*Proceedings of the 2014 Winter Simulation Conference*
*A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.*

# PREDICTING HALFWAY THROUGH SIMULATION: EARLY SCENARIO EVALUATION USING INTERMEDIATE FEATURES OF AGENT-BASED SIMULATIONS

Satoshi Hara
Rudy Raymond
Tetsuro Morimura
Hidemasa Muta

IBM Research – Tokyo
5-6-52, Toyosu, Koto-ku
Tokyo, 135-8511, JAPAN

## ABSTRACT

Agent-based simulations are indisputably effective for analyzing complex processes such as traffic patterns and social systems. However, human experts often face the challenges in repeating the simulation many times when evaluating a large variety of scenarios. To reduce the computational burden, we propose an approach for inferring the end results in the middle of simulations. For each simulated scenario, we design a feature that compactly aggregates the agents' states over time. Given a sufficient number of such features we show how to accurately predict the end results without fully performing the simulations. Our experiments with traffic simulations confirmed that our approach achieved better accuracies than existing simulation metamodeling approaches that only use the inputs and outputs of the simulations. Our results imply that one can quickly evaluate all scenarios by performing full simulations on only a fraction of them, and partial simulations on the rest.

## 1 INTRODUCTION

Simulation is effective for many complex problems that do not have closed-form solutions, and which resist analytical, empirical, or statistical modelling approaches. Examples include designing materials with desirable properties that requires atomistic simulation methods, and deciding on the best action in managing traffic to handle future events, such as natural disasters, which require modelling traffic flows. Agent-based simulation is an effective tool for decision making in economic and social complex problems because it does not require detailed knowledge of the macro-dynamics. It suffices to define the behavioral rules of an agent, an autonomous entity with a specific set of properties (e.g., a car, a person, etc.), and then simulate the interactions of a large number of such agents to measure the emergent macro-level results using key performance indexes, or KPIs, such as the total length of the traffic jams and the $CO_2$ emissions. Differing from analytical and statistical models, simulation does not require overly simplistic assumptions.

There are many agent-based tools developed for traffic simulations (Gomes, May, and Horowitz 2004, Behrisch et al. 2011, Osogami et al. 2012). Guidelines for their rigorous use in marketing research (Rand and Rust 2011) have also accelerated their applications in analysing consumers purchasing behaviors (Libai et al. 2013). However when compared with analytical and statistical modelling, they suffer from high computational costs that limit the scale of the simulations and the number of scenarios that can be evaluated within a limited amount of decision time. These limitations can be severe when one needs to evaluate the worst-case guarantees for all of the possible scenarios. Even for a typical case when finding a good-enough scenario suffices, (heuristic) optimization algorithms need to run the simulation many times as the more scenarios simulated, the better the outcome.

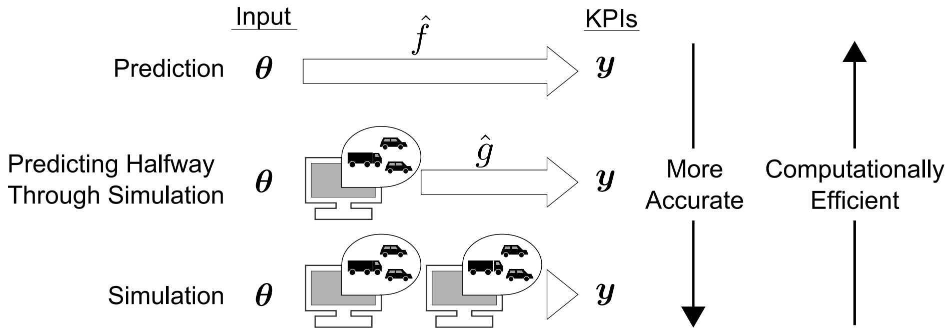



Figure 1: Three ways to evaluate KPIs: The proposed approach (Predicting Halfway Through Simulation) bridges the two extreme ends: Prediction and (full) Simulation. It balances two important properties: accuracy and computational efficiency.

In this paper, we address the challenges of quickly evaluating scenarios in agent-based simulations without fully performing them by using analytical and statistical approaches. Suppose we have fully performed agent-based simulations for $N$ different scenarios, and we want to evaluate another $M$ scenarios. The key questions are: (*i*) how to predict the KPIs of the $M$ scenarios, and (*ii*) how to find the best $K$ scenarios from the $N+M$ scenarios. Here, the prediction results and the best scenarios are measured against the ground truth KPIs that we would obtain if we fully simulated all of the scenarios, and the scenarios differ in the settings of simulation inputs.

Existing approaches, such as Kriging (Staum 2009) and Gaussian process regression (Rasmussen and Williams 2005) that predict the KPIs from the simulation inputs, may suffice when the relationships between the inputs and the KPIs are simple, and when there are sufficient number of past simulations to reveal the relationships. These approaches will answer questions (*i*) and (*ii*) by simply comparing the inputs of the $M$ scenarios against those of past simulations. However, in many cases of agent-based simulations, the relationships are complicated, and the past simulations only cover a small fraction of the possible scenarios. For example, it is hard to predict the effects of building a new road in reducing the traffic congestion in a city center when there were no past simulations of adding new roads. However, by running a simulation with the new road halfway, it is possible to compare its intermediate states with those of previous simulations.

We propose a framework for the prediction models built from past simulation results to infer the future simulation KPIs. In contrast to previous work, our proposal uses simulation intermediate data that consist of all of the agents' information at each time step. We refer to the framework as *predicting halfway through simulation*. The role of the intermediate data is clear from the observation that simulations starting with completely different parameters might result in similar evolutions of the agents' states, which eventually produce similar KPIs. For example, the traffic patterns after the construction of a new road (not included in the previous simulations) may turn out to be similar to those of a particular scenario simulated before. Our proposal can be considered as an attempt to find and use *emergence* within an agent-based simulation, which is one of grand challenges in the field according to Charles Macal in Taylor et al. (2013).

We note that our proposal lies between the two extremes of evaluating KPIs: Running $M$ full simulations at one end, and prediction without any additional simulations at the other end, as shown in Figure 1. The predicting halfway through simulation framework is expected to be more accurate than the predictions, and also more computationally efficient than the full simulations.

To summarize, our contribution in the domain of agent-based traffic simulation is twofold. First, we present a novel method to incorporate simulation intermediate data to predict the KPIs of ongoing simulations. To this end, we construct features from intermediate states of the simulations. The method allows us to quickly evaluate $M$ future scenarios by only simulating them to the midpoint to obtain their intermediate features. We then use these features to predict the KPIs of the $M$ simulation scenarios. The main challenge is in designing compact but representative features from high-dimensional intermediate states of the agents in the simulations.

