THE ONE DIMENSIONAL RANDOM PAIRING PROBLEM IN A CELLULAR ROBOTIC SYSTEM

OMER EGECIOGLU †
Department of Computer Science
University of California, Santa Barbara, CA 93106

BEATRIX ZIMMERMANN
Center for Robotics Systems
University of California, Santa Barbara, CA 93106

Abstract

A typical reconfiguration problem for an autonomous robotic system on the one dimensional grid is considered. The global goal of the system is to self-organize into units of physically adjacent pairs of robots separated by empty seats. By making use of randomization in the decision making process of each robot, the evolution of the system is modeled as a Markov chain, where each state represents a nonuniform random walk. The chain is absorbing, showing that the desired configuration will be reached with probability one.

1. Introduction

As robotics systems grow more complex and the demand for reliability increases, distributed systems in which many robots cooperate to accomplish a given task become a natural domain of study [1]. Given that robotics systems employing centralized control are susceptible to performance problems when the load on the central control increases, and to complete failure if the central control breaks down, the desirability of a fully distributed robotics system becomes apparent both in terms of reliability, and simplicity of design for a system of identical autonomous robots.

We consider a typical reconfiguration problem for an autonomous robotic system on the one dimensional grid. Each robot is of a relatively simple complexity with identical internal algorithm and limited sensing power constrained to its neighboring cells. The robotic system in question is further characterized in having no centralized control, no centralized data base, no shared memory, and no synchronous clock. These are the properties that essentially characterize Cellular Robotics Systems [3, 5].

The one dimensional pairing problem specified here utilizes an asynchronous, distributed algorithm which uses randomization. Even though the use of randomization results in a relatively simple protocol for the problem, the verification that the system will indeed accomplish its stated task requires careful analysis.

The One Dimensional Random Pairing Problem (RPP) has the following basic characteristics:

- The system is made up of a large, even number of identical robots.
- The robot motion takes place in a one dimensional unbounded grid of infinite *seats*.
- The robots know their left and right directions.
- The motion of a robot is limited to one step to the right or one step to the left provided such moves are possible.
- Each robot can sense the neighborhood around itself and cannot see through other robots. The neighborhood of a unit is defined as the first two seats to its left and the first two seats to its right.
- The internal algorithm of each robot contains a *randomized* component in the sense that decisions can be made depending on the outcome of a coin toss.

The global goal of the system is for the robots to self-organize into units of pairs. These pairs are required to be spatially separated by at least one empty seat in the final (or stable) configuration.

In the algorithm that we propose here, each robot starts out by selecting a random direction to move in. If the robot has empty seats surrounding it and there are no other neighboring robots moving towards it, then it will attempt to move in the direction chosen since its ultimate goal is to find other robots. If the robot is neighboring one or more robots, then it will perform one of a number of different possible state changes depending on the status of its neighbors and its own position relative to its local configuration.

The analysis of the algorithm first assumes that no two units move at exactly the same time, and thus there is a temporal order to the steps taken by the individual robots. In this case it is possible to model the problem as a finite Markov chain in which each individual state itself is a nonuniform random walk in an unbounded region of a possibly large dimensional Euclidean space. We prove that each one of the random walks is null-recurrent, and the underlying Markov chain is absorbing, where the absorbing state corresponds to the desired configuration of the robots into pairs. Thus with probability one, the robots will settle down into a formation of groups of twos, spatially separated from one another. The number of steps required for convergence however, seems to be difficult to estimate since the process depends heavily on the initial configuration of the units. The asynchronous case relies on the fact that the protocol disallows two robots to attempt to move into the same seat. In this case the evolution of the system can be viewed as taking place by a succession of equiprobable moves of individual robots, and the process can be modeled by a Markov chain as before.

