Network Optimization Visualizations

Dan Calderone

***Abstract*—This tutorial details matrix notation and visualizations for network flow and optimization problems. We review standard incidence matrix notation and affine and vertex constraint representations for graph flow problems including routing and cycle problems. We then present primal and dual linear programming formulations of shortest path problems along with variable interpretations. Detailed visualizations of the geometry of these problems is provided throughout.**

I. Introduction

Algebraic graph theory has become a staple modern engineering problems. In this tutorial paper, we review basic algebraic grapth theory constructions with references and visualizations. We then present mass flow constraint formulations standard to many modern optimization problems as well as linear programming formulations associated with shortest path problems. We note that while the visualizations in this paper are more extensive and thorough than normal, virtually none of the mathematical content is original; we have sought to provide the appropriate references throughout.

Markov decision process have become a staple of modern machine learning problems. In this tutorial paper, we collect a matrix notation for Markov decision processes, present the well known MDP linear programming formulations in terms of this notation, and provide a thorough set of visualizations of both the primal and dual linear programs. The matrix notation also allows us to make connections with the underlying graph structure of the MDP and several algebraic graph theory constructions (most notably the node-edge incidence matrix of a directed graph).

Markov decision processes (MDPs) are a staple of modern machine learning and control theory used for modeling discrete decision processes with a discrete state space. In this paper, we review and consolidate the excellent matrix notation for MDPs presented in [1] and make connections with standard algebraic graph theory constructions (most notably the node-edge incidence matrix of a directed graph.) We present well-known linear programming formulations of MDPs in the infinite horizon (average-reward and discounted reward) and finite horizon (total reward) settings [2] in both their primal and dual forms. Exposition, compact proofs, and minor extensions are offered throughout. Extensive visualizations are included throughout.[1]

The paper is organized as follows. In Section II, we define notational preliminaries and present several illustrative examples of our visualization techniques. (For more thorough explanation of the visualization techniques, we suggest the linear programming tutorial offered by these authors as well [**?**].)

In Section **??**, we present notation and visualizations for stochastic transition kernels, policies, and the resulting Markov chains. We also relate these concepts algebraically to the underlying graph structure. In Section **??**, we present the infinite horizon, average reward LP formulation of an MDP and provide visualizations. In Section **??**, we examine the effect of a discount factor on the transition kernel, present the infinite horizon, discounted reward LP formulation of an MDP and provide visualizations. In Section **??**, we present the finite horizon, total (and discounted) reward LP formulations of an MDP and provide visualizations. Appendix **??** contains a table summarizing notation for reference.

This paper assumes the knowledge and notation in the following monographs.

- Vector visualizations
- Matrix column geometry
- Linear programming geometry

II. Notation and Visualization techniques

A. Basic Notation

Let $I_{n\times n}$ refer to the identity of dimension n, or usually just I in context. Let A_j refer to the jth column of the matrix $A \in \mathbb{R}^{m\times n}$ and $\bar{A}_i$ refer to the ith row of the matrix

$$A = \begin{bmatrix} | & & | \\ A_1 & \cdots & A_n \\ | & & | \end{bmatrix} = \begin{bmatrix} - & \bar{A}_1 & - \\ & \vdots & \\ - & \bar{A}_m & - \end{bmatrix}$$

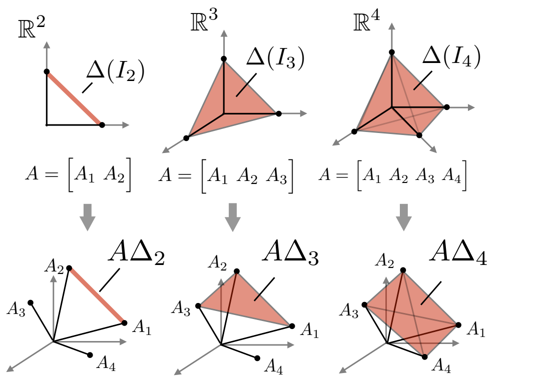

Fig. 1: Visualizations of simplicies in $\mathbb{R}^2$, $\mathbb{R}^3$, and $\mathbb{R}^4$ mapping to the convex hull of A.

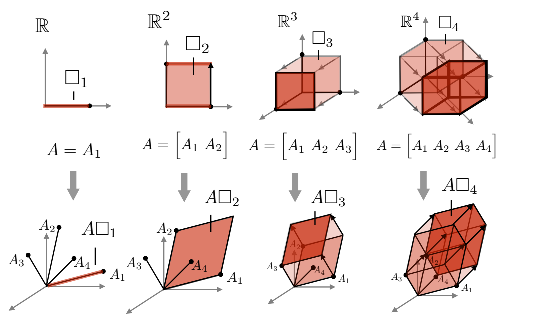

Fig. 2: Visualizations of unit cubes in $\mathbb{R}^2$, $\mathbb{R}^3$, and $\mathbb{R}^4$ mapping through A

Applying this to the identity gives I_j to be the jth standard basis vector and $A_j = AI_j$.

Let Δ_n represent the simplex of dimension n.

$$\Delta_n = \left\{x \in \mathbb{R}^n \mid \mathbf{1}^T x = 1,\ x \geq 0\right\}$$

where $\mathbf{1}$ is the vector of all 1's of the appropriate length. We will use $A\Delta_n$ (or just $A\Delta$ with the dimensions determined in context) to represent the convex hull of the columns of $A \in \mathbb{R}^{m \times n}$

$$A\Delta_n = \left\{y \in \mathbb{R}^m \mid y = Ax,\ x \in \Delta_n\right\}$$

. Similarly, let $\square_n$ represent the n-dimensional unit cube.

$$\square_n = \left\{x \in \mathbb{R}^n \mid \mathbf{0} \leq x \leq \mathbf{1}\right\}$$

and let $A\square_n$ (or, again, $A\square$ with n determined by context) be the set

$$A\square_n = \left\{y \in \mathbb{R}^m \mid y = Ax,\ x \in \square_n\right\}$$

. These sets are visualized in Fig. **??** & 2.

