A critical exponent for shortest-path scaling in continuum percolation

Tim Brereton[1], **Christian Hirsch**[1], **Volker Schmidt**[1], **Dirk Kroese**[2]

[1] Institute of Stochastics, University of Ulm, Germany

[2] School of Mathematics and Physics, University of Queensland, Australia

Abstract. We carry out Monte Carlo experiments to study the scaling behavior of shortest path lengths in continuum percolation. These studies suggest that the critical exponent governing this scaling is the same for both continuum and lattice percolation. We use splitting, a technique that has not yet been fully exploited in the physics literature, to increase the speed of our simulations. This technique can also be applied to other models where clusters are grown sequentially.

1. Introduction

A natural measure of distance on percolation clusters is chemical distance — the number of edges in the shortest path between two points. This contrasts with Euclidean distance, which measures geometric distance rather than path length. The relationship between chemical distance and Euclidean distance plays an important role in describing percolation clusters; see, e.g., [1], [2] and [3]. One object of interest is the scaling behavior of the chemical distance between sites that are close to one another in the sense of Euclidean distance; see, for example, [4, 5, 6, 7]. It has been shown that, for geometrically close points, the tail of the chemical distance distribution exhibits power law behavior in the vicinity of the percolation threshold; see [8]. Estimates of the critical exponent are given in [5] and [6].

In this paper, we investigate the scaling behavior of chemical distance in a continuum percolation setting. More precisely, we carry out Monte Carlo simulations in two dimensions that show that the exponent in the prototypical 'overlapping spheres' continuum setting appears to be the same as the exponent in the lattice setting, in agreement with the general belief that lattice and continuum percolation are in the same universality class; see, e.g., [9, 10, 11]. We introduce a technique from the rare-event simulation literature, multilevel splitting (or simply 'splitting'), to increase the speed and accuracy of our estimation procedure. This technique can be applied, more generally, to improve the efficiency of simulations in both continuum and lattice settings that are carried out in a sequential fashion. Examples of such simulation techniques include the popular Leath growth model (introduced in [12]) and the Alexandrowicz method (introduced in [13]). Splitting has recently attracted interest in the physics literature as a technique for rare-event estimation (see [14] and [15]). To the best of our knowledge, it has not yet been applied as a sampling technique. However, similar sequential techniques have been used; see, for example, [16] and [17].

Our definition of chemical distance in a continuum percolation setting is a natural extension of chemical distance as defined in lattice percolation settings. It also plays a role in the study of minimal spanning forests, which are related to the study of spin-glass models; see [18, 19]. Note that the asymptotic behavior of the chemical distance distribution is well understood for supercritical continuum percolation; see [20]. For a survey of further recent developments in continuum percolation, see [21].

2. Setting

The 'overlapping spheres' model consists of spatially uncorrelated points, specifically the points of a homogenous Poisson point process in $\mathbb{R}^d$ with fixed intensity (we set the intensity to 1 to simplify calculations). A sphere of radius $\gamma/2$ is drawn around each point. Edges are placed between points if their spheres overlap; that is, if the Euclidean distance between them is less than γ. This model has been extensively studied; see, for example, [22, 23, 24] and [25]. Connecting points in this manner results in a random graph. If the intensity is held constant and γ is allowed to increase, percolation occurs above the critical threshold, γ_c, which is dependent on d, the dimension of the model.

We are interested in the scaling behavior of chemical distance for points that are close to one another. Therefore, we consider a modified version of the 'overlapping spheres' model, where we add two points to the Poisson process and consider the random graph, G^γ, defined using the connection rule given above. We fix one of these two additional points, o, as the origin and place the other point, v_γ, so that it lies a distance γ from o. Specifically, we set $v_\gamma = (\gamma, 0, \ldots, 0)$.

We consider the distribution of the chemical distance, X^γ, between o and v_γ. A similar quantity for lattice percolation is considered in [5], where it is argued that this quantity follows a power law with exponent λ_{lat} and, in two dimensions, an estimate is given of $\lambda_{\mathsf{lat}} = 2.1055 \pm 0.0010$. The exponent λ_{lat} is closely connected to the shortest path fractal dimension, d_{min}, which describes the rate at which the average chemical distance between two points grows as a function of their Euclidean distance. Indeed, in [7], it was shown that $\lambda_{\mathsf{lat}} = 1 + 5/(4d_{\mathsf{min}})$. Thus, determining λ_{lat} is equivalent to determining d_{min} and vice versa. Using this result, [7] provided a refined estimate, based on the exact determination of a related exponent, of $\lambda_{\mathsf{lat}} = 2.1056 \pm 0.0003$. More recent estimates of d_{min}, obtained in [26], result in an estimate of $\lambda_{\mathsf{lat}} = 2.10544 \pm 0.00002$. In the two dimensional continuum setting, we provide strong numerical evidence that, for $\gamma = \gamma_c$, we have $\mathbb{P}(X^\gamma = x) \sim x^{-\lambda}$ as $x \to \infty$. We then estimate the exponent, λ, and show that it appears likely to have the same value as the exponent estimated in [5] and [7].

3. Simulating X^γ and Estimating λ

Our estimation procedure involves generating independent and identically distributed copies of X^γ, the chemical distance defined above, then using maximum likelihood to estimate λ.

3.1. Simulating X^γ

Generating realizations of X^γ amounts to finding the shortest path (if it exists) between o and v_γ in G^γ. Our approach to simulating X^γ is based on the method introduced in [5]. This is an extension of the classical Leath model, introduced in [12], where clusters are grown in shells. A similar approach to growing continuum percolation clusters was used in a three dimensional setting in [27].

