat{Q}(\alpha) \) and (ii) keeping this Transient Performance close to \( Q^{\ast}(\alpha) \), which is the quantity one actually wishes to minimize.
Identifying $\nu$ and $w$ that approximate (8) is not straightforward because $Q^* (\alpha_i)$ is only known at the end of Algorithm 2, but $\nu$ and $w$ need to be known at its start. Progress can be made by exploiting the iterative formulation of Algorithm 1, where one has in each Improvement Step access to all realizations of Algorithm 2 evaluated in all previous Improvement Steps. The following Adaptation Problem is hence solved in every Improvement Step $R > 1$ of Algorithm 1:

$$\text{Adaptation Problem } \min_{\nu, w} \sum_{r=1}^{R-1} \sum_{n=M+1}^{N^{(r)}} \left( \| \tilde{Q}^{(r)}_{\alpha_i} - Q^{(r)}_{\alpha_i} \| - (\nu \cdot e_{\alpha_i}^{(r)} + w \cdot u_{\alpha_i}^{(r)}) \right)^2,$$

s.t. $\nu, w \geq 0,$

which is composed of the following elements. The outer sum runs over all so far completed Improvement Steps. In each Improvement Step $r = 1 \ldots R - 1$, Algorithm 2 was used to approximately solve the corresponding Selection Problem. The inner sum runs over the iterations $n = M + 1 \ldots N^{(r)}$ of that realization of Algorithm 2, with $N^{(r)}$ being the respectively last iteration. In each such iteration, one Approximate Selection Problem was solved, and each addend within the double-sum evaluates how closely its solution matches (8): $\tilde{Q}^{(r)}_{\alpha_i}$, $e_{\alpha_i}^{(r)}$, $u_{\alpha_i}^{(r)}$ are the optimal Transient Performance, Equilibrium Gap, and Uniformity Gap that were obtained when solving the considered Approximate Selection Problem (in Improvement Step $r$ and in iteration $n$ of Algorithm 2), and $Q^{(r)}_{\alpha_i}$ is the objective function value of the decision variable ultimately returned by Algorithm 2 (in Improvement Step $r$). Setting the weight $\gamma \in [0, 1]$ strictly below one has the effect that longer-ago Improvement Steps are eventually forgotten and hence enables greater adaptivity in the $\nu, w$ parameters.

Overall, the Adaptation Problem requires to set the discount parameter $\gamma$, solves for the parameters $\nu$ and $w$, and relies otherwise on readily available (since previously computed) numbers. This is considered advantageous because $\nu$ and $w$ are highly problem-specific, whereas the plausible value range for $\gamma$ is limited: Setting $\gamma$ in the range $[0.9, 1.0]$ yields discount weights in the range $[0.35, 1.0]$ after ten Improvement Steps and $[0.12, 1.0]$ after twenty Improvement Steps. For problems where the computational cost of one Improvement Step is in the order of one full simulation run (this will turn out to be a realistic assumption) and the computational budget consists of a few dozen simulation runs (as previously discussed), discount values in this range strike a sensible balance between adaptivity and exploitation of previously collected information.

This completes the specification of the proposed acceleration method. It is inserted into a given Embedding Solution Technique by specifying concrete instances of the Selection Problem and the Adaptation Problem in Algorithm 1. The Selection Problem is approximately solved by Algorithm 2. The Adaptation Problem, which is stated above, constitutes a two-dimensional least squares problem with simple bound constraints. In the absence of other information, the parameters $\nu$ and $w$ can be initialized with zero values, leading to a greedy first Improvement Step that does not account for out-of-equilibrium conditions or the presence of simulation stochasticity. All other operations of Algorithm 1 are problem-specific.

2.3. Discussion of problem and method formulation

2.3.1. Role of convex combinations in the Approximate Selection Problem

The objective function of the Approximate Selection Problem relies heavily on convex combinations. This can be given an intuitive meaning. Assuming for now a deterministic simulator, the information a single transient simulator state provides about the corresponding stationary state depends on how close the transient state has already come to that stationary state. Unless the transient state is already close to the stationary state, only little can be learned from it.

Given linear simulator dynamics, consider the following convex combination of decision variables $u_i$, and resulting simulator equilibrium points $\bar{x}_i = \bar{u}(u_i)$ (the index $i$ ranges from one to $N$ throughout):

$$\bar{x}_i = A\bar{x}_i + B u_i$$

$$\Rightarrow \left( \sum_i \alpha_i \bar{x}_i \right) = A \left( \sum_i \alpha_i \bar{x}_i \right) + B \left( \sum_i \alpha_i u_i \right).$$

It follows that using the decision variable combination $\sum_i \alpha_i u_i$ makes the simulator converge to the stationary state $\sum_i \alpha_i \bar{x}_i$. This is, however, only meaningful if these convex combinations can actually be processed by the simulator. If one interprets the convex combination coefficients $\alpha_i$ as probabilities (they are non-negative and sum up to one), then (i) the convex decision variable combination $\sum_i \alpha_i u_i$ coincides with the expected decision variable given that one samples decision variables according to the distribution ($\alpha_i$), and (ii) the corresponding convex stationary state combination $\sum_i \alpha_i \bar{x}_i$ coincides with the expected stationary state of the simulator given that one uses a randomly sampled decision variable from the distribution ($\alpha_i$) in each of its transitions. This follows from the linearity of both the simulator and the expectation operator.

This reasoning can be taken further. Consider the following convex combination of non-converged simulator states and transitions:

$$x_i = A(x_i - \Delta x_i) + B u_i$$

$$\Rightarrow \left( \sum_i \alpha_i x_i \right) = A \left( \sum_i \alpha_i x_i \right) + B \left( \sum_i \alpha_i u_i \right) - \left[ A \sum_i \alpha_i \Delta x_i \right].$$


If the square-bracketed term disappears then the remainder of the last equation coincides with (10). In this case, the stationary simulator state $\sum \alpha_i \bar{x}_i$ resulting from the linear decision variable combination $\sum \alpha_i u_i$ can be reconstructed from transient simulator states $\sum \alpha_i x_i$ only.

