Optimal distributions for multiplex logistic networks

Luis E. Sola Conde, Javier Used, and Miguel Romance

This paper presents some mathematical models for distribution of goods in logistic networks based on spectral analysis of complex networks. Given a steady distribution of a finished product, some numerical algorithms are presented for computing the weights in a multiplex logistic network that reach the equilibrium dynamics with high convergence rate. As an application, the logistic networks of Germany and Spain are analyzed in terms of their convergence rates. *Published by AIP Publishing.*

We are living in a global-scale world. Globalization is the process of international integration of products, ideas, culture, and science that has changed our way of understanding the world. Advances in communication and transportation that took place along the last Century allowed shortening the distances and making that our society moved from the local-scale to a global scale, generating new interdependences of economic, political, social, and cultural activities along the world. This process also has affected the business and commerce processes and the way that traditional trade is performed, changing from buying and selling to our closest neighbors to exchanging products with costumers in any country in the world. This change in the commerce's scales forces a new way of distributing the goods that has shifted from the direct supplier to costumer perspective to a new distributed point of view. Supply chain management (SCM),¹¹ which is also known as Logistics Network Analysis, is a part of the Operational Research that has been extensively studied in recent years to analyze the newly distributed perspective of commerce arises from the Globalization. In this new approach, the classic point-to-point distribution that directly connects the supplier (the factory) with the final costumer is replaced by the use of logistic networks and distributed global-scale transportation. This paper presents some mathematical models for distribution of several products simultaneously by using multiplex networks analysis, and some effective algorithms are included in order to spot the optimal logistic network, in terms of distribution rate.

I. INTRODUCTION

Supply chain management (SCM) is the branch of the Operational Research that studies optimal methods for providing materials to manufactures, transforming these materials into intermediate and finished products, and the distribution of these finished products to customers.^{1,7,11} It is an interdisciplinary field that combines techniques coming

from operational research with tools of many other fields such as optimization, risk management, or game theory among many others.¹⁰

The main goal of this paper is presenting some mathematical models for optimizing the distribution of a finished product from the manufacturers to the final costumers based on the spectral theory of complex networks⁴ and how to compute the optimal logistic network numerically. Roughly speaking, the problem we are dealing with is the following: We consider a factory that produces a finished product and we want to distribute this product to several cities along a country by using a transportation network (such as a road, railway, or airline network) optimizing the supply time. Therefore, we take a transportation network between n cities $e_1, \dots, e_n$, we fix the location of the manufacturer in one of these cities and we consider the amount of the finished product that must be supplied to each city. By using these ingredients, the solution of the problem will be the self-organized weight distributions on the connections of the transport network that guaranties that the right amount of product arrives every city as soon as possible. In fact, the problem analyzed is the distribution of $m > 1$ different products manufactured in different places but distributed along a fixed transport network, and therefore, the model used deals with the multiplex networks analysis³ with some interplays between layers.

Section II is devoted to state the precise mathematical models and to prove the theoretical results that guarantee the existence of an optimal solution to the problem. The theoretical results presented about the existence of optimal solutions are sharp but non-constructive in general and therefore not much useful in real applications. In order to solve this inconvenience, Section III presents some effective numerical algorithms that compute almost optimal solutions to the distribution problems modeled in Section II, and therefore, they give a tool for solving the distribution problem in real applications. Section IV includes the use of the algorithms included in Section III, by showing a comparative analysis of the distribution of finished products through the transport networks of Germany and Spain. Finally, Section V summarizes

the conclusions derived from the results presented and lists some future works that could be started from this paper.

II. MATHEMATICAL MODELS AND RESULTS

As we have stated in Section I, the problem we are dealing with is, roughly speaking, the following: We take a fixed transportation network between cities $c_1, \dots, c_n$, and we want to deliver several products that have been manufactured in some of the cities (i.e., the factory of each product is one of the cities considered) by using a distributed and collaborative model. The delivery method will be modeled as a dynamical system in discrete time such that:

- In the beginning, all the amount of each product is in its corresponding factory (i.e., in the city that manufactures the product).
- At each time step, a fraction of each product is delivered from each city to its neighbors by using the transportation network. The amount of product delivered from a city is a fraction of the product that it was in the city at the end of the previous time step. These fractions delivered are modeled as weights in each link between two cities. The proportion of product delivered from one city to another depends on the product considered and the two cities involved.
- The delivery process repeats until some equilibrium is reached (a steady state).

In this section, we will translate this idea into a solvable mathematical model, analyzing the existence of a steady state, the convergence speed to the steady state, and in Section III, we will present some numerical methods for locating a delivery method with optimal (maximal) convergence speed.

We start with a network consisting of n nodes $\{e_1, \dots, e_n\}$ and links between them that we represent with an $n \times n$ matrix

$$T = (T_{ij})_{i,j}, \quad T_{ij} = \begin{cases} 1 & \text{if } e_j \text{ is linked to } e_i, \\ 0 & \text{otherwise.} \end{cases}$$

Here, the condition that “ e_j is linked to e_i ” means that the node e_j is capable of distributing products to the node e_i . In principle, we do not assume A to be symmetric, but we set T_{ii} to be equal to 1 for all i (we allow each node to distribute products to itself).

Now we consider a set of $m \in \mathbb{N}$ non-negative $n \times n$ matrices

$$D_r = (D_{ijr})_{i,j}, \quad D_{ijr} \geq 0, \quad \text{for all } 1 \leq i, j \leq n, 1 \leq r \leq m.$$

Satisfying the following conditions:

- (N1) (Prescribed topology) $D_{ijr} = 0$ if $T_{ij} = 0$, for all i, j, r . This condition means that the distribution is realized upon the network modelled by T .
- (N2) (Stochasticity) $\sum_i D_{ijr} = 1$, for all r, j . Every node j distributes the total amount of their products r between their neighbours (including itself).

