

PAPER

# Models for information propagation on graphs

Oliver R. A. Dunbar<sup>1</sup>, Charles M. Elliott<sup>2</sup> and Lisa Maria Kreusser<sup>3</sup> 

<sup>1</sup>California Institute of Technology, Pasadena, CA, USA

<sup>2</sup>Mathematics Institute, University of Warwick, Coventry, UK

<sup>3</sup>Department of Mathematical Sciences, University of Bath, Bath, UK

**Corresponding author:** Lisa Maria Kreusser; Email: [lmk54@bath.ac.uk](mailto:lmk54@bath.ac.uk)

**Received:** 11 May 2023; **Revised:** 26 September 2024; **Accepted:** 22 December 2024; **First published online:** 24 January 2025

**Keywords:** wave front sets in context of PDEs; Hamilton–Jacobi equations; shortest paths; numerical methods for difference equations

**2020 Mathematics Subject Classification:** 35A18, 35F21 (Primary); 52B05, 68R10 (Secondary)

## Abstract

We propose and unify classes of different models for information propagation over graphs. In a first class, propagation is modelled as a wave, which emanates from a set of *known* nodes at an initial time, to all other *unknown* nodes at later times with an ordering determined by the arrival time of the information wave front. A second class of models is based on the notion of a travel time along paths between nodes. The time of information propagation from an initial *known* set of nodes to a node is defined as the minimum of a generalised travel time over subsets of all admissible paths. A final class is given by imposing a local equation of an eikonal form at each *unknown* node, with boundary conditions at the *known* nodes. The solution value of the local equation at a node is coupled to those of neighbouring nodes with lower values. We provide precise formulations of the model classes and prove equivalences between them. Finally, we apply the front propagation models on graphs to semi-supervised learning via label propagation and information propagation on trust networks.

## 1. Introduction

Information propagation (also known as diffusion, cascade or spread) is of great importance in complex networks where, given information at a small number of nodes of the network, the aim is to understand the propagation to all the nodes. Social media networks provide typical examples including the breaking of a news story and the spread of product advertisements, internet memes and misinformation to different users. The ability to predict propagation plays a key role in tasks such as informing how to seed information for obtaining maximal coverage and influence [24, 26], or for identifying likely sources of information provided that the times are given when the information was received [34]. Models may be used for control and management of the propagation.

Our starting point is to model the underlying network as a given graph. The aim of this work is to formulate models for inspired by the propagation of waves passing through continuous media. Elements of the approach are that information has either arrived at a graph vertex or not, that information is transmitted to a node only from neighbouring nodes at which information has arrived already and that there is an arrival time for each node. These models for information propagation can then be used in applications ranging from social media networks to semi-supervised learning.



### 1.1. Continuum front propagation

In the continuum setting, there are three common viewpoints for modelling waves: front propagation, first arrival times and local equations. To introduce these viewpoints, we consider an open bounded domain  $\Omega \subset \mathbb{R}^d$  for  $d \geq 1$  with a Lipschitz boundary  $\Gamma$ , a given point  $x_0 \in \Omega$  and a continuous, positive function  $s: \bar{\Omega} \rightarrow \mathbb{R}$  which can be regarded as the impedance of the medium  $\bar{\Omega}$ .

A first approach proposes a propagating front separating the region for which the wave has arrived from the remainder. The fronts initiate at  $x_0$  and are characterised by being level surfaces of the arrival time from  $x_0$ . The impedance  $s(x)$  is specific for the underlying medium and controls the additional time required for the front to travel through the medium at  $x$ . We also refer to this approach as front propagation.

A second classical approach consists of formulating a model based on finding the smallest travel time over a set of possible paths and hence results in an optimisation problem. The aim of this model is to determine the shortest travel time along any path from  $x_0$  to every  $x \in \bar{\Omega}$ ,  $x \neq x_0$ , in the medium  $\bar{\Omega}$  for a given impedance  $s$ . This task can be expressed as the minimisation problem

$$u(x) = \inf_{\substack{\xi \in W^{1,\infty}([0,1],\bar{\Omega}), \\ \xi(0)=x_0, \quad \xi(1)=x}} \left\{ \int_0^1 s(\xi(r)) \|\xi'(r)\|_2 dr \right\}, \quad (1.1)$$

cf. [11], where  $\|\cdot\|_2$  denotes the 2-norm in  $\mathbb{R}^d$  and  $\xi(\cdot)$  is a parameterised path in the Sobolev space  $W^{1,\infty}$ . Note that  $\xi \in W^{1,\infty}([0,1])$  is locally Lipschitz continuous and hence the integral in (1.1) is well defined. Since large values of  $s$  slow down the movement and increase the travel time within the medium, we sometimes refer to  $s$  as the slowness function, while its inverse  $\frac{1}{s}$  can be regarded as a velocity. We also refer to this approach as first arrival times.

A third approach arises when regarding an optimal value  $u$  of (1.1) as a solution to the eikonal equation, an isotropic static Hamilton-Jacobi partial differential equation. The eikonal equation is given by

$$\|\nabla u\|_2 = s \quad \text{in } \Omega \setminus \{x_0\} \quad (1.2)$$

with boundary conditions

$$\begin{aligned} u(x_0) &= 0, \\ \nabla u(x) \cdot \nu(x) &\geq 0 \quad \text{for } x \in \Gamma, \end{aligned} \quad (1.3)$$

where  $\nu$  is the unit outer normal to  $\Gamma$ . We also refer to this approach as a local equation. Also it is possible to pose and solve eikonal equations on connected (sub)Riemannian manifolds, see e.g. [23].

These three approaches of wave propagation in continuum settings have been exploited to advance different fields of research. The optimisation over paths (also referred to as first arrival times) arises in modelling of optimal logistics such as accessibility, evacuation planning, robot navigation and ray models. The study of the graph eikonal equation (i.e. a local equation) is of importance for proving theoretical results on existence and uniqueness of solutions with certain monotonicity properties. Efficient numerical methods such as fast marching algorithms take advantage of the front propagation approach when solving the continuum eikonal equation [27, 37, 38]. This demonstrates that diverse perspectives on modelling waves are crucial in the continuum setting for getting more insights into modelling, analysis and numerical methods of the underlying continuum problem.

In contrast to the continuum setting, only a scattered picture is currently available for graphs, including shortest paths, Dijkstra's algorithm and graph-eikonal models. Motivated by the continuum setting, the aim of this work is to propose and unify corresponding perspectives in the graph setting. We formulate and relate several classes of models based on front propagation, first arrival time over sets of admissible paths and a local equation considering arrival times at a given node and its neighbours. As part of this, we introduce appropriate graph-based generalisations of the continuum counterparts for the three classes of models. In the context of the Dijkstra algorithm, for instance, the Dijkstra algorithm can be regarded as a front propagation model. For the local equation, we replace (1.2)–(1.3) in the continuum

setting by a graph-based version of the local equation

$$\|\nabla u\|_p = s \quad \text{in } \Omega \setminus \{x_0\}$$

for  $p = \infty$  with boundary conditions (1.3), which leads to an  $\ell^\infty$  graph-eikonal equation. We also propose a first arrival time model, based on the travel time over paths, and prove its equivalence to Dijkstra's algorithm. Motivated by the special case  $p = \infty$  for the local equation, we derive front propagation, first arrival time and local equations for other cases of  $p$ . The main contribution of this paper is to model wave propagation in the graph-based setting using three perspectives (front propagation, first arrival times and local equations). We prove the equivalence of the models for special cases of  $p$ . It is important to note that in the models we do not embed the vertices in any ambient Euclidean space.

## 1.2. Applications

It is natural to introduce the concept of information propagation to data classification and semi-supervised learning. Motivated by this, we apply front propagation on graphs to classical examples in semi-supervised learning such as the *Two moons problem* and *Text classification datasets*. Here the information consists of a given finite set of labels, and the aim is to label all vertices in a graph based on the knowledge of the labels on given small number of nodes. Labels are attached by ordering the magnitudes of the arrival times of the information. In addition, we apply information propagation to *Trust networks*. These are social networks whose users rate each other by trustworthiness. Examples include collaborative networks such as a community of software engineers, or partners of a transaction within cryptocurrency exchanges. Applied to the software community dataset *soc-advogato* [35], we show that information propagation can use local trust information to create rankings of any collaborator on the network. Our model-rankings are resistant to Sybil attack [1, 15, 44], where users artificially inflate their reputation, by creating a group of fake users to giving them positive ratings.

## 1.3. PDE approaches

Many computational methods for semi-supervised and unsupervised classification [2, 6, 45] are based on variational models and PDEs [22]. Examples include algorithms based on phase fields [4] and the MBO scheme [31], as well as  $p$ -Laplacian equations [20, 29]. In a series of papers, Elmoataz et al. [12, 13, 19, 39] postulate discrete eikonal equations and investigate label propagation on graphs with applications in imaging and machine learning. Current analytical results include an investigation of viscosity solutions for Hamilton-Jacobi equations on networks [10], the well-posedness of nonlinear PDEs such as the Eikonal equation on finite graphs [32] and an approximation scheme for an eikonal equation on a network [9], producing an approximation of shortest paths to the boundary. In addition, limits and consistency of non-local and graph approximations to the time-dependent (local) eikonal equation have been studied in [21]. The robustness of the solution to the eikonal equation for  $p = 1$  and its convergence to the shortest path distance as  $p \rightarrow \infty$  is shown in [8].

## 1.4. Contributions

Our contributions are as follows:

- Derivation of general model formulations for three perspectives (front propagation models, first arrival time models, and local equations) in the graph-based setting that include established models (Dijkstra's algorithm, shortest paths, and  $\ell^p$  graph-eikonal equations for  $p \geq 1$ ) as special instances.
- Unification of the three perspectives in the graph-based setting by proving equivalence of the models (front propagation, first arrival times and discrete generalised eikonal models) depending on  $p$ .
- Application of front propagation on a weighted social network to calculate metrics of trust securely.

- Application of front propagation on graphs to classical problems in semi-supervised learning for point cloud datasets (two moons problem, text classification datasets Cora and CiteSeer).

## 1.5. Outline

We introduce several models for travel times on a graph in Section 2. Equivalences between certain instances of the models are established in Section 3. In Section 4.1, we apply information propagation to trust networks. The use of front propagation on graphs to semi-supervised learning via label propagation is illustrated in Section 4.2. Finally, we make some concluding remarks in Section 5.