Because we are interested in the asymptotic behavior of X^γ, we increase the speed of our algorithm by only considering values of X^γ greater than 2. That is, we assume there are no points in $B_\gamma(o) \cap B_\gamma(v_\gamma)$, where $B_\gamma(v) = \{x \in \mathbb{R}^d : \|x - v\| < \gamma\}$.

We 'grow' clusters from the 'seed' points o and v_γ in a sequential fashion by generating Poisson processes in unexplored regions within distance γ of existing points. We are able to proceed in this sequential manner due to the independence property of the Poisson process. We begin by generating points in $B_\gamma(o) \setminus B_\gamma(v_\gamma)$, which form the initial points of the first cluster, and points in $B_\gamma(v_\gamma) \setminus B_\gamma(o)$, which form the initial points of the second cluster. We mark the points o and v_γ as explored and mark the newly generated points as unexplored, as they have not yet been explicitly considered by our algorithm. We then proceed to sequentially examine the unexplored points, which are ordered by their chemical distance from the seed point of their cluster. If two points have the same chemical distance, we choose one at random. We examine an unexplored point, v, by generating a Poisson process in $B_\gamma(v)$, the open ball centered at v with radius γ. We delete all new points in this ball whose Euclidean distance to an explored point is less than γ. We add the surviving new points to the appropriate clusters. If the two clusters connect, we stop the algorithm and return X^γ. If it is no longer possible to add a point (i.e., we have explored all regions within distance γ of an existing point and there are no surviving new points), we stop the algorithm and return the empty set, $\emptyset$. By choosing the new points to explore based on their chemical distance to the seed of their cluster, we are essentially using a variation of Dijkstra's algorithm, i.e., breadth-first search; see [28]. This means that, if the two clusters connect, we can immediately determine the shortest path between o and v_γ.

We use two sequences of sets, $(U_m)_{m\geq 0}$ and $(E_m)_{m\geq 0}$, to describe the unexplored and explored points generated by our algorithm. The set U_m describes the unexplored points remaining at the mth step and the set E_m describes the points that have already been explored at the mth step. These sets consist of triples of the form $(v, c, t) \in \mathbb{R}^d \times \{1, 2\} \times \mathbb{N}$.

The Euclidean coordinates of a point are given by v. If a point is in the cluster containing the origin then $c = 1$ and if a point is in the cluster containing v_γ then $c = 2$. The chemical distance from a point to the originating point of its cluster is given by t.

Because the running time of the algorithm is potentially unbounded, we introduce a stopping condition as in [5]. If the sizes of the clusters grow beyond a certain threshold without meeting, we terminate the algorithm and return $\emptyset$. We measure the sizes of the clusters using the sequence $(L_m)_{m \geq 0}$, where L_m is the minimal chemical distance from a point in U_m to the originating point of the cluster containing it. We denote the threshold by L_{max}. Note that this stopping condition means we are simulating X^γ conditional on $X^\gamma \leq 2L_{\mathsf{max}} + 1$.

The algorithm is described precisely in the following.

Algorithm (Simulating X^γ)

1. *Set* $U_0 = \emptyset$ *and* $E_0 = \{(o, 1, 0), (v_\gamma, 2, 0)\}$. *Set* $m = 0$.
2. *Generate* $Y \sim \mathsf{Pois}(\kappa_d \gamma^d)$ *points, where* κ_d *denotes the volume of* $B_1(o)$ *in* $\mathbb{R}^d$, *in* $B_\gamma(o)$ *and remove all points that also lie in* $B_\gamma(v_\gamma)$. *Add the remaining points to* U_0 *with* $c = 1$ *and* $t = 1$.
3. *Generate* $Y \sim \mathsf{Pois}(\kappa_d \gamma^d)$ *points in* $B_\gamma(v_\gamma)$ *and remove all points that also lie in* $B_\gamma(o)$. *Add the remaining points to* U_0 *with* $c = 2$ *and* $t = 1$.
4. *Choose a point* $p = (v, c, t)$ *from the current set of unexplored points,* U_m, *with minimum chemical distance* t. *Set* $U_{m+1} = U_m \setminus \{p\}$, $E_{m+1} = E_m \cup \{p\}$, *and* $L_m = t$.
5. *Let* $p_1^o = (v_1^o, c_1^o, t_1^o), \ldots, p_J^o = (v_J^o, c_J^o, t_J^o)$ *be the points in* E_m *that are not in the same cluster as* p. *Let* $\mathcal{J}_S = \{j : v \in B_\gamma(v_j^o)\}$ *be the set of indices associated with those points from* $\{v_1^o, \ldots, v_J^o\}$ *that can be connected to* v *by an edge with length less than* γ. *If* $\mathcal{J}_S \neq \emptyset$ *terminate the algorithm and return* $X^\gamma = t + t_{\mathsf{min}} + 1$, *where* $t_{\mathsf{min}} = \min\{t_j^o : j \in \mathcal{J}_S\}$.
6. *Generate* $Y \sim \mathsf{Pois}(\kappa_d \gamma^d)$ *points,* $\widetilde{v}_1, \ldots, \widetilde{v}_Y$, *distributed uniformly in* $B_\gamma(v)$. *Set* $y = 1$.
7. *Let* $v_1^e, \ldots, v_K^e$ *be the first components of the points in* E_m. *If* $\widetilde{v}_y \notin \bigcup_{k=1}^K B_\gamma(v_k^e)$, *set* $U_{m+1} = U_{m+1} \cup \{(\widetilde{v}_y, c, t + 1)\}$.
8. *If* $y < Y$, *set* $y = y + 1$ *and repeat from step* 7.
9. *If* $L_m > L_{\mathsf{max}}$, *terminate the algorithm and return* $\emptyset$.
10. *If* $U_{m+1} \neq \emptyset$, *set* $m = m + 1$ *and repeat from step* 4. *Otherwise, return* $\emptyset$.

