Wilson, R. E., & Smith, D. A. (2005). Two-cluster dynamics for cellular automata with shuffle update.

Two-Cluster Dynamics for Cellular Automata with Shuffle Update

David A. Smith* and R. Eddie Wilson†
Bristol Centre for Applied Nonlinear Mathematics,
Department of Engineering Mathematics, University of Bristol,
Queen's Building, University Walk, Bristol BS8 1TR, United Kingdom
(Dated: November 18, 2005)

The random shuffle update method for the asymmetric exclusion process (ASEP) is introduced and the cluster dynamics technique is extended in order to analyse its dynamics. A sequence of approximate models is introduced, the first element of which corresponds to the classical parallel update rule whose two-cluster dynamics is reviewed. It is then shown how the argument may be extended inductively to solve for the two-cluster probabilities for each element of the sequence of approximate models. A formal limit is then taken, and macroscopic velocities and flow rates are derived.

PACS numbers: 89.40.-a, 02.50.-r

I. INTRODUCTION

The general context of this paper is the modelling of unidirectional road traffic or pedestrian flow with one-dimensional cellular automata of Nagel-Schreckenberg type [1], with the maximum velocity parameter $v_{\max}$ set equal to one. This type of model is sometimes referred to as the Asymmetric Exclusion Process (ASEP) [2, 3]. In this well-known set-up, space is discretised into a one-dimensional array of cells each of which is either empty or occupied by exactly one agent, and each agent moves according to a pair of very simple microscopic rules:

1. If the cell immediately downstream is occupied, remain stationary. (R1)
2. If the cell downstream is unoccupied, move forward into it with probability p, $0 < p \leq 1$. (R2)

The only remaining subtlety (and the subject of this paper) concerns the precise order in which rules (R1,2) are applied.

At each time step in the *parallel update* scheme [1, 4], rules (R1,2) are applied simultaneously to all agents. No conflict resolution is necessary, since rule (R1) automatically prevents multiple occupancy.

In contrast, at each time step of the *random sequential update* scheme [5, 6], rules (R1,2) are applied to a single agent chosen at random. For simulation purposes, the most attractive feature of the random sequential update is that single-occupancy is automatically preserved by (R1), and this would hold even in multi-dimensional situations, where the parallel update scheme would need a conflict resolution algorithm. However, a disadvantage of random sequential update is that with small probability, a single agent might receive a large number of consecutive turns, and thus, in low density situations, achieve an unphysical velocity.

*D.A.Smith@bristol.ac.uk
†RE.Wilson@bristol.ac.uk

This paper however is concerned with the dynamics of rules (R1,2) under the *shuffle update* scheme, which has received very little attention in the literature to date [7–10]. At each time step in this scheme, rules (R1,2) are applied to each individual agent in turn, according to a random order generated at the beginning of the time step, which contains each agent exactly once. After all agents have applied rules (R1,2), a new random order is generated and the next time step begins.

The shuffle update is similar to the random sequential scheme in that the occupancy of cells is updated incrementally as each agent applies its rules and consequently, it does not require conflict resolution to preserve single-occupancy (even in multi-dimensional extensions). However, the shuffle update enjoys the modelling advantage that the velocities of individual agents are bounded.

The chief result of this paper is an extension of the *two-cluster* analysis of Schreckenberg *et al* [11], which analyses (R1,2) under the parallel update scheme, to the more complicated case of the shuffle update. The argument here is more involved than [11] because under the shuffle update, it is possible for large blocks of contiguous agents to move forward in a single time step, if their turns are served in upstream order.

The paper is laid out as follows. In Sec. II, we illustrate with simple examples some features of the shuffle update scheme, and we introduce a sequence of approximations to it under which the number of agents moving forward from a contiguous block is bounded. Then in Sec. III, we describe the cluster dynamics method, and solve for the statistically stationary two-cluster probability for the two coarsest approximate schemes. Secs. IV and V then extend the argument inductively, to solve for the two-cluster probability in a sequence of truncated models which approximate the full scheme as close as we like. A formal limit is taken, then Sec. VI derives quantities such as the distribution of block lengths, average velocity, and average flow, as a function of the system density. Throughout we assume (i) that the system has reached statistical stationarity, (ii) that it is large, and (iii) that periodic boundary conditions are in force so

that agents leaving the right hand end of the system rejoin on the left, so that the total number of agents, and therefore the density, is conserved.

However, the analysis presented here is still approximate, since the two-cluster approximation is based on a spatial independence assumption that is unproven for the shuffle update: namely that the probabilities of three-clusters, four-clusters and so on factorise into products of conditional probabilities of two-clusters. This limitation and other conclusions are discussed further in Sec. VII.

II. SHUFFLED DYNAMICS AND TRUNCATED PROCESSES

In this section we use three simple examples to illustrate features of rules (R1,2) under the shuffle update scheme. We also introduce a family of truncated schemes which are used in the analysis that follows.

Each example, depicted in Figs. 1–3, starts at time t^* with an identical pattern of agents which are labelled $1, 2, \ldots, 7$ in upstream order. To start the time step, a turn-taking order must be chosen in which each agent appears exactly once. For all three examples we assume the same order

$$7 \;\; 1 \;\; 5 \;\; 2 \;\; 3 \;\; 4 \;\; 6. \qquad (1)$$

We then follow the sequence of moves through the time step to derive the state at time $t = t^* + 1$. In each figure, parts (a)-(g) show the decision making and incremental updating of the state, and part (h) shows the consequent state at time $t = t^* + 1$.

There are subtle details in the model used in each example which result in different sets of moves and different final states.

Example 1. See Fig. 1. Here we assume $p = 1$ in rule (R2), so that if an agent is served with an empty cell downstream of it, then it is sure to move so as to occupy it. The set of moves under turn-taking sequence (1) is as follows: (a) agent 7 is blocked and does not move; (b) agent 1 moves; (c) agent 5 moves; (d) agent 2 moves; (e) agent 3 moves; (f) agent 4 moves; (g) agent 6 moves. Note here that the contiguous block of agents 1–4 all move, since they are served in upstream order. In contrast, under the parallel update scheme, only agent 1 would move from this block. However, agent 7 does not get to move even with the shuffle update, since it is served before agent 6 which heads its block. Thus the turn-taking order is crucial, and this scheme is stochastic even when $p = 1$ because the turn-taking order is chosen randomly. Note further that holes cannot recombine, since the leading agent of each block will always move. In fact it may be shown that as $t \to \infty$, one obtains either a situation where the holes occur only as singletons, or where the agents occur only as singletons, depending on whether the mean density is greater or less than one half respectively.

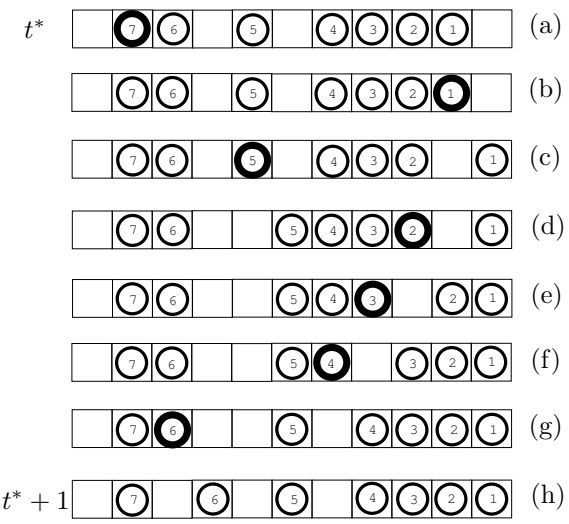

FIG. 1: One time step of the shuffled dynamics with $p = 1$ and turn-taking order 7,1,5,2,3,4,6: see Example 1. Each line of the figure in turn shows an agent highlighted in bold applying rules (R1,2). Note that the contiguous block of agents 1–4 all move.