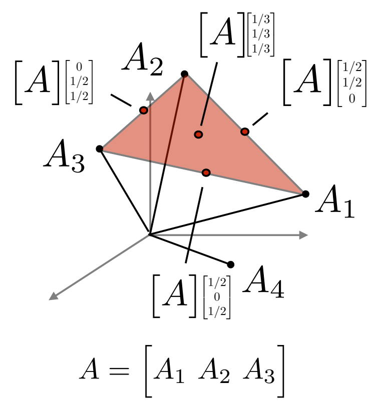

Fig. 3: Specific points labeled in $A\Delta$ for Δ_3.

B. Visualization Techniques

Fig. 4 illustrates more specifically how several key points in Δ_n map to points in the convex hull of A. We use this method for visualization convex combinations quite extensively throughout this paper.

Traditionally, vectors are visualized in as arrows in 2 or 3-dimensional spaces. If we think of each element of a vector as giving a displacement in a particular axis direction, we can draw 2D projections of higher dimensional vectors by simply assigning directions in 2D for each axis and then adding up the displacements along the various axes to get the overall vector. Examples of this for drawing 4D vectors are shown in **??** and 2. Again the result is a 2D projection, but we can still get a lot of intuition from it especially when visualizing convex hulls. We should note that in 2D projections, a certain amount of information, the "depth" direction(s), in the image gets lost. If we're drawing a 3D vector, depth is one dimensional (out of the page). If we're drawing a 4D vector, depth is 2-dimensional; for a 5D vector, depth is 3-dimensional, etc. Any intuition derived from projections of higher dimensional sets should be verified with rigorous proof.

It is also possible to visualize vectors as stacks of various heights (possibly negative) determined by the values in the vector. For example, $r^T = [r_1\ r_2\ r_3]$ can be represented as three stacks of heights r_1, r_2, and r_3. The value of $r^T x$ for any point $x \in \Delta$ can then be visualized by drawing the convex hull of the tops of the stacks and locating x relative to this convex hull; the height of x relative to the convex hull is $r^T x$. This visualization method is illustrated in Fig. **??**.

C. Linear Programs

To illustrate basic versions of the visualization techniques we will use

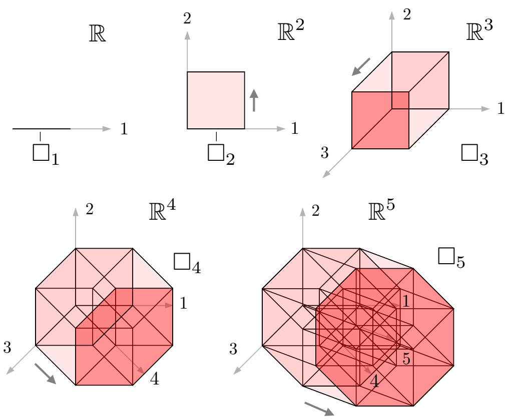

Fig. 4: Multi-dimensions visualization.

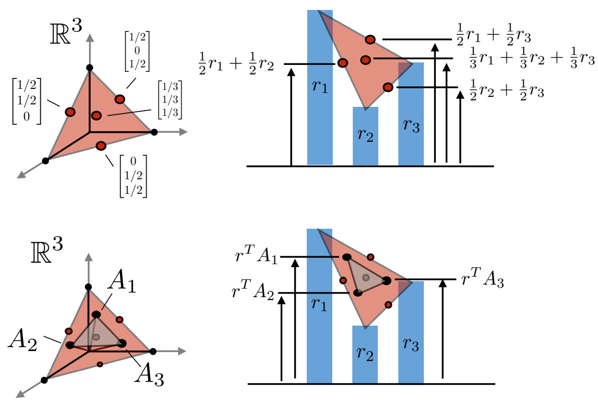

Fig. 5: Visualizing $r^T x$ for $r^T = [r_1\ r_2\ r_3]$ and $x \in \Delta_3$

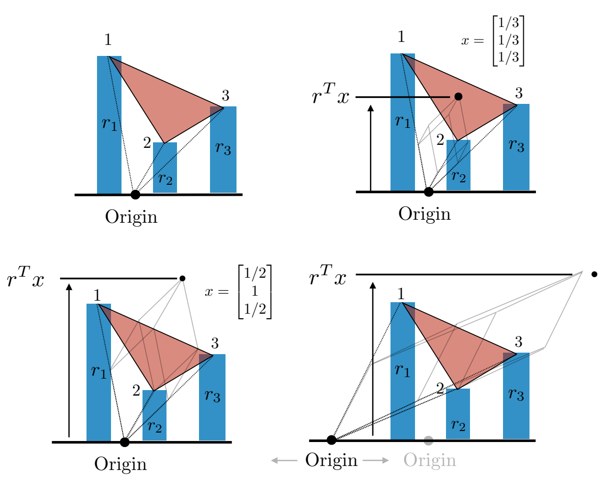

Fig. 6: Visualizing $r^T x$ for $r^T = [r_1\ r_2\ r_3]$ and $x \in \Delta_3$

III. Node-Edge Incidence Matrix

Let $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ be a graph with nodes or *states* $\mathcal{S}$ and directed edges $\mathcal{E}$ where an edge $e \in \mathcal{E}$ runs from s to s'. Let $\rho \in \mathbb{R}_+^{|\mathcal{S}|}$ represent a mass distributions on the states and $x \in \mathbb{R}_+^{|\mathcal{E}|}$ represent a mass distribution over the edges.

We can define graph indicator matrices $E_o, E_i \in \{0,1\}^{|\mathcal{S}|\times|\mathcal{E}|}$ whose e-th columns indicates the starting and ending nodes of edge e, respectively.

$$[E_o]_{se} = \begin{cases} 1 & ;\ \text{if edge } e \text{ starts at state } s \\ 0 & ;\ \text{otherwise} \end{cases}$$

$$[E_i]_{se} = \begin{cases} 1 & ;\ \text{if edge } e \text{ ends at state } s \\ 0 & ;\ \text{otherwise} \end{cases}$$

The matrix $E = E_o - E_i$ is the standard node-edge incidence matrix of a directed graph. If we think of a vector $x \in \mathbb{R}^{|\mathcal{E}|}$ that corresponds to amounts of flow on each edge of the graph, the equation $Ex = 0$ corresponds to mass conversation at each node in the graph.