2. Cellular Robotic Systems

A Cellular Robotic System (CRS) is a relatively new concept and thus it is desirable to work with a well defined, simplified model in which general problems can be studied and the correctness of the distributed algorithms verified. The components and the basic properties of a general CRS model is presented below. We formally define a *robot* as a quadruple $R = (T, r_0, R, f)$ where:

T - *the type of robot,*

r_0 - *initial state,*

R - *set of possible states,*

f - *transition function from one state to another.*

The transition function f is dependent on the current state and the state of the robot's neighborhood which is defined below.

A CRS is characterized by the following properties:

† Supported in part by NSF Grant No. DCR-8603722.

0-8186-2012-9/89/0000/0076$01.00 © 1989 IEEE

1. There are m independent robots where $m \geq 1$. The robots operate autonomously and thus are not synchronous. Note that each specific type of robot has the same R and f.
2. There are T types of robots where $T \geq 1$.
3. The space S in which the robots move is a k-dimensional connected grid. Each position in the grid is called a *seat* and is specifiable by its integral Cartesian coordinates with respect to some chosen origin and set of k axes in S. Two seats, $s_1, s_2 \in S$ are *adjacent* if they satisfy some relation $s_1 P s_2$. In the current model used $s_1 P s_2$ if and only if s_1 and s_2 differ by 1 in exactly one coordinate.
4. The seats which do not contain any robots are referred to as *empty seats*. The *empty space* E of the given grid is defined as the union of all the empty seats.
5. Each robot can sense and can be sensed by its *neighborhood*. This is the only communication the robot has available. In the current model, the d-neighborhood of a robot residing in a seat s is defined as the union of seats in S which are of the form $s + (d_1, d_2, \ldots, d_k)$ where $-d \leq d_i \leq d$ for $i = 1, 2, \ldots, k$.

A robot may post a set of messages to be seen by the robots in its neighborhood (possibly a different message for each neighbor). To simplify matters, the neighboring robots are guaranteed to see the relevant messages in a finite amount of time $\delta > 0$.

6. There is no centralized control or centralized data base. A robot operates and makes decision based solely on the messages it senses in its neighborhood and its internal algorithm f.
7. The resources of the system and in particular the number m of robots remains constant. Thus this model is a closed model.
8. The robots can *move* throughout the space S. Legal movement is defined in the following manner: Suppose there is a robot r residing in seat s_1 at time t. If at time t the seat $s_2 \in E$ is adjacent to s_1 then at time $t + \delta$ the robot r may reside in seat s_2 with $s_1 \in E$.

3. The One dimensional Random Pairing Problem

3.1 The CRS Model for the RPP

The RPP model being studied has the following characteristics:

1. It initially contains a relatively large even number of the same type of robots. Thus $T = 1$ and $m = 2N$ for some positive integer N.
2. The space S under consideration is the one-dimensional grid of lattice points which we take to be on the x-axis. Thus in this case S and E have infinitely many seats.
3. The robots can only sense the neighborhood around themselves and cannot see through other robots. The neighborhood of a robot is the 2-neighborhood as defined above.
4. The robots are equipped with a left-right orientation.
5. The movements of the robots is limited to the seats immediately adjacent to them on the left and on the right provided these seats are empty.
6. Each robot may post two signs *left_occupied* and *right_move* to its right side, and similarly two signs *right_occupied* and *left_move* to its left side.

The goal of this system is for the robots to to self-organize into groups of twos. These groups of twos must be spatially separated by at least one empty seat. Call a robot *happy* when it has one and only one neighboring robot. Individually, each robot seeks happiness and takes certain actions depending on the state of its neighboring cells to achieve this goal. The system is *stable* when all the robots are happy.

Even though the robots are unaware of their location in relation to the other robots, we shall label them with 1 through $2N$ from left to right. Since the robots cannot jump over other robots and the model is closed, this ordering remains fixed throughout the life of the system.

