Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

COORDINATED TRAFFIC SIGNAL CONTROL VIA BAYESIAN OPTIMIZATION FOR HIERARCHICAL CONDITIONAL SPACES

Hidetaka Ito
Kyota Tsutsumida
Tatsushi Matsubayashi
Takeshi Kurashima
Hiroyuki Toda

NTT Service Evolution Laboratories
NTT Corporation
1-1 Hikari-no-oka, Yokosuka
Kanagawa, 239-0847, JAPAN

ABSTRACT

We study the problem of coordination control of multiple traffic signals to mitigate traffic congestion. The parameters we optimize are the coordination pattern and offsets. A coordination pattern indicates which traffic signals are coordinated. Offsets show how traffic signals can be coordinated. We aim at finding the optimal combination of the coordination pattern and offsets. In this paper, we treat it as an optimization problem whose search space is a conditional space with hierarchical relationships; the coordination pattern determines the controllability of the offsets. Then, we tackle the problem by proposing a novel method built upon Bayesian Optimization, called BACH. Experiments demonstrate that BACH successfully optimizes coordination control of traffic signals and BACH outperforms various state-of-the-art approaches in the literature of traffic signal control and Bayesian Optimization in terms of best parameters found by these methods with a fixed budget.

1 INTRODUCTION

Metropolitan areas are characterized by the sheer number of traffic signals, and traffic congestion can occur anywhere. Achieving the optimal control of those signals will greatly benefit society (Koonce et al. 2008). Traffic signal control has been studied for many years (Papageorgiou et al. 2003) and simulation-based approaches are popular (Osorio and Chong 2012). Traffic signals, vehicles, roads, obstacles, and their interaction are simulated using digital models (Bowman and Miller 2016). Our aim is the global optimization of traffic; we minimize the metric of *time loss*. Time loss quantifies traffic congestion, and is the sum of the time each vehicle lost by being driven slower than the desired speed. We consider all vehicles on the various routes taken. In this paper, we focus on standard fixed-time control; the signal phase plan and phase times are fixed. Most existing signals are fixed-time.

Coordination control of traffic signals is especially important to mitigate traffic congestion as shown in Figure 1. For example, consider two traffic signals on the same road with vehicles passing through them. Vehicles pass the first traffic signal, then they reach the front of the second one. If the second traffic signal is green when they arrive at the second one, they can pass through without stopping. As reducing the number of times vehicles must stop relieves traffic congestion, coordination control is very desirable. However, some traffic signals should not be coordinated. The cycle length is a parameter traffic signals are characterized by. This indicates the time for an entire sequence of signal phases at each intersection, where a signal phase is a timing unit associated with the control system. To establish the coordinated control of two traffic signals, the cycle lengths of the two traffic signals must be the same. As each traffic signal has

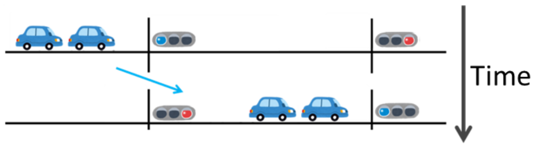

Figure 1: Illustration of coordination control. Vehicles can cross intersections without stopping by setting appropriate offsets.

a different optimal cycle length, this requirement leads to an increase in time loss. Therefore, we need to carefully determine which traffic signals should be coordinated and how to coordinate them.

Two types of parameters for this problem are involved in the problem as shown in the left part of Figure 2: *the coordination pattern* and *offsets*. These parameters are associated not with individual intersections, but pairs of intersections linked by a connected road. A coordination pattern represents a set of decisions as which pairs of intersections should be coordinated. In Figure 2, we show two examples of coordination patterns. Additionally, each offset represents a decision of how to coordinate a pair of intersections. Offsets are defined by the time difference between the beginning of the first signal phases of the intersection pairs. What makes the problem difficult is the hierarchical structures present in the coordination pattern and offsets. We can control only those offsets associated with coordinated intersection pairs. Controllable offsets are shown in dark gold lines in the left part of Figure 2. The number of controllable offsets depends on the coordinated pairs found in the coordination pattern.

We model the time loss as a function of the coordination pattern and offsets. Then, the problem is to find the optimal combination of the coordination pattern and offsets. The direct approach is to use trial and error via computer simulations to find the optimum combination. Unfortunately, the cost of realistic simulations usually restricts the number of simulation runs. Recently, there has been a surge of interest in Bayesian optimization (BO). BO is a gradient-free method for the global optimization of black-box functions with as few evaluations as possible (Shahriari et al. 2016). An evaluation of black-box functions can be thought as a simulation run. BO iterates the process of parameter selection and simulation run. BO selects parameters which seem promising by using surrogate models that predict the simulation result without making a direct simulation run. BO has been gaining attention from the simulation optimization community (Pearce and Branke 2017; Poloczek et al. 2016). Figure 2 illustrates application of BO to traffic signal control. Unfortunately, ordinary BO suffers from the curse of dimensionality. Ordinary BO considers all combinations of the coordination pattern and offsets in a naive manner. The large number of parameter combinations makes it very difficult to apply ordinary BO effectively.

Therefore, we propose a novel method called *BAyesian optimization for Conditional spaces with Hierarchical relationships (BACH)*. BACH is based on a simple assumption: optimal offsets for a given coordination pattern can be taken as suggestions for other similar coordination patterns (see Figure 3). Similar coordination patterns can be expected to exhibit similar movements of vehicles and relations among intersections. Thus, similar environments lead to similar optimal control schemes. In this case, we can transfer the knowledge gained from the offsets optimized for a given coordination pattern to solve similar patterns. Based on this assumption, we find combinations of a coordination pattern and offsets by the following strategy. First, we restrict the targeted coordination patterns and search for optimal offsets for the targeted coordination patterns. Once we find promising offsets for them, we can apply knowledge transfer to optimize the offsets of other coordination patterns. This allows us to efficiently search for the optimal combination of the coordination pattern and offsets.

In an experiment on traffic signal control, we show that BACH can find promising combinations of the coordination pattern and offsets. To the best of our knowledge, this paper is the first to apply a framework based on BO to traffic signal control. Furthermore, BACH outperforms state-of-the-art baselines in the literature of traffic signal control and Bayesian Optimization in terms of the best parameters found by these methods with fixed iteration numbers.