- (N3) (Homeostasis) $\sum_r v_r D_{ijr} \leq C_i$, for all i . In this formula, we have constants v_r, C_i that shall be determined for each particular problem: v_r is a measure of the volume of the product r , and C_i measures the total capacity of the node i . Obviously, we need to impose that $v_r \leq C_r$, for all r (since one should be able to assume that in the initial state the whole production of the node r is stored in r).

The $n \times n \times m$ tensor D is then used to distribute the products of the different nodes by the following recursive rule:

$$\begin{cases} X_D(t)_{ir} := \text{amount of prod. } r \text{ at node } i \text{ (instant } t), \\ X_D(0) := (n \times m) \text{ identity matrix,} \\ X_D(t) = D \cdot X(t-1), \text{ for all } t, \text{ that is,} \\ X_D(t)_{ir} = \sum_k D_{ikr} X_D(t-1)_{kr}, \text{ for all } m, i, r. \end{cases} \quad (1)$$

The tensor D is then used as a linear operator that may be represented as the $nm \times nm$ block matrix

$$D = \begin{pmatrix} D_1 & 0 & \cdots & 0 \\ 0 & D_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D_m \end{pmatrix}.$$

A. Convergence of a distribution model

It is well known (see, for instance, Ref. 9, chapter 8) that the process described in (1) converges to a steady state under the following topological conditions:

- (C1) *Strong connectivity* of the network: given any two indices i and j , we may go from the node e_i to the node e_j following the links of the network.
- (C2) *Primitivity* of the matrices D_r : the only eigenvalue of D_r with norm equal to the spectral radius λ of D_r is λ . This may be expressed by saying that the k -th power of the matrix D_r is positive, for some $k > 0$, by Frobenius criterion, or, equivalently, if there exist no permutation of indices such that the matrix D_r is a block permutation matrix (see Ref. 9, page 680).

The first condition implies that a primitive network is, in particular, strongly connected, while the second shows that primitivity is indeed a topological condition, depending only on the set of entries of a matrix that are not zero and not on their values.

In our case, since we are assuming $T_{ii} = 1$ for all i , we get the following:

Proposition 2. *With the same notation as above, the network determined by the matrix T is strongly connected if and only if T is primitive. Moreover, if T is symmetric, both conditions are equivalent to the connectivity of the network.*

Remark 3. Note that, in the way we have defined them, even if T is primitive, the matrices D_r can be non-primitive.

The following is a particular case of the classical Perron–Frobenius theorem (see, for example, Ref. 9).

Theorem 4. Assume that the matrix D_r is primitive, for every r . Then, the following limit exists:

$$X_D(\infty) := \lim_{t \rightarrow \infty} X_D(t),$$

and $X_D(\infty)$ is an eigenvector associated to the spectral radius 1 of the linear operator D , which is the only eigenvalue with norm equal to one.

Definition 5. With the same notation as above, we will call $X_D(\infty)$ the Perron matrix of D .

B. Measuring the convergence speed

It is also known that the convergence speed of the power method depends on the norms of the eigenvalues of D different from the dominant one.⁹ In our case, the spectrum of D is the union of the spectra of the matrices D_r and, in particular, its spectral radius is equal to one. The value 1 is indeed an eigenvalue of the operator D of multiplicity n ; however, since the process defined by D behaves like n independent linear dynamical systems, given by the matrices D_i , it is obvious that the convergence speed depends on the eigenvalues of norm smaller than one.

In order to define an appropriate concept of convergence speed, we start by recalling the following well known statement (cf. Ref. 6, Section 7.3.1):

Theorem 6. Denoting by λ_2 an eigenvalue of D whose norm is maximum among the eigenvalues of D different from one, we have

$$\|X_D(t) - X_D(\infty)\| = O(|\lambda_2(D)|^t), \text{ for } t \text{ large.}$$

Later on, we will also study how variations on the coefficients of D may help to improve the convergence speed. Hence, we may want to describe the infinitesimal dependence of λ_2 on the coefficients of the matrix D , which can be done as follows.

For any two square matrices A and B , the characteristic polynomial of $A(t) = At + B$ can be written as $P(t, \lambda) = a(t) \prod (\lambda - \lambda_i(t))$ where the functions $\lambda_i(t)$ are differentiable at 0, by the inverse function theorem, if A has only simple eigenvalues. Then we may choose left and right eigenvalues associated to each of these eigenvalues $\lambda_i(t)$, denoting them by $w_i(t)$ and $v_i(t)$, respectively, and noting that they depend differentiably on t around $t=0$, as well. We denote $w_i = w_i(0)$, $v_i = v_i(0)$. We may then write, whenever this quotient is defined, the following formula:

$$\lambda_i'(t) = \frac{w_i(t)A(t)v_i(t)}{w_i(t)v_i(t)}.$$

By deriving at $t=0$, and taking into account that $w_i(t)A(t) = \lambda_i(t)w_i(t)$, $A(t)v_i(t) = \lambda_i(t)v_i(t)$, we may then write

$$\lambda_i'(0) = \frac{w_i(0)Bv_i(0)}{w_i(0)v_i(0)}.$$

It then follows that the gradient of λ_i at A can be written as the matrix

$$\nabla_A \lambda_i = \frac{w_i^T v_i^T}{w_i v_i},$$

where the superscript T denotes transposition here. Note that this formula applies also in the case in which λ_i is not real.