Example 2. See Fig. 2. We take $p < 1$, so that agents may turn down the chance to occupy an empty cell downstream. Here the sub-steps are as follows: (a) agent 7 is blocked and does not move; (b) agent 1 is able to move and choses to; (c) agent 5 is able to move and choses to; (d) agent 2 is able to move but choses not to; (e) agent 3 is blocked and does not move; (f) agent 4 is blocked and does not move; (g) agent 6 is able to move but choses not to. The first point to note is that now only the lead agent from block 1–4 has moved, because agent 2, in choosing not to move, has impeded those that follow. Nevertheless all four agents might have moved in a different instantiation. Secondly, note that for $p < 1$, hole blocks may recombine, here by the singleton agent 5 choosing to move but agent 6 refusing the space in front of it.

An interesting feature of the shuffle update, which we have seen demonstrated in Example 1, but which by chance we have avoided in Example 2, is that large blocks of contiguous agents may move forward together in a single time step, if the turn-taking order permits. This potentially unbounded behaviour renders intractable the cluster dynamics method that we apply later. We fix this problem by defining truncated processes as follows.

Definition. By the *truncated process of order n*, we mean that rules (R1,2) are applied under the shuffle update scheme, with the proviso that the opportunity to move is offered only to agents who are in the first n positions of a contiguous particle block at the beginning of the time step.

Example 3. See Fig. 3. We apply the truncated process of order two, and for simplicity we assume $p = 1$. The sequence of moves is as follows: (a) agent 7 is blocked

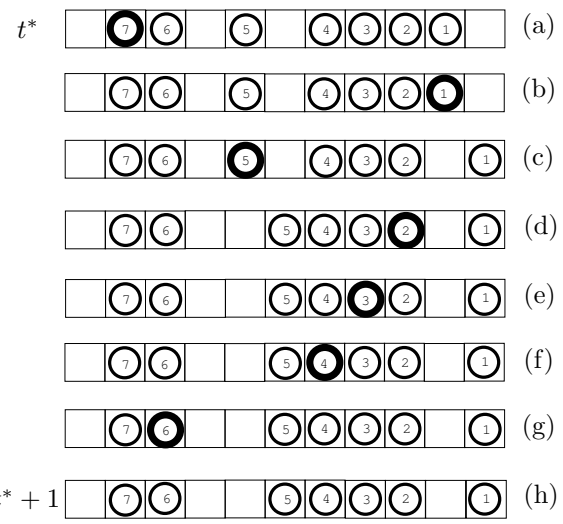

FIG. 2: One instantiation of one time step of the shuffled dynamics with $p < 1$ and turn taking order (1): see Example 2. Note that for $p < 1$, holes may recombine.

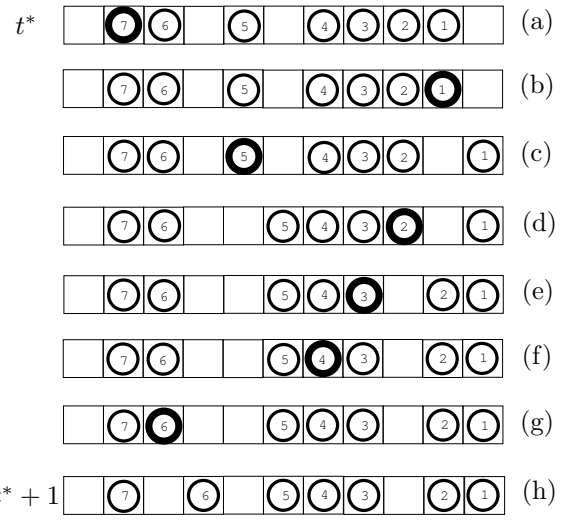

FIG. 3: One time step of the truncated process with $n = 2$, $p = 1$ and turn-taking order (1): see Example 3. Note that only the front two agents of block 1–4 are now able to move.

and does not move; (b) agent 1 is able to move and does; (c) agent 5 is able to move and does; (d) agent 2 is able to move and does (position 2 of its block at time t^*); (e) agent 3 is able to move, but does not because of the truncation (position 3 of its particle block at time t^*); (f) agent 4 is blocked and does not move; (g) agent 6 is able to move and does. Note that the final state (Fig. 3(h)) is quite different from Fig. 1(h).

An interesting point, that we examine in more detail shortly, is that the truncated process of order one is identical to the parallel update scheme. However, the limit of interest is that of truncated processes of order n, $n \to \infty$, under which one converges to the dynamics of the (full, untruncated) shuffle update scheme.

III. CLUSTER DYNAMICS METHOD

In order to compute distributions of block sizes and hence mean velocities and flow rates, we generalise the cluster dynamics method of Schreckenberg *et al* [11].

The chief idea is to compute the so-called *two-cluster* probabilities P_2 for the possible states of two adjacent cells, i.e., the probabilities of two adjacent cells having the states $(1,0)$, $(0,1)$, $(1,1)$ and $(0,0)$, where 0 and 1 denote empty and occupied respectively. Throughout we consider a statistically stationary situation where the two-cluster probabilities have converged to time-independent values. The argument is simplified by noting that the probabilities of the two-clusters $(0,1)$, $(1,1)$ and $(0,0)$ may be computed from the probability of the $(1,0)$ two-cluster alone, which we denote y. To see this, note

$$P_2(1,0) + P_2(0,1) + P_2(0,0) + P_2(1,1) = 1, \quad (2)$$

$$P_2(1,0) = P_2(0,1), \ (=: y), \qquad \text{and} \quad (3)$$

$$\frac{1}{2}\left(P_2(0,1) + P_2(1,0) + 2P_2(1,1)\right) = c, \quad (4)$$

where c denotes the prescribed mean density, i.e., the probability that a single cell is occupied. We thus have three equations in four unknowns and the goal is to derive an algebraic equation to solve for y, which then fixes other quantities thus:

$$P_2(1,1) = c - y, \quad (5)$$

$$P_2(0,0) = 1 - c - y. \quad (6)$$

Next we consider a window of adjacent cells surrounding a two-cluster whose time evolution is monitored: see e.g. Fig. 4 for the case of the parallel update scheme (identical to the truncated process of order one). Here the window is four cells wide, and we label the cells 0, 1, 2, 3 in downstream order, and we denote their contents by τ_0, τ_1, τ_2, τ_3 at the backward time step $t = t^* - 1$ and by σ_0, σ_1, σ_2, σ_3 at $t = t^*$. Our task is to catalogue all of the states at time $t = t^* - 1$ that can give rise to $(\sigma_1, \sigma_2) = (1,0)$ at time $t = t^*$. Such window states are listed in the left hand column of Fig. 4. Note that if a cell's occupancy is denoted by **?**, then all moves carry through with the same transition probabilities irrespective of whether that cell is occupied or not.

