A Physarum Inspired Dynamic to Solve Semi-Definite Programming

Andreas Karrenbauer
karrenba@mpi-inf.mpg.de

Hamidreza Kamkari
hamidkamkari@ce.sharif.edu

Mohammadamin Sharifi
mmdsharifi@ce.sharif.edu

Abstract

Physarum Polycephalum is a Slime mold that can solve the shortest path problem. A mathematical model based on the Physarum's behavior known as the Physarum Directed Dynamics can solve positive linear programs. In this paper, we will propose a Physarum based dynamic based on the previous work and introduce a new way to solve positive Semi-Definite Programming (SDP) problems that are harder than positive linear programs. Empirical results suggest that this extension of the dynamic can solve the positive SDP showing that the nature-inspired algorithm can solve one of the hardest problems in the polynomial domain. In this work, we will formulate an accurate algorithm to solve positive and some non-negative SDPs and formally prove some key characteristics of this solver thus inspiring future work to try and refine this method.

1 Introduction

The Physarum computing model is an analog computing model motivated by the network dynamics of the slime mold Physarum Polycephalum. In wet-lab experiments, it was observed that the slime mold is apparently able to solve shortest path problems [1]. A mathematical model for the dynamic behavior of the slime was proposed in [2]. It models the slime network as an electrical network with time-varying resistors that react to the amount of electrical current flowing through them.

A variant of the Physarum dynamics, the directed Physarum dynamics, is known to solve positive linear programs in standard form [3, 4]. A positive linear program asks to minimize a linear function $c^T x$ with a positive cost vector $c \in \mathbb{R}^n > 0$ subject to the constraints $Ax = b$ and $x \geq 0$. Here $A \in \mathbb{R}^{m\times n}$and $b \in \mathbb{R}^m$. Formally:

$$min\{c^T x \mid Ax = b\ ,\ x \geq 0\} \tag{1}$$

The Physarum dynamics facilitates a vector q which can be achieved by solving the following problem:

$$q(x) = \underset{f}{argmin}\{f^T R f \mid Af = b\} \tag{2}$$

where R is defined as a diagonal matrix where $R_{i,i} = \frac{c_i}{x_i}$. The R matrix is known as the **Resistor** matrix and $G = R^{-1}$ is known as the **Conductance** matrix. This optimization problem is referred to as the **Update Problem** and solving it is easier than the SDP problem given in (1) because it does not restrict f to be a positive vector. That is to say, one can find the solution to this optimization problem via Lagrange multipliers.

The Physarum directed dynamic requires to solve the update problem and obtain the solution $q(x)$. Afterward, $x(t)$ is characterized using the following differential equation:

$$\dot{x}(t) = q(x(t)) - x(t), \tag{3}$$

One can also define a discretized version of the dynamic by considering $\dot{x}(t) = \frac{x_{after}(t) - x_{before}(t)}{h}$ and reformulating as the following:

$$x_{after} = h.q(x_{before}) + (1-h).x_{before} \tag{4}$$

It has been proven that by choosing $x(0)$ cleverly and large enough the dynamic will converge to the optimum solution of (1).

Theorem 1.1. *[3, 4] Consider I to the union of the support of all the optimum solution vectors of* (1)*:*

$$I = \{i \mid i \in supp(x^*) \text{ for some optimal solution } x^*\},$$

and consider F to be the set of all feasible solutions of (1)*:*

$$F = \{x \mid x \text{ satisfies the constraints of the LP }\}$$

Then, if the dynamics (3) *has a solution in which all elements of x are positive, then:*

1. $\inf_t x_i(t) > 0$ *for all* $i \in I$
2. $dist(x(t), F) \to 0$ *as* $t \to \infty$
3. $\lim_{t\to\infty} c^T x(t) = c^T x^*$

Numerous results have been derived by altering the dynamics. Some showcase using the discretized version of the dynamics in (4) and choose h such that the processing time of the dynamics improves, [5] Others add a coefficient and create non-uniform dynamic of the form $x_i(t) = d_i(q_i(t) - x_i(t))$ [6]. A detailed proof of the directed dynamics for the base positive LP problem can also be seen in [7]. In this document, we try to generalize the Physarum directed dynamics and obtain a new dynamic system which can solve positive SDPs.

An SDP can be written as one of the primal-dual pair

$$\begin{aligned}&\min\{tr(C^TX) : tr(A_\ell^TX) = b_\ell \forall \ell \in [m],\, X \succeq 0\}\\ &\geq \max\{\sum_{\ell=1}^m b_\ell y_\ell : \sum_{\ell=1}^m y_\ell A_\ell + S = C,\, S \succeq 0\},\end{aligned}$$

where we may assume without loss of generality that C and all A_ℓ are symmetric. This is due to the fact that X is symmetric and each formula of form $tr(MX)$ is equivalent to $tr(\frac{M+M^T}{2}X)$; that being said, we rewrite all of these conditions in this form where C will be replaced by $\frac{C+C^T}{2}$ and each A_ℓ matrices will be replaced by $\frac{A_\ell + A_\ell^T}{2}$. We will call an SDP **positive** if the cost matrix is positive semi-definite. One can write the positive SDP primal and dual pair as below:

$$\begin{aligned}&\min\{tr(CX) : tr(A_\ell X) = b_\ell \forall \ell \in [m],\, X \succeq 0\},\ C \succ 0\\ &\geq \max\{\sum_{\ell=1}^m b_\ell y_\ell : \sum_{\ell=1}^m y_\ell A_\ell + S = C,\, S \succeq 0\},\end{aligned} \tag{5}$$

Note that one can also write the SDP in a vectorized notation where $vec(M)$ for an $n \times n$ matrix is an n^2 vector whose $n(i-1)+j$th element is $M_{i,j}$, i.e, it is obtained by stacking up the columns of M on top of each other. Consider Ω to be a $m \times n^2$ matrix defined as below:

$$\Omega = \begin{bmatrix} vec(A_1)^T \\ vec(A_2)^T \\ ... \\ vec(A_m)^T \end{bmatrix}. \tag{6}$$

With these notations one can rewrite the primal of the SDP as follows:

$$min\{vec(C)^T vec(X) \mid \Omega.vec(X) = b,\ X \succeq 0\},\ C \succ 0.$$

In this document, we propose a dynamic and investigate to what extent this Physarum-inspired dynamical system is able to solve positive SDPs. positive SDPs are an important subset of SDPs that are useful in applications such as an approximation for max-cut.

In Section 2 we will introduce our extension of the Physarum dynamics. Similar to the Physarum dynamics introduced for LP, we will define a matrix function $X(t)$ which is charactarized by a differential equation similar to (3). In Section 3 we define an update problem similar to (2) and formulate its solution. Sections 4 and 5 are dedicated to proving this dynamic converges to a certain optimum point given a set of conditions known as Convergence Conditions (CC). In Section 6 we will propose an extension of positive SDP that complies to all of the convergence conditions previously defined and we will show that the solution to this SDP yields an optimum result an arbitrary positive SDP.

Section 7 gathers all the theoretical results from the previous sections. In that, we will introduce a complete solver for positive SDP problems and run some experiments to evaluate its performance. These experiments have shown promising results indicating the Physarum dynamic converges to an optimum solution of the SDP. Interestingly, empirical findings suggest some of the convergence conditions proposed are not needed and the dynamic converges to the optimum even when they do not hold. All things considered, the Physarum based algorithm to solve SDPs is a novel approach to solve positive SDPs and with extra effort might compete with other well-known optimization methods.

2 Proposed Method

To solve the positive SDP we will introduce the matrix function $X(t)$ using the following dynamics:

$$\dot{X}(t) = Q(X(t)) - X(t), \tag{7}$$

this differential equation characterizes $X(t)$ which is changing over time $t \geq 0$. We will study how does the dynamic behave for solving the positive SDP.

Moreover, $Q(X(t))$ is known as the solution to an optimization problem that we define as the **SDP update problem** which is analogous to the update problem in the LP case:

$$\min\{vec(Q)^T R(C, X(t)) vec(Q) \mid \Omega vec(Q) = b,\ Null(X(0)) \subseteq Null(Q),\ Q^T = Q\} \tag{8}$$

Analogous to the previous work, for a matrix F that complies to the constraints above, we refer to the objective value $vec(F)^T R(C, F) vec(F)$ as the **Energy** of F and $Q(X(t))$ minimizes this energy. We define the R matrix above as the **resistor** matrix which is calculated using $X(t)$ and C and will be explained in details in Section 3.

The constraint of the form $\Omega \cdot vec(Q) = b$ which is equivalent to

$$\forall \ell \in [m] : tr(A_\ell Q) = b_\ell$$

are referred to as the **linear equality constraints** and the constraints of the form

$$Null(X(0)) \subseteq Null(Q)$$

as the **null space** constraints.

We try to show that given a certain set of conditions known as the **Convergence Conditions (CC)**, the proposed dynamic converges to the optimum solution of the positive SDP. The conditions are as below:

(CC.1) $X(0) \succeq 0$.

(CC.2) $Null(C) \subseteq Null(X(0))$.

(CC.3) $C^\dagger_{(k_0)}$ is a linearly feasible solution of the SDP. i.e., $\forall \ell \in [m] : tr(A_\ell C^\dagger_{(k_0)}) = b_\ell$.

(CC.4) $X(0)$ complies to the linear feasibility constraints, i.e., $\forall \ell \in [m] : tr(A_\ell X(0)) = b_\ell$.

$C^\dagger_{(k_0)}$ is a constant matrix that will be defined in Section 3.

We aim to ultimately show the following claim holds: (5).

Claim 2.1. *If $X(t)$ is a matrix function that follows the dynamic* (7) *and (CC.1), (CC.2), (CC.3), and (CC.4) hold, $X(t)$ will converge to an X^* value which optimizes the following for $C \succeq 0$:*

$$\min\{tr(CF) \mid \forall \ell \in [m] : tr(A_\ell F) = b_\ell,\ F \succeq 0,\ Null(X(0)) \subseteq Null(F)\}$$

Note that in case $C \succ 0$ and the SDP is positive if we choose X as a full-rank matrix, then if the claim holds $X(t)$ will converge to the optimum primal of (5). Among the convergence conditions, (CC.3) and (CC.4) are the most restrictive ones. In Section 6 we will introduce a new SDP problem based on (5) known as the **Augmented SDP** which has a certain useful property and complies to (CC.3). In this SDP we can easily choose $X(0)$ such that (CC.1), (CC.2), and (CC.4) also hold. We will then discuss that solving this particular SDP leads to solving the positive SDP in the most general case. In the sections before 6 we assume convergence for the case where (CC.1) to (CC.4) hold.

Additionally, we will discuss in Section 7 that (CC.3) and (CC.4) might not necessarily be needed as the empirical results show convergence even when the problem is not augmented. In these experiments, $X(0)$ is initialized by a very large matrix, e.g. $X(0) = B \times C$ where B is a large number with respect to the problem. These experiments show that despite not having a formal proof, (CC.3) and (CC.4) are redundant.

3 The Update Problem

In this section, we will define the update problem which was mentioned in (8) accurately. To this end, we will first solve a transformed version of the update problem to come up with an appropriate resistor matrix. Then we will propose a method to solve the update problem using Lagrange multipliers. This section is divided into four parts:

- In the first part, we will do a diagonalization of $X(t)$ and introduce the matrix $U(t)$ and a diagonal matrix $\Lambda(t)$. These matrices will represent $X(t)$ as $U(t)\Lambda(t)U^T(t)$ at each point of time.

