Geometric properties of Poisson matchings

Jonges, J.

J. Jonges

Geometric properties of Poisson matchings

Bachelorscriptie, 30 juni 2012

Scriptiebegeleider: Dr. M.O. Heydenreich

Mathematisch Instituut, Universiteit Leiden

Contents

1 Introduction

In this thesis we discuss Poisson matchings, which can be seen as 'random graphs' with an infinite set of random vertices $U \subset \mathbb{R}$. The set 'U' is the set of 'arrival times' in a Poisson process. It is our goal to gain more insight in a specific matching type called the 'stable multi-matching', which has some extra nice properties. We are mainly concerned with the question whether infinite components exist in these random graphs . In this paper we will study this question for the stable multi-matching with a specific degree distribution. Showing the existence of an infinite component in this matching is still an open question, but an overwhelming amount of simulation results seems to suggest a positive answer. We will also add our simulation results in support of the conjecture that such an infinite component does exist (with probability 1). We will mainly use and study the work of Alexander Holroyd and for many results will refer to his papers on this subject.

Also, we try to give the reader some impression of the arguments involved in proving statements about geometric properties of the Poisson matchings. Intuitively some of these problems can seem misleadingly simple, but often there's many subtleties and difficult mathematics involved in proving statements about geometric properties. So, don't be misleaded by easy questions with difficult answers!

2 What is Poisson Process ?

In this section we give a small introduction to Poisson processes, and give some of its basic features. For more extensive studies in Poisson processes we refer to [4].

Definition 1. A *point process* $\{C(t), t \in \mathbb{R}\}$ is a integer-valued stochastic process satisfying the following properties:

(1) $C(t) \geq 0$ for all t,

(2) $C(t)$ is an integer,

(3) $C(t)$ is non-decreasing in t, i.e., if $s \leq t$ then $C(s) \leq C(t)$.

If $s < t$ then $C(t) - C(s)$ denotes the number of events that occured in the time-interval $(s, t]$.

The most important examples of counting processes are Renewal processes, and Poisson processes. A Poisson process is a special case of a Renewal process.

Definition 2. A *Poisson process* $\{N(t), t \in \mathbb{R}\}$ is a point process with the following additional properties:

(i) $N(0) = 0$,

(ii) The process has independant increments,

(iii) $\mathbb{P}(N(t+s) - N(s) = k) = \frac{e^{-\lambda t}(\lambda t)^k}{k!}$, $k = 0, 1, 2, \cdots$.

A stochastic process has independant increments if the number of events in disjoint intervals are independant. This means that $N(t)-N(s)$ and $N(v)-N(u)$ are independant variables for disjoint intervals $(s, t]$ and $(u, v]$. Property (iii) states that the number of events in any interval with length t follows a Poisson distribution with mean λt. Here λt is the expected number of events in an interval of length t.
It follows that the process has stationary increments, meaning that the number of events in any interval $(s, s + t]$ only depends on the length t, and not on the particular position in time s. The parameter $\lambda > 0$ is called the intensity-rate of the process. In the case that that λ changes over time, that is $\lambda = \lambda(t)$ is a non-constant function of t, the process is called a non-homogenous Poisson process. We will only deal with homogenous Poisson processes here.

Each realization of the process $\{N(t), t \in \mathbb{R}\}$ is an integer valued step-function (Fig. 1). Let $\mathcal{P}$ denote a realization of a Poisson process. We are interested in the set of jump times

$$[\mathcal{P}] := \left\{ t \geq 0 : \lim_{\epsilon \uparrow 0} N_{t-\epsilon} \neq N_t \right\},$$

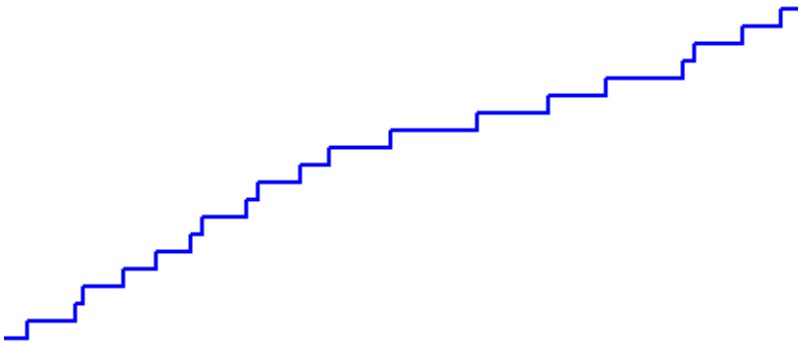

Figure 1: sample process $N(t)$

which we call the **support** of $\mathcal{P}$. The set of jump times in $(0,t)$ follows a uniform distribution. To see this, note that any subinterval of $(0,t)$ of equal length has the same probability to contain the event due to property (ii). Formally, we show this by conditioning on one event in the interval $(0,t]$, and proving that the time upto the occurance follows a uniform distribution on $(0,t)$.

Lemma 3. *Let $\mathcal{P}$ a Poisson-process with finite intensity λ. Then the set of jump times in $(0,t)$ follows a uniform distribution.*

Proof. Denote by T_1 the time upto the first event. Let $0 \leq s \leq t$, then we have:

$$\begin{aligned}
\mathbb{P}(T_1 \leq s | N(t) = 1) &= \frac{\mathbb{P}(\text{one event in } (0,s], \text{no event in } (s,t])}{\mathbb{P}(N(t)=1)} \\
&= \frac{\mathbb{P}(N(s)=1)\mathbb{P}(N(t-s)=0)}{\mathbb{P}(N(t)=1)} \\
&= \frac{e^{-\lambda s}(\lambda s)\cdot e^{-\lambda(t-s)}}{e^{-\lambda t}(\lambda t)} \\
&= \frac{s}{t}.
\end{aligned}$$

We see that T_1 is indeed uniformly distributed on $(0,t)$. This property is used for computer simulations that will be discussed later.

To avoid ambiguity we will now call the events from a Poisson process 'occurances'. We like to add that a Poisson process can be characterized in many but equivalent ways. An alternative definition is the following. Denote by W the inter-arrival time, which is the elapsed time between 2 consecutive arrivals. Keep conditions (i),(ii) and replace (iii) by 'W obeys an Exp(λ) distribution with mean $\frac{1}{\lambda}$'.
We show these sets of conditions to be equivalent.

