2017 American Control Conference
Sheraton Seattle Hotel
May 24–26, 2017, Seattle, USA

Offset Optimization of Signalized Intersections via the Burer-Monteiro Method

Eric S. Kim, Cheng-Ju Wu, Roberto Horowitz, Murat Arcak

Abstract— **This paper tackles the offset optimization problem which seeks to coordinate traffic signals in a large traffic network. By assuming that all signals have a common cycle time and that the arrival/departure rates at each intersection can be approximated by sinusoids, the original non-convex offset optimization problem can be relaxed into a semidefinite program (SDP). SDP solvers unfortunately run out of memory for larger networks with thousands of intersections. The Burer-Monteiro (BM) method [1] for solving large SDPs avoids conic constraints and solves a lower dimensional problem but is non-convex. A synthetic New York City example with 1771 intersections showcases the scalability of the BM method. Another example involving 420 intersections in Los Angeles demonstrates that the BM method recovers optimal solutions of the SDP. Moreover, a detailed microsimulation of the Los Angeles network shows that the optimized offsets result in reduced delay time and smaller queues.**

I. Introduction

In arterial road networks, vehicle idling time at signalized intersections is a major source of delays and emissions. Minimization of idling time can be achieved by aligning signal offsets to encourage sequences of green lights along heavily utilized roads. The offset optimization problem as proposed in [2] approximated queueing processes with sinusoids and sought to minimize queue oscillations which appear as a symptom of ill timed offsets. The problem is amenable to semidefinite relaxation (SDP), which was used to both certify a lower bound on the minimization problem and as a heuristic to obtain near optimal offsets.

For larger networks with thousands of intersections, interior point solvers used in SDPs typically run out of memory. We use the Burer-Monteiro (BM) method [1] for solving SDPs to avoid the expensive conic constraints and dramatically reduce the number of optimization variables. This method is a non-linear heuristic for solving SDPs with low rank solutions with surprisingly good results in practice; prior uses of the BM method include e.g. matrix completion [3] and angular synchronization [4].

Two examples showcase both the algorithm's scalability as well as improvements to common traffic metrics in a microscopic simulation. The first is a synthetic example with 1771 signalized intersections in the Manhattan borough of New York City, highlighting the scalability of the Burer-Monteiro method. Our second example is a microscopic simulation of Pasadena, a suburb of Los Angeles, California with 420 intersections. This simulation uses real green splits, cycle times, and empirically observed demands and was provided by the Connected Corridors project at UC Berkeley. The optimized offsets outperform existing offsets with respect to average delay and queue occupancy.

This work was supported in part by NSF grant CNS-1446145 and the NSF Graduate Research Fellowship Program. Authors Kim and Arcak are with the Department of Electrical Engineering and Computer Sciences and and authors Wu and Horowitz are with the Department of Mechanical Engineering at the University of California, Berkeley. Emails: {eskim, chengju, horowitz, arcak}@berkeley.edu

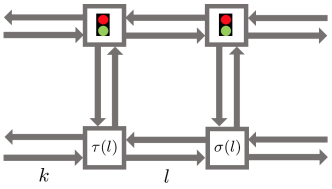

Fig. 1. Example network

The paper is organized as follows. Section II presents the traffic flow model and assumptions used in this study. Section III presents the offset optimization model and Section IV introduces the semidefinite relaxation and the Burer-Monteiro method. We showcase two benchmark examples in Section V and provide conclusions in Section VI.

II. Traffic Flow Model

We describe the routing model and queue dynamics adopted previously for the offset optimization problem by Coogan et al. [2].

A traffic network is modeled as a set of links/roads $\mathcal{L}$ and nodes/signalized intersections $\mathcal{S}$. Each link has a vehicle queue where vehicles wait to be discharged. Let $\sigma : \mathcal{L} \mapsto \mathcal{S}$ map a link to the signalized intersection that actuates a link's queue. Similarly, $\tau : \mathcal{L} \mapsto \mathcal{S} \cup \{\epsilon\}$ maps a link to the signalized intersection from which vehicles enter the link. When $\tau(l) = \epsilon$ then link l is an entry link and no such upstream link exists; such a l is an entry link and $\mathcal{E} = \{l \in \mathcal{L} : \tau(l) = \epsilon\}$. Vehicles on l flow from the *upstream* intersection $\tau(l)$ to the *downstream* intersection $\sigma(l)$. An illustration of the notation is provided in Figure 1.