Unfortunately, choosing to define the convergence speed of D as $-\log(|\lambda_2(D)|)$, as one may be tempted to in view of Theorem 6, has certain inconveniences. First of all, this theorem only tells us that the second largest eigenvalue models the convergence for m large. In other words, it tells us how the system behaves on a small neighborhood of its equilibrium state: it is a measure of the stability of the equilibrium state. If we want to understand how the dynamical system converges, starting from any state, we need to keep track of the rest of eigenvalues of the matrix and, in fact, simulations show that convergence is usually faster than the exponential of base $|\lambda_2|$. On the other hand, λ_2 may be complex, hence $\nabla_D \lambda_2$ may have not real components, which is a problem if we want to identify this gradient as the direction of maximal growth of λ_2 . Finally, even if this gradient is real, a modification of the coefficients of D in the opposite direction to this gradient may change the order of the eigenvalues, so that the new λ_2 is not the second largest anymore.

All those problems suggest choosing a different way of measuring the convergence speed of D . First of all, we consider every D_r separately and define

$$\Psi(D_r) := \sum_{i=2}^n |\lambda_i|^2 = \sum_{i=2}^n (\lambda_i \bar{\lambda}_i), \quad (2)$$

where $\lambda_2, \dots, \lambda_n$ are all the eigenvalues of D_r different from $\lambda = 1$. Then we define

$$\Psi(D) := \max_r \Psi(D_r). \quad (3)$$

Definition 9. With the same notation as above, the convergence measure of D is defined as

$$v(D) := -\frac{1}{2} \log(\Psi(D)),$$

or, equivalently,

$$v(D) = \min_r v(D_r), \quad \text{where} \quad v(D_r) := -\frac{1}{2} \log(\Psi(D_r)).$$

Now, Ψ is a real nonnegative differentiable function, smaller than or equal to $|\lambda_2|$, whose gradient may be computed as follows:

$$\nabla_D \Psi = 2 \sum_{i=1}^n \bar{\lambda}_i \frac{w_i^T v_i^T}{w_i v_i}. \quad (4)$$

Note that, for each r , the Schur decomposition of D_r (see, for example, Ref. 9) provides three matrices Q , E , N satisfying that $Q^{-1} = \bar{Q}^T$ (here, the overlining indicates complex conjugation, and T the standard transposition), E is diagonal (with the same eigenvalues as D_r), N is strictly upper triangular, and $D_r = Q(E + N)\bar{Q}^T$. Whence

$$\begin{aligned}
\Psi(D_r) &= \text{tr}(E\bar{E}^T) \\
&= \text{tr}(D_r D_r^T) - \text{tr}(N\bar{N}^T) = \|D_r\|_2 - \|N\|_2 \\
&\leq \|D_r\|_2,
\end{aligned}$$

where $\|\cdot\|$ denotes the Frobenius norm of a matrix. This is telling us that, in order to find matrices D_r with high convergence measure—this will be one of our goals, see Problem 14 below—one should look for matrices D_r with small Frobenius norm. This approach has the advantage of being computationally more economic than considering the convergence measure.

C. The polytope of distributions of a given network

So far, we have studied in this section the existence of a steady state of a weight distribution supported on a fixed transportation network and how to measure the convergence speed to this asymptotic state. Therefore, the working sequence was the following: We take a weight distribution supported on a fixed transportation network that fulfills the conditions proposed by the model (prescribed topology, stochasticity, and Homeostasis), and we look for the steady state that this distribution gives and how fast the dynamical system converges to this steady state. In Section IID, we are going to change the perspective and we will try to find weight distribution that produces a fixed steady state (if it exists) with the highest convergence speed as possible (this will be analyzed in Section III). Roughly speaking, the problem we are dealing with in the rest of this section is the following:

Problem 11 (Inverse Distribution Problem). *Given a fixed transportation network T between cities $e_1, \dots, e_n$, values $v_r, C_i > 0$, and an stochastic $n \times n$ matrix u (that corresponds to a prescribed final delivery distribution of the products in each city), can we find weight distribution D supported on T and whose steady state is $X_D(\infty) = u$?*

Note that inverse distribution problem is a natural problem in logistics, since it is usual to have a fixed transportation network and consider a prescribed final delivery distribution of the products according to the customers' choices and we want to plan the weight distribution in order to fulfill this delivery balance.

In order to analyze the existence of weight distribution D supported on a transportation network T with a prescribed steady state, we have to study the structure of the submanifold of weighted matrices supported on T and if there are some of them with the prescribed steady state. The techniques used for analyzing such sub-varieties of the matrix space include methods coming from convex geometry and polytope analysis. In Section IID, we are going to construct such sub-varieties from this convex geometry's perspective in order to solve properly the inverse distribution problem.

For every index j , let us denote

$$S_j := \left\{ x \in \mathbb{R}^n \mid \sum_i x_i = 1, \text{ and } 0 \leq x_i \leq T_{ij}, \text{ for all } i \right\},$$

where every vector $x \in \mathbb{R}^n$ is written as $x = (x_1, \dots, x_n)$. Then the set

$$S := \prod_j S_j,$$

considered as a subset of the space of $n \times n$ matrices with real coefficients, is the set of all the possible right-stochastic matrices for which the topology of the associated network is given by T . Geometrically, it may be described as a product of n simplices S_j of dimensions smaller than or equal to $n - 1$.

All the possible tensors D defined as above, for our given network T , belong to the Cartesian product S^n , and we may consider

$$\mathcal{O} := \{D \in S^n \mid D \text{ satisfies (N1), (N2), (N3)}\},$$

which is a linear section $\mathcal{O}$ of S^n . Inside of $\mathcal{O}$ we have an open subset (possibly empty)

$$\mathcal{O}_1 := \{D \in \mathcal{O} \mid D_r \text{ is primitive, for all } r\} \subset \mathcal{O}.$$

Alternatively, we may consider the open subset $S_1^n \subset S^n$ of primitive tensors, and then $\mathcal{O}_1 = \mathcal{O} \cap S_1^n$ is a linear section of S_1^n . Note that S^n and $\mathcal{O}$ are convex polytopes, and $S_1^n \subset S^n$ and $\mathcal{O}_1 \subset \mathcal{O}$ are open subsets. It is obvious that these open sets are defined as the complementary subsets of a certain union of faces of the polytopes, corresponding to the elements D for which at least one of the D_r 's is non-primitive.