There are two remaining tasks. Firstly, we must find the transition probabilities W for each of the left hand to right hand column moves. These are calculated from the rules of the cellular automata and involve the parameter p. The second (more problematic) task is to to compute the probability of each left hand column window state in terms of the two-cluster probability y and the mean density c. Once these two steps are complete, $y = P_2(1,0) = P(\sigma_1 = 1, \sigma_2 = 0)$ may be expressed as a

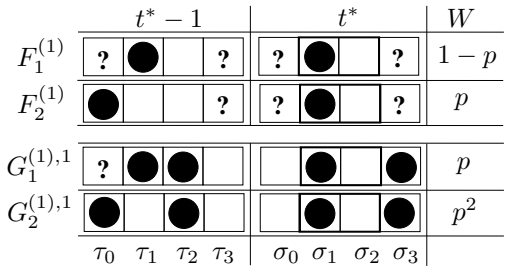

FIG. 4: The list of all possible transitions to a $(\sigma_1, \sigma_2) = (1, 0)$ two-cluster (highlighted in bold) at time step t^*, for the truncated process of order $n = 1$. The list is identical to that for the parallel update rule. Transition probabilities are denoted by W. Cells marked by **?** can be either occupied or empty, with no effect on the transition probability calculation since the movement or lack of movement of an agent in this cell cannot affect the monitored (σ_1, σ_2) two-cluster. The families of left hand column states labelled by $F_i^{(n)}$, $G_i^{(n),m}$ are the building blocks of the inductive process that follows later.

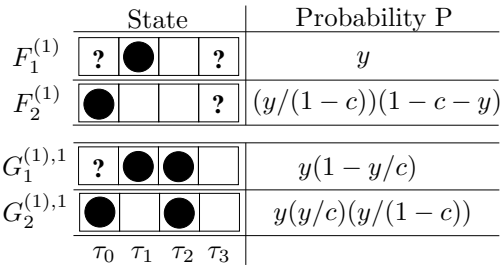

FIG. 5: Probabilities of the window states for the left hand column of Fig. 4. These are calculated approximately by combining two-cluster probabilities, and are expressed in terms of $y = P_2(1, 0)$ and the mean density c. Cells marked as **?** are effectively ignored, since their probability contribution is equal to 1.

sum product of the left hand window state probabilities (involving c and y) and the transition probabilities (involving p). Thus if c and p are fixed, we obtain a single scalar equation to solve for y, which may then be used to compute all the other quantities of interest.

Unfortunately the above argument is not rigorous, because the left hand window states involve three- and four-clusters, and consequently there is a spatial independence approximation involved in expressing their probabilities in terms of the two-cluster probability y. In fact, the results of the two-cluster analysis have been proven exact for the parallel update case [11], which as we have remarked, is the same as our truncated process of order $n = 1$. However, it remains to be shown whether the exactness holds or fails as we increase n.

Recall that so far, our discussion has centred not on the shuffle update, but on Fig. 4 which concerns the parallel update (equivalent to the truncated process of order $n = 1$). When using the shuffle update, there is no longer an upper bound on the number of agents who may move forward in a single time step from a contiguous block of agents. There is consequently the difficulty that no finite window of cells gives a full catalogue of the transitions to the $(1, 0)$ two-cluster.

Our procedure is thus to approximate the full shuffle update by the sequence of truncated processes for increasing $n \to \infty$. The argument is laid out as follows. In the remainder of this section, we provide the remaining details for the $n = 1$ case, in particular explaining the procedure for approximating large cluster probabilies. We then show how the argument may be extended to the truncated process with $n = 2$, by considering a larger catalogue of states of five-wide cell windows. Then, in Sec. IV, we show how inductively the argument may be extended to consider the truncated process for any finite n. At each level, the calculation involves a window which is $n+3$ cells wide, and a family of states with $2n+2$ members.

A. Truncated process with $n = 1$

This section completes the two-cluster analysis of the truncated process of order $n = 1$. Since this process is identical to the parallel update rule, this material is standard but it is necessary to present it here as it is the first step of our inductive process.

As we have explained, Fig. 4 lists all of the four-wide windows of cells at $t = t^* - 1$ which give rise to $(\sigma_1, \sigma_2) = (1, 0)$ at time $t = t^*$. There are two sets of quantities to be worked out: (i) transition probabilities W (results listed in Fig. 4), and (ii) probabilities of left hand states in terms of the two-cluster probability y (results listed in Fig. 5).

Take for example the left hand state $F_1^{(1)}$. This gives the corresponding right hand state if the occupant of cell 1, which has a hole in front of it, remains stationary, and all other moves, in particular the motion of the occupant of cell 3 (if there is one) are irrelevant. Thus $W(F_1^{(1)}) = 1 - p$, i.e. the probability of non-movement given by rule (R2).

For a second example, consider $G_2^{(1),1}$. The agent in cell 0 must move (with probability p) and independently the agent in cell 2 must move (with probability p). Consequently $W(G_2^{(1),1}) = p^2$. The remaining transition probabilities are worked out in a similar fashion.

We now calculate the window state probabilities. The easy case is $P(F_1^{(1)}) = P_2(1, 0), = y$. However, in general the probabilities of three- and even four-clusters must be computed. For example, $P(F_2^{(1)}) = P(\tau_0 = 1, \tau_1 = 0, \tau_2 = 0) = P_3(1, 0, 0)$, extending the cluster probability notation in the most natural way. Similarly, $P(G_1^{(1),1}) = P_3(1, 1, 0)$ and $P(G_2^{(1),1}) = P_4(1, 0, 1, 0)$.

To approximate higher order cluster probabilities, we use the conditional two-cluster probabilities which are

given as follows:

$$P_2(\underline{1},1) = P_2(1,\underline{1}) = 1 - y/c, \tag{7}$$
$$P_2(\underline{0},0) = P_2(0,\underline{0}) = 1 - y/(1-c), \tag{8}$$
$$P_2(\underline{1},0) = P_2(0,\underline{1}) = y/c, \tag{9}$$
$$P_2(\underline{0},1) = P_2(1,\underline{0}) = y/(1-c). \tag{10}$$

Here we have adopted the notation of [11], so that e.g., $P_2(\underline{1},0)$ denotes the probability of a cell being unoccupied if its left hand neighbour is occupied. Quantities (7-10) are computed simply from (3) and (5,6).

We now write, for example,

$$P(F_2^{(1)}) = P(\tau_0 = 1, \tau_1 = 0, \tau_2 = 0), = P_3(1,0,0)$$
$$\simeq P_2(1,\underline{0})P_2(0,0), \quad = \frac{y}{1-c}(1-c-y). \tag{11}$$

This is only an approximate result since the factorisation of the three-cluster probability relies on spatial independence which has not been proven. For a second example, we have

$$P(G_2^{(1),1}) = P_4(1,0,1,0)$$
$$\simeq P_2(1,\underline{0})P_2(0,\underline{1})P_2(1,0) \quad = \frac{y^3}{c(1-c)}. \tag{12}$$

The remaining window state probabilities are approximated similarly. In what follows we suppose that the above factorisation is exact, so that we may drop the $\simeq$ symbol.

