INFORMATION AND CONTROL 36, 245–255 (1978)

The Parallelism Principle: Speeding Up the Cellular Automata Synchronization

FRANCESCO ROMANI

Istituto di Elaborazione della Informazione del CNR, via S. Maria 46, Pisa, Italy

In recent years we have seen many generalizations of the well-known Firing Squad Synchronization problem. Now it is possible to synchronize networks of finite automata arbitrarily connected. This paper shows a general principle according to which synchronization time of figures with particular symmetry properties will decrease considerably.

1. INTRODUCTION

The Firing Squad Synchronization problem was stated by Myhill in 1957 (Moore, 1964) and solved by Waksman (1966) and Balzer (1967) among others. This paper deals with a generalization of the problem stated as follows.

Consider a set of n identical finite state machines; at time t everyone enters a new state that is a function of the states at time $t-1$ of the automaton itself and of some other automata, called its neighbors, arbitrarily chosen but limited in number; the neighborhood relation is supposed symmetrical.

At time $t = 0$ all the automata are in quiescent states, called *Soldier* states, but one is in an active state G called the *General*. We wish to define the single automaton structure so that after a sequence of transitions, at time $t = t_f$ all the automata enter, for the first time, the state F called the *Firing* state.

It has been shown by Rosenstiehl (1966; Rosenstiehl *et al.*, 1973) and Romani (1976) that each connected network of n automata can be synchronized in a time less than or equal to $2n$.

We will show that it is possible to achieve considerably lower times, using our knowledge of particular properties of the figures we deal with. Let us introduce the problem using two examples.

EXAMPLE 1. Consider the class of linear arrays with odd length and the General in the center. Using the general and optimal method of synchronization due to Moore and Langdon (1968) we reach the Firing status in $2n - k - 2$ steps, where k is the distance between the General and the near end of the array; in this case $k = (n-1)/2$ and

$$t_f = 2n - 2 - (n-1)/2 = 3/2(n-1).$$

0019-9958/78/0363-0245$02.00/0

But using our knowledge that the General is in the middle of the array, we can synchronize concurrently the two identical arrays departing from the General; their length is $(n + 1)/2$ and so

$$t_f = 2(n + 1)/2 - 2 = n - 1.$$

This synchronization time improvement is obtained using the knowledge of the position of the General.

EXAMPLE 2. A linear array, with the General at one end, can be synchronized in $n - 1$ steps if all the automata "know" the array length. Everyone has $n + 1$ states $Q_0, \ldots, Q_n$; Q_0 is the soldier state, Q_n the General state. At time $t = 0$ the General (the automaton X_0) is in the state Q_n. At time $t = 1$ X_0 and X_1 enter the state Q_{n-1}. At time $t = k$ $(k < n - 1)$ $X_0, X_1, \ldots, X_k$ enter the state Q_{n-k}. Finally at time $t = n - 1$ all the automata enter the state Q_1 (Fire).

Clearly this automaton can work only on linear arrays of length n.

In Section 2 we enunciate a general principle to decrease the synchronization time for large sets of figures. Then we define five types of applications of the above principle, classifying synchronization methods of many figures previously studied separately, namely, multidimensional arrays, information-lossless, and fixed-oriented.

2. THE PRINCIPLE OF PARALLELISM

Consider the network of Fig. 1; it consists of a ring from which four linear arrays depart. Clearly it is possible to synchronize such a figure in less time than using general methods: assume that the General is an automaton in the ring,

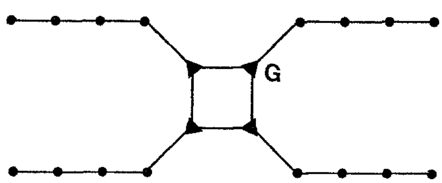


FIG. 1. A composite graph.

then the process acts as follows. At time $t = 0$ the ring synchronization starts, at time $t = f'$ the Firing state of the ring automata causes the arrays' synchronization to start simultaneously and, because the arrays are identical, they will fire together and the second Firing status is the true Firing status for the whole network.