Definition 12. *The polytope $\mathcal{O}$ is called the polytope of distributions of the network T , and $\mathcal{O}_1$ is called the set of primitive distributions of T .*

We introduce now the equilibrium state $X_D(\infty)$ into the above picture. Note first that we have a differentiable map, associating to every element D of $\mathcal{O}_1$ its corresponding eigenvector $X_D(\infty)$

$$\psi : D \in \mathcal{O}_1 \mapsto X_D(\infty) \in M_n(\mathbb{R}).$$

Moreover, for every D , the vector $X_D(\infty)$ lies in the simplex of stochastic matrices

$$\Delta := \left\{ u \in M_n(\mathbb{R}) \in \mathbb{R}^n \mid \sum_i u_{ir} = 1, u_{ir} \geq 0, \forall i, r \right\}.$$

The set $\mathcal{O}_1(u) := \psi^{-1}(u)$ may be described as the solutions $D \in \mathcal{O}_1$ of the linear system $D \cdot u = u$. Denoting by H_u the affine subvariety of the space of $n \times n \times n$ tensors D satisfying $D \cdot u = u$, we may write

$$\mathcal{O}_1(u) = H_u \cap \mathcal{O}_1.$$

This is an open set (possibly empty) of the polytope $\mathcal{O}(u) := H_u \cap \mathcal{O}$.

Definition 13. *The polytope $\mathcal{O}(u) := H_u \cap \mathcal{O}$, in which $\mathcal{O}_1(u)$ is an open set, is called the polytope of distributions of the network T with equilibrium state u .*

According to the view presented in the Introduction, given a certain prescribed equilibrium state u , within

$\mathcal{O}_1(u) \subset \mathcal{O}(u)$ we will be interested in finding elements D having high convergence. In other words, we are interested in the following mathematical problem, which is the precise transcription of the inverse distribution problem to the convex geometry's language.

Problem 14. *Given a network T , real values $v_r, C_i > 0$, and an stochastic $n \times n$ matrix u , find an element $D \in \mathcal{O}_1$ satisfying that $X_D(\infty) = u$ (that is, $D \in \mathcal{O}_1(u)$), and that the convergence measure $v(D)$ is maximal for the matrices satisfying that property.*

D. The polytopes S^n , $\mathcal{O}$ and $\mathcal{O}(u)$

In this subsection, we will discuss the equations and vertices of the polytopes appearing in our problem. We start by defining the following vector subspace of the space of $n \times n$ matrices

$$V := \text{v.s. gen. by } \{E_{ij} \mid T_{ij} = 1\},$$

where E_{ij} is the matrix whose coordinates are all zero, except $(E_{ij})_{ij} = 1$, for every i, j . We denote by (z_{ij}) the coordinates in V with respect to the basis $\{E_{ij}\}$ (ordered in lexicographic order $j\bar{i}$). Inside V , S may be described, either:

- as the convex hull of the elements of the form $\sum_j E_{f(j)j}$, where f is any map assigning to every index j an index $f(j)$ satisfying that $T_{f(j)j} = 1$, or
- as the polytope defined by the intersection of the following half-planes and hyperplanes: $z_{ij} \geq 0$, for all i, j , $\sum_i z_{ij} = 1$, for all j .

Now the linear span of our polytope S^n is the n -th power V^n , and considering coordinates (z_{ij}) for each factor V provides a system of coordinates (z_{ijr}) on V^n . Then we may use the above description of S , to describe S^n . For instance, the facet representation of S^n is

$$S^n = \left\{ (z_{ijr}) \in V^n \mid z_{ijr} \geq 0, \sum_i z_{ijr} = 1, \forall i, j, r \right\}. \quad (5)$$

Then the polytope $\mathcal{O}$ is defined by

$$\begin{cases} z_{ijr} \geq 0, & \text{for all } i, j, r, \\ \sum_i z_{ijr} = 1, & \text{for all } j, r, \\ \sum_{r,j} v_r z_{ijr} \leq C_i, & \text{for all } i, \end{cases}$$

and the polytope $\mathcal{O}(u)$ by

$$\begin{cases} z_{ijr} \geq 0, & \text{for all } i, j, r, \\ \sum_i z_{ijr} = 1, & \text{for all } j, r, \\ \sum_{r,j} v_r z_{ijr} \leq C_i, & \text{for all } i, \\ \sum_k z_{ikr} u_{kr} = u_{ir}, & \text{for all } r, i. \end{cases}$$

Proposition 16. *With the same notation as above, assume that $nv_r < C_r$ for all r . Then the polytope $\mathcal{O}$ has the same dimension as S^n .*

Proof: It is enough to note that under this assumption $\mathcal{O}$ contains an open neighbourhood of the identity tensor I

(given by $z_{ijr} = 1$ if and only if $i = j$, for all r) within the polytope S^n . $\square$

Moreover, if we denote

$$m(i) = \{j \mid T_{ij} = 1\}, \quad \text{for all } i = 1, \dots, n,$$

we can pose the following straightforward statement:

Proposition 17. *Assume that the values v_j, C_i satisfy the following condition:*

$$\sum_r \sum_{j \in m(i)} v_j \leq C_i, \quad \text{for all } i. \quad (6)$$

Then the third set of inequalities in the definition of $\mathcal{O}$ or $\mathcal{O}(u)$ is superfluous.