We may now write down the probability $y = P_2(1,0)$ in terms of the sum product of the probabilities of left hand window states and their corresponding transition probabilities. We have

$$y := \sum_{i=1,2} \Big(W(F_i^{(1)})P(F_i^{(1)}) + W(G_i^{(1),1})P(G_i^{(1),1}) \Big), \tag{13}$$

which on substitution of the calculated probabilities simplifies to

$$\left(\frac{p^2}{c(1-c)}\right)y^2 - \left(\frac{p}{c(1-c)}\right)y + p = 0. \tag{14}$$

This quadratic has one valid root between 0 and 1 given by

$$y = \frac{1}{2p}\big(1 - \sqrt{1 - 4pc(1-c)}\big), \tag{15}$$

see [11]. As we extend the analysis to truncated processes of higher order n, we shall see that y solves in general a polynomial of degree $n+1$.

B. Truncated process with $n = 2$

We now extend the (so-far standard) two-cluster analysis to the truncated process with $n = 2$, where up two agents may move from the front of each block in one time step. Later, in Sec. IV, we extend this procedure to arbitrarily large n, thus analysing truncated processes arbitrarily close to the full system.

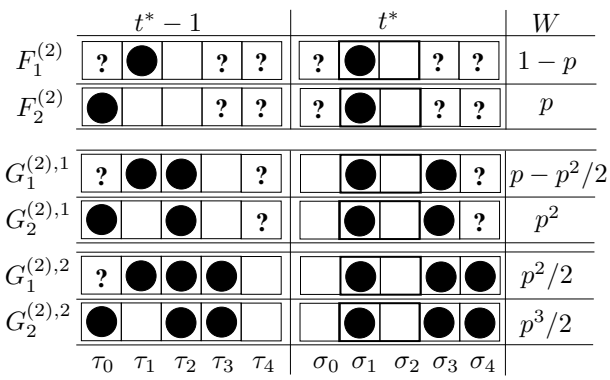

FIG. 6: The list of all possible transitions to a $(1,0)$ two-cluster at time step t^* for the $n = 2$ case. $W(F_i^{(n)}), W(G_i^{(n),m})$ denote the transition probabilities from the left column to the right column. The highlighted cells indicate the $(1,0)$ two-cluster that we seek. A **?** symbol means that the cell can be filled or empty, and effectively ignored since it has no effect on the monitored (σ_1, σ_2) two-cluster, and contributes nothing to the window state, or transition, probability.

The first step is to extend Fig. 4, which deals with the $n = 1$ process and which lists windows of cells at time $t^* - 1$ which can give rise to $(\sigma_1, \sigma_2) = (1,0)$ at time t^*. Whereas with $n = 1$ there were four such window states to consider, with $n = 2$ there are now six, labelled $F_{1,2}^{(2)}$ and $G_{1,2}^{(2),1,2}$, see Fig. 6. Here the parenthesised superscript denotes the order n of the truncated process, and the subscripts and letters F and G describe an hereditary relation with the window states for $n = 1$. Note that the G states also have a second superscript whose meaning we explain shortly. In some sense, we may think of the $n = 2$ states as descendants of those for $n = 1$, and the crux is to understand how the 'propagation' works.

Firstly, note that the propagation of the F family of states is trivial. To see this, note that the state $F_1^{(1)}$ has $(\tau_1, \tau_2) = (1,0)$: consequently, it produces $(\sigma_1, \sigma_2) = (1,0)$ via the occupant of cell 1 not moving. Likewise, state $F_2^{(1)}$ has $(\tau_0, \tau_1, \tau_2) = (1,0,0)$: so we require only the agent in cell 0 to move forward. Since the occupant of cell 0 is either an isolated agent, or the head of a block, the nature and probability of this transition is unaffected by the order n of the truncated process.

The difference between the $n = 1$ and $n = 2$ cases concerns the G family of states. Each of these has $\tau_2 = 1$, and hence requires the occupant of cell 2 to move to produce $\sigma_2 = 0$. In the $n = 1$ case, this can only happen if the occupant of cell 2 is either the head of a block (case $G_1^{(1),1}$) or isolated (case $G_2^{(1),1}$). When $n = 2$, there are more exotic possibilities. For example, we could have

$(\tau_2, \tau_3) = (1, 1)$, and provided the occupant of cell 3 is the head of its block, it is possible for both the occupants of cells 2 and 3 to move, leaving $\sigma_2 = 0$. To determine whether the occupant of cell 3 is the head of its block, it is thus necessary to consider the occupancy of the next cell downstream. Therefore, for the truncated process with $n = 2$, we must consider a five-wide window of cells.

We now turn our attention to the calculation of transition probabilities W: results are summarised in Fig. 6. For the $n = 2$ process we allow two agents to move from each block in one time step, but the second agent is only able to move if the front one has moved before it: with the (random) shuffle update order there is a probability $1/2$ of the front agent having its turn before the second. We thus have the following five cases to consider:

1. Front agent is updated and moves, then second agent is updated and moves, probability $= p^2/2$.
2. Front agent is updated and moves, then second agent is updated and does not move, probability $= p(1-p)/2$.
3. Front agent is updated and does not move, then second agent is updated and cannot move, probability $= (1-p)/2$.
4. Second agent is updated and cannot move, then front agent is updated and moves, probability $= p/2$.
5. Second agent is updated and cannot move, then front agent is updated and does not move, probability $= (1-p)/2$.

For states $F_i^{(2)}$, we have $W(F_i^{(2)}) = W(F_i^{(1)})$ because there is no second agent in a block to come in to play. We also have $W(G_2^{(2),1}) = W(G_2^{(1),1})$ since there are no blocks of length greater than one. However, we do see changes in the transition probabilities for the other $G_i^{(2),m}$ states. For $G_1^{(2),1}$ we have the front agent of the block moving but not the second: this can occur through either case 2 or case 4 as listed above, giving transition probability $p(1-p)/2 + p/2$. So $W(G_1^{(2),1}) = p - p^2/2$.

The transition probabilities for the $G_i^{(2),2}$ states are related to those for $G_i^{(1),1}$ by $W(G_i^{(2),2}) = (p/2)W(G_i^{(1),1})$, which comes from there being one more agent needing to move in the $n = 2$ case, doing so with probability p. Also this agent is the second in a block, so there is a probability of $1/2$ that the update order will allow it to move. This completes the discussion of transition probabilities.

We now see how the window state probabilities relate to those for $n = 1$ (see Fig. 7). Here we consider five-wide cell windows, since we need to consider $v_{\max}$ cells upstream, and n cells downstream of our monitored two-cluster. The top four window states in the left column of Figs. 6, 7 can be easily identified with the $n = 1$ window states (Figs. 4, 5), since the fifth cell takes the **?** state, contributing a factor of 1 to the probability. So we have

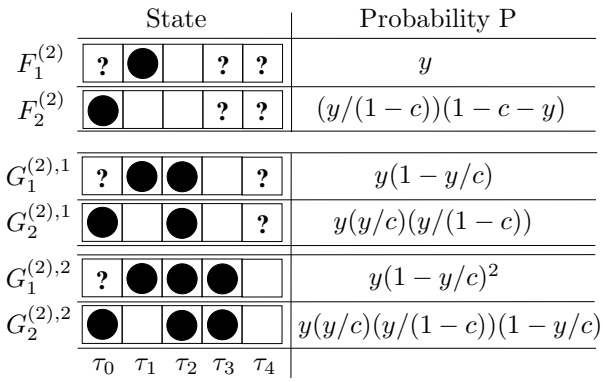

FIG. 7: Probabilities of the window states for the left hand column of Fig. 6. These are calculated approximately by combining two-cluster probabilities, and are expressed in terms of $y = P_2(1, 0)$ and the mean density c. Cells marked as **?** are effectively ignored, since their probability contribution is equal to 1.