In practice, the data structures used are chosen carefully in order to improve the performance of the above algorithm. In particular, the performance of steps 3 and 5 is improved considerably by using a data structure for the family of sets $(E_m)_{m \geq 0}$ that orders the points according to their Euclidean coordinates. For instance, one can discretize the Euclidean space into boxes and use an array structure to access all points in a given box; see, for example, the approaches taken in [9, 10, 27], and [29].

3.2. Illustration of the Simulation Algorithm

For the planar case, $d = 2$, the basic idea of the algorithm is illustrated in the following. We begin with the two original vertices (which have different symbols in order to distinguish the two clusters); see Figure 1, left. The area around them is still unexplored. We first examine the left point, o. The pattern marks the area where we know there are no points. We generate a homogeneous Poisson process in the region of radius γ surrounding o. In this case, one point is generated; see Figure 1, center. We mark the new points as belonging to cluster 1 and add the explored point to the set of explored points (marked in black). The same is done for the other original vertex, v_γ. Again, only one point is generated; see Figure 1, right. We next generate points in the region surrounding the point in the first cluster of chemical distance 1 to the originating point. The region that has already been explored (i.e., in which the Poisson process has already been generated) is marked in gray; see Figure 2, left. We mark this point as explored and add the new points to the list of points to be explored. We do the same for the point with chemical distance 1 to the originating point of the second cluster; see Figure 2, center. Next, we consider one of the points in the first cluster that is of chemical distance 2 to its originating point. This point is within distance γ of an explored point in the second cluster, forming a path between o and v_γ of chemical length 4; see Figure 2, right. The algorithm terminates.

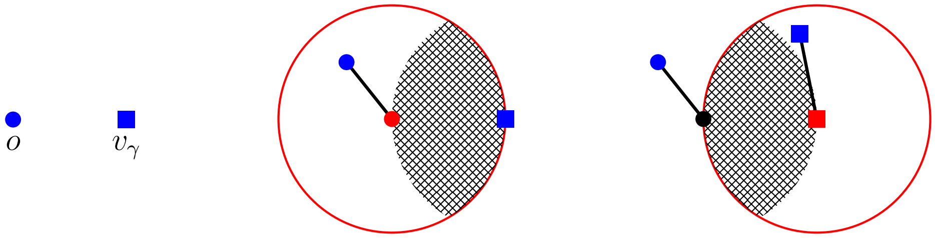

Figure 1. Illustration of the algorithm (steps 1–3).

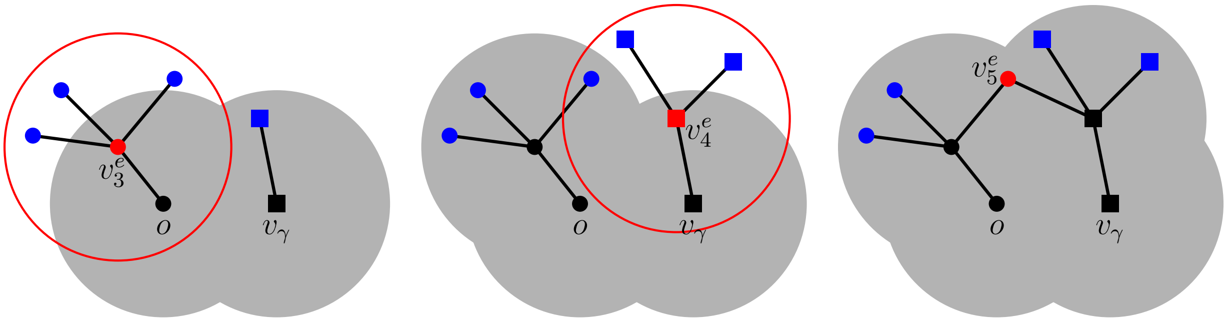

Figure 2. Illustration of the algorithm (steps 4–10).

3.3. Estimating λ

As stated above, for $\gamma = \gamma_c$, we expect that the chemical distance displays power law behavior. For $d = 2$, we provide numerical evidence of this in Section 5. Thus, estimating the critical exponent λ amounts to estimating the parameters of a (right-truncated) power law distribution. There are two commonly used techniques: linear regression and maximum likelihood. It has been demonstrated in a number of studies that maximum likelihood is a more robust approach (see [30, 31]). In particular, almost all the standard assumptions about statistical error made for least squares estimation are violated, resulting in inaccurate estimators of λ and incorrect estimators of standard errors. We therefore follow the approach described in [30] and [32]. That is, we assume that there exists an x_{min} such that, for all $x > x_{\mathsf{min}}$,

$$\mathbb{P}(X^\gamma = x) = cx^{-\lambda}, \tag{1}$$

for some $c > 0$. Given a sample $X_1^\gamma, \ldots, X_N^\gamma$, generated according to the algorithm described in Section 3.1, we determine λ and x_{min} as follows.

The random variable X^γ conditioned on being less than or equal to $x_{\mathsf{max}} = 2L_{\mathsf{max}} + 1$ has probability mass function $\ell : \{x_{\mathsf{min}}, \ldots, x_{\mathsf{max}}\} \to [0, 1]$ with

$$\ell(x) = \mathbb{P}(X^\gamma = x \,|\, x_{\mathsf{min}} \leq X^\gamma \leq x_{\mathsf{max}}) = \frac{x^{-\lambda(x_{\mathsf{min}})}}{\sum_{y=x_{\mathsf{min}}}^{x_{\mathsf{max}}} y^{-\lambda(x_{\mathsf{min}})}}. \tag{2}$$

This yields the maximum likelihood estimator $\widehat{\lambda}(x_{\mathsf{min}})$, which is the solution of

$$\frac{\sum_{y=x_{\mathsf{min}}}^{x_{\mathsf{max}}} y^{-\lambda} \log y}{\sum_{y=x_{\mathsf{min}}}^{x_{\mathsf{max}}} y^{-\lambda}} = \frac{1}{N} \sum_{i=1}^{N} \log X_i^\gamma.$$

Thus, $\widehat{\lambda}$ can be easily calculated via standard numerical procedures.