In other words, under these conditions, our system may be thought of as n independent networks (one for each product that we are distributing), and the polytope $\mathcal{O}(u)$ may be simply described as the product of the polytopes $\mathcal{O}(u^r)$, $r = 1, \dots, n$, where u^r denotes the r -th column of u , and each $\mathcal{O}(u^r)$ is defined by the equations

$$\begin{cases} z_{ijr} \geq 0, & \text{for all } i, j, \\ \sum_i z_{ijr} = 1, & \text{for all } j, \\ \sum_k z_{ikr} u_{kr} = u_{ir}, & \text{for all } i. \end{cases}$$

Definition 19. *We say that Problem 14 is unsolvable if $\mathcal{O}(u)$ does not contain any primitive element.*

Even if the conditions in Proposition 16 hold, it is not true in general that Problem 14 is solvable for every possible u . However, since the determination of vertices of a polytope is a computationally complicated problem, determining the solvability of our problems may be involved. In order to determine, in a practical way, if our problem is solvable, we make the following observation.

Remark 20. Since $\mathcal{O}$ and $\mathcal{O}(u)$ are polytopes containing the identity tensor I , for all u , we may say that Problem 14 is solvable if the linear sub-variety H_u contains a primitive element in an open neighborhood of I .

In other words, we may first look for solutions close enough to I . Close to I , the polytope $\mathcal{O}$ looks like the polyhedral cone $\mathcal{C}$ of vertex I generated by the edges of $\mathcal{O}$ passing by I . Note that we have one of these edges for every link of the network defined by T between two different nodes. If we denote by α_j the number of nodes $i \neq j$ that can be reached from j , we may say that, under the assumptions of Proposition 17 (i.e., assuming no homeostasis), $\mathcal{O}$ has precisely $n \prod_i (\alpha_i + 1)$ vertices, whilst $\mathcal{C}$ has one vertex, I , and $n \sum_i \alpha_i$ extremal rays. This makes computationally much easier to compute the intersection $H_u \cap \mathcal{C}$ than $H_u \cap \mathcal{O}$.

III. NUMERICAL ESTIMATES FOR THE OPTIMAL DISTRIBUTIONS

Once we have analyzed the existence of weight distribution supported on a fixed transportation network and with a prescribed steady delivery state (Inverse Distribution Problem),

we are now focusing on optimizing the weights in order to get the fastest convergence rate to the steady state. This is also a natural problem in logistics since a weight distribution with the highest convergence speed means a more efficient distribution of the products on the cities, less costs and a more clever distribution network that could adapt faster to the costumers' needs.

For simplicity, along this section, we are going to assume the hypotheses of Proposition 17, so that in particular, our system behaves like n independent distribution networks that may be then treated separately. This in particular, tells us that Problem 14 reduces to calculate optimal elements in the polytopes $\mathcal{O}(u^r)$, for every $r = 1, \dots, n$. The methods that we are going to describe here may be applied also in the general case, but we have then to consider the whole set polytope $\mathcal{O}(u)$.

Let us then describe the strategy that we are going to follow in searching for an optimal element in $\mathcal{O}(u^r)$, for fixed r . It will consist of two steps:

- (1) Computing a set of elements in $\mathcal{O}(u^r)$, close to the identity $n \times n$ matrix I .
- (2) Optimization of the elements obtained in the previous step. We propose here two different ways to obtain an approximation D'_r of the optimal D_r ; one based on the classic *Gradient Descent Method* and a second one, coarser but computationally more economic, that we call the *Shrapneling algorithm*.

A. Determination of a set of elements of $\mathcal{O}(u^r)$

As in Section II D, the polytope $\mathcal{O}(u^r)$ may be described as the intersection of the polytope S , described in Section II D, with a linear space H_{u^r} . In order to compute points in $S \cap H_{u^r}$, one may, of course, consider the inequalities defining S and the equations defining H_{u^r} , use any vertex determination algorithm (see Refs. 2 and 8 and the references therein) to determine *all* the vertices of $\mathcal{O}(u)$, and then take convex combinations of these vertices. Unfortunately, this approach has a very high computational cost. On the other hand, given any point in S , checking if it is contained in H_{u^r} is a procedure of linear order. We will use this simple idea to produce the following alternative geometric method to compute points in $\mathcal{O}(u^r)$. As in Remark 20, we will find such points in the vicinity of the identity I . Hence, we consider coordinates with origin at I , with respect to the basis generated by the vertices of S that share an edge of S with I , and consider the metric that makes this basis orthonormal. We may write the equations of H_{u^r} with respect to these coordinates and denote by π the orthogonal projection of S into H_{u^r} (that may be encoded into an $(\dim(H_{u^r}) \times \dim(S))$ -matrix).

We may now choose random points $A_1, A_2, \dots$, in the convex hull of I and the edges passing by it, and project them onto H_{u^r} , checking, for every point, if the coordinates of the projection with respect to the basis are all positive. If they are, then the corresponding projected points provide primitive elements $D_r(1), D_r(2), \dots \in S \cap H_{u^r}$ that we may use as initial matrices for the optimization process. Figure 1 sketches the polytopes, some points and their projections.

Note that this method produces not only a set of non-optimal (in the sense that the convergence speed is not maximal for them) solutions of Problem 14, but it may also be used to estimate the relative size of $S \cap H_{u^r}$ with respect to S , if we compute the percentage of random points whose projection lies into $S \cap H_{u^r}$. In particular, it may help us to estimate if a particular problem of the type of Problem 14 is unsolvable.