Second, we show by experiments how our method can perform better than existing approaches for questions (*i*) and (*ii*). We performed our experiments with the agent-based traffic simulator Megaffic (Osogami et al. 2012) that runs on a massively parallel environment for simulating large numbers of agents. We studied the case when a traffic planner in Nairobi, Kenya, must evaluate 1,024 different scenarios of traffics at one hour later. We found that our method gives better predictions for the rest of the $M$ scenarios that are simulated only till the middle. The accuracies improved as the number of training scenarios $N$ and the number of time steps till the midpoint increases.

## 2 EARLY SCENARIO EVALUATION THROUGH PREDICTION

We present a framework of using intermediate states of agents in simulations that we think can be applied for quickly evaluating scenarios in various agent-based simulations. We refer to the framework as *predicting halfway through simulation*. Its specific details for traffic simulations will be given in the next section.

### 2.1 Agent-based Simulation

There are $B$ *agents* each having its own state $s_b(t) \in \mathscr{S}$ at time $t$ for $b = 1, 2, \ldots, B$ where $\mathscr{S}$ is some state space. At each time step $t$, an agent receives some feedback (from other agents or surrounding environments), responses with an action $a_b(t) \in \mathscr{A}$ in some action space $\mathscr{A}$, and changes its state into $s_b(t+1)$. The choices of actions and state changes are determined by a policy function and a transition function. Those functions determine the rules of behavior of agents that can give rise to interesting macro-level results.

The validity of an agent-based simulation is often tested by its macro-level summarized statistics, or KPIs, that are comparable to real-world observation, i.e., empirical output validation in Rand and Rust (2011). For example in traffics, they can be the length of roads with traffic congestion, or the number of cars passing through some roads within a time interval. Users often have to tune the rules of agents by many runs of simulations until the desired KPIs are derived. Each run (or scenario) consists of large time steps, and each time step is computationally intensive owing to updating every agent's states.

Existing approaches in simulation metamodeling evaluate the KPIs of a scenario from its inputs, or, its setting (such as the policy and transition functions, and initial environment). On the other hand, predicting halfway through simulation, our proposal, aims at inferring the KPIs during earlier time steps of a simulation by also incorporating the changes of $s_b(t)$ and $a_b(t)$.

### 2.2 Simulation Metamodeling

There is a large body of existing work in this direction that can be considered as function approximation. The method originates from the response surface methodology (Box and Draper 2006). Here, the simulator is cast as a black box function $f$ that outputs KPIs. Let $\theta \in \Theta$ be a simulation setting in the domain $\Theta$ and $y \in \mathbb{R}^P$ be a set of $P$ KPIs we want to compute. In the function approximation approach, we aim to construct a prediction function $\hat{f}$, as shown in the topmost of Figure 1, for the KPIs of the simulator $f$. This is essentially a prediction of KPIs $y$ from the inputs $\theta$.

The KPIs and inputs pairs from past simulations provide a training set for constructing the prediction function. Suppose we have fully conducted $N$ simulations and obtained their inputs-KPIs pairs $\{\theta^{(n)}, y^{(n)}\}_{n=1}^{N}$. We can use them to construct a function $\hat{f}$, $y \approx \hat{f}(\theta)$. On constructing the function $\hat{f}$, we can use existing methods such as Kriging (Beers and Kleijnen 2004, Kleijnen 2009) and Gaussian process regression (Rasmussen and Williams 2005).

### 2.3 Predicting Halfway Through Simulation

When prediction functions mentioned above cannot produce satisfactory results, the only choice is to run the simulation scenario. However, since the action and state of each agent in the simulation (or, simply intermediate states) is observable at each time step, there is a possibility to build a more reliable prediction function by running the simulation till some midpoint. If the predicted KPIs are far from satisfactory, we can halt the current simulation and move on to the next scenario. For some agent-based simulations that permit warm start, the halted scenario can be restarted later.

Agent-based simulations provide the intermediate states of their agents that influence the final KPIs, $y$, more closely than the inputs $\theta$. Formally, suppose we have an access to the past states and actions of each of $B$ agents until the time point $T$, $\Xi_T = \{\{s_b(t), a_b(t)\}_{b=1}^{B}\}_{t=1}^{T}$. Let $g$ be a subroutine of the simulator $f$ that generates KPIs from the inputs $\theta$ and the intermediate states $\Xi_T$. Our approach approximates the function $y = g(\theta, \Xi_T)$ with a function $\hat{g}$, as shown in the middle of Figure 1, namely, $y \approx \hat{g}(\theta, \Xi_T)$.

Applying general machine learning techniques that make use of $\Xi_T$ is challenging because $\Xi_T$ is high-dimensional due to the large number of agents and time steps. Some summarization of $\Xi_T$ is needed. We denote such a summarization function as $h$ that generates a compact feature $\xi = h(\Xi_T)$. The function $h$ depends on the problem domain and the forms of $s_b(t)$, $a_b(t)$. It is crucial for the performance of the proposed method, and should be tailored to the simulations.

Here, we describe one example of the function $h$ we use in Section 3.1.2. Suppose the state space and the action space are both discrete, which means, $\mathscr{S} = \{s_1, s_2, \ldots, s_{|\mathscr{S}|}\}$ and $\mathscr{A} = \{a_1, a_2, \ldots, a_{|\mathscr{A}|}\}$. We can then track the statistics of the agents at a fixed time interval: the number of agents in each state and the number of agents taking each action. Specifically, we denote the number of agents in the state $s_i$ at the $q$th time interval by $\xi_{\mathrm{s},i,q}$, and the number of agents taking the action $a_{i'}$ at the $q'$th time interval by $\xi_{\mathrm{a},i',q'}$. Hence, $\xi_{\mathrm{s}}$ and $\xi_{\mathrm{a}}$ are $|\mathscr{S}| \times Q$ and $|\mathscr{A}| \times Q$ matrices, respectively, where $Q$ is the number of time intervals. Recall that the size of the original intermediate data $\Xi_T$ is proportional to $B \times T$, we can see that the summarized features $\xi_{\mathrm{s}}$ and $\xi_{\mathrm{a}}$ are substantially smaller since $|\mathscr{S}|, |\mathscr{A}| \ll B$ in most practical cases.