Each intersection uses a fixed time control strategy where a periodic sequence of non-conflicting links are allowed to discharge vehicles.

Assumption 1: Each signalized intersection in the network has a common cycle period T or, identically, a common cycle frequency $\omega = \frac{2\pi}{T}$.

Each link has an associated queue where vehicles wait to be released. We assume a fluid model for queues, which

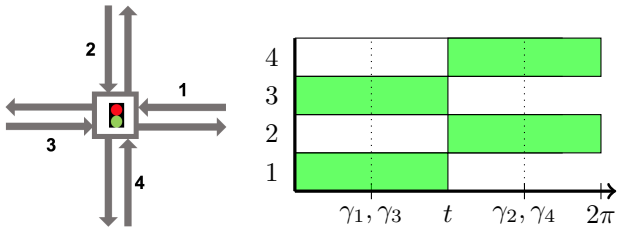

Fig. 2. Green splits of a four-way intersection. East-West links 1,3 have green splits $\gamma_1, \gamma_3 = \frac{\pi}{2}$ and North-South links 2,4 have green splits $\gamma_2, \gamma_4 = \frac{3\pi}{2}$

have a length $q_k(t) \in \mathbb{R}_{\geq 0}$ at time t and exhibit continuous dynamics driven by arrivals and departures for all $l \in \mathcal{L}$:

$$\dot{q}_l(t) = a_l(t) - d_l(t).$$

Link flows are controlled with red and green lights. Signals at all intersections obey a global clock and each intersection s has an *offset* $\theta_s \in [0, 2\pi)$ that represents a phase difference from that global clock. During each period, each link is actuated with a contiguous green subinterval. The middle of that subinterval is given by a *green split* $\gamma_l \in [0, 2\pi)$ and hence the middle of link l's green interval is actuated on $t = 2\pi n + \theta_{\sigma(l)} + \gamma_l$ for $n = 1, 2, \ldots$. Figure 2 shows a typical signal configuration for a four-way intersection, where the North-South movements and East-West movements are actuated on disjoint intervals.

Parameter $\beta_{kl} \in [0, 1]$ is the proportion of the vehicles exiting link k that enter link l. If A_k is the average flow of vehicles exiting link k, then the average flow of vehicles entering link l from k is $\beta_{kl} A_k$. Flows can only be non-zero on adjacent links; more precisely, $\beta_{kl} \neq 0$ only if $\tau(l) = \sigma(k)$. Furthermore, $\sum_{l \in \mathcal{L}} \beta_{kl} \leq 1$ where strict inequality signifies that some vehicles are exiting the network of modeled roads.

This framework is general enough to accommodate more granular models that include multiple lanes and unsignalized intersections such as stop signs or junctions. Section V-B provides further details on a mild reformulation made to apply our work on a microsimulation benchmark.

III. Minimizing Queue Oscillation

A. Approximation with Sinusoids

For a network where all signalized intersections have a common cycle time and the network can accomodate exogenous arrivals, the network has a periodic steady state [5].

Assumption 2: We assume that the network is under a periodic steady state with a period given by $T = \frac{2\pi}{\omega}$, i.e.,

$$\forall l \in \mathcal{L} \qquad x_l(t) = x_l(t + Tn) \text{ for } n = 1, 2, \ldots. \tag{1}$$

The partitioning of a period into red and green subintervals induces a platooning effect where the departure rate of vehicles has peaks and troughs. This effect is approximated by modeling arrival and departure rates as sinusoids.

1) Entry Links: Entry links $l \in \mathcal{E}$ are assumed to have a periodic arrival rate $a_l(t)$

$$a_l(t) := A_l + \alpha_l \cos(\omega t - \varphi_l) \tag{2}$$

where A_l represents the mean arrival rate, α_l encodes a fluctuation, and φ_l is centered at the peak arrival rate. It's clear that $A_l, \alpha_l \geq 0$ and $A_l \geq \alpha_l$.