- We will then introduce the transformed update problem with respect to $\Lambda(t)$ and $U(t)$ at each point of time.
- Using Lagrange multipliers we will solve the transformed update problem and define the resistor matrix $R(X(t), C)$.
- Finally, we will propose a solution based on Lagrange multipliers to solve (8) and obtain $Q(X(t))$ directly.

3.1 Diagonalization of X

Here, we will introduce matrices $\Lambda(t)$ and $U(t)$ at each point of time which are able to diagonalize $X(t)$. This diagonalization is of crucial importance and is key to defining a proper update problem for our purpose. The following theorem claims such matrices exist and points out key characteristics of them:

Theorem 3.1. *Assume $X(t)$ is a matrix function that follows the dynamics in (7) and that the resistor for the update problem is chosen arbitrarily, if $X(0)$ complies to (CC.2) and its rank is k_0, one can define two $n \times k_0$ matrices $U(t), \tilde{U}(t)$ with linearly independent columns and a diagonal $k_0 \times k_0$ matrix $\Lambda(t)$ such that:*

$$\begin{aligned} C^{\dagger}_{(k_0)} &= U(t)U^T(t) \\ C_{(k_0)} &= \tilde{U}(t)\tilde{U}^T(t) \\ X(t) &= U(t)\Lambda(t)U^T(t) \end{aligned}$$

in which $C_{(k_0)}$ and $C^{\dagger}_{(k_0)}$ are constant time-invariant even though $U(t)$ and $\tilde{U}(t)$ change with respect to t. For convenience we will denote $\tilde{U}(0)$ by $\tilde{O}$ and $C_{(k_0)} = \tilde{O}\tilde{O}^T$.

Moreover, if $F(t)$ is an arbitrary matrix function of time that can be written as $F(t) = U(t)\tilde{F}(t)U^T(t)$ for any time t given $U(t)$ then,

(i) $tr(CF(t)) = tr(C_{(k_0)}F(t)) = tr(\tilde{O}^T F(t)\tilde{O}) = tr(\tilde{F}(t))$

(ii) If $v \in Null(X(0))$, then $F(t)v = U(t)v = 0$ for any time t.

Additionally, if we indicate the ith the columns of U and $\tilde{U}$ with u_i and $\tilde{u}_i$ respectively and the ith element of the diagonal $\Lambda(t)$ with $\lambda_i(t)$, then,

$$\begin{aligned} &\tilde{u}_i^T u_j = u_j^T \tilde{u}_i = \begin{cases} 1 & \text{if } i = j \\ 0 & o.w \end{cases} \qquad \text{for all } 1 \leq i, j \leq k_0 \\ &X\tilde{u}_i = \lambda_i u_i. \end{aligned}$$

The following lemma points out a key characteristic of $Null(X(t))$ at each time t. Note that independent of how we choose the resistor matrix $R(X(t), C)$ the following lemma holds.

Lemma 3.2. *If $X(t)$ follows the Physarum dynamics for an arbitrary $R(X(t), C)$ and (CC.2) holds for $X(0)$, then for all $v \in Null(X(0))$ we have $X(t)v = 0$. Moreover, for any time $0 \leq t$, $Null(X(0)) \subseteq Null(X(t))$ and $k(0) \geq k(t)$.*

Proof. Consider the function $X(t)v$ for $v \in Null(X(0))$. It can be characterized using the following differential equation:

$$\frac{d}{dt}X(t)v = Q(X(t))v - X(t)v = -X(t)v$$

The null-space condition in (8) imposes that $Q(X(t))v = 0$. This differential equation has the unique solution of form:

$$X(t)v = e^{-t}X(0)v = 0$$

That being said, for each $v \in Null(X(0))$ we know that $X(t)v = X(0)v = 0$ and $v \in Null(X(t))$. This in turn proves the lemma. $\square$

We will refer to $\tilde{u}_i(t)$ as the **generalized eigenvectors** and $\lambda_i(t)$ values as the **generalized eigenvalues**. Now consider the eigen-decomposition of C as $W_{r_C}\Psi W_{r_C}^T$ and the rank of C to be r_C. The $n \times r_C$ matrix W_{r_C} contains eigenvectors that are not in the null-space of C and the diagonal $r_C \times r_C$ matrix Ψ contains the non-zero eigenvalues of C. If we write all the eigenvectors of C in W such that the first r_C columns correspond to W_{r_C}, we know according to Lemma 3.2 and (CC.2) that $Null(C) \subseteq Null(X(0)) \subseteq Null(X(t))$, hence the $n - r_C$ last columns of W contain vectors both in the null-space of $X(t)$ and in C.

We will denote the eigen-decomposition of $\Psi^{\frac{1}{2}}W_{r_C}^T X(t) W_{r_C}\Psi^{\frac{1}{2}}$ by $W'(t)\Psi'(t)W'^T(t)$. Now consider the following set:

$$A = \{i \mid 1 \leq i \leq r_C \text{ and } \Psi'_{i,i}(0) \neq 0\},\ |A| = \text{rank of } X(0) = k_0$$

If we reduce the diagonal matrix $\Psi'(t)$ to a diagonal matrix $\Lambda(t)$ by only choosing the intersection of rows and columns in A from $\Psi'(t)$, and if we reduce the columns of $W'(t)$ to obtain $W'_{k_0}(t)$ which has columns that are indexed in A, we can then show that:

$$W'(t)\Psi'(t)W'^T(t) = W'_{k_0}(t)\Lambda(t){W'}_{k_0}^T(t).$$

Moreover,

$$\Psi^{-\frac{1}{2}}W'_{k_0}(t)\Lambda(t){W'}_{k_0}^T(t)\Psi^{-\frac{1}{2}} = \Psi^{-\frac{1}{2}}\Psi^{\frac{1}{2}}W_{r_C}^T X(t)W_{r_C}\Psi^{\frac{1}{2}}\Psi^{-\frac{1}{2}} = W_{r_C}^T X(t) W_{r_C},$$

and,

$$\begin{aligned}
W^T X(t) W &= \begin{bmatrix} W_{r_C}^T X(t) W_{r_C} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \Psi^{-\frac{1}{2}}W'_{k_0}(t)\Lambda(t){W'}_{k_0}^T(t)\Psi^{-\frac{1}{2}} & 0 \\ 0 & 0 \end{bmatrix} \\
\Rightarrow X(t) &= \underbrace{W_{r_C}\Psi^{-\frac{1}{2}}W'_{k_0}(t)}_{U(t)}\Lambda(t)\underbrace{{W'}_{k_0}^T(t)\Psi^{-\frac{1}{2}}W_{r_C}^T}_{U^T(t)}
\end{aligned}$$

Hence, $U(t)$ and $\Lambda(t)$ can diagonalize $X(t)$ such that $X(t) = U(t)\Lambda(t)U^T(t)$. We may also define $\tilde{U}(t)$ as an $n \times k_0$ matrix equal to $W_{r_C}\Psi^{\frac{1}{2}}W'_{k_0}(t)$. Now we will try to show that the characteristics pointed out in Theorem 3.1 hold for such a choice of $U(t), \Lambda(t)$, and $\tilde{U}(t)$.

Note that,

$$\tilde{U}^T(t)U(t) = U^T(t)\tilde{U}(t) = {W'}_{k_0}^T(t)^T\Psi^{\frac{1}{2}}W_{r_C}^T W_{r_C}\Psi^{-\frac{1}{2}}{W'}_{k_0}^T(t) = I$$

Hence,

$$X\tilde{U} = U\Lambda U^T\tilde{U} = U\Lambda$$

which means $X\tilde{u}_i = \lambda_i u_i$ and that the dot product of $\tilde{u}_i$ and u_j is equal to one if $i = j$ otherwise it is zero.

Note that according to Lemma 3.2, the null-space of $X(0)$ is always contained within the null-space of $X(t)$. One can characterize the null space of $X(0)$ using a basis as follows. If we denote the ith column of W by w_i and the ith column of W' by w'_i, then one part of such a basis is the basis of C according to (CC.2) which is characterized by:

$$\mathcal{B} = \{w_{r_C+1}, w_{r_C+2}, ..., w_n\}$$

Another part is characterized by $W_{r_C}\Psi^{-\frac{1}{2}}w'_i$ values for $k_0 < i \leq r_C$:

$$\mathcal{B}' =< W_{r_C}\Psi^{-\frac{1}{2}}{w'}_{k_0+1}, ..., W_{r_C}\Psi^{-\frac{1}{2}}{w'}_n >$$

Therefore, according to Lemma 3.2, $\mathcal{B}\cup\mathcal{B}'$ are linearly independent vectors that will always remain in $Null(X(t))$.

We will show $C^{\dagger}_{(k_0)}$ and $C_{(k_0)}$ are time invariant:

- $$\begin{aligned} C^{\dagger}_{(k_0)} = U(t)U^T(t) &= W_{r_C}\Psi^{-\frac{1}{2}}{W'}_{k_0}(t){W'}^T_{k_0}(t)\Psi^{-\frac{1}{2}}W^T_{r_C} \\ &= W_{r_C}\Psi^{-1}W^T_{r_C} - W_{r_C}\Psi^{-\frac{1}{2}}\Big(\sum_{i=k_0+1}^{r_C} {w'}_i{w'}^T_i\Big)\Psi^{-\frac{1}{2}}W^T_{r_C} \\ &= C^+ - \underbrace{\Big(\sum_{i=k_0+1}^{r_C} W_{r_C}\Psi^{-\frac{1}{2}}{w'}_i\Big)}_{H}\underbrace{\Big(\sum_{i=k_0+1}^{r_C} {w'}^T_i\Psi^{-\frac{1}{2}}W^T_{r_C}\Big)}_{H^T} \end{aligned}$$

- $$\begin{aligned} C_{(k_0)} = \tilde{U}(t)\tilde{U}^T(t) &= W_{r_C}\Psi^{\frac{1}{2}}{W'}_{k_0}(t){W'}^T_{k_0}(t)\Psi^{\frac{1}{2}}W^T_{r_C} \\ &= W_{r_C}\Psi W^T_{r_C} - W_{r_C}\Psi^{\frac{1}{2}}\Big(\sum_{i=k_0+1}^{r_C} {w'}_i{w'}^T_i\Big)\Psi^{\frac{1}{2}}W^T_{r_C} \\ &= C - \underbrace{\Big(\sum_{i=k_0+1}^{r_C} W_{r_C}\Psi^{\frac{1}{2}}{w'}_i\Big)}_{\tilde{H}}\underbrace{\Big(\sum_{i=k_0+1}^{r_C} {w'}^T_i\Psi^{\frac{1}{2}}W^T_{r_C}\Big)}_{\tilde{H}^T} \end{aligned}$$

H and $\tilde{H}$ are time invariant because ${w'}_i$ values for $i > k_0$ are time invariant. Hence $C^{\dagger}_{(k_0)}$ and $C_{(k_0)}$ are also time invariant.