## 3 SCENARIO EVALUATION OF AGENT-BASED TRAFFIC SIMULATIONS

We now describe an example of how to apply the proposed framework for an agent-based traffic simulation.

First, we present a design of features aiming at predicting the KPIs that are the length of congested roads. The exact criteria of the congested roads will be given in the next section. In short, the features consist of a vector representing actions to regulate a set of roads (the inputs of the simulations), the intermediate KPIs (the length of congested roads during the simulation), and temporal statistics on the number of cars at each road (the states of the agents in the simulation). Then, we show how to use the features in a prediction function for evaluating future simulation scenarios. Because the features are also available at all scales of traffic flow modeling, from picoscopic to macroscopic as mentioned in Ni (2010), the proposed framework may also be applicable to non agent-based traffic simulations.

### 3.1 Prediction Features

We assume an agent-based traffic simulation for evaluating the KPIs of traffic-regulation scenarios at, say, $T = 60$min (one hour) after executing the regulations. For simplicity, we assume the KPIs are the length of congested roads at different parts of the city (thus, multi-objective).

#### 3.1.1 Input Features

These features are the most basic and describe how we regulate the flows on some roads. For simplicity, let us suppose the traffic-regulation scenarios are in the form of directing the traffic on a road by one of 4 actions: *no intervention*, *one-way*, *reverse one-way*, and *close*. We note that it is straightforward to consider other actions. Let also assume that we have choices to impose those actions on $r$ number of roads.

We use the *1-of-K coding* (Bishop 2006) for the input feature representation. Specifically, we let $\theta = (\theta_1^\top, \theta_2^\top, \ldots, \theta_r^\top)^\top \in \{0,1\}^{4r}$ where $\theta_i \in \{0,1\}^4 \, (i = 1,2,\ldots,r)$. The vector $\theta_i$ has $\theta_{i1} = 1$ if the regulation on the road $i$ is *no intervention* and 0 otherwise. Similarly, we set $\theta_{i2} = 1$ if the regulation is *one-way*, $\theta_{i3} = 1$ if the regulation is *reverse one-way*, and $\theta_{i4} = 1$ if the regulation is *close*, and 0 otherwise. As the result, the input feature $\theta$ is a $4r$-dimensional binary vector with exactly $r$ number of 1s.

### 3.1.2 Intermediate Features

These features are unique to our approach. They consist of two types: intermediate KPIs, which are the length of congestion at intermediate time, and agents' features.
**[Intermediate KPIs]** During a simulation, we compute congestion lengths at some time points. Specifically, we measure them at $T = 10\text{min}, 20\text{min}$, and $30\text{min}$, which we denote by the vectors $\xi_{10\text{min}}$, $\xi_{20\text{min}}$, and $\xi_{30\text{min}}$, respectively. We aims at predicting the KPIs at 60min during the simulation. We therefore construct the intermediate feature vector $\xi_{\text{KPI}}$ as follows. When we predict KPIs at $T = 10\text{min}$ of the simulation time point, we set $\xi_{\text{KPI}} = \xi_{10\text{min}}$, while we set $\xi_{\text{KPI}} = (\xi_{10\text{min}}^\top, \xi_{20\text{min}}^\top)^\top$ and $\xi_{\text{KPI}} = (\xi_{10\text{min}}^\top, \xi_{20\text{min}}^\top, \xi_{30\text{min}}^\top)^\top$ when predicting at $T = 20\text{min}$ and 30min, respectively.
**[Intermediate Agents' States]** The agent-based traffic simulator provides the locations and the choices of roads of its agents (cars). We summarize the massive information of agents' locations and choices of roads by keeping track the following statistics at a fixed time interval for each road in the simulation: *the number of agents going into, the number of agents going out from, the number of agents staying at* (see Section 2.3 for the general framework). The intuition of choosing these statistics is because the congestion length on a road is heavily influenced by the number of cars staying on the road, while the other numbers provide clues on how the congestion will develop in the future. To this end, we present the matrix notations of those statistics. We denote the number of cars going into the road $i$ at the $q$th time interval by $\xi_{\text{In},i,q}$. Hence, $\xi_{\text{In}}$ is a $R \times Q$ matrix where $R$ is the number of roads and $Q$ is the number of time intervals. The matrix $\xi_{\text{In}}$ now stores averaged states of agents and its size is independent of the number of agents in the simulation. Moreover, because road network is sparse, the matrix is also sparse and can be stored efficiently. Similarly, the number of cars going out from can be summarized in $\xi_{\text{Out}} \in \mathbb{R}^{R \times Q}$, and the number of cars kept staying in $\xi_{\text{Stay}} \in \mathbb{R}^{R \times Q}$, which are also sparse.

## 3.2 Prediction Function

We now show the KPI prediction functions for future simulations. Let $\{y^{(n)}, \theta^{(n)}, \xi_{\text{KPI}}^{(n)}, \xi_{\text{In}}^{(n)}, \xi_{\text{Out}}^{(n)}, \xi_{\text{Stay}}^{(n)}\}_{n=1}^N$ be the KPIs, the inputs, and intermediate features from past $N$ simulations. For ease of notation, we use $x_1 = \theta$, $x_2 = \xi_{\text{KPI}}$, $x_3 = \xi_{\text{In}}$, $x_4 = \xi_{\text{Out}}$, $x_5 = \xi_{\text{Stay}}$, and $x = \{x_1, x_2, x_3, x_4, x_5\}$. We construct a prediction function $\hat{g}$ such that $y \approx \hat{g}(x)$. To this end, we use kernel ridge regression, or ridge regression with RBF kernels (Bishop 2006), for prediction, which is a variant of Kriging and we can utilize multimodal features.