Theorem 4. *Impose conditions (i),(ii) on a counting process, then conditions (iii) and $W \sim Exp(\lambda)$ are equivalent.*

Proof. Consider a counting process $\{N(t), t \geq 0\}$ with conditions (i),(ii),(iii). Let T_k be the time uptil the k-th arrival for $k = 0, 1, 2, \cdots$, and $W_k = T_k - T_{k-1}$ for $k = 1, 2, \cdots$. Then clearly the number of arrivals before some fixed time t is less then k if and only if the waiting time uptil the k-th arrival is more then t. Therefore the event $[T_k > t]$ happens if and only if the event $[N(t) < k]$ happens, and consequently the probabilities are the same. That is

$$\mathbb{P}(T_k > t) = \mathbb{P}(N(t) < k).$$

Note that the set interarrival-times $\{W_k\}_{k \geq 1}$ follow the same distribution. In particular consider $W_1 = T_1 - T_0$, which is the waiting time until the first arrival. We have

$$\mathbb{P}(W_1 > t) = \mathbb{P}(N(t) = 0) = \mathbb{P}(N(t) - N(0) = 0) = \frac{e^{-\lambda t}(\lambda t)^0}{0!} = e^{-\lambda t}.$$

We see that W is indeed exponentially distributed with mean $\frac{1}{\lambda}$. Therefore the 2 different sets of conditions are equivalent, and both characterise a Poisson process. □

Another interesting fact is that no 2 arrivals may take place on the same point in time. This is an immediate consequence from the fact that W follows an Exp(λ) distribution. Note that $\mathbb{P}(W > 0) = e^{-\lambda \cdot 0} = 1$, therefore the inter-arrival time is strictly bigger then 0 with probability 1.

3 Poisson matching

In this thesis we discuss Poisson matchings which are constructed graphs with vertex set $[\mathcal{P}]$. Connected points are here viewed as 'partners' or 'matched pairs'. Our primary focuss will be on the cluster-forming, and component structure resulting from these matching processes, and we will study geometric properties of the resulting graphs. We present some major results by A. Holroyd, and either refer to proofs, or study some in more detail. We then move on to the concept of stable matching. First, we give a formal definition of the objects we will work with.

Let $\mathcal{P}$ be a Poisson process with intensity-rate 1. Let μ be a measure on the strictly positive integers. We consider the family of $\mathbb{N}$-valued random variables $\{D_x : x \in [\mathcal{P}]\}$ with joint law μ. Interpret D_x as the number of stubs of the vertex x. In this thesis we restrict our attention to the case $\mu(\{1,2\}) = 1$, in particular $D_x = 1$ or $D_x = 2$ for all x (i.e. $\mu(\{1\}) = 1$ or $\mu(\{2\}) = 1$ respectively). These cases are of special interest, as they result in planar graphs for the stable multi-matching on $\mathbb{R}$ (See section 6).
A **matching scheme** on $[\mathcal{P}]$ is a point process $\mathcal{M}$ on the space of unordered pairs of points in $\mathbb{R}$, with the property that almost surely for every pair $\{x, y\} \in [\mathcal{M}]$ we have $x, y \in [\mathcal{P}]$. We say that the matching scheme $\mathcal{M}$ is **translation-invariant** if the law of the joint process $(\mathcal{P}, \mathcal{M})$ is invariant under translations of $\mathbb{R}$. We are only interested in the case that $\mathcal{M}$ is a perfect matching, that is, every vertex x in the resulting graph $G = G([\mathcal{P}], [\mathcal{M}])$ has degree D_x. Also, the matchings under consideration are simple, i.e., G is a simple graph (meaning G almost surely has neither self-loops nor multiple edges).
The idea of Poisson matching can be generalised to Poisson processes in arbitrary dimension $d \geq 1$. We will restrict our attention to $d = 1$. Furthermore, we make a distinction between 2 kinds of matching schemes; if $\mathcal{M} = f(\mathcal{P})$ for some deterministic function f, then $\mathcal{M}$ is called a **factor** matching, and otherwise $\mathcal{M}$ is referred to as **randomized**. The focus of the present work is on factor matchings.

4 The one stub case

Let $\mathcal{P}$ be a Poisson process with intensity $\lambda = 1$, and assign to every $x \in [\mathcal{P}]$ one stub, i.e. $D \equiv 1$. We will give an example of a matching scheme $\mathcal{M}$ that is translation-invariant, and another matching which is not. For a factor matching translation-invariant means that applying $\mathcal{M}$ after translating the points in $[\mathcal{P}]$ by some $\theta \in \mathbb{R}$, results in the same graph as first applying $\mathcal{M}$ and then translating by θ. This may seem like a very natural property, but a simple matching for which this is not the case is the following.

Example 1.
Match the first point in $[\mathcal{P}]$ left from the origin to the first point right from the origin, and match the rest of the matching adjacent. By adjacent we mean that for any edge $\{x, y\} \in [\mathcal{M}]$ there's no point of $[\mathcal{P}]$ in (x, y). Translate the Poisson points in figure 2 by some $\theta > 0$ such that the first point left from zero becomes the first right point from zero. Apply the matching described above. Clearly we find a different graph from the one in figure 2 after translation by θ. Therefore the matching is not translation-invariant.

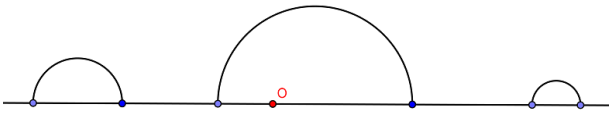

Figure 2:

In the case of $D \equiv 1$ it's not obvious that a translation-invariant factor matching exists. The stable (Gale-Shapely) matching is one example of a translation invariant matching for any degree distribution μ. We will come back to this later.

Example 2.
For $D \equiv 2$ adjacent matchings are translation-invariant. Note that adjacent matchings only exist for degree distributions with $\mu(\{1, 2\}) = 1$ (one or 2 stubs per vertex). There exists exactly 1 adjacent matching in the case $D \equiv 2$ which is obtained by either connecting every point to its closest left neighbor, or connecting every point to its closest right neighbor. Again, it's an easy verification that the matching is independent of any translation θ.

5 Some results on Poisson matching

The first result we present is for translation-invariant factor matchings with degree distribution $\mu(\{1\}) = 1$. (one stub per vertex). Let $\mathbb{P}$ be the probability measure governing $(\mathcal{P}, \mathcal{M})$ and $\mathbb{E}$ be the associated expectation operator. We are interested in the typical distance between matched pairs. Denote by $\mathcal{M}(x)$ the partner of x for any $x \in [\mathcal{P}]$.

Assume without loss of generality that $\mathcal{P}$ has intensity 1. For $r \in [0, \infty)$ consider the following quantity

$$F(r) = \mathbb{E}\#\{x \in [\mathcal{P}] \cap [0, 1) : |x - \mathcal{M}(x)| \leq r\}.$$

Lemma 5. *F defines a distribution function.*

Proof. Clearly $F(r) \to \mathbb{E}\#\{x \in [\mathcal{P}] \cap [0, 1)\} = 1$ as $r \to \infty$. Furthermore $F(r) = 0$ for $r \leq 0$. Let $0 < r_1 < r_2$. For any $x \in [\mathcal{P}] \cap [0, 1)$, if $|x - \mathcal{M}(x)| < r_1$

then $|x - \mathcal{M}(x)| < r_2$ implying $F(r_1) \leq F(r_2)$, and showing that F is monotone non-decreasing. $\square$

By lemma 5 we may introduce a random variable X with

$$\mathbb{P}^*(X \leq r) = F(r).$$

We can think of X as the typical (expected) distance between matched pairs.