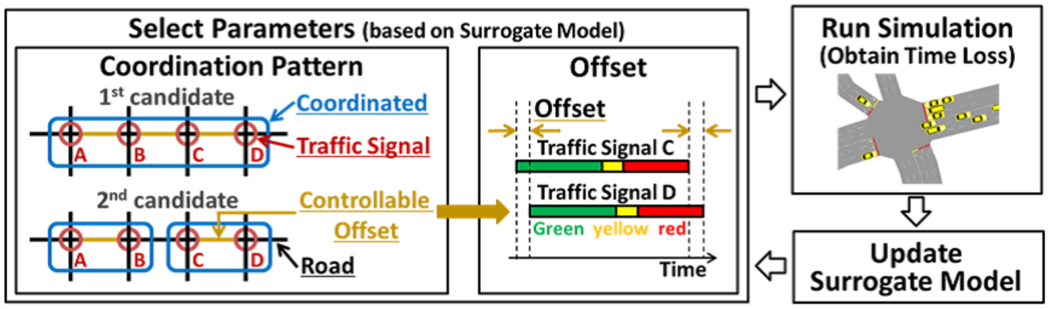

Figure 2: Illustration of traffic signal control via Bayesian Optimization (BO). The objective is to find the optimal parameters (a coordination pattern and offsets) that minimize the time loss. We consider pairs of intersections. A coordination pattern determines which pairs of intersections are coordinated. Two examples are represented by maps shown in the left part. For example, the 1st candidate is a coordination pattern indicating all pairs are coordinated. Each offset determines how a pair can be coordinated. Two types of parameters have hierarchical relationships. If the coordination pattern determines that the pair of intersections associated with the offset is coordinated, the offset is controllable; otherwise, it is uncontrollable. Controllable offsets are shown by the dark gold lines in the coordinate pattern examples. BO iterates the three processes shown in the figure.

Our contributions can be summarized in three points:

- Model formulation of the coordinated traffic signal control problem;
- Proposal of a method based on Bayesian Optimization to efficiently optimize two types of parameters with hierarchical relationships (the coordination pattern and offsets);
- Demonstration of the superiority of our method over state-of-the-art baselines.

2 RELATED WORKS

We introduce related works from two viewpoints.

Traffic Signal Control. Genetic Algorithm (GA) (Lertworawanich et al. 2011; Armas et al. 2017) and Particle Swarm Optimization (PSO) (Garcia-Nieto et al. 2013) are popular gradient-free global optimization methods used in simulation-based traffic signal parameter optimization. Osorio and Chong (2012) introduces a metamodel approach for simulation-based optimization. These methods as well as vanilla BO fail to well handle the hierarchical relationships of the coordination pattern and offsets. Previous works often omit the relationships by assuming that all traffic signals are coordinated (Lertworawanich et al. 2011) or by choosing coordination patterns beforehand using non-simulation-based methods (Wang et al. 2013). Our method can optimize both of them using just simulations. Furthermore, we empirically showed that BACH outperforms GA and PSO. This paper provides an advance in the practicality of simulation-based optimization of traffic signal control. Although reinforcement learning is a popular optimization method (Wei et al. 2018), it is applicable only to adaptive signals that change the signals dynamically according to observed traffic. Most existing signals are fixed-time, not adaptive, because adaptive ones are expensive. By contrast, BACH is applicable to standard fixed-time traffic signals, which indicates that BACH is practical.

Bayesian Optimization. The idea of using BO in conditional spaces with hierarchical relationships has already been explored in the literature. Independent BO uses independent GPs across different coordination patterns (Bergstra et al. 2011). As ordinary GPs cannot share information across different coordination patterns, Arc BO and SMAC enable BO to do so through the use of Arc-GPs (Swersky et al. 2014) and Random Forest (Hutter et al. 2011), respectively. BACH builds upon Arc BO. Even with the successes of BO in many applications, the curse of dimensionality is a major barrier to the application of BO to real world problems. Many approaches have been proposed to overcome the curse of dimensionality (Kandasamy

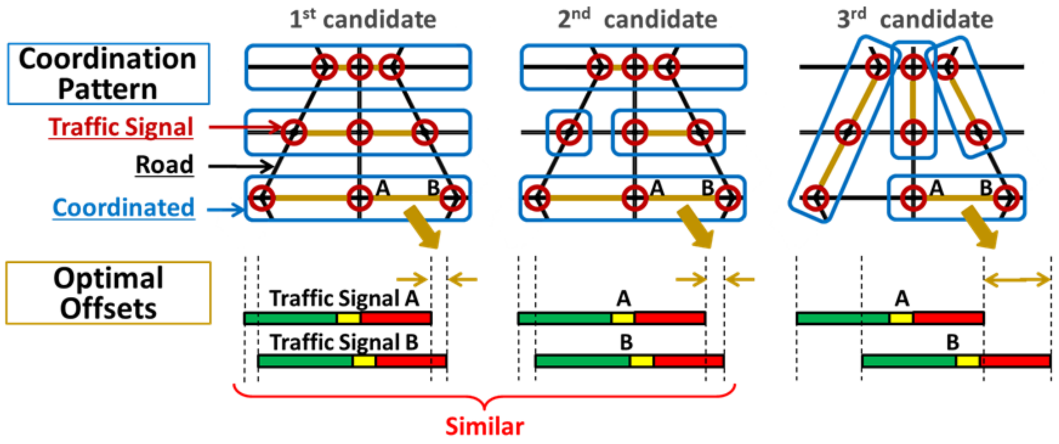

Figure 3: Illustration of the main assumption of the proposed method. Three coordination patterns and optimal offsets are shown. Each offset shown in the figure is associated with a pair of intersections located in the same position. The assumption states that when coordination patterns of the 1st and 2nd candidates are similar, their optimal offsets are also similar. The assumption originates from the idea that similar environments demonstrating similar coordination patterns exhibit similar optimal offsets.

et al. 2015). However, most are designed for spaces without hierarchical relationships. Few approaches tackle the curse of dimensionality for conditional spaces with hierarchical relationships. Tree BO has been developed toward the efficient optimization of parameters with tree-structured relationships (Jenatton et al. 2017). BACH outperformed Arc BO, Independent BO, Tree BO, and SMAC in all experiments conducted herein.

3 PROBLEM FORMULATION

In this paper, we construct the optimization problem of coordinated control of traffic signals as follows. Parameter x_i is the offset of the intersection pair i. Let $\boldsymbol{x} = [x_1, \dots, x_D]$ denote the concatenation of all offsets, where D is the number of intersection pairs. Note that we consider only intersection pairs directly connected by a road. In contrast to ordinary optimization problems that deal with only one kind of parameter, we tackle two parameters. Parameter z_i is a binary parameter that represents whether x_i is controllable or not. Let $\boldsymbol{z} = [z_1, \dots, z_D]$ denote the coordination pattern. When $z_i = 1$ ($z_i = 0$), x_i is controllable (uncontrollable). When $z_i = 0$, since we can not control x_i, the value of x_i is missing. Let $\mathscr{Z} \subseteq \{0,1\}^D$ denote the set of all possible coordination patterns, which is given as an input. Let $\mathscr{X}_{\boldsymbol{z}}$ denote the search space of $\boldsymbol{x}$ given coordination pattern $\boldsymbol{z}$. The spaces $\{\mathscr{X}_{\boldsymbol{z}} \mid z \in \mathscr{Z}\}$ are pairwise disjoint, i.e. $\mathscr{X}_{\boldsymbol{z}} \bigcap \mathscr{X}_{\boldsymbol{z}'} = \emptyset$, for all $\boldsymbol{z}, \boldsymbol{z}' \in \mathscr{Z}$, $\boldsymbol{z} \neq \boldsymbol{z}'$.

