Estimating Invasion Time in Real Landscapes[*]

Daniyah A. Aloqalaa
Department of Computer Science
University of Liverpool, UK
d.a.aloqalaa@liverpool.ac.uk

Jenny A. Hodgson
Department of Evolution, Ecology and Behaviour
University of Liverpool, UK
jenny.hodgson@liverpool.ac.uk

Dariusz R. Kowalski[†]
Department of Computer Science
University of Liverpool, UK
d.kowalski@liverpool.ac.uk

Prudence W.H. Wong
Department of Computer Science
University of Liverpool, UK
pwong@liverpool.ac.uk

ABSTRACT

Species are threatened by climate changes, unless their populations have the ability to invade landscapes to search for new regions of suitable climate and conditions. It is therefore of utmost importance for ecologists to estimate the invasion time, as it is a crucial parameter used for environmental planning and may even determine survivability of the species. From a computational perspective, estimating the invasion time by running simulations is very time consuming, as the full model is based on a Markov Chain of exponential number of states with respect to the landscape size; therefore, in practice, this method is not suitable especially in case of frequent environmental changes or for environmental planning. In this paper, we propose a new way to estimate the time of invasion process using a powerful computational approach based on conductance and network flow theory. More specifically, we give a new formula for estimating the invasion time using a combination of network flow methodologies, and prove asymptotic bounds on the quality of the obtained approximation. The proposed approach is analyzed mathematically and applied to real heterogeneous landscapes of the United Kingdom to estimate the duration of the process; the theoretical bounds obtained are compared with simulation results. The evaluations of the proposed approach demonstrate its accuracy and efficiency in approximating the invasion time.

CCS Concepts

•**Applied computing** → Computational biology; •**Theory of Computation** → *Design and analysis of algorithms;* Mathematical optimization; •**Networks** → Network algorithms;

[*]Supported by the initiative Networks Sciences & Technologies (NeST) by School of EEE & CS, University of Liverpool.

[†]Supported by Polish National Science Center (NCN) grant UMO-2017/25/B/ST6/02553.

ICCBB '18 October 11–13, 2018, Bari, Italy

DOI:

Keywords

Invasion process; Landscape; Simulations; Rumor spreading; Network flow; Conductance

1. INTRODUCTION

Species are being extinct faster than natural extinction due to changes in land use, climate change and pollution among others [1, 2, 3, 4]. Ecologists observed that species that are effectively able to respond to these threats do so by shifting/invading their geographical areas [5, 6], however, the availability of suitable habitats may limit the ability of species to shift [1, 3, 7, 8]. If species are shifting very slowly or failing to shift, they become more vulnerable to extinct [9]. Therefore, in order to protect ecosystem functioning and services in different environmental changes, it becomes an important need to improve species performance and species interactions especially by facilitating their shifts to new regions with more suitable climate and landscape environment [10]. Conservationists are facing challenges in finding out whether and how they can the facilitate shifting of species [10]. A number of ecological studies have been shown the importance of spatial arrangement of habitats on the speed of range shifting [4, 10]. Hodgson et al. [10] found the evidence of the benefits of creating new habitats as "corridors" or "stepping stones" to allow species to shift through unsuitable landscapes and to help them colonise new regions. However, this notion of habitat creation is essentially difficult to test in large landscapes and it becomes even more expensive computation problem when different scenarios need to be tested.

In this paper, we propose a new formula for estimating the time of the invasion process using conductance and network flow theory. Network flow [11] is one of the fundamental problems in many areas of computer science, including networks, optimization, distributed systems, and distributed databases. The input of the problem is a directed, weighted landscape graph $G = (V, E)$, where each vertex represents a patch of habitat (henceforth patch) in the landscape and each edge weight represents the probability of spreading the species in one step from the beginning to the end patch of the edge. We distinguish two sets of patches: the source patches represent initially populated patches in which species are located, and the target patches represent the target locations for the invasion process. The network is almost completed, in the sense that each vertex is connected to (almost) all other vertices. The invasion process is to populate any of the target patches when starting from the source patches, and we aim to approximate the time performance of this process. The vertex is populated by means of an invasion protocol. The protocol keeps checking edges, one after another (see definitions of synchronous and asynchronous executions later in this section), by generating a random number from 0 to 1; if the number is smaller than the weight of the edge and the beginning of the edge is populated, then the end of the edge becomes populated as well (unless it has been populated earlier). We restrict our attention to the invasion protocol introduced in [10]. In each round of the invasion process, each populated vertex tries to hit/populate (independently) all other unpopulated vertices in the landscape graph.

Throughout this paper, we consider discrete time steps, and we assume that one time step is sufficiently large so that the flow of the species between two vertices (i.e., population) completes. We analyze the invasion protocol in two models: *asynchronous* and *synchronous*. In the execution of the invasion protocol in the asynchronous model, simply called an asynchronous execution (of invasion protocol), at each step only one edge is selected, uniformly at random, among the m edges of the landscape graph. The selected edge is realized with a defined probability (weight) of populating unpopulated vertex w from populated vertex v, where vertices v and w are connected by the selected directed edge. In a synchronous execution of the invasion protocol, at each *synchronous round* all directed edges in the landscape graph are realized, independently with defined probabilities (weights); in this sense, each (synchronous) round consists of exactly m steps, each done with respect to a different edge. Sometimes, for simplicity, we will be calling asynchronous and synchronous executions of the invasion protocol by asynchronous and synchronous invasions, respectively.