Theorem 6. *Let $\mathcal{P}$ be a Poisson process of intensity 1. Any translation-invariant matching scheme satisfies $\mathbb{E}^*[X] = \infty$.*

This is a remarkable result, as it is impossible to make translation-invariant matching schemes with a bound on the expected edge length. Leaving out translation-invariance, it becomes quite easy to make matching schemes with finite expected edge length. An example is the matching scheme described in example 1. The expected edge length is here equal to the expected distance between 2 Poisson points, which is (as described in the introduction) the expected waiting time $E[W] = 1/\lambda = 1 < \infty$. We shall see, as in the proof of Theorem 6, that translation-invariance is quite a strong condition.

We postpone the proof of Theorem 6 and give a lemma that states we can not achieve an adjacent matching without randomization. One may find this not very surprising. In a factor matching, to achieve an adjacent matching one has to make one choice for an edge $\{x, y\}$ for 2 adjacent points $x, y \in [\mathcal{P}]$, then the rest of the matching is determined. One can imagine this is hard (actually impossible) to do in a translation-invariant way. In example 1 we saw that making the choice for the first edge (first points left and right from zero), directly sacrificed translation-invariance. Still, there might exist clever and sophisticated ways to make this choice. The following lemma proves that this is not the case; making any choice for an edge with 2 adjacent points by some deterministic function of $[\mathcal{P}]$ will be at the expense of the translation-invariance condition. In what follows we say that an edge $\{x, y\}$ of $\mathcal{M}$ **crosses** a site $z \in \mathbb{R}$ if $z \in (x, y)$.

Lemma 7. *Let $\mathcal{P}$ be a homogenous Poisson process on $\mathbb{R}$. There does not exist a translation-invariant factor matching scheme where the matching is almost surely adjacent.*

Proof. Suppose on the contrary that $\mathcal{M}$ is such matching scheme. Write $\mathcal{F}_S$ for the σ-algebra generated by the restriction of $\mathcal{P}$ to $S \subset \mathbb{R}$, and consider the event

$$A := \{ \text{ 0 is crossed by some edge } \}.$$

Since A is contained in $\mathcal{F}_{\mathbb{R}}$, we have that for every $\epsilon > 0$ exists $r = r(\epsilon) < \infty$ and an event $A_\epsilon \in \mathcal{F}_{[-r,r]}$ such that $\mathbb{P}(A \Delta A_\epsilon) < \epsilon$. This is a consequence of Levy's zero-one Law. (See Appendix). By translation-invariance of $\mathcal{M}$, we can find

$B_\epsilon \in \mathcal{F}_{[-2r,0]}$ such that $\mathbb{P}(\{-r \text{ is crossed by some edge }\} \Delta B_\epsilon) < \epsilon$. Furthermore, an adjacent matching is fully determined by whether some deterministic point $x \in \mathbb{R}$ is crossed or not. Hence there exists $L_\epsilon \in \mathcal{F}_{[-2r,0]} \subset \mathcal{F}_{(-\infty,0]}$ with $\mathbb{P}(A \Delta L_\epsilon) < \epsilon$. Since this is true for all $\epsilon > 0$, we have $A \in \overline{\mathcal{F}_{(-\infty,0]}}$, where the bar denotes the completion under $\mathbb{P}$. The same arguments can be used to show that $A \in \overline{\mathcal{F}_{[0,\infty)}}$. Now, since A is contained in 2 independant σ-algebra's, A must be independant of itself. That is $\mathbb{P}(A) = \mathbb{P}(A \cap A) = \mathbb{P}(A)^2$, and it follows that A has probability 1 or 0. But then we arrive at a contradiction since neither of the 2 matching schemes is translation-invariant. □

In the following lemma we use the fact that Poisson processes are **Ergodic**. Define a shift-operator $T_s : \mathcal{P} \to \mathcal{P}$ by

$$T_s\nu(t) = \nu(s+t) - \nu(s),\ s \geq 0.$$

Let $T_s^{-1}B = \{\nu \in \mathcal{P} : T_s(\nu) \in B\}$. A set B is said to be (translation) **invariant** if $T_s^{-1}(B) = B$ for all s. A stationary point process is called Ergodic if $\mathbb{P}(B) = 0$ or 1 for all invariant sets. More on ergodicity of the Poisson process can be found in [4].

Lemma 8. *Let $\mathcal{P}$ be a Poisson process of intensity 1, and $\mathcal{M}$ be a translation-invariant factor matching scheme. The number of edges crossing 0 is infinite.*

Proof. Let $\mathcal{P}$, and $\mathcal{M}$ be as given. Define

$$B = [0 \text{ is crossed by a finite number of edges}].$$

Our aim is to prove $\mathbb{P}(B) = 0$. Suppose by contradiction $\mathbb{P}(B) > 0$. On the event that 0 is crossed by a finite number of edges, a.s. the same is true for any other $r \in \mathbb{R}$, since the difference of the number of edges crossing r and the number of edges crossing 0 is at most the number of Poisson points between 0 and r (which is finite). This shows that B is an invariant set; by translation-invariance of the matching, for any point configuration $\nu \in \mathcal{P}$ we have $\nu \in B$ if and only if $T_s(\nu) \in B$. It follows by ergodicity that B either has probability 1, or 0. Since we assumed $\mathbb{P}(B) > 0$ we must have $\mathbb{P}(B) = 1$. Then define a new matching scheme $\mathcal{M}'$ as follows: Match 2 adjacent points x, and y if and only if the points $r \in (x, y)$ are crossed an odd number of times in the matching $\mathcal{M}$. The maching $\mathcal{M}'$ is an (translation-invariant) adjacent factor matching contradicting lemma 7. □

We almost have the tools to prove Theorem 6. The actual proof of Theorem 6 makes use of Palm measures. In the Palm process $\mathcal{P}^*$ the origin is added uniformly to the support of $\mathcal{P}$. If the typical edge length X is defined as the distance from the origin to its partner

$$X := |\mathcal{M}^*(0)|,$$

it turns out this yields a measure with the same CDF as defined above. For more on this topic see [5].