The $(v:\text{weighted})$ sum of Transient Performance and Equilibrium Gap in the Approximate Selection Problem can hence be interpreted as balancing two objectives: (i) Minimizing the Transient Performance as a proxy for the expected stationary objective function value that would result from continuing to draw decision variables according to the distribution $(\alpha_i)$; (ii) minimizing the Equilibrium Gap as a measure of the approximation error made in (i). Accounting for simulation stochasticity by including the Uniformity Gap has the additional effect of pulling $(\alpha_i)$ somewhat towards uniformity but does otherwise not invalidate the intuition developed here.

\subsection{Interpretation as a column generation heuristic}

Column generation aims at solving (often linear and integer) optimization problems with an unmanageable large number of decision variables (e.g. Lübbecke and Desrosiers, 2005). It can be sketched as iterating between a Master Problem and a Subproblem, where the Master Problem solves the problem at hand for only a small subset of decision variables and the Subproblem updates this subset.

To clarify the relationship between column generation and the proposed method, the objective function of the Approximate Selection Problem is rewritten as

$$\tilde{Q}(\alpha) = [Q(x_1) \cdots Q(x_N)]\alpha + v \cdot ||\Delta x_1 \cdots \Delta x_N||\alpha] + w \cdot \alpha^T \alpha.$$

This emphasizes that the Approximate Selection Problem is (for a given objective function $Q$ and parameters $v, w$) fully characterized by the Simulator Transitions $(x_i, \Delta x_i, u_i)$. Further, the by Assumption 2 linear Simulator Transition constraints are reformulated as follows:

$$x_i = A(x_i - \Delta x_i) + Bu_i + e_i, \quad i = 1 \ldots N$$

\begin{equation}
\Leftrightarrow \sum_{i=1}^{N} \alpha_i x_i = \sum_{i=1}^{N} \alpha_i [(x_i - \Delta x_i) + Bu_i + e_i] \quad \forall \alpha \in \Omega_{\text{relaxed}}.
\end{equation}

Letting

$$r_i = (I - A)x_i + A\Delta x_i - Bu_i - e_i, \quad i = 1 \ldots N,$

the constraint (15) can be rewritten as

$$[r_1 \cdots r_N] \alpha = 0 \quad \forall \alpha \in \Omega_{\text{relaxed}}.$$}

Adding one Simulator Transition $(x_{N+1}, \Delta x_{N+1}, u_{N+1})$ to the Approximate Selection Problem hence amounts to adding one column to the constraint matrix $[r_1 \cdots r_N]$ of (17) as well as to $[Q(x_1) \cdots Q(x_N)]$ and $[\Delta x_1 \cdots \Delta x_N]$ in (13). This motivates an interpretation of Algorithm 2 as a column generation heuristic for a nonlinear and stochastic problem, with the Approximate Selection Problem constituting the Master Problem and the Subproblem being solved by (i) the randomized selection of a simulation trajectory, combined with (ii) the advancement of the corresponding simulation by one iteration. It may be possible to develop and exploit this relationship with the column generation framework further.

\subsection{Interpretation as a ranking & selection (RS) heuristic}

Given a finite set of alternatives and means to repeatedly evaluate a stochastic performance measure for each alternative, RS techniques aim at efficiently identifying a near-optimal alternative (Swisher et al., 2003; Kim and Nelson, 2006b; Kim, 2013). There exist RS methods for stationary simulation processes, where computational efficiency is pursued by evaluating these processes one simulation iteration at a time (Goldsmann et al., 2002). RS techniques have been inserted into iterative search procedures such as Algorithm 1 (Hong et al., 2015). Overall, one may interpret the proposed method as an RS heuristic. There are, however, important differences.

There appears to exist no RS technique for the optimization of stationary performance measures that exploits information collected from transient (not-yet-converged) simulator iterations; the only treatment of transients mentioned in the relatively recent review of Kim (2013) is to discard them. This also holds for the $K^+\text{**}$ algorithm (Kim and Nelson, 2006a), a current standard in the field. Arguably, the approach of dropping transients has been adopted because dropping the stationary assumption instead would have hindered a mathematically rigorous development. The present article develops a method around the very idea of exploiting simulator transients. Mathematical tractability is maintained by adopting a number of (in particular, linear) approximations that are untypical for RS.

RS methods are designed to process raw stochastic simulator responses and to make their own inference about the statistical significance of the corresponding mean performance measures. Indeed, much of the complexity of going from independent samples to stationary simulation processes stems from the necessity to estimate the corresponding mean value variances from correlated samples (Goldsmann et al., 2002). The present work assumes simulator convergence to be identified by an external and problem-specific logic.
The proposed method has not been developed as a stand-alone (RS) technique but specifically for use within the search Algorithm 1. In this context, it is not meant to identify differences between arbitrary simulator configurations (as it is typically assumed in RS) but only between relatively small variations of a given configuration; as explained above, this adds credibility to its linearity assumptions. The accompanying Adaptation Problem is solved by a self-tuning mechanism that can be expected to at least in parts compensate for the adopted approximations.

2.3.4. Computational considerations

The dimension of the Approximate Selection Problem’s solution space equals the number of Simulator Transitions because there is one element in the Approximate Selection Problem’s decision vector \( \mathbf{a} \) for each Simulator Transition. Still, the computational effort of solving the problem can be expected to increase only slowly with that dimension. This is so because the objective function of Algorithm 2’s Nth Approximate Selection Problem results from adding a single term to the sums in the Transient Performance, Equilibrium Gap, and Uniformity Gap of the \((N-1)\)th problem. One can hence use the \((N-1)\)th solution vector with an added zero element \((a_N = 0)\) as a near-optimal starting point for the Nth problem.