If we want to reorder the nodes and/or edges of the graph we simply multiply the incidence matrix by permutation matrices U and V (of the appropriate dimensions) on the left and right respectively. The new incidence matrix is given by $E' = UEV$. If we want to flip the direction of an edge, we multiply that column of E by -1. Algebraically, this can be done by left multiplying by E by a diagonal matrix V with 1's and -1's on the diagonal.

A. Domain: Edge-Space

The domain of the incidence matrix E, $\mathbb{R}^{|\mathcal{E}|}$ is the space of mass flows on the edges of a graph. There are several fruitful interpretations of Ex for an edge flow vector $x \in \mathbb{R}^{|\mathcal{E}|}$.

1) Row Geometry: Flow into nodes: As discussed briefly above. Each row of E, $\bar{E}_i$, (corresponding to node i), the quantity $\bar{E}_i x$ sums the flow into node i, $[\bar{E}_i]_j x$ and subtracts the flow out of node i, $[\bar{E}_o]_j x$. If a vector x is orthogonal to $\bar{E}_i$ it means flow into node i is equal to flow out of node i, ie. $\bar{E}_i x = 0 \Rightarrow [\bar{E}_i]_j x = [\bar{E}_o]_j x$. If a flow vector x is orthogonal to all rows, ie. in the nullspace of E, then mass is conserved at each node. We will return to this discussion later.

2) Column Geometry: Tension between nodes: A single element of a flow vector x, x_e refers to the flow on edge e. If e runs from state s to s', the quantity $E_e x_e$ indicates that flow x_e must come out node s and flow into node s' in order to generate flow of x_e on edge e. A general linear combination of the columns, Ex indicates

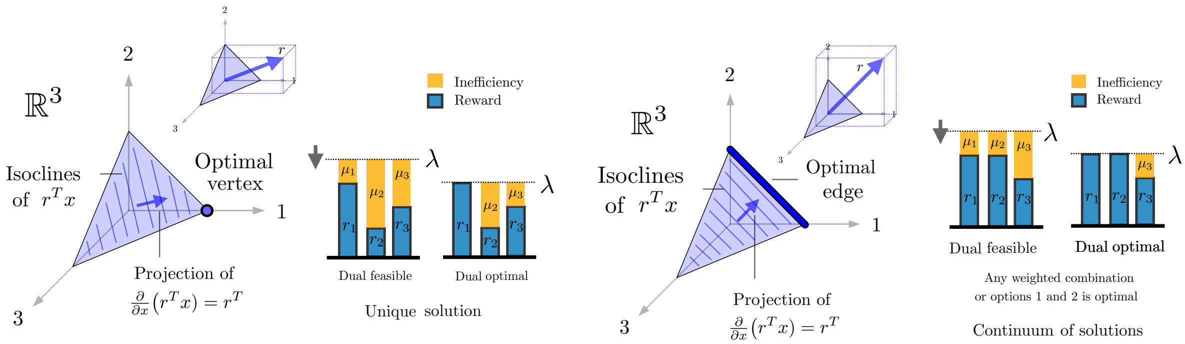

Fig. 7: Visualizing $r^T x$ for $r^T = [r_1\ r_2\ r_3]$ and $x \in \Delta_3$

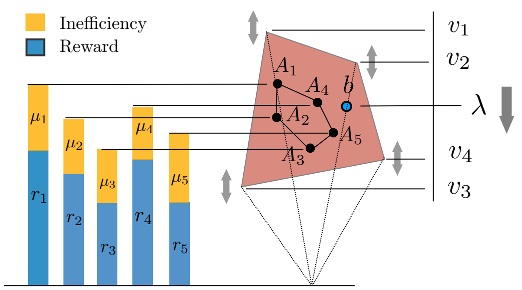

Fig. 8: Visualizing $r^T x$ for $r^T = [r_1\ r_2\ r_3]$ and $x \in \Delta_3$

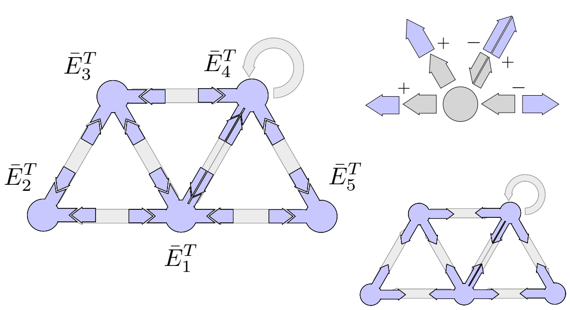

Fig. 9: Illustrating range of E^T

how much flow must come in and out of each node to generate the desired edge of x.

3) Sums of Edges: Effective Edge: We now consider specific linear combinations of the columns of E (with coefficients of 1 or -1) construct vectors of the same form of the columns with a 1 and a -1. These combina-

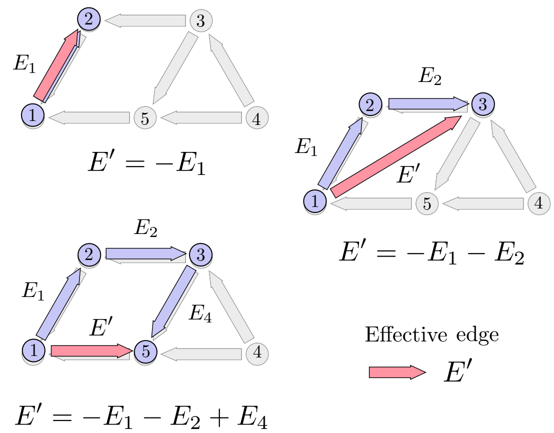

Fig. 10: Edge combinations

tions can form an indicator vector for an "effective" edge between two nodes. (See the discussion of routes below for an algebraic representation.) For example, for the graph shown in Figure **??**, taking a linear combination of the columns E_1, E_2, and E_4 with the appropriate signs creates an indicator for an edge between nodes 1 and 5. Note how the signs relate to the direction of the edges and how the values in the intermediate vectors cancel with each other. Using this method we can represent any edge in a cycle of the graph as a linear combination of the other edges in the cycle.

$$\begin{bmatrix}-1\\0\\0\\0\\1\end{bmatrix} = \underbrace{\begin{bmatrix}-1\\1\\0\\0\\0\end{bmatrix}}_{E_1} + \underbrace{\begin{bmatrix}0\\-1\\1\\0\\0\end{bmatrix}}_{E_2} - \underbrace{\begin{bmatrix}0\\0\\1\\0\\-1\end{bmatrix}}_{E_4}$$