The Parallelism Principle is a formalization of this synchronization schema. Given a connected graph E with m subgraphs $D_1, D_2, \ldots, D_m$ such that:

(i) every node of E belongs at least to one subgraph D_i,

(ii) it is possible to synchronize all the subgraphs in the same time p (henceforth we call figures with this property *equivalent* figures).

(iii) There exists a subgraph A of E, synchronizable in time q and for each D_i there is at least one node common to D_i and A: then it is possible to synchronize E as follows.

(a) The subgraph A synchronizes in q steps and all its automata enter simultaneously an intermediate Firing status.

(b) For each D_i there is an automaton of A that, entering the Firing status acts as General for the D_i synchronization.

Since all D graphs start simultaneously their synchronization process at time $p + q$ all the automata of the network enter the true Firing status together.

Synchronization time does not depend on the number m of the subgraphs D but on their size and organization, thus the larger is m the more useful this technique will be.

Moreover the General does not have to belong to A but any automaton in the network may be allowed to be the General: let "a" be the General and "b" an automaton of the graph A; "a" sends signals to all the automata in the network to start the process; when the signal reaches "b" it will start the A synchronization and this will happen after k steps, where k is the distance between "a" and "b," so the synchronization time will be $p + q + k$.

Obviously "b" must be in a different Soldier state or have unique neighborhood properties to be the only automaton to start the synchronization process for A.

In order to apply such a principle to a given kind of figures (i.e., a family of graphs with common properties), we require:

(i) analysis of the generic figure to determine the basic elements for the synchronization, namely, the subgraphs A, the D's, and the secondary Generals;

(ii) construction of an automaton capable of synchronizing every graph belonging to the set under analysis.

Problem (i) can be solved using an explicit differentiation of Soldiers at the beginning of the synchronization: we may have as initial states: the General state G, the secondary Generals S_i, the A Soldiers AA_i, and the D Soldiers DD_i.

Clearly this complete differentiation is not always necessary: for certain kind of figures some information can be obtained from the local properties of the graph (e.g., number of neighbors, the orientation, etc.). There are figures for which no special Soldier states are needed.

Moreover with the complete differentiation there can be synchronized only figures composed of a limited number of equivalent subgraphs because the number of states of the solution automaton must, however, be finite.

3. Classification of Parallelism Principle Applications

In this section we define five types of graph analyses according to the properties of graphs D and A. Each type of analysis can be used to synchronize large sets of figures.

We define *application of a type T analysis*:

(a) an automaton M with a set IM of possible initial states;

(b) the set of all the graphs E with the following property: "there exists an initial marking with states belonging to IM capable of synchronizing E using M as node automaton and following a type T analysis."

Thus each application allows us to synchronize a family of graphs with common properties using the same solution automaton. If the initial marking is of the standard type $\{G, S\}$ the application is an acceptable solution to the classical Firing Squad problem (i.e., with only two initial states allowed) for that family of graphs.

The first analysis we define is:

NIS (*Nonintersection Single*)

In this case the graph E to be synchronized is obtained joining m equivalent subgraphs D_1, D_2,..., D_m without intersection and the subgraph A is formed by only one node for each subgraph D.

For example, rectangular arrays satisfy a NIS-type analysis (see Fig. 2): rows form the D graphs and the column containing the General is the A graph.

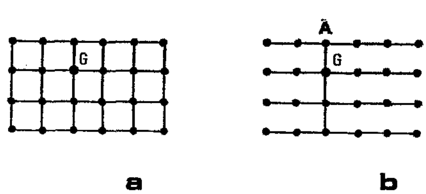


FIG. 2. (a) A rectangular array; (b) the same figure seen as union of four linear arrays.

To perform the synchronization, individuals of A and D can be recognized either implicitly, using neighborhood properties, or explicitly, using an initial marking of the type $\{G, S_2,..., S_m, DD_1,..., DD_m\}$; the AA states are not required here because A is composed by secondary Generals (we assume G to belong to A and to be the secondary General for D_1).