In order to measure the *time performance*, or *runtime*, of the invasion process, we measure the number of steps required so that every vertex in the target set is populated, regardless of which vertices are initially populated (worst case analysis). Since in asynchronous invasion only one edge is selected per step, while in synchronous invasion all edges are realized independently with corresponding probabilities (weights) per step, in order to compare them fairly we consider the concept of a *round*, that is a single synchronous round in case of synchronous executions and m consecutive steps in case of asynchronous executions.

In this work, we define the invasion time as the estimated time (total number of rounds) to populate any of the target patches in a given landscape. We will define an invasion landscape network for a given landscape and find a correspondence between conductance and network flows in this network and the invasion time.

1.1 Our Results

We introduce a new theoretical measure γ that estimates the expected runtime of invasion process for a given landscape. In particular:

- We show that the proposed measure estimates from above the expected number of rounds in asynchronous and synchronous execution, in the latter with additional logarithmic (additive) component (Theorem 2).
- We prove that the new measure is inversely proportional to the conductance of the proposed flow network with a logarithmic factor (Theorem 3).
- Based on our theoretical investigations, we introduce and justify more precise prediction formula IT for invasion time (Equation 2).
- We compare the obtained prediction formula with simulation results on real UK landscapes and obtain quite accurate estimates of invasion runtime in much lower computational cost.

We believe that our theoretical and simulational results form a convincing background for further work in this direction, to obtain even better accuracy in prediction of invasion time for real landscapes.

2. MODEL, NOTATIONS, PREVIOUS MEASURES AND RELATED WORK

We are given a 2-dimensional rectangular grid landscape of height/ rows H and width/columns W as an input, which we represent as a directed *landscape graph* $G = (V, E)$. Let $n = |V|$ be the number of vertices in the landscape graph, $m = |E|$ be the number of directed edges and $d(v)$ be the out-degree of a vertex $v \in V$. Let $q(v)$ denote the quality of patch v, where the quality is a number between zero and one given as input. We distinguish two sets of patches, S and T, where S denotes the set of populated source patches that are non-zeros in quality and T denotes the set of unpopulated target patches that are also non-zeros in quality. For any subset of vertices $X \subseteq V$, $\mathrm{vol}(X)$ denotes the *volume* of X and it is defined as $\sum_{v\in X} \sum_{w\in V,(v,w)\in E} p_{vw}$, where p_{vw} is the weight of edge (v, w). We define the maximum and minimum vertex degree as follows: $d_{max} = \max\{d(v) : v \in V\}$ and $d_{min} = \min\{d(v) : v \in V\}$. For any set of vertices $X \subseteq V$ and any vertex v, we denote by $d_X(v)$ the degree of v restricted to set X of out-neighbors. For any two sets of vertices $X, Y \subseteq V$ such that $X \cap Y = \emptyset$, we define $E(X, Y)$ to be the set of edges from vertices in X to vertices in Y. For any set of vertices $X \subseteq V$, we define the complement of X, denoted by X^c, as $\{v \in V : v \notin X\}$. For any subset of vertices $X \subseteq V$, $\partial(X)$ denotes the set of vertices in X^c such that each of them has at least one edge from some vertex in X.

2.1 Conductance Measure

Conductance is an important notion of expansion of a graph. There are different ways to define the conductance of a graph in the literature. Sinclair [12] gives a traditional definition of conductance, while there are also other, slightly different, definitions related to the estimation of the runtime of rumor spreading, c.f., Mosk-Aoyama et al. [13] and Censor-Hillel et al. [14, 15]. On the other hand, a definition of conductance that measures how *well-connected* the graph is, has been used by Giakkoupis [16] and Sauerwald and Stauffer [17].

In the next sections, we use a generalized version of this definition of conductance, due to its *graph theory* nature. The *conductance* $\Phi(G)$ of a graph $G = (V, E)$ is defined as follows

$$\Phi(G) = \min_{X\subseteq V: 0\leq \mathrm{vol}(X)\leq \mathrm{vol}(V)/2} \frac{\sum_{(v,w)\in|E(X,X^c)|} p_{v,w}}{\mathrm{vol}(X)}.$$

2.2 Related Work

The closest area to the estimation of invasion time is the analysis of rumor spreading protocols. The most popular rumor spreading protocols are: Push, in which an informed vertex selects randomly its (out-)neighbor to whom it sends a message, and Pull, in which an uninformed vertex selects its (in-)neighbor from whom it gets the message (provided that the chosen vertex has it). Their modifications and combinations were also considered. These processes differ from the invasion process in many ways. First, they were studied in undirected unweighted graphs; in other words, all weights were set to 1. Second, they run at vertices rather than at edges, which is the case of invasion. Nevertheless, in what follows we give an overview of related results in the rumor spreading area.

Runtime of Push-Pull has also been bounded on special graph classes. For complete graphs, it is known that $\Theta(\log n)$ rounds are sufficient [18, 19, 20, 21] while for social networks represented as Preferential Attachment graph, Chierichetti et al. [22] showed that Push-Pull protocol informs all vertices within $O(\log^2 n)$ rounds, whp. The latter has been further improved to $\Theta(\log n)$ by Doerr et al. [23]. Doerr et al. [23] also showed how to obtain optimal runtime for these types of graphs by modifying Push-Pull protocol. This line of research has been later generalized by Giakkoupis [24], who upper bounded the running time of push-pull process by $O((1/\Psi)\log n)$, where $\Psi(G)$ is the *vertex expansion* of a graph $G = (V, E)$: $\Psi(G) := \min_{X\subseteq V:1\leq|X|\leq n/2} \frac{|\partial(X)|}{|X|}$.

In relating runtime of rumor spreading with conductance in general graphs, Chierichetti et al. [25] first showed that Push-Pull protocol broadcasts the message within $O(\Phi^{-6}(G)\log^4 n)$ rounds whp, and later the same authors improved the bound to $O(\frac{\log^2 1/\Phi}{\Phi}\log n)$ in [26]. Giakkoupis [16] closed the gap by providing a tight bound of $\Theta((1/\Phi)\log n)$.