B. Shrapneling algorithm

We describe here the first optimization process. We start by noting that the method for obtaining points of $\mathcal{O}(u^r)$ described above is simple enough to be carried on a sufficient number of random points, so that, in case our problem is solvable, we should be able to obtain as many points of $\mathcal{O}(u^r)$ as we want (or, as our computing capabilities allow us to). On the other hand, it will provide only elements on a neighborhood of the identity, therefore, we may not expect, in general, one of these points to be a good approximation of an optimal element. But, since the set $\mathcal{O}(u^r)$ is convex, any element in it (in particular, an optimal element) may be joined to the identity I by a segment completely contained in $\mathcal{O}(u^r)$. Then we may expect that this optimal element admits a good approximation by a point lying in one of the half-lines passing by I and by one of the points $D_r(k)$ obtained by the projection method, if we produce a sufficient number of points in this way.

Then the Shrapneling algorithm may be described as the process of finding an optimal element in segments passing by I :

- (1) Compute a (sufficiently large) set of random solutions in the vicinity of I . Denote them by $D_r(k)$, for $k = 1, \dots, t$.
- (2) For every $t \in \mathbb{R}_{\geq 0}$ consider the elements of the form $D_r(k, t) = I + t(D_r(k) - I)$, contained in the polytope of solutions $S \cap H_{u^r}$. Note that, if the set of eigenvalues of $D_r(k)$ is $\{\lambda_j(k)\}_j$, then the set of eigenvalues of $D_r(k, t)$ is $\{\lambda_j(k) - 1)t + 1\}_j$, and $\Psi(D_r(k, t)) = \sum_j \|(\lambda_j(k) - 1)t_{\min} + 1\|^2$ is a quadratic equation in t . We need to impose also that our matrices $D_r(k, t)$ belong to $S \cap H_{u^r}$, a condition that can be easily rewritten as follows:

$$0 \leq t \leq T(D_r(k)) := \min_{D_r(k)_j \neq 1} \left\{ \frac{1}{1 - D_r(k)_j} \right\}.$$

Now a straightforward computation tells us that the minimum value of $\Psi(D_r(k, t))$ is obtained for the value

$$t_{\min} := \min \left\{ T(D_r(k)), \frac{n - \sum_i \lambda_i}{\sum_i \|\lambda_i - 1\|^2} \right\}.$$

- (3) Denote $D'_r(k) := D_r(k, t_{\min})$, for every k . Find the value $k = k_{\min}$ such that $\Psi(D'_r(k))$ is minimal, and choose $D'_r = D'_r(k_{\min})$ to be the approximation of the optimal D_r .

Remark 21. Clearly, the precision of the shrapneling algorithm may be improved by increasing the number of initial solutions $D_r(k)$. In order to measure this precision, we may use the function Ψ defined in Equation (2). More

concretely, except in the cases in which the optimal solution lies in the boundary of $S \cap H_{u^r}$, an optimal solution satisfies that $\nabla_A \psi = 0$. In other words, the smaller this gradient is, the better our approximation. Hence, it makes sense to define the precision of an approximation $D'_r \in \mathcal{O}(u^r)$ as

$$\rho(D'_r) = \|\nabla_{D'_r} \Psi\|_2.$$

C. Gradient descent algorithm

In the cases in which we consider that the number of points obtained by orthogonal projection is not large enough, we may optimize the solutions obtained by using a gradient descent algorithm (see, for example, Ref. 5 and the references therein), with respect to the differentiable function Ψ .

This means that we start with one of the matrices $D'_r(0) := D_r(k)$ obtained by projection, belonging to the relative interior of $\mathcal{O}(u^r)$, and we want to substitute it with a new element $D'_r(1)$ having larger convergence measure or, equivalently, smaller value of Ψ . The standard gradient descent algorithm tells us that we may do this by setting

$$D'_r(1) = D'_r(0) + \gamma(0) \nabla_{D'_r(0)} \Psi,$$

where $\gamma(0)$ has to be chosen conveniently so that the process converges. We refer the reader to Ref. 5 for more details about this topic. Note that, in our case $0 < \gamma(0) \in \mathbb{R}$ has to be chosen so that, moreover, $D'_r(1)$ belongs to the relative interior of $\mathcal{O}(u^r)$. Note also that in order to compute the matrix $\nabla_{D'_r(0)} \Psi$ we need to calculate all the eigenvectors of $D'_r(0)$, as formulated in (4).

We then proceed recursively by setting

$$D'_r(t+1) = D'_r(t) + \gamma(t) \nabla_{D'_r(t)} \Psi, \quad t > 0,$$

for some conveniently chosen $0 < \gamma(t) \in \mathbb{R}$. The process stops after N iterations, if the distance between $D'_r(N)$ and $D'_r(N-1)$ is smaller than a (prescribed) tolerance value.

Remark 22. Note that our gradient descent process may converge to a certain point in the relative boundary of $\mathcal{O}(u^r)$, and that this point may depend on the initial element $D'_r(0)$. This is caused by the fact that if we remove the boundary condition, the algorithm converges to a relative maximum in the linear space containing $\mathcal{O}(u^r)$. This problem may be solved by considering different initial elements $D'_r(0)$ and choosing the limit element that has the maximal value of Ψ .

IV. NUMERICAL COMPARISONS OF TWO LOGISTIC NETWORKS: GERMANY VS. SPAIN

In this section, we illustrate the use of the theoretical results and numerical algorithms presented in Sections II and III by comparing two logistic networks in order to see how the structure of the network affects the convergence speed, and how this fact can be applied to obtain an optimal distribution of a product.

Before starting with the numerical simulations, the *proximity networks* must be constructed, that is, suppose a

geometric network connecting n points in the plane $e_1, \dots, e_n$, that are the nodes of the network, is given together with n positive values $r_1, \dots, r_n > 0$, such that a node e_i is linked to e_j if the distance from e_i to e_j is smaller than or equal to r_i . If all the r_i 's are equal, the proximity network is called *uniform*, and in this case, the graph of the network is undirected.