The dimensionality of the simulator state space also has only a limited effect because no vector operations are needed while solving the Approximate Selection Problem. This follows from rewriting the Equilibrium Gap as \(\text{eg} (\mathbf{a}) = \sqrt{\sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \alpha_i \alpha_j \Delta \mathbf{x}_i^T \Delta \mathbf{x}_j}\) and observing that all involved inner products can be computed once prior to solving the problem. Increasing the number of involved state transition vectors by adding Simulator Transitions also does not add much complexity: Most of the inner products occurring in the Nth Equilibrium Gap have already been evaluated in the \((N-1)\)th Equilibrium Gap.

The Approximate Selection Problem is convex with linear constraints; the method of Frank and Wolfe (1956) is used to solve it in the subsequently presented experiments.

3. Experiments: dynamic road pricing with a complex simulator

The purpose of these experiments is to demonstrate how the proposed acceleration method helps to tackle a non-trivial optimization problem in conjunction with a detailed stochastic transport microsimulation that captures mobility at the level of individual travelers (“agents”).

3.1. Scenario description

The simulation scenario is based on the network shown in Fig. 1. All links are bi-directional and have identical outflow capacities of 1500 vehicles per hour and maximum velocities of 60 km/h, respectively. The (links connecting the) nodes 7, 3, 4, 8 constitute a city center, and the link directly connecting nodes 2 and 5 represents a bypass around that center.

5000 simulated travelers live at node 1 and work at node 6; these are subsequently referred to as the “remote citizens”. Another 5000 travelers live at node 7 and work at node 8; these are called the “central citizens”. The travel demand is dynamic in the following sense: Every traveler has to implement a home/work/home travel plan. The desired duration of the work activity is individual-specific and uniformly distributed between two and twelve hours. Performing the work activity is possible between 7:30 and 17:30, meaning that every simulated traveler has to select two departure times (one for the trip to work and one back home).

The travelers also have to choose a route to get to work and to return back home. The central citizens have no sensible alternative to traveling along the nodes 7, 3, 4, 8, whereas the remote citizens can select to either travel through the city center or take the bypass. Since the bypass is longer than the way through the city center, it only becomes an attractive alternative when the congestion-induced delay in the city center reaches a level that matches the detour travel duration. This leads to heavy congestion in the city center, from which also the central citizens suffer.

The introduction of a time-dependent cordon toll that is to be paid whenever crossing a link between nodes 2 and 3 and between nodes 4 and 5 is hence considered. Practical considerations (loosely inspired by the real Stockholm congestion charges) lead to the following requirements: (i) The toll levels are non-negative multiples of 0.25 Euro; (ii) there are at most three non-zero toll levels, subsequently called level one, two, and three; (iii) the toll changes only at discrete points in time that are located on a 30-min time grid. A concrete decision variable specification that complies with these requirements is given further below in Section 3.3.
3.2. Simulation configuration

The travel plan of each individual in the system consists of a departure time and route from home to work, and a departure time and route from work back home. Every simulated person (agent) \(a = 1 \ldots A\) attempts to maximize a utility function that assigns the following systematic utility to travel plan \(i\):

\[
V_{ai} = \beta_{\text{activity}} \cdot [t_{\text{home.planned},a} \cdot \log(t_{\text{home.realized},i}) + t_{\text{work.planned},a} \cdot \log(t_{\text{work.realized},i})] + \beta_{\text{toll}} \cdot \text{toll},
\]

where \(t_{\text{work.planned},a}\) and \(t_{\text{home.planned},a}\) are individual \(a\)'s desired work and home duration, \(t_{\text{work.realized},i}\) and \(t_{\text{home.realized},i}\) are travel plan \(i\)'s actually realized durations, \(\text{toll}\) is the toll paid when pursuing plan \(i\), and \(\beta_{\text{activity}} = 10 \text{ h}^{-1}\) and \(\beta_{\text{toll}} = 1 \text{ Euro}^{-1}\) are fixed parameters. Travel time is implicitly accounted for because time spent travelling is not available for activity implementation. The concrete functional form of this utility function is such that maximizing utility subject to a fixed time budget (length of a day minus a fixed travel time) and in the absence of a toll allocates to each activity an amount of time that is proportional to its desired duration. Linearizing (18) under the assumption that desired and realized activity durations coincide, one obtains a time value of \(\beta_{\text{activity}}/\beta_{\text{toll}} = 10 \text{ EUR/h.}\) meaning that a simulated traveler would accept to pay a toll of one Euro if this leads to a time saving of at least 6 min.

The simulation logic consists of the following steps.

1. Every traveler is given an \textit{ad hoc} generated initial travel plan. The maximum plan choice set size per traveler is five; whenever this size is exceeded by the creation of a new travel plan, the so far worst travel plan is discarded from the choice set.

2. Iterate:
   
   (a) Every traveler selects a travel plan according to the following rules.

   i. With probability 0.05, a new plan is generated by randomly varying the departure times of an existing plan. This is realized by adding uniform random variables from the interval \([-7200, 7200]\) s. The new plan is added to the traveler’s plan choice set and chosen for execution.

   ii. With probability 0.05, a new plan is generated by recomputing the paths of an existing plan. The new paths are what would have been fastest paths based on the realized travel times of the previous simulation iteration. The new plan is added to the traveler’s plan choice set and chosen for execution.

   iii. With probability 0.9, an existing plan is chosen from the choice set of previously generated plans according to a multinomial logit model.

(b) All travelers execute their chosen plans; network flows are simulated with a dynamic queueing model with spill-back.

(c) All travelers update the utility of their most recently executed plan based on the actually experienced travel times and tolls.

The way in which new plans are generated implies that the plans available to a given individual can differ in their routes and departure times. Many details of this logic are omitted; the deployed simulation system (MATSim, \texttt{www.matsim.org}) is rather complex. Comprehensive information can be found in Horni et al. (2016).