Suppose we aim to predict the $p$th KPI $y_p$ with a function $\hat{g}_p$ as $y_p \approx \hat{g}_p(x)$. In the kernel ridge regression, we model the function $\hat{g}_p$ as the linear combination of the kernel functions. In the current modelling, we adopt a Gaussian kernel $k(w, w') = \exp\left(-\|w - w'\|^2 / 2\sigma^2\right)$ for each of $x_1, x_2, \ldots, x_5$. The overall model of $\hat{g}_p$ is now given by

$$\hat{g}_p(x) = \sum_{\ell=1}^{L} \sum_{j=1}^{J} \alpha_{\ell j} k(x_j, u_j^{(\ell)}) + \beta, \tag{1}$$

where $\alpha_{\ell j}$, $\beta$ are some real values, $L$ is the number of kernel functions, $J$ is the number of feature types utilized, and $u_j^{(\ell)}$ is the $\ell$th kernel center for $x_j$. Typical choice of the kernel center is a training data and we set $u_j^{(\ell)} = x_j^{(\ell)}$ (thus, $L = N$). Equation (1) can also be written as $\hat{g}_p(x) = \text{Tr}[A^\top K(x)] + \beta$ where $A_{\ell j} = \alpha_{\ell j}$ and $K(x)_{\ell j} = k(x_j, u_j^{(\ell)})$. The construction of the function $\hat{g}_p$ has now reduced to finding $A$ and $\beta$ so that

$y_p \approx \hat{g}_p(x)$. To this end, we optimize the following penalized least-squares problem, which is also known as ridge regression:

$$\min_{A,\beta} \frac{1}{2N} \sum_{n=1}^{N} (y_p^{(n)} - \hat{g}_p(x^{(n)}))^2 + \frac{\rho}{2} \|A\|_{\mathrm{F}}^2,$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix and $\rho$ is a non-negative parameter. Here, let $Y_p = (y_p^{(1)}, y_p^{(2)}, \ldots, y_p^{(N)})^\top$ and $G = (\mathrm{vec}(K(x^{(1)})), \mathrm{vec}(K(x^{(2)})), \ldots, \mathrm{vec}(K(x^{(N)})))^\top$ where $\mathrm{vec}(\cdot)$ is a vecotrization operator. The optimal $A$ and $\beta$ can then be derived analytically as follows:

$$\begin{aligned} \mathrm{vec}(A) &= (G^\top H_N G + N\rho I_{LJ})^{-1} G^\top H_N Y_p = G^\top H_N (H_N G G^\top H_N + N\rho I_N)^{-1} Y_p, \\ \beta &= 1_N^\top (Y_p - G\,\mathrm{vec}(A))/N, \end{aligned}$$

where $1_N$ is an $N$-dimensional vectors of all ones, $I_N$ and $I_{LJ}$ are $N \times N$ and $LJ \times LJ$ identity matrices, and $H_N = I_N - 1_N 1_N^\top / N$. The last equality on $A$ follows from the Woodbury matrix identity (Petersen and Pedersen 2012). We note that the regularization parameter $\rho$ and the width parameter $\sigma$ of the Gaussian kernel remain as user-tuning parameters. We can decide these values using model selection methods such as cross validation.

## 4 EXPERIMENT

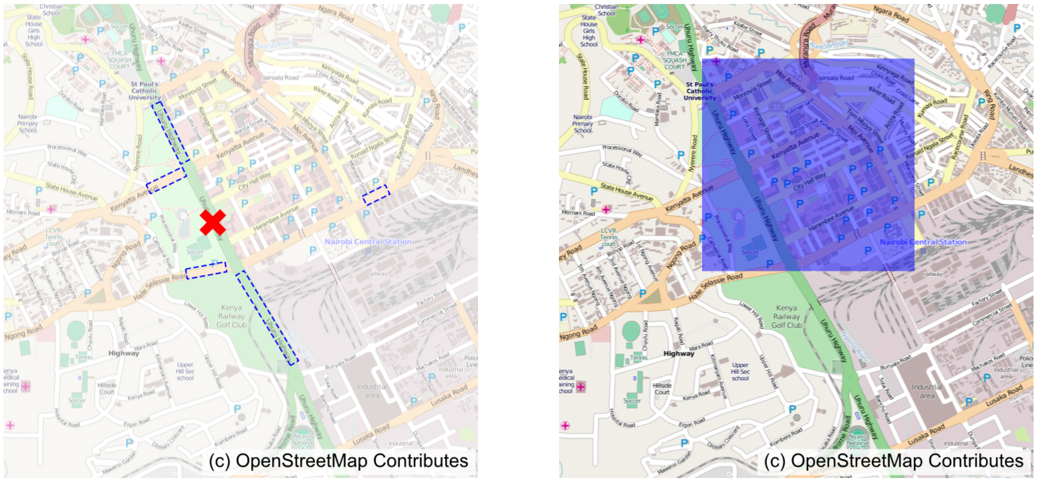
This section describes experiment results on the efficacy of the proposed method on predicting the KPIs of simulation scenarios, and on finding scenarios with the best KPIs. We used the agent-based traffic simulator Megaffic (Osogami et al. 2012) for simulating traffics in Nairobi, Kenya, with similar settings as Imamichi and Raymond (2013). The traffic simulator runs on the road network data provided by the OpenStreetMap (http://www.openstreetmap.org), and on the origin-destination table that provides trips of cars designed based on Gonzales, Chavis, Li, and Daganzo (2009). We used $R = 3857$, a smaller number of roads than Imamichi and Raymond (2013) because other roads were not traversed by generated trips.

### 4.1 Experiment Setting

We consider a case when traffic authorities evaluate traffic-regulation scenarios to resolve traffic congestion caused by a traffic accident on one of arterial roads, as marked by the red ”x” in the left of Figure 2, by directing traffic on the surrounding five roads, as marked by the blue dashed-boxes in the left of Figure 2. There are 4 possible actions for each road (see Section 3.1.1) and therefore the number of possible simulation scenarios is $4^5 = 1,024$. The KPIs of the scenarios are the congestion lengths at one hour after implementing the actions measured in both inside and outside the city central business district, called *Inside C.B.D* and *Outside C.B.D* as in the right of Figure 2, respectively. We precomputed the KPIs by fully simulating all scenarios to obtain the ground truth.

We consider two different definitions of congested roads. In the first definition, a road is congested if the average speed of cars on the road is less than 5km/h threshold, while in the second definition, the threshold is 10km/h. Thus, the corresponding KPIs $y \in \mathbb{R}^4$ are the congestion lengths in two parts of the city under two different congestion definitions which we denote by *Inside C.B.D (5km/h)*, *Inside C.B.D (10km/h)*, *Outside C.B.D (5km/h)*, and *Outside C.B.D (10km/h)*.

We conducted $N$ full simulations, with $N$ varied from 60 to 300, and derived their final KPIs and prediction features. We then built prediction functions using the proposed method. As $N$ gets large, we obtain much more intermediate features for the prediction model $\hat{g}$. Here, we refer to the model using prediction function in Equation (1) with $J = 5$ as the *full PHTS model* (after **P**rediction **H**alfway **T**hrough **S**imulation). As the baseline method, we adopt Kriging (Beers and Kleijnen 2004, Kleijnen 2009) which uses only the inputs for prediction. We can then verify how intermediate features are useful by contrasting the two models. In the experiment, we also consider the variant of the full PHTS model which uses only




(a) Accident site(cross) and five traffic regulation candidate roads(dashed line)

(b) Shaded region is Inside C.B.D. Outside C.B.D is an unshaded region.