proof of Theorem 6
We have the following inequalities

$$\begin{aligned}
\mathbb{E}\#\{\text{edges crossing } 0\} &\leq \mathbb{E}\#\{x \in [\mathcal{P}] \cap [0,\infty) : |x - \mathcal{M}(x)| > |x|\} \\
&= \sum_{r=0}^{\infty} \mathbb{E}\#\{x \in [\mathcal{P}] \cap [r, r+1) : |x - \mathcal{M}(x)| > |x|\} \\
&\leq \sum_{r=0}^{\infty} \mathbb{E}\#\{x \in [\mathcal{P}] \cap [r, r+1) : |x - \mathcal{M}(x)| > r\} \\
&= \sum_{r=0}^{\infty} \mathbb{E}\#\{x \in [\mathcal{P}] \cap [0, 1) : |x - \mathcal{M}(x)| > r\} \\
&= \sum_{r=0}^{\infty} (\mathbb{E}\#\{x \in [\mathcal{P}] \cap [0,1)\} - \mathbb{E}\#\{x \in [\mathcal{P}] \cap [0,1) : |x - \mathcal{M}(x)| \leq r\}) \\
&= \sum_{r=0}^{\infty} (1 - \mathbb{P}^*(X \leq r)) \\
&= \sum_{r=0}^{\infty} \mathbb{P}^*(X > r) \\
&= 1 + \sum_{r=1}^{\infty} \mathbb{P}^*(X > r) \\
&\leq 1 + \int_0^{\infty} \mathbb{P}^*(X > r) dr \\
&= 1 + \mathbb{E}^*[X].
\end{aligned}$$

Therefore Lemma 8 implies $\mathbb{E}^*[X] = \infty$. Note that we used translation-invariance of $\mathcal{M}$ in step 4. In the last inequality we used that X is a continuous stochastic variable on $[0, \infty)$.

6 Stable Matching

In this section we introduce a particular natural kind of matching scheme, called the **stable multi-matching**. The concept of **stable matching** goes back to the stable marriage problem introduced by Gale and Shapely. We start by defining stable multi-matching, and give an algorithm yielding the unique stable multi-matching. Also, we collect relevant results for the case $\mu(\{1,2\}) = 1$.

Definition 9. Let $\mathcal{P}$ be a Poisson process of intensity 1. A matching $\mathcal{M}$ is called stable multi-matching on $\mathbb{R}$, if a.s. for any 2 distinct points $x, y \in [\mathcal{P}]$ we either have an edge $\{x, y\}$, or x and y have no incident edges longer then $|x-y|$.

Definition 10. Let $U \subset \mathbb{R}^d$. A sequence $\{x_i\} \subset U$ is called a descending chain if $|x_i - x_{i-1}|$ is strictly decreasing.

It is proven in [1] that $[\mathcal{P}]$ can not contain descending chains. This property is relevant for showing the existence of the stable multi-matching on $\mathbb{R}$.

In the set $[\mathcal{P}]$ call 2 distinct points $x, y \in [\mathcal{P}]$ **mutually closest** if x is the closest point to y in $[\mathcal{P}] \setminus \{y\}$, and vice versa. For a Poisson process $\mathcal{P}$ on $\mathbb{R}$ and any degree distribution μ, the following algorithm exhausts all stubs, and yields the unique stable multi-matching in the sense of definition 9.

Gale-Shapely algorithm

Step 1. Consider the set $[\mathcal{P}]$ of all points. An edge is created between each mutually closest pair in this set, and one stub is removed from each of these points.

Step 2. Consider the set of points that still at least have one stub after step 1. Two points are called compatible if no edge was created between them in step 1. An edge is created between each two compatible mutually closest pair in this set, and one stub is removed from each of these points.

$\vdots$

Step n. Consider the set of points that still at least have one stub. Two such points are called compatible if no edge has been created between them. An edge is created between each compatible mutually closest pair in this set, and one stub is removed from each of these points.

$\vdots$

It is clear from the from the steps in the algorithm, the limiting graph will not contain any loops or multiple edges. The process is also clearly translation-invariant. We will not deal with all the details here, but a full proof for existence and uniqueness of the stable multi-matching can be found in [2].

We shall give an argument as to why the limiting graph is perfect. This is an argument by contradiction. Suppose that the limiting graph is not perfect, and consider the set of vertices $[\mathcal{P}']$ with at least one unused stub on them after the above matching procedure is completed. The set $[\mathcal{P}']$ a.s. contains no points, or infinitely many points. In the first case the matching was already perfect. To rule out the second case, call 2 points in $[\mathcal{P}']$ compatible if they don't have an edge between them in the configuration obtained from the matching procedure. Then create a directed graph G' with $[\mathcal{P}']$ as vertex set, by connecting every point to its nearest compatible point. Clearly no cycles with more then 2 points exist in G'. If $C = (x_1, x_2, \cdots, x_n, x_1)$ with $x_1 < x_2 < \cdots < x_n$ was such a cycle, then either x_1 has x_n as nearest compatible point, or x_n has x_1 as nearest compatible point which is clearly a contradiction. A cycle $C = (x, y)$ corresponds to 2 mutually closest compatible points, which can not be present in G' as the edge (x, y) would have been created at some stage in the algorithm. But then $[\mathcal{P}']$ contains an infinite descending chain, which is impossible.

The following results relate the existence of an infinite component to a certain property concerning edge lengths. Again, the Palm process $\mathcal{P}^*$ is considered here. We state these results without proof. Let M_x denote the length of the longest edge incident to $x \in [\mathcal{P}]$. We say that x **desires** a site $y \in \mathbb{R}$ if $|y - x| < M_x$, and write N for the number of points that desire the origin.

Theorem 11. *For a Poisson process on $\mathbb{R}$, consider the stable multi-matching.*

(i) For any degree distribution, if there is no infinite component, then $N = \infty$ almost surely.

(ii) If $\mu(\{1, 2\}) = 1$, and there is an infinite component, then $N < \infty$ almost surely.

This means that for degree distributions with $\mu(\{1, 2\}) = 1$, the existence of an infinite component is hence equivalent to $N < \infty$. Furthermore, assuming the existence of an infinite component, as $N < \infty$ we may expect only a finite number of edges crossing 0. The proof of Theorem 11, and Lemma 12 can be found in [3]. Let $M = M_0$.

Lemma 12. *For any translation-invariant matching scheme, we have that $\mathbb{E}^*[M] < \infty$ if and only iff $\mathbb{E}[N] < \infty$.*

Combining these two results, it follows that the existence of an infinite component in the stable multi-matching with $D \equiv 2$ implies $\mathbb{E}^*[M] < \infty$. Therefore the expected edge length is finite. We will come back to this later.

Definition 13. Edges (a, b) and (c, d) are **crossing** if $a < c < b < d$.

We now show that any stable multi-matching with $\mu(\{1, 2\}) = 1$ is a planar graph. That is, we have no crossing edges. This is an immediate consequence of

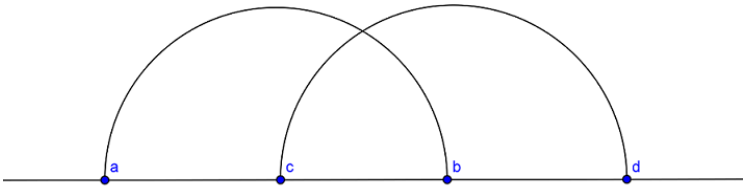

Figure 3: unstable

the stability property (definition 9). We shall illustrate this with the following picture.

Observe that edges (a, b) and (c, d) give rise to edges $(a, c), (c, b)$ and (b, d) (according to the stability property), which means that vertices c, b have degree (at least!) 3, which is impossible. This shows that the situation in figure 3 can not occur in any stable multi-matching with $\mu(\{1, 2\}) = 1$.

