

Optimal Control of Linear Cost Networks

David Ohlin, Emma Tegling and Anders Rantzer

Abstract—We present a method for optimal control with respect to a linear cost function for positive linear systems with coupled input constraints. We show that the Bellman equation giving the optimal cost function and resulting sparse state feedback for these systems can be stated explicitly, with the solution given by a linear program. Our framework admits a range of network routing problems with underlying linear dynamics. These dynamics can be used to model traditional graph-theoretical problems like shortest path as a special case, but can also capture more complex behaviors. We provide an asynchronous and distributed value iteration algorithm for obtaining the optimal cost function and control law.

I. INTRODUCTION

In many real-world systems the quantities of interest, like amounts of goods or distributions of probabilities, are intrinsically positive. Such systems naturally lead to models that are positive systems, where the state x is confined to the positive orthant \mathbb{R}_+^n . In the context of optimal control, positive systems exhibit many advantageous properties, as reviewed in [1]. Of special note is that such systems admit linear Lyapunov functions. This guarantees positive stage costs for a linear cost function, giving significant computational advantages in applications where such a cost function can accurately capture the desired objective. In contrast, current methods for optimal control of general linear systems with quadratic cost [2] give rise to Riccati equations where the number of variables scales quadratically with the dimension of the state, as opposed to linearly. Additionally, as noted in [1], systems on this form allow for sparse optimal state feedback $u = Kx$, giving favorable scaling results for high-dimensional inputs.

In this paper we exploit these advantages for the purpose of optimal control of a system where the magnitude of combinations of input signals are jointly constrained by a linear function of the state. These coupled constraints generalize the previous work [3], wherein all inputs are fully actuated in either the positive or negative direction, and give rise to trade-offs between several different inputs. This novel formulation covers a range of problems excluded by the constraints in [3]. In particular, it provides a natural framework for a general class of network routing and shortest path-like problems with underlying linear dynamics unfolding on a graph. Our formulation, which uses linear costs for the state and input,

The authors are with the Department of Automatic Control and ELLIIT Strategic Research Area at Lund University, Lund, Sweden. Email: {david.ohlin, anders.rantzer, emma.tegling}@control.lth.se

This work is partially funded by Wallenberg AI, Autonomous Systems and Software Program (WASP) and the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme under grant agreement No 834142 (ScalableControl).

allows for an explicit solution to the associated Bellman equation, just like in the familiar quadratic cost case. We also provide a method for obtaining the optimal cost function $J^*(x)$ by means of linear programming.

This paper focuses on control problems over networks, in which physical constraints and scalability concerns in applications typically call for distributed solution methods. We give a distributed implementation of the value iteration algorithm for the proposed problem formulation, similar to those treated in [4], and show that convergence results presented in [5] can be applied to the present setting. This allows for distributed and asynchronous computation of the local optimal state feedback, wherein nodes need only share their local estimates of the cost function and the cost of the optimal control action. An extended analysis of the problem setup in [3], leveraging established theory of Dynamic Programming and extending the problem to constraints based on other norms, can be found in [6], wherein upper bounds on the convergence rate of policy iteration for such systems are given. That analysis, however, also fails to capture the class of network routing problems treated here.

The formulation we present connects results from the field of optimal control to a rich history of works (e.g. [7], [8]) on graphs with finite state and action spaces, which becomes a special case of the problem as stated here. Our framework is, however, flexible enough to cover a wider range of problems with continuous state and action spaces. Notably, interpreting the continuous state as a probability distribution allows for the modeling of Markov Decision Processes (MDPs) as another special case where existing results on convergence of methods like policy iteration [9] can be applied. Most works in the related literature that treat dynamic versions of the shortest path problem use models where the structure of the graph changes [10]. Similar problems with dynamics in a continuous state space evolving on a graph structure are, however, relatively unexplored.

In the next section we present relevant theory and define notation. Section III contains the problem statement, required assumptions and main result in the form of Theorem 1. In Section IV, an algorithm is given for distributed and asynchronous value iteration that finds the fixed point of the operator given in Theorem 1, corresponding to the optimal cost function. The results are illustrated by an example in Section V, with application to a simplified model of a large-scale cooling system. Finally, Section VI presents conclusions and sets out possible avenues for future study.