3.2 Description of the Algorithm f

A robot starts out by sensing its neighborhood. If both seats to its left and to its right are empty, then the robot moves to one of them after selecting a random direction by a fair coin flip. If both seats are occupied, then the robot posts the left_occupied sign on its right, and the right_occupied sign to its left side. If the seat to the left of the robot is empty and the seat to the right occupied, then the robot posts the right_occupied sign on its left side and then checks the sign on the left side of its neighboring robot. If the right_occupied sign is posted there, then the robot attempts to move to the empty seat to its left. Otherwise, it is happy, and does nothing. Similarly, if the seat to the robot's right is empty but the one to its left is occupied, then the robot checks the sign posted on the left side of the neighboring robot. If left_occupied sign is flagged there, then it attempts to move to the empty seat to its right. Otherwise it is happy and makes no move.

Formally, the design of the algorithm f for each robot is as follows:

```
LOOP FOREVER
Sense to the left and right
IF both seats to the left and right are empty THEN
    BEGIN
    MOVE(left) or MOVE(right) with equal probability
    set left_occupied = right_occupied = FALSE
    END

ELSE if both seats to the left and right are occupied THEN
    BEGIN
    make no move
    set left_occupied = right_occupied = TRUE
    END

ELSE if left seat is empty and right seat occupied THEN
    BEGIN
    set right_occupied = TRUE
    IF right_occupied of the right neighbor = FALSE  THEN make no
    move
    ELSE set right_occupied = FALSE
    MOVE(left)
    END

    ELSE if right seat is empty and left seat occupied THEN
    BEGIN
    set left_occupied = TRUE
    IF left_occupied of the left neighbor = FALSE  THEN make no
    move
    ELSE set left_occupied = FALSE
    MOVE(right)
    END

END LOOP
```

Note that the above algorithm assumes that no conflicts occur when two robots decide to move into the same empty seat. Under this assumption (to be justified below), the changes in the configuration of the robots can be put in a discrete temporal sequence where from one configuration to the next only one robot seems to make a move. This allows us to model the evolution of the configuration as a random process.

3.3 The Move Algorithm

The justification of non-simultaneity of the movements of robots to the same empty seat can be resolved by replacing the MOVE(left) and

MOVE(right) commands in the algorithm f by actual subprocedures that resolve conflicts while still enabling us to model the changes in the configurations as a random process. In order to implement these subprocedures, we allow each robot to post a *left-move* sign to its left side and a *right_move* sign to its right, indicating its intention to move in either direction by flagging the proper sign. Note that the move algorithms themselves are probabilistic.

```
PROCEDURE MOVE(left)
{Assume the robot is in seat s }
BEGIN
set left_move = TRUE
IF the seat s−2 is empty THEN
    BEGIN
    move to the left seat
    set left_move = FALSE
    END
ELSE BEGIN
X: IF right_move of robot in s−2 is = FALSE THEN
    BEGIN
    move to the left seat
    set left_move = FALSE
    END
    ELSE BEGIN
    Y: Set left_move = TRUE/FALSE with probability ½
    IF seat s−1 is not empty THEN
        BEGIN
        set left_move = FALSE
        EXIT MOVE(left)
        END
        ELSE BEGIN
        IF left_move = FALSE THEN GOTO Y
        IF left_move =TRUE THEN GOTO X
    END
END
END MOVE(left)
```

The procedure MOVE(right) is similar to MOVE(left) and will be omitted.

4. The Coordinate-free Representation of the RPP

Suppose at time t, robot i is residing at the lattice point $r_i(t)$ in S. Then at time t, the configuration of the robots is completely described by the position vector $r(t)=(r_1(t), r_2(t), \ldots, r_{2N}(t))$ where $r_1(t) < r_2(t) < \cdots < r_{2N}(t)$ with the initial state $r(0)=(r_1(0), r_2(0), \ldots, r_{2N}(0))$. Since the separation of the robots rather then the actual number of empty seats between them is important for the RPP, we may represent the *pattern* of $r(t)$ by a coordinate-free binary string $b(t)=b_1(t)b_2(t)\cdots b_{2N-1}(t)$ of length $2N-1$ where for $i=1, 2, \ldots, 2N-1$

$$b_i(t) = \begin{cases} 0 & \text{if } r_{i+1}(t)-r_i(t)=1, \\ 1 & \text{if } r_{i+1}(t)-r_i(t)>1. \end{cases}$$