Now we will prove (i) and (ii) in Theorem 3.1. We know that any $v \in Null(X(0))$ can be formulated using a linear combination of elements in $\mathcal{B}$ and $\mathcal{B}'$. We will prove that $U^T(t)v = 0$ for such v values:

- $v \in \mathcal{B} \Rightarrow U^T(t)v = {W'}^T_{k_0}(t)\Psi^{-\frac{1}{2}}\underbrace{W^T_{r_C}w_i}_{0 \text{ for } i>r_C} = 0.$
- $v \in \mathcal{B}' \Rightarrow U^T(t)v = {W'}^T_{k_0}(t)\Psi^{-\frac{1}{2}}W^T_{r_C}W_{r_C}\Psi^{-\frac{1}{2}}{w'}_i = \underbrace{{W'}^T_{k_0}(t){w'}_i}_{0 \text{ for } i>k_0} = 0$

This proves part (ii). On the other hand, for part (i) we can easily show that $\tilde{H}U^T = U\tilde{H}^T = 0$. This will give us the following representation for C:

$$C = \tilde{U}\tilde{U}^T + \tilde{H}\tilde{H}^T = C_{(k_0)} + \tilde{H}\tilde{H}^T$$

Now if we plug this into the formula below:

$$\begin{aligned} tr(CF(t)) &= tr(C_{(k_0)}F(t)) + tr(\tilde{H}\tilde{H}^TF(t)) = tr(C_{(k_0)}F(t)) + tr(\tilde{H}^TU\tilde{F}U^T\tilde{H}) = tr(C_{(k_0)}F(t)) \\ &= tr(\tilde{O}^TF(t)\tilde{O}) = tr(\tilde{U}(0)^TU(t)\tilde{F}(t)U^T(t)\tilde{U}(0)) \\ &= tr({W'}^T_{k_0}(0){W'}_{k_0}(t)\tilde{F}(t){W'}^T_{k_0}(t){W'}_{k_0}(0)) = tr({W'}^T_{k_0}(t){W'}_{k_0}(0){W'}^T_{k_0}(0){W'}_{k_0}(t)\tilde{F}(t)) \\ &= tr(\tilde{F}(t)) - tr(\tilde{F}(t)({W'}_{k_0}(t)^T\sum_{i=k_0+1}^{r_C}{w'}_i{w'}^T_i{W'}_{k_0}(t)) \qquad {W'}^T_k(t){w'}_i = 0 \text{ for } i > k_0 \\ &= tr(\tilde{F}(t)) \end{aligned}$$

The calculations in this section are viable and the following corollary summarizes them.

Corollary 3.3. *Consider C to have rank r_C with the eigen-decomposition $W_{r_C}\Psi W_{r_C}^T$. Assume $X(t)$ follows the Physarum dynamics in (7) with an arbitrary resistor for the update problem and that $rank(X(0)) = k_0$. Consider the eigen-decomposition of $\Psi^{-\frac{1}{2}}W_{r_C}^T X(t) W_{r_C}\Psi^{-\frac{1}{2}}$ to be $W'(t)\Psi'(t)W'^T(t)$, and A to be the set of non-zero elements on $\Psi'(0)$:*

$$A = \{i \mid 1 \le i \le r_C \text{ and } \Psi'_{i,i}(0) \neq 0\},\ |A| = k_0$$

Then one can write $U(t), \tilde{U}(t)$, and $\Lambda(t)$ according to the following scheme such that Theorem 3.1 holds and that $\Lambda(0) \succ 0$.

- *By reducing $W'(t)$ to the columns in A we may obtain $W'_{k_0}(t)$ and by reducing $\Psi'(t)$ to an $k_0 \times k_0$ diagonal matrix by only choosing the rows and columns of $\Psi'(t)$ that are in A we will obtain $\Lambda(t)$.*
- *Choose $U(t) = W_{r_C}\Psi^{-\frac{1}{2}}W'_{k_0}(t)$.*
- *Choose $\tilde{U}(t) = W_{r_C}\Psi^{\frac{1}{2}}W'_{k_0}(t)$.*

3.2 The Transformed Update Problem

The definition of $R(C, X(t))$ relies on the $U(t)$ and $\Lambda(t)$ values presented in the previous section. In this section, we will show an alternative **transformed** representation of the Update problem in (8). Given this transformed representation, we will obtain (8) by accurately defining R. The symbols used in this section will repeat numerous times throughout the document.

We will introduce some new symbols in order to define the transformed update problem. In the following, U is an arbitrary $n \times k$ matrix for $1 \le k \le n$ with independent columns and Λ is a diagonal $k \times k$ matrix.

- (Transformed resistor) $\tilde{R}(\Lambda) = \left(\frac{1}{2}\Lambda \otimes I + \frac{1}{2}I \otimes \Lambda\right)^+$, This matrix is a diagonal $k^2 \times k^2$ with zeros or $\frac{2}{\lambda_i + \lambda_j}$ as the $(i-1)k + j$th element on the diagonal.
- (Transformed Conductance) $\tilde{G}(\Lambda) = \tilde{R}^+$.
- $\tilde{A}_\ell(U) = U^T A_\ell U$
- $\tilde{\Omega}(U) = \begin{bmatrix} vec(\tilde{A}_1) & \dots & vec(\tilde{A}_m) \end{bmatrix}^T$

Using this new representation we can write the following transformed update problem which aims to optimize $\tilde{Q}$:

$$\min\{vec(\tilde{Q})^T \tilde{R}(\Lambda) vec(\tilde{Q}) \mid \tilde{\Omega}(U) \cdot vec(\tilde{Q}) = b,\ \tilde{Q} = \tilde{Q}^T\} \tag{9}$$

Hence, $\tilde{Q}$ can be viewed as a $k \times k$ matrix which is a function of Λ and U. When $U(t)$ and $\Lambda(t)$ are chosen with respect to $X(t)$ according to Theorem 3.1 we can calculate $Q(t)$ using the transformation $Q(t) = U(t)\tilde{Q}(t)U^T(t)$.

We can now define the resistor

$$R = (\tilde{U} \otimes \tilde{U})\tilde{R}(\tilde{U} \otimes \tilde{U})^T$$

and show an optimum solution $\tilde{Q}$ for the transformed update problem maps to an optimum solution Q for the original update problem.

To this end, one can show that the null-space constraints in (8) are inherently present for $Q = U\tilde{Q}U^T$. For $v \in Null(X(0))$, according to Theorem 3.1, $Q(t)$ can be viewed as $U(t)\tilde{Q}(t)U^T(t)$ and according to part (ii) $Q(t)v = 0$ inherently for these cases.

For the linear equality constraints:

$$tr(\tilde{A}_\ell\tilde{Q}) = tr(U^TA_\ell U\tilde{Q}) = tr(A_\ell U\tilde{Q}U^T) = tr(A_\ell Q)$$

which means that if $b_\ell = tr(\tilde{A}_\ell\tilde{Q}) = tr(A_\ell Q)$.

Finally, the objective values of the transformed update problem maps to the objective value of (8):

$$\begin{aligned} vec(\tilde{Q})^T\tilde{R}vec(\tilde{Q}) &= vec(\tilde{U}^TU\tilde{Q}U^T\tilde{U})^T\tilde{R}vec(\tilde{U}^TU\tilde{Q}U^T\tilde{U}) \\ &= vec(U\tilde{Q}U^T)^T(\tilde{U}\otimes\tilde{U})\tilde{R}(\tilde{U}^T\otimes\tilde{U}^T)vec(U\tilde{Q}U^T) \\ &= vec(Q)^TRvec(Q) \end{aligned}$$

To conclude, the optimum solution $\tilde{Q}$ of the transformed update problem has the following relation with Q:

$$Q = U\tilde{Q}U^T,\ \tilde{Q} = \tilde{U}^TQ\tilde{U} \tag{10}$$

3.3 Solving the Update Problem

We will use Lagrange multipliers to solve a relaxed version of (9) and will not consider the condition that $\tilde{Q}$ needs to be symmetric. Additionally, instead of minimizing $vec(\tilde{Q})^T\tilde{R}vec(\tilde{Q})$ one can minimize $\frac{1}{2}vec(\tilde{Q})^T\tilde{R}vec(\tilde{Q})$ which gives us the following Lagrange function with the Lagrange multipliers denoted by p. The derivative of this Lagrange function should vanish when $\tilde{Q}$ is optimum hence,

$$L(\tilde{Q},p) = \frac{1}{2}vec(\tilde{Q})^T\tilde{R}vec(\tilde{Q}) + p^T(\tilde{\Omega}\cdot vec(\tilde{Q}) - b) \Rightarrow \frac{\partial L(\tilde{Q},p)}{\partial vec(\tilde{Q})} = \tilde{R}vec(\tilde{Q}) - \tilde{\Omega}^Tp = 0$$

We will **only** solve the update problem for cases where $\Lambda \succ 0$. This means $\tilde{R}$ and $\tilde{G}$ are strictly positive definite and that they are non-singular. Thus one can write:

$$vec(\tilde{Q}) = \tilde{R}^{-1}\tilde{\Omega}^Tp = \tilde{G}\tilde{\Omega}^Tp \tag{11}$$

We will then multiply both sides of (11) by $\tilde{\Omega}$. $vec(\tilde{Q})$ is feasible in the linear constraints, hence, the left hand side will be equal to b while the right hand side will be $\tilde{\Omega}\tilde{R}\tilde{\Omega}^Tp$. We will define M as below:

$$M = \tilde{\Omega}\tilde{R}\tilde{\Omega}^T,\ Mp = b \tag{12}$$

We will solve the above linear system of equation in order to obtain the Lagrange multipliers and plug the resulting p in formula (11) to obtain $\tilde{Q}$. This $\tilde{Q}$ will be an optimum for the relaxed version of (9) without the $\tilde{Q} = \tilde{Q}^T$ constraint. However, we can show that $\tilde{Q}$ for the relaxed problem is in fact symmetric which means the optimum value of the relaxed update problem is an optimum for the original update problem in (9). The following shows $\tilde{Q}$ calculated by (11) is in fact symmetric because it can be written as a linear combination of symmetric matrices

$$\begin{aligned} & vec(\tilde{Q}) = \tilde{G}\tilde{\Omega}^Tp \\ & \Rightarrow \tilde{Q} = \sum_{\ell=1}^{m} p_\ell(\tilde{A}_\ell * \mathcal{P}) \text{ where } \mathcal{P}_{i,j} = \frac{\lambda_i+\lambda_j}{2} = \mathcal{P}_{j,i},\ \text{for } 1 \le i,j \le k \end{aligned}$$

The term $\tilde{A}_\ell * \mathcal{P}$ that appears above means element-wise multiplication of matrices $\tilde{A}_\ell$ and $\mathcal{P}$. We know that both of them are symmetric hence their element-wise multiplication will also be symmetric.