A. Notation

Inequalities are applied element-wise for matrices and vectors throughout. Further, the notation \mathbb{R}_+^n is used to denote the closed nonnegative orthant of dimension n . The operator $\min\{A, 0\}$ extracts the minimum element of A , yielding zero if A has no negative elements. Let $\text{diag}(a)$ denote a square matrix with the elements of the vector a along the diagonal. The expressions $\mathbf{1}_{p \times q}$ and $\mathbf{0}_{p \times q}$ signify a matrix of ones or zeros, respectively, of the indicated dimension, with subscript omitted when the size is clear from the context. If the dimension is zero, this is to be interpreted as the empty matrix.

B. Problem setup

Consider the infinite-horizon optimal control problem

$$\begin{aligned} \text{Minimize} \quad & \sum_{t=0}^{\infty} [s^\top x(t) + r^\top u(t)] \text{ over } \{u(t)\}_{t=0}^{\infty} \\ \text{subject to} \quad & x(t+1) = Ax(t) + Bu(t) \\ & u(t) \geq 0, \quad x(0) = x_0 \\ & \mathbf{1}^\top u_1(t) \leq E_1^\top x(t) \\ & \quad \vdots \\ & \mathbf{1}^\top u_M(t) \leq E_M^\top x(t) \end{aligned} \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$ and $B = [B_1 \cdots B_M] \in \mathbb{R}^{n \times m}$ with $B_i \in \mathbb{R}^{n \times m_i}$ define the linear dynamics. The input $u \in \mathbb{R}^m$ is partitioned into M subvectors u_i , each containing m_i elements, so that $m = \sum_{i=1}^M m_i$. In the upcoming treatment, we will let the matrices B_i be further subdivided into individual columns as $B_i = [B_{i1} \cdots B_{im_i}]$ with $B_{ij} \in \mathbb{R}^n$. The costs connected to states and actions are $s \in \mathbb{R}_{>0}^n$ and $r \in \mathbb{R}_{\geq 0}^m$ with $r_i \in \mathbb{R}^{m_i}$ following the partition of u , and r_{ij} denoting the j th element of r_i . The constraints on the input $u(t)$ are given by $E = [E_1 \cdots E_M]^\top \in \mathbb{R}_{>0}^{M \times n}$.

Solving the problem (1) is equivalent to finding some optimal value function $J^*(x)$ that satisfies the corresponding Bellman equation

$$J^*(x) = \min_{u \in U(x)} [g(x, u) + J^*(Ax + Bu)]. \quad (2)$$

In the specific context of (1) the immediate cost is given by the value at each instant t of the cost function to be minimized, so $g(x(t), u(t)) = s^\top x(t) + r^\top u(t)$. Note that stabilizability of the system in (1) is a necessary condition for the problem to have a finite value. In the previous work [3] the Bellman equation (2) has an optimal linear solution $J^*(x) = p^\top x$ with $p \in \mathbb{R}^n$ for a certain class of positive linear systems. Our main result below proves that similarly favorable results can be obtained for systems on the form (1), which have a natural interpretation as network routing problems, but are not in general admissible in [3].

In anticipation of our main result, we introduce two assumptions on the dynamics of the system in (1).

Assumption 1: Any negative element of B_i is located on the i th row.

Assumption 2: The matrices A , B and E satisfy the element-wise inequality $A \geq |B^- E| \geq \mathbf{0}$ where

$$B^- = [\text{diag}([\min\{B_1, 0\} \cdots \min\{B_M, 0\}]) \mathbf{0}_{M \times (n-M)}]^\top.$$

Assumptions 1 and 2 guarantee positivity of the dynamics in (1). While this may seem restricting, the partition of u is natural in many relevant applications. In the case of a network problem where the inputs u correspond to quantities transferred over edges in the network, Assumption 1 requires that all edges leading from node i be included in the same constraint partition u_i . Since positivity of the system is often motivated by states corresponding to real quantities, Assumptions 1 and 2 mean that no more than the quantity already in node i at time t (or expected to be within one time step) can be transferred from it.