As an example of NIS-type application we expose a synchronization method for k-dimensional arrays. The process is structured as follows:

The General starts the synchronization of the linear array obtained varying the first-dimension index. Each element of the array, when firing, acts as secondary General for the $(k-1)$-dimensional array obtained varying all the other indices. This array is synchronized with the same method if $k > 2$, otherwise a standard method for linear arrays is used.

Let $(j_1, j_2, \ldots, j_k)$ be the position of the General in the array: j_i is the distance between the General and the near end of the array on the ith dimension. It is easy to see that the synchronization time for an array $m_1 \times m_2 \times \cdots \times m_k$ is:

$$T = \sum_{i=1}^{k} (2m_i - j_i - 2).$$

For cubic arrays with the General in position $(0, 0, \ldots, 0)$ we have $k = 3$, $m_i = m$, $j_i = 0$, and $T = 6m - 6$ that is not minimal but of the same order of the optimal result $3m - 3$ due to Shinahr (1974).

The second type of analysis is:

NIM(*Nonintersection Multiple*)

A graph E is synchronizable with a NIM-type method if it is composed of m subgraphs connected without intersection, and there exists a graph A containing one or more elements of each D_i.

To synchronize a NIM-type graph it is necessary to distinguish secondary Generals. We can use a complete marking like $\{G, S_2, \ldots, S_m, DD_1, \ldots, DD_m, AA_1, \ldots, AA_m\}$ but there are figures for which the standard marking $\{G, S\}$ can be used.

For example, consider the family of all the graphs composed by m equivalent figures with Von Neumann neighborhood joined to a $2m$-automata ring like in Fig. 3a.

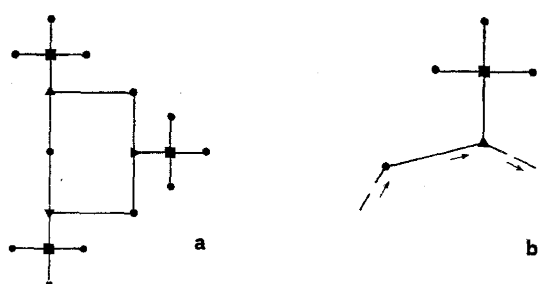


FIG. 3. (a) A NIM figure example; (b) one of the equivalent graphs.

Membership of an automaton to a subgraph can be determined on the basis of the number of neighbors and their orientation. Automata with one or four neighbors do not belong to the ring and are not involved in the first synchronization process; automata with two or three neighbors form the ring and are applied to the first synchronization; moreover a two-neighbour automaton acts as as secondary General for the D figure formed by: itself, their neighbor in the clockwise direction and the Von Neumann figure departing from it (Fig. 3b).

Clearly ring automata are supposed to be able to distinguish their clockwise, neighbors. The synchronization time is $p + 2m$ where p is the synchronization time of the D figure and $2m$ is the ring synchronization time.

In the following classifications automata are allowed to belong to different D graphs.

During synchronization, if the behavior of a common automaton is the same in all the figures where it appears, then it will act normally (see for an example the fixed-oriented synchronization method due to Grasselli, 1975). Otherwise a common automaton will simulate as many automata as sufficient to avoid interferences.

IO (*Intersection—One*)

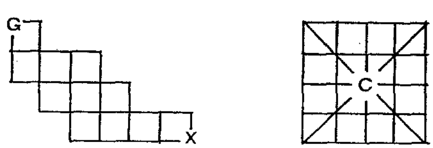
An analysis of a graph E is of the IO type if there exists one automaton common to all the equivalent figures D.

The set of initial states required to apply an IO method in a general case is $\{G\} \cup T$, where T is the set of all compound states formed by elements of $\{DD_1, DD_2, ..., DD_m\}$ and an automaton "a" is in the initial state $\langle DD_i, DD_j, ..., DD_k\rangle$ if "a" belongs to $D_i, D_j, ..., D_k$.

The A graph is formed by the General G that is the automaton common to all D's, so the first synchronization process is not required and the General can start directly the synchronization of all the equivalent figures.

As two examples of a particular IO application, where a standard marking $\{G, S\}$ is sufficient we can see:

(a) The fixed-oriented figures: the graph A is formed by the General and the equivalent figures are the minimal paths from the General to the opposite automaton X (see Fig. 4a).



a b

FIG. 4. (a) A fixed-oriented figure; (b) an information-lossless figure with the core formed by one automaton.