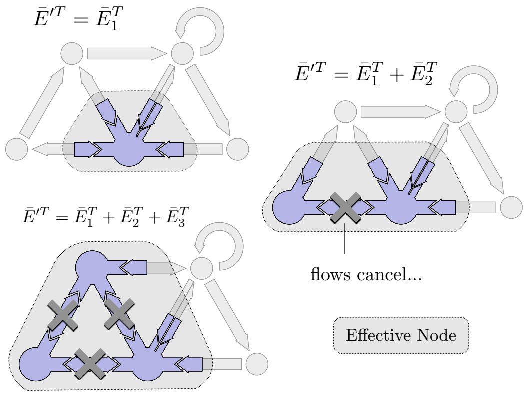

Fig. 11: Linear combinations of rows of E^T

Matrix: Graph Incidence (Node-Edge)

$$E = E_i - E_o \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{E}|}$$

Ex:

$$E = \overset{\leftarrow \text{ Edges } \rightarrow}{\begin{bmatrix} 0 & -1 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & -1 & -1 & 0 \end{bmatrix}} \updownarrow \text{Nodes}$$

↑ Self−loop ↑ Edge from node 2 to 3

Bases:

Range(E) :	F	Range(E^T) :	G^T
Null(E^T) :	$\bar{\mathbf{1}}^T$	Null(E) :	C

SVD: $\Sigma =$ square-root of Laplacian eigenvalues

Left vectors : $\underbrace{U}_{\text{Laplacian eigenvectors}}$ Right vectors : $E^T U^T \Sigma^{-1}$

B. Co-Domain: Node-space

1) *Column Geometry: Edge Tension:*
2) *Row Geometry: Node values/potentials:*
3) *Sums of Nodes: Effective Node:*

C. Tree-Cycle Decomposition

This edge representation idea leads to a general decomposition for incidence matrices based on spanning trees and the cycles of a graph. We note the following decomposition theorem for node-edge incidence matrices. Up to a reordering of the columns, any node-edge incidence matrix can be written in the form

$$E = FG = F\begin{bmatrix} I & C \end{bmatrix}$$

where $F \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|-1}$ is the incidence matrix for a spanning forest (named cause it is a set of spanning trees) and $C \in \mathbb{R}^{|\mathcal{E}|-|\mathcal{S}|+1}$ is a representation of the other edges as linear combinations in of edges in the spanning forest. Note that if we group nodes in spanning trees together, the above decomposition can be expanded as

$$E = \begin{bmatrix} T_1 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & T_{n_c} \end{bmatrix} \begin{bmatrix} I & \cdots & 0 & C_1 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & & I & 0 & \cdots & C_{n_c} \end{bmatrix}$$

where T_k is the incidence matrix for a spanning tree for each connected component and C_k is the corresponding representation of the cycle edges with respect to that spanning tree. Depending on context we may want to group edges of connected components together as well. In this case the appropriate decomposition becomes.

$$E = \begin{bmatrix} T_1 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & T_{n_c} \end{bmatrix} \begin{bmatrix} I & C_1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & I & C_{n_c} \end{bmatrix}$$

We note briefly that the columns E (ie. F) for a spanning forest are linearly independent (proof in Appendix XXX).

D. Domain/Co-Domain Descriptions

The domain and co-domain of the incidence matrix have intuitions descriptions in terms of edge flows and node values respectively. Intuitively if $x \in \mathbb{R}^{|\mathcal{E}|}$ is an edge flow vector, then the quantity $S = Ex$ is the mass flowing into and out of each node. If $v \in \mathbb{R}^{|\mathcal{S}|}$ is a set of values on the nodes of the graph, the quantity $v^T E = q^T$ is difference in values (or "tensions") along the edges. Both the domain and co-domain can be decomposed into two orthogonal subspaces based on the incidence matrix and the fundamental theorem of linear algebra. This decomposition is illustrated in the figure below. In the following sections, we give further descriptions and intuitions for these spaces as well further visualizations.

1) Range of E^T: The orthogonal complement of the space of cycles, the range of E^T is the space of flow into/out of each nodes. The rows of E are indicator vectors for which edges come into each node (with a negative sign indicating that the flow is actually out). For any edge flow x, the equation $\bar{E}_s^T x$ sums up the

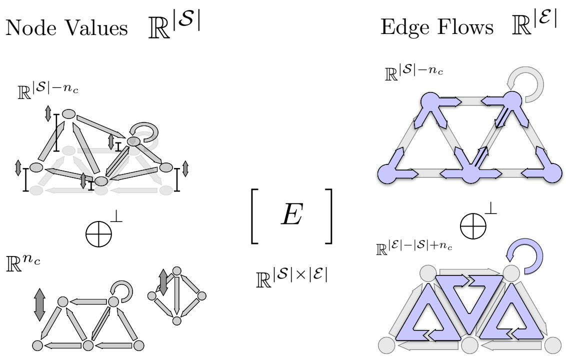

Fig. 12: Fundamental theorem decomposition of E

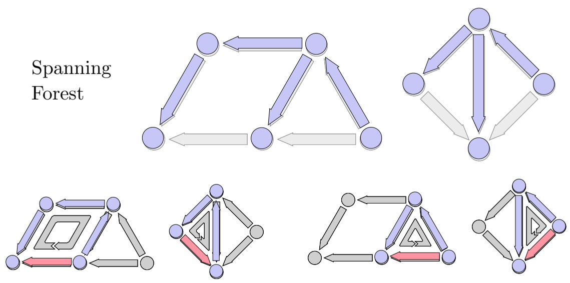

Fig. 13: Illustration of graph structure decomposed into forest and cycle edges

flow into that particular node. A basis for the range of E^T is given by the rows of

$$G = \left[\ I\ C\ \right]$$

Span is immediate from the decomposition and linear independence is the same as the proof for the nullspace basis.