Censor-Hillel and Shachnai [15] slightly modified the random protocol by adding some determinism in it. They analyzed their protocol based on a different notion they called "weak conductance". Recently, Censor-Hillel [27] modified Push-Pull protocol in order to solve the information dissemination problem with no dependence on conductance and showed that this new protocol solves the rumor spreading problem in at most $O(D + \text{polylog}(n))$ rounds in a graph of diameter D. Nevertheless, their protocol is not the classical Push-Pull protocol.

The results mentioned above apply to the synchronous model. In the asynchronous model, Sauerwald [28] showed that for Push protocol, the expected number of rounds for rumor spreading is asymptotically equivalent to the expected number of rounds in the synchronous model for Push protocol. He introduced a new measure that considers for each $1 \leq k \leq n-1$ the subset X of k vertices that minimizes $\sum_{v\in X} d_{X^c}(v)/d(v)$ (denoted as Φ_k). The expected runtime of Push protocol in the asynchronous model has then been shown to be at most $\sum_{k=1}^{n-1} 1/\Phi_k$. More recently, Kowalski and Thraves Caro [29] defined another new measure that is a relatively tight estimate of the runtime of rumor spreading by Push-Pull protocol.

3. RESULTS

3.1 The New Measure

In this section, we introduce the γ measure — a new measure to estimate invasion time. This measure can be seen as the weighted and edge-oriented version of the measure introduced by Kowalski and Thraves Caro [29] in the context of rumor spreading.

For a given network $N = (s, t, V, E)$, with source s, target t, set of intermediate nodes V and set of directed weighted edges E, we define the following new parameter:

$$\beta_k(N) = \min_{X:X\subseteq V, s\in X, t\notin X, |X|=k} \sum_{v\in X}\sum_{w\in X^c} p_{vw} \ ,$$

where p_{vw} is the weight of edge (v, w). This parameter captures the worst-case bi-directional expansion through a cut of one border of size k. Our new measure, used later for estimating invasion time, is defined as follows:

$$\gamma(N) = \sum_{k=1}^{n-1} \frac{1}{\beta_k(N)} \ .$$

We may skip parameter N from the above parameters $\gamma(N)$, $\beta_k(N)$ if network N is clearly understood from the context.

3.2 Estimating Parameters via Network Flow

We propose to use a network flow approach to model and estimate the invasion process. For this, we build a network to represent a given landscape graph G of size $H \times W$ in the case of invasion (recall the definition of invasion time in Section 1). We adapt the proposed sparsification method in [30] to compute the distance R for the given landscape graph G such one could estimate well the invasion time while restricting to links between patches of distance at most R; the authors developed the following formula:

$$R(G) = \frac{1}{\alpha} \cdot \ln\left(\frac{H + W}{d_{min}(G) \cdot \bar{q}(G)}\right) + d_{min}(G) \ , \qquad (1)$$

where $\bar{q}(G) = \frac{q_{min}(G)}{q_{max}(G)}$ and d_{min} is the minimum distance such that every node in T is reachable from some node in S in graph G. We number columns starting from 0 and we assume that the area (patches) between column 0 and 9 is populated. We also assume that column number 19, 29, 39, $\ldots$, and $W - 11$ as the first column of the target area, where the target area is of length 10 columns. For each prefix of size $H \times 29$, $H \times 39$, $\ldots$, $H \times W - 2$, we define a sub-landscape graph G' of size $H \times (\max\{0, 10i - 2R - 10\} : 10i - 1 + 9)$, where $i = [2, \frac{W-10}{10}]$. Then, we build an invasion network N to represent G' graph as follows.

Invasion network N and estimating invasion time IT. The invasion network N for the sub-landscape graph G' of size $H \times (\max\{0, 10i - 2R - 10\} : 10i - 1 + 9)$ is defined as follows. We distinguish two sets of vertices (each of length 10 columns): the initial populated set S which contains vertices between column $\max\{0, 10i - 2R - 10\}$ and $\max\{0, 10i - 2R - 1\}$, and the target set T which involves vertices in the area between column $10i - 1$ and $10i - 1 + 9$. We add a virtual source vertex s and connect it to each vertex in the initial populated set S by a directed edge with a weight λ, where λ is the maximum, over all vertices, of the sum of the weight of adjacent edges, $\lambda = \max\{\lambda_v : v \in V\}$, where $\lambda_v = \sum_{u\in V} p(v, u)$. We also add a virtual target vertex t and connect each vertex in the target set T to the additional target vertex t by a directed edge with weight λ. Each intermediate vertex (except the source vertex s and the target vertex t) is connecting to all other vertices by edges and given weights that equal to the transition probabilities

$p(v,u)$ between the patches. A constructed invasion network N for a given landscape is given in the Supplementary materials [31].

We compute the network flow of this constructed network with the weight as the capacity.

For a given landscape graph G of size $H \times W$, we consider column number 19, 29, 39, ..., $W-11$ as the first column of the target area and we compute invasion time for each target area. The estimated invasion time IT at target column $10i-1$ is defined as:

$$IT(10i{-}1) = \begin{cases} \frac{\ln(\hat{\#}[0:10i-1+9])}{\text{max-flow}([0:10i-1+9])} \cdot c \\ \qquad \text{if } 1 < i \le \frac{W-10}{10} \text{ and } 10i-1 < 2R+9 \\ IT(10i-11) + \frac{\ln(\hat{\#}[10i-2R-10:10i-1+9])}{\text{max-flow}([10i-2R-10:10i-1+9])} \cdot c \\ \qquad \text{if } 1 < i \le \frac{W-10}{10} \text{ and } 10i-1 \ge 2R+9 \end{cases} \tag{2}$$

where c is a small constant to be interpolated by simulation in Section 5 and $\hat{\#}$ is the total number of non-zero patches between two specified columns [start-column:end-column]. In the following Section 4 we develop a theory justifying that the formula defined by Equation 2 is a good asymptotic estimate of invasion time.