Our solution is to take a straightforward approach: we simultaneously optimize both offset $\boldsymbol{x}$ and coordination pattern $\boldsymbol{z}$. The whole search space is given by

$$\mathscr{X} = \bigcup_{\boldsymbol{z} \in \mathscr{Z}} \mathscr{X}_{\boldsymbol{z}}.$$

The entire search space is composed of subspaces with different input dimensions, which have to be separately explored. Note that, for all $\boldsymbol{x} \in \mathscr{X}$, there exists only one $\boldsymbol{z}$ that corresponds to $\boldsymbol{x}$. Let $\mathscr{F} : \mathscr{X} \to \mathbb{R}$ denote the time loss of the traffic simulation. The objective is to find the optimal combination of coordination pattern $\boldsymbol{z}^*$ and offsets $\boldsymbol{x}^* \in \mathscr{X}_{\boldsymbol{z}^*}$ that minimizes the time loss $\mathscr{F}(\boldsymbol{x})$. Note that, although $\mathscr{F}(\boldsymbol{x})$ can be represented as a function of just offsets $\boldsymbol{x}$, the coordination pattern $\boldsymbol{z}$ is implicitly related to $\mathscr{F}(\boldsymbol{x})$ as each $\boldsymbol{x}$ has a unique corresponding $\boldsymbol{z}$.

We note that $\mathscr{Z}$ is not always equal to $\{0,1\}^D$. Some coordination patterns represent the same groups of coordinated traffic signals. Therefore, we remove some coordination patterns beforehand so that a way of grouping coordinated traffic signals and a $\boldsymbol{z} \in \mathscr{Z}$ are in one-to-one correspondence. For example, we consider three intersections connected by roads, A, B, and C. All pairs are connected by a road. We should consider offsets of B from A, C from B, and A from C. $\boldsymbol{z}$ is a binary vector whose number of elements is three. In this case, as $\boldsymbol{z} = [1,1,1]$, $\boldsymbol{z} = [1,1,0]$, $\boldsymbol{z} = [1,0,1]$, and $\boldsymbol{z} = [0,1,1]$ represent the same groups of coordinated traffic signals. Therefore, we select one $\boldsymbol{z}$ included in $\mathscr{Z}$ from them before optimization is commenced as follows. First, we count the number of vehicles passing on each road. Then, we find the intersection pair corresponding to the road carrying the least number of vehicles. Finally, we regard this intersection pair as not being coordinated. In the example of three traffic signals, when the number of vehicles passing on the road between A and B is the least, and the first element of $\boldsymbol{z}$ represents whether A and B are coordinated, $\boldsymbol{z} = [0,1,1]$ is included in $\mathscr{Z}$. By pruning elements of $\mathscr{Z}$ beforehand, we can remove some coordination patterns that would otherwise introduce extra constraints on offsets from $\mathscr{Z}$. For example, $\boldsymbol{z} = [1,1,1]$ indicates that all traffic signals are coordinated. In this case, when the offsets of B from A and C from B are given, the offset of A from C is determined. Nevertheless, as this $\boldsymbol{z} = [1,1,1]$ and $\boldsymbol{z} = [0,1,1]$ represent the same groups of coordinated traffic signals, $\boldsymbol{z} = [1,1,1]$ is not included in $\mathscr{Z}$. $\boldsymbol{z} = [0,1,1]$ does not have such an issue.

4 GAUSSIAN PROCESSES AND BAYESIAN OPTIMIZATION

In this section, we give an explanation of Gaussian Processes, Arc Gaussian Processes, and Bayesian Optimization.

Gaussian Processes. Gaussian processes (GPs) are nonparametric models specified by a mean function $m : \mathscr{X} \to \mathbb{R}$ and a covariance kernel $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ (Rasmussen and Williams 2006). We use the constant mean function $m(\boldsymbol{x}) = m_0$. A typical kernel is the squared exponential kernel $k(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\|\boldsymbol{x}-\boldsymbol{x}'\|^2/(2h^2)\right)$, where hyper-parameter $h \in \mathbb{R}_+$ defines a length-scale. Squared exponential kernels are used in the experiments. GPs use Gaussian distributions for prediction; they can express both belief and confidence. Let $\boldsymbol{y}_n = [y_1,\ldots,y_n]$ and $\mathscr{D}_n = \{(\boldsymbol{x}_i,y_i) \mid i = 1,\ldots,n\}$ denote the concatenation of outputs and the set of input-output pairs, respectively. To deal with noisy outputs, we assume $y_i = \mathscr{F}(\boldsymbol{x}_i) + \varepsilon_i$ where $\varepsilon_i \sim N(0,\sigma^2)$ for all $i = 1,\ldots,n$. Then, the predicted mean $\mu_n(\boldsymbol{x})$ and covariance $\sigma_n^2(\boldsymbol{x})$ are given by

$$\mu_n(\boldsymbol{x}) = \boldsymbol{k}_n(\boldsymbol{x})\left(\boldsymbol{K}_n + \sigma^2\boldsymbol{I}\right)^{-1}\left(\boldsymbol{y}_n^\top - m_0\right), \qquad \sigma_n(\boldsymbol{x}) = k(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{k}_n(\boldsymbol{x})\left(\boldsymbol{K}_n + \sigma^2\boldsymbol{I}\right)^{-1}\boldsymbol{k}_n(\boldsymbol{x})^\top,$$

where $\boldsymbol{k}_n(\boldsymbol{x}) = [k(\boldsymbol{x}_1,\boldsymbol{x}),\ldots,k(\boldsymbol{x}_n,\boldsymbol{x})]$ and $\boldsymbol{K}_n = (k(\boldsymbol{x}_i,\boldsymbol{x}_j) \mid i,j = 1,\ldots,n)$.

Arc Gaussian Processes. Arc Gaussian processes (Arc GPs) are based on GPs and used for sharing information across coordination patterns (Swersky et al. 2014). Our method builds on them. As uncontrollable offsets are recorded as missing values, vanilla GP methods cannot deal with such offsets. Arc GPs embed D dimensional input data with missing elements $\boldsymbol{x}$ into $2D$ dimensional spaces with cylindrical coordinates. Let x_i denote the ith element of the D dimensional parameter $\boldsymbol{x}$. Bounds of x_i are defined as $\mathscr{X}_i = [x_{i,\max}, x_{i,\min}]$. For all i, x_i is embedded into

$$\mathscr{T}(x_i) = \begin{cases} [0,0] & \text{if } z_i = 0 \\ \left[\sin \pi\rho_i \frac{x_i}{R_{x_i}}, \cos \pi\rho_i \frac{x_i}{R_{x_i}}\right] & \text{if } z_i = 1, \end{cases}$$

where $\rho_i \in [0,1]$ and $R_{x_i} = x_{i,\max} - x_{i,\min}$. We can use GPs with standard covariance kernels in this space. Hyper-parameter ρ_i controls whether $\boldsymbol{z}$ is stronger than $\boldsymbol{x}$ in altering the prediction of GPs. As each x has a unique corresponding coordination pattern z, GPs model only x. Models for z are not necessary because z is implicitly modeled by GPs that model x.