It was found experimentally that 500 simulation iterations are sufficient to reach the proximity of a stationary state and that averaging the simulation outputs over another 500 iterations allows to evaluate objective function values with a precision in the order of one decimal digit. A simulation is hence considered converged after 1000 iterations, and its expected stationary state is approximated by averaging over the last 500 iterations.

3.3. Objective function and Embedding Solution Technique

The objective function to be minimized is defined as

\[
Q = \frac{1}{A} \sum_{a=1}^{A} (0.9 \cdot \beta_{\text{toll}} \cdot T_a - U_a)
\]

where \(T_a\) and \(U_a\) are the actually experienced toll and travel plan utility of agent \(a\) in a given simulation iteration. If the individual addends in this sum were given by \(1.0 \cdot \beta_{\text{toll}} \cdot T_a - U_a\) then the toll would effectively cancel out in the objective function because it also enters \(U_a\) with a negative sign, cf. (18). This would reflect that the collected toll could be fully redistributed and hence would not constitute a societal cost. Weighting the toll in (19) by 0.9 implies that 10 percent of that toll cannot be redistributed; this is meant to reflect the operating cost of the toll collection infrastructure. The objective function can be evaluated directly from the simulator outputs; there is no need to reconstruct it from a vector-valued state space approximation as it was formally assumed during the earlier method development.

The time structure and levels of the toll are parametrized as follows, resulting in a 13-dimensional discrete decision variable vector (3 levels on a 0.25 Euro grid, 10 switching times on a 30 min grid).

- No toll before the morning peak.
- Five switching times during the morning peak: start of level 1 toll, start of level 2 toll, start of level 3 toll, back to level 2 toll, back to level 1 toll.
• Level 1 toll persists between morning and evening peak.
• Five switching times during the evening peak: start of level 2 toll, start of level 3 toll, back to level 2 toll, back to level 1 toll, back to no toll.
• No toll after the evening peak.

Candidate decision variables are created through random variations of the currently best solution as follows. Every toll level is drawn uniformly from the set \{\theta − 25\text{ cent}, \theta, \theta + 25\text{ cent}\} with \theta being the respective toll level value of the currently best solution. Every switching time is drawn uniformly from the set \{t − 30\text{ min}, t, t + 30\text{ min}\} with t being the respective switching time of the currently best solution. For every new solution that is generated by this logic, a second solution is created using the opposite toll and time variations. No distinct decision variable is created more than once. If a value is resampled, this draw is discarded. The toll levels and switching times within every newly generated solution are arranged such that they are in increasing order. Overall, there are about $3^{13} \approx 10^9$ possible variations of a given decision variable because there are three variation levels for each of the 13 decision variable elements.

The Approximate Selection Problem relies on a real-valued vector representation of the simulator state space. For the given microsimulation system (and arguably many others), this can only approximately be achieved: The transition from one simulation iteration to the next is defined by the information based on which each simulated traveler decides which travel plan to execute in the upcoming iteration. If the plan choice sets were fixed and the plan choice model only evaluated plan utilities, then this information could be represented by the vector of utilities of all plans in the choice sets of all agents. In the present example, however, one would have to additionally account for all information entering the plan choice set update mechanism. An approximation of the state space is hence constructed. It consists of stacking, for all 16 unidirectional links in the network, their time-dependent occupancy realizations (in vehicles) from the last simulation iteration into one vector, with time being discretized into 30-min bins. This results in an overall $16 \cdot 48 = 768$-dimensional state space.

Different candidate solution set sizes $M$ are used in different experiments; an upper limit for $M$ results from allocating at most 10 GB of Random Access Memory to an experiment. The discounting weight $\gamma$ used in the Adaptation Problem is set to 0.95. The initial solution used in all experiments is an all-zeros toll with the switching times set to the full hours 8, 9, 10, 11, 12, 14, 15, 16, 17, 18. Not varying the initial point makes it likely that at most a local (near-)optimum can be identified, which is considered acceptable given that the purpose of this study is to demonstrate the proposed acceleration method and not to globally explore the case study objective function.

As explained in Section 1, the proposed acceleration method is designed to be compatible with a broad class of Embedding Solution Techniques. To demonstrate its added value with greatest clarity, a very basic instance of Algorithm 1 is used in all experiments where (i) new candidate decision variables are generated by random variations of the currently best decision variable as explained above; (ii) the currently best decision variable is not added to the candidate set; and (iii) the currently best decision variable is set to the best candidate decision variable only if that yields an improvement. The initial algorithm control parameters are set to $B^{(0)} = (vw) = (00)$.

A single execution of Algorithm 1 is subsequently referred to as an optimization run (even though no optimality claims are made). A computational budget of 20,000 simulation transitions is allowed for in each optimization run, corresponding to the computational cost of evaluating twenty full simulation runs. This budget is evaluated as follows. Each optimization run keeps track of the total number of evaluated simulator transitions. A new Improvement Step is started only if this total number is still within the computational budget, otherwise the algorithm terminates. Since the number of simulator transitions evaluated per Improvement Step is variable, an optimization run may somewhat exceed the computational budget. The motivation for this setting is to be able to evaluate the result of the last Improvement Step, which would be wasted computing time if terminated half-way.

The considered overall problem is of nontrivial size and complexity. The network is small, but the number of simulated travelers is not. Every single traveler is simulated as an individual decision maker with an individual plan choice set and an individual desired working duration, meaning that simulating the travel choices of a population of 10,000 travelers is, in coarse but not completely unrealistic approximation, computationally as demanding as simulating route and departure time choice for 20,000 (number of travelers times number of trips per traveler) origin/destination pairs. The total number of toll switching time combinations is $\binom{39}{0} > 2.8 \cdot 10^9$ because there are 49 possible switching times (00:00, 00:30, ..., 23:30, 24:00) out of which 10 are to be selected. The total number of possible toll level combinations coming on top of this is infinite because there is no upper bound on the toll levels (0 cents, 25 cents, 50 cents, etc.). Computational performance is subsequently measured in terms of “number of evaluated simulator transitions” in order to decouple the analysis as much as possible from the concrete scenario settings and the used simulation program.