4. ESTIMATES OF INVASION RUNTIME IN REAL LANDSCAPES

In this section we analyze the new measure γ to approximate the number of rounds (round complexity) of the invasion protocol.

THEOREM 1. *For any landscape graph G, the expected* asynchronous *round complexity of invasion protocol is $O(\gamma(N))$, where N is the invasion landscape network of landscape graph G.*

PROOF. We partition an asynchronous execution of invasion protocol into consecutive phases as follows: phase k contains steps in which exactly k vertices are populated. Note that some phases may be empty, and the partition into phases is related with the partition into asynchronous rounds, though the former depends on the random choices made in asynchronous steps while the latter is fixed (i.e., each round contains exactly m asynchronous steps).

Observe that the expected number of steps in phase k is at most $1/\beta_k(N)$. Indeed, let W be the set of k populated vertices during phase k. The probability that a single population operation populates some unpopulated vertex in W^c from some populated vertex in W (i.e., the probability of selecting an edge between W and W^c) is $\frac{1}{m}\sum_{v\in W}\sum_{w\in W^c} p_{vw}$.

Therefore, the probability of populating some unpopulated vertex, when the phase will terminate, is at least

$$\frac{1}{m}\sum_{v\in W}\sum_{w\in W^c} p_{vw} \ge \frac{\beta_k(N)}{m} \ .$$

Because population operation is applied independently for edges in consecutive steps, the expected time for termination of the phase is at most $\frac{m}{\beta_k(N)}$. Hence, the expected time of invasion, in terms of steps, is the sum of expected numbers of steps over $1 \le k \le n-1$, which gives at most $\sum_{k=1}^{n-1} \frac{m}{\beta_k(N)}$ steps, which in turn is equal to $\gamma(N)$ asynchronous rounds. $\square$

The proof of the following theorem is analogous to the one in [29] (the main difference is that we consider random sequence of edges instead of vertices) and we give the proof in the Supplementary materials [31] for completeness.

THEOREM 2. *For any landscape graph G, the number of rounds in an asynchronous execution of invasion protocol is asymptotically not bigger than the number of rounds in a synchronous execution of invasion protocol plus $\log n$, with high probability. On the other hand, the expected number of rounds in a synchronous execution of invasion protocol on landscape graph G is $O(\gamma(N) + \log n)$, where N is the invasion landscape network of landscape graph G.*

It follows that the asymptotic bounds on synchronous invasion protocol apply also to asynchronous invasion protocol, modulo an additive logarithm of n.

The following theorem is the main theoretical finding, which will be later used for comparison with simulation results. Denote $\max_{v\in V} \text{vol}(v)$ by $\text{vol}_{\max}$ and $\min_{v\in V} \text{vol}(v)$ by $\text{vol}_{\min}$, and assume that they are constants (depending on the landscape).

THEOREM 3. *For every landscape with n patches and its invasion landscape network N, it holds that the invasion time is $O\left(\lceil \frac{\log n}{\Phi(N)} \rceil\right)$.*

PROOF. By Theorem 2, it is enough to show that $\gamma(N) = O\left(\frac{\log n}{\Phi(N)}\right)$. The following holds:

$$\begin{aligned}\beta_k(G) &= \min_{X:X\subseteq V, |X|=k} \sum_{v\in X} \sum_{w\in X^c,(v,w)\in E} p_{vw} \\ &\ge \min_{X:X\subseteq V, |X|=k} \left(k \cdot \text{vol}_{\min} \cdot \frac{\sum_{v\in X}\sum_{w\in X^c,(v,w)\in E} p_{vw}}{\sum_{v\in X}\text{vol}(v)}\right) \\ &\ge k \cdot \text{vol}_{\min} \cdot \Phi(N) \ .\end{aligned}$$

Finally, we get

$$\begin{aligned}\gamma(G) &= \sum_{k=1}^{n-1} \frac{1}{\beta_k(G)} \le \sum_{k=1}^{n-1} \frac{1}{k\cdot \text{vol}_{\min}\Phi(N)} = \frac{H_{n-1}}{\text{vol}_{\min}\Phi(N)} \\ &= O\left(\frac{\log n}{\Phi(N)}\right) \ .\end{aligned}$$

$\square$

Observe that, although the exact computation of conductance (or γ) is hard, for small landscapes N of width $2R$, where R is the sparsification distance from Equation 1, $\Phi(N)$ could be well estimated by minimum cut in the graph, which is equal to maximum flow in the graph (by the well-known min-max theorem). This combined with Theorem 3 justifies the (asymptotic) prediction formula IT introduced in Equation 2: the formula is a sum of estimates of the invasion times over a sequence of small and partly overlapping landscapes (of width $2R$). In Section 5 we will interpolate constants in the formula, and in Section 6 we verify it for randomly selected landscape.

5. SIMULATIONS — ADJUSTING THE PREDICTION FORMULA (EQUATION 2)

In this section, we describe the simulation environment and adjust the prediction formula (Equation 2) using simulations.