Now, we will consider $Mp = b$ which has answers of the form below:

$$\{M^+ b + p' \mid p' \in Null(M)\}$$

We will show that choosing any of the answers from the above set will result in the same $\tilde{Q}$ matrix. Note that $M = \tilde{\Omega}\tilde{G}\tilde{\Omega}^T$ is positive semi-definite and its null space can be characterized by vectors p' such that $p'^T M p' = 0$. Now consider $p'^T \tilde{\Omega}\tilde{G}\tilde{\Omega}^T p' = 0$. Since $\tilde{G} \succ 0$, the left hand side will be zero if and only if $\tilde{\Omega}^T p' = 0$. If we plugin an arbitrary answer from the set above to (11) we have

$$vec(\tilde{Q}) = \tilde{G}\tilde{\Omega}^T(M^+ b + p') = \tilde{G}(\tilde{\Omega}^T M^+ b + \underbrace{\tilde{\Omega}^T p'}_{0}) = \tilde{G}\tilde{\Omega}^T M^+ b.$$

which means $\tilde{Q}$ is calculated regardless of p'. Hence from now on we will consider $p = M^+ b$ and calculate $\tilde{Q}$ using the formula in (11). This will give us the optimum solution to (9) which can lead to the optimum of (8) via a reverse transformation shown in (10). However, all of these formulas and computations seem rather inefficient and we will try to come up with a formula to directly obtain Q.

3.4 Calculating Q Directly

Here, we will achieve simple formulas to calculate Q directly. We will use the prior knowledge in solving the transformed version of the update problem in (9) to obtain these formulas.

First and foremost, in order to calculate Q we will first formulate matrix M:

$$\begin{aligned} M &= \tilde{\Omega}\tilde{G}\tilde{\Omega}^T \\ \Rightarrow M_{i,j} &= vec(U^T A_i U)^T (\frac{1}{2} I \otimes \Lambda + \frac{1}{2}\Lambda \otimes I) vec(U^T A_j U) \\ &= tr(U^T A_i U \Lambda U^T A_j U) = tr(U U^T A_i X A_j) \\ &= tr(C^\dagger_{(k_0)} A_i X A_j) \end{aligned}$$

Note that one can use sparse matrix multiplication when $C^\dagger_{(k_0)}, X, A_i, A_j$ are all sparse matrices. By calculating M we can obtain $p = M^+ b$. Now recall (10), we will calculate Q using the p multipliers and the above formula.

$$\begin{aligned} vec(Q) &= vec(U\tilde{Q}U^T) = (U \otimes U) vec(\tilde{Q}) = (U \otimes U)\tilde{G}\tilde{\Omega}^T p \\ &= (U \otimes U)\tilde{G} vec(\sum_{\ell=1}^m p_\ell U^T \tilde{A}_\ell U) = (U \otimes U)\tilde{G} vec(U^T [\sum_{\ell=1}^m p_\ell A_\ell] U) \\ &= (U \otimes U)\tilde{G}(U^T \otimes U^T) vec(\sum_{\ell=1}^m p_\ell A_\ell) \end{aligned}$$

We can now define G as the **conductance** matrix according to the following:

$$G = (U \otimes U)\tilde{G}(U^T \otimes U^T) \tag{13}$$

this gives us the following formula for the vectorized notation of Q:

$$vec(Q) = G\Omega^T p. \tag{14}$$

We can also obtain a formula without any vectorization:

$$\begin{aligned}
vec(Q) &= (U \otimes U)(\frac{1}{2}I \otimes \Lambda + \frac{1}{2}\Lambda \otimes I)(U^T \otimes U^T)vec(\sum_{\ell=1}^{m} p_\ell A_\ell) \\
&= \big(\frac{1}{2}(UU^T) \otimes X + \frac{1}{2}X \otimes (UU^T)\big)vec(\sum_{\ell=1}^{m} p_\ell A_\ell) \\
&= \frac{1}{2}vec(C^\dagger_{(k_0)}[\sum_{\ell=1}^{m} p_\ell A_\ell]X + X[\sum_{\ell=1}^{m} p_\ell A_\ell]C^\dagger_{(k_0)}) \\
&=vec\big(\ \frac{1}{2}\sum_{\ell=1}^{m} p_\ell(C^\dagger_{(k_0)}A_\ell X + XA_\ell C^\dagger_{(k_0)})\big)
\end{aligned}$$

One may also obtain an efficient way to calculate Q using sparse matrix multiplications according to the formula below:

$$Q = \sum_{\ell=1}^{m} p_\ell(\frac{1}{2}C^\dagger_{(k_0)}A_\ell X + \frac{1}{2}XA_\ell C^\dagger_{(k_0)}) \tag{15}$$

4 Feasibility of $X(t)$

In this section, we will show that given (CC.1) to (CC.4) if $X(t)$ is a matrix function defined by (7) it will always remain feasible. In addition to that, we will prove that $rank(X(T)) = k_0$ for any finite time T.

The feasibility of $X(t)$ consists of two types of conditions:

- Linear constraints of the form $tr(A_\ell X(t)) = b_\ell$ which can be summarized in $\Omega \cdot vec(X) = b$.
- Positive semi-definiteness of $X(t)$, i.e., $X(t) \succeq 0$.

We will begin with the linear constraints. The following lemma is useful in this respect:

Lemma 4.1. *If $X(t)$ is characterized by the dynamics in* (7)*, and if we define*

$$\Delta_\ell(t) = tr(A_\ell X(t)) - b_\ell,$$

then $\Delta_\ell(t) = e^{-t}\Delta_\ell(0)$. That is to say, when we choose $X(0)$ to be feasible in the linear constraints, $X(t)$ remains linearly feasible and for a non-feasible $X(0)$, $X(t)$ converges to a linearly feasible solution exponentially fast.

Proof. We can write the following differential equation:

$$\begin{aligned}
\frac{d}{dt}\big(tr(A_\ell X(t)) - b_\ell\big) =& tr(A_\ell \dot{X}(t)) = tr(A_\ell Q) - tr(A_\ell X) \\
=& b_\ell - tr(A_\ell X) = -\big(tr(A_\ell X(t)) - b_\ell\big)
\end{aligned}$$

The function $\Delta_\ell(t) = \big(tr(A_\ell X(t)) - b_\ell\big)$ can be characterized by $\dot{\Delta}_\ell(t) = -\Delta_\ell(t)$ which is a differential equation with answer $\Delta_\ell(t) = e^{-t}\Delta_\ell(0)$; this means:

$$tr(A_\ell X(t)) - b_\ell = e^{-t}(tr(A_\ell X(0)) - b_\ell).$$

In case $X(0)$ is feasible in the linear constraints, $\Delta(0) = 0$ and $\Delta(t) = 0$. In other words, $X(t)$ will remain linearly feasible for any time $t \geq 0$. If $X(0)$ is not linearly feasible, the e^{-t} coefficient ensures $\Delta(t)$ converges to zero in infinity, i.e., $X(t)$ will converge to a linearly feasible solution exponentially. □

Now that we have proven linear feasibility, we will try to prove that $X(t)$ remains positive semi-definite throughout the dynamic if (CC.1), (CC.2), (CC.3), and (CC.4) hold. However, for this to hold we will need the additional condition that $C^{\dagger}_{(k_0)}$ is also feasible. In Section 6 we will show that (CC.3) and (CC.4) hold in the augmented case.

To prove this, we introduce the work function

$$W(X(t)) = \ln\det(\tilde{O}^T X(t)\tilde{O}) \tag{16}$$

This function explodes when $\tilde{O}^T X(t)\tilde{O}$ obtains a zero eigenvalue. To show $X(t) \succeq 0$, we will show this function does not explode and that $\tilde{O}^T X(T)\tilde{O} \succ 0$ for any finite time T.

We know that $W(X(t))$ is continuously differentiable because it is obtained by taking the log determinant of a continuously differentiable matrix $\tilde{O}^T X(t)\tilde{O}$. Now we will try to write down its derivative in order to show $W(X(t))$ will not explode for any finite time.

To that end, we will also need to facilitate the following theorem.

Theorem 4.2. *(Theorem 6.8 [8]) Assume that $X(t)$ is symmetric and continuously differentiable in an interval I of t. Then there exists n continuously differentiable functions $\mu_i(t)$ on I that represent the repeated eigenvalues of $X(t)$.*

Lemma 4.3. *If (CC.1), (CC.2), (CC.3), and (CC.4) hold,*

- *While $\tilde{O}^T X(t)\tilde{O} \succ 0$, $\frac{d}{dt}W(X(t))$ is lower-bounded with a constant.*
- *For any finite time T, $W(T)$ does not explode to $-\infty$. Moreover, $\tilde{O}^T X(T)\tilde{O}$ remains strictly positive definite for any finite time T and that $\lambda_i(T)$ values will also always remain positive.*
- *$X(t) \succeq 0$ for any time t.*

Proof.

- While $\tilde{O}^T X(t)\tilde{O} \succ 0$ then $\tilde{O}^T U(t)\Lambda(t)U^T(t)\tilde{O} \succ 0$, $\Lambda(t) \succ 0$, and $\tilde{O}^T U(t)$ is non-singular. Hence $rank(X(t)) = k_0$ and we may use the formulas in Section 3.

$$\begin{aligned}
&\frac{d}{dt}\ln\det(\tilde{O}^T X(t)\tilde{O}) \\
&= tr\big((\tilde{O}^T X(t)\tilde{O})^{-1}(\frac{d}{dt}\tilde{O}^T X(t)\tilde{O})\big) = tr\big((\tilde{O}^T X(t)\tilde{O})^{-1}(\tilde{O}^T Q(t)\tilde{O})\big) - k_0 \\
&= tr\big([\tilde{O}^T U(t)]^{-T}\Lambda^{-1}(t)[\tilde{O}^T U(t)]^{-1}[\tilde{O}^T U(t)]\tilde{Q}(t)[\tilde{O}^T U(t)]^T\big) - k_0 \\
&= tr(\Lambda^{-1}\tilde{Q}) - k_0 = vec(\Lambda^{-1})^T\tilde{G}\tilde{\Omega}^T p - k_0 \\
&= vec(\Lambda^{-1})^T\big(\frac{1}{2}I\otimes\Lambda + \frac{1}{2}\Lambda\otimes I\big)\tilde{\Omega}^T p - k_0 \\
&= vec(\frac{1}{2}\Lambda^{-1}\Lambda + \frac{1}{2}\Lambda\Lambda^{-1})\tilde{\Omega}^T p - k_0 \\
&= vec(I)^T\tilde{\Omega}^T p - k_0 = \sum_{\ell=1}^m p_\ell vec(I)^T vec(U^T A_\ell U) - k_0 = \sum_{\ell=1}^m p_\ell tr(U^T A_\ell U) - k_0 \\
&= \sum_{\ell=1}^m p_\ell tr(A_\ell U U^T) - k_0 \qquad U(t)U^T(t) = C^{\dagger}_{(k_0)} \text{ is feasible} \\
&= b^T p - k_0 = p^T M p - k_0 \geq -k_0 \qquad M \text{ is positive semi-definite}
\end{aligned}$$

- $W(X(t))$ exploding to $-\infty$ is equivalent to $\tilde{O}^T X(t)\tilde{O}$ obtaining a zero eigenvalue. We know that $\tilde{O}^T X(0)\tilde{O} = \Lambda(0) \succ 0$ for a positive definite starting point. According to the previous part the work function will be lower bounded by $-k_0$ for some time.