The following theorem states that we can have at most one infinite component in the stable multi-matching.

Theorem 14. *For any Poisson process on $\mathbb{R}$ and any degree distribution μ, there is at most one infinite component in the stable multi-matching.*

This result is proven using the mass-transport principle, to show that an infinite component in any translation-invariant matching scheme must be unbounded in 2 directions. Furthermore, 2 distinct components can not have crossing edges. This is also easily seen from figure 9. Since the edge (c, b) must be present we see that crossing edges must be part of the same component. This shows there can be at most one infinite component, since 2 distinct infinite components would necessarily have crossing edges.

In the next result we show that if we have percolation in the stable multi-matching on $\mathbb{R}$, there must be a strictly positive probability p^* to be part of the infinite component. That is

$$\begin{aligned} p^* &:= \mathbb{P}^*(\ 0 \text{ is in the infinite cluster}) \\ &= \mathbb{E}\#\left\{x \in [0,1) \cap [\mathcal{P}] : x \text{ is in the infinite cluster}\right\}. \\ &> 0. \end{aligned}$$

This follows from the bound on the expected edge length (Lemma 11). By boundedness of the expected edge length, we may also obtain a bound on the expected number of Poisson points under the edges of the infinite component. That is, for any edge $\{x, y\}$ the number of Poisson points between x, and y must be smaller then some finite number N. Therefore the probability that a given vertex belongs to the infinite component is bigger then $1/N$. Let $\mathcal{C}_\infty$ denote the infinite component. We will prove last observation more rigorously in next theorem.

Theorem 15. *For any stable multi-matching on $\mathbb{R}$, if there exists an infinite component, then $\mathbb{P}^*(0 \in \mathcal{C}_\infty) > 0$.*

Proof. Assume there exists an infinite component $\mathcal{C}_\infty$. We will show that this implies the following

$$\frac{|\mathcal{C}_\infty \cap (0, N]|}{N} \to p > 0 \text{ a.s. for } N \to \infty.$$

Define $X_i := |\mathcal{C}_\infty \cap (i-1, i]|$, a random variable that counts the total number of points of the infinite cluster in the interval $(i-1, i]$. Then we have

$$\frac{|\mathcal{C}_\infty \cap (0, N]|}{N} = \frac{1}{N} \sum_{i=1}^{N} X_i \to \mathbb{E}[X_1] \text{ a.s. },$$

according to Birkhoff's Ergodic Theorem (see Appendix). Also, we have

$$\mathbb{E}[X_1] \geq \mathbb{E}[\mathbb{I}_{\{X_1 > 0\}}] = \mathbb{P}(X_1 > 0).$$

In case of percolation, we have an infinite component and the last probability must be strictly positive since

$$\begin{aligned} 0 <& \mathbb{P}(\text{ an infinite component exists }) \\ &= \mathbb{P}\left(\bigcup_{i \in \mathbb{Z}} \{X_i > 0\}\right) \\ &\leq \sum_{i \in \mathbb{Z}} \mathbb{P}(X_i > 0). \end{aligned}$$

Since all the probabilities $\mathbb{P}(X_i > 0)$ are equal (because of translation-invariance) they must all be greater then 0. In particular $\mathbb{P}(X_1 > 0) > 0$. □

7 Stable multi-matching $D \equiv 2$

Let $\mathcal{P}$ a Poisson process of finite intensity with $D_x = 2$ for all $x \in [\mathcal{P}]$ and apply the Gale-Shapely matching. In an effort to understand the component structure that arises, we try to answer several questions. How do local structures look like? Can we make predictions about component size? But most importantly, does there exist an infinite component? Although the answer to the last question seems affirmative, which is heavily supported by computer simulations, a rigorous proof has not yet been found. We will first give some extra insight into this problem, and give a statistical proof of this fact later. But first, what do we expect to see?

Recall that the resulting matching is a planar graph. This directly gives a lot of insight into its behaviour. For instance, since we have no crossing edges, all finite components must be of the form figure 4 (consisting of at least 3 points). We already have quite a clear picture now of what we can expect to see in the

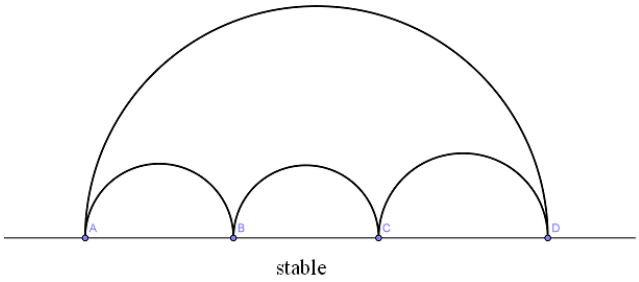

Figure 4: Stable cluster with 4 points. Any finite cluster is of this form, but may consist of an arbitrary number of points.

stable multi-matching with $D \equiv 2$. If there is an infinite component, it must be a double infinite path consisting of degree 2 vertices. Also, by the planarity of the graph, all finite components are literally trapped under the edges of the infinite component. Every finite component is either trapped under an edge of another finite component, or only under an edge of the infinite component. One can imagine how this looks like. Still it's unclear how the vertices are distributed among cluster sizes, and what edge length we may expect. In the next section we present simulation results in support of the conjecture that an infinite component indeed does exist, and we try to get a better picture of the resulting graph.

8 Simulations

Take a sample of N points $X_1, \cdots, X_N$ uniformly on the 1-dimensional torus S_1 equiped with its canonical distance

$$d(x, y) = \min\{|x - y|, 1 - |x - y|\}.$$

Apply the Gale-Shapely matching and denote by S_N the corresponding simulation with N points. Let C_N denote the maximal cluster in S_N. If in fact an infinite component does exist, by Theorem 16 there's a strictly positive probability p that a given vertex belongs to this component, and we expect to see the following happen:

$$|C_N|/N \to p \in (0, 1) \text{ as } N \to \infty.$$

Thus, we expect the largest component C_N to grow linearly with the system size. At the same time we expect the fraction $|C_N^{(2)}|/N$, where $C_N^{(2)}$ is the second largest component, to decrease. To be more precise,

$$|C_N^{(2)}|/N \to 0 \text{ as } N \to \infty.$$

Note that every finite cluster must be locked in between 2 points of the infinite cluster (if it exists!). That's why we expect all clusters smaller then C_N to be locked in between points of C_N, and expect to see $C_N^{(2)}$ increase exponentially slow. Results of these computer simulations are summarised in the next table.