3.4. Results

Each experiment consists of ten replications of an optimization run. The replications within an experiment differ only in their random seed. Seven experiments are performed to investigate the performance of the proposed acceleration method when inserted into the given Embedding Solution Technique. These experiments differ only in the candidate solution set size $M = 2, 4, 8, 16, 32, 64, 128$. Figs. 2–8 display the results. Each figure consists of three subfigures. Subfigure (a) shows the evolution of the objective function value of the currently best solution over the number of evaluated simulator transitions.
Fig. 2. Candidate solution set size $M = 2$.

Fig. 3. Candidate solution set size $M = 4$.

Fig. 4. Candidate solution set size $M = 8$.

(in thousands). Counting from the left, the $r$th box in this figure summarizes the situation after the $r$th Improvement Step. Its lower and upper edge indicate the 25% and 75% percentile of the empirical distribution of the objective function values over all ten replications. Its left and right edge indicate the 25% and 75% percentile of the empirical distribution of the total number of evaluated simulation transitions over all ten replications. The cross inside the box is located at the respective 50% percentiles. Box and cross are only drawn if all ten replications are available to compute the percentiles; this may not be the case towards the end of an experiment because the number of simulator transitions evaluated per Improvement Steps varies,
meaning that some replications may evaluate less Improvement Steps than others before exhausting their computational budget. The small circles indicate the results of the individual replications. Two dotted lines are added to guide the eye during the following discussion. Subfigures (b) and (c) show, in the same manner, the evolution of the Equilibrium Gap weight \( v \) and of the Uniformity Gap weight \( w \), respectively.

The objective function value dynamics (subfigures (a)) are considered first. One can overall observe that increasing the candidate solution set size \( M \) from two to four and from four to eight leads to distinct improvements in search performance and that increasing \( M \) even further leads to only small and eventually diminishing improvements. The same trend can
be observed for the variability obtained per Improvement Step: The larger $M$, the smaller this variability, with the further reduction becoming relatively small as one goes beyond $M = 8$. Given that the experiments with $M \geq 8$ yield very similar results, it is postulated that $M \geq 8$ is an adequate parametrization for this problem and the following discussion focuses exclusively on these experiments.

All but one optimization run come very close to a final objective function value of $-192.7$, which is indicated by the horizontal dotted line. The single exception, which is graphically represented by the upper circle on the very right-hand side of Fig. 6a, still reaches an objective function value of $-192.4$ after overall 23,351 simulator transitions. The majority of optimization runs obtains objective function values below $-192.0$ within half of the computational budget (the equivalent of ten simulation runs, indicated by the vertical dotted line). It appears likely that a local minimum with an objective function value around $-192.7$ exists and that the search approaches overall reliably the proximity of this solution.

The Equilibrium Gap weights (subfigures (b)) and Uniformity Gap weights (subfigures (c)) exhibit all comparable trends, in that the Equilibrium Gap weight first jumps to a relatively large value and then decays again whereas the Uniformity Gap weight grows upwards at about the same relative pace as the Equilibrium Gap weight decays. This can be explained by the fact that each optimization run, both in terms of decision variables and simulator dynamics, first goes through a transient phase and then settles in a stationary phase. During the transient phase, the Equilibrium Gap dominates the Uniformity Gap in the objective function of the Approximate Selection Problem, whereas during the stationary phase it is the other way around. This is plausibly reflected by the corresponding weights.

The Equilibrium and Uniformity Gap weights exhibit relatively large within-experiment variability when compared to the respective objective function variability. This can be explained and justified by the following re-writing of the Equilibrium Gap (all sum indices run from 1 to $N$):

$$
\text{eg}(\alpha) = \sqrt{\sum_i \alpha_i^2 \|\Delta x_i\|^2 + \sum_{i \neq j} \alpha_i \alpha_j \Delta x_i \cdot \Delta x_j}.
$$

(20)

If all state space transitions were of same magnitude ($\|\Delta x_i\| = \|\Delta x_j\| \forall i, j$) and orthogonal to each other ($\Delta x_i \cdot \Delta x_j = 0 \forall i \neq j$) then the above expression would collapse into $\text{eg}(\alpha) \propto \sqrt{\sum_i \alpha_i^2} = \sqrt{\text{ug}(\alpha)}$, meaning that Equilibrium and Uniformity Gap were perfect substitutes for each other. This would imply that the regression parameters $v$ and $w$ in the Adaptation Problem were not separately identifiable; increasing one and decreasing the other correspondingly would not change the Adaptation Problem’s objective function. Indeed, a Pearson’s correlation coefficient of $-0.803$ between the $v$ and $w$ values obtained at the end of all optimization runs with $M \geq 8$ can be observed. However, given that the Uniformity and the Equilibrium Gap capture systematically different effects (transient conditions vs. the presence of stochasticity), it is in general advisable to use separate $v$ and $w$ parameters. A possible dependence between them does not impair the Approximate Selection Problem, which only evaluates them in combination. Only if the simulator is deterministic (noise-free), it makes sense to constrain $w$ to zero.

Fig. 9 illustrates the simulated transport reality behind the experiment with $M = 128$. The Box plots in Fig. 9a summarize the obtained toll levels. Their variability appears relatively larger than that of the corresponding objective function values in Fig. 8; this suggests that the objective function is at least in some directions relatively flat near the identified solution region. Fig. 9b and c show Box plots of the hourly detour flows without and with the toll, respectively. Continuous lines are added to guide the eye. The toll has clearly the desired effect of pushing traffic out of the city center and onto the detour route.