Figure 2: Nairobi City Map

the inputs and the intermediate KPIs (that is, $J = 2$ in (1)). We refer to the model as the *simple PHTS model*. The intention behind this model is to show that the KPIs cannot be simply inferred from the same prediction functions as the full PHTS model using the intermediate KPIs alone.

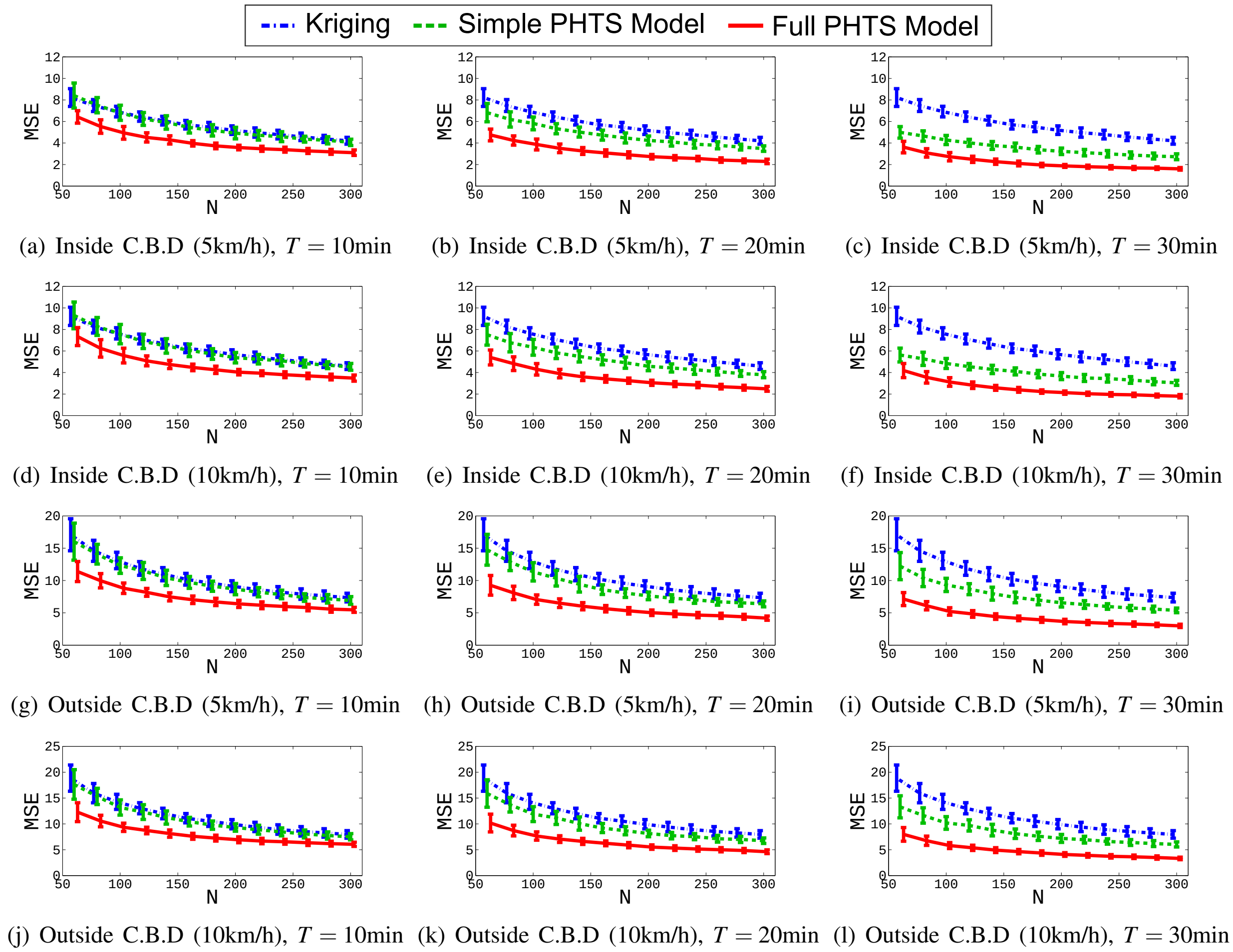
### 4.2 Result - Predicting the KPIs

We compared the accuracies of the models to predict the KPIs of fully-simulated scenarios. We randomly selected $N$ fully-simulated scenarios out of 1,024, and $M = 1024 - N$ halfway-simulated scenarios, to obtain prediction features for the two PHTS models. The comparison were repeated 50 times.

The results can be found in Figure 3 where each graph corresponds to the prediction performance of each model at different midpoints of the simulations $T = 10\text{min}, 20\text{min}$, and 30min, respectively. The horizontal axis of each graph denotes the number of fully-simulated scenarios $N$ and the vertical axis denotes the mean square error(MSE) of predicted KPIs on the remaining $M = 1024 - N$ scenarios defined as $\text{MSE} = \sum_{n'=N+1}^{1024} (y_p^{(n')} - \hat{g}_p(x^{(n')}))^2 / M$. Each line in the graph corresponds to the median MSE of 50 repetitions for each method. The blue dot-dashed and the green dashed lines, denote, resp., the MSEs of Kriging and the simple PHTS models. The red solid line denotes those of the full PHTS model.

We can see the advantage of the full PHTS model over the others: it attains the smallest MSEs for predicting all types of KPIs on any range of $N$ fully-simulated scenarios and the simulation midpoint of the rest $M$ scenarios. Moreover, we can also observe that as more intermediate features were available (as $T$ grows from 10min to 30min), the accuracies of the full PHTS model significantly became better relative to the other models (as the MSE gaps in the figure became larger). In particular at $T = 30$min, the MSE of the full PHTS model at $N = 60$ is comparable to the ones of Kriging at $N = 300$. This indicates that the proposed framework, predicting halfway through simulation, can gain much more information from intermediate features. Hence, the difference of the two PHTS models revealed the importance of what intermediate information to use: the simple PHTS model merely uses the inputs and the intermediate KPIs and completely discards micro-level information of agent-based simulations, which resulted in the inferior performance than the full PHTS model.

We now show the detailed differences of the prediction accuracies of Kriging and the full PHTS model at $N = 60$ in Figure 4. In the figure, the horizontal axis denotes the ground-truth KPIs and the vertical axis denotes the predicted KPIs. The scatter plots of the full PHTS model are clearly closer to the diagonal lines than those of Kriging, that also confirm the better reliability of the full PHTS model.