Thus in $b(t)$, ones represent one or more spaces between robots and the zeros indicate that the robots are touching. As an example the following configuration of six robots

$$\ldots\_\_\_\times\times\_\_\_\times\_\_\_\times\times\_\_\_\_\times\_\_\ldots$$

where the symbol $\times$ denotes the position of a robot and $\_$ an empty seat, is represented by the binary string

$$0\ 1\ 1\ 0\ 1\ .$$

The system is in a stable state when for some $t \geq 0$,

$$b(t)=b_a=0101\cdots010=0(10)^{N-1}\ . \tag{4.1}$$

This means that the system has converged to the desired pattern, since the configuration represented by $0(10)^{N-1}$ consists of pairs of robots separated by empty seats.

We now consider the effect of the change of state of the robots from $r(t)$ to $r(t+1)$ on the string $b(t)$. When robot i with $1 < i < 2N$ makes a move, the change in the string $b(t)$ only depends locally on the substring $b_{i-1}(t)\,b_i(t)\,b_{i+1}(t)$ and affects only this portion. The possible local transformations on $b(t)$ are indicated below where the arrows indicate the direction of time:

$$\begin{array}{ll} A) & 111 \rightarrow 101 \\ B) & 110 \longleftrightarrow 100 \\ C) & 011 \longleftrightarrow 001 \\ D) & 100 \rightarrow 010 \end{array} \tag{4.2}$$

In the special case of the leftmost robot ($i=1$) the local transformations become

$$\begin{array}{ll} E) & 10\ldots \longleftrightarrow 00\ldots \\ F) & 11\ldots \longleftrightarrow 01\ldots \end{array} \tag{4.3}$$

and for the rightmost robot ($i=2N$) they are

$$\begin{array}{ll} G) & \ldots 01 \longleftrightarrow \ldots 00 \\ H) & \ldots 11 \longleftrightarrow \ldots 10 \end{array} \tag{4.4}$$

Now consider the finite Markov chain M whose state space consists of all binary strings b of length $2N-1$. Note that in going from $r(t)$ to $r(t+1)$, either $b(t+1)=b(t)$, or $b(t+1)$ is obtained from $b(t)$ by the application of one of the local transformations A through H given in (4.2)-(4.4). The state space of the Markov chain M for four robots is shown in Figure 1.

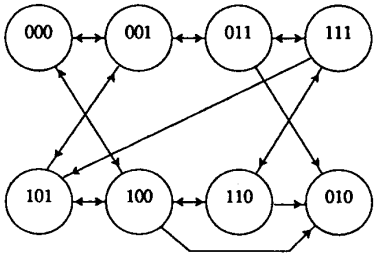

Figure 1: *The Markov chain M associated to the RPP. The absorbing state is $b_a=010$.*

5. The Convergence of the RPP

We want to show that starting at any initial configuration, the stable state represented by b_a is reached with probability one i.e., M is an *absorbing chain.* Note that since there are no transitions out of b_a, this state is absorbing.

The proof of convergence can be broken down into two parts.

(I) Show that starting in any non-absorbing state b there is zero probability of remaining in that state and a positive probability of transition to any neighboring state given by the transformations A - H above.

(II) The absorbing state b_a can be reached from every state b.