2) Nullspace of E: From the above, tree-cycle decomposition we can write a basis for the nullspace as

$$\mathbf{C} = \begin{bmatrix} C \\ -I \end{bmatrix}$$

Note that each column of this matrix corresponds to traveling around a cycle: the first part through the spanning tree (the particular column of C) and the second part back along the edge not in the spanning tree. Intuitively, this is consistent with $Ex = 0$ describing flows where mass is conserved, ie. cyclic flows. We can also prove

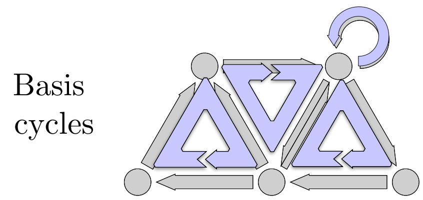

Fig. 14: Illustrating cycle basis

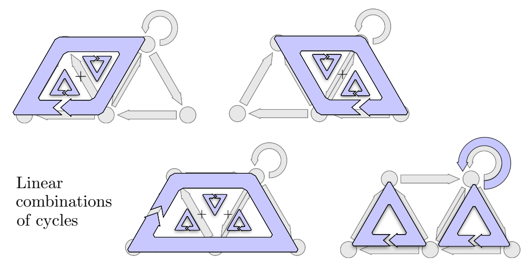

Fig. 15: Illustrating linear combinations of cycles

concisely that $\mathbf{C}$ is a basis.

$$\text{LIN IND:} \qquad \mathbf{C}x = 0 \Rightarrow \begin{bmatrix} Cx \\ -x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow x = 0$$

$$\text{SPAN:} \qquad Ex = 0 \Rightarrow F\begin{bmatrix} I\ C \end{bmatrix}\begin{bmatrix} x' \\ x'' \end{bmatrix} = 0$$

$$\Rightarrow x' = -Cx'' \Rightarrow x = \begin{bmatrix} C \\ -I \end{bmatrix}(\text{-}x'')$$

where the beginning the second line depends on the columns of F being linearly independent.

We note that the above decomposition gives a canonical basis for the nullspace of the incidence matrix that gives the cycle space of the graph. Specifically, the columns of the matrix

$$\mathbf{C} = \begin{bmatrix} C \\ -I \end{bmatrix}$$

form a basis for the cycle space. A canonical orthonormalization of this basis is given by

$$\begin{aligned} \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1/2} &= \begin{bmatrix} C \\ -I \end{bmatrix}(C^TC + I)^{-1/2} \\ &= \begin{bmatrix} C(C^TC+I)^{-1/2} \\ -(C^TC+I)^{-1/2} \end{bmatrix} \end{aligned}$$

Note that another option for orthonormalizing $\mathbf{C}$ would be to do Gram-Schmidt on the columns, ie. perform a QR-decomposition on $\mathbf{C}$.

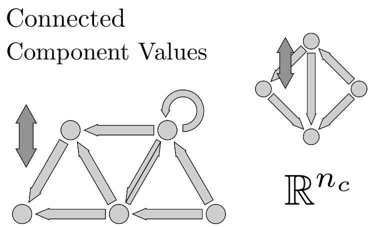

Fig. 16: Linear combinations of rows of E^T

Matrix: Cycle Indicator Matrix

$$\mathbf{C} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{C}|}$$

Ex:

$$\mathbf{C} = \overset{\leftarrow \text{ Cycles } \rightarrow}{\begin{bmatrix} 1/3 & 0 & 0 \\ 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/2 \\ 1/3 & 0 & 0 \\ 0 & 1/3 & 1/2 \end{bmatrix}} \begin{matrix}\uparrow \\ \text{Edges} \\ \downarrow\end{matrix}$$

Properties: Column stochastic

3) Nullspace of E^T: The co-domain intuitively is the space of values on the nodes. The equation $v^T E$ gives the difference between values along edges and thus the nullspace is the set of equal shifts among all the elements in a connected component. A basis is given by the matrix

$$\bar{\mathbf{1}}^T = \textbf{blkdiag}\Big([\underbrace{\mathbf{1}^T, \cdots, \mathbf{1}^T}_{\times n_c}] \Big)$$

where the constant vectors $\mathbf{1}^T$ are have the length corresponding to connected components of the graph.

4) Range of E: The range of E is the space of relative values on the nodes in each connected component. For a single connected component it is not surprising that a spanning tree should define a basis for the range; defining the difference along edges in a spanning tree is sufficient to fix the differences in the values between the nodes. Algebraically, we see that a spanning forest incidence matrix spans the range immediately from the tree-cycle decomposition. We also note that linear independence can be shown using the the transformation given in Appendix XXX of the form $F' = FW$. This new basis is equivalent to measuring difference in edges from a particular root node as illustrated in Figure XXX. Depending on context both of these decompositions can be useful.

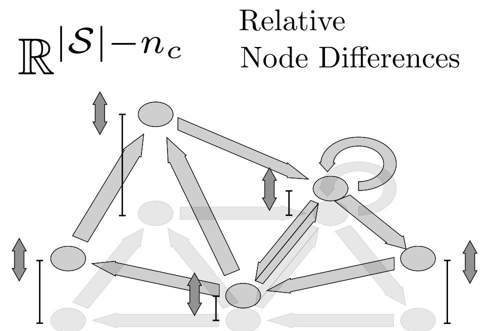

Fig. 17: Linear combinations of rows of E^T

IV. Laplacian/Degree/Adjacency

One of the shape matrices associated with the incidence matrix EE^T is called the graph Laplacian and is fundamental in algebraic graph theory $L = EE^T \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$. Spectral graph theory studies the eigenstructure of the Laplacian and it's relation to properties of the graph. Here we only note that the square-root of the Laplacian $L^{\frac{1}{2}} = (EE^T)^{\frac{1}{2}}$ is positive semi-definite shape matrix of the incidence matrix and thus it's eigenvalues are the singular values of the incidence matrix and the eigenvectors of the Laplacian are the left-singular vectors of E. Explicitly we have that if L has EVD given below, then the SVD of E is given by

$$L = EE^T = U \begin{bmatrix} \Sigma^2 & 0 \\ 0 & 0 \end{bmatrix} U^T, \quad \Rightarrow \quad E = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^T$$

where U is an orthonormal basis of eigenvectors for the Laplacian and V is an orthonormal basis given by

$$V^T = \begin{bmatrix} - \ V'^T \ - \\ - \ V''^T \ - \end{bmatrix} = \begin{bmatrix} - \ U^T E \ - \\ - \ V''^T \ - \end{bmatrix} = \begin{bmatrix} - \ U^T F \begin{bmatrix} I & C \end{bmatrix} \ - \\ - \ V''^T \ - \end{bmatrix}$$

where V'' completes a basis for the domain with components in the nullspace of E. This can be taken to be any orthonormal basis for the cycle space.