(a) Inside C.B.D (5km/h), $T = 10$min (b) Inside C.B.D (5km/h), $T = 20$min (c) Inside C.B.D (5km/h), $T = 30$min

(d) Inside C.B.D (10km/h), $T = 10$min (e) Inside C.B.D (10km/h), $T = 20$min (f) Inside C.B.D (10km/h), $T = 30$min

(g) Outside C.B.D (5km/h), $T = 10$min (h) Outside C.B.D (5km/h), $T = 20$min (i) Outside C.B.D (5km/h), $T = 30$min

(j) Outside C.B.D (10km/h), $T = 10$min (k) Outside C.B.D (10km/h), $T = 20$min (l) Outside C.B.D (10km/h), $T = 30$min

Figure 3: Median Prediction Errors: The horizontal axis denotes the number of full simulations $N$ and the vertical axis denotes the mean square error(MSE). The vertical bars extend from 25% to 75% quantiles.

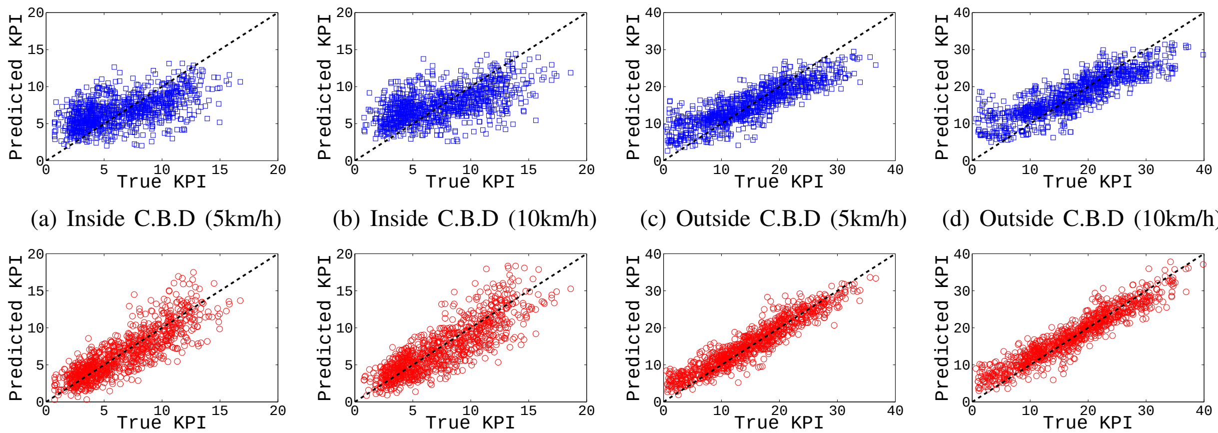
### 4.3 Result - Finding the Top-100 Best Scenarios

We compared the performance of the models in identifying the top-100 best scenarios, those with small congestion lengths, on different parts of the city. The traffic authorities can then, for example, put more priorities to explore the identified scenarios with full simulation in conjunction with the ranking and selection (R&S) approaches (Kim and Nelson 2007) that require simulating all scenarios.

We combined the $N$ KPIs of $\{y_p^{(n)}\}_{n=1}^N$ from full simulations and the $M$ predicted KPIs $\{\hat{g}_p(x^{(n')})\}_{n'=N+1}^{1024}$ for the identification. We let $z^{(n)} = y_p^{(n)}$ for $n = 1, 2, \ldots, N$ and $z^{(n')} = \hat{g}_p(x^{(n')})$ for $n' = N+1, N+2, \ldots, 1024$. We then sorted the values $\{z^{(n)}\}_{n=1}^{1024}$ in an ascending order and extracted the predicted top-100 scenarios. The accuracy is measured by the number of true top-100 scenarios identified by the models from the above extraction procedure. Figure 5 shows the accuracy comparison of the models. We again can observe that the full PHTS model attains the best result for all cases of $N$ and the simulation midpoints.

## 5 RELATED WORK

There are many established work in prediction and finding optimal scenarios by uncovering the inputs-KPIs relations (a.k.a. metamodeling or response surfaces) in simulations. For example, see a brief overview of Design of Experiments by Kleijnen (2008). Popular methods include regression (Kleijnen 2007), sequential




(a) Inside C.B.D (5km/h) (b) Inside C.B.D (10km/h) (c) Outside C.B.D (5km/h) (d) Outside C.B.D (10km/h)

(e) Inside C.B.D (5km/h) (f) Inside C.B.D (10km/h) (g) Outside C.B.D (5km/h) (h) Outside C.B.D (10km/h)

Figure 4: True KPI values and Predicted KPI values when $N = 60$: Upper figures (a)–(d) denote results from Kriging while the lower figures (e)–(h) represent results from the full PHTS model at $T = 30$min. The diagonal dashed line denotes the ideal result where the true values and the predicted values coincide.