(b) The information-lossless figures with the core formed by one automaton (Fig. 4b). In this case the General is the central automaton C and the D graphs are the minimal paths from C to the outermost layer.

In these examples the equivalent subgraphs are linear arrays of the same length and the automata belonging to mor than one array appear in the same position, therefore there are no interferences.

IA (*Intersection—All*)

The characteristic of this analysis type is that in the graph A each automaton acts as secondary General for some D graphs. In order to avoid ambiguities we require the identification of the D graphs and their secondary Generals: this can be done with a complete marking similar to the IO case with secondary General states added ($\langle S_{i'}, S_{i''}, ..., S_{i^*}\rangle$) for the automata of A.

As an example of a synchronization method of IA type with a standard initial marking $\{G, S\}$ we present the synchronization method for information-lossless figures (Fig. 5) due to Grasselli (1975).

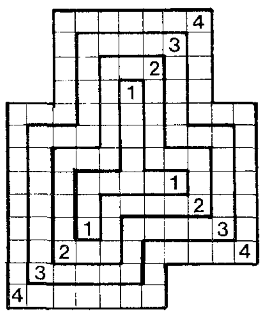


FIG. 5. An information-lossless figure with evidenced layers.

The A graph is the outermost layer and the D graphs are the direct paths from the outermost layer to the core. In this case there are no interferences because each automaton has the same position in all the direct paths it belongs to. Moreover the General is not required to belong to the outermost layer but may be any automaton in the figure.

I (*Intersection*)

This is the analysis type with the weakest conditions: in the Parallelism Principle applications of the I-type intersection of equivalent graphs is allowed and the A graph may contain automata not acting as secondary Generals.

The initial marking must be sufficient to distinguish the membership of each automaton to D graphs, to the A graph and, if it is a secondary General, the set of D graphs for which it acts.

As an example of I-type application we give a synchronization method valid for a set of figures composed of identical square arrays. This method works for figures with the following characteristics.

(a) the figure is composed by identical square arrays with Von Neumann neighborhood connected with intersection allowed;

(b) each automaton may belong only to one or two different squares;

(c) the number of squares is $4m$, their dimension is $l \times l$, and they form a symmetrical "pseudoring" (Fig. 6). A figure of this kind is totally determined once given the parameters m, l, and k ($k < l/2$).

(d) The automata are supposed capable to distinguish the neighbor orientation (N, E, S, W).

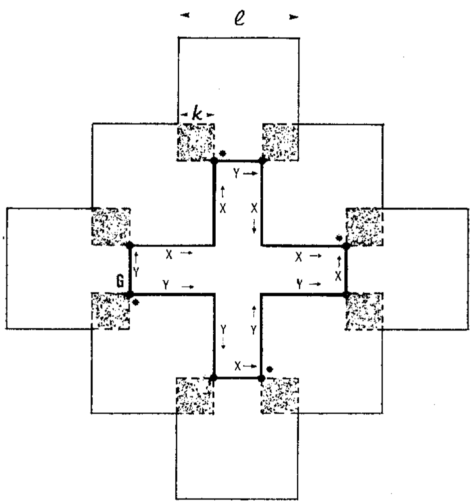


FIG. 6. An I-type figure: there are eight identical square arrays, the internal ring is evidenced, and the flow of activating signals X and Y is shown; the starred automata are the secondary Generals.

It is easy to see that the outermost layer of our figure is not connected but is formed by two rings. The A graph we choose is the internal ring and the secondary Generals are one-half of the points of A belonging to two squares.

We use an initial marking of the form $\{G, S, SS\}$: the General belongs to the internal ring and to two squares, S marks automata belonging to one square, SS automata belonging to two squares.

The General, when starting the internal ring synchronization, sends two Y

signals in both directions; when a Y signal reaches an SS autmaton, it is changed into an X signal; when the X signal reaches an SS, it is changed into a Y and the automaton is marked as secondary General. So half of the SS automata in the ring is marked as secondary General and, when the ring fires, each of them will start the synchronization of the two squares it belongs to.