## 1.6. Notation

Following the terminology and setting in [17, 19, 22], we consider a finite, connected weighted graph  $G = (V, E, w)$  with vertices  $V = \{1, \dots, n\}$ , edges  $E \subset V^2$  and nonnegative edge weights  $w$ . We assume that the graph is simple, i.e. there exists at most one edge between any two vertices. We suppose that there is a decomposition of  $V := \partial V \cup \overset{\circ}{V}$  into two disjoint non-empty sets  $\partial V$  and  $\overset{\circ}{V}$ . The edge between node  $i$  and node  $j$  is denoted by  $(i, j)$ . For ease of notation, we regard the weights  $w$  as a weight matrix  $w \in \mathbb{R}^{n \times n}$  with entries  $w_{ij}$ , where we assume that there exists an edge  $(i, j) \in E$  if and only if  $w_{ij} > 0$ , while  $w_{ij} = 0$  if  $(i, j) \notin E$ . Since  $G$  is not necessarily undirected,  $w_{ij} \neq w_{ji}$  in general. This framework also includes unweighted graphs corresponding to the cases in which  $w_{ij} = 1$  for all  $(i, j) \in E$ . Given a graph  $G$ , we denote by  $N(i) \subset V$  the set of neighbours of node  $i \in V$ . We define  $j \in N(i)$  if there exists an edge  $(j, i) \in E$ , and in general this does not imply existence of  $(i, j) \in E$ . The direction of this relationship is chosen for convenient notation in the following. We introduce the notion of a path from node  $x \in V$  to  $y \in V$  and write  $p_{x,y} = (x = i_1, \dots, y = i_{n(p_{x,y})})$  for a path with  $n(p_{x,y})$  nodes and  $n(p_{x,y}) - 1$  edges  $(i_{m-1}, i_m) \in E$  for  $m = 2, \dots, n(p_{x,y})$  such that all nodes  $i_m$  for  $m \in \{1, \dots, n(p_{x,y})\}$  are distinct, i.e. a path must not self-intersect. Due to the assumption that the graph  $G$  is connected, for every  $x, y \in V$  there exists a path  $p_{x,y}$  connecting  $x$  and  $y$ , i.e. there exists  $n(p_{x,y}) > 1$  such that  $p_{x,y} = (x = i_1, \dots, y = i_{n(p_{x,y})})$  is a path with edges  $(i_{m-1}, i_m) \in E$  for  $m = 2, \dots, n(p_{x,y})$ . For a graph with  $|V| = n$  nodes, we denote by  $\mathcal{H}^n$  the function space of all functions defined on  $V$ , i.e. all  $v \in \mathcal{H}^n$  are of the form  $v: V \rightarrow \mathbb{R}$ . For  $v \in \mathcal{H}^n$ , we write  $v_x = v(x)$  for  $x \in V$ . We also assume that there is a given *slowness* function  $s \in \mathcal{H}^n$  with  $s \geq 0$ .

## 2. Description of models

In this section, we propose several models for the propagation of information on graphs. The common elements of the models are

- We suppose that either all information has arrived at a vertex or none.
- We introduce the variable  $u \in \mathcal{H}^n$  with  $u_i$  for  $i \in V$  to denote the *arrival time* of information at vertex  $i$ .
- We assume that  $u$  is prescribed on  $\partial V$  and we set  $u = 0$  on  $\partial V$ , though in general the models can accommodate a wider class of boundary conditions.
- We suppose that information propagation is local. That is, information arrives at a vertex only by propagation from a neighbouring vertex for which information has arrived. Thus, there is a unique travel time  $u_i$  at each node  $i$  that can only depend on travel times at nodes  $j \in N(i)$  with  $u_j < u_i$ .
- The edge weights reflect the distance or resistance to propagation along an edge.
- The function  $s \in \mathcal{H}^n$  is a measure of slowness or resistance associated with each vertex.

The aim of a model is to associate a travel time  $u_i$  with each vertex of the graph. Since the graph is finite,  $u = \{u_i, i = 1, 2, \dots, n\}$  attains an unknown number of  $J + 1 \in \mathbb{N}$  distinct values consisting of

prescribed initial data  $U_0 \in \mathbb{R}$  and unknown values  $U_1, \dots, U_J \in \mathbb{R}$  ordered so that  $U_0 < \dots < U_J$ . We set  $V_0 := \partial V$  as the set of initially labelled vertices and prescribe the initial data  $U_0$ , i.e.  $u_i = U_0$  for all  $i \in V_0$ . In the following, we set  $U_0 = 0$ . We consider three classes of models. The first class of models is based on the propagation of discrete fronts from an initial front  $\partial V$  (Model 1). The second class of models considers first arrival times of sets of paths that link vertices in the initial set  $\partial V$  to vertices in  $\hat{V} = V \setminus \partial V$  (Model 2). For the third class of models, we postulate a generalised discrete  $\ell^p$  eikonal equation model (Model 3) depending on parameter  $p$ . We mainly focus on  $p \in \{1, 2, \infty\}$  below. Note that some of the model instances may look rather complicated. However, the main motivation is to unify graph-based models from three perspectives (front propagation, first arrival times, and local equations) by proving their equivalence.

## 2.1. Front propagation models

In this approach, we view information propagation as an evolving front, i.e. a boundary that separates the region for which the wave has arrived from the remainder. We decompose the set  $\hat{V}$  of initially unlabelled vertices into  $J$  disjoint sets  $V_1, \dots, V_J$  such that for  $j \in \{1, \dots, J\}$  all vertices  $i \in V_j$  satisfy  $u_i = U_j$ . We define *known* sets  $K_0, \dots, K_J$  and *candidate* sets  $C_0, \dots, C_J$  as follows:

$$K_l = \bigcup_{j \in \{0, \dots, l\}} V_j, \quad C_l = \bigcup_{j \in K_l} N(j) \setminus K_l.$$

Under the assumption that  $U_j$  and  $V_j$  for  $j = 0, \dots, k-1$  are known, implying that the value of  $u_i$  for all  $i \in K_{k-1}$  is known, our task is to determine  $U_k$  and  $V_k$ . The front  $F_{k-1}$  consists of all vertices in  $K_{k-1}$  with neighbours in  $C_{k-1}$  and with  $F_0 = V_0$ . We determine candidate values  $\tilde{u}_i$  for each  $i \in C_{k-1}$  using a model (specified below), and we define  $U_k$  by choosing the smallest candidate value in the candidate set  $C_{k-1}$ :

$$U_k := \min_{i \in C_{k-1}} \tilde{u}_i. \quad (2.1)$$

We then define  $V_k \subset C_{k-1}$  to be the set where the minimum is attained and we set  $u_i = U_k$  for all  $i \in V_k$ . The above procedure depends on the definition of candidate values  $\tilde{u}_i$  for  $i \in C_{k-1}$ . We define relationships for  $\tilde{u}_i$  that depend upon the set  $N(i) \cap K_{k-1}$ . Using (2.1), the values  $U_1, \dots, U_L$  of the solution  $u$  can then be determined. By construction, the solution  $u$  is unique for the function  $i \mapsto \tilde{u}_i$ .

### 2.1.1. Model 1(i)

Given the known arrival time  $u_j$  for  $j \in K_{k-1}$ , and let  $j \in N(i)$  so edge  $(j, i)$  exists, then a candidate for the arrival time at  $i$ , is given by  $u_j + \frac{s_i}{w_{j,i}}$ . Choosing the smallest value of all these possible candidate values results in the candidate

$$\tilde{u}_i = \min_{j \in N(i) \cap K_{k-1}} \left\{ u_j + \frac{s_i}{w_{j,i}} \right\} \quad (2.2)$$

for  $i \in C_{k-1}$ . Here,  $u_j + \frac{s_i}{w_{j,i}}$  is the sum of the first arrival time  $u_j$  at node  $j$  and  $\frac{s_i}{w_{j,i}}$ , which is the travel time from  $j$  to  $i$  along edge  $(j, i)$ . The travel time along  $(j, i)$  only depends on the slowness  $s_i$  at the endpoint of  $(j, i)$  and the edge weight  $w_{j,i}$ . The term  $\frac{s_i}{w_{j,i}}$  is inspired from the continuum setting (1.1) which suggests that the travel time along an edge  $(i, j)$  is antiproportional to the velocity  $\frac{1}{s_i}$  and hence proportional to  $s_i$ . (1.1) also suggests that the travel time is proportional to the length of an edge and thus proportional to  $\frac{1}{w_{ij}}$  if we regard  $w_{ij}$  as a characterisation of the connectivity of vertices  $i$  and  $j$ .

As the minimum in (2.2) can be associated with the  $\ell^\infty$ -norm, we will also see later that this model is equivalent to the  $\ell^\infty$  graph-eikonal equation.

### 2.1.2. Model I(ii)

While only the smallest neighbouring value has been considered in (2.2), which can be associated with the  $\ell^\infty$ -norm, we consider a more averaging approach in the following instance of a front propagation model motivated by weighing neighbouring known values in an  $\ell^2$ -sense. We define  $z_i^2 := \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2$  for  $i \in C_{k-1}$ , i.e.  $z_i^2 = \|(w_{j,i})_{j \in N(i) \cap K_{k-1}}\|_2^2$ . For  $i \in C_{k-1}$ , we set

$$\tilde{u}_i = \mu_i + \sqrt{\frac{s_i^2}{z_i^2} - \sigma_i^2}. \quad (2.3)$$

Here,

$$\mu_i = \frac{1}{z_i^2} \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 u_j$$

can be regarded as the weighted mean travel time between any node  $j \in N(i) \cap K_{k-1}$  and node  $i$  as  $\frac{1}{z_i^2} \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 = 1$ . The weighted mean travel time to  $i$  balances the travel time to each known node  $j$  with the squared weights between  $i$  and  $j$ . Further, we set

$$\sigma_i^2 = \sum_{j \in N(i) \cap K_{k-1}} \left( \frac{w_{j,i}^2}{z_i^2} u_j^2 \right) - \mu_i^2$$

as the variance of the weighted mean travel time.

As an interpretation of (2.3), we can regard the wavefront of information travelling simultaneously from all known nodes  $j \in K_{k-1}$  to candidate node  $i$  where the averaged wavefront (in the  $\ell^2$ -sense) depends on the weighted mean travel time  $\mu_i$  and its variance  $\sigma_i^2$ . With this model, one can interpret the neighbours' values as forming an estimate of a candidate value  $\tilde{u}_i$  from below, with a weighted mean square error  $(\tilde{u}_i - \mu_i)^2 + \sigma_i^2 = \frac{s_i^2}{z_i^2}$ . We will also see later that this model is equivalent to the  $\ell^2$  graph-eikonal equation.

### 2.1.3. Model I(iii)

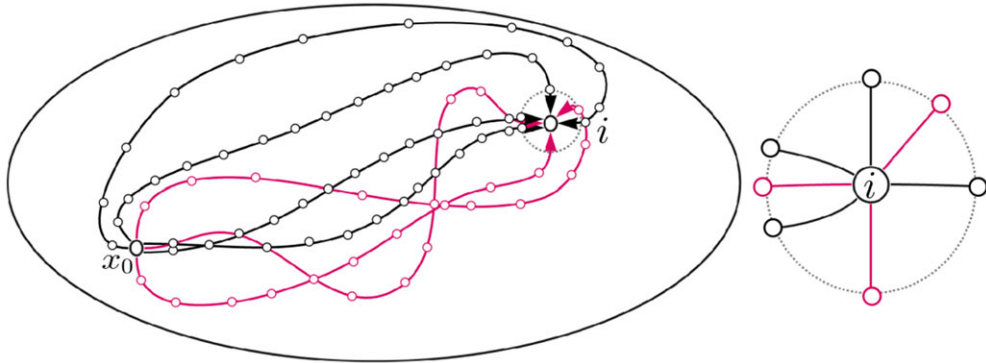
Similarly to (2.3), we consider an averaging approach in the following instance of a front propagation model, but here we weigh neighbouring known values in an  $\ell^1$ -sense. For  $i \in C_{k-1}$ , we define  $M_{i,k} = |N(i) \cap K_{k-1}|$  and  $y_i := \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}$ , i.e.  $y_i = \|(w_{j,i})_{j \in N(i) \cap K_{k-1}}\|_1$ . We set

$$\tilde{u}_i = \frac{1}{y_i} \sum_{j \in N(i) \cap K_{k-1}} (w_{j,i} u_j) + \frac{s_i}{y_i} = \frac{1}{y_i} \sum_{j \in N(i) \cap K_{k-1}} w_{j,i} \left( u_j + \frac{s_i}{M_{i,k} w_{j,i}} \right) \quad (2.4)$$

for  $i \in C_{k-1}$ . The first term in (2.4) can be regarded as a weighted mean travel time to  $i$ , obtained by balancing the travel time from each known node  $j$  with the weight  $w_{j,i}$  between  $j$  and  $i$ , while the second term  $\frac{s_i}{y_i}$  can be interpreted as bias. Like for the other instances, we can interpret (2.4) as the wavefront of information travelling simultaneously from all known nodes  $j \in K_{k-1}$  to candidate node  $i$  where the averaged wavefront (in the  $\ell^1$ -sense) depends on the weighted mean travel time and its bias. We will also see later that this model is equivalent to the  $\ell^1$  graph-eikonal equation.