The intuition for the columns of U is vibration modes of the graph and the rows of $U^T E$ are the resulting edge tensions from the vibration modes.

Matrix: Graph Laplacian

$$L = EE^T = D - A \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$$

Ex:

$$L = \underbrace{\begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}}_{D} - \underbrace{\begin{bmatrix} 0 & 1 & 1 & 2 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 2 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}}_{A}$$

$$= \begin{bmatrix} 5 & -1 & -1 & -2 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ -2 & 0 & -1 & 4 & -1 \\ -1 & 0 & 0 & -1 & 2f \end{bmatrix}$$

Properties: PSD, rank: $|\mathcal{S}| - n_c$

EVD/SVD:

Eigen-values $\underbrace{0, \cdots, 0}_{\text{\# connected components}} < \lambda_1 \leq \cdots \lambda_{|\mathcal{S}|-n_c}$

V. Flow Conservation Constraints

Given an edge flow x, the flow through the nodes on the graph is given by $\rho = E_i x = E_o x$. A constraint of the form $Ex = E_o x - E_i x = 0$ represents conservation of flow on the edges of the graph and thus the nullspace of E is closely related to the cycles of the graph. Let $c \subset \mathcal{E}$ refer to a cycle on the directed graph and let $\mathcal{C} \subset 2^{\mathcal{E}}$ be the set of all cycles on the graph. Let $\mathbf{C} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{C}|}$ be a matrix indicating which edges are in which cycles (scaled by each cycle length).

$$\big[\mathbf{C}\big]_{ec} = \begin{cases} \frac{1}{|c|} & \text{if } e \in c \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $|c|$ is the length of cycle c. Note that for a fully connected graph, only $|\mathcal{E}| - |\mathcal{S}| - 1$ cycles need be enumerated in order to span the nullspace of E (by the rank-nullity theorem). Given a flow distribution on the cycles of a graph $z \in \mathbb{R}_+^{|\mathcal{C}|}$, the resulting flow distribution on the edges of the graph is given by $x = \mathbf{C}z$.

Note that E_i, E_o, and $\mathbf{C}$ are all column stochastic

$$\mathbf{1}^T E_i = \mathbf{1}^T, \quad \mathbf{1}^T E_o = \mathbf{1}^T, \quad \mathbf{1}^T \mathbf{C} = \mathbf{1}^T \tag{2}$$

which results in mass conservation between state flows, edge flows, and cycle flows

$$\mathbf{1}^T \rho = \mathbf{1}^T E_o x = \mathbf{1}^T E_i x = \mathbf{1}^T x = \mathbf{1}^T \mathbf{C} z = \mathbf{1}^T z$$

Steady state flow of mass $m = 1$ on a graph can be represented by the constraints.

$$\begin{aligned} \mathcal{X} &= \Big\{ x \in \mathbb{R}^{|\mathcal{E}|} \;\Big|\; E_o x = E_i x,\ \mathbf{1}^T x = 1,\ x \geq 0 \Big\} \\ &= \Big\{ x \in \mathbb{R}^{|\mathcal{E}|} \;\Big|\; Ex = 0,\ x \in \Delta_{|\mathcal{E}|} \Big\} \end{aligned}$$

If we enumerate the cycles of the graph, we can reparametrize this set as

$$\begin{aligned} \mathcal{X} &= \Big\{ x \in \mathbb{R}^{|\mathcal{E}|} \;\Big|\; x = \mathbf{C}z,\ \mathbf{1}^T z = 1,\ z \geq 0,\ z \in \mathbb{R}^{|\mathcal{C}|} \Big\} \\ &= \Big\{ x \in \mathbb{R}^{|\mathcal{E}|} \;\Big|\; x = \mathbf{C}z,\ z \in \Delta_{|\mathcal{C}|} \Big\} = \mathbf{C}\Delta \end{aligned}$$

A. Path Flows

1) Source-Sinks: A path or route is a set of edges that go from a source to a sink node. Source and sink vectors $S_i, S_o \in \mathbf{R}^{|\mathcal{S}|}$ indicate the source and sink nodes

$$\big[S_o\big]_s = \begin{cases} 1 & ; s \text{ is source} \\ 0 & ; \text{ o.w.} \end{cases}, \quad \big[S_i\big]_s = \begin{cases} 1 & ; s \text{ is sink} \\ 0 & ; \text{ o.w.} \end{cases},$$

and we can also define a source-sink vector $S = S_i - S_o \in \mathbb{R}^{|\mathcal{S}|}$.

$$\big[S\big]_s = \begin{cases} -1 & ; s \text{ is source} \\ 1 & ; s \text{ is sink} \\ 0 & ; \text{ o.w.} \end{cases},$$

Note that the vector S is the same as an indicator vector for an edge from the source to the sink.

We will denote a set of routes as $\mathcal{R} \subset 2^{\mathcal{E}}$ and $r \in \mathcal{R}$. We often represent a set of routes as columns of an indicator matrix $\mathbf{R}^{|\mathcal{E}|\times|\mathcal{R}|}$

$$\big[\ \mathbf{R}\ \big]_{er} = \begin{cases} 1 & ; \text{ if } e \in r \\ 0 & ; \text{ otherwise} \end{cases}$$

It is useful to consider how a route indicator vector interacts with the incidence matrix E. Specifically, if $\mathbf{R}_r$ is a route indicator vector (a column of $\mathbf{R}$) then we have

$$E\mathbf{R}_r = \sum_{e \in r} \begin{bmatrix} | \\ E_e \\ | \end{bmatrix} \big[R_r\big]_e = S$$

as illustrated in Figure XXX. Intuitively, flow along a route from node s to s' is the same (from a mass conversation perspective) as flow along an edge directly between those node.

2) Route vs. Cycle Flows: Note that here we can construct cycle flows from route flows and vice versa as illustrated in Figure XXX.