  Consider the first time T where $\tilde{O}^T X(t)\tilde{O}$ is not positive definite. At this time $\tilde{O}^T X(T)\tilde{O}$ should obtain a zero eigenvalue; otherwise according to Theorem 4.2 if this matrix obtains some negative eigenvalues there will be a time $T' < T$ where these eigenvalues pass zero. Since according to the first part $W(X(t)) \geq -k_0$ for $t \in [0, T)$ then $\lim_{t\to T} W(X(t)) = W(X(T)) \geq W(X(0)) - k_0 T$ which is a finite value. However, $W(X(T))$ is infinitely small which contradicts with the fact that $W(X(t))$ is continuously differentiable. The contradictions yields the statement of this part and states that for any finite time $\tilde{O}^T X(t)\tilde{O}$ and $\Lambda(t)$ should be strictly positive definite. This means none of the $\lambda_i(t)$ values will become non-positive, otherwise $\tilde{O}^T X(t)\tilde{O}$ will not remain strictly positive definite.

- Note that $X(t) = U(t)\Lambda(t)U^T(t)$ and since $\Lambda(t) \succ 0$ for any finite time t and $\Lambda(t) \succeq 0$ for any time t, then we can easily conclude $X(t) \succeq 0$.

□

Lemma 4.3 proves that $\Lambda(t) \succ 0$ for any finite time meaning $X(t) = U(t)\Lambda(t)U^T(t)$ will also remain positive semi-definite. This Lemma coupled with Lemma 4.1 prove that $X(t)$ will always remain feasible if (CC.1) to (CC.4) hold.

Corollary 4.4. *If (CC.1), (CC.2), (CC.3), and (CC.4) hold and if $\Lambda(t)$ represents the generalized eigenvalues of $X(t)$, then $\Lambda(T)$ always remains strictly positive definite for any finite time T and $X(t)$ will remains feasible for any time "t", i.e., feasible in the linear constraints and positive semi-definite. Moreover, $\tilde{R}(\Lambda(t))$ will always remain strictly positive definite for any finite time.*

5 Convergence of the Dynamic

In this section we will first show that the dynamics converges to an equilibrium and that

$$\lim_{t\to\infty} [X(t) - Q(X(t))] = 0.$$

We will then discuss that the Lagrange multipliers p obtained in the update problem via the formula (12) can be viewed as a dual candidate solution for a specific primal-dual pair of SDP. We will only use weak duality in this case.

First and foremost, we need to prove the following lemma:

Lemma 5.1. *For $X(t)$ following the Physarum dynamics when (CC.1), (CC.2), (CC.3), and (CC.4) hold. Then for any finite time T:*

(i) $\tilde{R}(\Lambda(T))vec(\Lambda(T)) = vec(I)$.

(ii) $vec(X(T))^T R(X(T))vec(X(T)) = vec(\Lambda(T))^T \tilde{R}(\Lambda(T))vec(\Lambda(T)) = tr(\Lambda(T)) = tr(CX(T))$.

Proof.

(i) According to Corollary 4.4 $\Lambda(T)$ and $\tilde{R}(\Lambda(T))$ will remain strictly positive definite. We will calculate $\tilde{R}vec(\Lambda)$ at time t element by element. The $(i-1)k+j$th row of $\tilde{R}$ contains zeros except for the $((i-1)k+j, (i-1)k+j)$th element which corresponds to the (i, j) the element in Λ. The respective elements will be $\frac{2}{\lambda_i+\lambda_j}$ and $(\Lambda)_{i,j}$ because $\tilde{R}$ is non-singular and hence will not have any zero elements on the diagonal. In case $i \neq j$, the respective element in Λ is zero. Otherwise the multiplication will be $\frac{2}{\lambda_i+\lambda_i} \times \lambda_i = 1$. That being said,

$$[\tilde{R}vec(\Lambda)]_{(i-1)k+j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{o.w.} \end{cases}$$

(ii)

$$\begin{aligned} vec(X)^T Rvec(X) &= vec(X)^T (\tilde{U} \otimes \tilde{U}) \tilde{R} (\tilde{U}^T \otimes \tilde{U}^T) vec(X) \\ &= vec(\tilde{U}^T X \tilde{U})^T \tilde{R} vec(\tilde{U}^T X \tilde{U}) \\ &= vec(\tilde{U}^T U X U^T \tilde{U})^T \tilde{R} vec(\tilde{U}^T U X U^T \tilde{U}) = vec(\Lambda)^T \tilde{R} vec(\Lambda) \\ &= vec(I)^T vec(\Lambda) = tr(\Lambda) \qquad \text{According to the last part} \\ &= tr(\tilde{U}^T U \Lambda U^T \tilde{U}) = tr(\tilde{U}\tilde{U}^T X) = tr(C_{(k_0)} X) \end{aligned}$$

According to Theorem 3.1 part (i) $tr(C_{(k_0)}X) = tr(CX)$.

□

We will then prove that the dynamic converges to some equilibrium:

Lemma 5.2. *(Reaching Equilibrium) If $X(t)$ is a matrix function following the Physarum dynamics in (7) and (CC.1) to (CC.4) all hold, we will define the function $\mathcal{L}(X(t)) = tr(CX(t))$. Then,*

(i) $\frac{d}{dt}\mathcal{L}(X(t)) \leq 0$.

(ii) $\frac{d}{dt}\mathcal{L}(X(t))$ *will become* 0 *only if* $X(t) = Q(X(t))$.

(iii) The dynamic in (7) *converges to an equilibrium point, i.e.,* $\underset{t\to\infty}{X(t)} = \underset{t\to\infty}{Q(X(t))}$.

Proof.

(i) we can then upper-bound the derivative of this function as follows:

$$\begin{aligned} \frac{d}{dt}\mathcal{L}(X(t)) &= \frac{d}{dt} tr(CX(t)) \qquad \text{according to Theorem 3.1 part (i)} \\ &= tr(\tilde{O}^T (Q(t) - X(t)) \tilde{O}) = tr(\tilde{O}^T Q(t) \tilde{O}) - tr(\tilde{O}^T X(t) \tilde{O}) \\ &= tr(\tilde{Q}(t)) - tr(\Lambda(t)) \\ &= vec(I)^T vec(\tilde{Q}) - tr(\Lambda) \qquad \text{using Lemma 5.1} \\ &= vec(\Lambda)^T \tilde{R} vec(\tilde{Q}) - tr(\Lambda) \\ &= [\tilde{R}^{\frac{1}{2}} vec(\Lambda)]^T [\tilde{R}^{\frac{1}{2}} vec(\tilde{Q})] - tr(\Lambda) \qquad \text{Analogous to Cauchy-Schwartz} \\ &\leq \sqrt{vec(\Lambda)^T \tilde{R} vec(\Lambda)} \sqrt{vec(\tilde{Q})^T \tilde{R} vec(\tilde{Q})} - tr(\Lambda) \qquad \text{Optimality of } \tilde{Q} \\ &\leq vec(\Lambda)^T \tilde{R} vec(\Lambda) - tr(\Lambda) \qquad \text{Using Lemma 5.1} \\ &= tr(\Lambda) - tr(\Lambda) = 0 \end{aligned}$$

We have used the fact that if $X(t)$ remains feasible for each time according to Corollary 4.4 then $\Lambda(t)$ complies to the constraints in the simplified transformed update problem in (9). Note that:

$$\forall \ell \in [m] : b_\ell = tr(A_\ell X(t)) = tr(A_\ell U \Lambda U^T) = tr(U^T A_\ell U \Lambda) = tr(\tilde{A}_\ell \Lambda)$$

This gives us the inequality $vec(\tilde{Q})^T \tilde{R} vec(\tilde{Q}) \leq vec(\Lambda)^T \tilde{R} vec(\Lambda)$ because Λ complies in all the constraints of the update problem in (9) but $\tilde{Q}$ is the optimum solution for such matrices.

(ii) The above inequality becomes equal if and only if both the Cauchy-Schwartz and

$$vec(\Lambda)^T \tilde{R} vec(\Lambda) \leq vec(\tilde{Q})^T \tilde{R} vec(\tilde{Q})$$

become equality. The Cauchy-Schwartz becomes an equality only if $\tilde{R}^{\frac{1}{2}} vec(\Lambda)$ and $\tilde{R}^{\frac{1}{2}} vec(\tilde{Q})$ are linearly dependent, i.e, $\tilde{R}^{\frac{1}{2}} vec(\Lambda) = g\tilde{R}^{\frac{1}{2}} vec(\tilde{Q})$. We also know that equality holds only if Λ is an optimal solution for the shrunken transformed update problem. That being said, the objective obtained from Λ and $\tilde{Q}$ should be equal.

With that in mind, g should be equal to one otherwise the energies will differ. Thus,

$$\tilde{R}^{\frac{1}{2}} vec(\Lambda) = \tilde{R}^{\frac{1}{2}} vec(\tilde{Q}).$$

Since $\tilde{R}$ is non-singular for any finite time, Λ and $\tilde{Q}$ should be equal. This leads to X and Q being equal because:

$$Q - X = U(\tilde{Q} - \Lambda)U^T = 0$$

(iii) We know that $\mathcal{L}$ at time 0 is finite and its derivative is negative. The derivative remains negative as long as $X \neq Q$, we also know for a fact that $\mathcal{L}(X(t)) = tr(CX(t))$ is lower bounded by zero because $X(t)$ always remains positive semi-definite. Hence, $X(t)$ and $Q(t)$ become arbitrarily close and the dynamic approaches an equilibrium.

□

That being said, the dynamic might have trajectories to multiple equilibrium points. If we define the set EQ as:

$$EQ = \{(U, \Lambda) \mid \tilde{Q}(U, \Lambda) = \Lambda \text{ and } X = U\Lambda U^T \text{ is feasible for the SDP constraints and } \Lambda \succ 0\}$$

then there exists $(U, \Lambda) \in EQ$ such that $X(t)$ converges to $U\Lambda U^T$.

We propose the following conjecture which is analogous to Theorem 1.1 of the LP part 1:

Conjecture 5.3. *For $X(t)$ following the Physarum dynamic in* (7) *such that all (CC.1) to (CC.4) hold. If it converges to $U\Lambda U^T$ for $(U, \Lambda) \in EQ$, then there exists $\hat{X}$ such that $U\hat{X}U^T$ is the optimum to the following SDP:*

$$\min\{tr(CF) \mid \Omega \cdot vec(F) = b,\ F \succeq 0,\ Null(X(0)) \subseteq Null(F)\}$$

We should then prove the following lemma which shows useful properties for $(U, \Lambda) \in EQ$.