To obtain the departure process, first note that under the steady state assumption

$$\int_T a_l(t)dt = \int_T d_l(t)dt. \tag{3}$$

The departure rate $d_l(t)$ is then approximated with a sinusoid with a peak at the offset $\varphi_{\sigma(l)} + \gamma_l$.

$$d_l(t) := A_l(1 + \cos(\omega t - \varphi_{\sigma(l)} - \gamma_l)) \quad \forall l \in \mathcal{E}. \tag{4}$$

2) Internal Links: For non-entry link l the arrival rate is a sum of the departure rates of upstream links, except with an appropriate scaling factor from turning ratios and a travel time $\lambda_l \in [0, 2\pi)$ on link l

$$a_l(t) = \sum_{k \in \mathcal{L}} \beta_{kl} A_k (1 + \alpha_l \cos(\omega t - \theta_{\sigma(k)} - \gamma_k - \lambda_l)). \tag{5}$$

If $\tau(l) = \sigma(k)$ for any k, l such that $\beta_{kl} \neq 0$ then

$$a_l(t) = A_l + \alpha_l \cos(\omega t - \theta_{\tau(l)} - \varphi_l) \tag{6}$$

where α_l and φ_l for $l \in \mathcal{L} \setminus \mathcal{E}$ are

$$\alpha_l = \sqrt{\left(\sum_{k \in \mathcal{L}} \beta_{kl} A_k \cos(\gamma_k)\right)^2 + \left(\sum_{k \in \mathcal{L}} \beta_{kl} A_k \sin(\gamma_k)\right)^2} \tag{7}$$

$$\varphi_l = \lambda_l + \tan^{-1}\left(\frac{\sum_{k \in \mathcal{L}} \beta_{kl} A_k \sin(\gamma_k)}{\sum_{k \in \mathcal{L}} \beta_{kl} A_k \cos(\gamma_k)}\right). \tag{8}$$

Note that conservation of mass ensures that

$$\sum_{k \in \mathcal{L}} \beta_{kl} A_k = A_l. \tag{9}$$

The departure process is identical as for entry links:

$$d_l(t) := A_l(1 + \cos(\omega t - \varphi_{\sigma(l)} - \gamma_l)) \quad \forall l \in \mathcal{L}. \tag{10}$$

An internal queue's rate of change is given by.

$$\begin{aligned} \dot{q}_l(t) &= a_l(t) - d_l(t) && (11) \\ &= \alpha_l \cos(\omega t - \theta_{\tau(l)} - \varphi_l) \\ &\quad - A_l \cos(\omega t - \theta_{\sigma(l)} - \gamma_l). && (12) \end{aligned}$$

Through a simple application of Euler's identity and phasors, the sum can be simplified to:

$$\dot{q}_l(t) = M_l(\theta) \cos(\omega t - \eta_l)$$

for some phase shift η_l and amplitude:

$$M_l(\theta) = \sqrt{A_l^2 + \alpha_l^2 - 2\cos\left((\theta_{\sigma(l)} + \gamma_l) - (\theta_{\tau(l)} + \varphi_l)\right)}. \tag{13}$$

The resulting explicit representation of $q_l(t)$ as a function of θ is:

$$q_l(t) = \frac{M_l(\theta)}{\omega}\cos(\omega t - \eta_l) + C_l \tag{14}$$

where $C_l > 0$ is a constant representing the average queue length with respect to time.

B. Optimization

For each link consider the cost function:

$$J_l(\theta) = \left(\frac{M_l(\theta)}{\omega}\right)^2$$

and observe from (13) that this shares a minimum with $R_l(\theta) = -A_l\alpha_l\cos\big((\theta_{\sigma(l)}+\gamma_l) - (\theta_{\tau(l)}+\varphi_l)\big)$. Both $J_l(\theta)$ and $R_l(\theta)$ are minimized when:

$$\theta_{\sigma(l)} + \gamma_l = \theta_{\tau(l)} + \varphi_l. \tag{15}$$

Condition (15) intuitively means that center of the green light $\theta_{\sigma(l)} + \gamma_l$ of intersection $\sigma(l)$ should match the time $\theta_{\tau(l)} + \varphi_l$ when the vehicle arrival rate from upstream intersection $\tau(l)$ peaks. Any satisfying θ is a minimizer of $J_l(\theta)$ and $R_l(\theta)$. Condition (15) is easily satisfied with a single link l and a pair of intersection offsets $\theta_{\sigma(l)}, \theta_{\tau(l)}$ but this is generally not the case for entire networks with complex topologies. Since such constraints cannot be satisfied simultaneously, we consider the sum $\sum_{l\in\mathcal{L}} R_l(\theta)$ as an objective function:

$$\min_{\theta} -\sum_{l\in\mathcal{L}} A_l\alpha_l\cos(\theta_{\tau(l)} + \varphi_l - (\theta_{\sigma(l)} + \gamma_l)). \tag{16}$$

The A_l component of the coefficient weights prioritizes links with higher flow. The α_l component prioritizes links with higher arrival rates fluctuation; these links would benefit more from well aligned offsets.