Finally, the question is addressed if the proposed acceleration method provides a computational benefit. For this, the acceleration method is removed from the Embedding Solution Technique. For simplicity, the Embedding Solution Technique with/without the proposed acceleration method is subsequently referred to as the accelerated/reference algorithm. The
reference algorithm differs from the accelerated algorithm only in that it tackles the Selection Problem by evaluating each candidate decision variable based on one full simulation run. Maintaining the computational budget of 20 full simulation runs per optimization run, only relatively small candidate decision variable sets of size \( M = 2 \) and \( M = 4 \) are used; this allow for \( 20/2 = 10 \) respectively \( 20/4 = 5 \) Improvement Steps per reference optimization run. Given the fixed computational budget, the use of even larger candidate decision variable set sizes in the reference algorithm would allow for even less Improvement Steps and would hence give the reference algorithm even less opportunity to move away from the initial solution point. In other words, using a relatively small \( M \) in the reference algorithm is not arbitrary but a necessary consequence of the fact that it evaluates full simulation trajectories.

Fig. 10 compares the accelerated algorithm with \( M = 128 \) (same data as shown in Fig. 8a) to the reference algorithm with \( M = 2 \) in Fig. 10a and \( M = 4 \) in Fig. 10b. The individual objective function values of the reference algorithm are indicated by triangles, those of the accelerated algorithm as before by circles. The layout of the 25/75\% percentile boxes is identical in either case; those corresponding to the accelerated algorithm can be identified by being located systematically below those of the reference algorithm. Only two out of twenty reference runs (identifiable as the lowest triangle sequences in Fig. 10a) come close to the previously identified objective function value of -192.7 by reaching values of –192.3 and –192.5,
respectively. Overall, the reference algorithm exhibits a less distinct trend and greater performance spread across replications, indicating that the accelerated algorithm performs a more directed search. No hypothesis testing is needed to conclude that the accelerated algorithm delivers much better solutions than the reference algorithm for the same computational budget.

Fig. 11 extends Fig. 10 in that it gives the reference algorithm a computational budget of 100,000 simulation transitions. The reference algorithm gets relatively quickly stuck in very different solutions for both $M = 2$ and $M = 4$. The accelerated algorithm is able to use much more ($M = 128$) candidate decision variables, and its Approximate Selection Problem also considers all possible convex combinations of the simulator states visited with these decision variables. As explained in Section 2.3.1, this can be interpreted as searching through all convex combinations of the candidate decision variables. The ability to perform a much more diverse search than the reference algorithm appears to protect the accelerated algorithm from getting stuck at inferior solutions.

It needs to be acknowledged that a performance assessment in terms of simulator transitions neglects the computational overhead of solving a sequence of Approximate Selection Problems in the proposed acceleration method. As explained in Section 2.3.4, this overhead can in general be assumed to be relatively insensitive to the number of evaluated Simulator Transitions and to the dimensionality of the simulator state space. This means that the larger the considered simulation system is (i.e. the longer the evaluation of a single simulation transition takes), the more negligible this overhead becomes.

3.5. Implications for further experimentation

The present experiments illustrate the added value of the proposed acceleration method when being inserted into a very basic Embedding Solution Technique. Little additional insight about the proposed acceleration method can be expected from comparing the present combination of acceleration method and Embedding Solution Technique to an alternative search technique from the literature because performance differences could not be related to the acceleration method in isolation. Future experimentation could instead aim at (i) comparing the proposed method to alternative (approximate) Selection Problem solvers or at (ii) inserting the proposed method into more sophisticated Embedding Solution Techniques. Some insights along both directions can already be obtained from the present experiments.
A comparison of the proposed method to a ranking & selection (RS, cf. Section 2.3.3) technique is of interest. As in the present experiments, this would require to first specify an Embedding Solution Technique and to then tackle the corresponding Selection Problem either with the proposed method or with the RS technique. Recalling that there appears to currently exist no RS technique that can process transient simulator responses and that RS is not useful if the simulator responses are deterministic, one can already now state the following.

1. Before a given Selection Problem can be tackled with an RS technique, the simulation trajectories of all candidate decision variables have to be run to stationarity. This overhead becomes more prominent the more iterations the simulation needs to attain stationarity.
2. RS is designed to quickly distinguish unknown expected values based on stochastic (and stationary) realizations. This means that the potential benefit of an RS technique increases with the stochasticity of the simulator responses.
3. RS does not make the linearity assumptions of the proposed acceleration method. This means that it is compatible with very explorative candidate decision variable generation strategies, where the linearity assumptions of the proposed method may break down.

When it comes to inserting the proposed acceleration method into more advanced Embedding Solution Techniques, one may consider (variations of) Simulated Annealing (SA; Kirkpatrick et al., 1983). The genuine SA logic creates only a single candidate decision variable at a time and may even make deterioration steps (i.e. it may switch to inferior candidates) according to a probabilistic logic that aims at escaping local minima. In the present experiments, the reference algorithm would likely have benefited from such a mechanism given its tendency to get stuck at inferior solutions. It is noteworthy that inserting the proposed acceleration method does not only improve the initial search speed but also avoids stagnation in local solutions, despite of the greedy Embedding Solution Technique that only accepts improved candidate decision variables. This first indication that the proposed acceleration method helps to escape inferior solutions without the additional computational cost of making deterioration steps may render it particularly compatible with otherwise greedy search strategies.

4. Summary and directions for future research

This work proposes a new method to accelerate the approximate solution of optimization problems with real-valued and/or discrete decision variables subject to equilibrium constraints that are given in the form of a possibly stochastic simulation process. The method is based on a new problem approximation for which several desirable properties are proven under somewhat idealized but not implausible assumptions. The entire approach is constructed such that it can be inserted into a broad class of Embedding Solution Techniques and such that it can be used in conjunction with general simulators, even if their black-box implementation prohibits an analysis of their mathematical properties; it is sufficient to be able to iterate the simulator towards a deterministic or stochastic equilibrium. The feasibility to use the new acceleration method for real problem solving is demonstrated through a single but comprehensive simulation case study, which indicates that substantial computational speedups can be achieved. Clearly, the experimental validation of a new method based on one (or two, or three) simulation studies can only be anecdotic; an expectation that the proposed method will perform well in a broader problem range is based on the proven mathematical properties of the problem approximation it uses.