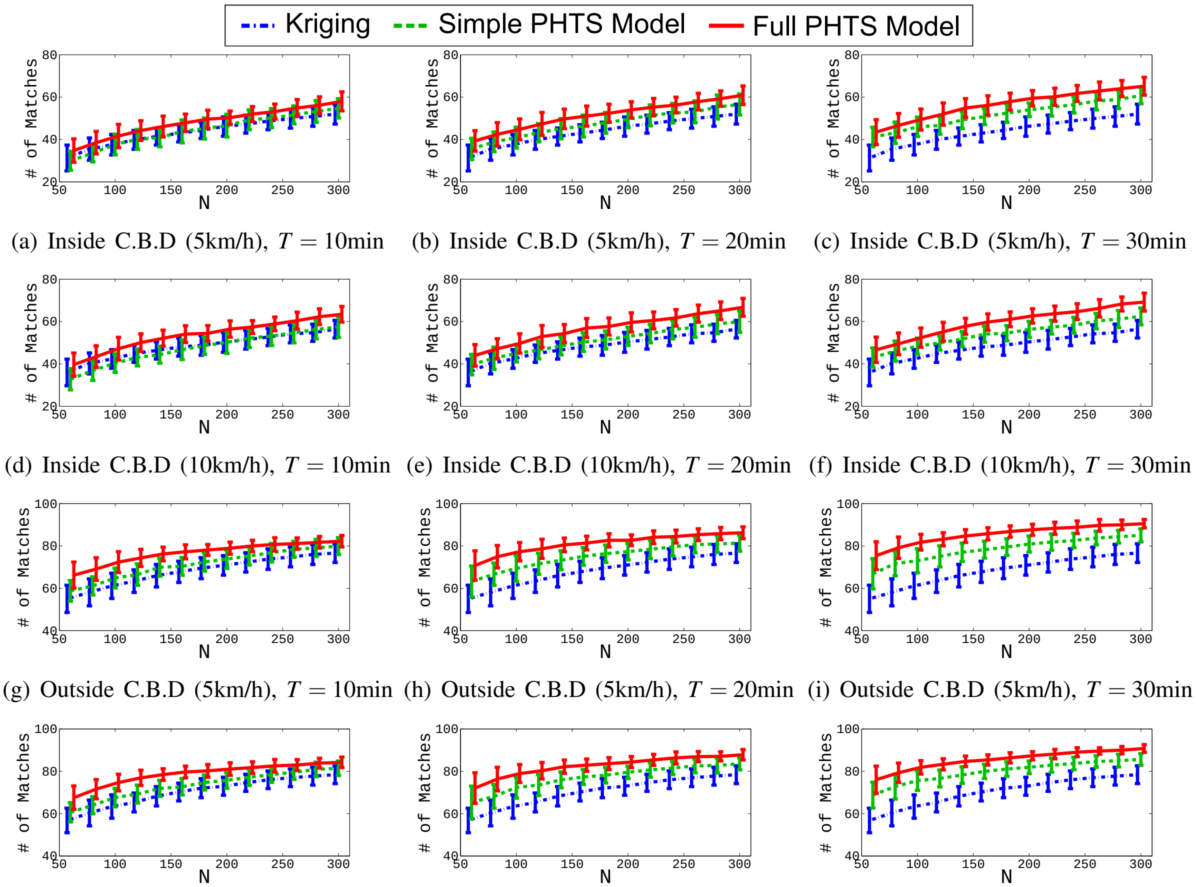
bifurcation (Bettonvil 1990), Kriging (Beers and Kleijnen 2004, Kleijnen 2009, Staum 2009, Ankenman, Nelson, and Staum 2010), Bayesian optimization (Snoek, Larochelle, and Adams 2012), and approximate Bayesian computation (Csilléry, Blum, Gaggiotti, and François 2010), that are mainly developed for sensitivity analysis and optimization. Recently, there are significant progresses on metamodeling for high-dimensional designs, such as, in Shan and Wang (2010), and Salemi, Nelson, and Staum (2012).

In contrast to the methods proposed in this paper, however, the previous methods treat simulations as black boxes, i.e., only use the I/O of simulations. Some of their advantages is that they are generic and independent of simulation types, and therefore applicable to a wide variety of simulations. On the other hand, the generic property sacrifices some accuracies due to ignoring intermediate data that might capture the trends emerging during the simulations. Our approach is ad-hoc but tailored to typical intermediate data of an agent-based traffic simulations. Recently, there have been approaches to construct dynamic bayesian networks (Dean and Kanazawa 1989) representing changes of intermediate states of simulations that can be used to infer the KPIs of simulations (Poropudas and Virtanen 2011, Poropudas, Pousi, and Virtanen 2011, Pousi, Poropudas, and Virtanen 2013). The drawbacks of such approaches are that, e.g., they require experts knowledge and many repetitions of simulations to determine the structure of the bayesian network.

## 6 CONCLUSION

We presented a novel method to use the intermediate features of agent-based traffic simulations to evaluate scenarios that are simulated only until their midpoints. In particular, we showed how to aggregate the agents' states obtained during the simulations and use them as parameters of a state-of-the-art prediction method. We confirmed the superiority of the proposed method over standard approaches that regard the simulations as black boxes in predicting the KPIs of the scenarios and in identifying the best scenarios.

There are several directions of interesting future work, such as how to take into account more intermediate features to improve the accuracies of the proposed method. It will require advanced techniques in dimension reduction and big data acquisition, and carefully balancing the accuracy and computational resources between the prediction and simulation. Another direction is development of the decision making algorithm (Csilléry, Blum, Gaggiotti, and François 2010, Snoek, Larochelle, and Adams 2012) that incorporates the proposed framework. For this purpose, we need to decide when to halt the current simulation based on the predicted KPIs and which scenarios to run to completion to access their exact KPIs. For example, in our traffic simulation study, the prediction at 30min time point seemed to be sufficiently good, although this may




(a) Inside C.B.D (5km/h), $T = 10$min (b) Inside C.B.D (5km/h), $T = 20$min (c) Inside C.B.D (5km/h), $T = 30$min

(d) Inside C.B.D (10km/h), $T = 10$min (e) Inside C.B.D (10km/h), $T = 20$min (f) Inside C.B.D (10km/h), $T = 30$min

(g) Outside C.B.D (5km/h), $T = 10$min (h) Outside C.B.D (5km/h), $T = 20$min (i) Outside C.B.D (5km/h), $T = 30$min

(j) Outside C.B.D (10km/h), $T = 10$min (k) Outside C.B.D (10km/h), $T = 20$min (l) Outside C.B.D (10km/h), $T = 30$min

Figure 5: Median Top-100 Identification: The horizontal axis denotes the number of full simulations $N$ and the vertical axis denotes the number of correct identification. The vertical bars extend from 25% to 75% quantiles.

not be the case if the target system is chaotic because it may require nearly complete simulations due to its long-term unpredictability. We think the exploration-exploitation trade-off in reinforcement learning, etc. (Sutton and Barto 1998, Auer, Cesa-Bianchi, and Fischer 2002, Osogami and Kato 2007) might be useful for this direction.

## ACKNOWLEDGMENTS

SH, RR, and TM were supported by JST, CREST. We thank Dr. Takashi Imamichi for his comments.

## REFERENCES

Ankenman, B. E., B. L. Nelson, and J. Staum. 2010. “Stochastic Kriging for Simulation Metamodeling”. *Operations Research* 58 (2): 371–382.

Auer, P., N. Cesa-Bianchi, and P. Fischer. 2002. “Finite-time Analysis of the Multiarmed Bandit Problem”. *Machine Learning* 47:235–256.

Beers, W. C. M. V., and J. P. C. Kleijnen. 2004. “Kriging Interpolation in Simulation: A Survey”. In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S.

Smith, and B. A. Peters, 113–121: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers (IEEE), Inc.

Behrisch, M., L. Bieker, J. Erdmann, and D. Krajzewicz. 2011. "SUMO - Simulation of Urban Mobility: An Overview". In *Proceedings of International Conference on Advances in System Simulation*, 55–66.