IV. Computing Lower Bounds via Semidefinite Relaxation

A. Reformulating as a Semidefinite Program

The offset optimization problem (16) is non-convex but is readily amenable to semi-definite relaxation. After introducing the substitution $z = (x, y)$ where $x_s = \cos(\theta_s)$ and $y_s = \sin(\theta_s)$, we can construct a W using trigonometric sum and difference identities such that the objective in (16) is equal to $z^T W z$. For $s, u \in \mathcal{S}$, let

$$W_1[s,u] = -\sum_{l\in\mathcal{L}_{s\to u}} A_l\alpha_l\cos(\varphi_l - \gamma_l)$$
$$W_2[s,u] = -\sum_{l\in\mathcal{L}_{s\to u}} A_l\alpha_l\sin(\varphi_l - \gamma_l)$$

where

$$\mathcal{L}_{s\to u} = \{l \in \mathcal{L} | \tau(l) = s \text{ and } \sigma(l) = u\}.$$

Finally, we arrive at W:

$$\bar{W} = \begin{bmatrix} W_1 & W_2 \\ -W_2 & W_1 \end{bmatrix}, W = \frac{1}{2}(\bar{W} + \bar{W}^T).$$

Each pair x_s, y_s must also satisfy $x_s^2 + y_s^2 = 1$. This constraint is encoded via a set of matrices M_s which essentially act as indicator functions that select elements of z. Let $E_s \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ have only one non-zero element such that $E_s[s,s] = 1$. Then,

$$M_s = \begin{bmatrix} E_s & 0 \\ 0 & E_s \end{bmatrix}.$$

The optimal value of problem (17) is equivalent to problem (16).

$$\min_{z\in\mathbb{R}^{2|\mathcal{S}|}} z^T W z \tag{17}$$
$$s.t.\ z^T M_s z = 1 \quad \forall s \in \mathcal{S}.$$

Problem (17) is a non-convex quadratically constrained quadratic program that can be reformulated to a semidefinite program (SDP) with a rank constraint by using the circular property of the trace: $\mathbf{Tr}(z^T W z) = \mathbf{Tr}(W z z^T)$.

$$\min_{Z\succeq 0} \mathbf{Tr}(WZ) \tag{18}$$
$$s.t.\ \mathbf{Tr}(M_s Z) = 1 \quad \forall s \in \mathcal{S}$$
$$\text{rank}(Z) = 1.$$

Dropping the rank constraint yields a convex optimization problem:

$$\min_{Z\succeq 0} \mathbf{Tr}(WZ) \tag{19}$$
$$s.t.\ \mathbf{Tr}(M_s Z) = 1 \quad \forall s \in \mathcal{S}.$$

Because (19) is a relaxation, its minimum provides a lower bound on (17) and when a minimizing Z is rank 1, then it is a tight lower bound.

B. Recovering a Feasible Solution from the SDP

For rank 1 solutions to the SDP we can transform back to the offsets via

$$\theta_s = \arctan\left(\frac{y_s}{x_s}\right) \quad \forall s \in \mathcal{S}. \tag{20}$$

In general, the solution to the semidefinite relaxation is not guaranteed to yield a rank one solution and be feasible for the original problem. One may consider taking the principal eigenvector of the minimizer $X = RR^T$ and feeding it into Equation (20); however, this is often not a local minimum, let alone a global minimum. A reasonable heuristic is to use the principal eigenvector as an initial point for our non-linear solver routine applied to the original offset optimization problem (16).