The SS automata will simulate two independent automata to perform synchronization without interferences.

The ring synchronization time is:

$$4((l - 2k) + 2(m - 1) \cdot (l - k)) = 8m(l - k) - 4l.$$

The square synchronization time is $2l - k - 2 + 2l - 2$ so the Firing status is reached in:

$$8m(l - k) - 4l + 2l - k - 2 + 2l - 2 = 8m(l - k) - k - 4.$$

Note that the number of automata in the figure is $4m(l^2 - k^2)$ and using a general method the synchronization time would be about $8m(l^2 - k^2)$.

Table I summarizes all the analyses defined in this paper. In Fig. 7 an inclusion

TABLE I

Summary of the Classes Introduced in the Text

	Equivalent figures	
Graph A	Intersection not allowed	Intersection allowed
One automaton only	—	IO
Each automaton is a secondary General	NIS	IA
There are automata of A not acting as secondary Generals	NIM	I

diagram is shown of the various analyses and of the already known classes of graphs whose synchronization falls into Parallelism Principle applications (arrays, fixed oriented, information lossless).

Conclusion

A general principle has been presented which allows us to use the knowledge of the figure properties to speed up the synchronization.

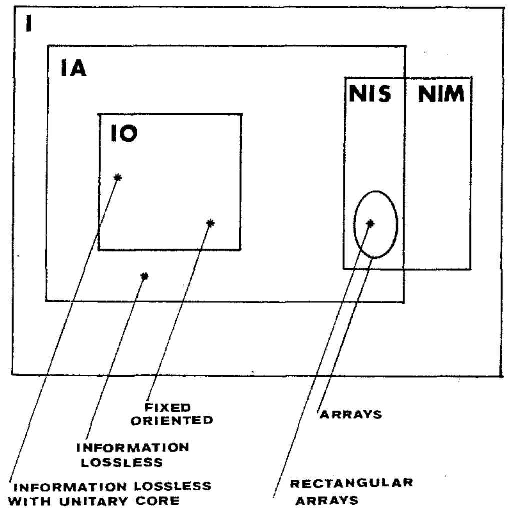


FIG. 7. Inclusion diagram of the various classes.

Five classes of applications of this principle have been introduced. It has also been shown how some well-known fast synchronization methods can be seen as Parallelism Principle applications.

ACKNOWLEDGMENTS

The author thanks Professor A. Grasselli and Professor A. Albano for their helpful suggestions and guidance.

RECEIVED: September 25, 1975; REVISED: March 31, 1977

REFERENCES

BALZER, R. (1967), An 8-state minimal time solution to the firing squad synchronization problem, *Inform. Contr.* **10**, 22–42.

GRASSELLI, A. (1975), Synchronization of cellular arrays: The firing squad problem in two dimensions, *Inform. Contr.* **28**, 113–124.

MOORE, E. F. (1964), The firing squad synchronization problem, *in* "Sequential Machines. Selected Papers" (E. F. Moore, Ed.), pp. 213–214, Addison–Wesley, Reading, Mass.

MOORE, F. R., AND LANGDON, G. G. (1968), A generalized firing squad problem, *Inform. Contr.* **12**, 212–220.

ROMANI, F. (1976), Cellular automata synchronization, *Inform. Sci.* **10**, 299–318.

ROSENSTIEHL, P. (1966), Existence d'automates finis capables de s'accorder bien qu'arbitrairement connectes et nombreux, *Internat. Comput. Centre Bull.* **5**, 215–244.

ROSENSTIEHL, P., FISKEL, J. R., AND HOLLINGER, A. (1973), Intelligent graphs: Networks of finite automata capable of solving graph problems, *in* "Graph Theory and Computing" (R.C. Read, Ed.), pp. 219–265, Academic Press, New York.

SHINAHR, I. (1974), Two and three-dimensional firing squad synchronization problems, *Inform. Contr.* **24**, 163–180.

WAKSMAN, A. (1966), An optimum solution to the firing squad problem, *Inform Contr.* **9**, 66–78.