$P(F_i^{(2)}) = P(F_i^{(1)})$ and $P(G_i^{(2),1}) = P(G_i^{(1),1})$. Then, as indicated by our choice of F and G notation, we relate the remaining two $n = 2$ window states $G_i^{(2),2}$ to the $n = 1$ states $G_i^{(1),1}$. We can imagine them as the same window state, but with an occupied fifth cell inserted into the middle, giving a contribution to the probability of $P_2(\underline{1}, 1) = (1 - y/c)$. So we have $P(G_i^{(2),2}) = (1 - y/c)P(G_i^{(1),1})$.

We can now write down the probability y of finding $(\sigma_1, \sigma_2) = (1, 0)$ according to the formula

$$y := P_2(1,0) = \sum_i W(F_i^{(2)})P(F_i^{(2)}) + \sum_{m\le 2}\sum_i W(G_i^{(2),m})P(G_i^{(2),m}). \quad (16)$$

On substitution of the transition and window state probabilities, this simplifies to

$$0 = f_2(y;c,p) := p - y\Big(\frac{p}{c(1-c)} + \frac{p^2}{2c}\Big) + y^2\Big(\frac{p^2}{c(1-c)} + \frac{p^2}{2c^2} + \frac{p^3}{2c(1-c)}\Big) - y^3\Big(\frac{p^2}{2c^2(1-c)}\Big), \quad (17)$$

which we want to solve for y between 0 and 1.

IV. INDUCTIVE CONSTRUCTION FOR TRUNCATED PROCESSES

In the previous section we showed explicitly how the calculation is done for the truncated processes with $n = 1$ and $n = 2$. We saw how the window states needed for

consideration at order $n = 2$ (Fig. 6) were related to those at order $n = 1$ (Fig. 4). We now generalise this so that we can look at the truncated process for any order n.

Our choice of labels for the states was chosen with this in mind, and we treat the $F_i^{(n)}$ and $G_i^{(n),m}$ separately as they have quite different extensions into higher n.

As was seen for the $n = 1, 2$ cases, the states denoted $F_i^{(n)}$ propagate unaltered as n increases (see Fig. 8). The cell window widens by one cell (on the right) at each step up in n, and this cell may take value either $\tau_{n+2} = 0$ or 1, with no effect on the calculations: this effect is indicated on the figures by **?**. We thus have $P(F_i^{(n)}) = P(F_i^{(1)})$ for all n. Similarly, we find $W(F_i^{(n)}) = W(F_i^{(1)})$ for all n because we need never consider the motion of more than one agent. In the $F_i^{(n)}$ states, we require that the front agent of a block does not move, so that all other motion is blocked and does not depend on the order n. Agents downstream have no effect on the monitored (σ_1, σ_2) two-cluster. In the $F_2^{(n)}$ states, the only motion is the front agent of a block moving into σ_1, and it does not matter how many agents move behind this one, as they do not enter the monitored (σ_1, σ_2) two-cluster. Again, agents downstream have no effect on the monitored (σ_1, σ_2) two-cluster.

The $G_i^{(n),m}$ states are more interesting in their propagation in n. They breed new states as well as propagating themselves (see Fig. 9). We define those states who, in Fig. 9, have arrows linking the n state to two $n+1$ states as 'breeding' states, while those with only one (horizontal) linking arrow we denote 'dormant' states. For clarity, we refer to extension along the horizontal arrows as propagation, and extension along the diagonal arrows as breeding. We now define the properties that make up these breeding and dormant states, before going on to discuss how their transition probabilities W to the monitored $(1, 0)$ two-cluster, and their window state probabilities P, change with increasing n.

To explain the inductive process, we use Figs. 9 and 10. In the $G_i^{(n),m}$ notation, the n, m specify the horizontal and vertical coordinates of the corresponding pair of G states measured from the top-left hand corner. The superscript m also gives the number of agents which are required to move from a single block in the corresponding window states. For instance, for $G_1^{(3),2}$ to have $(\sigma_1, \sigma_2) = (1, 0)$ at the next time step, two agents must move from the block, and the third must remain stationary.

Breeding states are those labelled $G_i^{(n),n}$, and are characterised by having $\tau_2, \ldots, \tau_{n+2} = 1$ (indicated on Fig. 9 by the dashed outlines). Thus n agents are required to move from a block in order that $\sigma_2 = 0$ at $t = t^*$. We say that these states 'breed' because $G_i^{(n),n}$ can be related to both $G_i^{(n+1),n}$ and $G_i^{(n+1),n+1}$. By examining Fig. 9, we see that the $G_i^{(n+1),n}$ state is obtained along a horizontal arrow by the window state growing in the manner of the F states, with an added right-hand **?**. The $G_i^{(n+1),n+1}$ state manifests itself by a diagonal arrow, with the block length growing accordingly.

These breeding states account for all changes in probabilities W and P as n increases, see Fig. 10. We see that along the diagonal arrows, state probabilities P gain a factor of $P_2(\underline{1}, 1) = 1 - y/c$, from the increased length of the block, while transition probabilities W gain a factor of p/n from the extra agent that is being required to move. On the horizontal lines from breeding states, we see that the window state probabilities are unchanged, while the transition probabilities only alter for the $G_1^{(n),m}$ states. Here we have $W(G_1^{(n),m}) = W(G_1^{(n-1),m}) - (p/n)W(G_1^{(n-1),m})$, which can be interpreted as Prob ($n-1$ agents move)$-$Prob (n agents move). We have this relation because with the increase in n, it becomes neccesary to specify that the last agent in the block does not move.

States labelled $G_i^{(n),m}$ with $n \neq m$ are 'dormant' and propagate unaltered in the same manner as the $F_i^{(n)}$ states. Transition probabilities do not change, as the number of agents required to move is already less than n and so is independent of n as it increases.

We can then summarise the probabilities shown in Fig. 10 inductively. For the state probabilities, we have $P(G_i^{(n),m}) = P(G_i^{(n-1),m})$ and $P(G_i^{(n),m}) = (1 - y/c)P(G_i^{(n-1),m-1})$, and for the transition probabilities, we have $W(G_i^{(n),m}) = (p/n)W(G_i^{(n-1),m-1})$; then (i) for $m < n$, $W(G_i^{(n),m}) = W(G_i^{(n-1),m})$; and (ii) for $m = n$, $W(G_1^{(n),m}) = W(G_1^{(n-1),m}) - (p/n)W(G_1^{(n-1),m})$ and $W(G_2^{(n),m}) = W(G_2^{(n-1),m})$.