Bayesian Optimization. Given search space $\mathscr{X} \subset \mathbb{R}^D$ and black-box function $\mathscr{F} : \mathscr{X} \to \mathbb{R}$, Bayesian optimization (BO) is a derivative-free algorithm for the following global optimization problem, $\boldsymbol{x}^* =$

$\arg\min_{\boldsymbol{x}\in\mathscr{X}} \mathscr{F}(\boldsymbol{x})$. (Shahriari et al. 2016). We note that finding an optimizer means the best result among iterations is optimal when the iteration number approaches infinity in our problem as actual applications pick up only the best parameter set. It differs from methods whose final iteration result is evaluated.

BO is a sequential model based approach that applies three main processes in each iteration t: updating the surrogate model to fit $\mathscr{F}(\boldsymbol{x})$, selecting the next parameters $\boldsymbol{x}_t$ for the simulation run by using the surrogate model, and running a simulation. In the first part, GPs are a common choice as surrogate models. The models are updated with the set of previous observations $\mathscr{D}_{t-1}$. In the second part, leveraging the surrogate model, we take the maximizer of the acquisition function in the search space as the parameters for the next simulation run,

$$\boldsymbol{x}_t = \arg\max_{\boldsymbol{x}\in\mathscr{X}} \alpha(\boldsymbol{x}).$$

Acquisition function $\alpha(\boldsymbol{x})$ is used to balance exploration and exploitation; the acquisition function is high when the prediction uncertainty is high (exploration) and the predicted result is promising (exploitation). A commonly-used acquisition function is expected improvement (EI) (Vazquez and Bect 2010). EI is written in closed form as $\alpha(\boldsymbol{x}) = (\tau - \mu(\boldsymbol{x}))\Phi(\gamma) + \sigma(\boldsymbol{x})\phi(\gamma)$, where τ denotes the current minimum of time losses of the simulation results and $\gamma = (\tau - \mu(\boldsymbol{x}))/\sigma(\boldsymbol{x})$. The functions $\Phi(\gamma)$ and $\phi(\gamma)$ are the standard normal cumulative distribution function and the standard normal probability density function, respectively. Optimality guarantees of this method are given by Vazquez and Bect (2010).

5 BACH

Here, we introduce BAyesian optimization for Conditional spaces with Hierarchical relationships (BACH). The strategy used to select parameters differentiates BACH from ordinary BO. We speed up the optimization process by deliberately restricting the search space in acquisition function optimization. In each iteration t, we select the parameters for the next simulation run in the following way. A formal description is given in Algorithm 1. Let $\boldsymbol{z}_t$ denote the candidate of the coordination pattern to be simulated. We substitute the coordination pattern selected in the previous iteration $\boldsymbol{z}_{t-1}$ into $\boldsymbol{z}_t$ (line 5), which is the start point of the candidate. Then, given $\boldsymbol{z}_t$, we obtain the maximizer of the acquisition function, $\boldsymbol{x}_t = \arg\max_{\boldsymbol{x}\in\mathscr{X}_{\boldsymbol{z}_t}} \alpha(\boldsymbol{x})$ (line 6). Note that the search space is restricted; it is $\mathscr{X}_{\boldsymbol{z}_t}$, not the whole space $\mathscr{X}$. Next, we define neighbors of a coordination pattern as the coordination patterns most similar to it; i.e. neighbors of $\boldsymbol{z}_t$ are obtained by replacing one of the elements of $\boldsymbol{z}_t$ by 1 (if it is 0) or 0 (if it is 1). For example, imagine $z = [0,0]$. As $[1,0]$ and $[0,1]$ are the closest to z in Euclidean distances, they are neighbors of z. We run the following iteration for all $\acute{\boldsymbol{z}}$ in neighbors of $\boldsymbol{z}_t$. Given $\acute{\boldsymbol{z}}$, we obtain the maximizer of the acquisition function, $\acute{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in\mathscr{X}_{\acute{\boldsymbol{z}}}} \alpha(\boldsymbol{x})$ (line 9). Then, we compare the acquisition function value $\alpha(\boldsymbol{x}_t)$ with $\alpha(\acute{\boldsymbol{x}})$. If $\alpha(\acute{\boldsymbol{x}})$ is higher than $\alpha(\boldsymbol{x}_t)$, we update $\boldsymbol{z}_t$ and $\boldsymbol{x}_t$ with $\acute{\boldsymbol{z}}$ and $\acute{\boldsymbol{x}}$, respectively (lines 10, 11). By running the iteration, candidate $\boldsymbol{z}_t$ is updated to the coordination pattern such that the maximum value of the acquisition function given this coordination pattern is highest among the candidate coordination pattern and its neighbors. If $\boldsymbol{z}_t$ is updated to a neighbor, we return to the beginning of the iteration with the new $\boldsymbol{z}_t$. This means that we obtain neighbors of the new $\boldsymbol{z}_t$ and update $\boldsymbol{z}_t$ by the procedure stated above. When $\boldsymbol{z}_t$ is not updated; i.e. $\alpha(\boldsymbol{x}_t)$ is higher than $\alpha(\acute{\boldsymbol{x}})$ for all $\acute{\boldsymbol{z}}$ in neighbors of $\boldsymbol{z}_t$, we do not return to the beginning of the iteration (lines 12, 13). Finally, we confirm $\boldsymbol{z}_t$ and $\boldsymbol{x}_t$ as the parameters for the next simulation run. This selection strategy restricts coordination patterns searched in each iteration. Note that in the next iteration $t+1$, as the start point of the coordination pattern candidate $\boldsymbol{z}_{t+1}$ is $\boldsymbol{z}_t$, which differs from $\boldsymbol{z}_{t-1}$, we iterate the procedures using the different start point.

The initialization procedure (line 2) is as follows. First, we sample $\boldsymbol{x}_1^{\text{init}}, \ldots, \boldsymbol{x}_{N_0}^{\text{init}}$ uniformly and randomly from $\mathscr{X}$. In the experiments, we set $N_0 = 5$. Then, we run simulations and obtain $y_i^{\text{init}} = \mathscr{F}(\boldsymbol{x}_i^{\text{init}})$ for $i = 1, \ldots, N_0$. The initial set of input-output pairs $\mathscr{D}_0$ is given by $\mathscr{D}_0 = \{(\boldsymbol{x}_i^{\text{init}}, y_i^{\text{init}}) \mid i = 1, \ldots, N_0\}$. Finally, the start point of the candidate coordination pattern $\boldsymbol{z}_0$ is such that $\boldsymbol{x}_0 \in \mathscr{X}_{\boldsymbol{z}_0}$, where $\boldsymbol{x}_0 = \arg\min_{\boldsymbol{x}\in\{\boldsymbol{x}_1^{\text{init}}, \ldots, \boldsymbol{x}_{N_0}^{\text{init}}\}} \mathscr{F}(\boldsymbol{x})$.