## 2.2. First arrival times

In this approach, we optimise travel times over path sets as a generalisation of travel times over paths. For this, we define useful quantities for describing path sets. Then, we define some generalised travel time models and first arrival times over path sets. In Remark 2.2, we show how this generalises the standard travel time defined over paths. For two nodes  $x_0, i \in V$ , let  $\mathbb{P}_{x_0,i}$  be the set of admissible paths  $p_{x_0,i}$  from  $x_0$  to  $i$ . Since the graph  $G = (V, E, w)$  is connected,  $\mathbb{P}_{x_0,i}$  is non-empty. Let  $P_{x_0,i} \subset \mathbb{P}_{x_0,i}$  denote a non-empty



**Figure 1.** An illustration of a path set and its truncation. On the left we represent the set of all paths  $\mathbb{P}_{x_0,i}$  between two nodes  $x_0$  and  $i$  with black arrows from  $x_0$  to  $i$ . We represent a path set  $P_{x_0,i} \subset \mathbb{P}_{x_0,i}$  in pink. In particular, the path set  $P_{x_0,i}$  contains three paths. On the right of the figure, we zoom into the neighbourhood  $N(i)$ , represented as nodes on dotted circle; the pink nodes on the dotted circle represent the penultimate truncation  $K(P_{x_0,i}) \subset N(i)$  of the path set. The pink edges therefore can be written as  $(j, i)$  such that  $j \in K(P_{x_0,i})$ .

subset of paths from  $x_0$  to  $i$ , and we refer to  $P_{x_0,i}$  as a *path set*. We define the *penultimate truncation* of a path  $p_{x_0,i} \in \mathbb{P}_{x_0,i}$  as a path  $p_{x_0,j}$ , where  $j \in N(i)$  and  $p_{x_0,i} = (p_{x_0,j}, (j, i))$ . Similarly, for a path set  $P_{x_0,i}$ , we define the *penultimate truncations* of  $P_{x_0,i}$  as the set  $\{p_{x_0,j} : j \in K(P_{x_0,i})\}$  where  $K(P_{x_0,i}) \subset N(i)$  such that for every  $j \in K(P_{x_0,i})$  there exist a path  $p_{x_0,j}$  and a path  $p_{x_0,i} \in P_{x_0,i}$  such that  $p_{x_0,i} = (p_{x_0,j}, (j, i))$ . Note, unlike the set  $N(i)$  which depends only on the graph structure,  $K(P_{x_0,i})$  depends on the choice of the path set  $P_{x_0,i}$ . An illustration of a path set and its penultimate truncation is shown in Figure 1.

We assume that there exists a formula for a generalised travel time  $T(Q)$  for any path set  $Q \subset \mathbb{P}_{x_0,i}$ . Some specific examples are introduced below. We define  $u_i$  for  $i \in V$ , as the first arrival travel times over path sets by

$$u_i = \min_{x_0 \in \partial V} \min_{P_{x_0,i} \subset \mathbb{P}_{x_0,i}} T(P_{x_0,i}). \quad (2.5)$$

For boundary nodes  $x_0 \in \partial V$ , we set  $u_{x_0} = 0$ . The inner minimisation in (2.5) is not over paths  $p_{x_0,i} \in \mathbb{P}_{x_0,i}$ , but over *path sets*  $P_{x_0,i} \subset \mathbb{P}_{x_0,i}$ .

We define a travel time  $T$  over a path set  $P_{x_0,i}$  with a local formula over the penultimate truncations of  $P_{x_0,i}$ . In particular,  $T(P_{x_0,i})$  is calculated as a function of  $T(P_{x_0,j}^i)$  with  $j \in K(P_{x_0,i})$ , where  $P_{x_0,j}^i = \{p_{x_0,j} \in \mathbb{P}_{x_0,i} : (p_{x_0,j}, (j, i)) \subset P_{x_0,i}\}$ . By definition  $P_{x_0,j}^i$  is also a path set. Since all nodes of a path are distinct by definition, for all  $p_{x_0,j} \in P_{x_0,j}^i$  we have  $i \notin p_{x_0,j}$ .

The models we propose for the travel time  $T$  share similarities with the front propagation models 1(i), 1(ii), 1(iii) in Section 2.1 and are specified further below.

### 2.2.1. Model 2(i)

Similar to Model 1(i) in (2.2), we define

$$T(P_{x_0,i}) = \min_{j \in K(P_{x_0,i})} \left\{ T(P_{x_0,j}^i) + \frac{s_i}{w_{j,i}} \right\}. \quad (2.6)$$

We will see later that this model is equivalent to the  $\ell^\infty$  graph-Eikonal equation.



### 2.2.2. Model 2(ii)

Similar to Model 1(ii) in (2.3), we consider

$$T(P_{x_0,i}) = \mu_{x_0,i} + \sqrt{\frac{s_i^2}{z_{x_0,i}} - \sigma_{x_0,i}^2} \quad (2.7)$$

where

$$z_{x_0,i} = \sum_{j \in K(P_{x_0,i})} w_{j,i}^2, \quad \mu_{x_0,i} = \frac{1}{z_{x_0,i}} \sum_{j \in K(P_{x_0,i})} w_{j,i}^2 T(P_{x_0,j}^i)$$

and

$$\sigma_{x_0,i}^2 = \sum_{j \in K(P_{x_0,i})} \left( \frac{w_{j,i}^2}{z_{x_0,i}} (T(P_{x_0,j}^i))^2 \right) - \mu_{x_0,i}^2.$$

We will see later that this model is equivalent to the  $\ell^2$  graph-Eikonal equation.

### 2.2.3. Model 2(iii)

Similar to Model 1(iii) in (2.4), we define

$$T(P_{x_0,i}) = \frac{1}{y_{x_0,i}} \sum_{j \in K(P_{x_0,i})} w_{j,i} T(P_{x_0,j}^i) + \frac{s_i}{y_{x_0,i}} \quad (2.8)$$

where  $y_{x_0,i} := \sum_{j \in K(P_{x_0,i})} w_{j,i}$ . We will see later that this model is equivalent to the  $\ell^1$  graph-Eikonal equation.

**Remark 2.1.** Due to the assumption that the graph  $G$  is connected and the weights  $w_{j,i}$  are positive, there exists a solution to (2.5) for all the above choices of the travel time  $T$ . Clearly, first arrival time solutions are well defined and unique. However, the minimising path sets are not unique in general.

**Remark 2.2.** Consider a singleton path set  $P_{x_0,i} = \{p_{x_0,i}\} = \{(x_0 = i_1, \dots, i = i_M)\}$ . We observe that the value of  $T(P_{x_0,i})$  calculated using models 2(i), 2(ii) or 2(iii) is equal to the following:

$$\begin{aligned} T(\{p_{x_0,i}\}) &= T(\{p_{x_0,i_{M-1}}\}) + \frac{s_{i_M}}{w_{i_{M-1},i_M}} = T(\{p_{x_0,i_{M-1}}\}) + T(\{(i_{M-1}, i_M)\}) \\ &= \sum_{m=2}^M T(\{(i_{m-1}, i_m)\}), \end{aligned} \quad (2.9)$$

where we used that the models 2(i), 2(ii) and 2(iii) satisfy

$$T(\{(i_{m-1}, i_m)\}) = \frac{s_{i_m}}{w_{i_{m-1},i_m}}. \quad (2.10)$$

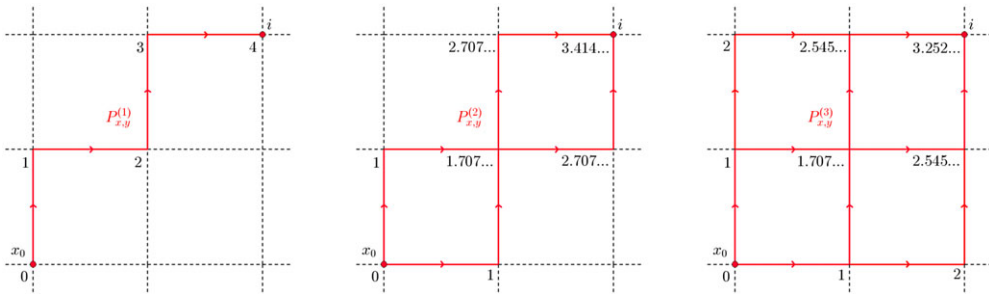
If we suppose that  $w_{i_{m-1},i_m}$  characterises the connectivity between nodes  $i_{m-1}$  and  $i_m$ , and thus  $\frac{1}{w_{i_{m-1},i_m}}$  is proportional to the travel time, the form of the travel time (2.9) can be regarded as a discretisation of  $\int_0^1 s(\xi(r)) \|\xi'(r)\|_2 dr$  in (1.1).

Classically, there is a known relationship between the discretisation of problem (1.1) and the minimisation problem

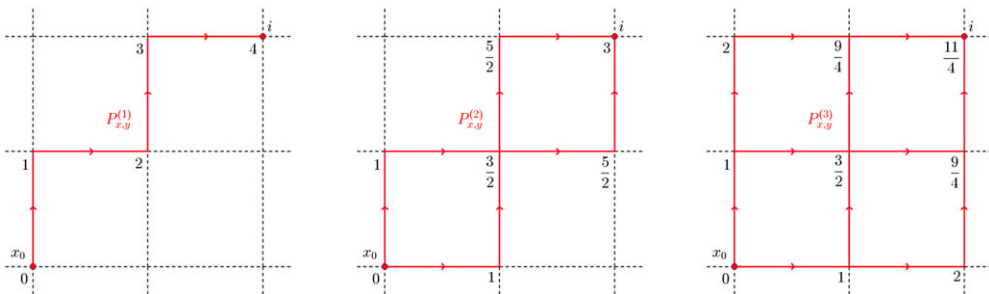
$$u_i = \min_{x_0 \in \partial V} \min_{p_{x_0,i} \in \mathbb{P}_{x_0,i}} T(\{p_{x_0,i}\}), \quad (2.11)$$

where  $u_{x_0} = 0$  on boundary nodes  $x_0 \in \partial V$ . Under the assumption that only singleton sets  $P_{x_0,i} = \{p_{x_0,i}\}$  may be considered in (2.5), then (2.5) reduces to (2.11).





**Figure 2.** Three different path sets shown in red on a square grid with  $w_{j,i} = 1$  and  $s_i = 1$  for all nodes. The numbers correspond to the values of the generalised travel time  $T(P_{x_0,i}^{(i)})$  for model 2(ii) for each path set.