These assumptions are sufficient to ensure invariance of \mathbb{R}_+^n in the general case, but in special cases where parts of the state space are unreachable they may be relaxed. In such cases it is, however, possible to remodel the system in compliance with the assumptions, which can therefore be made without loss of generality. Given these assumptions on the matrices of the problem (1), we are now ready to state our main result:

Theorem 1: Under Assumptions 1 and 2, the following three statements are equivalent:

- (i) The problem (1) has a finite value for every $x_0 \in \mathbb{R}_+^n$.
- (ii) There exists $p \in \mathbb{R}_+^n$ satisfying the equation

$$p = s + A^\top p + \sum_{i=1}^M \min\{r_i + B_i^\top p, 0\} E_i. \quad (3)$$

- (iii) The value of the linear program

$$\begin{aligned} \text{Maximize} \quad & \mathbf{1}^\top p \text{ over } p \in \mathbb{R}_+^n \\ \text{subject to} \quad & p \leq s + A^\top p - \sum_{i=1}^M z_i E_i \\ & z_i \geq r_{ij} + B_{ij}^\top p \text{ for } j = 1, \dots, m_i \\ & z_i \geq 0 \end{aligned}$$

is bounded and has solution p satisfying (3).

Given the existence of a p as in (ii), the minimal value of (1) is $p^\top x_0$. The optimal linear state feedback law is then given by $u_i(t) = K_i x(t)$ with

$$K_i := \begin{bmatrix} \mathbf{0}_{j-1 \times n} \\ E_i^\top \\ \mathbf{0}_{m_i - j \times n} \end{bmatrix} \text{ for } i = 1, \dots, M, \quad (4)$$

where the vector E_i^\top enters at the j th row with j being the index of the minimal element of $r_i + B_i^\top p$, provided it is negative. If all elements are nonnegative then $K_i = \mathbf{0}_{m_i \times n}$.

Proof: We first show that \mathbb{R}_+^n is invariant under the dynamics in (1). It holds that

$$Ax + Bu = Ax + [B_1 \ \cdots \ B_M] \begin{bmatrix} u_1 \\ \vdots \\ u_M \end{bmatrix} = Ax + \sum_{i=1}^M B_i u_i.$$

As a consequence of Assumption 1 and the constraints given in (1) this expression has a lower bound given by

$$\begin{aligned} Ax + \sum_{i=1}^M B_i u_i &\geq Ax + \begin{bmatrix} \min\{B_1, 0\}E_1^\top \\ \vdots \\ \min\{B_M, 0\}E_M^\top \\ \mathbf{0}_{(n-M) \times n} \end{bmatrix} x \\ &= (A + B^- E)x \end{aligned}$$

recalling that $\min\{B_i, 0\}$ denotes the minimum element of B_i , yielding zero if all elements are nonnegative. This final expression is contained in \mathbb{R}_+^n due to Assumption 2.

Next, let $J_k(x) = p_k^\top x$. Given the ansatz $p_0 = 0$, we compute each subsequent iterate p_{k+1} by applying the Bellman equation

$$\begin{aligned} &\min_{u \in U(x)} [g(x, u) + J_k(f(x, u))] \\ &= \min_{u \in U(x)} [s^\top x + r^\top u + p_k^\top (Ax + Bu)] \\ &= s^\top x + p_k^\top Ax + \min_{u \in U(x)} [(r + B^\top p_k)^\top u] \end{aligned}$$

where $U(x)$ denotes the set of all inputs u satisfying the constraints in (1). We partition the vector $r + B^\top p_k$ in the same fashion as u into M subvectors $r_i + B_i^\top p_k \in \mathbb{R}^{m_i}$. Since each subvector u_i is only bound by one of the upper input constraints, we can minimize the expression above independently for each u_i :

$$\begin{aligned} \min_{u \in U(x)} (r + B^\top p_k)^\top u &= \sum_{i=1}^M \min_{\substack{\mathbf{1}^\top u_i \leq E_i^\top x \\ u_i \geq 0}} (r_i + B_i^\top p_k)^\top u_i \\ &= \sum_{i=1}^M \min\{r_i + B_i^\top p_k, 0\} E_i^\top x \end{aligned} \quad (5)$$

where the minimum in the final expression is taken over all elements in the vector $r_i + B_i^\top p_k$ and 0. The Bellman iteration becomes

$$\begin{aligned} &\min_{u \in U(x)} [g(x, u) + J_k(f(x, u))] \\ &= s^\top x + p_k^\top Ax + \sum_{i=1}^M \min\{r_i + B_i^\top p_k, 0\} E_i^\top x \\ &= \left(s + A^\top p_k + \sum_{i=1}^M \min\{(r_i + B_i^\top p_k, 0) E_i\} \right)^\top x \\ &= p_{k+1}^\top x. \end{aligned}$$

As a consequence, we have an increasing sequence of functions generated by iteration of the Bellman equation, and the cost functions all take the form $J_k(x) = p_k^\top x$.