Algorithm 1 BAyesian optimization for **C**onditional spaces with **H**ierarchical relationships (BACH)

```
1: Input: black-box function (time loss) $\mathscr{F}(\boldsymbol{x})$, acquisition function $\alpha(\boldsymbol{x})$, set of coordination patterns
   $\mathscr{Z}$, search spaces given each coordination pattern $\{\mathscr{X}_{\boldsymbol{z}} \mid \boldsymbol{z} \in \mathscr{Z}\}$, maximum number of iterations $T$
2: Initialize $\mathscr{D}_0$ and $\boldsymbol{z}_0$
3: for $t = 1, 2, \ldots, T$ do
4:     Fit an Arc GP on $\mathscr{D}_{t-1}$
5:     $\boldsymbol{z}_t \leftarrow \boldsymbol{z}_{t-1}$
6:     $\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x} \in \mathscr{X}_{\boldsymbol{z}_t}} \alpha(\boldsymbol{x})$                ▷ given $\boldsymbol{z}_t$
7:     while True do
8:         $L \leftarrow \{\acute{\boldsymbol{z}} \mid \acute{\boldsymbol{z}}$ is a neighbor of $\boldsymbol{z}_t\}$
9:         for all $\acute{\boldsymbol{z}}$ in $L$ do
10:            $\acute{\boldsymbol{x}} \leftarrow \arg\max_{\boldsymbol{x} \in \mathscr{X}_{\acute{\boldsymbol{z}}}} \alpha(\boldsymbol{x})$                ▷ given $\acute{\boldsymbol{z}}$
11:            if $\alpha(\acute{\boldsymbol{x}}) > \alpha(\boldsymbol{x}_t)$ then
12:                Update $\boldsymbol{z}_t \leftarrow \acute{\boldsymbol{z}}$, $\boldsymbol{x}_t \leftarrow \acute{\boldsymbol{x}}$
13:        if no neighbors of $\boldsymbol{z}_t$ have higher acquisition function values than $\alpha(\boldsymbol{x}_t)$ then
14:            break
15:    Simulation run $y_t \leftarrow \mathscr{F}(\boldsymbol{x}_t)$
16:    $\mathscr{D}_t \leftarrow \mathscr{D}_{t-1} \cup \{(\boldsymbol{x}_t, y_t)\}$
17: Return best parameters recorded during iterations
```

Why do we deliberately restrict the search space? See Figure 4 for a graphic overview of BACH. Let $\boldsymbol{x}^*_{\boldsymbol{z}}$ denote sets of optimal offsets for given coordination pattern $\boldsymbol{z}$, $\boldsymbol{x}^*_{\boldsymbol{z}} = \arg\min_{\boldsymbol{x} \in \mathscr{X}_{\boldsymbol{z}}} \mathscr{F}(\boldsymbol{x})$. We consider finding $\boldsymbol{x}^*_{\boldsymbol{z}}$ for each $\boldsymbol{z} \in \mathscr{Z}$ as sub-goals as shown in the left figure. With sub-goals, our strategy is as follows. First, we target at a small number of similar coordination patterns. Let $\boldsymbol{z}_a$ denote one of targeted patterns. To search for $\boldsymbol{x}^*_{\boldsymbol{z}_a}$, we run simulations with various offsets while restricting the coordination pattern to $\boldsymbol{z}_a$. Selecting coordination patterns similar to $\boldsymbol{z}_a$ also helps the search for $\boldsymbol{x}^*_{\boldsymbol{z}_a}$ as we use the Arc GP model which can share information across coordination patterns. After we select targeted patterns a sufficient number of times, we find offsets near $\boldsymbol{x}^*_{\boldsymbol{z}_a}$, which we call the knowledge of $\boldsymbol{x}^*_{\boldsymbol{z}_a}$. Then, with the assumption discussed in Section 1, when another coordination pattern $\boldsymbol{z}_b$ is similar to $\boldsymbol{z}_a$, we transfer the knowledge of $\boldsymbol{x}^*_{\boldsymbol{z}_a}$ to search for $\boldsymbol{x}^*_{\boldsymbol{z}_b}$ as shown in the center figure. This leads to the quick acquisition of the new knowledge of $\boldsymbol{x}^*_{\boldsymbol{z}_b}$. We iterate the use of knowledge to gain new knowledge by searching coordination patterns not previously explored. While doing this, we gradually vary the targeted coordination patterns in the direction of the optimal one as shown in the right figure to discover $\boldsymbol{x}^*$.

Our strategy has the following advantages. First, because of knowledge transfer, we cut many simulation runs by not searching for unpromising offsets. If the coordination pattern we target is similar to an already searched one, we already know the candidates of offsets whose results are likely to be promising. By selecting them for the next simulation run, we can efficiently discover the optimal offsets for the targeted coordination pattern. Second, provided that we already know promising offsets for each of several coordination patterns, we can identify the best of these coordination patterns with ease. Assume that we do not know promising offsets. Even if the time losses of some simulation results differ, we do not know if this difference is due to the coordination pattern or offsets. In contrast, assume that the optimal offsets for several coordination patterns are known. As the simulation results do not have uncertainty in the time losses created by unoptimized offsets, we know that the difference originates from these coordination patterns. This makes comparison of these coordination patterns easier. This keeps us from selecting undesirable coordination patterns. Efficiently selecting promising coordination patterns is important to reduce the number of simulation runs. We note that we merely state that BACH is based on this assumption. BACH performs well only if this assumption is true for a problem.

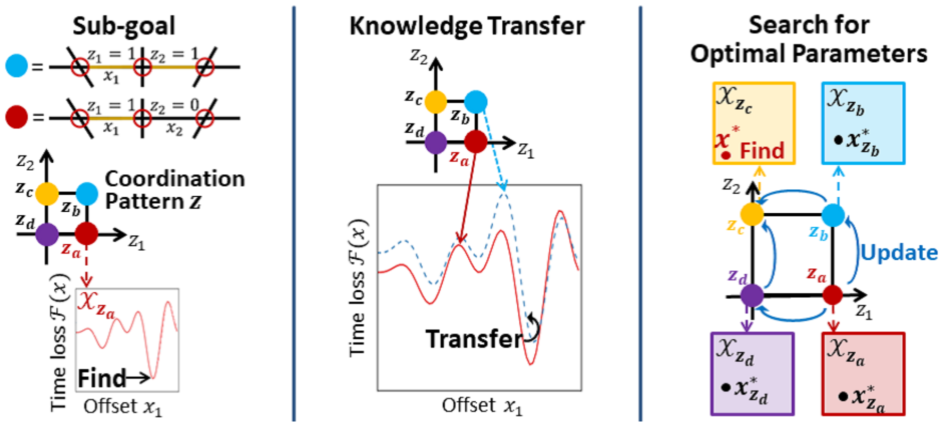

Figure 4: Graphic overview of BACH in a 2-dimensional case. (left) Each filled circle represents a coordination pattern. We consider finding the optimal offsets for a given coordination pattern as a sub-goal. (center) The illustration shows that we transfer the knowledge of the red solid line to the optimization of the blue dashed line. (right) The illustration shows that we search coordination patterns similar to those already searched and that targeted patterns gradually approach the optimal one.