By applying these rules we can write down all the terms needed to construct the algebraic equation $f_n(y; c, p) = 0$ (which we look to solve for y), in terms of $f_{n-1}(y; c, p)$. This task is tackled in the next section.

V. GENERAL SOLUTION FOR THE TWO-CELL CLUSTER PROBABILITY

In the previous section we have demonstrated the structure present in the sequence of truncated approximations. Our next task is to translate that structure into an equation to solve for y for any n, m, in the form

$$y = \sum_i P(F_i^{(1)})W(F_i^{(1)}) + \sum_{i,m} P(G_i^{(n),m})W(G_i^{(n),m}). \tag{18}$$

As n is increased some terms, corresponding to the $F_i^{(n)}$ states remain the same. Further new terms appear, and accumulate, corresponding to the breeding and dormant $G_i^{(n),m}$ states. We therefore seek an iterative process in the form $y =$ base terms + dormant terms + new terms.

The base term comes from the contribution of the $F_i^{(1)}$ states, since this remains unaltered for all n. The con-

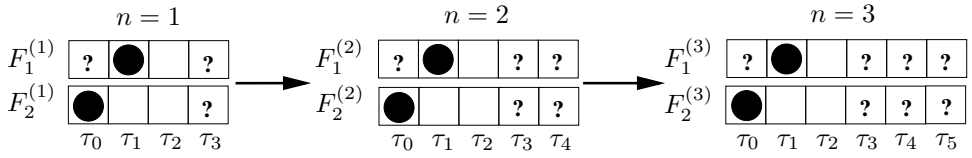

FIG. 8: The propagation of $F_i^{(n)}$ states as n increases. These states correspond to the left-hand columns in Figs. 4 and 6. The added right hand cell in each window takes the value **?** meaning that it can be either occupied or empty but we need not consider which, since it has no effect on the ability or probability to produce $(\sigma_1, \sigma_2) = (1, 0)$.

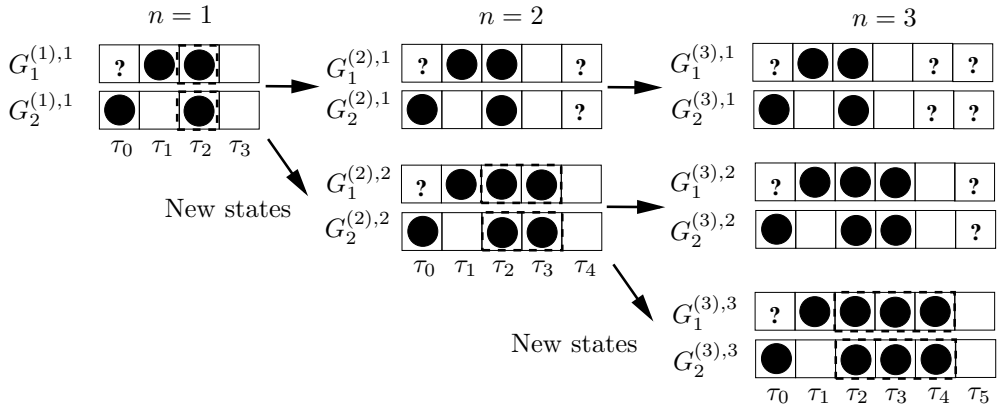

FIG. 9: Propagation of the $G_i^{(n),m}$ states as n increases. Existing states breed new ones as well as propagating in the same manner as the $F_i^{(n)}$ states (shown in Fig. 8). The characteristic feature of states which breed is that all cells with the dashed outline should be filled. For any given n, $F_i^{(n)}$ and $G_i^{(n),m}$ encompass all states capable of producing $(\sigma_1, \sigma_2) = (1, 0)$ at the next time step. The breeding of $G_i^{(n),m}$ means that the number of states increases by two each time n is increased by one.

tributing terms are therefore those we saw for $n = 1$ in Sec. III A in the form $\sum_{i=1}^{2} P(F_i^{(1)})W(F_i^{(1)})$. Thus we have

$$\begin{aligned} \text{base} &= y(1-p) + \frac{y}{1-c}(1-c-y)p, \\ &= y - \frac{py^2}{1-c}. \end{aligned} \tag{19}$$

As dormant states and breeding states all come from the $G_i^{(n),m}$ states, which all originate from $G_i^{(1),1}$ via the inductive development, we consider them together.

We start by writing down the terms contributed by the $G_i^{(1),1}$ states (which are classed as breeding states) and then build the inductive argument from there. These terms are of the form $\sum_{i=1}^{2} P(G_i^{(1),1})W(G_i^{(1),1})$. Thus

$$\text{breeding}_1 = y\big(1 - \frac{y}{c}\big)p + y\big(\frac{y}{c}\big)\big(\frac{y}{1-c}\big)p^2. \tag{20}$$

We note here that by solving

$$y = \text{base} + \text{breeding}_1,$$

we obtain equation(15) for the $n = 1$ case, as we would expect.

Now, to proceed with our inductive argument we recall that the breeding states are those labelled $G_i^{(n),n}$ and the dormant states are those labelled $G_i^{(n),m}$ with $m < n$. So new dormant states and new breeding states are produced by the respective propagation and breeding of $G_i^{(n-1),n-1}$ states. The first dormant states appear for $n = 2$, and are labelled $G_i^{(2),1}$ in Figs. 6, 9. We examine the $G_i^{(2),m}$ contribution to the equation for y as an extension of the breeding$_1$ terms, and note which correspond to the dormant states, and which to the breeding states.

By applying the rules outlined in Sec. IV, we have for $n = 2$,

$$\begin{aligned} y = \text{base} + \text{breeding}_1\big(1 + \frac{p}{2}\big(1 - \frac{y}{c}\big)\big) \\ - \big(\frac{p}{2}\big)P(G_1^{(1),1})W(G_1^{(1),1}), \end{aligned} \tag{21}$$

where

$$\text{dormant}_2 = \text{breeding}_1 - \big(\frac{p}{2}\big)P(G_1^{(1),1})W(G_1^{(1),1}), \tag{22}$$

and

$$\text{breeding}_2 = \frac{p}{2}\big(1 - \frac{y}{c}\big)\text{breeding}_1. \tag{23}$$

There was nothing special about the extension of $n = 1$ to $n = 2$, since the inductive form holds generally. By