Number of points	$\lvert C_N\rvert/N$	$\lvert C_N^{(2)}\rvert/N$
12.500	0.2679 ± 0.0254	0.0086 ± 0.0049
25.000	0.2930± 0.0185	0.0036± 0.0014
50.000	0.2899± 0.0071	0.0023± 0.0012
100.000	0.2818± 0.0064	0.0021 ± 0.00073959
200.000	0.2930 ± 0.0038	0.00066± 0.0001342

The proportion of points in the largest and second largest component are indicated as "Sample mean ± standard deviation" for a sample of size 5. Note that these results are consistant with our expectations, suggesting the existence of an infinite component with $p \approx 0.3$. Indeed, the fraction of points comprised by the largest component C_N seems to be relatively consistant. As expected we see the standard deviation of $|C_N|/N$ decrease as the number of points is increased, while its value floats around 0.3. At the same time we see the fraction of points comprised by the second largest component $C_N^{(2)}$ drop to zero.

Also, it would be interesting to understand how local structures are formed. For example, how are vertices distributed among different cluster sizes?. Is there a linear or exponential relation between cluster-size and fraction of points?. For large N, the behaviour in distribution seems relatively consistant. Thus to get a good idea, we set out the rounded average percentages against component size in the next table for $N = 200.000$ (from a sample of size 5).

cluster-size	pecentage of points
C_N	30
3	42
4	11
5	3
remaining	14

We see there's a big explosion of small clusters, and we can already see there's a slim probability to be taken up by a larger cluster (C_N being an exception). This is may be somewhat surprising. We shall give an appeal to intuition in an effort to explain this behaviour. If we examine the stability property a little closer, it's not hard to imagine that small clusters are easily formed. Also, suppose somewhere in the matching process the next situation occurs.

Now, these 6 points can become a cluster if the edge $\{x, y\}$ is added. But this means that there's no available point to match x with, on a distance larger then $|x-y|$ left from x. The same must hold for y in the right direction. This is very good possible, but since the points are uniformly distributed we may rather expect the opposite (and find at least one available point for x or y). In this case the component will be one point larger, and becoming a closed component will

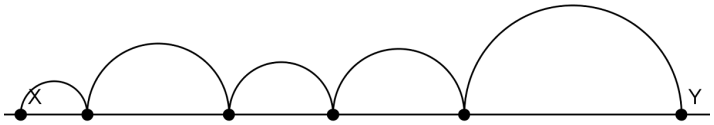

Figure 5:

be even harder. The larger $|x-y|$ becomes (as more points added), the harder this becomes. Still, this is only an appeal to intuition, as many of the points left from x, or right from y may be already organised configurations. That is, we may find many small groups with relatively small distances to eachother. Still, we may expect pre-organised configurations not to be very large. One can understand that the probability of having a very large pre-organised group points is quite small. To create a picture, we may filter out all pre-organised figurations of small groups, and apply the matching procedure to the remaining points. Distances between points will now be larger, and one might expect new pre-organised groups to form. Repeating this procedure, at some point pre-organised configurations must cease to exist. The remaining points will form the infinite cluster (C_N in the simulation). Exactly how this happens remains a mysterie.

Also, this doesn't fully explain why so little larger finite components exist, absorbing the vertices in between (already) organised smaller configurations. The best thing we can say about this, is that small configurations easily form, and comprise a high percentage of the points. In fact, so many points that there's simply not many points left to be absorbed in larger components. Picture 5 may not extremely convincing, but if it was a very large number of points, intuitively we may understand that it's not at all easy (examining the stability property) to become a closed component. Loosely speaking, we expect a sequence $(x_1, x_2), (x_2, x_3), \cdots, (x_{n-1}, x_n)$ of edges with vertices $x_1 < x_2 < \cdots < x_{n-1} < x_n$ having increasingly more difficulty becoming a closed component for increasing n. In the next section this idea will be formalized in a constructive statistical proof.
Note that (from table 2) the distribution of vertices among finite clusters (clusters smaller then C_N) seems to exhibit exponential behaviour. This is further supported by the computer simulations. Let $\mathbb{P}(|\mathcal{C}| > n)$ denote the conditional probability that a vertex is absorbed in a finite cluster larger then n points. We set $\mathbb{P}(|\mathcal{C}| > n) = e^{-c(p)\cdot n}$ where $c(p)$ is a constant to be retrieved (estimated) from the simulation data. A sample of size 5 with $N = 100.000$ gave an estimation $c(p) = 0.2901 \pm 0.0074$ which is around the same value as p.

9 A statistical proof

In the last section we saw that simulations on large cycles indicate the existence of an infinite component with $p \approx 0.3$. But we like to stress this is only an indication since convergence of $|C_N|/N$ may happen very slow. In this section we present more compelling statistical evidence of percolation for the 2-stub stable multi-matching. In order to do so, we will present a theorem by A. Holroyd wich allows us to give statistical proof of this fact.

The idea is to show that percolation in the 2-stub stable multi-matching follows from the assumption that a certain finite event has sufficiently large probability. There is also overwhelming statistical evidence for this assumption (also from simulations). This approach is very different from the one in the last section, as we're not approximating an infinite event with an finite event. Instead the problem is brought down to proving something about a *finite* event.

Theorem 16. *Let $\mathcal{P}$ be a Poisson process of intensity 1 on $\mathbb{R}$. If for some L we have $\mathbb{P}([0, L]$ is good $) > 0.968$, then the 2-stub stable multi-matching has an infinite component.*

Simulation results that suggest $\mathbb{P}([0, 13000]$ is good $) > 0.968$ can be found in [3].

We give a small outline of the proof. We are interested in the Gale-Shapely matching on $I \cap [\mathcal{P}]$ for some bounded interval $I = [a, b]$, but only with those edges that are also present in the stable multi-matching on $\mathbb{R}$. This matching is called the core (stable) multi-matching on I. The idea of the proof is to show that if the probability of the core multi-matching having a connected component from the first quarter of I to the last quarter of I is large enough, then the 2-stub stable multi-matching on $\mathbb{R}$ has a infinite component. All of this will be made precise below.

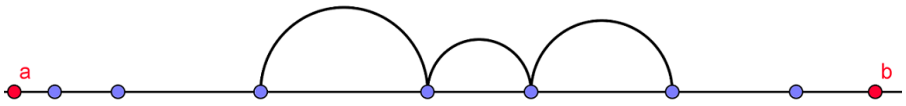

Figure 6: Example of a core multi-matching on some bounded interval $I = [a, b]$. Note that if a blue point is closer to a boundary point then to its closest neighbour in I, we can not connect the 2, for there may be closer (available) points behind the boundary points. The edges depicted are the only ones that can be added without knowing the configuration outside I.

The core multi-matching in the general setting of arbitrary dimensions and number of stubs is defined as follows. Let $S \subset \mathbb{R}^d$ be a bounded set, let $P \subset S$ be a finite set of points, and let $(D_x)_{x \in P}$ be the positive numbers representing

numbers of stubs. Let $\tilde{P} = P \cup \{S^c\}$. Assume that all distances of $\tilde{P}$ are distinct. Assign D_x stubs to each point $x \in P$ and one stub to S^c. Repeat the following operations:

From each point $x \in P$ that currently has an unused stub, assign an arrow pointing to the closest other element of $\tilde{P}$ among those that have at least one unused stub and do not allready have an edge to x. Then, for every $x, y \in P$ whose arrows point to each other, connect them with an edge and remove one stub from each. Erase all arrows and repeat. After a finite number of such iterations, no more edges are added. The core multi-matching of (P, D) in S is defined to be the resulting graph. Note that the degree of $x \in P$ is at most D_x but may be strictly less.