The choice of x_{min} involves a trade-off. If a too small value of x_{min} is chosen, it may not be true that equation (1) holds, even approximately. However, if x_{min} is too large, only a small fraction of the total sample will be larger than x_{min} and the estimator of λ will be inaccurate. In practice, we estimate $\lambda(x_{\mathsf{min}})$ for $x_{\mathsf{min}} = x_0, x_0 + 1, x_0 + 2, \ldots, x_1$, where $0 \leq x_0 < x_1 \leq x_{\mathsf{max}} - 1$. We then evaluate the goodness-of-fit of each $\widehat{\lambda}(x_{\mathsf{min}})$ by computing the Kolmogorov-Smirnov statistic

$$D(x_{\mathsf{min}}) = \max_{x_{\mathsf{min}} \leq x \leq x_{\mathsf{max}}} |F(x) - \widehat{F}(x)|, \tag{3}$$

where, given the estimator $\widehat{\lambda}(x_{\mathsf{min}})$, $F(x)$ is the theoretical distribution function based on equation (2) and $\widehat{F}(x)$ is the empirical distribution function. We choose the values of x_{min} and $\widehat{\lambda}(x_{\mathsf{min}})$ corresponding to the smallest value of D.

4. Splitting

Accurate estimation of the critical exponent λ requires a sufficiently large sample of values in the right tail of X^γ. We use an approach from rare-event simulation, multilevel splitting, to significantly reduce the simulation effort required to generate values of X^γ larger than a prespecified threshold x. This approach, originally introduced in [33], has primarily been used as a tool for estimating small probabilities, see, for example, [14, 34] and [35], but we use it instead as a tool for sampling from the extremes of a distribution. Although this idea is applied in the specific context of estimating the chemical distance exponent, λ, described above, it can be applied much more generally to simulation models using the Leath method [12], the Alexandrowicz method [13], or any other approach where the simulation is carried out in a sequential fashion.

The basic idea of splitting is to identify paths of a stochastic process that seem likely to generate an event of interest, here $\{x < X^\gamma < \infty\}$, and 'split' these paths to increase the number of paths resulting in this event occurring. In the case where a percolation cluster is generated sequentially the generations of these clusters can be thought of as the time steps of a stochastic process. When this stochastic process satisfies a certain criterion, we 'split' the process by creating a number of replicates. If the splitting criterion is chosen correctly, these replicates should have a high chance of reaching the rare event set.

In our case, the stochastic process that generates X^γ is implicit in the algorithm described in Section 3.1. For every $m \geq 0$ denote by $Y_m = (E_m, U_m, L_m)$ the mth time step of the process consisting of the set of explored vertices, E_m, the set of unexplored vertices, U_m, and L_m, the shortest distance from any of the unexplored points to its seed. See Section 3.1 for the precise definitions of these objects. The process $(Y_m)_{m\geq 0}$ is clearly a time-homogeneous Markov chain. Define

$$\tau_0 = \inf\{m \geq 0 : \text{ the two clusters in } E_m \text{ connect or } U_m \text{ is empty or } L_m > L_{\mathsf{max}}\}.$$

This is the time at which the algorithm stops running and either returns a chemical distance or $\emptyset$. That is, $X^\gamma = f(Y_{\tau_0})$, where

$$f(Y_{\tau_0}) = \begin{cases} 0 & \text{if } U_{\tau_0} = \emptyset \\ t + t_{\mathsf{min}} + 1 & \text{if the two clusters in } E_{\tau_0} \text{ are connected} \\ \infty & \text{if } L_{\tau_0} > L_{\max} \end{cases}.$$

Given that $X^\gamma > x$ only if $\max_{m\geq 0}\{L_m\} \geq (x-1)/2$, we are able to identify paths of $(Y_m)_{m\geq 0}$ that are likely to produce values of X^γ that are larger than x. More precisely, if we define $\tau_1 = \inf\{m \geq 0 : L_m > l_1\}$, where $l_1 \leq (x-1)/2$, then it is clear that $\{x < X^\gamma < \infty\}$ occurs only if $\tau_1 \leq \tau_0$ (i.e., L_m becomes larger than l_1 before the algorithm terminates). This means that in order to generate values of X^γ larger than x, the only important paths are those where $(Y_m)_{m\geq 0}$ enters the set $\{Y = (E, U, L) : L \geq l_1\}$. We increase the proportion of the simulation effort spent on paths resulting in the event $R = \{x < X^\gamma < \infty\}$ by 'splitting' these paths. That is, every time that a process $(Y_m^i)_{m\geq 0}$ hits the level l_1, we stop simulating this process and instead start simulating s new processes, $(Y_m^{i,1})_{m\geq 0}, \ldots, (Y_m^{i,s})_{m\geq 0}$, each beginning at Y_{τ_1} (i.e., $Y_0^{i,1} = \cdots = Y_0^{i,s} = Y_{\tau_1}^i$).

The splitting approach is illustrated in Figure 3. Originally, three paths are generated. These paths continue, with non-decreasing L values, until either an event in the description of τ_0 occurs or an event in the description of τ_1 occurs. One path returns a chemical distance. Another path fails to return a chemical distance (i.e., U becomes empty before a path between o and v_γ can be found). The third successfully reaches the set $\{Y = (E, U, L) : L \geq l_1\}$. This path (which is thicker than the others) splits into four paths. One successfully reaches the target set, R, and the other three prematurely terminate.