6 EXPERIMENTS

We evaluate BACH in two experiments. We first study the property of BACH for the optimization of testing functions. Then, we show it yields optimum traffic signal control.

Experimental Setup. For BACH and GP-based baselines, we update the hyper-parameters of the GP kernel by maximizing the GP marginal likelihood in each iteration. Additionally, following (Kandasamy et al. 2015), we allocate a budget of 200D function evaluations when maximizing the acquisition function. We use EI as an acquisition function in the experiments. We use multi-start local search when maximizing acquisition functions. This method randomly chooses some initial points and obtains local optima by using L-BFGS. To obtain error bars, we performed multiple independent optimization runs of each method and plot medians results, along with 25% and 75% percentiles as uncertainties.

6.1 On Testing Functions

We consider testing functions given by

$$\mathscr{F}(\boldsymbol{x},\boldsymbol{z}) = \sum_{i=1}^{D} 4(x_i - 1/2)^2 z_i + \lambda \exp\left(\|\boldsymbol{z} - \hat{\boldsymbol{z}}\|^2/a\right).$$

The objective is to minimize the function. The first term represents the loss model induced by offsets. We call it the *offset term*. The term is a sum of sphere functions. Uncontrollable offsets do not influence the term since they are indicated by $z_i = 0$. Coordination patterns do not directly affect the term; they are related to the term only to exclude uncontrollable offsets. The term is minimized when $x_i = 1/2$ for all i such that $z_i = 1$. The second term represents the Gaussian-like loss model induced by the coordination patterns. We call this the *coordination term*. The term is minimized when $\boldsymbol{z} = \hat{\boldsymbol{z}}$. We set $\hat{\boldsymbol{z}} = [1,0,1,0,1,0,1,0,1,0]^\top$ and $a = 10$ in the experiment. Parameter λ controls the balance between the two terms. Results shown below are for case of $\lambda = 1$ but similar results were obtained for $\lambda = 0.1, 0.2, 0.5, 2$, and 5. Since the function can be broken down into two terms, we can discuss the losses induced from the coordination pattern and offsets separately. This contributes to an understanding of method behavior. We call these functions *Sphere-Gaussian functions*. We set $D = 10$. The number of coordination patterns is 1,024. For each method and parameter λ, we performed 20 independent optimization runs of 100 iterations.

Baselines. We compare BACH with Random2x, SMAC (Hutter et al. 2011), Tree BO (Jenatton et al. 2017), Independent BO (Bergstra et al. 2011), and Arc BO (Swersky et al. 2014). Random2x is random search with twice the budget. BACH and baselines (other than SMAC and random2x) are based on GPs. For SMAC, we use publicly available software (Hutter et al. 2011). For the others, we wrote our own.

Results. Figure 5(a) shows the optimization results for $\lambda = 1$. The vertical axis plots the optimality gap, the gap between the best function value obtained so far and the optimal value. The result shows that BACH outperforms all baselines in terms of the optimality gap with 100 iterations. The optimality gap of BACH plunges before the number of iterations reach 50. Figure 5(b) shows the breakdown results of BACH and Arc BO. The figure shows that BACH successfully optimized the function using the strategy stated in Secs. 1 and 5. At the beginning of the optimization run, BACH offers a faster decrease in the offset term than Arc BO. This indicates that offsets can be optimized for at least one coordination patterns by targeting a small number of similar coordination patterns. The main component of total optimality gap is the coordination term. Next, the coordination term drops to 0 within 50 iterations. BACH finds the optimal coordination pattern faster than Arc BO in terms of the median performance. After the coordination term attenuates, the offset term becomes the main factor determining the total optimality gap. The offset term is kept small when the optimal coordination pattern is found. The drop of the coordination term explains the plunge in the total optimality gap, which yields the superior performance of BACH.

6.2 On Traffic Signal Control

For the simulations, we employed an open-source microscopic road traffic simulator, Simulation of Urban Mobility (SUMO) (Krajzewicz et al. 2005). We use the Trapezoid and Luxembourg scenarios. The Trapezoid scenario was synthetically generated and contains 9 intersections shown in Figure 6(a). The road shape of this scenario is trapezoid-like, hence its name. The scenario contains 12 offsets and 3102 coordination patterns. The Luxembourg scenario is a scenario based on the traffic in Luxembourg (Codeca et al. 2015). We target 10 intersections in the vicinity of the city center shown in Figure 6(b). Traffic signals during the morning rush hour, from 08:00 to 08:30, are targeted for optimization. Both BACH and baselines calculates the optimal offsets and coordination pattern for the whole 30 min. As real signals need to gradually change parameters while taking several cycles of signal indication sequences to adjust new cycle length and offset setting, updating parameters more frequently is not practical. The scenario contains 11 offsets and 1968 coordination patterns. In all scenarios, the number of vehicles and routes of vehicles are fixed beforehand. Vehicles do not detour. As discussed above, we minimize the metric of *time loss*. Let T_d and T_a denote the sum of elapsed time when vehicles are driven at the desired speed and the sum of actual travel time. Time loss is given by $T_a - T_d$. The desired speed is the maximum speed. Time loss increases as vehicle speed falls. We note that BACH can work with arbitrary metrics. In addition to offsets, traffic signals have two other types of parameters: cycle lengths and splits. They are optimized by Webster method (Webster 1958). Although initial offsets of uncoordinated intersection pairs are set to 0, they are meaningless because they will deviate. For each method and scenario, we performed 20 optimization runs of 100 iterations.

Additional Baselines. In addition to those stated in Section 6.1, we added GA4x (Mitchell 1998) and PSO4x (Poli et al. 2007) to the baselines. GA4x and PSO4x are implemented with four times the budget because these methods often require more simulation runs than BO. We used publicly available software to implement GA and PSO (Fortin et al. 2012).