5.1 The *Full* Simulation Method

For a landscape graph G of size $H \times W$, we use the formula of colonization probability proposed by Hodgson et al. [1] to define the *transition probability* between patches v and u as $p(v,u) = q(v) \cdot \frac{\exp(-\alpha d(v,u))}{\left(\frac{2\pi}{\alpha^2}\right)-1}$, where $\alpha > 0$ is the dispersal coefficient assumed to be the same for all patches and $d(v,u)$ is the Euclidean distance between patches v and u. We simulate the behavior of the invasion process by building a simulator that uses the *full* method. In each round of the invasion process, for every pair of patch vertices v and u such that v is populated and u is not, we determine whether v populates u or not by the probability $p(v,u)$. In the *full* invasion method, each populated patch vertex in the landscape tries to populate every other unpopulated patch in the whole landscape. This is in contrast to other methods where only vertices of limited distance apart are considered (see [30] for more details).

More formal description of the algorithm implementing the *full* method is given in the Supplementary materials [31]. The generic structure of the *full* method contains inputs, outputs, and COUNT ROUNDS function. The COUNT ROUNDS function counts the number of rounds required for invasion. The function includes nested loops of three levels. The main loop (starts at line 7) counts the number of rounds to populate any of the target patches. The second level loop (starts at line 9) is for all populated patches that are trying to populate unpopulated patches. The inner level loop (starts at line 12) is for all unpopulated patches. Each unpopulated patch becomes populated if the *transition probability* between the populated and unpopulated patches is greater than a randomly generated number between zero and one (lines 15-18). We consider only populating a patch with non-zero quality. The COUNT ROUNDS function terminates when any of the non-zero target patches become populated and returns the number of rounds needed for a successful invasion as well as the execution time of simulation.

This full simulation method is time consuming, and what is more, it is difficult to analyze and stop at the proper time as it is based on a Markov Chain of exponential size wrt the size of the landscape. Therefore, we aim to develop new methods to estimate the number of rounds of the invasion process in a more efficient way.

5.2 The Studied Landscapes

The study landscapes were of size 5 rows/height (pixels) and 299 columns/width (pixels) extracted from UK Land-cover Map 2007 (LCM2007) data [32] and gridded at 100 m cell resolution (see [31]). Each patch (pixel) in each extracted landscape of low and medium quality provides the percentage cover of semi-natural grassland aggregate class across the UK, while a pixel in a landscape of high quality presents the percentage cover of the improved grassland UK map [32]. The percentage cover at each patch is considered as the quality of the patch. For examination purposes, three groups of landscape qualities namely: low quality, medium quality, and high quality have been formulated to represent the quality of each extracted landscape. In such an extracted landscape, if the average of all patches' qualities is between 0% and 5%, 5% and 25%, 25%, and 100%, then the landscape is of low, medium, and high quality, resp.

It has been assumed that all patches at the first 10 columns of each landscape are occupied by species and the goal is to estimate the time needed to populate any of the patches at a target area. It has been assumed that the length of the target area is 10 columns. Each of the column numbers $19, 29, 39, \ldots, W-11$ has been considered as the first column of the target areas for the occupied patches in the first 10 columns of the considered landscapes. Therefore, each landscape has been extracted such that at least one of the patches at each target area (i.e., 19-28, 29-38, 39-48 etc.) has non-zero quality.

5.3 Simulation Setting and Adjusting Prediction

The simulation is directed at three goals. The first is to compute the invasion time using the *full* simulation method and investigate how the dispersal coefficient α and landscape quality affect on the invasion duration. The second is compute the invasion time based on the proposed prediction method in Section 3.2 (Equation 2) which is based on computing maximum flow for the constructed invasion landscape network N using one of the known algorithms[1]. The third goal is to adjust the prediction formula IT (Equation 2) using simulations. Finally, we aim to combine results from simulation and the proposed prediction method for invasion and compare them.

One important characteristic to consider when computing the total time of invasion is also the number of iterations needed for the (average) invasion time to stabilize on the outputted duration of invasion. Accordingly, we define the stabilization time (ST) for a given landscape as the time t such that the change in the average number of rounds for the invasion time (IT) between t and $2t$ is less than or equal to 2%: $\forall \tau \in (t, 2t]$, $|IT(\tau) - IT(t)| \leq 0.02 \cdot IT(t)$, where $IT(\tau) = \frac{\sum_{j=1}^{\tau} Rounds(j)}{\tau}$ and $Rounds(j)$ is the number of rounds needed for invasion at iteration j.

For each prefix $5 \times 30, 5 \times 40, 5 \times 50, \ldots, 5 \times 299$ in each extracted rectangular landscape we run the following procedure. We run *full* simulation until stabilization point and compute the average number of rounds over 2ST independent iterations (approximated invasion time) and the execution time of simulation. Then, we compute the invasion time based on computing maximum flow for the constructed invasion landscape network N (as described in Section 3.2) and the execution time of computing network flows (see the Supplementary materials [31]). Finally, we compare the invasion time produced by these two methods by computing the ratio between them (invasion time by simulation over invasion time by prediction method). That has been done for α equals: 0.25, 0.5, 1, and 2.

Interpolating constant c in IT formula (Equation 2) based on simulations. In order to interpolate constant c in the IT prediction formula (Equation 2), we do the following. We first compute the ratio of invasion time (simulation results over prediction results assuming $c = 1$) for each prefix in each of the studied rectangular landscapes (low, medium, high). Then, we divide the ratio of the medium and high qualities by the ratio of low quality. Finally, we compute constant c that gives the smallest distance between simulation and prediction curves (the best c that makes the invasion time ratio closer to 1). Table 1 shows the interpolated constant c (by simulation) for each landscape quality and dispersal coefficient.

[1]We use NetworkX package in Python programming language to calculate maximum flow.

Table 1: The interpolated constant c for each landscape quality and dispersal coefficient α.