As a working example, we may consider the capitals of the provinces of Spain. Restricting ourselves to the Spanish provinces lying in the Iberian Peninsula, we have a set of 47 cities that will be the nodes over which we will construct some proximity networks.

In order to construct the network, we will consider a threshold distance r , and say that two cities on the list are linked if and only if the distance between them is smaller than or equal to r . A typical feature of proximity networks is that the shortest path between points A and B may pass by a third point C . The process that consists of eliminating all the links AB for which there exist a point C for which the sum of the distances AC and CB equals the distance AB is called *threshing*. Figure 2 shows the difference between a proximity network and the result of the threshing process in the Spanish settings.

By using the Tarjan's algorithm to detect the (strongly) connected components of the subjacent graph,¹² we obtain that the minimum value of r so that the network is strongly connected is $r = 236$.

Once we have the threshed networks from the Spanish and German transport networks (as it can be seen in Top panels of Figure 4), we can start comparing both logistic networks. Consider the next problem: we have a product that we want to distribute to all the nodes of a network, what is the faster way of doing it? For solving it, we consider a certain steady state, u_f , where every node (city) of the network has a given amount of the product we want to distribute, and starting from an initial state, u_0 , what is the matrix D'_r for which the convergence speed, defined as $v(D'_r) := -\frac{1}{2} \log(\Psi(D'_r))$ (see Definition 9) is maximal?. In our case, we take as initial state a state where all the product to be distributed is in only one node, e_i , and as steady state u_f the state where every node of the network receives the same quantity of product. This problem is solved using the algorithms shown in Section III: gradient descent and Shrapneling algorithms.

It is important to point out the fact that the proportion of the elements obtained randomly from $\mathcal{O}(u^r)$ in the numerical algorithms strongly depends on the steady distribution u^r

FIG. 1. Using projections to produce points in $\mathcal{O}(u^r)$.

FIG. 2. Strongly connected network from Spain before the thresholding process (left panel) and after (right panel).

considered. In order to show these phenomena, we can consider an uniparametric family of steady distributions u^r given by

$$u^r = \lambda u_f + (1 - \lambda) u_p,$$

where $\lambda \in [0, 1]$, u_f is the uniform distribution on the cities (i.e., the steady state consists on giving the same fraction of product to each city) and u_p is the distribution proportional to the population of each city (i.e., u^r is a convex combination of u_f and u_p). Figure 3 shows the variation of the proportion of the elements obtained randomly from $\mathcal{O}(u^r)$ for the Spanish logistic network in terms of the value λ and it is clear that if $u^r \approx u_f$ the proportion of elements in $\mathcal{O}(u^r)$ is close to 50%, while if u^r is similar to u_p , then the proportion of elements in $\mathcal{O}(u^r)$ is almost null. In the rest of this section, we will fix $u^r = u_f$, i.e., the steady distribution will be uniform on the cities of the logistic network, but essentially the same phenomena happen if we consider other steady distribution (as $u^r = u_p$ proportional to each city population).

When we compute the problem and obtain the matrices D'_r for every network with both methods, the evolution of the

FIG. 3. Fraction of the elements obtained randomly from $\mathcal{O}(u^r)$ (on the Y-axis) as a function of λ (on the X-axis) for the Spanish, where u^r is the steady state given by $u^r = \lambda u_f + (1 - \lambda) u_p$, where u_f is the uniform distribution on the 47 cities and u_p is the distribution on the 47 cities proportionally to their population.

system can be analyzed studying the successive iterations $u_{n+1} = D'_r \cdot u_n$. Point out that what we have defined as convergence speed only depends on the structure of the network and on the algorithm used to obtain it, but the number of iterations that is necessary to apply to reach the steady state has a great dependence on the initial state. It is to say, as closer from the steady state the initial state with less iterations will be needed to reach the steady state (we consider that the steady state is reached when the norm of the difference vector between the actual and the steady state is less than 2% of the norm of the initial state). Tables I and II present the five cities which arrive in less iterations to the steady state for the Spanish and German logistic networks, respectively. This list also depends on the algorithm that is used to obtain the matrix D'_r .

As the number of iterations needed to reach the steady state strongly depends on the initial state, we cannot obtain too much information by comparing the evolution of the system from two particular cities. An idea about the convergence of the network can be deduced with the mean value of all the evolutions of the difference vector between the actual and the steady state (see bottom panels of Figure 4).

Finally, we can analyze the dynamics on both networks by comparing the trajectories from the initial state until the steady state in the case of the simplified Spanish and German networks. Since these two networks have different nodes, it is not straightforward to choose similar initial conditions for both networks. In order to avoid this inconvenience, we could compare nodes that play similar relative roles in their corresponding networks according to some structural criterion. Following this idea, Figure 5 depicts the time evolution of $\|u_n - u_f\|$ (where u_n is the distribution at time step n and u_f is the steady state) in terms of the time step for the initial conditions that give the maximal convergence speed for both networks (left panel in Figure 5) and for the initial conditions that give the minimal convergence speed for both networks (left panel in Figure 5). Note that the initial condition with the highest (lowest) convergence speed can be understood as

FIG. 4. Top panels: thresholded network from Spain (left) and Germany (right). Bottom panels: Evolution of the mean value of the difference vector between the initial and the actual states for all the cities in the network. The evolution of the states is obtained with the D'_i matrices obtained using the Shrapneling algorithm (left) and the gradient descent algorithm (right). For some points, the standard deviation is plotted in the error bars. Spain (red) and Germany (blue). For the sake of clarity in both plots we have used logarithmic scale.

the best (worst) located city in terms of optimal location for placing a factory that guarantees the best (worst) efficient delivery.