Given that the problem (1) has a finite value according to (i), the sequence of value functions generated by this iteration has the upper limit $J^*(x) = p^\top x$. This limit fulfills the equation in (ii). The converse also holds since, given that a p satisfying (ii) exists, the increasing sequence of $\{p_k\}_{k=0}^\infty$ must satisfy $p_k \leq p$ for all k . This means that the Bellman equation has a nonnegative and finite solution, implying that (i) holds. To find the optimal controller we seek the policy $\mu(x)$ minimizing the Bellman equation for the optimal cost function $J^*(x) = p^\top x$. Taking the expression from (5) we get

$$\mu(x) = \arg \min_{u \in U(x)} (r + B^\top p)^\top u$$

which, partitioned like the minimization in (5), can be performed separately for each subvector μ_i yielding the expression

$$\mu_i(x) = \underbrace{\begin{bmatrix} \mathbf{0}_{j-1 \times n} \\ E_i^\top \\ \mathbf{0}_{m_i-j \times n} \end{bmatrix}}_{=K_i} x.$$

where the nonzero row vector E_i^\top enters on the j th row and $j \in \{1, \dots, m_i\}$ is the index of the largest in magnitude negative element of $r_i + B_i^\top p$. If all elements are positive then $\mu_i(x) = \mathbf{0}_{m_i \times n}$ is the minimizing argument. As a result, the state feedback law for each partition u_i is stationary and given by $u_i(t) = K_i x(t)$.

Finally, we address (iii) in the same way as treated in [3]. If (ii) holds then the p fulfilling Equation 3 clearly also solves the program

$$\begin{aligned} &\text{Maximize } \mathbf{1}^\top p \text{ over } p \in \mathbb{R}_+^n \\ &\text{subject to } p \leq s + A^\top p + \sum_{i=1}^M \min\{r_i + B_i^\top p, 0\} E_i \end{aligned}$$

which is equivalent to the linear program in (iii). Any solution fulfilling the Bellman inequality gives a lower bound on the corresponding optimal cost function. Thus the solution of the linear program must also be the optimal cost function and, as a consequence, (ii) implies (iii). To show the converse, assume that the linear program in (iii) is bounded by some argument p_k . Calculate

$$p_{k+1} = s + A^\top p_k + \sum_{i=1}^M \min\{r_i + B_i^\top p_k, 0\} E_i.$$

We now have $p_{k+1} \geq p_k$ and as a consequence

$$\begin{aligned} p_{k+1} &= s + A^\top p_k + \sum_{i=1}^M \min\{r_i + B_i^\top p_k, 0\} E_i \\ &= \min_{u \in U(x)} [s^\top x + r^\top u + p_k^\top (Ax + Bu)] \\ &\leq \min_{u \in U(x)} [s^\top x + r^\top u + p_{k+1}^\top (Ax + Bu)] \\ &= s + A^\top p_{k+1} + \sum_{i=1}^M \min\{r_i + B_i^\top p_{k+1}, 0\} E_i \end{aligned}$$

which is exactly the constraint in the reformulated program above. Since we assumed that p_k attains the maximum of the program and $p_{k+1} \geq p_k$, the only possible conclusion is that $p_{k+1} = p_k$. Thus the maximizing argument of the program in (iii), if it is bounded, also solves Equation 3, implying (ii). This concludes the proof. ■

To illustrate the above theorem we apply it to a simple shortest-path problem. A detailed example showing more general possibilities is presented in Section V.

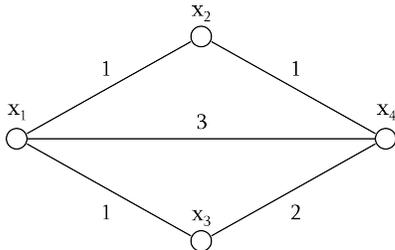


Fig. 1: Weighted graph in Example 1.