The considered class of stochastic process simulators, formalized in (1), allows for stochasticity in simulator states but not in simulator input parameters (such as road capacities). It may be feasible to develop the framework further such that input stochasticity can also be accounted for; possibly resulting in a robust optimization approach.

Additional performance gains can be expected from combining the proposed acceleration method with more sophisticated Embedding Solution Techniques than the basic random search used in the present article. Specialized further developments, for instance for only real-valued decision variables or for macroscopic simulators that do not exhibit simulation noise, may be feasible. Variations of Algorithm 2’s approximate solution of the Selection Problem, for instance by sampling decision variables differently than proportionally to their \( \alpha \) coefficients, and alternative formulations of the corresponding Adaptation Problem may yield further speedups.

Overall, this work is driven by the need to deal with real and computationally heavy transport simulation model systems. Approximations and simplifications have hence been adopted. A more rigorous theoretical analysis, in particular the establishment of performance bounds in transient conditions, is equally desirable as it appears challenging. The close relationship of the proposed method with column generation and ranking & selection techniques is yet to be fully explored.

Acknowledgments

Michel Bierlaire suggested the column generation interpretation. This research was in parts funded by the ERA-NET Transport III Flagship Call 2013 “Future Traveling” and Vinnova - the Swedish Governmental Agency for Innovation Systems, project “Smart Adaptive Public Transport (SMART-PT)”. 
Appendix A. Proof of Proposition 2

The by Assumption 3 linear objective function $c^T \sum_{i=1}^{N} \alpha_i \tilde{x}_i$ of the Relaxed Selection Problem evaluates a convex combination $\sum_{i=1}^{N} \alpha_i x_i$ of expected stationary simulator states. This convex combination is approximated in the Transient Performance term of the Approximate Selection Problem by a convex combination $\sum_{i=1}^{N} \alpha_i x_i$ of possibly transient and noisy simulator states, using the same $\alpha_i$ coefficients. Assume that the magnitude of the corresponding approximation error can be bounded by

$$\left\| \sum_{i=1}^{N} \alpha_i x_i - \sum_{i=1}^{N} \alpha_i \tilde{x}_i \right\| \leq \frac{1}{\|c\|} (v \cdot eg(\alpha) + w \cdot ug(\alpha))$$ \hspace{1cm} (21)

with all involved variables as in the Approximate Selection Problem. From this follows

$$Q_{exact}(\alpha) = c^T \sum_{i=1}^{N} \alpha_i \tilde{x}_i$$ \hspace{1cm} (22)

$$\leq \max_{z} c^T \left( \sum_{i=1}^{N} \alpha_i x_i + z \right) \text{ s.t. } \|z\| \leq \frac{1}{\|c\|} (v \cdot eg(\alpha) + w \cdot ug(\alpha))$$ \hspace{1cm} (23)

$$= c^T \sum_{i=1}^{N} \alpha_i x_i + v \cdot eg(\alpha) + w \cdot ug(\alpha)$$ \hspace{1cm} (24)

$$= \tilde{Q}(\alpha).$$ \hspace{1cm} (25)

Proposition 2 bounds the probability that this inequality does not hold. Its pre-condition (21) is hence analyzed in the following. For this, the following consequence of the triangular inequality is observed:

$$\left\| \sum_{i=1}^{N} \alpha_i x_i - \sum_{i=1}^{N} \alpha_i \tilde{x}_i \right\| = \left\| \sum_{i=1}^{N} \alpha_i e_i + \sum_{i=1}^{N} \alpha_i (x_i - e_i - \tilde{x}_i) \right\|$$ \hspace{1cm} (26)

$$\leq \left\| \sum_{i=1}^{N} \alpha_i e_i \right\| + \left\| \sum_{i=1}^{N} \alpha_i (x_i - e_i - \tilde{x}_i) \right\|. \hspace{1cm} (27)

The two addends on the right-hand side (r.h.s.) of (27) are now individually bounded from above.

A1. A bound on the first addend on the r.h.s. of (27)

A variation of Chebychev’s inequality by Ferentinos (1982) is used. Given a d-dimensional random vector $Z = (Z_j)$ and stacking the variance of each element of $Z$ in another vector $v = (\text{VAR}[Z_j])$, it reads as follows:

$$\text{Pr}(\|Z - E[Z]\| \geq \delta \|v\|) \leq 1/\delta^2.$$ \hspace{1cm} (28)

Noting that $\|v\| \leq \sum_{j=1}^{d} \text{VAR}[Z_j]$, it follows that

$$\text{Pr}(\|Z - E[Z]\| \geq \delta \sum_{j=1}^{d} \text{VAR}[Z_j]) \leq 1/\delta^2.$$ \hspace{1cm} (29)

Assume that the simulator has not yet been evaluated, meaning that the simulator noise sequence $e_1, \ldots, e_N$ is composed of a priori unknown random vectors. Letting $Z = \sum_{i=1}^{N} \alpha_i e_i$ implies by Assumption 1 that $E[Z] = 0$ and $\text{VAR}[Z_j] = \sum_{i=1}^{N} \alpha_i^2 \text{VAR}[e_{ij}] \leq \sigma_j^2 \sum_{i=1}^{N} \alpha_i^2$. Inserting this in (29), one obtains

$$\text{Pr}(\sum_{i=1}^{N} \alpha_i e_i \geq \delta \sum_{j=1}^{d} \sigma_j^2 \sum_{i=1}^{N} \alpha_i^2) \leq 1/\delta^2.$$ \hspace{1cm} (30)

This means that, prior to evaluating the simulator, the probability that the first addend on the r.h.s. of (27) exceeds $\delta \sum_{j=1}^{d} \sigma_j^2 \sum_{i=1}^{N} \alpha_i^2$ is at most $1/\delta^2$. 