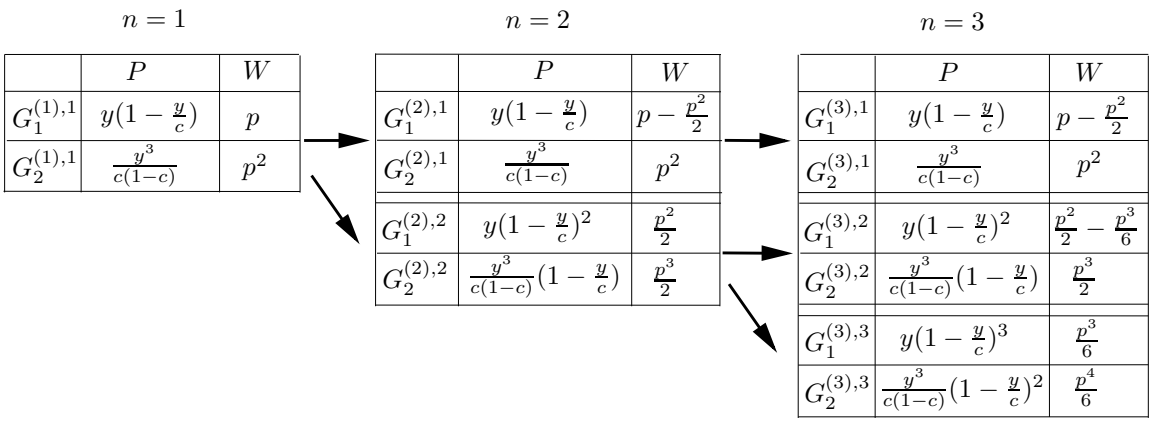

FIG. 10: The state and transition probabilities for the $G_i^{(n),m}$ for increasing n. We see unaltered state probabilities along the propagating (horizontal) arrows, but gaining a factor of $P_2(\underline{1},1) = 1 - y/c$ on the breeding (diagonal) arrows. Transition probabilities gain a factor of p/n on diagonal arrows, while $W(G_1^{(n),n-1}) = W(G_1^{(n-1),n-1})(1-p/n)$, and others remain unchanged on the horizontal arrows.

recalling that the dormant terms remain in our equation for all higher n, we can write down the complete equation

$$y = \text{base} + \sum_{j=1}^{n-1}\Big(\text{breeding}_j - \big(\frac{p}{j+1}\big)P(G_1^{(j),j})W(G_1^{(j),j})\Big) + \frac{p}{n}\big(1-\frac{y}{c}\big)\text{breeding}_{n-1}, \quad (24)$$

as a sum of all terms up to n. Here, the sum contains the terms from all the $2(n-1)$ dormant $G_i^{(n),m}$ states, and the last term is the contribution from the breeding $G_i^{(n),n}$ states.

We consider the breeding terms shortly, but now we write down the form of $P(G_1^{(j),j})W(G_1^{(j),j})$ (see Fig. 10). We can do this because this state always consists of $j+1$ contiguous occupied cells $(\tau_1,\ldots,\tau_{j+1})$ and one empty cell (τ_{j+2}), as well as a **?** cell with no contribution to the state probability. Thus we have

$$P(G_1^{(j),j}) = y\big(1-\frac{y}{c}\big)^j, \quad (25)$$

which is derived from the two-cluster probabilities for $(1,0)$ and $(1,\underline{1})$. Then also (see Fig. 10),

$$W(G_1^{(j),j}) = \frac{p^j}{j!}. \quad (26)$$

So

$$y = \text{base} + \sum_{j=1}^{k-1}\Big(\text{breeding}_j - \big(\frac{p^{j+1}}{(j+1)!}\big)y\big(1-\frac{y}{c}\big)^j\Big) + \frac{p}{k}\big(1-\frac{y}{c}\big)\text{breeding}_{k-1}, \quad (27)$$

which we can write as

$$y = \text{base} + \sum_{j=1}^{k-1}\Big(\text{breeding}_j - y\frac{p^{j+1}}{(j+1)!}\sum_{i=1}^{j+1}\binom{j}{i-1}\Big(\frac{-y}{c}\Big)^{i-1}\Big) + \frac{p}{k}\big(1-\frac{y}{c}\big)\text{breeding}_{k-1}. \quad (28)$$

In order to fully express our equation for y, we need to write explicitly, and generally, the terms from breeding states. These, unlike the dormant state terms, do not accumulate. There is one pair of breeding states at each order n, labelled $G_i^{(n),n}$; the information on lower order breeding states is included in the dormant state terms.

Again we refer to the breeding rules described in Sec. IV and Figs. 9 and 10. We see that in the change from n to $n+1$, the cell window has become one cell wider, and this extra cell is accounted for by an extra agent in the main block. Therefore, using the two-cluster method to write down state probabilities, the extra occupied cell manifests itself as the inclusion of a extra factor $P_2(1,\underline{1}) = \big(1-y/c\big)$. The transition probabilities obey $W(G_i^{(n),n}) = \big(p/n\big)W(G_i^{(n-1),n-1})$, since one more agent is required to move with probability p, and there is a $1/n$ probability that the agent is allowed to move due to the update order. We can then write any breeding_n terms by building inductively from the breeding_1 terms (20). We have already seen (23) that

$$\begin{aligned}\text{breeding}_2 &= \frac{p}{2}\big(1-\frac{y}{c}\big)\text{breeding}_1 \\ &= y\big(1-\frac{y}{c}\big)^2\frac{p^2}{2} \\ &\quad + y\big(\frac{y}{c}\big)\big(\frac{y}{1-c}\big)\big(1-\frac{y}{c}\big)\frac{p^3}{2}. \end{aligned} \quad (29)$$

We can go on to say that for general n,

$$\text{breeding}_n = y\big(1-\frac{y}{c}\big)^n\frac{p^n}{n!} + y\big(\frac{y}{c}\big)\big(\frac{y}{1-c}\big)\big(1-\frac{y}{c}\big)^{n-1}\frac{p^{n+1}}{n!}, \quad (30)$$

which we can rewrite as

$$\text{breeding}_n = y\frac{p^n}{n!}\sum_{i=1}^{n}\binom{n-1}{i-1}\left(\frac{-y}{c}\right)^{i-1} \times\left(1-\frac{y}{c}+\big(\frac{y}{c}\big)\big(\frac{py}{1-c}\big)\right). \quad (31)$$

We now have all the necessary ingredients to write down our equation to solve for y in the form

$$y = \text{base terms} + \text{dormant terms} + \text{breeding}_n. \quad (32)$$

We divide this through by y, and rearrange to get $f_n(y;c,p)=0$, which we then need to solve for y. We have

$$\begin{aligned} f_n(y;c,p) = &-\frac{py}{1-c} \\ &+\sum_{j=1}^{n-1}\frac{p^j}{j!}\sum_{i=1}^{j}\binom{j-1}{i-1}\left(\frac{-y}{c}\right)^{i-1} \\ &\quad\times\left(1-\frac{y}{c}+\big(\frac{y}{c}\big)\big(\frac{py}{1-c}\big)\right) \\ &-\frac{p^{j+1}}{(j+1)!}\sum_{i=1}^{j+1}\binom{j}{i-1}\left(\frac{-y}{c}\right)^{i-1} \\ &+\frac{p^n}{n!}\sum_{i=1}^{n}\binom{n-1}{i-1}\left(\frac{-y}{c}\right)^{i-1} \\ &\quad\times\left(1-\frac{y}{c}+\big(\frac{y}{c}\big)\big(\frac{py}{1-c}\big)\right), \end{aligned} \quad (33)$$

which can be simplified to give

$$\begin{aligned} f_n(y;c,p) = &\,p-\frac{py}{1-c} \\ &+\sum_{i=1}^{n}\left(\frac{-y}{c}\right)^{i}\sum_{j=i}^{n}\frac{p^j}{j!}\binom{j-1}{i-1}\left(1-\frac{py}{1-c}\right). \end{aligned} \quad (34)$$

Note that we must solve $f_n(y;c,p)=0$ for the two-cluster probability $y=P_2(1,0)$ for the truncated process of order n. In fact, (34) has an $n\to\infty$ limit which we may exploit to find an equation $f(y;c,p)=0$ to solve for the full untruncated process. We have

$$\begin{aligned} f(y;c,p) = &-(1-p) \\ &+\frac{1}{c-y}\left(1-\frac{py}{1-c}\right)\left(c-ye^{p(1-y/c)}\right). \end{aligned} \quad (35)$$

To solve this, in general, one needs to use numerical methods, but once that is done, y can be used as a building block for more interesting quantities, discussed in the next section.