Example 1: Consider the (trivial) problem of finding the shortest path from node x_1 to node x_4 in Figure 1. This can be formulated as an optimal control problem on the form (1) with $n = 4$, $m = 10$ (one input for each direction of traversal of the edges) in the following way: Let the states x_i correspond to the nodes, select $E = A = I$ and let B be a node-link incidence matrix:

$$B = \begin{bmatrix} -1 & -1 & -1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & -1 & -1 & -1 \end{bmatrix}.$$

We partition the input vector $u = [u_1^\top u_2^\top u_3^\top u_4^\top]^\top$ to collect all actions corresponding to directed edges originating in node i into u_i . The weights are selected as

$$s^\top = [1 \ 1 \ 1 \ 0], \quad r^\top = [0 \ 0 \ 2 \ 0 \ 0 \ 1 \ 0 \ 2 \ 0 \ 1]. \quad (6)$$

We now have the problem on the form required by Theorem 1 and finding a p satisfying (3) by fixed point iteration we get the optimal cost function as

$$J^*(x) = [2 \ 1 \ 2 \ 0]^\top x.$$

The optimal control law is given by (4) which yields the optimal routing in each node.

We can easily verify that Assumption 1 and Assumption 2 are satisfied by the above choice of A , B and E . The shortest path problem assigns cost only to traversal of the edges, while the formulation (1) also associates a cost s_i with each state x_i and allows for the possibility of taking no action ($u = 0$). However, since the dynamics are not asymptotically stable ($A = I$), the optimal policy will never be to remain in i as long as the cost s_i is strictly positive. This means the cost s_i will be incurred once per edge traversal in the optimal policy. By reallocating some of the cost from the edges to all nodes except the endpoint x_4 (from r to s), we guarantee that the optimal policy will correspond to the shortest path. The weights (6) result in the same cost as in the

original formulation for all paths that do not stay more than one time step in each node, i.e. all paths that are possible candidates for the optimal solution.

IV. DISTRIBUTED IMPLEMENTATION

As the dimensions of the state and input spaces grow in (1), it is desirable that the complexity of the solution method scales favorably. One method for finding the optimal control of (1) is to solve the linear program in Theorem 1. The number of parameters scales linearly with the dimension of the state and the program is thus computationally tractable even for high-dimensional problems. In many applications, however, global communication and access to information is not feasible. An alternative method is to distribute the the solution of the problem, requiring only local communication and knowledge of the local dynamics. Below we present an algorithm for distributed and asynchronous calculation of the optimal cost function for agents with limited knowledge of the global state.

Consider the dynamics (1) when each state x_i belongs to a separate agent i with its own constraint $\mathbf{1}^\top u_i \leq E_i^\top x$, so that $M = n$. The optimal cost function $J^*(x) = p^\top x$ with p fulfilling (3) can be found by asynchronous distributed value iteration starting from the ansatz $p_0 = 0$. Agent i affects other agents through either the dynamics A , the constraints E or the inputs governed by the structure of B . Here, let $W^{(i)} \in \mathbb{R}^{n \times n}$ denote the local incidence matrix of node i given by $W^{(i)} = B_i B_i^\top$. We define a set of neighbors of i for each of these interactions:

$$\begin{aligned} \mathcal{N}_i^A &:= \{j : A_{ji} \neq 0, j \neq i\} \\ \mathcal{N}_i^E &:= \{j : E_{ji} \neq 0, j \neq i\} \\ \mathcal{N}_i^B &:= \{j : W_{ij}^{(i)} \neq 0, j \neq i\}. \end{aligned}$$

As a consequence of Assumption 2 we have $\mathcal{N}_i^E \subseteq \mathcal{N}_i^A$ with equality for the choice $E = A$. Note that these sets are defined to include only out-neighbors of i . Next, index the columns of A as $A = [A_1 \ \dots \ A_n]$ and let each agent i have local estimates $\hat{p}^{(i)}, \hat{q}^{(i)} \in \mathbb{R}$. Further, let $\hat{p} = [\hat{p}^{(1)} \ \dots \ \hat{p}^{(n)}]^\top$ denote the vector containing all agents' estimates of the local value function. We have shown in the proof of Theorem 1 that (3) has a solution p in \mathbb{R}_+^n . The positive orthant is invariant under iteration of the Bellman equation (2). As a consequence, if p satisfying (3) exists, then value iteration can be performed in a distributed fashion following Algorithm 1 with local estimates converging to the optimal cost function [5, Proposition 2.6.2], under the additional assumption that all agents update their estimate an infinite number of times as $t \rightarrow \infty$. Note that steps 8 and 9 can (with slight abuse of notation) be evaluated despite agent i not having access to the full vector \hat{p} , since the unknown terms in both cases are multiplied by zero. As stated in Algorithm 1, each agent stores only local estimates $\hat{p}^{(i)}$ and $\hat{q}^{(i)}$ which converge to the fixed point corresponding to the solution of (3). In order to extract the optimal policy, agents would additionally need to store the index of the element that minimizes the expression in step 5.