C. Burer-Monteiro Algorithm

Interior point solvers for semidefinite programs begin to run out of memory roughly when $|\mathcal{S}|$ is in the low thousands. The Burer-Monteiro algorithm for solving SDPs [1] applies the nonlinear change of variables $Z = RR^T$ where $R \in \mathbb{R}^{2|\mathcal{S}|\times r}$ has a rank upper bound of $r \le 2|\mathcal{S}|$.

$$\min_{R\in\mathbb{R}^{2|\mathcal{S}|\times r}} \mathbf{Tr}(WRR^T) \tag{21}$$
$$s.t.\ \mathbf{Tr}(M_s RR^T) = 1 \quad \forall s \in \mathcal{S}.$$

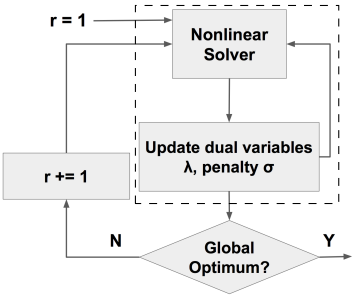

Fig. 3. Control flowchart of the Burer-Monteiro Algorithm. The dashed box denotes the rank r subroutine in Algorithm 1.

Although we pay the cost of transforming an originally convex optimization problem to a nonlinear program, we are free of conic constraints and dramatically reduce the dimension of the search space. Moreover, we are able to use memory-efficient first order and quasi-Newton methods.

Two key insights highlight the intuition behind the algorithm. First, for problems with compact search spaces there exists a global optimum of (19) of rank at most r^* where $\frac{r^*(r^*+1)}{2} \leq |\mathcal{S}|$ [6][7]. Thus, for $r^* \leq r$ the minima of (19) and (21) are equivalent. Second, it is commonly the case that a local minimum of (21) with $R \in \mathbb{R}^{2|\mathcal{S}|\times r}$ is either a global minimum or can be escaped by increasing the rank of the search space to $R \in \mathbb{R}^{2|\mathcal{S}|\times(r+1)}$. Conditions for when this escape property holds have been a subject of theoretical interest; recent results [8] show that for SDPs where the solution set is a smooth compact manifold, the set of problematic matrices W where local minima do not have an escape is of measure 0. Both of these insights suggest that one can use the flowchart in Figure 3 to solve the SDP, where the dashed box encloses a subroutine to find local minima which we explain next.

Problem (21) is an equality constrained non-linear programming problem which we opt to solve with the method of multipliers [9]. The method of multipliers enforces the trace condition $\mathbf{Tr}(M_s RR^T) = 1$ with an augmented Lagrangian, which utilizes dual variables λ and penalty factor σ to penalize infeasibility:

$$L(R,\lambda,\sigma) = \mathbf{Tr}(WRR^T) - \sum_{s=1}^{|\mathcal{S}|}\lambda_s(\mathbf{Tr}(M_sRR^T)-1) + \frac{\sigma}{2}\sum_{s=1}^{|\mathcal{S}|}(\mathbf{Tr}(M_sRR^T)-1)^2. \quad (22)$$

For an appropriate (λ^*, σ^*) pair, a minimizer of (22) will recover a local minimizer of (21) [10]. To identify that pair, we use the iterative method shown in Algorithm 1. It first uses a nonlinear solver to minimize the augmented Lagrangian for a fixed (λ, σ), then it updates the dual variable λ while sufficient progress is being made and increases the penalty factor σ as soon as progress stalls. In our implementation, We opt to use the limited memory BFGS algorithm to minimize the augmented Lagrangian, but in principle any nonlinear solver that converges to a local minimum can be used. Parameter $\eta < 1$ specifies the rate of desired progress while $\gamma > 1$ specifies how quickly the penalty rate grows. In practice, we chose $\gamma = 2$ and $\eta = .9$ because much of the information about infeasibility was encoded in the dual variables λ. These parameters prevented ill-conditioning from the penalization factor growing too rapidly and ensured that the dual variable could successfully guide the local search to the feasible region.