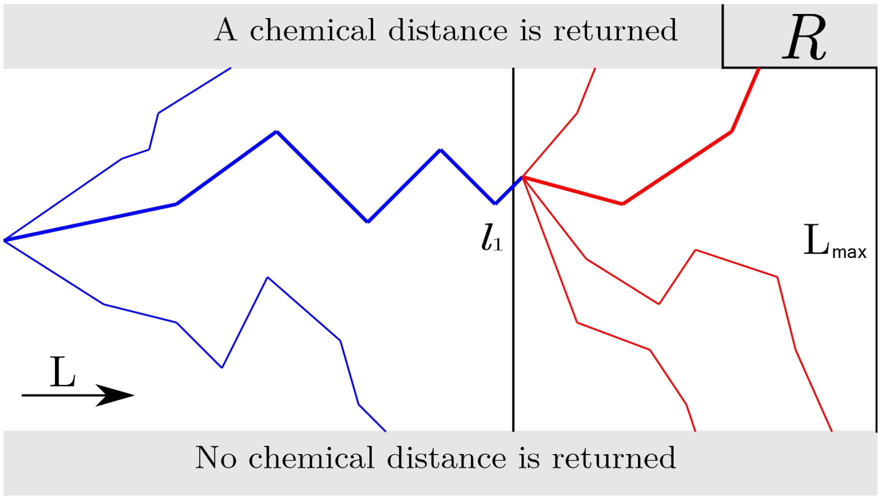
A chemical distance is returned
R
l_1
L_{max}
L
No chemical distance is returned

Figure 3. A schematic illustration of the splitting approach. One path of $(Y_m)_{m\geq 0}$ successfully hits l_1. This path is thicker than the others. It is then split into four separate paths, one of which successfully reaches the rare event set. Again, this path is thicker than the others.

If we only take as our sample the resulting values of X^γ that are bigger than x, it is easy to see that these satisfy

$$\mathbb{P}\left(f\left(Y_{\tau_0}\right) = x \,|\, L_{\tau_1} > l_1, x < f\left(Y_{\tau_0}\right) < \infty\right) = \mathbb{P}\left(X^\gamma = x \,|\, x < X^\gamma < \infty\right).$$

Of course, the use of splitting means that the samples are no longer independent. However, if the splitting factor, s, is not too large relative to the number of paths that hit l_1, this is not an issue in practice.

When we apply splitting in this manner, the expected number of points larger than x is s times larger than it would be using standard Monte Carlo. In other words, we only need to begin with N/s processes in order to get a sample size in the right-tail that would require simulating N processes using standard Monte Carlo. We can repeat this process a number of times by using a sequence of levels $l_1 < l_2 < \cdots < l_K = (x-1)/2$ and splitting a process s times when it crosses a level. Using this approach, we generate roughly s^K times more samples of X^γ conditioned on $\{x < X^\gamma < \infty\}$ than we would have using conventional Monte Carlo. In our setting, the optimal x_{min} is small enough that this is not necessary, but in many other settings multiple levels are required.

The appropriate way to measure the effectiveness of the splitting method is by comparing the work required to generate a sample from $\{X^\gamma : X^\gamma > x\}$ using splitting to the work required using standard Monte Carlo. In the continuum setting, the cost of the splitting, i.e. copying the data structure s times, is considerable. However, for sufficiently large x, there is a substantial computational saving in that far fewer samples must be generated in order to generate s samples of X^γ conditioned on being greater than x. This saving outweighs the cost of copying the data structures and results in a significant speed increase and ability to generate samples from much further in the tail of X^γ. In the numerical results given below, it takes approximately 30% more time to generate a sample using conventional Monte Carlo than it does using the splitting approach. Note that we are not in the classical setting of splitting, where the cost of replicating the relevant stochastic process is negligible. Thus the performance increases we obtain are not comparable to those obtained in that setting. When we performed simulations in the lattice setting, where the data structures are simpler and smaller than in the continuum setting, we achieved much more substantial reductions in computation time. We are confident that research into new and efficient data structures for investigating continuum percolation should lead to significant improvements in the performance of the splitting approach.

5. Numerical Results

The numerical results in the following section were obtained using programs implemented in Java and running in the background on a single PC with 32 GB RAM and an Intel i7-4770K CPU.

5.1. Exponential vs. Polynomial Decay

We performed our simulation not only for $\gamma = \gamma_c$, but also for other choices of γ. The resulting simulations provide strong evidence that polynomial decay of the conditional probability mass function of X^γ, given in (2), occurs only for $\gamma = \gamma_c$. The results are shown in Figure 4.

5.2. Estimates using the Standard Monte Carlo Sample

We carried out a numerical investigation of the critical exponent for $d = 2$. The critical percolation threshold for this model, γ_c, is estimated in [10] to be $\gamma_c \approx 1.198468$; see, [36]

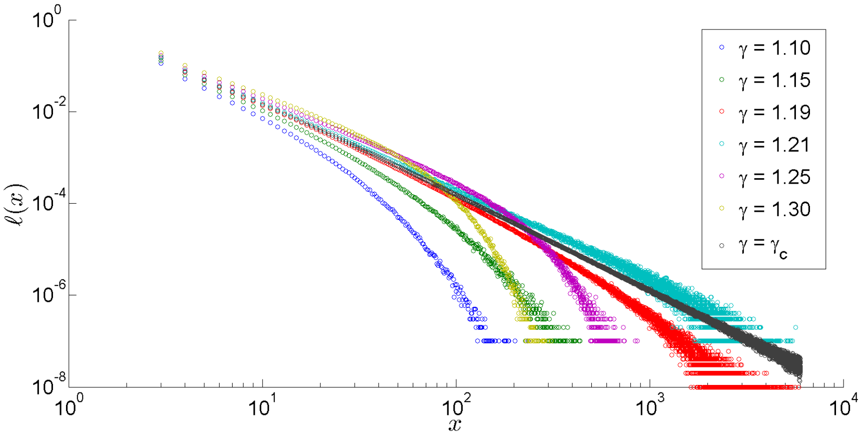

Figure 4. Estimates of $\ell(x) = \mathbb{P}(X^\gamma = x \mid X^\gamma < x_{\mathsf{max}})$ for a variety of different edge lengths.

and [37] for two additional papers where this threshold is estimated.