Algorithm 1 Asynchronous value iteration

- 1: local estimates $\hat{p}^{(i)}, \hat{q}^{(i)} \leftarrow 0$ for $i = 1, \dots, n$
 - 2: local terminating conditions $c_i \leftarrow 1$ for $i = 1, \dots, n$
 - 3: **while** $\exists i$ such that $c_i = 1$ **do**
 - 4: sample agent $i \in \{1, \dots, n\}$
 - 5: $\hat{p}_{\text{old}}^{(i)} \leftarrow \hat{p}^{(i)}$
 - 6: receive $\hat{p}^{(j)}$ from each neighbor $j \in \mathcal{N}_i^A \cup \mathcal{N}_i^B$
 - 7: receive $\hat{q}^{(j)}$ from each neighbor $j \in \mathcal{N}_i^E$
 - 8: $\hat{q}^{(i)} \leftarrow \min\{r_i + B_i^\top \hat{p}, 0\}$
 - 9: $\hat{p}^{(i)} \leftarrow s_i + A_i^\top \hat{p} + \hat{q}^{(i)} E_{ii} + \sum_{j \in \mathcal{N}_i^E} \hat{q}^{(j)} E_{ji}$
 - 10: **if** $\hat{p}^{(i)} = \hat{p}_{\text{old}}^{(i)}$ **then**
 - 11: $c_i \leftarrow 0$
 - 12: **else**
 - 13: $c_i \leftarrow 1$
 - 14: **end if**
 - 15: **end while**
-

To perform Algorithm 1, agent i needs to store and access, in addition to the sets of in- and out-neighbors, only the following: A_{ji} for $j \in \mathcal{N}_i^A$, E_{ji} for $j \in \mathcal{N}_i^E$ and B_i . From the perspective of privacy we note that agent i requires only the quantities \hat{p}_j and $\hat{q}_j = \min\{r_j + B_j^\top \hat{p}, 0\}$ from its neighbors, which contains no information about which action attains the minimum. Agents are not required to reveal their optimal action in order to perform the algorithm.

Remark 1: The algorithm as stated here does not include a truly distributed terminating condition. In settings where the optimal controller is calculated online and system parameters are potentially subject to change, it is realistic to run the algorithm indefinitely. Various methods of termination detection for distributed and asynchronous algorithms exist dating back to the seminal work [11], but applying these to the results presented here in a computationally efficient manner is a non-trivial problem in its own right and beyond the scope of this paper.

V. EXAMPLE: LINEAR FLOW DYNAMICS

We now present a more intricate, yet physically motivated, example to demonstrate some of the nuances that can be captured by the presented framework beyond the simpler shortest path problem. In the following example an optimal controller is derived for a system where endpoints are not specified a priori and the routing problem is complicated by the underlying dynamics of the network.

Example 2: Consider a simplified liquid cooling system serving some type of large-scale industrial plant. The system is represented by a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where the set of links \mathcal{E} models travel across the pipes that transport the liquid coolant between nodes \mathcal{V} . Each node $i \in \mathcal{V}$ corresponds to a different location in the plant and has an associated state x_i that specifies the heat contained there. The dynamics are given by (1) with $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$. Let A_{ii} model the loss of heat at location i due to both dissipation and diffusion to other states, while the off-diagonal elements $A_{ij}, j \neq i$ model the heat diffusing from i to other states j . For all