Landscape quality	α=0.25	α=0.5	α=1	α=2
Low	1	1	1	1
Medium	1.3	1.45	1.65	1.8
High	1.35	1.65	1.95	2.35

5.4 Simulation Results vs. Prediction Results

For each prefix $5 \times 30, 5 \times 40, 5 \times 50, \ldots, 5 \times 299$ in each extracted rectangular landscape of low, medium, and high quality. We compute the approximated invasion time (i.e., average number of rounds over 2ST independent repetitions) using *full* simulation method and the proposed method (i.e., Equation 2 with constants in Table 1) which based on computing maximum flow in the constructed invasion landscape network. Both simulation and prediction results show that the invasion time grows with the growth in the prefix width. Prefixes of low quality take longer time (larger number of rounds) than medium quality to be invaded, while high quality prefixes need shorter time. The increase in the dispersal coefficient α from 0.25 to 2 shows an increase in the invasion time in all qualities (see the Supplementary materials [31]).

Importantly, based on theoretical analysis of the new γ measure to approximate the number of rounds (invasion time) in Section 4 we compute the ratio between the computed invasion time by simulation and prediction for each prefix. Comparison of the obtained results based on the *full* and flow methods shows that our proposed flow way (Equation 2 with constants in Table 1) gives a good approximation for the invasion time, as shown in Figure 1. The invasion time ratio is around one in all types of quality (low, medium, and high) and for different values of the dispersal coefficient α.

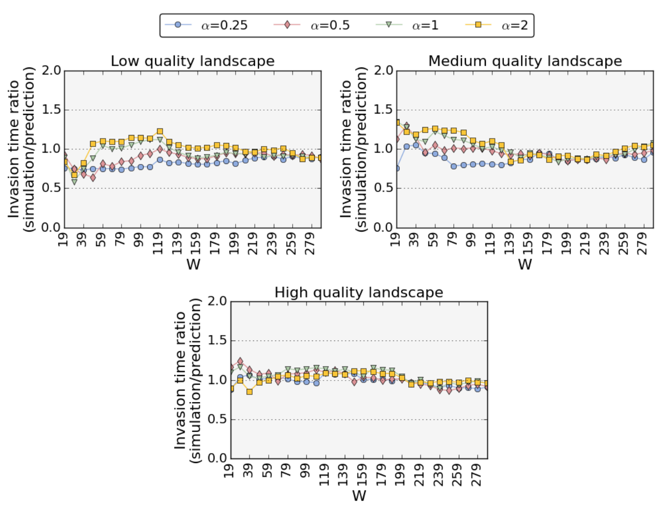

Figure 1: The ratio of invasion time (simulation over prediction) for each prefix in each of the studied landscapes (low, medium, high) when α takes four values: 0.25, 0.5, 1 and 2.

6. VERIFICATION OF THE PREDICTION FORMULA (EQUATION 2 WITH TABLE 1)

In order to test the robustness of our proposed prediction method (i.e., Equation 2 with constants in Table 1), we performed verification in a landscape of mixed quality which is selected randomly from semi-natural grassland UK map [32] (see the Supplementary materials [31]). To ensure robustness, we verify our new method on a landscape of 10 rows and 299 columns, where the height is different than the height in landscapes used in Section 5.

For each prefix $10 \times 30, 10 \times 40, 10 \times 50, \ldots, 5 \times 299$ on this extracted landscape and for each considered value of the dispersal coefficient α, we run *full* simulation method and our proposed prediction method (i.e., Equation 2 with constants in Table 1) independently, in order to get the estimated time of invasion as well as the execution time of simulation and prediction. Then, we compute the ratio of the estimated time of invasion (i.e., invasion time by simulation over invasion time by prediction).

Figure 2 gives the ratio of invasion time. As presented in the Figure, the ratio is between 0.5 and 1 for α equals to 0.25 and 0.5 and is much better and much closer to one for α equals to 1 and 2 (for detail results, see the Supplementary materials [31]).

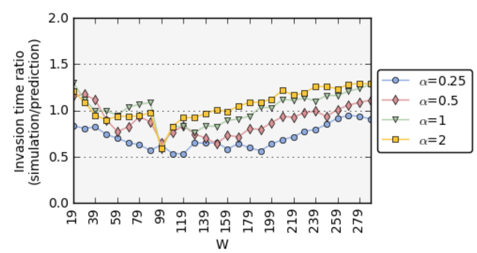

Figure 2: The ratio of invasion time (simulation over prediction) for each prefix in 10×299 landscape of mixed qualities when α equals: 0.25, 0.5, 1, and 2.

Additionally, we compute the error rate between the presented invasion time ratios in Figures 1 and 2 to see the precision of our prediction when the height of the landscape is doubled. The error rate for a fixed α is defined as: $\frac{\sum_{i=1}^{j}(a-b)^2}{j}$, where a is the average of the invasion time ratio over all three types of quality, b is the invasion time ratio in the mixed qualities landscape, and j is the total number of prefixes. It has been illustrated by the verification experiment that the accuracy of our new prediction method is very good as when the height of the landscape is increased to the double and the chosen landscape is of mixed qualities, the error rate is small between 2% and 4%. The error rates are 0.04, 0.02, 0.04, and 0.04 when α are 0.25, 0.5, 1, and 2, respectively.

To summarize, the verification experiment has confirmed that the desired accuracy (simulation over prediction $\simeq$1) has been achieved using the developed formula for the invasion time. Moreover, the error rate between the invasion time ratios (Figures 1 and 2) is small (0.02 & 0.04) for all values of the dispersal coefficient α. On the other hand, it seems that the precision of our prediction drops with the increase in the height of the landscape.