$\sigma = \sigma_0, \lambda_s = \mathbf{0}$;
$k = 0$;
$v_0 = \frac{\sigma}{2}\sum_{s=1}^{|\mathcal{S}|}(\mathbf{Tr}(M_s(R^k)(R^k)^T)-1)^2$;
while $|\nabla L(R,\lambda^k,\sigma^k)| > \epsilon$ *and* $\sigma_k < 2\times 10^6$ **do**
  if $v \leq \eta v_k$ **then**
    $\lambda_i^{k+1} = \lambda_i^k - \sigma_k(\mathbf{Tr}(M_s(R^k)(R^k)^T)-1)$ for all $s \in \mathcal{S}$;
    $\sigma_{k+1} = \sigma_k$;
    $v_{k+1} = v$;
  else
    $\lambda_i^{k+1} = y_i^k$ for all $s \in \mathcal{S}$;
    $\sigma_{k+1} = \gamma\sigma_k$;
    $v_{k+1} = v_k$;
  end
  R^{k+1} = Nonlinear Solver($R_{init} = R^k$);
  $k = k+1$
end

Algorithm 1: Method of multipliers update for dual variable λ and penalty factor σ.

V. Benchmarks

We present two examples of real-world networks that showcase the results of the Burer-Monteiro algorithm. The first example highlights the computational scalability of this problem while the second validates the model assumptions with a microscopic simulation. All benchmarks were performed on a 2013 Macbook Pro with 2.4GHz processor, 8GB memory, and with `python 2.7`.

The algorithms tested included "cold start" gradient descent for problem (16) before the coordinate transformation $z = (x, y)$. Lower bounds via the semidefinite relaxation are computed with `CVXOPT` [11] and the Burer-Monteiro method. The principal eigenvector from both algorithms are then used as "warm starts" for gradient descent. We also consider applying the method of multipliers to problem (21) without the rank update in the flowchart of Figure 3.

A. Manhattan

We demonstrate the scalability of the BM-algorithm by constructing a large synthetic network from `OpenStreetMap` data [12]. Our network is based on Manhattan in New York City and contains 1771 nodes and 3923 links. Green splits φ were assigned based on the orientation of the link as it enters the intersection in such a way that the green splits for North-South links are completely out of phase of East-West links. Travel times γ

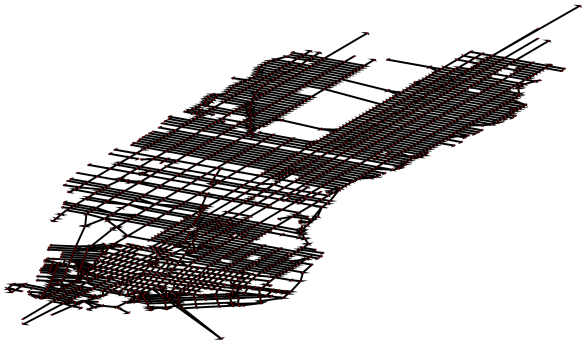

Fig. 4. Depiction of the Manhattan network.

Solver	Objective	Runtime (s)	Rank
Burer-Monteiro	-14895.66	347.63	3
CVXOPT	-	MemOut	-
Method of Mult. ($r = 1$)	-14888.02	145.02	-
Warm Start GD (BM)	-14887.26	421.25	-
Warm Start GD (CVXOPT)	-	-	-
Cold Start GD	-14834.58	275.90	-

TABLE I

MANHATTAN OFFSET OPTIMIZATION BENCHMARKS. FIELDS WITH DASHES ARE EITHER NOT COMPUTED OR CONSTANT DUE TO THE NATURE OF THE SOLVER.

Cycle Time (s)	60	70	80	90	120
Nodes	85	59	138	59	79
Queues	759	552	1365	562	887

TABLE II

OPTIMIZATION SUBPROBLEM SIZE FOR EACH CYCLE TIME.

Offsets	Total Objective	Total Runtime (s)	Total Rank
Pre-existing	-46561.95	-	-
Random	-2063.15	-	-
CVXOPT	-136767.71	3.14	188
Burer-Monteiro	-136767.70	0.004	5
Cold Start GD	-136265.80	51.55	-
Method of Mult. ($r = 1$)	-136767.70	0.004	-
Warm Start CVXOPT	-136767.71	11.07	-

TABLE III

PASADENA, CA OPTIMIZATION BENCHMARKS. TOTAL OBJECTIVE AND RANK COLUMNS ARE SUMS OF THE FIVE SUBPROBLEMS. PRE-EXISTING AND RANDOM OFFSETS ARE INCLUDED FOR COMPARISON.

are proportional to link length. Turn ratios β between links are chosen so that a vehicle is more likely to travel straight than turn. Let $A_{\mathcal{E}}$ be a vector with $A_l = 0$ for internal links while A_l is given by (2) for $l \in \mathcal{E}$. The mean traffic flow is computed with $A = (I - \beta)^{-1} A_{\mathcal{E}} = \sum_{i=0}^{\infty} \beta^i A_{\mathcal{E}}$. A is finite because turn ratios by construction guarantee that β is a stochastic matrix.