Proof of Part (II) :

Assuming part (I) for the moment, we first prove the easier part (II). For the proof, the following inductive argument is used:

Given a nonabsorbing state b, by transformations A, B, C, F, and H above, it is clear that from b one can reach a state with no two consecutive ones. Thus without loss of generality, we may assume that b has no two consecutive ones. Furthermore, by using the transformation G if necessary, we may also assume that $b_{2N-1}=0$. Now consider the two rightmost digits. If these are (10) then this section is stable. If the next digit to the left is b_1, then then either $b_1=0$ and we are done, or b_1 can be changed to zero by transformation H. Thus we may assume that there is another pair of digits to the left adjacent to this (10). If this pair is (00) then the left zero must change to a one. If there is a one to the left of the (00), it is possible for the left zero to become a one by transformation B. If there is a zero to the left of the (00), that zero can change to a one if there is a one to its left. Now consider the leftmost one in b. If there are no ones to the left of this position in b, one can be created by the transformation G above. The leftmost one can be propagated to the right by transformation B or D until there is a one to the left of the (00) pair in question which can then produce the (10) stable pair. The next leftmost pair can then be examined. If it is a (01) it can be changed to a (00) and the previous case applies. Clearly, the final string reached is b_a. □

Example 1 :

00000
10000 (F)
01000 (D)
00100 (D)
00010 (D)
(now consider the next two positions to the left)
10010 (F)
01010 (D)

Example 2 :

1110111
1110110 (G)
1010110 (A)
1010100 (B)
1010010 (D)
(now consider the next two positions to the left)
1001010 (D)
(now consider the next two positions to the left)
0101010 (D)

Proof of Part (I)

For the proof of this part of the convergence result, we recall that in the binary encoding $b(t)=b_1(t)b_2(t)\cdots b_{2N-1}(t)$, the digit $b_i(t)=1$ if and only if the robots i and $i+1$ are separated by a positive number of seats. We first consider an exemplary case. When the process is in state $1(01)^{N-1}$, this means that the robots 1 and 2 are separated by some $r_2(t)-r_1(t)=X_1(t)>0$ seats, the robots i and $i+1$ are paired up for $i=2,3,\ldots,2N-2$, and the robots $2N-1$ and $2N$ are separated by $r_{2n}(t)-r_{2N-1}(t)=X_{2N-1}(t)>0$ seats. Thus the projections $X_1(t)$ and $X_{2N-1}(t)$ define two independent one-dimensional random walks $\mathbf{X}_1$ and $\mathbf{X}_2$ with the identical one-step transition function

$$P(0,x)=\begin{cases}\frac{1}{4} & \text{if } x=\pm 1,\\ \frac{1}{2} & \text{if } x=0,\\ 0 & \textit{otherwise}.\end{cases} \tag{5.1}$$

The *hitting time* of a one-dimensional random walk $\mathbf{X}$ with $X(0)=x(0)>0$ is defined by

$$T(x)=\min\{\,t\,|X(t)=0\,\}\ .$$

It is well-known [2, 4] that the hitting time of a one-dimensional random walk $\mathbf{X}$ with the one-step transition function

$$P(0,x)=\begin{cases}p & \text{if } x=\pm 1,\\ 1-2p & \text{if } x=0,\\ 0 & \textit{otherwise}.\end{cases}$$

for any p with $0<p\le\frac{1}{2}$ is *null-recurrent*. In other words

$$Pr\{T(x)<\infty\}=1\ ,$$

$$E\{T(x)\}=\infty\ .$$

Thus in M, there is zero probability of remaining in state $1(01)^{N-1}$ and a positive probability of transition to any neighboring state. Clearly, a similar argument can be applied for the states $0(01)^{N-1}$ and $(10)^{N-1}0$.

In the case of a general state b of M, we consider the subwords of the string b consisting of maximal consecutive sequences of zeros and ones, each of length >1.

Example :

$$1110100011110 \quad\rightarrow\quad 111,000,1111\ .$$

A maximal consecutive sequence of zeros corresponds to a configuration of consecutive robots of the form

and a maximal sequence of ones to a configuration of the form

$$\times\_\ldots\times\_\ldots\times\_\ \cdots\ \times\_\ldots\times$$

where the robots are separated by a positive number of seats.

Algorithm f allows no conflicts of movement between the boundaries of such blocks. Thus the analysis can be restricted to independent random walks that correspond to each such maximal sequence. In the case of a maximal sequence of zeros, the situation is identical to the case of the analysis of the state $1(01)^{N-1}$ since here only the extreme zeros can turn into ones. Thus we only need to study the random walk that corresponds to a maximal consecutive sequence of ones in b.