The set of flows between a source and a sink can be represented using an incidence matrix

$$\mathcal{X} = \Big\{\ x \in \mathbb{R}^{|\mathcal{E}|} \;\Big|\; Ex = Sm,\ x \geq 0 \Big\}$$

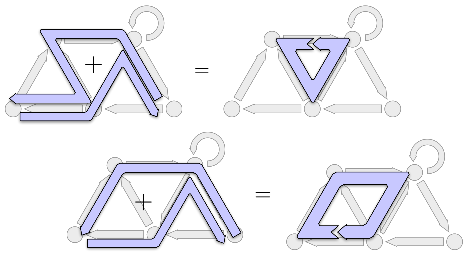

Fig. 18: Illustrating cycles as differences between routes

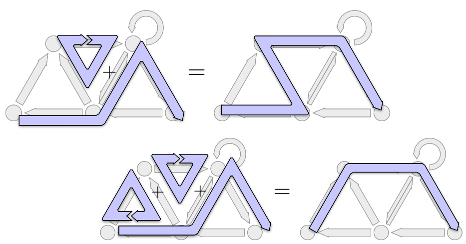

Fig. 19: Illustrating adding cycles to routes to produce new routes

We can also represent this set by enumerating the routes

$$\mathcal{X} = \left\{ x \in \mathbb{R}^{|\mathcal{E}|} \;\middle|\; x = \mathbf{R}z + \mathbf{C}z',\ \mathbf{1}^T z = m,\ x \geq 0 \right\}$$

For these representations to be the same set we only need $\mathbf{R}$ to have at least one route and $\mathbf{C}$ must span the nullspace of the incidence matrix. Other route flows can then be represented by combinations of the base route and cycle flows as illustrated in the Figure XXX.

Note here it is important that we don't enforce that $z \geq 0$. In some contexts, we may also want to represent a set of routes as a convex combination of vertices defined by routes

$$\mathcal{X} = \left\{ x \in \mathbb{R}^{|\mathcal{E}|} \;\middle|\; x = \mathbf{R}z + \mathbf{C}z',\ \mathbf{1}^T z = m,\ x \geq 0 \right\}$$

Note that in general these representations are not the same most obviously in that the first set is compact and the second set is not. In general, if we are to use the vertex presentation, the matrix $\mathbf{R}$ must enumerate as many routes as we want to consider. For any network routing problem of substantial size the number of routes $|\mathcal{R}|$ is prohibitively large and the incidence matrix representation should be used. This vertex representation is commonly used in many routing game problems and can be quite useful for small scale path studies and counter-examples.

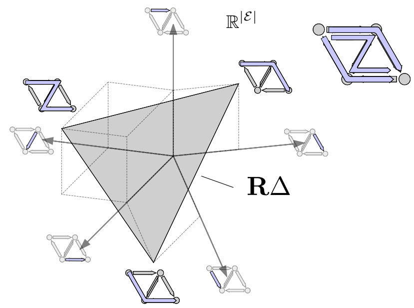

Fig. 20: Illustration of set of route flows.

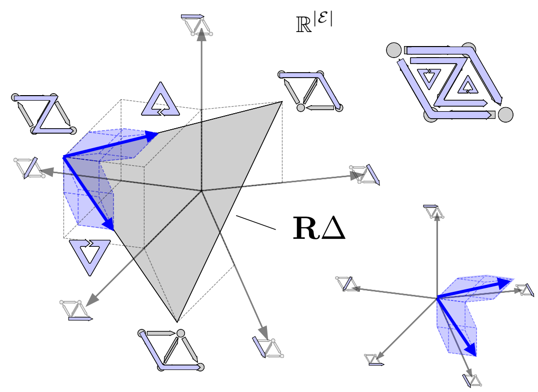

Fig. 21: Illustration of relationships between route flows and cycle flows.

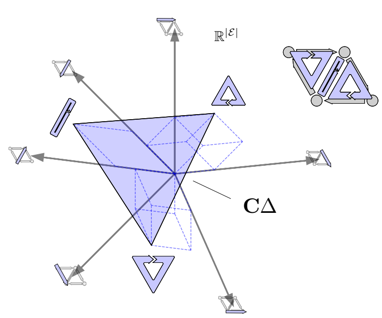

Fig. 22: Illustration of set of cycles .

Example

We give a

We give a geometric illustration of these algebraic graph theory objects for the fully connected graph on three states, $|\mathcal{S}| = 3$. $|\mathcal{E}| = 9$. The incidence matrices are given by

$$E_o = \begin{bmatrix} E_{o1} & E_{o2} & E_{o3} \end{bmatrix}$$

where

$$E_{o1} = \delta_1 \mathbf{1}_1^T,\ E_{o2} = \delta_2 \mathbf{1}_2^T,\ E_{o3} = \delta_3 \mathbf{1}_3^T,$$

$$E_i = \mathbf{1}^T \otimes I_3 = \begin{bmatrix} E_{i1} & E_{i2} & E_{i3} \end{bmatrix}$$

where

$$E_{i1} = E_{i2} = E_{i3} = I_3$$

and the cycle indicator matrix is given by

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/3 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/3 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 1/3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The graph structure, space of state flows, cycle structure, and columns of E_o, E_i, and E; are all illustrated in Fig. **??**. The space of edge flows and $\mathbf{C}\Delta$ are illustrated in Fig. **??**.

VI. Shortest-Path Optimization

Given the above mass conversation constraints, we can formulate several linear programs to solve for shortest path type problems. We note that these formulations may not be the most efficient in practice for these type problems but simply conceptually have merit as well.

In order to formulate the appropriate optimization problem, we define a vector of costs associated with each edge in the graph $\ell \in \mathbb{R}^{|\mathcal{E}|}$. The cost associated with a particular route $r \in \mathcal{R}$ can be computed as the quantity $\ell^T \mathbf{R}_r$. Given the above description of the space of edge flows, we can frame the given optimization problem as

If we enumerate all possible routes between a source and a sink in a routing matrix $\mathbf{R}$ then we can frame the shortest path problem as the optimization problem by solving the linear program as

$$\left\{ \max_{z}\ \ell^T \mathbf{R} z \;\middle|\; \mathbf{1}^T z = 1,\ z \geq 0 \right\}$$

Here $z \in \mathbb{R}_+^{|\Gamma|}$ is the mass distribution over the the set of routes and $\ell^T \mathbb{R}$ is vector of rewards for each route. This optimization problem is illustrated in the Figure XXX.