VI. STEADY STATE VELOCITIES AND FLOW RATES

Once we have found y we can write down the probabilities for different block lengths. We formulate this as 'given an agent, what is the probability it is in a block of length l?' and construct it from the same two-cluster probabilities we used in the previous section. For instance, the probability of a block of length 2 is the probability that, given an agent, there is an empty cell on one side of it, and an occupied cell on the other side, and then an empty cell next to that, i.e. $\mathcal{P}_2 = P_2(\underline{1},0)P_2(\underline{1},1)P_2(\underline{1},0) = \big(y/c\big)^2\big(1-y/c\big)$. We can extend this similarly for other lengths, to give

$$\mathcal{P}_l = \big(\frac{y}{c}\big)^2\big(1-\frac{y}{c}\big)^{l-1}. \quad (36)$$

From this we can calculate the average velocity of the system. Since all peak velocities are one, the average velocity $\hat{v}$ is given by the proportion of agents moving at each time step.

There are now two possible approaches. Firstly, we may select an agent at random, find the probability that it is in a block of length l, and then calculate the probability that the agent moves from the kth position in that block. The second approach is to select a block of agents at random, find the probability that it has length l and then calculate the probabilities of k agents moving from the block. Here we use the first method.

Using the agent-oriented approach the probability that an agent chosen at random is in a block of length l is $l\mathcal{P}_l/\sum_i i\mathcal{P}_i$. But $\sum_i i\mathcal{P}_i = 1$, so we have obtained $l\mathcal{P}_l$. We then use the fact that the agent is equally likely to be in any position within the block, introducing a $1/l$ term, and we sum over all k the probability that it moves from the kth position. Using the block length probability found in equation (36), we have

$$\begin{aligned} \hat{v} &= \sum_{l=1}^{\infty} l\mathcal{P}_l\frac{1}{l}\sum_{k=i}^{l}P(\text{block serves at least } k \text{ agents}), \\ &= \sum_{l=1}^{\infty}\big(\frac{y}{c}\big)^2\big(1-\frac{y}{c}\big)^{l-1}\sum_{k=1}^{l}\frac{p^k}{k!}, \\ &= \big(\frac{y}{c}\big)^2\sum_{k=1}^{\infty}\frac{p^k}{k!}\sum_{l=k}^{\infty}\big(1-\frac{y}{c}\big)^{l-1}. \end{aligned} \quad (37)$$

We can simplify this expression, since the sum in l is the tail of a geometric series. We therefore have

$$\begin{aligned} \hat{v} &= \big(\frac{y}{c}\big)^2\sum_{k=1}^{\infty}\frac{p^k}{k!}\big(1-\frac{y}{c}\big)^{k-1}\sum_{l=0}^{\infty}\big(1-\frac{y}{c}\big)^{l}, \\ &= \big(\frac{y}{c-y}\big)\left(\exp\big(\frac{p}{c}(c-y)\big)-1\right). \end{aligned} \quad (38)$$

So, with a global density c and an average velocity $\hat{v}$,

we can write down flow in the form

$$\begin{aligned} q &= c\hat{v}, \\ &= \big(\frac{cy}{c-y}\big)\Big(\exp\big(\frac{p}{c}(c-y)\big)-1\Big). \end{aligned} \tag{39}$$

This result agrees with that found by Wölki *et al* [8], using a car-oriented mean field (COMF) method. The relationship between our site-oriented (SOMF) method and that paper is that P_0 in their notation is equivalent to $P_2(\underline{1},1) = 1 - y/c$ here.

VII. CONCLUSIONS AND DISCUSSION

In this paper we have extended the two-cluster analysis of the well-known Nagel-Schreckenberg cellular automata model, with $v_{\max} = 1$, to the case of shuffle update.

We introduced the shuffle update model, and we discussed the similarities and differences with the standard parallel update and random sequential update models. We then defined the truncated process of order n as an approximation to the full model, where a maximum of n agents are allowed to move from any contiguous block, and we noted that for $n = 1$ this is equivalent to the standard parallel update model.

The two-cluster method of analysis gives exact results for the parallel update model because neighbouring two-cluster probabilities are independent of each other. We extended this method to the $n = 2$ truncated process, and then by considering the general extension from n to $n + 1$ we constructed a sequence of approximations to the full shuffle update dynamics. We have therefore shown that, if the spatial independence of two-clusters is maintained for $n > 1$, we can obtain full expressions for the distribution of agents on the lattice, and further, their velocity and flow rates.

To investigate whether or not the spatial independence condition holds, one should investigate higher order (i.e. three- and four-) clusters. If the results for higher order clusters agree with the two-cluster method, then it would seem likely that spatial independence holds (and hence the results here would be exact), but it would still need to be proven.

The particular formulae obtained here are in agreement with the recent print by Wölki *et al* [8], which came to our attention while our manuscript was in preparation. Their analysis was done by using a car-oriented mean field (COMF) method and they considered the lengths of the hole blocks in front of a given agent. The results can be compared directly, since their quantity P_0 is equivalent to our conditional probability $P_2(\underline{1},1)$. Further, in [8] it is shown that the results are exact for $p = 1$.

For future work it would be interesting to study the shuffle update model for $v_{\max} > 1$ and compare with the parallel and random sequential models. Also an extension into multi-lane traffic may be worth pursuing, since as we have seen here, the shuffle update scheme guarantees collision avoidance, even in higher dimensions.

Acknowledgments

We thank Andreas Schadschneider and Marko Wölki for useful discussions, and George Lunt for deriving some of the preliminary results.

[1] K. Nagel and M. Schreckenberg, J. Phys. I **2**, 2221 (1992).
[2] N. Rajewsky, L. Santen, A. Schadschneider, and M. Schreckenberg, J. Stat. Phys. **92**, 151 (1998).
[3] D. Chowdhury, L. Santen, and A. Schadschneider, Phys. Rep. p. 199 (2000).
[4] A. Schadschneider, Eur. Phys. J. B **10**, 573 (1999).
[5] B. Derrida, E. Domany, and D. Mukamel, J. Stat. Phys. **69**, 667 (1992).
[6] K. Nagel, Ph.D. thesis, Universität zu Köln (1995).
[7] G. Lunt, Master's thesis, University of Bristol (2001).
[8] M. Wölki, A. Schadschneider, and M. Schreckenberg, preprint, arXiv:cond-mat/0509546v1 21 Sep 2005.
[9] T. Meyer-König, H. Klüpfel, and M. Schreckenberg, *A microscopic model for simulating mustering and evacuation processes onboard passenger ships* (2001), tIEMS.
[10] H. Klüpfel, Ph.D. thesis, Universität Duisberg (2003).
[11] M. Schreckenberg, A. Schadschneider, K. Nagel, and N. Ito, Phys. Rev. E **51**, 2939 (1995).