Consider a maximal sequence of ones in b of length n. The corresponding random walk

$$\mathbf{X}[\mathbf{n}]=\{\,\mathbf{X}(t)=(X_1(t),X_2(t),\ldots,X_n(t))\,\}$$

in n–dimensional Euclidean space has the one-step transition function

$$P(0,\mathbf{x})=\begin{cases}\frac{1}{2h} & \text{if } \mathbf{x}\in\{\pm e_1,\pm e_n,\pm(e_i-e_{i+1})\}\\ & \text{for } 1\le i<n,\\ 1-\frac{1}{2h} & \text{if } \mathbf{x}=0,\\ 0 & \textit{otherwise},\end{cases}$$

where $e_1,e_2,\ldots,e_n$ are the standard basis vectors and $h\ge 2$ is the number of robots in pattern b which can potentially make a move. If the string b has i_1 singletons 0 and i_l blocks of zeros of length l in its decomposition, then h is given by

$$h=2N-2i_1-\sum_{l=2}^{2N-1}(l-1)\,i_l\ .$$

As an example of the random walk $\mathbf{X}[\mathbf{n}]$, the transitions for the case $n=2$ are given in Figure 2. Note that the positive axes are absorbing boundaries. Absorption at the x_1-axis corresponds to the transformation $11\rightarrow 01$, and absorption at the x_2-axis corresponds to the transformation $11\rightarrow 10$.

Note that in general the robot that corresponds to the first 1 in the maximal sequence of ones under consideration may not be the leftmost robot in the system, forcing X_1 to be bounded. A similar situation holds for the rightmost robot. However, these cases do not affect the following

analysis in any significant way.

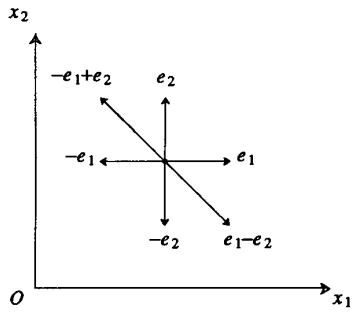

Figure 2 : *The random walk* $\mathbf{X}[2]$.

The projections of $\mathbf{X}[\mathbf{n}]$ define n one-dimensional random walks $\mathbf{X}_i$ with transition function

$$P_i(0, x_i) = \begin{cases} \frac{1}{h} & \text{if } x_i = \pm 1, \\ 1 - \frac{1}{h} & \text{if } x_i = 0, \\ 0 & \textit{otherwise}. \end{cases}$$

Analogous to (5.1), the i^{th} *-hitting time* of $\mathbf{X}_i$ with $X_i(0) = x_i > 0$ is defined by

$$T_i(\mathbf{x}) = \min \{ t \mid X_i(t) = 0 \} .$$

The *hitting time* of $\mathbf{X}[\mathbf{n}]$ where

$$\mathbf{X}(0) = \mathbf{x} = (x_1, x_2, \ldots, x_n), \quad x_i > 0 \quad \text{for } 1 \le i \le n$$

is defined by

$$T(\mathbf{x}) = \min \{ T_1(\mathbf{x}), T_2(\mathbf{x}), \ldots, T_n(\mathbf{x}) \} .$$

Set

$$p_i = Pr \{ T_i(\mathbf{x}) < \infty \} .$$

For the convergence of the RPP, we need to show that $p_i > 0$ and $p_1 + p_2 + \ldots + p_n = 1$. To show $p_i > 0$, consider the region R in n-dimensional space defined by $x_j \ge 1$ for $j \ne i$, and $x_i \ge 0$. If $i \ne n$, then the steps $e_i + e_{i+1}$ repeated x_i times is a sample path of positive probability in R which ends at $x_i = 0$. Thus $p_i > 0$. If $i = n$, then the steps $-e_n$ can be chosen instead. To show $p_1 + p_2 + \ldots + p_n = 1$, it suffices to show that $T(\mathbf{x})$ is null-recurrent.