Using the standard Monte Carlo procedure, we generated a sample of size $N = 1.5 \times 10^9$ with $L_{\mathsf{max}} = 4000$. Generating this sample took approximately 26.7 days computer time. We estimated $\lambda(x_{\mathsf{min}})$ for $x_{\mathsf{min}} = 3, \dots, 7990$. According to the Kolmogorov-Smirnov statistic, given in (3), the optimal value of x_{min} is $x_{\mathsf{min}} = 458$. This gives a rate estimate of $\widehat{\lambda}(458) = 2.1025$.

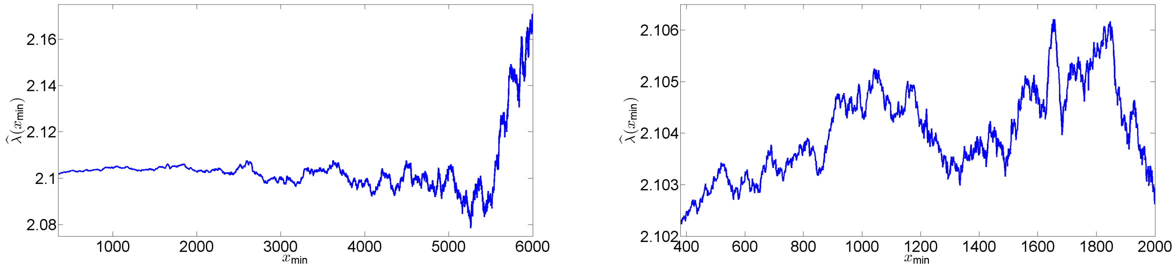

Figure 5. Left: Values of $\widehat{\lambda}(x_{\mathsf{min}})$ estimated using the standard Monte Carlo sample for $x_{\mathsf{min}} = 380, \dots, 6000$. Right: Values of $\widehat{\lambda}(x_{\mathsf{min}})$ estimated using the standard Monte Carlo sample for $x_{\mathsf{min}} = 380, \dots, 2000$.

The estimates of $\lambda(x_{\mathsf{min}})$ for $x_{\mathsf{min}} = 380, \dots, 6000$ are shown on the left of Figure 5. The estimates of $\lambda(x_{\mathsf{min}})$ for $x_{\mathsf{min}} = 380, \dots, 2000$ are shown on the right of Figure 5. The values of $\widehat{\lambda}(x_{\mathsf{min}})$ appear fairly stable for at least the first 2000 values of x_{min}, with the subsequent loss of stability due to the smaller sample sizes used for large values of x_{min}. Closer investigation of values in this region shows that $\widehat{\lambda}(x_{\mathsf{min}})$ fluctuates between 2.102 and 2.106 with the estimate obtained using the Kolmogorov-Smirnov statistic at the lower end of the range. Note that this degree of fluctation shows the sensitivity of the estimator to the choice of x_{min}.

Note that, prior to the simulations with $L_{\mathsf{max}} = 4000$, we carried out simulations using $L_{\mathsf{max}} = 3000$ with a sample size of $N = 10^9$. These simulations gave an optimal value of $x_{\mathsf{min}} = 854$ with $\widehat{\lambda}(854) = 2.1064$. This suggests that there is considerable variability in the estimator and also sensitivity to the cut-off point L_{max}.

5.3. Estimates using the Splitting Sample

We also used splitting to generate samples of X^{γ_c}, carrying out the splitting at the level 400. This produced a sample of X^{γ_c} conditioned on being larger than 800. The splitting point was chosen based on the standard Monte Carlo sample with $L_{\mathsf{max}} = 3000$ and $N = 10^9$ which returned an optimal choice of $x_{\mathsf{min}} = 854$. We calculated a sample of $N = 1.5 \times 10^7$ with a splitting factor of $s = 100$, giving us a sample in the tail of equivalent size to the sample gained using standard Monte Carlo with $N = 1.5 \times 10^9$. This took approximately 20.2 days to generate, which is roughly 75% of the time required by the standard Monte Carlo approach. The optimal value of x_{min}, according to the Kolmogorov-Smirnov statistic, given in (3), is $x_{\mathsf{min}} = 1065$. This gives a rate estimate of $\widehat{\lambda}(1065) = 2.1051$.

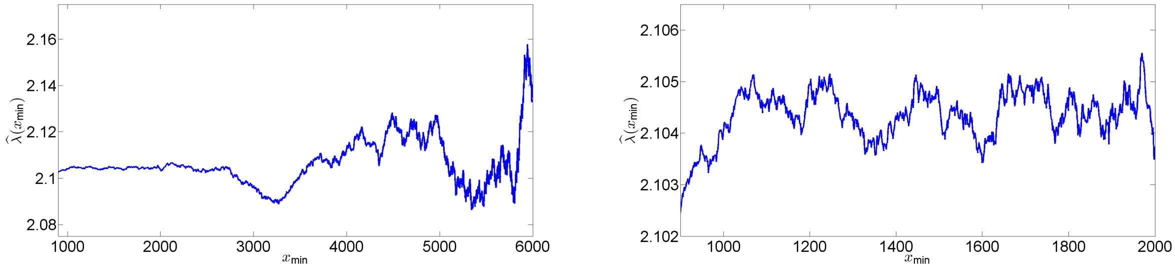

Figure 6. Left: Values of $\widehat{\lambda}(x_{\mathsf{min}})$ estimated using the splitting sample for $x_{\mathsf{min}} = 900, \ldots, 6000$. Right: Values of $\widehat{\lambda}(x_{\mathsf{min}})$ estimated using the splitting sample for $x_{\mathsf{min}} = 900, \ldots, 2000$.