**Figure 3.** Three different path sets shown in red on a rectangular grid with  $w_{j,i} = 1$  and  $s_i = 1$  for all nodes. The numbers correspond to the values of the generalised travel time  $T(P_{x_0,i}^{(i)})$  for model 2(iii) for each path set.

To understand the behaviour of model 2(i) in (2.6), substituting its definition in (2.5), we obtain (2.11). Indeed,

$$u_i = \min_{x_0 \in \partial V} \min_{P_{x_0,i} \subset \mathbb{P}_{x_0,i}} \min_{j \in K(P_{x_0,i})} \left\{ T(P_{x_0,j}^i) + \frac{s_i}{w_{j,i}} \right\} \\ = \min_{x_0 \in \partial V} \min_{j \in K(\mathbb{P}_{x_0,i})} \left\{ T(P_{x_0,j}^i) + \frac{s_i}{w_{j,i}} \right\} = \min_{x_0 \in \partial V} \min_{P_{x_0,i} \in \mathbb{P}_{x_0,i}} T(\{P_{x_0,i}\}).$$

Thus, when using model 2(i), a minimisation over path sets is thus reduced to a minimisation over paths.

To understand the behaviour of models 2(ii) and 2(iii), we calculate the generalised travel time of some simple path sets over the square grid in two space dimensions with constant unit weights and slowness function; see Figures 2 and 3, respectively. In each case, we calculate the travel times for the three path sets  $P_{x_0,i}^{(1)}$ ,  $P_{x_0,i}^{(2)}$  and  $P_{x_0,i}^{(3)}$ , where  $x_0 = (0, 0)$  and  $i = (2, 2)$ . Let  $U$  and  $R$  be the paths travelling ‘up’ and ‘right’ from a node to a neighbour on the square grid. We set  $P_{x_0,i}^{(1)} = \{(U, R, U, R)\}$ ,  $P_{x_0,i}^{(2)} = P_{x_0,i}^{(1)} \cup \{(R, U, R, U)\}$  and  $P_{x_0,i}^{(3)} = P_{x_0,i}^{(2)} \cup \{(U, U, R, R), (R, R, U, U)\}$ , so these path sets have 1, 2 and 4 elements, respectively. We show the generalised travel time for path sets  $P_{x_0,i}^{(1)}$ ,  $P_{x_0,i}^{(2)}$  and  $P_{x_0,i}^{(3)}$  for models 2(ii) and 2(iii) in Figures 2 and 3, respectively. Here, the numbers at nodes along the different paths denote the generalised travel time from the origin  $x_0$  to the respective nodes. We see that  $P_{x_0,i}^{(3)}$  is optimal for model 2(ii) and 2(iii) among  $\{P_{x_0,i}^{(1)}, P_{x_0,i}^{(2)}, P_{x_0,i}^{(3)}\}$  as shown in Figures 2 and 3. In fact,  $P_{x_0,i}^{(3)}$  is an optimal path set for model 2(ii) and 2(iii) among all subsets of  $\mathbb{P}_{x_0,i}$  on the square grid.

The properties of minimising path sets are left to future investigation. Heuristically, we see that the travel times given by model 2(ii) or 2(iii) are small for path sets that contain short paths or paths, which have many cross-overs among themselves (i.e. multiple distinct paths pass through common nodes). Such

behaviour is observed in Figures 2 and 3, where the support of the minimizing paths is the rectangular lattice between nodes  $x_0$  and  $i$ .

**Remark 2.3.** The notion of a minimising path in (2.5) also includes the case of a single element of  $\partial V$ , which corresponds to one label, i.e.  $\partial V = \{x_0\}$  in which case

$$u_i = \min_{P_{x_0,i} \subset \mathbb{P}_{x_0,i}} T(P_{x_0,i}).$$

**Remark 2.4.** It is possible that Gromov's theory (e.g. [23]) provides a suitable framework with which to view these constructions. In [23], a metric space  $(G, d)$  is endowed with an additional length structure over curves between points in the space. A path-metric space is then defined if  $d(x, y)$  for  $x, y \in G$  is equal to the shortest length of the curve connecting the  $x$  and  $y$ . This theory applies to the path-distance metric  $d$  on a connected graph  $G$ , defining a length structure by (2.6). It will be an interesting future direction of research to see if (2.7) and (2.8) define length structures and along with a suitable metric  $d$  form a path-metric space.

### 2.3. Discrete generalised eikonal models

For  $i \in V$ , we define one-sided edge derivatives  $\nabla_w^+ u_i \in \mathbb{R}^{|N(i)|}$  by

$$\nabla_w^+ u_i = (w_{j,i}(u_i - u_j)^+)_j, \quad j \in N(i).$$

Set,

$$\|\nabla_w^+ u_i\|_p = \left( \sum_{j \in N(i)} (w_{j,i}(u_i - u_j)^+)^p \right)^{1/p} \quad \text{for } 1 \leq p < \infty, \quad (2.12)$$

and

$$\|\nabla_w^+ u_i\|_\infty = \max_{j \in N(i)} \{w_{j,i}(u_i - u_j)^+\}. \quad (2.13)$$

#### 2.3.1. Model 3(p)

Motivated by monotone discretisations of the continuum eikonal equation, we consider for any  $1 \leq p \leq \infty$ ,

$$\begin{aligned} \|\nabla_w^+ u_i\|_p &= s_i, & i \in \mathring{V}, \\ u_i &= 0, & i \in \partial V. \end{aligned} \quad (2.14)$$

Note that (2.14) with  $p = 2$  is of the same form as the continuum eikonal equation (1.2). We can rewrite (2.14) as

$$\begin{aligned} \sum_{j \in N(i)} (w_{j,i}(u_i - u_j)^+)^p &= s_i^p, & i \in \mathring{V}, \\ u_i &= 0, & i \in \partial V, \end{aligned} \quad (2.15)$$

for  $1 \leq p < \infty$ , and

$$\begin{aligned} \max_{j \in N(i)} \{w_{j,i}(u_i - u_j)^+\} &= s_i, & i \in \mathring{V}, \\ u_i &= 0, & i \in \partial V, \end{aligned} \quad (2.16)$$

for  $p = \infty$ . The models satisfy a monotonicity condition characteristic of discrete Hamilton-Jacobi equations (c.f. [11]). Using a monotonicity condition and comparison principles, it has been shown that the boundary value problems admit a unique solution and are well-posed, see [12] and [8, Th. 12]. The

**Table 1.** We summarise proved equivalences between the front propagation, arrival time (path and path set) and discrete Eikonal models

Front propagation		First arrival		Discrete Eikonal
Model 1(i)	$\Leftrightarrow$	Model 2(i)	$\Leftrightarrow$	Model 3( $p = \infty$ )
Model 1(ii)	$\Leftrightarrow$	Model 2(ii)	$\Leftrightarrow$	Model 3( $p = 2$ )
Model 1(iii)	$\Leftrightarrow$	Model 2(iii)	$\Leftrightarrow$	Model 3( $p = 1$ )

authors in [8, Th. 12] also construct sub- and supersolutions of the unique solution, resulting in explicit lower and upper bounds of the solution which are both linked to the graph distance.

Note that the  $\ell^\infty$  eikonal equation is related to shortest path graph distances that approximate geodesic distances. However, this is not the case for the  $\ell^p$  eikonal equation with  $p$  finite as interaction between neighbouring nodes is of importance here.

### 3. Relations between models

In this section, we investigate relations between the different modelling approaches, that is front propagation, first arrival time and discrete eikonal models, which are introduced in Section 2. The relationships we prove between the models are summarised in Table 1, and the proofs are provided in the following sections.

#### 3.1. Equivalence of front propagation and discrete eikonal models.

In this section, we show the equivalence of front propagation models (2.2), (2.3), (2.4) (i.e. models 1(i),(ii),(iii)) and discrete eikonal models (2.15) for  $p = 1, p = 2$  and (2.16) for  $p = \infty$  (i.e. models 3( $p = 1$ ), 3( $p = 2$ ), 3( $p = \infty$ )).

##### 3.1.1. Equivalence of models 1(i) and 3( $p = \infty$ )

Let  $i \in \mathring{V}$  be given. Hence, there exists  $k \in \{1, \dots, L\}$  such that  $i \in V_k$ . For this  $k$ , the definition of sets  $V_k, K_{k-1}$  and  $C_{k-1}$ , and of model 1(i) (2.2), give the value of  $u_i$  as

$$u_i = U_k = \min_{j \in N(i) \cap K_{k-1}} \left\{ u_j + \frac{s_i}{w_{j,i}} \right\},$$

that is

$$\max_{j \in N(i) \cap K_{k-1}} \left\{ \frac{w_{j,i}(u_i - u_j) - s_i}{w_{j,i}} \right\} = 0.$$

Since  $w_{j,i} > 0$  for all edges  $(j, i) \in E$ , the model is equivalent to

$$\max_{j \in N(i) \cap K_{k-1}} \{w_{j,i}(u_i - u_j)\} - s_i = 0.$$

From minimality of  $u_i \in C_{k-1}$ , we have  $u_j \geq u_i$  for all  $j \in V \setminus K_{k-1}$ . Recall  $w_{j,i} > 0$  and  $s_i > 0$ , then extending the set over which the maximum is taken from  $N(i) \cap K_{k-1}$  to all of  $N(i)$  does not affect the maximum value. Similarly,  $K_{k-1}$  necessarily contains at least one point  $j$  with  $u_j < u_i$ , therefore replacing  $(u_i - u_j)$  with  $(u_i - u_j)^+$  does not affect the maximum. This leaves

$$\max_{j \in N(i)} \{w_{j,i}(u_i - u_j)^+\} = s_i,$$

which is precisely (2.16), i.e. model 3( $p = \infty$ ). The counter direction runs exactly the same, with the exception that one must show that  $u_i$  is minimal over  $C_{k-1}$ , however, this follows by monotonicity of the construction, as any  $j$  with  $u_j < u_i$  must belong to  $K_{k-1}$  and cannot be in  $C_{k-1}$ .