Results. Figures 5(c,d) show the optimization results for the two scenarios. Since GA and PSO run multiple simulations in each iteration, the result is updated after each iteration. The first part of the optimization curve is not shown because the first iteration is not finished. Interestingly, BACH outperforms all baselines in both Trapezoid and Luxembourg scenarios in terms of time losses with 100 simulation runs. BACH demonstrates behavior similar to that in the test function experiment; optimization speed of BACH rapidly increases in the middle of the optimization runs. Optimization speed of BACH rapidly increases in the middle of the optimization runs. We note that mean wall-clock times required for 100 iterations

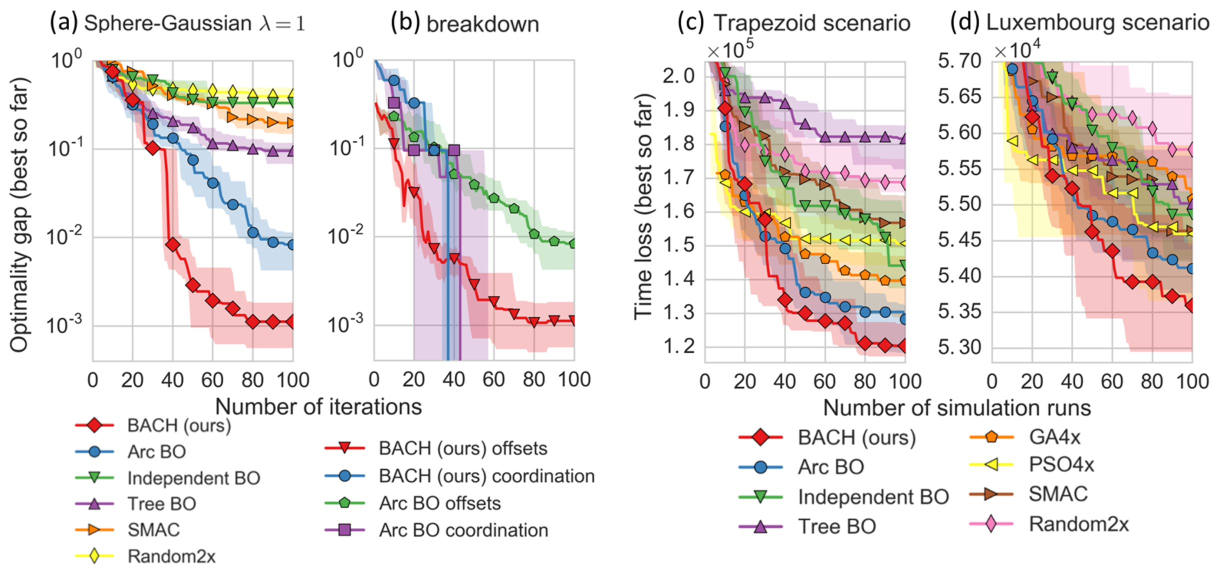

Figure 5: (a) Comparison of BACH and baselines for optimizing a testing (Sphere-Gaussian) function with $\lambda = 1$. We plot the optimality gap curves of methods versus the number of iterations. We plot median and quartile of the best performance across 20 runs. BACH is more efficient than the other methods. (b) Breakdown of the optimality gap curves shown in the left figure into the offset and coordination terms, which represent the loss induced by offset and coordination, respectively. BACH finds favorable offsets for a coordination pattern and then discovers the combination of the best coordination pattern and promising offsets. (c,d) Performance comparison of BACH and other methods for optimizing traffic signal control in the (c) Trapezoid and (d) Luxembourg scenarios. In the Trapezoid scenario, competitors include Oracle BO (i.e., knows the optimal coordination pattern and uses BO to simply tune offsets for the given coordination pattern). We plot the time loss curves of the methods versus the number of simulation runs. Lower is better. Among the methods, only BACH converges toward Oracle BO in the Trapezoid scenario. Moreover, BACH is more efficient than the other methods in all scenarios.

of BACH excluding simulation times are 309 s and 306 s in the Trapezoid and Luxembourg scenarios, respectively. Because those of methods using GPs are similar, BACH is efficient in terms of wall-clock times. In addition, mean simulation times are 176 s and 15,244 s in the Trapezoid and Luxembourg scenarios, respectively. Even light methods can run less than 3 times as more simulations as BACH in the Trapezoid scenario and most of times are elapsed by simulation runs in the Luxembourg scenario. Since we demonstrate that BACH outperform GA and PSO with four times the budget, BACH is faster than them.

7 CONCLUSION

We addressed coordination control of traffic signals in detail. Parameters of the problem are the coordination pattern and offsets. Due to their hierarchical relationships, each coordination pattern has some offsets that are not controllable. To tackle the problem, we presented BACH, an effective method for optimization in conditional parameter spaces that exhibit hierarchical relationships. With the assumption that the knowledge of the offsets optimized for a given coordination pattern can be transferred to those of similar patterns, BACH starts by intensively selecting similar coordination patterns to collect knowledge, and then shifting its target to finding the optimal combination of the coordination pattern and offsets by leveraging knowledge transfer. BACH effectively exploits the structure of the parameter spaces. We thoroughly evaluated its performance in optimization experiments that examined simulation-based traffic signal control. We demonstrated its superior efficacy over a variety of state-of-the-art methods.

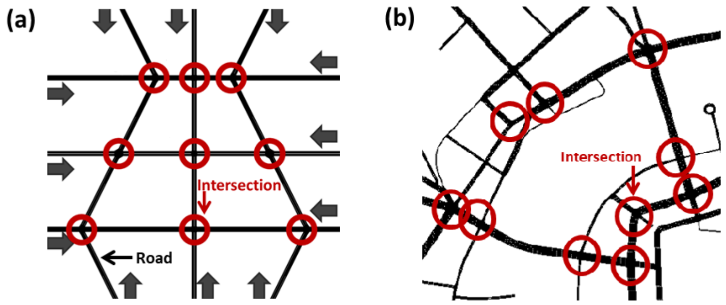

Figure 6: Illustrations of road networks of (a) synthetic and (b) Luxembourg scenarios.

REFERENCES

Armas, R., H. Aguirre, F. Daolio, and K. Tanaka. 2017. "Evolutionary Design Optimization of Traffic Signals Applied to Quito City". *PloS one* 12(12):1–37.

Bergstra, J., R. Bardenet, Y. Bengio, and B. Kégl. 2011. "Algorithms for Hyper-parameter Optimization". In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, edited by J. Shawe-Taylor and R.S. Zemel and P.L. Bartlett and F. Pereira and K.Q. Weinberger, 2546–2554. Red Hook, New York: Curran Associates Inc.

Bowman, C. N., and J. A. Miller. 2016. "Modeling Traffic Flow Using Simulation and Big Data Analytics". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 1206–1217. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Codeca, L., R. Frank, and T. Engel. 2015. "Luxembourg SUMO Traffic (LuST) Scenario: 24 Hours of Mobility for Vehicular Networking Research". In *2015 IEEE Vehicular Networking Conference*. July 16th-18th, Kyoto, Japan, 1–8.

Fortin, F.-A., F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné. 2012, jul. "DEAP: Evolutionary Algorithms Made Easy". *Journal of Machine Learning Research* 13:2171–2175.