The estimates of $\lambda(x_{\mathsf{min}})$ for $x_{\mathsf{min}} = 900, \ldots, 6000$ are shown on the left of Figure 6 and the estimates of $\lambda(x_{\mathsf{min}})$ for $x_{\mathsf{min}} = 900, \ldots, 2000$ are shown on the right of Figure 6. As in the standard Monte Carlo case, the values of $\widehat{\lambda}(x_{\mathsf{min}})$ appear fairly stable for at least the first 2000 values of x_{min}, with the subsequent loss of stability due to the smaller sample sizes used for large values of x_{min}. Closer investigation of values in this region shows that, at least for $x_{\mathsf{min}} > 1000$, $\widehat{\lambda}(x_{\mathsf{min}})$ fluctuates between about 2.1035 and 2.1055 with the estimate obtained using the Kolmogorov-Smirnov statistic at the higher end of the range.

5.4. Estimates using the Joint Sample

We then carried out an estimate of λ using the combined sample from the standard Monte Carlo and splitting approaches. This sample of X^{γ_c} conditioned on being larger than 800 was equivalent to that obtained using standard Monte Carlo with a sample size $N = 3 \times 10^9$. The optimal value of x_{min}, according to the Kolmogorov-Smirnov statistic, is $x_{\mathsf{min}} = 991$. This gives a rate estimate of $\widehat{\lambda}(991) = 2.1045$.

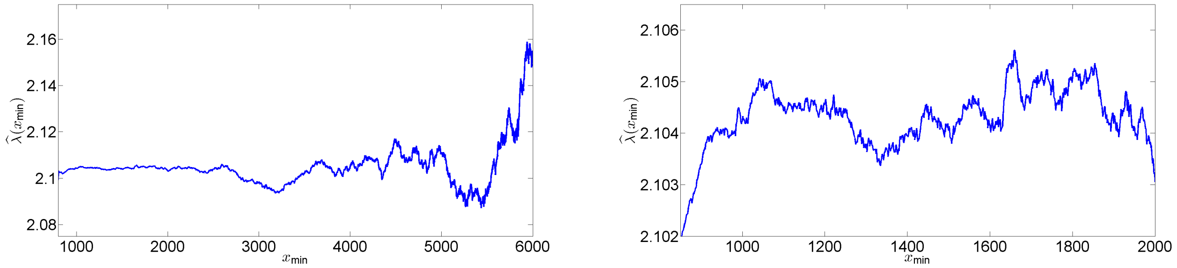

Figure 7. Left: Values of $\widehat{\lambda}(x_{\mathsf{min}})$ estimated using the joint sample for $x_{\mathsf{min}} = 850, \ldots, 6000$. Right: Values of $\widehat{\lambda}(x_{\mathsf{min}})$ estimated using the joint sample for $x_{\mathsf{min}} = 850, \ldots, 2000$.

The estimates of $\lambda(x_{\mathsf{min}})$ for $x_{\mathsf{min}} = 900, \ldots, 6000$ are shown on the left of Figure 7. The estimates of $\lambda(x_{\mathsf{min}})$ for $x_{\mathsf{min}} = 900, \ldots, 2000$ are shown on the right of Figure 7. As in both the standard Monte Carlo and splitting cases, the values of $\widehat{\lambda}(x_{\mathsf{min}})$ appear fairly stable for at least the first 2000 values of x_{min}. Closer investigation of values in this region shows that, as in the splitting case, for $x_{\mathsf{min}} > 1000$, $\widehat{\lambda}(x_{\mathsf{min}})$ fluctuates between about 2.1035 and 2.1055. Here, the estimate obtained using the Kolmogorov-Smirnov statistic is in the middle of the range.

5.5. Summary of Numerical Results

The four estimates we have obtained are $\widehat{\lambda} = 2.1025$ for the standard Monte Carlo sample with $L_{\mathsf{max}} = 4000$ and $N = 1.5 \times 10^9$, $\widehat{\lambda} = 2.1064$ for the standard Monte Carlo sample with $L_{\mathsf{max}} = 3000$ and $N = 10^9$, $\widehat{\lambda} = 2.1051$ for the splitting sample with $L_{\mathsf{max}} = 4000$ and $N = 1.5 \times 10^9$, and $\widehat{\lambda} = 2.1045$ for the joint sample with sample size equivalent to a standard Monte Carlo sample of $N = 3 \times 10^9$. We note that standard errors are not available for these estimators, which are also biased (though not as biased as estimates obtained using least squares). Nevertheless, these estimates broadly agree with the results for lattice percolation given in [5] and [7], as well as the estimate based on the results in [26].

Acknowledgments

This work was supported by the DAAD (German Academic Exchange Service) / Go8 Australia–Germany Joint Research Cooperation Scheme. Dirk Kroese acknowledges the support from the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) under grant number CE14010004.

References

[1] S. Havlin, B. Trus, G. H. Weiss, and D. Ben-Avragam. The chemical distance distribution in percolation clusters. *Journal of Physics A: Mathematical and General*, 18:L247–L249, 1985.

[2] S. Havlin and D. Ben-Avraham. Diffusion in disordered media. *Advances in Physics*, 36:695–798, 1987.

[3] H. J. Herrmann and H. E. Stanley. The fractal dimension of the minimum path in two- and three-dimensional percolation. *Journal of Physics A: Mathematical and General*, 21:L829–L833, 1988.