Table I showcases the attained objective values and runtimes for a suite of algorithms. The Burer-Monteiro algorithm certifies a lower bound on the optimization problem while `CVXOPT` ran out of memory. Note that the warm start BM runtime is the sum of the BM runtime and the nonlinear solver with a warmstart. The Burer-Monteiro algorithm returned a result that is infeasible for the original problem because it has rank higher than 1. We use the eigenvector associated with the principal eigenvalue as a warm start for a gradient descent algorithm. The cold start gradient descent method on the original problem (16) performs remarkably well but the runtime is longer than even the method of moments. We attribute this to the amount of time it takes to compute a gradient for (16) which involves a large summation along $\mathcal{L}$ for each component.

B. Microsimulation of Pasadena, California

The results from our offset optimization algorithm are also validated with a microscopic simulation of a network in Pasadena, California, USA depicted in Figure 5. Simulations were run with Aimsun, which is a transportation modeling software used by traffic practitioners to model networks and evaluate control strategies. The network contains 1537 nodes, of which 459 are signalized intersections. This network is under ongoing development at the Connected Corridors Program at the U.C. Berkeley Institute of Transportation Studies. The travel demands are obtained from area traffic studies between 2006 and 2014 and calibrated using in-built Aimsun routines. The geometric details of the network (lane counts, turn bays, etc) were manually verified against aerial imagery from Google Maps.

The microsimulation takes into consideration detailed parameters such as driver reaction time, vehicle-to-vehicle gaps, and vehicle dynamics. The Aimsun model includes features such as unsignalized intersections and multiple lanes within a road. Lanes can easily be accommodated by increasing the number of links in the network. The model in Section II assumes all intersections are signalized, so in the optimization problem all unsignalized intersections are ignored and the travel times between pairs of lane queues are updated to reflect the travel time of the shortest path between the queues plus any delay from stop signs.

There is not a single cycle time shared amongst all intersections, but 420 intersections have one of five cycle times: 60, 70, 80, 90, and 120 seconds. For each cycle time, the cost associated with a queue entering an intersection of a different cycle time are ignored. Table II shows how the problem size changes as different cycle times are considered. All right turn queues are ignored because vehicles in the microsimulation are allowed to turn right with a red light. An optimization routine is then run on the smaller network with the modified cost function and optimization variables only over the intersections with the chosen cycle time. The minimizing offsets for each cycle time are then concatenated to create a full 420 dimension offset vector.

All offset optimization subproblems took a few seconds and the summation of the objectives, runtimes, and rank for each of the five sub-problems is shown in Table III. The Burer-Monteiro method recovers the same value as `CVXOPT`, but with a feasible, rank 1 objective each time.

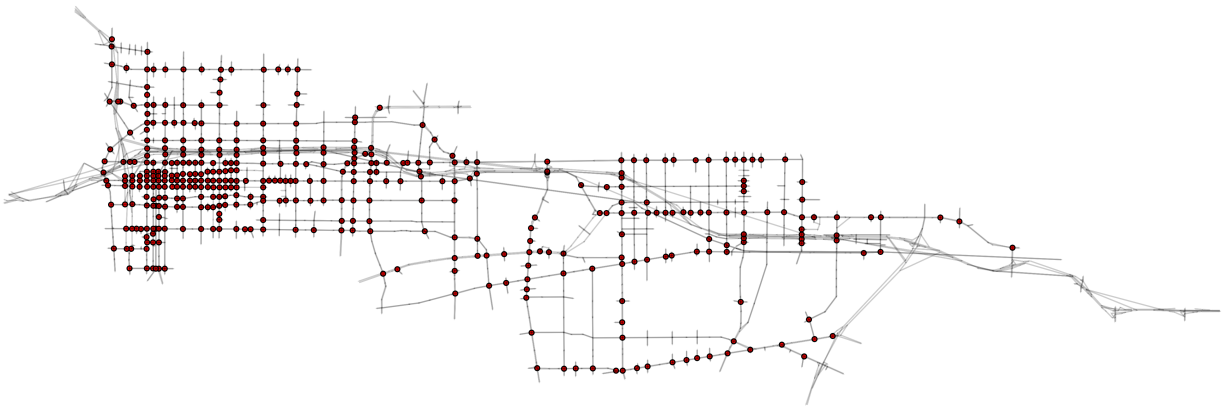

Fig. 5. Aimsun network of Pasadena, CA, USA. The 420 optimized intersections are highlighted with a dark red circle.