Figure 5 shows also the time evolution by using either Shrapneling and gradient descent algorithms, and again, it is shown that the gradient descent algorithm gives much better

results than the Shrapneling algorithm for both networks and for the fastest and lowest convergence speed. In addition to this, Figure 5 also confirms that the simplified German logistic network gives better delivery's performance than the simplified Spanish one, either for the fastest or lowest convergence speeds.

TABLE I. Top-five ranking of the five Spanish cities for which the convergence from its initial state, that is, all the product is in this city, to the steady state takes less iterations, by using either Shrapneling or gradient descent algorithms.

	Spain	
	Shrapneling	GDA
1	Madrid	Toledo
2	Ávila	Ávila
3	Badajoz	Madrid
4	Cáceres	Cuenca
5	Guadalajara	Cáceres

TABLE II. Top-five ranking of the five German cities for which the convergence from its initial state, that is, all the product is in this city, to the steady state takes less iterations, by using either Shrapneling or gradient descent algorithms.

	Germany	
	Shrapneling	GDA
1	Leipzig	Erfurt
2	Kassel	Kassel
3	Koblenz	Köln
4	Trier	Essen
5	Frankfurt/Main	Aachen

FIG. 5. Temporal evolution of the system for the fastest (left) and the slowest (right) cities from Spain and Germany. The matrices D_r have been obtained using the algorithms presented in this work. For the sake of clarity in both plots we have used logarithmic scale.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a mathematical model for the analysis of logistic networks based on the spectral analysis of dynamical systems. The distribution model is considered as the steady state of a dynamical system, understood as a weighted tensor supported on a transportation network. Given a (fixed) transportation network that connects n cities and a weighted matrix on such a graph that quantifies the amount of some given products that should be delivered from the supply depot of one city to its neighbors, the existence of a steady state (that corresponds to the final distribution of the product along the cities) is characterized in terms of structural properties and the convergence speed to the steady state is considered in terms of spectral properties of the weighted matrices.

The main achievement of this paper deals with the inverse distribution problem: Given a transportation network that connects n cities, some of them corresponding to manufacturers of some products, and a final desired amount of each product that should be delivered to each city, we should find (if it exists) a weighted tensor supported on the transportation network whose steady state corresponds to the desired amount of product to be delivered to each city. Some theoretical sharp results about the existence of such matrices in terms of convex geometry techniques have been proved. Furthermore, two efficient numerical algorithms for computing such matrices with optimal distribution speed are included in this paper, by using results from computational geometry and numerical optimization (based, for example,

on the well known gradient descent technique of Cauchy's work).

The main differences between the two numerical algorithms presented in the paper are analyzed and illustrated by considering two real logistic networks: the simplified (continental) Spanish and the simplified German logistic networks. In general, the numerical tests presented show that both methods behave quite differently from a numerical point of view. While the gradient descent algorithm gives better solutions (in terms of convergence speed) than the Sharpneling algorithm, the computational cost of the former algorithm is significantly higher than the later one, especially for transportation networks with many cities. In addition to this, it has been shown that the efficiency of both methods strongly depends on the steady state chosen, as it is depicted in Figure 3, since the proportion of possible solutions obtained from the methods has significant variations depending on the steady state. The influence of the structural properties of the underlying transportation network is pointed out by the fact that the convergence speed can seriously vary for two similar transportation networks (such as the Spanish and German transportation networks which have similar size).

The comparison between the simplified (continental) Spanish and the simplified German logistic networks shows that in this particular case, the convergence speed in the German network is higher than in the Spanish case for both methods, which suggests that the structural properties of the underlying German's transportation network fit better to the optimal product distribution than the Spanish one. The analysis of the better location for a manufacturer in each network (included in Tables I and II) is consistent, since there are no big discrepancies between the results obtained by both methods, and the ranking of the top-five cities corresponds to cities in the center of the transportation system.

The use of this mathematical formalism in the analysis of logistic networks and the tools and results included in this paper open new optimization questions about the interplay between the structural properties of the underlying transportation network and the distribution of products, including the optimal locations for a family of several manufacturers, the resilience of the distribution system or the evolution of the distribution process when the transportation network changes (due to some improvements or failures in their structure), among many other. These topics and others should be considered and analyzed carefully in future works.

ACKNOWLEDGMENTS

This work has been partially supported by Project Nos. MTM2012-32670, MTM2014-59906, and FIS2013-40653P (Spanish Ministry) and the grant URJC-Grupo de Excelencia Investigadora GARECOM (2014-2016).

¹H. Arndt, *Supply Chain Management* (Springer-Verlag, Berlin, 2004).

²D. Avis and K. Fukuda, *Discrete Comput. Geom.* **8**(3), 295–313 (1992).

³S. Boccaletti, G. Bianconi, R. Criado, C. I. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin, *Phys. Rep.* **544**, 1–122 (2014).

⁴S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, *Phys. Rep.* **424**, 175–308 (2006).

- ⁵M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms* (John Wiley & Sons, 2013).
- ⁶G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins Studies in the Mathematical Sciences (Johns Hopkins University Press, Baltimore, MD, 2013).
- ⁷D. M. Lambert and M. C. Cooper, *Ind. Mark. Manage.* **29**(1), 65–83 (2000).
- ⁸T. H. Matheiss and D. S. Rubin, *Math. Oper. Res.* **5**(2), 167–185 (1980).
- ⁹C. Meyer, *Matrix Analysis and Applied Linear Algebra* (SIAM, 2000).
- ¹⁰P. M. Reyes, *Appl. Math. Comput.* **168**, 1419–1431 (2005).
- ¹¹D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi, *Designing and Managing the Supply Chain: Concepts, Strategies and Cases*, 2nd ed. (McGraw-Hill, New York, NY, 2003).
- ¹²R. E. Tarjan, *SIAM J. Comput.* **1**(2), 146–160 (1972).