[4] N. V. Dokholyan, Y. Lee, S. V. Buldyrev, S. Havlin, P. R. King, and H. E. Stanley. Scaling of the distribution of shortest paths in percolation. *Journal of Statistical Physics*, 93(3-4):603–613, 1998.

[5] P. Grassberger. Pair connectedness and shortest-path scaling in critical percolation. *Journal of Physics A: Mathematical and General*, 32(35):6233–6238, 1999.

[6] M. Porto, S. Havlin, H. E. Roman, and A. Bune. Probability distribution of the shortest path on the percolation cluster, its backbone and skeleton. *Physical Review E*, 58(5):R5205–R5208, 1998.

[7] R. M. Ziff. Exact critical exponent for the shortest-path scaling function in percolation. *Journal of Physics A: Mathematical and General*, 32(43):457–459, 1999.

[8] P. Grassberger. Numerical studies of critical percolation in three dimension. *Journal of Physics A: Mathematical and General*, 25:5867–5888, 1992.

[9] E. T. Gawlinski and H. E. Stanley. Continuum percolation in two dimensions: Monte Carlo tests of scaling and universality for non-interacting discs. *Journal of Physics A: Mathematical and General*, 14:L291–L299, 1981.

[10] S. Mertens and C. Moore. Continuum percolation thresholds in two dimensions. *Physical Review E*, 86(6):061109, 2012.

[11] S. Torquato. *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. Springer, New York, 2001.

[12] P. L. Leath. Cluster size and boundary distribution near percolation threshold. *Physical Review B*, 14(11):5046–5055, 1976.

[13] Z. Alexandrowicz. Critically branched chains and percolation clusters. *Physics Letters*, 80A(4):284–286, 1980.

[14] D. A. Adams, L. M Sander, and R. M. Ziff. The barrier method: A technique for calculating very long transition times. *The Journal of Chemical Physics*, 133, 2010.

[15] D. A. Adams. *Algorithms for Measuring Extremely Rare Events in Statistical Physics*. PhD thesis, The University of Michigan, 2011.

[16] P. Grassberger. Pruned-enriched Rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000. *Physical Review E*, 56(3):3682–3693, 1997.

[17] P. Grassberger. Sequential Monte Carlo methods for protein folding. In *NIC Symposium 2004*, Juelich, Germany, 2004. `http://arxiv.org/abs/cond-mat/0408571`.

[18] C. Hirsch, T. Brereton, and V. Schmidt. Percolation and convergence properties of graphs related to minimal spanning forests. Preprint (submitted), 2014.

[19] R. Lyons and Y. Peres. *Probability on Trees and Networks*. Cambridge University Press, in preparation.

[20] C.-L. Yao, G. Chen, and T.-D. Guo. Large deviations for the graph distance in supercritical continuum percolation. *Journal of Applied Probability*, 48(1):154–172, 2011.

[21] P. Balister, A. Sarkar, and B. Bollobás. Percolation, connectivity, coverage and colouring of random geometric graphs. In B. Bollobás, R. Kozma, and D. Miklós, editors, *Handbook of Large-Scale Random Networks*, pages 117–142. Springer, Berlin, 2008.

[22] S. Torquato, J. D. Beasley, and Y. C. Chiew. Two point cluster function for continuum percolation. *The Journal of Chemical Physics*, 88(10):6540–6547, 1988.

[23] S. B. Lee and S. Torquato. Pair connectedness and mean cluster size for continuum percolation models. *The Journal of Chemical Physics*, 89(10):6427–6433, 1988.

[24] M. D. Rintoul and S. Torquato. Precise determination of the critical threshold and exponents in a three-dimensional continuum percolation model. *Journal of Physics A: Mathematical and General*, 30:L585–L592, 1997.

[25] S. N. Chiu, D. Stoyan, W. S. Kendall, and J. Mecke. *Stochastic Geometry and its Applications*. J. Wiley & Sons, Chichester, third edition, 2013.

[26] Z. Zhou, J. Yang, Y. Deng, and R. M. Ziff. Shortest-path fractal dimension for percolation in two and three dimensions. *Physical Review E*, 86:061101, 2012.

[27] C. D. Lorenz and R. M. Ziff. Precise determination of the critical percolation threshold for the three-dimensional "Swiss cheese" model using a growth algorithm. *Journal of Chemical Physics*, 114(8):3659–3661, 2001.

[28] D. Jungnickel. *Graphs, Networks and Algorithms*. Springer, Berlin, 1999.

[29] G. Paul, R. M. Ziff, and H. E. Stanley. Percolation threshold, Fisher exponent, and shortest path exponent for four and five dimensions. *Physical Review E*, 64:026115–1–026115–8, 2001.

[30] H. Bauke. Parameter estimation for power-law distributions by maximum likelihood methods. *European Physical Journal B*, 58(2):167–173, 2007.

[31] M. L. Goldstein, S. A. Morris, and G. G. Yen. Problems with fitting to the power-law distribution. *European Physical Journal B*, 41(2):255–258, 2004.

[32] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[33] H. Kahn and T. E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards Applied Mathematics Series*, 12:27–30, 1951.

[34] M. J. J. Garvels. *The Splitting Method in Rare Event Simulation*. PhD thesis, University of Twente, 2000.

[35] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. Multilevel splitting for estimating rare event probabilities. *Operations Research*, 47(4):585–600, 1999.

[36] J. A. Quintanilla and R. M. Ziff. Near symmetry of percolation thresholds of fully penetrable disks with two different radii. *Physical Review E*, 76(5):051115, 2007.

[37] J. A. Quintanilla, S. Torquato, and R. M. Ziff. Efficient measurement of the percolation threshold for fully penetrable discs. *Journal of Physics A: Mathematical and General*, 33:L399–L407, 2000.