### 3.1.2. Equivalence of models 1(ii) and 3( $p = 2$ )

Let  $i \in \mathring{V}$ , that is, there exists  $k \in \{1, \dots, L\}$  such that  $i \in V_k$ . First, we show that model 3( $p = 2$ ) in (2.15) follows from model 1(ii) in (2.3). For this  $k$ , the definition of sets  $V_k$ ,  $K_{k-1}$  and  $C_{k-1}$ , the definition  $z_i = \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2$ , and (2.3) implies that  $u_i$  satisfies

$$\sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 u_i = \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 u_j + \sqrt{\left( \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 u_j \right)^2 - z_i \left( \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 u_j^2 - s_i^2 \right)}.$$

For this, we square both sides of the equality which yields

$$(z_i u_i)^2 - 2u_i z_i \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 u_j = z_i s_i^2 - z_i \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 u_j^2.$$

Since  $z_i > 0$ , we obtain

$$\sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 (u_i - u_j)^2 = s_i^2, \quad (3.1)$$

From the definition of  $K_{k-1}$ , the sum can be expanded to the entire neighbourhood  $N(i)$ , by introducing the maximum with zero,

$$\sum_{j \in N(i)} w_{j,i}^2 ((u_i - u_j)^+)^2 = s_i^2, \quad (3.2)$$

This is equivalent to model 3( $p = 2$ ) in (2.15)

Next, we start from model 3( $p = 2$ ) in (2.15) for  $p = 2$ , or equivalently (3.1), and show that model 1(ii) in (2.3) follows. Note that (3.1) can be regarded as a quadratic equation in  $u_i$  whose solution  $u_i$  satisfies

$$u_i = \frac{1}{z_i} \left( \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 u_j \pm \sqrt{\left( \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 u_j \right)^2 - z_i \left( \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 u_j^2 - s_i^2 \right)} \right).$$

The discriminant is nonnegative due to the existence of a unique real solution to (2.15). Since

$$\frac{1}{z_i} \sum_{j \in N(i) \cap K_{k-1}} w_{j,i}^2 u_j \leq \max_{j \in N(i) \cap K_{k-1}} u_j \leq u_i,$$

this implies that the smaller solution contradicts the definition of  $i \in V_k$  and the larger solution of the quadratic equation has to be considered, i.e.

$$u_i = \frac{1}{w_i} \left( \sum_{j \in \tilde{N}(i)} w_{j,i}^2 u_j + \sqrt{\left( \sum_{j \in \tilde{N}(i)} w_{j,i}^2 u_j \right)^2 - w_i \left( \sum_{j \in \tilde{N}(i)} w_{j,i}^2 u_j^2 - s_i^2 \right)} \right),$$

which yields (2.3), that is model 1(ii).  $u_i$  is minimal over  $C_{k-1}$  by construction, as any  $j$  with  $u_j < u_i$  must belong to  $K_{k-1}$  and thus cannot be in  $C_{k-1}$ .

### 3.1.3. Equivalence of models 1(iii) and 3( $p = 1$ )

Let  $i \in \mathring{V}$  be given. Hence, there exists  $k \in \{1, \dots, L\}$  such that  $i \in V_k$ . For this  $k$ , the definition of sets  $V_k$ ,  $K_{k-1}$  and  $C_{k-1}$ , and Model 1(iii) in (2.4) show that  $u_i$  satisfies

$$u_i = \frac{1}{y_i} \left( s_i + \sum_{j \in N(i) \cap K_{k-1}} w_{j,i} u_j \right),$$

which is equivalent to model 3( $p = 1$ ) in (2.15) by the definition of  $y_i$  and the properties of  $i \in V_k$ , i.e.  $\sum_{j \in N(i)} w_{j,i} (u_i - u_j)^+ = s_i$ .

### 3.1.4. Derivation of model of type 1 from model 3( $p$ ) for general $p$

We have proved in the previous subsections that there exists a model of type 1 for any model 3( $p$ ), for  $p \in \{1, 2, \infty\}$ . In this subsection, we provide a procedure for deriving such a model of type 1.

For any finite  $p \geq 1$ , the solution  $u$  of 3( $p$ ) satisfies (2.15). Starting from the boundary condition  $\partial V$ , we initialise the front propagation algorithm. At the  $k$ th iteration, the following steps are done:

1. From  $K_{k-1}$  and the graph neighbourhood structure, create  $C_{k-1}$ .
2. By construction of solutions to Model 1, any admissible solution  $\tilde{u}_i$  has to satisfy  $\tilde{u}_i > u_j$  for all  $j \in N(i) \cap K_{k-1}$  and  $\tilde{u}_i \leq u_j$  for all  $j \in V \setminus K_{k-1}$ . To compute the traveltimes  $\tilde{u}_i$  at candidates  $i \in C_{k-1}$ , we use (2.15). Due to the properties of admissible solutions, it is sufficient to restrict the sum in (2.15) to  $N(i) \cap K_{k-1}$  instead of  $N(i)$ . Over this domain, the restriction to the positive part  $(\cdot)^+$  may be removed, and the problem is reduced to solving a polynomial equation in  $u_i$  (via analytic formulae or numerical solvers). As  $s_i$  and  $w_{j,i}$  are positive, there exists at least one admissible solution  $\tilde{u}_i$ . The uniqueness of  $\tilde{u}_i$  follows from contradiction: suppose that (2.15) has two admissible solutions  $\tilde{u}_i$  and  $\hat{u}_i$  with  $\tilde{u}_i > \hat{u}_i \geq u_j \forall j \in N(i) \cap K_{k-1}$ . Then,

$$s_i^p = \sum_{j \in N(i) \cap K_{k-1}} (w_{j,i} (\tilde{u}_i - u_j))^p > \sum_{j \in N(i) \cap K_{k-1}} (w_{j,i} (\hat{u}_i - u_j))^p = s_i^p$$

This is clearly a contradiction, and thus there is exactly one admissible solution. For all  $i \in C_{k-1}$ , we denote this admissible solution by  $\tilde{u}_i$  and determine  $U_k$  with (2.1).

3. Add all nodes  $i \in C_{k-1}$  with  $U_k = \tilde{u}_i$  into  $V_k$ , then generate  $K_k$ .

## 3.2. Equivalence of first arrival times over path sets and discrete eikonal models.

In this section, we equate the arrival time model (2.5) with travel times (2.6), (2.7), (2.8) (collectively models 2(i), (ii), (iii)) of Section 2.2 with the discrete eikonal models, i.e. model 3( $p = \infty$ ) in (2.16), and models 3( $p = 1$ ), 3( $p = 2$ ) in (2.15).

### 3.2.1. Equivalence between models 2(i) and 3( $p = \infty$ )

Substituting travel time (2.6) of model 2(i) into (2.5) and using the definition of  $K(P_{x_0,i})$  for  $P_{x_0,i} \subset \mathbb{P}_{x_0,i}$  yields

$$\begin{aligned} u_i &= \min_{x_0 \in \partial V} \min_{P_{x_0,i} \subset \mathbb{P}_{x_0,i}} T(P_{x_0,i}) \\ &= \min_{x_0 \in \partial V} \min_{P_{x_0,i} \subset \mathbb{P}_{x_0,i}} \min_{j \in K(P_{x_0,i})} \left( T(P_{x_0,j}^i) + \frac{s_i}{w_{j,i}} \right) \\ &= \min_{x_0 \in \partial V} \min_{K \subset N(i)} \min_{\{P_{x_0,i} \subset \mathbb{P}_{x_0,i} : K(P_{x_0,i}) = K\}} \min_{j \in K} \left( T(P_{x_0,j}^i) + \frac{s_i}{w_{j,i}} \right) \end{aligned}$$

$$\begin{aligned}
 &= \min_{x_0 \in \partial V} \min_{K \subset N(i)} \min_{j \in K} \min_{(P_{x_0,j}^i, (j,i)) \subset \mathbb{P}_{x_0,i}} \left( T(P_{x_0,j}^i) + \frac{s_i}{w_{j,i}} \right) \\
 &= \min_{x_0 \in \partial V} \min_{j \in N(i)} \min_{(P_{x_0,j}^i, (j,i)) \subset \mathbb{P}_{x_0,i}} \left( T(P_{x_0,j}^i) + \frac{s_i}{w_{j,i}} \right)
 \end{aligned}$$

Note that  $P_{x_0,j}^i$  contains paths between  $x_0$  and  $j$  not containing node  $i$ . If we now consider  $P_{x_0,j} \subset \mathbb{P}_{x_0,j}$ , then there may be a path from  $x_0$  to  $j$  via  $i$  in  $P_{x_0,j}$ , but it is not a minimiser. To see that a path  $p_{x_0,j}$  with  $i \in p_{x_0,j}$  is indeed not a minimiser, we consider  $p_{x_0,j} = (i_1 = x_0, \dots, i_k = i, \dots, i_M = j)$  for some  $M \in \mathbb{N}$  and  $1 < k < M$ , implying that  $i_{k-1} \in N(i)$  and hence  $p_{x_0,i} = (i_1 = x_0, \dots, i_{k-1}, i_k = i) \in (P_{x_0,\tilde{j}}^i, (\tilde{j}, i)) \subset \mathbb{P}_{x_0,i}$  for  $\tilde{j} = i_{k-1} \in N(i)$  and some path set  $P_{x_0,\tilde{j}}^i \subset \mathbb{P}_{x_0,\tilde{j}}$ . As the travel time is nonnegative on every edge by (2.10), the travel time is monotone over increasing path length, and we have  $T(p_{x_0,i_{k-1}}) < T(p_{x_0,j})$  with  $i_{k-1}, j \in N(i)$ , implying that  $p_{x_0,j}$  with  $i \in p_{x_0,j}$  cannot be a minimiser. Hence, we write

$$\begin{aligned}
 u_i &= \min_{x_0 \in \partial V} \min_{j \in N(i)} \min_{P_{x_0,j} \subset \mathbb{P}_{x_0,j}} \left( T(P_{x_0,j}) + \frac{s_i}{w_{j,i}} \right) \\
 &= \min_{j \in N(i)} \left( \left( \min_{x_0 \in \partial V} \min_{P_{x_0,j}^i \subset \mathbb{P}_{x_0,j}} T(P_{x_0,j}) \right) + \frac{s_i}{w_{j,i}} \right) = \min_{j \in N(i)} \left( u_j + \frac{s_i}{w_{j,i}} \right).
 \end{aligned}$$

We move  $u_i$  to the right-hand side and use that  $\min(x) = -\max(-x)$ , so that we obtain

$$0 = \max_{j \in N(i)} \left( u_i - u_j - \frac{s_i}{w_{j,i}} \right) = \max_{j \in N(i)} \left( \frac{w_{j,i}(u_i - u_j) - s_i}{w_{j,i}} \right).$$

Due to the positivity of  $w_{ij}$ , this is equivalent to  $\max_{j \in N(i)} (w_{j,i}(u_i - u_j) - s_i) = 0$ , and as  $u_i \geq u_j$ , this yields  $\max_{j \in N(i)} (w_{j,i}(u_i - u_j)^+) = s_i$ , that is, we obtain model 3( $p = \infty$ ) in (2.16).