Garcia-Nieto, J., A. C. Olivera, and E. Alba. 2013, Dec. "Optimal Cycle Program of Traffic Lights With Particle Swarm Optimization". *IEEE Transactions on Evolutionary Computation* 17(6):823–839.

Hutter, F., H. H. Hoos, and K. Leyton-Brown. 2011. "Sequential Model-based Optimization for General Algorithm Configuration". In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, edited by Carlos A. Coello Coello, 507–523. Berlin, Germany: Springer-Verlag.

Jenatton, R., C. Archambeau, J. González, and M. Seeger. 2017, 06–11 Aug. "Bayesian Optimization with Tree-structured Dependencies". In *Proceedings of the 34th International Conference on Machine Learning*, edited by D. Precup and Y. W. Teh, 1655–1664. Sydney, Australia: Proceedings of Machine Learning Research .

Kandasamy, K., J. Schneider, and B. Poczos. 2015, 07–09 Jul. "High Dimensional Bayesian Optimisation and Bandits via Additive Models". In *Proceedings of the 32nd International Conference on Machine Learning*, edited by F. Bach and D. Blei, 295–304. Lille, France: Proceedings of Machine Learning Research .

Koonce, P., L. Rodegerdts, K. Lee, S. Quayle, S. Beaird, C. Braud, J. Bonneson, P. Tarnoff, and T. Urbanik. 2008. *Traffic Signal Timing Manual*. Washington, DC: National Academies of Sciences, Engineering, and Medicine.

Krajzewicz, D., E. Brockfeld, J. Mikat, J. Ringel, C. Rössel, W. Tuchscheerer, P. Wagner, and R. Wösler. 2005. "Simulation of Modern Traffic Lights Control Systems Using the Open Source Traffic Simulation SUMO". In *Proceedings of the 3rd Industrial Simulation Conference*. June 9th-11th, Berlin, Germany, 229–302.

Lertworawanich, P., M. Kuwahara, and M. Miska. 2011. "A New Multiobjective Signal Optimization for Oversaturated Networks". *IEEE Transactions on Intelligent Transportation Systems* 12(4):967–976.

Mitchell, M. 1998. *An Introduction to Genetic Algorithms*. Cambridge, Massachusetts: MIT Press.

Osorio, C., and L. Chong. 2012. "An Efficient Simulation-based Optimization Algorithm for Large-scale Transportation Problems". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 423:1–423:11. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Papageorgiou, M., C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. 2003. "Review of Road Traffic Control Strategies". *Proceedings of the IEEE* 91(12):2043–2067.

Pearce, M., and J. Branke. 2017. "Bayesian Simulation Optimization with Input Uncertainty". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 181:1–181:11. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Poli, R., J. Kennedy, and T. Blackwell. 2007. "Particle Swarm Optimization". *Swarm intelligence* 1(1):33–57.

Poloczek, M., J. Wang, and P. I. Frazier. 2016. “Warm Starting Bayesian Optimization”. In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 770–781. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rasmussen, C. E., and C. K. I. Williams. 2006. *Gaussian Processes For Machine Learning.* Cambridge, Massachusetts: MIT Press.

Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. 2016. “Taking the Human out of the Loop: a Review of Bayesian Optimization”. *Proceedings of the IEEE* 104(1):148–175.

Swersky, K., D. Duvenaud, J. Snoek, F. Hutter, and M. A. Osborne. 2014. “Raiders of the Lost Architecture: Kernels for Bayesian Optimization in Conditional Parameter Spaces”. In *Proceedings of NIPS Workshop on Bayesian Optimization 2014*. December 10th, Lake Tahoe, Nevada, USA.

Vazquez, E., and J. Bect. 2010. “Convergence Properties of the Expected Improvement Algorithm with Fixed Mean and Covariance Functions”. *Journal of Statistical Planning and Inference* 140(11):3088–3095.

Wang, L., X. Liu, Z. Wang, and Z. Li. 2013. “Urban Traffic Signal System Control Structural Optimization Based on Network Analysis”. *Mathematical Problems in Engineering* 2013:1–9.

Webster, F. V. 1958. “Traffic Signal Settings”. Road Research Technical Paper No. 39. Road Research Laboratory, London.

Wei, H., G. Zheng, H. Yao, and Z. Li. 2018. “IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control”. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, edited by R. Rosales and J. Tang, 2496–2505. New York City, New York: Association for Computing Machinery.

AUTHOR BIOGRAPHIES

HIDETAKA ITO is a researcher at NTT Service Evolution Laboratories, Japan. He received a B.E. and M.E. in Aeronautics and Astronautics from the University of Tokyo, Japan, in 2014 and 2016, respectively. His current research focuses on efficient global optimization algorithms for applications in simulation optimization. His email address is hidetaka.ito.gt@hco.ntt.co.jp.

KYOTA TSUTSUMIDA is a research engineer at NTT Service Evolution Laboratories, Japan. He received a Master of Media and Governance from Keio University, Japan, in 2009. His research interests include intelligent transportation system, geo-spatial information science, recommender systems, and natural language processing. He is a member of the Information Processing Society of Japan (IPSJ) and the Japanese Society for Artificial Intelligence (JSAI). His email address is kyouta.tsutsumida.nm@hco.ntt.co.jp.

TATSUSHI MATSUBAYASHI is a senior research engineer at NTT Service Evolution Laboratories, Japan. He received the B.S. degree in Physics from Kyoto University, Japan, in 2000. He received the M.S. and Ph.D. degrees in Astrophysics from Tokyo Institute of Technology, Tokyo, Japan, in 2002 and 2006, respectively. His research interests lie in the area of high-performance computing, information visualization, and machine learning. He is a member of the Information Processing Society of Japan (IPSJ). His email address is tatsushi.matsubayashi.tb@hco.ntt.co.jp.

TAKESHI KURASHIMA Takeshi Kurashima is a senior research engineer at NTT Service Evolution Laboratories, Japan. He received the B.S. degree from Doshisha University, Japan, in 2004, the M.S. degree from Kyoto University, Japan, in 2006, and the Ph.D. degree in Informatics from Kyoto University, Japan, in 2014. His research interests include data mining and recommender systems. His email address is takeshi.kurashima.uf@hco.ntt.co.jp.

HIROYUKI TODA is a senior research engineer, supervisor, at NTT Service Evolution Laboratories, Japan. He recieved a B.E. and M.E. in materials science from Nagoya University in 1997 and 1999, and a Ph.D. in computer science from University of Tsukuba in 2007. He joined NTT in 1999. His current research interests include information retrieval, data mining, and ubiquitous computing. He is a member of the Information Processing Society of Japan (IPSJ), the Institute of Electronics, Information and Communication Engineers (IEICE), the Japanese Society for Artificial Intelligence (JSAI), the Database Society of Japan (DBSJ), and the Association for Computing Machinery (ACM). His email address is hiroyuki.toda.xb@hco.ntt.co.jp.