Lemma 5.4. *For $(U, \Lambda) \in EQ$ if $p(U, \Lambda)$ resembles the Lagrange multipliers calculated with respect to a transformed update problem parameterized by U and Λ, then:*

(i) $tr(\Lambda) = b^T p(U, \Lambda)$

(ii) $I = \sum_{\ell}^{m} p_\ell U^T A_\ell U$

Proof. We know that for $(U, \Lambda) \in EQ$, $\Lambda = \tilde{Q}$ and that $U\Lambda U^T$ is feasible meaning:

$$b = \Omega \cdot vec(U\Lambda U^T) = \tilde{\Omega}(U) vec(\Lambda)$$

(i)

$$\begin{aligned} tr(\Lambda) = tr(\tilde{Q}) &= vec(I)^T vec(\tilde{Q}) = vec(I)^T \tilde{G}\tilde{\Omega}^T p \\ &= vec(I)^T \big(\frac{1}{2} I \otimes \Lambda + \frac{1}{2}\Lambda \otimes I\big)\tilde{\Omega}^T p = vec(\Lambda)^T \tilde{\Omega}^T p \\ &= b^T p \end{aligned}$$

(ii) Consider $\tilde{R}vec(\tilde{Q})$ and $\tilde{R}vec(\Lambda)$ which are equal. The latter is I according to Lemma 5.1 and the former is $\tilde{\Omega}^T p$ if we plug in the formula $vec(\tilde{Q}) = \tilde{R}^{-1}\tilde{\Omega}^T p$. Hence,

$$vec(I) = \tilde{\Omega}^T p = vec(\sum_{\ell}^{m} p_\ell U^T A_\ell U)$$

□

Assume $X(t)$ converges to $U_{eq}\Lambda_{eq}U_{eq}^T$ for $(U_{eq}, \Lambda_{eq}) \in EQ$, now we will try to prove Claim 2.1 by considering the following primal dual pair:

$$\begin{aligned} &\min\{tr(\hat{X}) \mid \forall \ell \in [m] : tr(U_{eq}^T A_\ell U_{eq}\hat{X}) = b_\ell,\ \hat{X} \succeq 0\} \\ &\max\{b^T y \mid I - \sum_{\ell=1}^{m} y_\ell U_{eq}^T A_\ell U_{eq} \succeq 0\} \end{aligned} \tag{17}$$

According to Conjecture 5.3 one may instead solve the primal in (17) to obtain the optimal solution for Claim 2.1. In the end an optimal solution X^* for (17) will map to an optimal solution $U_{eq}X^*U_{eq}^T$ for Claim 2.1 because all of the properties for $U(t)$ mentioned in Theorem 3.1 hold for U_{eq}:

$$tr(A_\ell \underbrace{U_{eq}X^*U_{eq}^T}_{F}) = tr(U_{eq}^T A_\ell U_{eq} X^*)$$

$$tr(C\underbrace{U_{eq}X^*U_{eq}^T}_{F}) = tr(C_{(k_0)}U_{eq}X^*U_{eq}^T) = tr(U_{eq}^T\tilde{U}_{eq}\tilde{U}_{eq}^T U_{eq}X^*) = tr(X^*)$$

$$X^* \succeq 0 \Longleftrightarrow U_{eq}X^*U_{eq}^T \succeq 0$$

According to Theorem 3.1 the null-space condition of the form $Null(X(0)) \subseteq Null(F)$ will be inherently present for our dynamics because F can be written as $U_{eq}X^*U_{eq}^T$. That being said, we will prove that Λ_{eq} and $p(U_{eq}, \Lambda_{eq})$ form the optimum primal and dual for (17) meaning $U_{eq}\Lambda_{eq}U_{eq}^T$ is the optimum for Claim 2.1.

We may infer using Lemma 5.4. Note that p_{eq} and Λ_{eq} are dual and primal feasible candidates according to part (ii). Moreover, according to weak duality $p^T b \leq tr(\Lambda)$ for any feasible primal and dual. However, according to Lemma 5.4 part (ii) these objectives become equal in the equilibrium point meaning that both produce the optimum for their objectives.

6 Augmentation

Here, we will introduce the augmented SDP and show in this case $X(0)$ can be chosen such that all of (CC.1), (CC.2), (CC.3), and (CC.4) hold. We will then see how the solution to the augmented problem relates to the original problem.

To this end, we augment all matrices in (5) by a further row and column. We define $\bar{C}$ and $\bar{A}_\ell$s as below using a certain γ:

$$\bar{C} = \begin{pmatrix} \gamma C & 0 \\ 0 & 1 \end{pmatrix}, \bar{A}_i = \begin{pmatrix} A_i & 0 \\ 0 & \alpha_i \end{pmatrix} \quad , \quad \alpha_i = b_i - \frac{tr(A_i C^+)}{\gamma}$$

where C^+ is the Moore-Penrose pseudo-inverse.

We will then define the following SDP as the **Augmented SDP** and try to solve it:

$$\min\{tr(\bar{C}\bar{X}) : tr(\bar{A}_\ell\bar{X}) = b_\ell \forall \ell \in [m],\ \bar{X} \succeq 0\}. \tag{18}$$

We will also refer to an arbitrary matrix T as **augmented** if all entries of the right column and bottom row except bottom right entry are zero. It can be seen that the augmented SDP contains augmented matrices $\bar{C}$, $\bar{X}$, $\bar{A}_\ell$. We claim that solving (18) yields the answer to (5):

Conjecture 6.1. *There exists an answer $\bar{X}_{opt}$ to the augmented SDP in (18) if γ is chosen small enough, such that $(\bar{X}_{opt})_{(1:n),(1:n)}$ which contains the first n rows and columns of $\bar{X}_{opt}$ is also the solution to our base SDP problem in Claim 2.1. That being said, the Physarum dynamics introduced in (7) converges to such a $\bar{X}_{opt}$ in the augmented case.*

We will differentiate each of the augmented matrices and the normal matrices using a "bar" symbol. The Physarum dynamic in the augmented case is shown by:

$$\dot{\bar{X}}(t) = \bar{Q}(\bar{X}(t)) - \bar{X}(t).$$

If we choose $\bar{X}(0) = \bar{C}^+$, then all (CC.1), (CC.2), (CC.3), and (CC.4) hold:

- (CC.1): C is positive semi-definite hence $\bar{C}$ and $\bar{C}^+ = X(0)$ are also positive semi-definite.
- (CC.2): Since $Null(\bar{C}) = Null(\bar{C}^+)$ and $X(0) = \bar{C}^+$ we know that $Null(\bar{X}(0)) = Null(\bar{C})$.
- (CC.3): If $\bar{X}(0) = \bar{C}^+$ then the eigen-decomposition $\Psi^{\frac{1}{2}} W_{r_{\bar{C}}}^T X(0) W_{r_{\bar{C}}} \Psi^{\frac{1}{2}}$ wille be:

$$\Psi^{\frac{1}{2}} W_{r_{\bar{C}}}^T \bar{C}^+ W_{r_{\bar{C}}} \Psi^{\frac{1}{2}} = \Psi^{\frac{1}{2}} W_{r_{\bar{C}}}^T W_{r_{\bar{C}}} \Psi^{-\frac{1}{2}} W_{r_{\bar{C}}}^T W_{r_{\bar{C}}} \Psi^{\frac{1}{2}} = I$$

  meaning $U(0) = W_{r_{\bar{C}}} \Psi^{-\frac{1}{2}}$ and that:

$$\bar{C}^{\dagger}_{(k_0)} = U(t)U^T(t) = U(0)U^T(0) = W_{r_{\bar{C}}} \Psi^{-\frac{1}{2}} \Psi^{-\frac{1}{2}} W_{r_{\bar{C}}}^T = \bar{C}^+$$

  So in the augmented case $\bar{C}^+ = \bar{C}^{\dagger}_{(k_0)}$ and we will show that $\bar{C}^+$ is feasible:

$$tr(\bar{A}_\ell \bar{C}^+) = tr(\frac{A_\ell C^+}{\gamma}) + \alpha_\ell = tr(\frac{A_\ell C^+}{\gamma}) + \left[b_\ell - tr(\frac{A_\ell C^+}{\gamma})\right] = b_\ell$$

- (CC.4): $\bar{X}(0) = \bar{C}^+$ and it is feasible according to the previous part.

All the formulas previously discussed in Section 3 hold when we put a bar symbol above the matrices. According to (15) we can write the following. Note that in the augmented case $\bar{C}^{\dagger}_{(k_0)} = \bar{C}^+$:

$$\bar{Q} = \sum_{\ell=1}^{m} p_\ell (\frac{1}{2} \bar{C}^+ \bar{A}_\ell \bar{X}(t) + \frac{1}{2} \bar{X}(t) \bar{A}_\ell \bar{C}^+). \tag{19}$$

We will show that given the current dynamics, $X(t)$ will always remain augmented.

Lemma 6.2. *Given the dynamics in (7) in augmented case where $X(0) = C^+$, $X(t)$ will always remain augmented.*

Proof. Consider $\bar{X}(t) = \begin{pmatrix} X(t) & x(t) \\ x(t)^T & \beta(t) \end{pmatrix}$ then according to (19),

$$\bar{Q}(\bar{X}(t)) = \sum_{\ell=1}^{m} p_\ell (\frac{1}{2} \begin{pmatrix} \frac{1}{\gamma} C^+ & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A_\ell & 0 \\ 0 & \alpha_\ell \end{pmatrix} \begin{pmatrix} X(t) & x(t) \\ x(t)^T & \beta(t) \end{pmatrix} + \frac{1}{2} \begin{pmatrix} X(t) & x(t) \\ x(t)^T & \beta(t) \end{pmatrix} \begin{pmatrix} A_\ell & 0 \\ 0 & \alpha_\ell \end{pmatrix} \begin{pmatrix} \frac{1}{\gamma} C^+ & 0 \\ 0 & 1 \end{pmatrix})$$

We will show that $x(t) = 0$ for each time t. For this, we will consider the following differential equation for $x(t)$:

$$\begin{aligned}\dot{x}(t) &= \sum_{\ell=1}^{m} p_\ell(\frac{1}{2\gamma}C^+ A_\ell x(t) + \frac{1}{2\gamma}\alpha_\ell x(t)) \\ &= \big[\sum_{\ell=1}^{m} p_\ell(\frac{1}{2\gamma}C^+ A_\ell + \frac{1}{2\gamma}\alpha_\ell I)\big] x(t) = Px(t)\end{aligned}$$

We can write the answer to a matrix differential equation of the form $\dot{x}(t) = Px(t)$ as $x(t) = e^{Pt}x(0)$. We also know that $x(0)$ is the upper right $n \times 1$ vector of $\bar{C}^+$ which is zero. Hence $x(t) = 0$ and $X(t)$ remains augmented. $\square$

The multiplication and addition of each two augmented matrices yield an augmented matrix. Hence both $\bar{Q}(t)$ and $\bar{X}(t)$ can be considered as augmented matrices. We will write them as below:

$$\bar{X}(t) = \begin{pmatrix} X(t) & 0 \\ 0 & \beta(t) \end{pmatrix}, \bar{Q}(t) = \begin{pmatrix} Q(X(t)) & 0 \\ 0 & b^T p(U(t), \Lambda(t)) \end{pmatrix}$$

Each feasible solution of the original problem maps to a feasible solution X of the modified problem by adding a row and a column with all zeros. The objective value of such an $\bar{X}$ is then scaled by γ, i.e., $tr(\bar{C}\bar{X}) = \gamma tr(CX)$. By an appropriate choice of γ, these objective values can be made smaller than the objective value of $\bar{C}^+$, which is a constant. Hence we have the following:

$$tr(\bar{C}\bar{X}_{opt}) \leq \gamma tr(CX_{opt}) \tag{20}$$

In the augmented case, we have proven that the dynamic converges to an optimal solution. If we can somehow obtain that $\lim_{t\to\infty}\beta(t) = 0$, we will know that,

$$\begin{aligned} tr(\bar{C}\bar{X}_{opt}) =& tr(\bar{C}\bar{X}(\infty)) = \gamma tr(C\bar{X}(\infty)_{(1:n),(1:n)}) + \beta(\infty) \\ =& \gamma tr(C\bar{X}_{(1:n),(1:n)}) \geq \gamma tr(CX_{opt}) \end{aligned} \tag{21}$$

So by considering both (20) and (21) we can show that $tr(\bar{C}\bar{X}_{opt}) = \gamma tr(CX_{opt})$ and $\bar{X}_{(1:n),(1:n)}$ is an optimum solution for our original problem.