Theorem 5.1 The hitting time $T(\mathbf{x})$ is null-recurrent.

Proof : Associate with the process $\mathbf{X}[\mathbf{n}]$ the n one-dimensional random walks $\mathbf{X}_1^*, \mathbf{X}_2^*, \ldots, \mathbf{X}_n^*$ defined by deleting from the sample path of each projection $X_i(t)$ the parts of the path at which $X_i(t)$ does not change. Then the transition function of $\mathbf{X}_1^*$ is

$$P(0, x) = \begin{cases} \frac{1}{2} & \text{if } x = \pm 1, \\ 0 & \textit{otherwise}. \end{cases}$$

Furthermore, $T_i^*(\mathbf{x}) \le T_i(\mathbf{x})$. Note that the projection of the process $\mathbf{X}[\mathbf{n+2}]$ onto the first n coordinates is just $\mathbf{X}[\mathbf{n}]$ and

$$\min \{ T_1^*, T_2^*, \ldots, T_n^*, T_{n+2}^* \} \le \min \{ T_1^*, T_2^*, \ldots, T_n^* \} \le \min \{ T_1, T_2, \ldots, T_n \}$$

But the process $\mathbf{X}_{n+2}^*$ is independent of the processes $\mathbf{X}_1^*, \mathbf{X}_2^*, \ldots, \mathbf{X}_n^*$ and hence

$$E \left\{ \min \{ T_1^*, T_2^*, \ldots, T_n^*, T_{n+2}^* \} \right\} = E \left\{ \min \{ T_1^*, T_2^*, \ldots, T_n^* \} \right\} E \left\{ T_{n+2}^* \right\}$$

But the one dimensional random walk $\mathbf{X}_{n+2}^*$ is null-recurrent. Thus $E \{ T_{n+2}^* \} = \infty$. Thus $T(\mathbf{x})$ itself must be null-recurrent. □

In view of Theorem 5.1, starting in any non-absorbing state b there is zero probability of remaining in b and a positive probability of transition to any neighboring state given by the transformations A through H. Therefore, starting at any initial configuration, the system will converge to the desired state of pairs of robots separated by empty seats with probability one.

Conclusion

The robotics system considered in this paper is cellular, an is essentially characterized in having no centralized control, no centralized data base, no shared memory, and no synchronous clock. Each robot in the system executes an identical internal algorithm and is equipped with limited sensing power.

A typical reconfiguration problem called the Pairing Problem for such an autonomous robotic system on the one dimensional grid is studied. Here the global goal of the system is to self-organize into units of physically adjacent pairs separated by empty seats. By making use of randomization in the decision making process of each robot, the evolution of the system can be modeled as a Markov chain, where each state represents a nonuniform random walk. The chain is absorbing, showing that the desired configuration will be reached with probability one.

To estimate the expected time for the system to converge seems difficult. However, with proper modifications in the distributed algorithm, the probabilistic approach taken here should imply convergence to the desired pattern when the underlying space is bounded with reflecting boundary walls.

Acknowledgement

We would like to thank Professor Alan Konheim for the proof of Theorem 5.1.

References

[1] G. Beni, "The Concept of Cellular Robotic System", 3rd IEEE International Symposium on Intelligent Control, Arlington, Virginia, August 1988.

[2] E. Cinlar, *Introduction to Stochastic Processes*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1975.

[3] S. Hackwood and J. Wang, "The Engineering of Cellular Robotics Systems", 3rd IEEE International Symposium on Intelligent Control, Arlington, Virginia, August 1988.

[4] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, D. Van Nostrand Co., Inc., Princeton, New Jersey, 1960.

[5] J. Wang and G. Beni, "Pattern Generation in Cellular Robotic Systems", 3rd IEEE International Symposium on Intelligent Control, Arlington, Virginia, August 1988.