A2. A bound on the second addend on the r.h.s. of (27)

Assumptions 2 in conjunction with (4) yields

\[ \sum_{i=1}^{N} \alpha_i(x_i - e_i - \bar{x}_i) = \left[ A \sum_{i=1}^{N} \alpha_i(\Delta x_i) + B \sum_{i=1}^{N} \alpha_i u_i \right] - \left[ \sum_{i=1}^{N} \alpha_i \bar{x}_i \right]. \tag{31} \]

The first term in square brackets is the result of a noise-free simulator transition \( \sum_{i=1}^{N} \alpha_i(\Delta x_i - e_i) \) starting out from the state \( \sum_{i=1}^{N} \alpha_i(x_i - \Delta x_i) \) and using the decision variable \( \sum_{i=1}^{N} \alpha_i u_i \). The second term in square brackets is the stationary point that would be attained if the noise-free simulator was run to convergence using the decision variable \( \sum_{i=1}^{N} \alpha_i u_i \). This again follows from the linearity Assumption 2, which implies that the stationary state resulting from a convex combination of decision variables is equal a convex combination of the stationary states resulting from each individual decision variable.

Eq. (31) hence represents the deviation between a transient state of a deterministic mapping and its by Assumption 4 existing and unique fixed point. Let \( L \in [0, 1) \) be the Lipschitz constant of this mapping. Banach’s fixed point theorem (e.g. Berinde, 2007, Chapter 2) allows to bound the magnitude of this deviation based on the length of the transition having led to the transient state:

\[ \left\| \sum_{i=1}^{N} \alpha_i \Delta x_i \right\| \leq \frac{L}{1-L} \left\| \sum_{i=1}^{N} \alpha_i \Delta x_i \right\| + \frac{L}{1-L} \left\| \sum_{i=1}^{N} \alpha_i e_i \right\|. \tag{32} \]

This yields together with (31) the following bound on the second addend on the r.h.s. of (27):

\[ \left\| \sum_{i=1}^{N} \alpha_i(x_i - e_i - \bar{x}_i) \right\| \leq \frac{L}{1-L} \left\| \sum_{i=1}^{N} \alpha_i \Delta x_i \right\| + \frac{L}{1-L} \left\| \sum_{i=1}^{N} \alpha_i e_i \right\|. \tag{34} \]

Combining (27), (30) and (34) finally leads to

\[ \text{Pr}\left( \left\| \sum_{i=1}^{N} \alpha_i x_i - \sum_{i=1}^{N} \alpha_i \bar{x}_i \right\| \geq \frac{1}{\|c\|} \left\| \sum_{i=1}^{N} \alpha_i \Delta x_i \right\| + \|c\| \left( 1 + \frac{L}{1-L} \delta \sum_{i=1}^{N} \sum_{i=1}^{N} \alpha_i^2 \right) \right\| \leq \frac{1}{\delta^2}. \tag{35} \]

which means in conjunction with Proposition 2’s definition of \( \nu \) and \( \omega \) that (21) is violated with probability of at most \( 1/\delta^2 \). The same then also holds for its implication (22)-(25), as claimed by Proposition 2.

Appendix B. Proof of Proposition 3

Let \( u^* \) be an optimal solution of the Exact Selection Problem and let the corresponding stochastic simulation process of length \( K \) comprise the state sequence \( \{x_k\} \), the state transitions \( \{\Delta x_i\} \) and the noise vectors \( \{e^*_i\} \). The by Assumption 2 linear simulator dynamics allow to decompose this process into a deterministic mean process and a zero-mean stochastic disturbance process:

\[ x^*_k = E[x^*_k] + y_k \tag{36} \]

\[ E[x^*_k] = \begin{cases} x_0 & \text{if } k = 0 \\ A E[x^*_{k-1}] + Bu^* & \text{otherwise} \end{cases} \tag{37} \]

\[ y_k = \begin{cases} 0 & \text{if } k = 0 \\ Ay_{k-1} + e^*_{k-1} & \text{otherwise}. \end{cases} \tag{38} \]

Inserting \( \tilde{\alpha} = (\tilde{\alpha}_i) \) with \( \tilde{\alpha}_i = 1(u_i = u^*)/K \) into the objective function of the Approximate Selection Problem leads to

\[ \hat{Q}(\tilde{\alpha}) = c^T \frac{1}{K} \sum_{k=1}^{K} x^*_k + v \left\| \frac{1}{K} \sum_{k=1}^{K} \Delta x^*_k \right\| + \frac{W}{K} \tag{39} \]

\[ = c^T \frac{1}{K} \sum_{k=1}^{K} E[x^*_k] + c^T \frac{1}{K} \sum_{k=1}^{K} y_k + v \left\| \frac{1}{K} (E[x^*_k] + y_k - x_0) \right\| + \frac{W}{K} \tag{40} \]
The first term in square brackets converges deterministically to $Q_k^{\text{relaxed}}$ as $K$ goes to infinity because $\frac{1}{K} \sum_{k=1}^{K} E[x_k^n | v] \rightarrow 0$ and $w/K$ go to zero. The second term converges in mean square to zero because $\frac{1}{K} \sum_{k=1}^{K} y_k + v' \frac{1}{K} y_k$ for $\hat{Q}(\alpha) \geq \min_{\alpha \in \Omega} \hat{Q}(\alpha)$ then follows Proposition 3.

References


In press, available online.


Smith, M., 1980. Local traffic control policy which automatically maximises the overall travel capacity of an urban road network. Traffic Eng Control 21 (6), 298–302.