### 3.2.2. Equivalence between models 2(ii) and 3( $p = 2$ )

Starting with (2.5) and considering travel time of model 2(ii) in (2.7) yields

$$\begin{aligned}
 u_i &= \min_{x_0 \in \partial V} \min_{P_{x_0,i} \subset \mathbb{P}_{x_0,i}} T(P_{x_0,i}) \\
 &= \min_{x_0 \in \partial V} \min_{P_{x_0,i} \subset \mathbb{P}_{x_0,i}} \left( \frac{1}{z_{x_0,i}} \sum_{j \in K(P_{x_0,i})} w_{j,i}^2 T(P_{x_0,j}^i) \right. \\
 &\quad \left. + \frac{1}{z_{x_0,i}} \sqrt{\left( \sum_{j \in K(P_{x_0,i})} w_{j,i}^2 T(P_{x_0,j}^i) \right)^2 + z_{x_0,i} s_i^2 - z_{x_0,i} \sum_{j \in K(P_{x_0,i})} w_{j,i}^2 (T(P_{x_0,j}^i))^2} \right)
 \end{aligned}$$

where  $z_{x_0,i} = \sum_{j \in K(P_{x_0,i})} w_{j,i}^2$ . We can write  $u_i$  as

$$\begin{aligned}
 u_i &= \min_{x_0 \in \partial V} \min_{K \subset N(i)} \min_{\{P_{x_0,i} \subset \mathbb{P}_{x_0,i} : K(P_{x_0,i}) = K\}} \left( \frac{1}{z_K} \sum_{j \in K} w_{j,i}^2 T(P_{x_0,j}^i) \right. \\
 &\quad \left. + \frac{1}{z_K} \sqrt{\left( \sum_{j \in K} w_{j,i}^2 T(P_{x_0,j}^i) \right)^2 + z_K s_i^2 - z_K \sum_{j \in K} w_{j,i}^2 (T(P_{x_0,j}^i))^2} \right),
 \end{aligned}$$

where  $z_K = \sum_{j \in K} w_{j,i}^2$ . Since  $T(P_{x_0,j}^i)$  is the only term depending on  $x_0 \in \partial V$  and  $P_{x_0,j}^i$  satisfying  $P_{x_0,i} = (P_{x_0,j}^i, (j, i)) \subset \mathbb{P}_{x_0,i}$  with  $j \in K(P_{x_0,i})$ , we may pull the minimisation with respect to these parameters inside

the expression and replace the minimisation with respect to  $P_{x_0,i} = (P_{x_0,j}^i, (j, i)) \subset \mathbb{P}_{x_0,i}$  with  $j \in K(P_{x_0,i})$  by  $P_{x_0,j} \subset \mathbb{P}_{x_0,j}$  as in Section 3.2.1. This yields

$$u_i = \min_{K \subset N(i)} \left( \frac{1}{z_K} \sum_{j \in K} w_{j,i}^2 u_j + \frac{1}{z_K} \sqrt{\left( \sum_{j \in K} w_{j,i}^2 u_j \right)^2 + z_K s_i^2 - z_K \sum_{j \in K} w_{j,i}^2 u_j^2} \right)$$

where  $u_j = \min_{x_0 \in \partial V} \min_{P_{x_0,j} \subset \mathbb{P}_{x_0,j}} T(P_{x_0,j})$  by definition. Moving  $u_i$  to the right-hand side and using  $\min(x) = -\max(-x)$  provides

$$0 = \max_{K \subset N(i)} \left( \frac{1}{z_K} \sum_{j \in K} w_{j,i}^2 (u_i - u_j) - \frac{1}{z_K} \sqrt{\left( \sum_{j \in K} w_{j,i}^2 u_j \right)^2 + z_K s_i^2 - z_K \sum_{j \in K} w_{j,i}^2 u_j^2} \right).$$

To achieve that the expression vanishes, we require that the first term is nonnegative which is equivalent to  $K \subset N(i)$  such that  $u_j \leq u_i$  for all  $j \in K$ . Note that the first term is maximal for the set  $\{j \in N(i) : u_j \leq u_i\}$  and the magnitude of the second term decreases as the size of the set  $K$  increases. Hence, the maximiser  $K$  with  $K = \{j \in N(i) : u_j \leq u_i\}$  satisfies

$$z_K u_i - \sum_{j \in K} w_{j,i}^2 u_j = \sqrt{\left( \sum_{j \in K} w_{j,i}^2 u_j \right)^2 + z_K s_i^2 - z_K \sum_{j \in K} w_{j,i}^2 u_j^2}.$$

Squaring both sides and dividing by  $z_K$  yields

$$z_K u_i^2 - 2u_i \sum_{j \in K} w_{j,i}^2 u_j = s_i^2 - \sum_{j \in K} w_{j,i}^2 u_j^2,$$

i.e.

$$s_i^2 = \sum_{j \in K} w_{j,i}^2 (u_i - u_j)^2 = \sum_{j \in N(i) : u_j \leq u_i} w_{j,i}^2 (u_i - u_j)^2 = \sum_{j \in N(i)} w_{j,i}^2 ((u_i - u_j)^+)^2,$$

that is model 3( $p=2$ ) in (2.15).

### 3.2.3. Equivalence between models 2(iii) and 3( $p=1$ )

We begin by using the first arrival model (2.5) with travel time  $T$  given as in model 2(iii) by (2.8) which yields

$$\begin{aligned} u_i &= \min_{x_0 \in \partial V} \min_{P_{x_0,i} \subset \mathbb{P}_{x_0,i}} T(P_{x_0,i}) \\ &= \min_{x_0 \in \partial V} \min_{P_{x_0,i} \subset \mathbb{P}_{x_0,i}} \frac{1}{\sum_{j \in K(P_{x_0,i})} w_{j,i}} \left( \sum_{j \in K(P_{x_0,i})} w_{j,i} T(P_{x_0,j}^i) + s_i \right) \\ &= \min_{K \subset N(i)} \min_{x_0 \in \partial V} \min_{\{P_{x_0,i} \subset \mathbb{P}_{x_0,i} : K(P_{x_0,i})=K\}} \frac{1}{\sum_{j \in K} w_{j,i}} \left( \sum_{j \in K} w_{j,i} T(P_{x_0,j}^i) + s_i \right) \\ &= \min_{K \subset N(i)} \frac{1}{\sum_{j \in K} w_{j,i}} \left( \sum_{j \in K} w_{j,i} \min_{x_0 \in \partial V} \min_{P_{x_0,j} \subset \mathbb{P}_{x_0,j}} T(P_{x_0,j}) + s_i \right) \\ &= \min_{K \subset N(i)} \frac{1}{\sum_{j \in K} w_{j,i}} \left( \sum_{j \in K} w_{j,i} u_j + s_i \right), \end{aligned}$$

where we can use a similar argument as in Section 3.2.1 in the fourth equality to consider the sets  $P_{x_0,j} \subset \mathbb{P}_{x_0,j}$  instead of the sets  $P_{x_0,i} = (P_{x_0,j}^i, (j, i)) \subset \mathbb{P}_{x_0,i}$  with  $j \in K(P_{x_0,i})$ . Then we rearrange the equation resulting in



$$\min_{K \subset N(i)} \frac{1}{\sum_{j \in K} w_{j,i}} \left( \sum_{j \in K} w_{j,i} (u_j - u_i) + s_i \right) = 0,$$

and as  $\sum_{j \in K} w_{j,i} > 0$ , we obtain

$$s_i = - \min_{K \subset N(i)} \left( \sum_{j \in K} w_{j,i} (u_j - u_i) \right) = \max_{K \subset N(i)} \sum_{j \in K} w_{j,i} (u_i - u_j).$$

If  $u_j \leq u_i$ , then the summand is positive and therefore the maximiser over  $K \subset N(i)$  is the set  $\{j \in N(i) : u_j \leq u_i\}$ . Hence, we arrive at

$$s_i = \sum_{j \in N(i) : u_j \leq u_i} w_{j,i} (u_i - u_j) = \sum_{j \in N(i)} w_{j,i} (u_i - u_j)^+,$$

that is, model 3( $p = 1$ ) in (2.15).

## 4. Applications

In this section, we use information propagation in two applications: trust networks and semi-supervised learning.

### 4.1. Applications to trust networks

In this section, we apply information propagation to a trust network, which is a weighted directed graph, with nodes being users of a social network. The edges and their trust weights are reviews of trust among users, for example,  $\omega_{i,j}$  implies that  $i$  trusts  $j$  with a rating  $\omega_{i,j}$ . The neighbourhood structure is therefore built around *trusting nodes* and their neighbours of *trusted nodes*. It is a directed relationship, as trust may not be reciprocated (i.e.  $(i, j)$  may exist, but  $(j, i)$  does not), and may not be symmetric ( $\omega_{i,j} \neq \omega_{j,i}$ ).

An example of a trust network is the advocato dataset [30], specifically soc-advocato obtained from [35]. The dataset is a snapshot in time of a social network comprised of around 5000 software developers, with four categorical weightings of trust based on users' perceived contributions to open software and programming skills. These weightings have been numerically equated, though somewhat arbitrarily [30], to four values  $\omega_{i,j} \in (0.4, 0.6, 0.8, 1.0)$  where larger values correspond to larger levels of trust. Structural information of the graph is found in [35]. For this experiment, we will investigate only the largest connected component of this graph, which contains 5167 nodes (of which 4017 are trusting, and 4428 are trusted) and 47,337 edges.

Consider the application of a software team searching for a new collaborator from the network. The team seeks a notion of trustworthiness for each candidate collaborator. One can assess the level of trust in each candidate by the team by using the information from the trust network. A seemingly simple way to do this is to aggregate the trust given to them directly by other users (i.e. the weights from users to the candidate). Unfortunately, a common subversion of this method is a Sybil attack [1, 15, 44]. In its simplest form, a candidate creates a network of artificial community members, colloquially called 'Sybils', who have high trust with each other and with the candidate. This will boost the candidate's aggregate trust. Instead of a neighbourhood-only measurement of trust, we propose using information propagation of distrust to provide a ranking of candidates from the perspective of the team in a way that is resistant to Sybil attack.

We perform the propagation of distrust by setting boundary nodes  $\partial V$  as the software team. We then set  $s = 1$  at all nodes. We define the distrust weights  $w_{j,i} = \frac{1}{\omega_{j,i}}$ , a reciprocal of the trust weights. Candidates are selected on the network, and we calculate the (first arrival) times for the information to propagate to all candidates from the team members by using a propagation model. The candidates

**Table 2.** Ranking of trust of candidates A-H, for two experiments: a control experiment (Ctrl) and an experiment with a cluster of 50 Sybils around candidate G (GSyb). Candidates A-H are alphabetically assigned by the order of the first column. The columns give trust rankings from different information propagation models ( $p = 1, 2, \infty$ ), or from using the average of neighbourhood distrust (neighbour). The measure of absolute distrust of the candidate is given in brackets: for the first three columns, this is the travel time, in the final two columns, this is the averaged distrust over the neighbourhood of the candidate

Trust Rank (1 = most trust)	$p = 1$ Ctrl & GSyb	$p = 2$ Ctrl & GSyb	$p = \infty$ Ctrl & GSyb	Neighbour Ctrl	Neighbour GSyb
1	A (1.170)	A (1.728)	B (2.0)	A, D, H (1.0)	A, D, H (1.0)
2	B (1.360)	B (2.001)	A, C, E (2.2)	A, D, H (1.0)	A, D, H (1.0)
3	C (1.371)	C (2.019)	A, C, E (2.2)	A, D, H (1.0)	A, D, H (1.0)
4	D (1.573)	E (2.084)	A, C, E (2.2)	B (1.167)	<b>G (1.029)</b>
5	E (1.784)	D (2.341)	F, G (2.4)	C, F (1.25)	B (1.167)
6	F (2.005)	F (2.354)	F, G (2.4)	C, F (1.25)	C, F (1.25)
7	G (2.047)	G (2.358)	H (2.6)	<b>G (1.333)</b>	C, F (1.25)
8	H (2.148)	H (2.574)	D (2.8)	E (1.458)	E (1.458)



**Figure 4.** Result of the distrust propagation from a four-member software team, to eight candidates. Edge arrows indicate direction of trust. The left panel shows the software team (magenta) and candidates (cyan). The right panel shows the solved travel time field using model  $p = 1$ , with node colour indicating the level of distrust of this community member by the software team.

with greater arrival time are less trustworthy according to the model. This method of measuring distrust accounts for both the degree of separation between the team and the candidates, as well as the trust of each review along such paths. It is resistant to Sybil attacks, as 'Sybils' form a (largely) disconnected cluster around a candidate, and so have little or no effect on path structures between the team and the candidate.