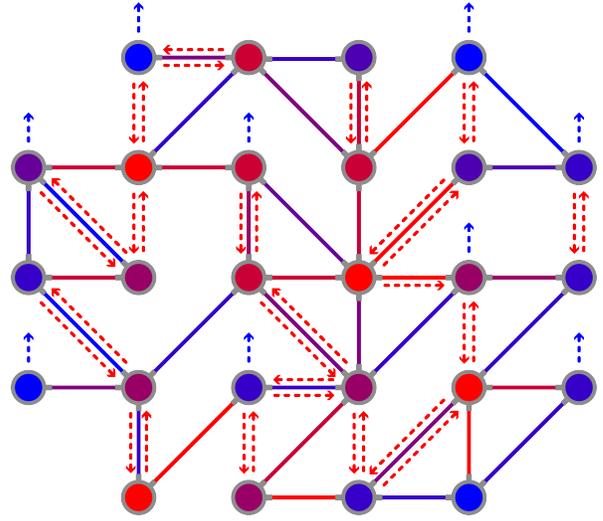


Fig. 2: Graph illustrating the cooling system in the example. The coloring of nodes and edges represent the cost vectors s and r , ranging from bright blue, corresponding to a cost of 0.2 per unit of heat transported, to bright red, corresponding to a cost of 1. Intermediate hues represent costs on that interval. Dashed red lines signify diffusion of heat between nodes and dashed blue lines represent dissipation.

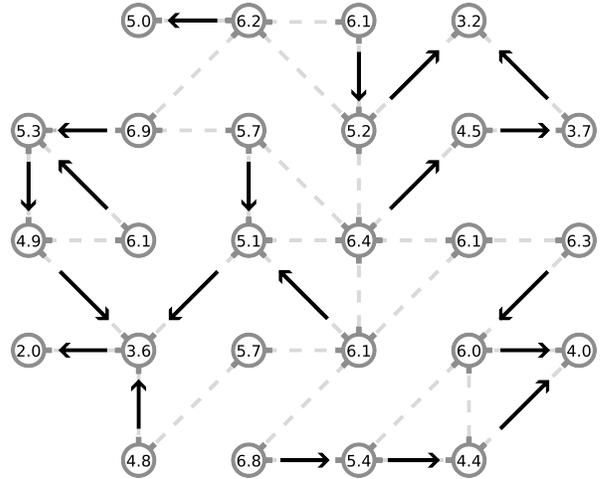


Fig. 3: The optimal cost function $J^*(x) = p^\top x$ given by Theorem 1 with each node showing the corresponding element of p . Dashed lines indicate the possible inputs (in both directions) and solid arrows show the direction of the optimal stationary control policy.

states we have $\sum_j A_{ji} \leq 1$, where equality means that no dissipation occurs in state i .

In each node i , the heat can be transferred by selecting the input vector u_i , dictating how much heat is transported across each of the links e_{ji} leading from i . The amount transferred is limited by the current amount present in state x_i , giving the constraint $1^\top u_i \leq E_i^\top x$ where $E_{ii} = A_{ii}$ and $E_{ij} = 0$ for $i \neq j$. The matrix B has one column corresponding to each directed link e_{ij} (from j to i), in which the j th element is -1 and the i th is a number in the interval $[0.95, 1)$ modeling