Lemma 17. *Let P be any discrete set of points in $\mathbb{R}$, let $(D_x)_{x\in P}$ be positive integers, and S be a bounded set. Every edge in the core multi-matching of $(P \cap S, D)$ in S is present in every stable multi-matching of (P, D) on $\mathbb{R}$.*

Lemma 17 is a straightforward verification that all edges in the core stable multi-matching are indeed present in the stable multi-matching on $\mathbb{R}$. We now restrict our attention to the case of main interest, that is $d = 1, S = I = [a, b]$, a bounded interval. Before we can prove Theorem 16, we need some lemma's and definitions.

Let $\mathcal{P}$ a Poisson process of intensity 1 on $\mathbb{R}$ with 2 stubs per vertex, i.e. $D \equiv 2$. From this point we refer to the core matching on I as the core multi-matching of $([\mathcal{P}] \cap I, D)$ on I.

Definition 18. The interval $I = [a, b]$ is called **good** if the core matching on I has a connected component with a point in the first quarter $[a, \frac{3}{4}a + \frac{1}{4}b]$ and a point in the last quarter $[\frac{1}{4}a + \frac{3}{4}b, b]$.

Definition 19. A **monotone path** in the multi-matching is a sequence of vertices $x_1 < x_2 < \cdots < x_k$ with the edges $(x_1, x_2), (x_2, x_3), \cdots, (x_{k-1}, x_k)$ all present.

Recall that we have no crossing edges in the 2-stub stable multi-matching, and therefore the same must hold in a core multi-matching. If I is good, it follows that I contains a monotone path from the first quarter to the last quarter. Such path is called **spanning path** of the good interval.

The following 2 lemma's are the building blocks for Theorem 16. The second lemma is a corollary of the first and is the key point in the proof. It provides a way of showing the existence of spanning paths in arbitrary large bounded intervals by assuming the existence of a certain number of smaller paths.

Lemma 20. *Let $a < b < c < d$ be points of $\mathcal{P}$, and suppose that the intervals $[a, b]$ and $[c, d]$ are both longer then $[b, c]$. If the 2-stub stable multi-matching has a monotone path α from a to b, and a monotone path δ from c to d, then it has a monotone path from a to d which contains α and δ.*

The proof of lemma 20 can be found in [3, lemma 4.3]. The idea is to extend α as far as possible to the right, idem for δ to the left. Then we get endpoints b' for α and c' for δ. (See fig. 4). A simple argument then shows that if there's no edge (b', c'), they must form an unstable pair. Therefore the edge (b', c') must be present, and we obtain a monotone path from a to d.

Figure 7: α is the monotone path on the left, δ on the right. Here b', and c' form an unstable pair, therefore the edge (b', c') must be present.

Lemma 21. *If at least 8 of the 9 intervals $[0, x], [x, 2x], \cdots, [8x, 9x]$ are good, then so is $[0, 9x]$. Furthermore, under the same assumption, given any spanning paths, one of each of the good short subintervals, there is a spanning path of the long interval containing all of them.*

Let the configuration outside $I := [0, 9x]$ be arbitrary and consider the stable multi-matching. Write $I_k = [(k-1)x, kx]$. Note that any 2 consecutive good intervals I_k, I_{k+1} contain monotone paths longer then $x/2$ with a gap of less then $x/2$ in between. Thus, applying lemma 4 we obtain a spanning path for $I_k \cup I_{k+1}$. So, for any sequence of consecutive good intervals $I_a, I_{a+1}, \cdots, I_b$, we obtain a monotone path, containing all spanning paths from $I_a, I_{a+1} \cdots, I_b$ and reaching to within distance $x/4$ of each end of $\bigcup_{k=a}^{b} I_k$. This means that the union of any number of consecutive good intervals is another good interval. Now, suppose 1 out of the 9 intervals $I_1, I_2, \cdots, I_9$ is bad. If I_3, I_4, I_5, I_6, I_7 are all good, then the resulting path reaches to within distance $2x + x/4 = 9x/4$ of each end of I, as required. On the other hand, if one of I_3, I_4, I_5, I_6, I_7 is bad (but the other 8 subintervals are good) then we have 2 monotone paths of length at least $3x/2$ on either side of the bad subinterval, with a gap of less then $3x/2$ in between. Again we may apply lemma 4 to obtain the desired spanning path.

Proof of Theorem 16. Let

$$I^k := \left[-\frac{9^k L}{2}, \frac{9^k L}{2}\right],$$

and let p_k the probability that I^k is good. By Lemma 21 we have $p_{k+1} \geq f(p_k)$, where

$$f(p) = p^9 + 9p^8(1-p).$$

First, we show $\sum_k (1-p_k) < \infty$. The derivative $f'(p) = 72p^7(1-p)$ is decreasing on the interval $[0.99, 1]$. By the mean value theorem it follows that

$$f'(c) = \frac{f(1) - f(p_k)}{1 - p_k} \geq \frac{1 - p_{k+1}}{1 - p_k},$$

for some c with $p_k < c < 1$. Hence for $p_k > 0.99$ we have

$$1 - p_{k+1} < f'(0.99)(1 - p_k) < 0.7(1 - p_k).$$

Let $p_0 \in [0.968, 1]$. Since f is increasing on $[0.968, 1]$ and $p = 1$ is the only fixpoint of f in this interval, we have $f^N(p_0) > 0.99$ for some finite number N. Hence, for some constant r the following bound holds

$$\sum_k (1 - p_k) < r \cdot \sum_k 0.7^k < \infty.$$

By the Borel-Cantelli lemma it follows that a.s. I^k is good for all sufficiently large k. That is, for some K and $k > K$ we have $p_k = 1$. Moreover, it follows that

$$\sum_k (1 - f(p_k)) < \sum_k (1 - p_{k+1}) < \infty.$$

Hence, by the same lemma, for all sufficiently large k we have $f(p_k) = 1$. This means that the interval I^k can be divided into 9 equal intervals of which at least 8 are good. By Lemma 21 it follows that for some (random) K we may find monotone paths $\pi_K, \pi_{K+1}, \pi_{K+2}, \cdots$, each contained in the next, where π_k is a spanning path of I^k for each k. Then

$$\mathcal{C}_\infty := \bigcup_{k \geq K} \pi_k$$

is an infinite connected component in the stable multi-matching.

References

[1] D.J. Daley and G. Last. Descending chains, the lilypond model, and mutual-nearest-neighbour matching. *Advances in applied probability*, 37(3):604–628, 2005.

[2] M. Deijfen, O. Häggström, and A.E. Holroyd. Percolation in invariant poisson graphs with iid degrees. *Arkiv för Matematik*, pages 1–18, 2010.