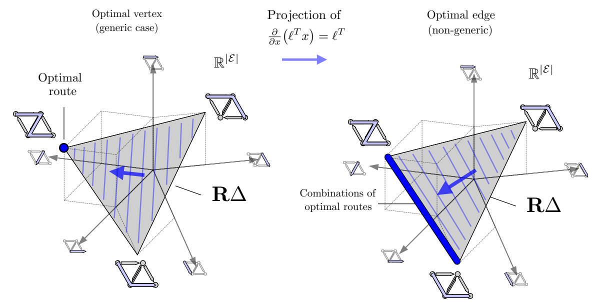

Fig. 23: Primal linear programming problem illustration.

The dual formulation of this problem is given by

$$\left\{ \min_{\lambda, \nu}\ \lambda \;\middle|\; \lambda \mathbf{1}^T = \ell^T \mathbf{R} + \nu^T,\ \nu^T \geq 0 \right\}$$

where the dual variables are $\lambda \in \mathbb{R}$ for the equality constraint and $\nu \in \mathbb{R}_+^{|\Gamma|}$ for the inequality constraint. λ is lower bound on the minimum travel cost and ν_π is the inefficieny of the route r. Intuitively, solving the dual optimization problem can be pictured as pulling λ up as high as possible while keeping $\nu \geq 0$ as shown in Fig. **??**. Optimality is guaranteed by complementary slackness $\nu_\pi z_\pi = 0$, ie. no mass chooses an inefficient route, Since the overall reward is a linear function of the route distributions and the set of steady-state distributions is characterized by a polytope as discussed above. Since this is an LP on, a single route is purely optimal (if not uniquely optimal). Geometrically this corresponds to that a constant cost vector ℓ in general points into a vertex; and if it points precisely into a face or edge any distribution of mass over that face (including all mass on any vertex) has the same cost.

Note that while conceptually pleasing this optimization, enumerating the routing matrix is wildly inefficient.

A. State Formulation

Alternatively, we can characterize the feasible set of routes using the incidence matrix constraint given above.

$$\left\{ \max_{y}\ \ell^T x \;\middle|\; Ex = Sm,\ \mathbf{1}^T x = 1,\ x \geq 0 \right\}$$

The dual problem for this formulation is given by

$$\left\{ \min_{\lambda, v, \mu}\ \lambda \;\middle|\; \lambda \mathbf{1}^T = r^T + v^T (E_o - E_i) + \mu^T,\ \mu^T \geq 0 \right\}$$

The dual variable $v \in \mathbb{R}^{|\mathcal{S}|}$ represents a value function on each state that encodes the minimum distance from that node to the destination. $\mu \in \mathbb{R}_+^{|\mathcal{A}|}$ represents the inefficiency of taking a particular edge. The optimum can be found by maximizing the travel cost as much as possible while ensuring that each edge is either optimal or suboptimal, ie. have a positive inefficiency. At optimum, the complementary slackness constraint $y_a \mu_a = 0$ ensures that no suboptimal edges are chosen and all mass travels down the fastest route.

Constraint Set: Cycle Flows $\mathcal{X}$

$$\underbrace{(E_i - E_o)x = 0,}_{\text{Local mass conserve.}} \quad \underbrace{\mathbf{1}^T x = m,}_{\text{Global mass conserve}} \quad \underbrace{x \geq 0}_{\text{Mass pos.}}$$

Remark 1. *The constraint above can be rewritten element-wise as*

$$v_s = \ell_e + v_{s'} + \mu_a \tag{3}$$

where edge e runs from state s to state s'. Summing up this condition along a route, we see that the values at the nodes cancel out and we're left with the equation

$$\sum_{e \in r} \ell_e = v_o - v_d$$

In matrix form this is simply given by right multiplying the constraint by a route indicator vector $E\mathbf{R}_r = Sm$ and at $\mathbf{1}^T y = 1$, $y \geq 0$) gives

$$\begin{aligned} \ell^T \mathbf{R}_r &= v^T E \mathbf{R}_r + \mu^T \mathbf{R}_r \\ &= v^T S m + \mu^T \mathbf{R}_r \end{aligned}$$

This relationship is illustrated in Fig. **??**. *Note here is determined by the relative difference between the value at the origin destination and the route inefficiency is $\mu^T \mathbf{R}_r$. This last term is 0 for optimal routes.*

B. Cycle Flow Optimization

We note a modified of this problem that solves for flows on routes. This problem is slightly less useful; however it has deep connections with a linear programming formulation of Markov decision processes presented in part 2 of this paper. It also has some subtle yet interesting differences from the routing problem shown above.

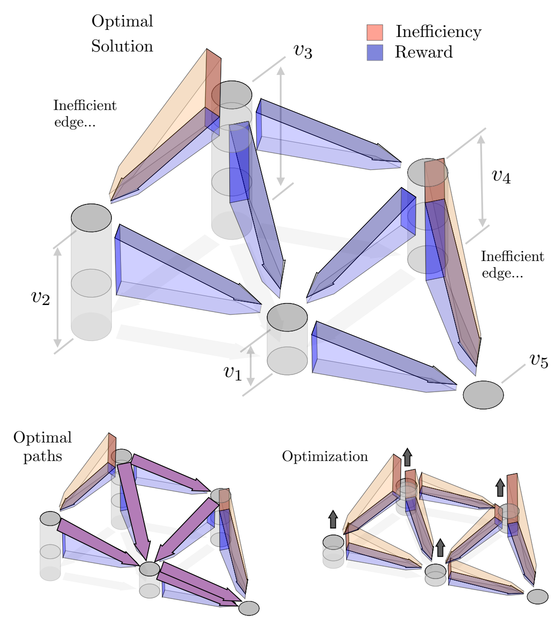

Fig. 24: Shortest path optimization problem (dual version)

References

[1] T. Wang, M. Bowling, and D. Schuurmans, "Dual representations for dynamic programming and reinforcement learning," in *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*. IEEE, 2007, pp. 44–51.

[2] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.