Conjecture 6.3. *In the augmented setting, if $\beta(t)$ is the lower-right corner of $\bar{X}(t)$ in each step, then $\lim_{t\to\infty}\beta(t) = 0$.*

If we consider the work function for the augmented problem, we know that $\beta(t)$ maps to a generalized eigenvalue and the respective generalized eigenvector is the standard vector e_{n+1}. That being said, the work function defined in (16), would go to $-\infty$ if $\beta(t) \to 0$. This in fact holds:

Lemma 6.4. *If $W(X(t))$ is the work function defined in (16) for the augmented problem with a small enough γ such that $\bar{C}^+$ is not the optimum solution, and $\bar{X}(0) = \bar{C}^+$, then $\frac{d}{dt}W(X(t)) < 0$.*

Proof. Using Lemma 4.3 we know that $\frac{d}{dt}W(X(t)) = b^T p - k_0$. Hence,

$$\begin{aligned}\frac{d}{dt}W(X(t)) &= b^T p - k_0 \qquad \bar{X}(t) \text{ is feasible} \\ &= vec(\Lambda)^T \tilde{\Omega}^T p - k_0 \\ &= vec(\Lambda)^T \tilde{R}^T vec(\tilde{Q}) - k_0 \qquad \text{Using Lemma 5.1} \\ &= vec(I)^T vec(\tilde{Q}) - k_0 = tr(\tilde{Q}) - k_0 \qquad \text{Using Lemma 5.2} \\ &\leq tr(\Lambda) - k_0 = \mathcal{L}(t) - k_0 \leq \mathcal{L}(0) - k_0 = tr(\bar{C}\bar{X}(0)) - k_0 = tr(\bar{C}\bar{C}^+) - k_0 \leq 0\end{aligned}$$

The inequality will only be an equality when at time 0 we converge; in other words, $\bar{X}(0) = \bar{C}$ is the optimum. For a small enough choice of γ this is not the case, hence the inequality is strict. $\square$

The above lemma shows that the work function has a negative derivative at each point of time. Hence $W(t)$ will go to $-\infty$ eventually. This means that one of the diagonal elements of $\Lambda(t)$ converges to zero which hints that maybe $\beta(t)$ is that certain generalized eigenvalue. To prove Conjecture 6.3 one might consider what happens if some other eigenvalue of $\tilde{O}^T\bar{X}(t)\tilde{O}$ converges zero.

7 Algorithm and Experimental Results

In this section, we will present an algorithm which encompasses all which we have presented until now. At first, we will present an algorithm that can solve positive SDPs given the conditions (CC.1) to (CC.4). Afterward, we will try to improve the algorithm. To that end, we will need to make the algorithm numerically stable. One major problem in these cases is M becoming ill-conditioned. Note that:

$$M = \tilde{\Omega}\big(\frac{1}{2}I \otimes \Lambda + \frac{1}{2}\Lambda \otimes I\big)^{-1}\tilde{\Omega}^T$$

Hence when very small λ_i values appear the dynamic tends to become numerically unstable. We will introduce a method to work around this issue and obtain a modified algorithm.

In the end, we will gather our experimental results and also experiment with the case that the problem is not augmented. The results interestingly suggest despite not having a formal proof (CC.3) and (CC.4) are not necessary for the dynamic to converge to the optimum. The modified algorithm presents a method that does not even augment the problem.

To simulate the continuous dynamic in an algorithm, we will discretize it using the following formula:

$$X(t+1) = hQ(X(t)) + (1-h)X(t) \tag{22}$$

This will simulate the dynamics in (7) when $h \to 0$. That is to say, h should be chosen infinitely small or just simply small enough for practical goals.

The algorithms in this section are implemented in the GitHub repository [1] and tested thoroughly using tests generated in the SDPLib standard format [9]. All the tests in the original SDPLib contained indefinite C matrices, hence we created a data set of 140 SDPs with different specification to validate our solver. In all these tests, C is chosen as a strictly positive definite matrix and the SDP is positive.

7.1 Dataset

Table 1 shows a set of 140 positive SDP samples. The tests are generated using the two following schemes:

- **Random-Tests:** These tests are generated randomly. Given a certain n and m - the size of matrices and the number of conditions - a random positive definite matrix C and random symmetric matrices A_ℓ are generated. The datasets which are created using this scheme consist of 'testset1' to 'testset3', 'large1', and 'large2'.
- **Vertex-Cover:** To generate these datasets we create a random graph with n vertices numbered from 1 to n and m edges. We will then create an SDP problem which is able to approximate the minimum vertex cover problem. For each random graph with n vertices C will be equal to the identity matrix of size $(n+1)\times(n+1)$. Then the following linear condition matrices will be generated:

[1] `https://github.com/HamidrezaKmK/PhysarumSDPSolver`

- For each vertex v in the graph an A_ℓ will be created according to the following:

$$A_{i,j} = \begin{cases} -1 & \text{if } min(i,j) = 1 \text{ and } max(i,j) - 1 = v \\ 2 & \text{if } i - 1 = j - 1 = v \\ 0 & \text{o.w} \end{cases}$$

  and the respective b_ℓ will be 0.
- For each edge (v, w) in the graph an A_ℓ will be created according to the following:

$$A_{i,j} = \begin{cases} -1 & \text{if } min(i,j) - 1 = min(v,w) \text{ and } max(i,j) - 1 = max(v,w) \\ 1 & \text{if } min(i,j) = 1 \text{ and } max(i,j) - 1 \in \{v, w\} \\ 0 & \text{o.w} \end{cases}$$

  and the respective b_ℓ will be 2.
- An additional A_ℓ is created which is fully zero and has a one on the upper-left corner and the respective b_ℓ is 1.

These tests have $n + m + 1$ conditions and matrices of size $n + 1$.

Some of the datasets are split into two subsets according to their difficulty. The tests are split according to numerical difficulty. In numerous experiments using different implementations of the dynamics, some particular tests appeared to be harder for the Physarum solver. In specific, the vertex cover tests tended to have a large number of linear constraints which made it particularly challenging for the Vanilla solver. That being said, we have separated each set to have a more accurate comparison on the challenge tests. For validation, the answers are compared with the standard SDPASolver [2] as ground truth.

TestSet Description	TestSet Name	No. of tests
Randomly generated definite SDP samples with $n = 5, m \in [1, 3]$	testset1	20
Randomly generated definite SDP samples with $n = 10, m \in [1, 5]$	testset2-1	14
	testset2-2	16
Randomly generated definite SDP samples with $n = 25, m \in [1, 10]$	testset3-1	16
	testset3-2	4
Randomly generated vertex cover problems with $\lvert V(G)\rvert = 5, \lvert E(G)\rvert \in [1, 10]$	vertexcover1-1	13
	vertexcover1-2	7
Randomly generated vertex cover problems with $\lvert V(G)\rvert = 20, \lvert E(G)\rvert \in [1, 130]$	vertexcover2-1	4
	vertexcover2-2	16
Randomly generated definite SDP samples with $n = 50, m \in [5, 10]$	large1	10
Randomly generated definite SDP samples with $n = 100, m \in [5, 20]$	large2	20
Randomly generated vertex cover problems with $\lvert V(G)\rvert = 50, \lvert E(G)\rvert \in [10, 20]$	vertexcover3	10

Table 1: Standard SDPLib dataset that is generated to validate the Physarum dynamics.

7.2 Vanilla Algorithm

The algorithm for our Physarum solver would consist of a sequence of iterations where in each iteration we will solve the update problem and obtain Q. Afterward, we update $X(t)$ with the

[2] `http://sdpa.sourceforge.net/`

discretized dynamics and step h to obtain $X(t+1)$. This process continues until $X(t)$ and $Q(t)$ become close to each other which indicates reaching an equilibrium point. We will stop iterating whenever $||Q(t)-X(t)||$ is smaller than a small constant ϵ. Thus we obtain the following algorithm:

Algorithm 1: Vanilla Physarum SDP solver

Input : $C, A_1, \dots, A_m \in S_n$, $b \in \mathbb{R}^m, X(0)$.
Output: $(X^{eq} \succeq 0, p^{eq})$.

Calculate $U(0), \Lambda(0), k_0$ using Corollary 3.3 from $X(0), C$;
Let $C^{\dagger}_{(k_0)} = U_k(0)U_k^T(0)$;
Let $t = 0$;
repeat
  Let $Q(t), p(t) = SolveUpdateProblem(C^{\dagger}_{(k_0)}, A_1, ..., A_m, b, X(t))$ using Algorithm 2;
  Calculate small enough h;
  Update $X(t+1) = hQ(t) + (1-h)X(t)$;
  Increment t;
until $||Q(t) - X(t)||$ *is more than* ϵ;
return $(X(t), p(t))$

Each iteration needs to solve the update problem. The algorithm to solve the update problem is obtained from Section 3 as below:

Algorithm 2: Solve Update Problem

Input : $C^{\dagger}_{(k_0)}, A_1, \dots, A_m \in S_n$, $b \in \mathbb{R}^m, X$.
Output: Q, p.

Calculate the $m \times m$ matrix M and let $M_{i,j} = tr(C^{\dagger}_{(k_0)} A_i X A_j)$.;
Calculate $p = M^+ b$.;
Calculate $Q' = \sum_{\ell}^{m} p_\ell C^{\dagger}_{(k_0)} A_\ell X$.;
Let $Q = \frac{Q' + Q'^T}{2}$;
return (Q, p)

As mentioned before, h should be chosen as a small value. One obvious upper-bound for h is obtained using the fact that the update should preserve positive semi-definiteness of $X(t)$. Moreover, for $W(X(t))$ defined in (16) not to explode $\tilde{O}^T X(t) \tilde{O}$ should remain positive definite. This leads to the following upper-bound for h:

$$\begin{aligned}
&O^T\big[h(Q(X(t)) + (1-h)X(t)\big]O \succ 0 \\
\Rightarrow\ &hO^T U(t)\tilde{Q}(t)U^T(t)O + (1-h)O^T U(t)\Lambda(t)U^T(t)O \succ 0 \qquad O^T U(t) \text{ is non-singular} \\
\Rightarrow\ &\Lambda(t) \succ h(\Lambda(t) - \tilde{Q}(t)) \\
\Rightarrow\ &I \succ h(I - \Lambda^{-\frac{1}{2}}(t)\tilde{Q}\Lambda^{-\frac{1}{2}}(t)) \\
\Rightarrow\ &h < \frac{1}{1 - \lambda_{\min}(\Lambda^{-\frac{1}{2}}\tilde{Q}\Lambda^{-\frac{1}{2}})} = h_{bound}
\end{aligned}$$

In Algorithm 1 we choose h to be the minimum of a small constant such as h_c and $\frac{h_{bound}}{2}$. In some tests where we aim for a higher accuracy we set h_c to a smaller number thus jeopardizing speed for accuracy.

We have proven Algorithm 1 works good on SDP problems when (CC.1) to (CC.4) hold. We will run the vanilla algorithm on the Augmented setting in which we will augment all the matrices using a γ and run Algorithm 1 to obtain a solution and then obtain an answer for the original problem by considering the upper-left corner of X_{eq} and dividing the matrix by γ.