Bettonvil, B. 1990. *Detection of Important Factors by Sequential Bifurcation*. Tilburg University Press.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Box, G. E., and N. R. Draper. 2006. *A Model for Reasoning about Persistence and Causation*. John Wiley& Sons, Inc.

Csilléry, K., M. G. Blum, O. E. Gaggiotti, and O. François. 2010. "Approximate Bayesian Computation (ABC) in Practice". *Trends in Ecology & Evolution* 25 (7): 410–418.

Dean, T., and K. Kanazawa. 1989. "A Model for Reasoning about Persistence and Causation". *Computational Intelligence* 5 (2): 142–150.

Gomes, G., A. May, and R. Horowitz. 2004. "Congested Freeway Microsimulation Model Using VISSIM". *Transportation Research Record: Journal of the Transportation Research Board* (1876): 71–81.

Gonzales, E. J., C. Chavis, Y. Li, and C. F. Daganzo. 2009. "Multimodal Transport Modelling for Nairobi, Kenya: Insights and Recommendations with an Evidence-Based Model". Technical Report UCB-ITS-VWP-2009-5, UC Berkeley Center for Future Urban Transport, A Volvo Center of Excellence.

Imamichi, T., and R. Raymond. 2013. "Simple and Fast Trip Generation for Large Scale Traffic Simulation". In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 2020–2031: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers (IEEE), Inc.

Kim, S.-H., and B. L. Nelson. 2007. "Recent Advances in Ranking and Selection". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 162–172: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers (IEEE), Inc.

Kleijnen, J. P. C. 2007. "Regression models and Experimental Designs: A Tutorial for Simulation Analysts". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 183–194: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers (IEEE), Inc.

Kleijnen, J. P. C. 2008. "Design Of Experiments: Overview". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 479–488: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers (IEEE), Inc.

Kleijnen, J. P. C. 2009. "Kriging Metamodeling in Simulation: A Review". *European Journal of Operational Research* 192 (3): 707–716.

Libai, B., E. Muller, and R. Peres. 2013. "Decomposing the Value of Word-of-Mouth Seeding Programs: Acceleration Versus Expansion". *Journal of Marketing Research* 50 (2): 161–176.

Ni, D. 2010. "A Spectrum of Traffic Flow Modeling at Multiple Scales". In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 554–566: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers (IEEE), Inc.

Osogami, T., T. Imamichi, H. Mizuta, T. Morimura, R. Raymond, T. Suzumura, R. Takahashi, and T. Idé. 2012. "IBM Mega Traffic Simulator". Technical Report RT0896, IBM Research – Tokyo.

Osogami, T., and S. Kato. 2007. "Optimizing System Configurations Quickly by Guessing at the Performance". In *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 145–156.

Petersen, K. B., and M. S. Pedersen. 2012. "The Matrix Cookbook". *Transportation Research Record: Journal of the Transportation Research Board*. Version 20121115.

Poropudas, J., J. Pousi, and K. Virtanen. 2011. "Multiple Input and Multiple Output Simulation Metamodeling Using Bayesian Networks". In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain,

R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 569–580: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers (IEEE), Inc.

Poropudas, J., and K. Virtanen. 2011. “Simulation Metamodeling with Dynamic Bayesian Networks”. *European Journal of Operational Research* 214 (43): 644–655.

Pousi, J., J. Poropudas, and K. Virtanen. 2013. “Simulation Metamodelling with Bayesian Networks”. *Journal of Simulation* 7 (4): 297–311.

Rand, W., and R. T. Rust. 2011. “Agent-based Modeling in Marketing: Guidelines for Rigor”. *International Journal of Research in Marketing* 28 (3): 181–193.

Rasmussen, C. E., and C. K. I. Williams. 2005. *Gaussian Processes for Machine Learning*. The MIT Press.

Salemi, P., B. L. Nelson, and J. Staum. 2012. “Moving Least Squares Regression for High Dimensional Simulation Metamodeling”. In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 1–12: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers (IEEE), Inc.

Shan, S., and G. G. Wang. 2010. “Metamodeling for High Dimensional Simulation-based Design Problems”. *Journal of Mechanical Design* 132 (5): 051009.

Snoek, J., H. Larochelle, and R. P. Adams. 2012. “Practical Bayesian Optimization of Machine Learning Algorithms”. In *Proceedings of Advances in Neural Information Processing Systems*, 2960–2968.

Staum, J. 2009. “Better Simulation Metamodeling: The Why, What, and How of Stochastic Kriging”. In *Proceedings of the 2009 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 119–133: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers (IEEE), Inc.

Sutton, R. S., and A. G. Barto. 1998. *Reinforcement Learning*. The MIT Press.

Taylor, S. J. E., S. E. Chick, C. M. Macal, S. C. Brailsford, P. L’Ecuyer, and B. L. Nelson. 2013. “Modeling and Simulation Grand Challenges: An OR/MS Perspective”. In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 1269–1282: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers (IEEE), Inc.

## AUTHOR BIOGRAPHIES

**SATOSHI HARA** is a researcher at IBM Research – Tokyo. He received his Ph.D. in Electrical Engineering from Osaka University in 2013, and a M.E. degree in Computer Science from Tokyo Institute of Technology in 2010, and a B.E. degree in Applied Physics from Waseda University in 2008. His research interests are machine learning algorithms and their applications in industries. His email address is satohara@jp.ibm.com.

**RUDY RAYMOND** is a researcher at IBM Research – Tokyo. He studied in Kyoto University and obtained the Bachelor of Engineering, Master, and Doctor of Informatics degrees in 2001, 2003, and 2006, respectively. In IBM Research, he has been working with various clients in telecommunication and manufacturing industries. His email address is raymond@jp.ibm.com.

**TETSURO MORIMURA** is a researcher at IBM Research – Tokyo. He received the M.E. and Ph.D. degrees in engineering from Nara Institute of Science and Technology, Japan. His research interests include reinforcement learning, machine learning, and its applications to neuroscience and business intelligence. His email address is tetsuro@jp.ibm.com.

**HIDEMASA MUTA** is a researcher at IBM Research – Tokyo. He received his Ph.D. degrees in engineering from The University of Electro-Communications, Tokyo, Japan. He has worked on various projects in IBM for 23 years. His research interests include multi-objective optimization, agent-based traffic simulation, high performance computing, and computer vision. His email address is hmuta@jp.ibm.com.