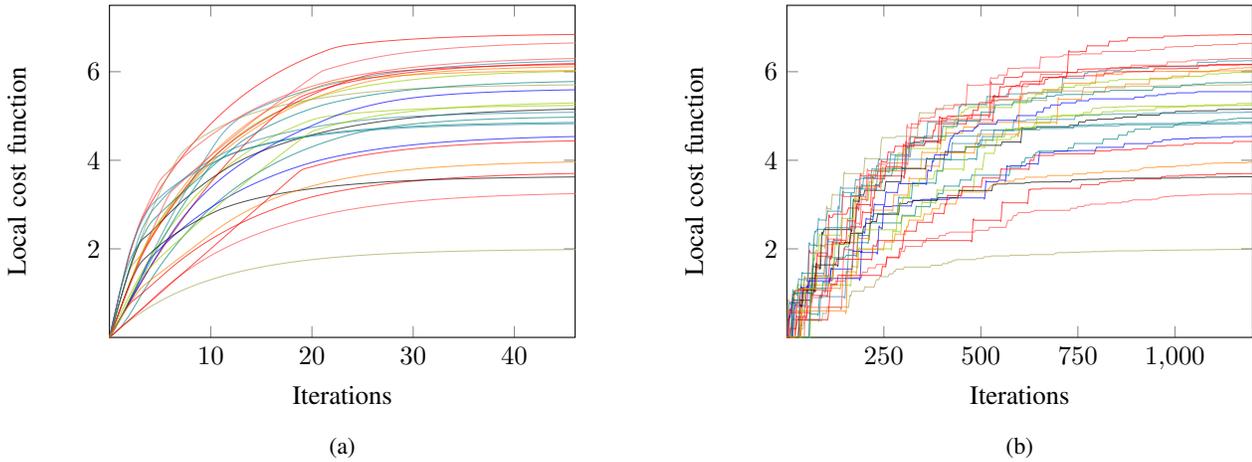


Fig. 4: (a) Elements of the cost function iterate p_k evolving under fixed point iteration of (3), starting from $p_0 = \mathbf{0}$. Each element corresponds to a node in Example 2. (b) Local cost function estimates $\hat{p}^{(i)}$ when performing distributed value iteration according to Algorithm 1 for the nodes in Example 2. Note that the apparent difference in convergence rate compared to Figure 4a is a result of only updating one node in each iteration of Algorithm 1. Scaling the axis of iterations by a factor $n = 26$ illustrates that the rate of convergence for the two methods is similar.

the heat loss over the pipe. The weights r and s signify the relative importance of cooling the different sections of the plant. Since each pipe is represented by two links e_{ij} and e_{ji} , the weights and heat dissipation for the two directions are set to be identical.

Figure 2 illustrates the dynamics of an example system with $n = 26$. The coloring of the nodes and links represent the relative costs r and s of transporting and storing one unit of heat respectively. The stationary control law found by applying Theorem 1 is illustrated in Figure 3 by indicating the optimal routing of heat across the graph. In any node i that has no arrow originating in it, the optimal control is $u_i = \mathbf{0}$, letting the heat diffuse and dissipate passively. As the dissipation only occurs in certain nodes, all heat must necessarily be transported to these either through passive diffusion or active heat transfer in order to obtain a finite value of the optimization problem (1).

Figure 4a shows the elements of p_k when applying value iteration (fixed point iteration of (3)) to find p for the above problem, with initialization $p_0 = \mathbf{0}$. In Figure 4b the evolution of the local cost estimates $\hat{p}^{(i)}$ when performing Algorithm 1 for the same problem are displayed. Both methods converge to the fixed point corresponding to the optimal cost function in Figure 3 and the rate of convergence differs approximately by a factor $n = 26$, since only one agent updates their state in each iteration of Algorithm 1. In other words, the figures show both methods converging after approximately the same number of agent updates.

VI. CONCLUSION

We have presented a framework for calculating optimal control of positive network dynamics in a scalable way. The resulting controller is a state feedback law $u(t) = Kx(t)$ where K is sparse given some inherent sparsity of the

dynamics. Furthermore, we have given a distributed and asynchronous algorithm for finding the optimal cost function and corresponding controller. This framework admits a wide range of network problems, both well and lesser studied. This indicates a possible direction for further exploration in adapting tools developed on shortest path problems and Markov decision processes for use in optimal control.

REFERENCES

- [1] A. Rantzer and M. E. Valcher, "A tutorial on positive systems and large scale control," in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 3686–3697.
- [2] D. P. Bertsekas, "Stable optimal control and semicontractive dynamic programming," *SIAM Journal on Control and Optimization*, vol. 56, no. 1, pp. 231–252, 2018.
- [3] A. Rantzer, "Explicit solution to Bellman equation for positive systems with linear cost," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 6154–6155.
- [4] D. Bertsekas, "Multiagent value iteration algorithms in dynamic programming and reinforcement learning," *Results in Control and Optimization*, vol. 1, p. 100003, 2020.
- [5] —, *Abstract Dynamic Programming: 3rd Edition*, ser. Athena scientific optimization and computation series. Athena Scientific, 2022.
- [6] Y. Li and A. Rantzer, "Exact dynamic programming for positive systems with linear optimal cost," arXiv: 2306.02072 [math.OC], 2023.
- [7] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, p. 269–271, 1959.
- [8] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [9] Y. Ye, "The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate," *Mathematics of Operations Research*, vol. 36, no. 4, pp. 593–603, 2011.
- [10] J. Gao, Q. Zhao, W. Ren, A. Swami, R. Ramanathan, and A. Bar-Noy, "Dynamic shortest path algorithms for hypergraphs," in *2012 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2012, pp. 238–245.
- [11] S.-T. Huang, "Termination detection by using distributed snapshots," *Information Processing Letters*, vol. 32, no. 3, pp. 113–119, 1989.