The experiment configuration is shown overlaying a relevant portion of the network in the left panel of Figure 4, we randomly select both a four-member team in magenta, and eight candidates in cyan that we label A-H. For illustration, the travel times (level of distrust) of the displayed nodes according to the propagation model with  $p = 1$  are given in the right panel of Figure 4. We perform two control experiments. In the first control experiment (Ctrl), we directly use the network of soc-advocato. In a second experiment (GSyb), we modify the network by adding a fully connected Sybil cluster (of size 50) to candidate G. The members of this cluster are given by the highest trust weighting 1 from each

other and the candidate  $G$ , and vice versa. For each experiment, we use different information propagation models and the neighbour-averaging approach to calculate a trust ranking of the eight candidates. The results of the experiments are given in Table 2. Candidates  $A$ – $H$  are alphabetically assigned by the order of the first column. The first three columns show ranks given by propagation model for  $p = 1, 2, \infty$ , which is based on globally averaging distrust. The final two columns show ranks for candidate  $j$  based on locally averaging distrust  $w_{ij}$  over  $i$  such that  $j \in N(i)$ .

First, we look at experiment (Ctrl) in Table 2. Comparing the propagation-based and neighbour-based ranks, we offer an interpretation of some interesting candidates. As candidate  $A$  is deemed trustworthy across all methods, this implies that both locally and globally  $A$  is a trustworthy candidate. Candidate  $H$ , on the other hand, is deemed trustworthy locally but suffers globally which indicates that there is an overall untrustworthy pathway of reviewers between the team and  $H$ . Candidate  $D$  shows a difference in ranking between travel time models  $p = 1, 2$  and  $p = \infty$ , which implies that although the most trustworthy review path from the team to  $D$  is not very trustworthy ( $p = \infty$  ranks  $D$  eighth), there are many similarly trusted pathways from the team to  $D$  ( $p = 1, 2$  rank  $D$  fourth and fifth). In this way, the models for  $p < \infty$  encode a concept of confidence over the network uncertainty into their travel time. In general, we see that for  $p = 1$  and  $p = 2$ , candidates enjoy similar rankings, while the  $p = \infty$  model tends to group candidates together, as for this model the travel times can only take more restrictive discrete values. These preliminary results suggest that  $p < \infty$  approaches provide solutions richer in information from the network and may be more robust in discrete settings, and so we advocate further investigation of their use in graph-based algorithms where  $p = \infty$  may be the state of art.

Finally, the key result of comparing between the experiments (Ctrl) and (GSyb) is that the travel time-based ranking did not change between the experiments, whereas the neighbour-averaged distrust of candidate  $G$  reduced from 1.333 in (Ctrl) to 1.029 in (GSyb), thus increasing their rank from 7 to 4. This provides concrete evidence to the susceptibility of neighbourhood-based approaches, while arrival time approaches are completely resilient to this form of attack.

## 4.2. Applications to label propagation/semi-supervised learning

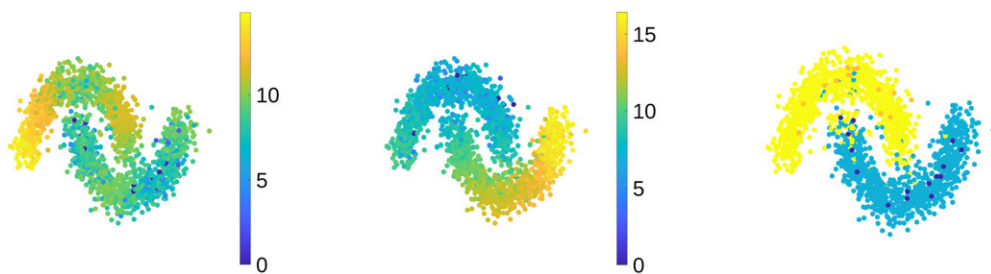
In this section, we consider an application to a semi-supervised learning approach to label propagation. The model consists of attaching  $L > 1$  labels to  $n > 1$  sets of features  $f_j \in \mathcal{F}_j$ ,  $j = 1, 2, \dots, n$ , where  $\mathcal{F}_j = \{\mathcal{F}_i^j \in \mathbb{F}_i\}_{i=1}^m$ ,  $i = 1, 2, \dots, m$  and  $\mathbb{F}_i$  is either  $\mathbb{R}$  or  $\mathbb{B}$ ,  $\mathbb{B} = \{0, 1\}$ . The first step consists of assigning weights  $w_{ij} \geq 0$  whose reciprocal measures the distance between features  $f_i$  and  $f_j$ . If the distance between features is sufficiently large according to some criterion, then the weight is set to zero. From this we obtain the graph with vertices  $V = \{1, 2, \dots, n\}$  and edges  $E \in V^2$  consisting of the pairs satisfying  $w_{ij} > 0$ . We assume there is a set of nodes  $\partial V_\ell$  for each category  $\ell = 1, \dots, L$  so that  $\text{label}(i) = \ell$  for all  $i \in \partial V_\ell$ , that is, a set where the classification is known. Our semi-supervised learning task is to provide all unlabelled nodes in  $V \setminus \{\cup_\ell \partial V_\ell\}$  with a label. The front propagation semi-supervised learning model is to assign

$$\text{label}(i) = \left\{ \ell \mid u_i^{(\ell)} = \min_{k=1, \dots, L} u_i^{(k)} \right\} \quad (4.1)$$

for any  $i \in V \setminus \{\cup_\ell \partial V_\ell\}$ , where  $u_i^{(k)}$  is the solution to a discrete eikonal equation (2.14) on a weighted graph for some  $p \in [1, +\infty]$ , with boundary  $u_i = 0$  for  $i \in \partial V_k$ . We assume for this model that the slowness function  $s \equiv 1$ . In this way,  $i$  is assigned the label  $\ell$  if the smallest travel time is found between  $\partial V_\ell$  and  $i$  among all sets of labels. This model requires to solve the discrete eikonal problem per label category, which can be performed independently in parallel to each other. For each of the following experiments, we carry out 20 simulations with differing random choices of known initial labels. We present the average (and standard deviation) of the classification accuracy over these 20 simulations. The labelling accuracy is calculated as the percentage of unlabelled nodes that are correctly classified.

**Table 3.** Mean (standard deviation) of classification for the two moons example

$w_{i,j}$	Eikonal model	Two moons accuracy %
$\exp\left(-\frac{\ x_i - x_j\ ^2}{\sqrt{d_{10}(x_i)d_{10}(x_j)}}\right)$	$p = 1$	92.7 (3.81)
	$p = 2$	92.0 (2.80)
	$p = \infty$	89.5 (2.96)



**Figure 5.** Example travel time fields and classification for two moons problem projected into two dimensions. The left and centre panels show the travel time field for labels 1 and 2, respectively. The right panel shows the resulting classification with predicted label 1 (blue) and predicted label 2 (yellow) solved with initially known labels 1 (orange), and 2 (dark blue). In this example, the accuracy was 94.7%.

#### 4.2.1. High-dimensional two moons problem

We follow the construction of the two moons problem for classification as in [5, 7]. The feature vectors here are taken to be the spatial coordinates of  $n$  nodes in  $\mathbb{R}^m$ , i.e.  $\mathbb{F}_i = \mathbb{R}$ ,  $\forall i$ . The construction is formed by considering two planar half circles of radius 1. One is centred at the origin and the other is rotated by  $\pi$  and centred (1, 0.5). We take  $n = 2000$  points on these initial planar circles and then embed them in  $\mathbb{R}^{100}$  by adding uniform Gaussian noise  $N(0, 0.02I_{100})$  where  $I_{100}$  is the identity matrix in  $\mathbb{R}^{100}$ . We define a classification problem by giving points on each initial circle a different binary label; for visualisation we project back onto the plane as seen in Figure 5. We proceed again as in [5, 7] by calculating distances between pairs of points in  $\mathbb{R}^{100}$  and then setting all weights  $w_{i,j} = 0$  unless point  $j$  is within the 10 nearest neighbours of point  $i$ . The non-zero weights are then set according to the weight function of [43]; a squared exponential function of distance, weighted by a local scaling  $d_{10}(x_i) = \|x_i - x_{j(i,10)}\|$ , where  $j(i, 10)$  is the 10<sup>th</sup> nearest neighbour of  $i$  (see Table 3). We perform each of the experiments by choosing at random 15 nodes per moon to have known labels. The illustration of the travel time-based classification is given in Figure 5. The accuracy results are given in Table 3. Here we observe high accuracy, with all choices of eikonal model comparable to experiments of unsupervised clustering in [5] with near optimal parameter choices. Our method has no tuneable parameters, though the experiment suggests best performance for  $p = 1$ .

#### 4.2.2. Text classification dataset

We demonstrate the performance on the standard Cora and CiteSeer document classification datasets [36]. In both cases, the graph nodes correspond to journal articles, and links between them are obtained from citations, forming a directed graph. The feature vectors are binary valued of length 1433 (Cora, i.e.  $\mathbb{F}_i = \mathbb{B}$ ,  $\forall i$ ) and 3703 (CiteSeer  $\mathbb{F}_i = \mathbb{B}$ ,  $\forall i$ ), based on whether or not the article contained specific words from a unique dictionary. Following [28, 42], we symmetrise the adjacency matrix for each citation link. We benchmark with the resulting largest connected component of each dataset. The resulting graphs have 2485 nodes and 5069 edges (Cora) and 2110 nodes and 3694 edges (CiteSeer). The reference

**Table 4.** Mean (standard deviation) of classification accuracy given as percentages, for the examples using different choices of weights. The function  $d_{\max}(x)$  is the Euclidean distance from  $x_i$  to its furthest neighbour

$w_{i,j}$	Eikonal model	Cora accuracy	CiteSeer accuracy
$1/\ x_i - x_j\ _{\ell^2}$	$p = 1$	69.0 (7.49)	64.3 (1.64)
	$p = 2$	68.9 (6.86)	62.6 (1.87)
	$p = \infty$	68.1 (3.86)	61.0 (2.26)
$\exp\left(-\frac{\ x_i - x_j\ _{\ell^2}^2}{500}\right)$	$p = 1$	72.4 (1.58)	64.3 (1.91)
	$p = 2$	71.8 (1.88)	62.5 (2.12)
	$p = \infty$	69.2 (2.50)	60.8 (2.25)
$\exp\left(-\frac{\ x_i - x_j\ _{\ell^2}^2}{100\sqrt{d_{\max}(x_i)d_{\max}(x_j)}}\right)$	$p = 1$	72.4 (1.56)	64.3 (2.06)
	$p = 2$	71.7 (1.91)	62.5 (2.08)
	$p = \infty$	69.0 (2.42)	60.7 (2.22)

did not provide suggestion for the graph weights, thus some naive choices were taken, based on the  $\ell^2$ -norm over binary vectors (see Table 4). There are  $L = 7$  (Cora) and  $L = 6$  (CiteSeer) labels, respectively, for each dataset, representing journal categories that we wish to classify. We take 20 labels from each category. The classification accuracy experiments for the different datasets were applied to 20 random seeds, and we display the results of the eikonal models for  $p \in \{1, 2, \infty\}$ . We assume for this application that the slowness function  $s \equiv 1$ . The results are shown in Table 4. Performance is robust across seeds and eikonal models chosen. The experiment suggests best performance at  $p = 1$ . The exponential-based weighted graphs outperform the reciprocal distance-based weights and have less variation due to random seeding. For this graph,  $d_{\max}$  was relatively constant and did not aid performance. We did not optimise the constants appearing in the weight functions and the algorithms performed similarly across an order of magnitude. Several approaches have applied to these datasets in [42]. Here comparisons are qualitative, as different methods (e.g., [3, 25, 41]) use differing levels of information. On these datasets, the front propagation approach performs comparably to Planetoid-T and Planetoid-I, the flagship methods of [42].