7. TIME AND MEMORY IMPROVEMENT

This section shows how much computational time and memory we save when using the proposed prediction method (Equation 2) for properly selected distance R (using Equation 1) to compute the invasion time.

7.1 Saved Time

The carried experiments in this paper demonstrate that the actual execution time needed to compute the estimated duration of the invasion process is substantially reduced by our new prediction method (Equation 2) for all landscapes of different qualities. Figure 3 illustrates how much the prediction method is faster than the *full* simulation method for all four considered landscapes. For many cases, the *full* simulation method takes 10-10000 times longer to compute invasion time and this ratio can become as high as 94000 for low quality landscape. We note that in general, for a given landscape width, the speedup of the prediction method increases as the dispersal coefficient α increases. On the other hand, in most cases, when α is fixed, the speedup increases as the width of landscape increases.

In more details, in the landscape of size 5×299 and of low quality, the execution time of *full* simulation is 9086.29 seconds while it takes only 0.09 seconds to compute invasion time using the new prediction method. That means the *full* simulation method takes around 94000 times longer to compute. We could envisage that when we are running the *full* simulation in a very large landscape e.g., a landscape of size 500×500, the computation time will be substantially reduced from maybe days/weeks to hours.

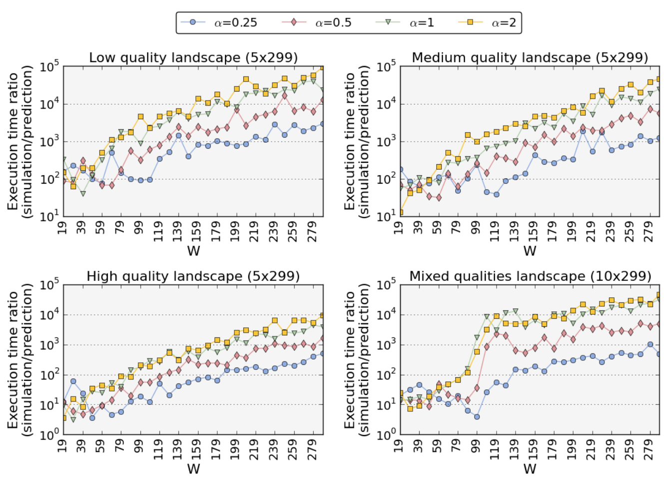

Figure 3: The ratio of the execution time (simulation over prediction) with logarithmic scale versus the landscape width in each of the considered landscape (low, medium, high, and mixed) when the dispersal coefficient α equals: 0.25, 0.5, 1, and 2.

7.2 Saved Memory

In order to investigate the saved memory, we compute the sparsification rate. It is defined as the ratio of the total number of edges between patches in the landscape over the total number of edges between patches in the constructed invasion landscape network N; in both numbers we count only edges with both ends of non-zeros quality. Supplementary materials [31] contain figures illustrating the sparsification rate versus the landscape width.

Generally, the relation between the sparsification rate and the landscape width can be seen as a linear tendency, where the linear coefficient grows linearly with α. As can be seen in the Supplementary materials [31], the sparsification rate depends on the computed local distance R for each α: the rate increases with the decrease of R. For instance, the rate increase in all qualities for $\alpha = 2$ refers to a decline in R; where R depends on α: R declines with the increase in α from 0.25 to 2. The computed local distance R (using Eq. 1) for the landscape of quality low, medium, high, and mixed, are as follows. For $\alpha = 0.25$ R values are 38, 40, 42, and 41, resp.; for $\alpha = 0.5$ R values are 21, 22, 22, and 22, resp.; for $\alpha = 1$ R values are 12, 13, 12, and 12, resp.; for $\alpha = 2$ R values are 8, 8, 7, and 10, resp.

To summarize, the practical part in this paper has confirmed the following: the desired accuracy (simulation over prediction $\simeq$1) has been achieved using the developed formula for the invasion time (Equation 2) and constants c in Table 1. Moreover, saving in computational time and memory is rapidly growing with the dispersal coefficient α and the landscape width.

8. CONCLUSION

This work was prompted by a desire to construct the *full* model that visualizes the invasion process using network modeling approach. The *full* model is based on the colonization probability of each patch (vertex) to be colonized/invaded, which has been presented in previous work. More importantly, the main contribution of this work is the introduction of a new theoretical measure γ that estimates the expected time performance for asynchronous and synchronous executions of the invasion protocol. The γ measure can be upper bounded by $\frac{\ln n}{\Phi(N)}$, which is a parameter that can be fastly approximated for small landscapes by using network flow algorithm. We extend this theoretical result into prediction formula IT, in which we interpolate constants by simulations. The capability of our model is to approximate accurately the number of rounds needed to invade large homogeneous landscapes in a short computational time; this has been clearly achieved as demonstrated in Figures 1 and 2 (accuracy) and in Figure 3 (computational time).

The simulations demonstrate that the dispersal coefficient α and the landscape quality affect on the invasion time and the computation time needed for it. A large α requires a long time to compute the invasion time, in all types of quality, while a short time is sufficient for a small α. Landscapes of low quality require the longest time to compute the invasion time among the three formed qualities (low, medium, high).

As for the future work, we aim to study algorithms that modify landscapes in order to improve/slow-down the invasion process within some pre-defined budget. This optimization task will need testing and studying the affect of different landscapes modifications on the invasion process. We believe that the network flow framework for the invasion process, introduced and tested in this work, could help to obtain satisfactory optimization results in a short computation time, as opposed to the optimization within the full simulation process (modeled as exponentially big Markov Chain) which is clearly not feasible.