Metric	Random	Baseline	MoM	Improvement
Stop time $\left(\frac{s}{veh \cdot hr}\right)$	12.33	12.01	11.57	3.65%
Queue (veh)	348.8	337.4	327.9	2.80%
Flow (veh/hr)	169.9	170.5	171.4	.5%

TABLE IV

METRICS FOR DIFFERENT OFFSETS. EACH VALUE IS A MEAN AMONG ALL QUEUES THAT ARE CONTROLLED BY AN INTERSECTION WITH AN OPTIMIZED OFFSET. THE IMPROVEMENT COLUMN COMPARES THE OPTIMIZED AGAINST THE BASELINE OFFSETS.

Like the Manhattan network, the output from gradient descent with a random initialization is nearly optimal. We also included random offsets and pre-existing offsets from the Aimsun model as a comparison.

To validate the assumptions involved in the optimization formulation, we collected simulations metrics for the same random, pre-existing, and optimized (via the method of multipliers) offsets that appear in Table III. Table IV shows that both the stoppage time and the mean-queue length improve for the optimized versus the baseline version, despite the fact that there was a small net increase in the average flow experienced by the signalized intersections.

VI. CONCLUSION

We have applied the Burer-Monteiro method to the offset optimization problem for a large network of signalized intersections. The Manhattan example showcases the scalability of the BM algorithm. Meanwhile for the Pasadena, CA network a comparison among a suite of algorithms demonstrates that the BM method reliably finds a global optimum of the semidefinite relaxation. A detailed microsimulation of Pasadena, California demonstrates that the optimization routine yields tangible improvements in delay time and mean queue occupancy, in spite of modeling assumptions such as the global cycle time, fluid queue model, sinusoidal approximation, and absence of congestion.

The speedup offered by the BM method raises the intriguing possibility of realtime coordination of large traffic networks. Future work will investigate methods to accommodate traffic demand forecasts with stochastic uncertainty and to incorporate actuated signals with detectors.

VII. ACKNOWLEDGEMENTS

The authors would like to thank Benjamin Recht for suggesting the Burer-Monteiro method, Samuel Coogan for helpful discussions, and both Gabriel Gomes and California Partners for Advanced Transportation Technology for providing the Pasadena Aimsun network.

REFERENCES

[1] S. Burer and R. D. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, 2003.

[2] S. Coogan, G. Gomes, E. Kim, M. Arcak, and P. Varaiya, "Offset optimization for a network of signalized intersections via semidefinite relaxation," 2015.

[3] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[4] A. Singer, "Angular synchronization by eigenvectors and semidefinite programming," *Applied and computational harmonic analysis*, vol. 30, no. 1, pp. 20–36, 2011.

[5] A. Muralidharan, R. Pedarsani, and P. Varaiya, "Analysis of fixed-time control," *Transportation Research Part B: Methodological*, vol. 73, pp. 81–90, 2015.

[6] A. I. Barvinok, "Problems of distance geometry and convex properties of quadratic maps," *Discrete & Computational Geometry*, vol. 13, no. 2, pp. 189–202, 1995.

[7] G. Pataki, "On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues," *Mathematics of operations research*, vol. 23, no. 2, pp. 339–358, 1998.

[8] N. Boumal, V. Voroninski, and A. S. Bandeira, "The non-convex burer-monteiro approach works on smooth semidefinite programs," *CoRR*, vol. abs/1606.04970, 2016.

[9] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[10] R. Fletcher, *Practical Methods of Optimization; (2Nd Ed.)*. New York, NY, USA: Wiley-Interscience, 1987.

[11] M. S. Andersen, J. Dahl, and L. Vandenberghe, "Cvxopt: A python package for convex optimization, version 1.1. 6," 2013.

[12] OpenStreetMap. https://wiki.openstreetmap.org.