## 5. Conclusion

In this paper, we proposed some models for information propagation on graphs. Underlying components of the models include a subset of nodes forming the initial source of information, the arrival times of information and the laws governing the transfer of information to nodes from their neighbours. The models are collected into three viewpoints: an information wavefront hitting time, an optimal travel time over sets of paths and a local equation for the time to receive information at a node given the times to receive information at its neighbours. We showed equivalences between these different views, as summarised in Table 1. In this framework, we provide examples such as a generalisation of classical equivalence between optimal paths and Dijkstra's algorithm [14]. We applied our models to a social network dataset soc-advogato [35], where directed edges are weighted by trust. Propagation of wavefronts from a group of nodes over such weighted networks define a notion of (dis)trust of this group on all other nodes defined by the travel times to other nodes. This notion of trust is robust to local Sybil attacks [15]. More generally, our models could be used as a back-end to replace path length or distance calculations in other cybersecurity strategies [1], as qualitatively the  $p < \infty$  approaches displayed better solution properties than  $p = \infty$ . Extending the work of [40], we applied these models to label propagation in a semi-supervised learning application. The eikonal-based classification algorithm obtains

comparable performance to clustering algorithms with two labels (e.g. [5]), and with simple choices of weight functions, it achieves comparable performance to machine-learning methods that learn graph embeddings (e.g., [42]) without any tuning or training. While graph Laplacian methods are often used to model information propagation on networks (e.g., [18, 20, 33]), the eikonal approach can also be applied and encapsulates control problems (using  $s$  or  $w$  as controllers). Procedures based on front propagation offer adjoint equations at no additional cost, leading to very efficient methods for inverse problems in these settings [11, 16].

**Financial support.** ORAD would like to acknowledge the support of Schmidt Sciences, LLC, the National Science Foundation (Grant No. AGS-1835860), the Cisco Foundation and the Office of Naval Research (Grant No. N00014-23-1-2654). LMK acknowledges support from the Warwick Research Development Fund through the project ‘Using Partial Differential Equations Techniques to Analyse Data-Rich Phenomena’, the European Union Horizon 2020 research and innovation programmes under the Marie Skłodowska-Curie grant agreement No. 777826 (NoMADS) and the Cantab Capital Institute for the Mathematics of Information and Magdalene College, Cambridge (Nevile Research Fellowship).

**Competing interests.** There are no competing interests to report.

## References

- [1] Al-Qurishi, M., Al-Rakhami, M., Alamri, A., Alrubaian, M., Rahman, S. M. M. & Hossain, M. S. (2017) Sybil defense techniques in online social networks: A survey. *IEEE Access* **5**, 1200–1219.
- [2] Belkin, M., Matveeva, I. & Niyogi, P. (2004). Regularization and semi-supervised learning on large graphs. In: *International Conference on Computational Learning Theory*, Springer. 624–638.
- [3] Belkin, M., Niyogi, P. & Sindhvani, V. (2006) Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.* **7**(85), 2399–2434.
- [4] Bertozzi, A. L. & Flenner, A. (2012) Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Model. Sim.* **10**(3), 1090–1118.
- [5] Bertozzi, A. L. & Flenner, A. (2016) Diffuse interface models on graphs for classification of high dimensional data. *SIAM Rev.* **58**(2), 293–328.
- [6] Blum, A. & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In: *Proceedings of the Eighteenth International Conference on Machine Learning, ICML, 2001*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. pp. 19–26.
- [7] Bühler, T. & Hein, M. (2009). Spectral clustering based on the graph p-laplacian. In: *Proceedings of the 26th annual international conference on machine learning*. pp. 81–88.
- [8] Calder, J. & Ettehad, M. (jan 2022) Hamilton-Jacobi equations on graphs with applications to semi-supervised learning and data depth. *J. Mach. Learn. Res.* **23**(1), 1–62.
- [9] Camilli, F., Festa, A. & Schieborn, D. (2013) An approximation scheme for a Hamilton-Jacobi equation defined on a network. *Appl. Numer. Math.* **73**, 33–47.
- [10] Camilli, F. & Marchi, C. (2013) A comparison among various notions of viscosity solution for Hamilton-Jacobi equations on networks. *J. Math. Anal. Appl.* **407**(1), 112–118.
- [11] Deckelnick, K., Elliott, C. M. & Styles, V. (2011) Numerical analysis of an inverse problem for the Eikonal equation. *Numerische Mathematik* **119**(2), 245–269.
- [12] Desquesnes, X., Elmoataz, A. & Lézoray, O. (2013) Eikonal equation adaptation on weighted graphs: Fast geometric diffusion process for local and non-local image and data processing. *J. Math. Imaging Vis.* **46**(2), 238–257.
- [13] Desquesnes, X., Elmoataz, A., Lézoray, O. & Ta, V.-T. (2010). Efficient algorithms for image and high dimensional data processing using Eikonal equation on graphs. In: *International Symposium on Visual Computing*. Springer, pp. 647–658.
- [14] Dijkstra, E. W. (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* **1**(1), 269–271.
- [15] Douceur, J. R. (2002). The Sybil attack. In: Druschel, Peter, Kaashoek, Frans & Rowstron, Antony (eds.), *Peer-to-Peer Systems*, Springer, Berlin, Heidelberg, pp. 251–260.
- [16] Dunbar, O. R. A. & Elliott, C. M. (2019) Binary recovery via phase field regularization for first-arrival traveltime tomography. *Inverse Probl.* **35**(9), 095004.
- [17] Dunlop, M. M., Slepčev, D., Stuart, A. M. & Thorpe, M. (2020) Large data and zero noise limits of graph-based semi-supervised learning algorithms. *Appl. Comput. Harmon. Anal.* **49**(2), 655–697.
- [18] Elmoataz, A., Desquesnes, X. & Toutain, M. (2017) On the game p-Laplacian on weighted graphs with applications in image processing and data clustering. *Eur. J. Appl. Math.* **28**(6), 922–948.
- [19] Elmoataz, A., Lezoray, O. & Bougleux, S. (2008) Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. *IEEE T. Image Process* **17**(7), 1047–1060.
- [20] Elmoataz, A., Toutain, M. & Tenbrinck, D., (2015) On the p-Laplacian and  $\infty$ -Laplacian on graphs with applications in image and data processing. *SIAM J. Imaging Sci.* **8**(4), 2412–2451.



- [21] Fadili, J., Forcadel, N., Nguyen, T. T. & Zantout, R. (2023) Limits and consistency of nonlocal and graph approximations to the Eikonal equation. *IMA J. Numer. Anal.* **43**(6), 3685–3728.
- [22] García Trillos, N. & Slepčev, D. (2016) Continuum limit of total variation on point clouds. *Arch. Ration. Mech. Anal.* **220**(1), 193–241.
- [23] Gromov, M. (2007, *Length Structures: Path Metric Spaces*, Boston, MA, Birkhäuser Boston, 1–25.
- [24] Jalili, M. & Perc, M. (2017) Information cascades in complex networks. *J. Complex Net.* **5**(5), 665–693.
- [25] Joachims, T. (1999). Transductive inference for text classification using support vector machines, In *ICML*, Vol. 99, pp. 200–209.
- [26] Kempe, D., Kleinberg, J. & Tardos, É. (2015) Maximizing the spread of influence through a social network. *Theo. Comput.* **11**(4), 105–147.
- [27] Kimmel, R. & Sethian, J. A. (1998) Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences* **95**(15), 8431–8435.
- [28] Kipf, T. N. & Welling, M. (2017) Semi-supervised classification with graph convolutional networks. In: *Proceedings of the 5th International Conference on Learning Representations*. ICLR: Toulon. arXiv preprint arXiv: 1609.02907.
- [29] Kreusser, L. M. & Wolfram, M.-T. (2020) On anisotropic diffusion equations for label propagation. arXiv preprint 2007.12516.
- [30] Massa, P., Salvetti, M. & Tomasoni, D. (2009). Bowling alone and trust decline in social network sites. In: *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, IEEE Computer Society, pp. 658–663.
- [31] Merkurjev, E., Kostic, T. & Bertozzi, A. L. (2013) An MBO scheme on graphs for classification and image processing. *SIAM J. Imaging Sci.* **6**(4), 1903–1930.
- [32] Oberman, Adam M., Sviridov, Alexander P. & Manfredi, Juan J. (2015) Nonlinear elliptic partial differential equations and p-harmonic functions on graphs. *Differ. Integral Equ.* **1**(2), 79–102.
- [33] Olfati-Saber, R., Fax, J. A. & Murray, R. M. (2007) Consensus and cooperation in networked multi-agent systems. *Proceed. IEEE* **95**(1), 215–233.
- [34] Paluch, R., Lu, X., Suchecki, K., Szymański, B. K. & Hołyst, J. A. (2018) Fast and accurate detection of spread source in large complex networks. *Sci. Rep.* **8**(1), 1–10.
- [35] Rossi, R. A. & Ahmed, N. K. (2015). The network data repository with interactive graph analytics and visualization, In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Austin, TX: AAAI Press, pp. 4292–4293.
- [36] Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B. & Eliassi-Rad, T. (2008) Collective classification in network data. *AI Mag.* **29**(3), 93–93.
- [37] Sethian, J. A. (1996) Theory, algorithms, and applications of level set methods for propagating interfaces. *Acta Numer.* **5**, 309–395.
- [38] Sethian, J. A. (1999) Fast marching methods. *SIAM Rev.* **41**(2), 199–235.
- [39] Ta, V.-T., Elmoataz, A. & Lézoray, O. (2009). Adaptation of eikonal equation over weighted graph. In: Tai, Xue-Cheng, Mørken, Knut, Lysaker, Marius & Lie, Knut-Andreas (eds.), *Scale Space and Variational Methods in Computer Vision*, Springer, Berlin Heidelberg, pp. 187–199.
- [40] Toutain, M., Elmoataz, A. & Lézoray, O. (2014). Geometric pdes on weighted graphs for semi-supervised classification. In: *2014 13th International Conference on Machine Learning and Applications*, pp. 231–236.
- [41] Weston, J., Ratle, F., Mobahi, H. & Collobert, R. (2012). Deep learning via semi-supervised embedding, In: *Neural Networks: Tricks of the Trade*, Springer, pp. 639–655.
- [42] Yang, Z., Cohen, W. & Salakhudinov, R. (2016). Revisiting semi-supervised learning with graph embeddings. In: *International conference on machine learning*, PMLR, pp. 40–48.
- [43] Zelnik-Manor, L. & Perona, P. (2004) Self-tuning spectral clustering. *Adv. Neur. Inf. Process. Syst.* **17**, 1601–1608.
- [44] Zhang, K., Liang, X., Lu, R. & Shen, X. (2014) Sybil attacks and their defenses in the internet of things. *IEEE Internet Things J.* **1**(5), 372–383.
- [45] Zhu, X. (2005). Semi-supervised learning literature survey, Technical Report 1530, Computer Sciences, University of Wisconsin–Madison.