9. REFERENCES

[1] Jenny A Hodgson, Chris D Thomas, Steve Cinderby, Howard Cambridge, Paul Evans, and Jane K Hill. Habitat recreation strategies for promoting adaptation of species to climate change. *Conservation Letters*, 4:289–297, 2011.

[2] Brian Huntley, Yvonne C Collingham, Stephen G Willis, and Rhys E Green. Potential impacts of climatic change on European breeding birds. *PLoS ONE*, 3:e1439, 2008.

[3] Steven J Phillips, Paul Williams, Guy Midgley, and Aaron Archer. Optimizing dispersal corridors for the cape proteaceae using network flow. *Ecological Applications*, 18:1200–1211, 2008.

[4] MS Warren, JK Hill, JA Thomas, J Asher, R Fox, B Huntley, DB Roy, MG Telfer, S Jeffcoate, P Harding, et al. Rapid responses of british butterflies to opposing forces of climate and habitat change. *Nature*, 414:65–69, 2001.

[5] I Chen, JK Hill, R Ohlemüller, DB Roy, and CD Thomas. Rapid range shifts of species associated with high levels of climate warming. *Science*, 333:1024–1026, 2011.

[6] G.-R. Walther, E. Post, P. Convey, A. Menzel, C. Parmesan, T. J. C. Beebee, J.-M. Fromentin, O. Hoegh-Guldberg, and F. Bairlein. Ecological responses to recent climate change. *Nature*, 416(6879):389–395, 2002.

[7] Olivier Honnay, Kris Verheyen, Jan Butaye, Hans Jacquemyn, Beatrijs Bossuyt, and Martin Hermy. Possible effects of habitat fragmentation and climate change on the range of forest plant species. *Ecol Lett*, 5:525–530, 2002.

[8] F. Skov and J. C. Svenning. Potential impact of climatic change on the distribution of forest herbs in europe. *Ecography*, 27:366–380, 2004.

[9] Chris D Thomas, Alison Cameron, Rhys E Green, Michel Bakkenes, Linda J Beaumont, Yvonne C Collingham, Barend FN Erasmus, Marinez Ferreira De Siqueira, Alan Grainger, Lee Hannah, et al. Extinction risk from climate change. *Nature*, 427:145–148, 2004.

[10] Jenny A Hodgson, Chris D Thomas, Calvin Dytham, Justin MJ Travis, and Stephen J Cornell. The speed of range shifts in fragmented landscapes. *PLoS ONE*, 7:e47141, 2012.

[11] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[12] Alistair Sinclair. *Algorithms for random generation and counting - a Markov chain approach*. Progress in theoretical computer science. Birkhäuser, 1993.

[13] Damon Mosk-Aoyama and Devavrat Shah. Fast distributed algorithms for computing separable functions. *IEEE Transactions on Information Theory*, 54(7):2997–3007, 2008.

[14] Keren Censor-Hillel and Hadas Shachnai. Partial information spreading with application to distributed maximum coverage. In *PODC*, pages 161–170, 2010.

[15] Keren Censor-Hillel and Hadas Shachnai. Fast information spreading in graphs with large weak conductance. In *SODA*, pages 440–448, 2011.

[16] George Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *STACS*, pages 57–68, 2011.

[17] Thomas Sauerwald and Alexandre Stauffer. Rumor spreading and vertex expansion on regular graphs. In *SODA*, pages 462–475, 2011.

[18] A. Demers, D. Greene, C. Houser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. *SIGOPS Oper. Syst. Rev.*, 22:8–32, January 1988.

[19] Benjamin Doerr and Mahmoud Fouz. Asymptotically optimal randomized rumor spreading. In *ICALP*, pages 502–513, 2011.

[20] Robert Elsässer. On the communication complexity of randomized broadcasting in random-like graphs. In *SPAA*, pages 148–157, 2006.

[21] Richard M. Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. In *FOCS*, pages 565–574, 2000.

[22] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumor spreading in social networks. *Theor. Comput. Sci.*, 412(24):2602–2610, 2011.

[23] Benjamin Doerr, Mahmoud Fouz, and Tobias Friedrich. Social networks spread rumors in sublogarithmic time. In *STOC*, pages 21–30, 2011.

[24] George Giakkoupis. Tight bounds for rumor spreading with vertex expansion. In *SODA*, pages 801–815, 2014.

[25] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumour spreading and graph conductance. In *SODA*, pages 1657–1663, 2010.

[26] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Almost tight bounds for rumour spreading with conductance. In *STOC*, pages 399–408, 2010.

[27] Keren Censor-Hillel, Bernhard Haeupler, Jonathan A. Kelner, and Petar Maymounkov. Global computation in a poorly connected world: fast rumor spreading with no dependence on conductance. In *STOC*, pages 961–970, 2012.

[28] Thomas Sauerwald. On mixing and edge expansion properties in randomized broadcasting. *Algorithmica*, 56:51–88, 2010.

[29] Dariusz R. Kowalski and Christopher Thraves Caro. Estimating time complexity of rumor spreading in ad-hoc networks. In *ADHOC-NOW*, pages 245–256, 2013.

[30] Daniyah A. Aloqalaa, Jenny A. Hodgson, and Prudence W. H. Wong. The impact of landscape sparsification on modelling and analysis of the invasion process. In *SEA*, pages 32:1–32:16, 2017.

[31] Daniyah A. Aloqalaa, Jenny A. Hodgson Dariusz R. Kowalski, and Prudence W. H. Wong. Estimating invasion time in real landscapes: Supplementary materials, 2018.

[32] D Morton, C Rowland, C Wood, L Meek, C Marston, G Smith, R Wadsworth, and I Simpson. Final report for lcm2007-the new uk land cover map. *Countryside Survey Technical Report No 11/07*, 2011.