[3] M. Deijfen, A.E. Holroyd, and Y. Peres. Stable poisson graphs in one dimension. *Electronic Journal of Probability*, 16:1238–1253, 2011.

[4] J. Grandell. Mixed poisson processes, 1997.

[5] A.E. Holroyd, R. Pemantle, Y. Peres, and O. Schramm. Poisson matching. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 45, pages 266–287. Institut Henri Poincaré, 2009.

A Ergodic Theory

A set B is said to be T-invariant if $T^{-1}(B) = B$.
A filtration of a measure space $(\Omega, \mathcal{F})$ is a sequence of σ-algebra's $\{F_t\}_{t\geq 0}$ with $\mathcal{F}_t \subset \mathcal{F}$ and such that $t_1 < t_2$ implies $\mathcal{F}_{t_1} \subset \mathcal{F}_{t_2}$.

Levy's zero-one law.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ a probability space and let X a random variable in L^1. Let $(\mathcal{F}_k)_{k\in\mathbb{N}}$ be any filtration of $\mathcal{F}$, and define $\mathcal{F}_\infty$ as the minimal σ-algebra generated by $(\mathcal{F}_k)_{k\in\mathbb{N}}$. Then

$$\mathbb{E}[X \mid \mathcal{F}_k] \to \mathbb{E}[X \mid \mathcal{F}_\infty] \text{ as } k \to \infty.$$

Birkhoff's Ergodic Theorem.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ a probability space and let $T : \Omega \to \Omega$ be a measure-preserving transformation. Let $\mathcal{I}$ denote the σ-algebra of T-invariant sets. Then for every $f \in L^1(\Omega, \mathcal{F}, \mathbb{P})$, we have

$$\frac{1}{n}\sum_{j=0}^{n-1} f(T^j x) \to \mathbb{E}(f \mid \mathcal{I}).$$

for $\mathbb{P}$-a.e. $x \in X$.

B simulation code

The following matlab code was used for the simulations in section 8. The simulations were done in Matlab R2010b.

```
clear all
clc

tic;

N = 1*(1e4);
P = 100;

A = sort(rand(1,N));
B = zeros(1,2*P);
C = zeros(N,2*P);
voorkeur = zeros(N,P);
E = zeros(1,2*P);
```

```
for i = 1:P
    I = [1:i-1 i+1:i+P N-P+i:N];
    B = min(abs(A(I)-A(i)),1-abs(A(I)-A(i)));
    [C(i,:),E] = sort(B);
    voorkeur(i,:) = I(E(1:P));
    favoriet(i)=voorkeur(i,1);
end
for i = P+1:N-P
    I = [i-P:i-1 i+1:i+P];
    B = abs(A(I)-A(i));
    [C(i,:),E] = sort(B);
    voorkeur(i,:) = I(E(1:P));
    favoriet(i)=voorkeur(i,1);
end
for i = N-P+1:N
    I = [i-P:i-1 i+1:N 1:P-N+i];
    B = min(abs(A(I)-A(i)),1-abs(A(I)-A(i)));
    [C(i,:),E] = sort(B);
    voorkeur(i,:) = I(E(1:P));
    favoriet(i)=voorkeur(i,1);
end

C(:,P+1:end) = [];
toc;

tic;
M=zeros(N,2);
graad=zeros(1,N);
locatie=ones(N,1);
b=-1;

try
    while sum(graad)<2*N; %Gale-Shapely algorithm

        if sum(graad)> b
        b=sum(graad);
        else error('meer voorkeuren nodig')
        end;

        for i=1:N
            if graad(i)<2&& locatie(i)<P;
                while graad(favoriet(i))==2&&locatie(i)<P;

                    locatie(i)=locatie(i)+1;
                    favoriet(i)=voorkeur(i,locatie(i));
```

```
                end
            end

            if graad(i)<2&& i==favoriet(favoriet(i))&&graad(favoriet(i))<2
           &&locatie(i)<=P&&locatie(favoriet(i))<P;

                if M(i,1)==0;
                    M(i,1)=favoriet(i);
                else M(i,2)=favoriet(i);
                end

                if M(favoriet(i),1)==0;
                    M(favoriet(i),1)=i;
                else M(favoriet(i),2)=i;
                end

                graad(i)=graad(i)+1;
                graad(favoriet(i))=graad(favoriet(i))+1;

                locatie(i)=locatie(i)+1;
                locatie(favoriet(i))=locatie(favoriet(i))+1;

                if locatie(i)<=P
                favoriet(favoriet(i))=voorkeur(favoriet(i), locatie(favoriet(i)));
                favoriet(i)=voorkeur(i,locatie(i));
                end
            end
        end

    end
catch error
    disp(error.message)
end
toc

tic;

z=find(graad<2);
m=numel(z);
locatiez=ones(m,1);

for i=1:m;
    k=min(abs(A(z)-A(z(i))),1-abs(A(z)-A(z(i)))); % new sortation for vertices with d<2.
    [Y,I]=sort(k);

    for j=1:m-1
```

```
    voorkeurz(i,j)=I(j+1);
    favorietz(i)=I(2);
    end
end

try
    while sum(graad)<2*N; %Gale-Shapely algorithm for leftover vertices.

        for i=1:m
            if graad(z(i))<2;
                while graad(z(favorietz(i)))==2;

                    locatiez(i)=locatiez(i)+1;
                    favorietz(i)=voorkeurz(i,locatiez(i));

                end
            end

            if graad(z(i))<2&& i==favorietz(favorietz(i));
                if M(z(i),1)==0;
                    M(z(i),1)=z(favorietz(i));
                else M(z(i),2)=z(favorietz(i));
                end

                if M(z(favorietz(i)),1)==0;
                    M(z(favorietz(i)),1)=z(i);
                else M(z(favorietz(i)),2)=z(i);
                end

                graad(z(i))=graad(z(i))+1;
                graad(z(favorietz(i)))=graad(z(favorietz(i)))+1;

                locatiez(i)=locatiez(i)+1;
                locatiez(favorietz(i))=locatiez(favorietz(i))+1;

                favorietz(favorietz(i))=voorkeurz(favorietz(i), locatiez(favorietz(i)));
                favorietz(i)=voorkeurz(i,locatiez(i));

            end
        end

    end
catch error
    disp(error.message)
end
toc;
```

```
tic
T=ones(N,1);
h=1;
k=1;

while sum(T)>0 % finding cluster lengths

    lengte=1;
    STOP=0;

    while STOP==0

        if T(M(k,1))==1
            T(k)=0;
            k=M(k,1);
            lengte=lengte+1;

        elseif T(M(k,1))==0&& T(M(k,2))==1;
            T(k)=0;
            k=M(k,2);
            lengte=lengte+1;

        else T(k)=0; STOP=1; cluster(h)=lengte;
            h=h+1;
        end

    end

    k=1;
    while T(k)==0 && k<N
        k=k+1;
    end
```