Table 2 contains the results of Algorithm 1 on each of the datasets. Each row contains one of the datasets in 1. As you can see, the solver provides accurate solutions on most of the datasets. In the 'vertexcover1' and 'vertexcover2' dataset however we have an increase in the number of linear conditions which makes M larger. This will both make the algorithm slower and more numerically unstable. Because one should solve the equation $Mp = b$ where M is an $m \times m$ matrix for each iteration which is a large matrix. In addition to that, the solver runs into numerical difficulties in these tests. We have figured these problems are caused due to two main reasons:

- h is not chosen sufficiently small. We have only used a naive approach to choose h and by tuning it we might get more accurate results.
- M becomes ill-conditioned due to some of $\lambda_i(t)$ values vanishing.

Of the problems above, the latter will be addressed in the next part and will help us solve the remaining tests albeit with a slower processing speed. We will also modify the whole algorithm in order to obtain a solver which solves all the created dataset.

Additionally, we have also shown the maximum $\beta(t)$ obtained in the last iteration for each dataset; these values have converged to small values supporting Conjecture 6.1. Moreover, the algorithm has converged rather fast to the optimum solutions in these cases making the vanilla algorithm a fast but inaccurate method.

TestSet	Ratio of tests with error below 10^{-2}	Average Time Spent	Maximum primal gap on tests with a lower than 10^{-2} primal gap	Maximum Infeasibility	Maximum β in last iteration for high accuracy tests
testset1	20/20	0.635 (sec)	2.7×10^{-8}	7.7×10^{-12}	7.4×10^{-54}
testset2-1	14/14	0.335 (sec)	9.5×10^{-7}	7.9×10^{-12}	9.5×10^{-7}
testset2-2	6/6	1 (sec)	1.3×10^{-6}	1.2×10^{-11}	3.8×10^{-114}
testset3-1	14/16	1.26 (sec)	7.9×10^{-7}	1.8×10^{-11}	9.3×10^{-30}
testset3-2	4/4	4.3 (sec)	4.7×10^{-7}	1.8×10^{-11}	7.3×10^{-189}
vertexcover1-1	13/13	1.4 (sec)	7×10^{-4}	0	7.4×10^{-5}
vertexcover1-2	5/7	1.8 (sec)	10^{-3}	7.2×10^{-5}	10^{-3}
large1	10/10	7.1 (sec)	1.2×10^{-8}	3.2×10^{-11}	7×10^{-54}
large2	20/20	25.8 (sec)	1.4×10^{-6}	6×10^{-11}	1.3×10^{-6}
vertexcover2-1	1/4	60 (sec)	2.6×10^{-9}	0	1.0×10^{-7}
vertexcover3	4/13	110 (sec)	4.8×10^{-5}	0	2.2×10^{-3}
vertexcover2-2	2/16	196 (sec)	3.3×10^{-9}	0	2.6×10^{-3}

Table 2: Results of the vanilla algorithm after augmenting the matrices with $\gamma = 0.01$. A test is considered accepted if the primal gap between the Physarum algorithm primal objective value and the ground truth is less than 10^{-2}.

7.3 Modified Algorithm

Here we will present a modified version of the algorithm that seems to work on all of the SDP samples. Moreover, we will not augment the problem in this experiment and set $X(0)$ to a very large matrix, i.e, $B \times C$ for a large number B with respect to the base problem.

We propose the following guess which is analogous to the findings in Physarum solvers for LP.

Conjecture 7.1. *If $X(t)$ is characterized by the Physarum dynamic in (7) aimed to solve the positive semi-definite SDP in (5), it will converge to the following optimum solution if $X(0) \succ X^{opt}$ for some optimum solution X^{opt} of the following:*

$$\underset{X}{argmin}\{tr(CX) \mid \forall \ell \in [m] : tr(A_\ell X) = b_\ell,\ X \succeq 0,\ Null(X(0)) \subseteq Null(X)\}$$

Now we would present a modified version of Algorithm 1 which is numerically more stable. In case one of the eigenvalues of $X(t)$ approaches zero, M will become ill-conditioned propagating errors. The idea is that according to Conjecture 7.1, if we start from a large enough positive semi-definite matrix $X(0)$ we will converge to the optimum primal of the SDP in (5) with an additional $Null(X(0)) \subseteq Null(X)$ condition added.

Suppose at some point of time $X(t)$ obtains a small eigenvalue which maps to a small generalized eigenvalue $\lambda_i(t)$. One may assume $\lambda_i(t)$ converges to zero and that $\tilde{u}_i(t)$ converges to a vector in the null-space of an optimum solution. So whenever we encounter a small enough generalized eigenvalue such as $\lambda_i(t)$, we may assume that $\tilde{u}_i(t)$ is in the null-space of an optimum solution. Hence we may restart from X_0' which contains that particular vector in the null-space, e.g., $X_0' = B \times (X(t) - \lambda_i(t)\tilde{u}_i(t)\tilde{u}_i^T(t))$. The new restart point X_0' contains $\tilde{u}_i(t)$ in the null-space inherently and according to Lemma 3.2 will always contain it in the null-space. If Conjecture 7.1 holds, will converge to a solution of the SDP with an additional constraint that $\tilde{u}_i$ is in the null-space; although this will not make our answer sub-optimal since $\tilde{u}_i$ is part of the null-space of some optimum solution.

The following algorithm runs in multiple epochs. In each epoch we will do a decomposition of $X = U\Lambda U^T$ and exclude the small generalized eigenvalues from Λ by getting rid of the respective rows and columns in Λ to get $\hat{\Lambda}$ and the respective columns in U to get $\hat{U}$. We will then choose $X' = \hat{U}\hat{\Lambda}\hat{U}^T$ as our new starting point in that epoch and continue iterating. This will reduce the condition number of $\tilde{G}$ and M which leads to a more numerically stable algorithm.

Algorithm 3: Modified Physarum SDP solver

Input : $C, A_1, \ldots, A_m \in S_n$, $b \in \mathbb{R}^m$, $X(0)$.
Output: $(X^{eq} \succeq 0, p^{eq})$.

repeat
  Calculate $U, \tilde{U}, \Lambda$ using Corollary 3.3 for $X(0), C$;
  For each i that $\Lambda_{i,i} \leq \epsilon$ add it into the set A;
  Obtain matrix $\hat{U}$ by considering the columns of U that are in A;
  Obtain matrix $\hat{\Lambda}$ by considering the intersection of rows and columns in A;
  Obtain $\tilde{O}$ by only considering the columns of $\tilde{U}$ that are in A;
  Set $X(0) = \hat{U}\hat{\Lambda}\hat{U}^T$ and $k_0 = |A|$;
  Set $C^{\dagger}_{(k_0)} = \hat{U}\hat{U}^T$;
  Let $t = 0$;
  repeat
    Let $Q(t), p(t) = SolveUpdateProblem(C^{\dagger}_{(k)}, A_1, ..., A_m, b, X(t))$ using Algorithm 2;
    Calculate small enough h;
    Update $X(t+1) = hQ(t) + (1-h)X(t)$;
    Increment t;
  until $||Q(t) - X(t)|| \geq \epsilon$ *and* $\lambda_{\min}(\tilde{O}^T X(t) \tilde{O}) \geq \epsilon$;
until $||Q(t) - X(t)|| \geq \epsilon$ *and* $\lambda_{\min}(\tilde{O}^T X(t) \tilde{O}) < \epsilon$;
return $(X(t), p(t))$

Table 3 shows the results for the modified dynamics on the datasets. As you can see, this solver is able to solve all of the tests in the dataset with descent accuracy. However, it takes a longer time for the dynamic to converge accurately. We believe by a more appropriate choice of h one can also obtain a fast and more accurate Physarum solver. Additionally, one can tune the value of B at the beginning of each epoch where we reset the dynamic and set $X(0) = B \times \hat{U}\hat{\Lambda}\hat{U}^T$; thus the algorithm will not need as many iterations to approach a viable solution again for each epoch.

Despite not having a formal proof, Table 3 supports Conjecture 7.1 showing that the (CC.3) and (CC.4) conditions described previously are not needed for the dynamic to converge if we choose

$X(0)$ large enough.

Testset Name	Average Time Spent	Maximum Primal Gap	Maximum Infeasibility
testset1	7.5 (sec)	3.6×10^{-5}	5.9×10^{-12}
testset2-1	2.7 (sec)	2.9×10^{-6}	4.8×10^{-6}
testset2-2	65 (sec)	2.48×10^{-5}	1.1×10^{-13}
testset3-1	52 (sec)	1.9×10^{-7}	5.1×10^{-6}
testset3-2	115 (sec)	8×10^{-4}	5×10^{-7}
vertexcover1-1	9 (sec)	1.3×10^{-6}	2.4×10^{-7}
vertexcover1-2	41 (sec)	5.9×10^{-4}	2.6×10^{-4}
vertexcover2-1	8 (min)	4×10^{-6}	3.88×10^{-6}
vertexcover2-2	22 (min)	1.8×10^{-2}	3.3×10^{-5}
large1	32 (sec)	1.02×10^{-4}	5.9×10^{-7}
large2	170 (sec)	7.2×10^{-4}	5.5×10^{-11}
vertexcover3	7 (min)	1.7×10^{-4}	1.8×10^{-4}

Table 3: Results of the modified Algorithm 3 on the dataset. Maximum Primal Gap is the difference between the Physarum SDP solver and the SDPA baseline. Infeasibility measures how much X conflicts with the constraints. It is calculated by taking the maximum magnitude of the negative eigenvalue of X^{eq} and the maximum $|b_\ell - tr(A_\ell X)|$ for $1 \leq \ell \leq m$.

References

[1] T. Nakagaki, M. Iima, T. Ueda, Y. Nishiura, T. Saigusa, A. Tero, R. Kobayashi, and K. Showalter, "Minimum-risk path finding by an adaptive amoebal network," *Physical review letters*, vol. 99, no. 6, p. 068104, 2007.

[2] A. Tero, R. Kobayashi, and T. Nakagaki, "A mathematical model for adaptive transport network in path finding by true slime mold," *Journal of theoretical biology*, vol. 244, no. 4, pp. 553–564, 2007.

[3] A. Johannson and J. Zou, "A slime mold solver for linear programming problems," in *Conference on Computability in Europe*, pp. 344–354, Springer, 2012.

[4] D. Straszak and N. K. Vishnoi, "On a natural dynamics for linear programming," *arXiv preprint arXiv:1511.07020*, 2015.

[5] R. Becker, V. Bonifaci, A. Karrenbauer, P. Kolev, and K. Mehlhorn, "Two results on slime mold computations," *Theoretical Computer Science*, vol. 773, pp. 79–106, 2019.

[6] E. Facca, A. Karrenbauer, P. Kolev, and K. Mehlhorn, "Convergence of the non-uniform directed physarum model," *Theoretical Computer Science*, vol. 816, pp. 184–194, 2020.

[7] V. Bonifaci, "On the convergence time of a natural dynamics for linear programming," *Algorithmica*, vol. 82, no. 2, pp. 300–315, 2020.

[8] T. Kato, *Perturbation theory for linear operators*, vol. 132. Springer Science & Business Media, 2013.

[9] B. Borchers, "Sdplib 1.2, a library of semidefinite programming test problems," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 683–690, 1999.