A partial analysis of Go*

By Edward O. Thorp† and William E. Walden‡

A game called Computer Go is defined. Computer Go differs from the game of Japanese Go only in that certain imprecisely defined conventions have been replaced by precise rules. Some general theorems on Computer Go are given, as well as a scheme for analysing the game with the aid of a computer. Several reduced versions of Computer Go were analysed, and the resultant strategies are briefly described.

1. Introduction

The board game called Go originated in China. There is some doubt as to the exact date of its beginning, but the game was well known in the tenth century B.C. Go was introduced into Japan in approximately 754 A.D. In Japan the game has become so popular and well known that championship play is followed by the general public. For the most part, Go has been introduced into the United States by persons from Japan. Although Go does not enjoy the popularity of Chess in the United States, it is fast becoming a popular intellectual game. For a complete discussion of the history of Go one should examine Falkener (1961), Lasker (1960), Smith (1956).

There are several books (Goodell, 1957; Lasker, 1960; Morris, 1951; Smith, 1956; Takagawa, 1958) that are principally devoted to the analysis of various board situations in Go. One book (Rosenthal, 1954) is devoted to the analysis of various board situations for a reduced version of the game. At least one computer program (Remus, 1962) has been written which simulates a reduced version of the game and learns "good" strategies.

In this discussion we have attempted a complete analysis for reduced versions of a game which we call Computer Go. At the outset we realized that the large number of possible moves would restrict us to very small boards. However, it was our hope that such an attempt would yield some general results about the game. After describing the game and its rules, we present here some general results, and best strategies for various reduced versions of the game.

Computer Go differs from Go as it is played only in that certain "colloquialisms" (imprecisely defined conventions) have been replaced by precise rules. As far as we know, these alterations make no essential difference in the game, and results for Computer Go will agree with ordinary Go whenever the latter is unambiguous.

We were initially motivated by the thought that if we could find the value of $N \times N$ Go for small values of N (perhaps $N = 2, \ldots, 7$), then insight into the value of Go for larger N might be obtained. For example, suppose we learned Go was a win for Black when $N = 3, 4, 5$, and was a draw for $N = 6, 7, 8, 9$. This would suggest strongly that Go was a draw for all $N \geqslant 6$. (It of course might be true that the results for even N and those for odd N should be considered separately.)

2. Description of Go

We now give a brief description of the game of Go. For detailed descriptions of the game, we suggest Lasker (1960), Smith (1956).

Go is played on a board that is marked with nineteen equidistant line segments of equal length parallel with each edge. These lines produce 361 points of intersection. One player, called Black, has 181 black stones. The other player, called White, has 180 white stones.

A move consists of the placing of a stone on one of the vacant points of intersection. The first turn always goes to Black. A group of stones are captured and removed from the board when they are completely surrounded, both inside and out, by the opponent's stones. Fig. 1 has several examples of groups of black stones that have been captured and are ready for removal. A stone cannot be placed on a vacant point of intersection if by so doing the capture of the stone is caused. We were tempted to rephrase this as "Suicide is (morally) forbidden". A stone cannot be placed on a vacant point of intersection if such a move would cause the board to be identical to its configuration after the player's previous move.

A player is not required to move. He may choose to pass on his turn. However, the game is terminated after two consecutive passes.

At the end of the game, a player's final score is obtained by adding the number of captured stones in his possession and the number of vacant points that are surrounded by his stones. The player with the highest score wins.

* This research was supported in part by Grant NSF-G 25058, Grant AF-AFOSR 457-63, and by The Atomic Energy Commission.

† *Mathematics Department New Mexico State University.*
‡ *Los Alamos Scientific Laboratory, University of California, Los Alamos, New Mexico.*

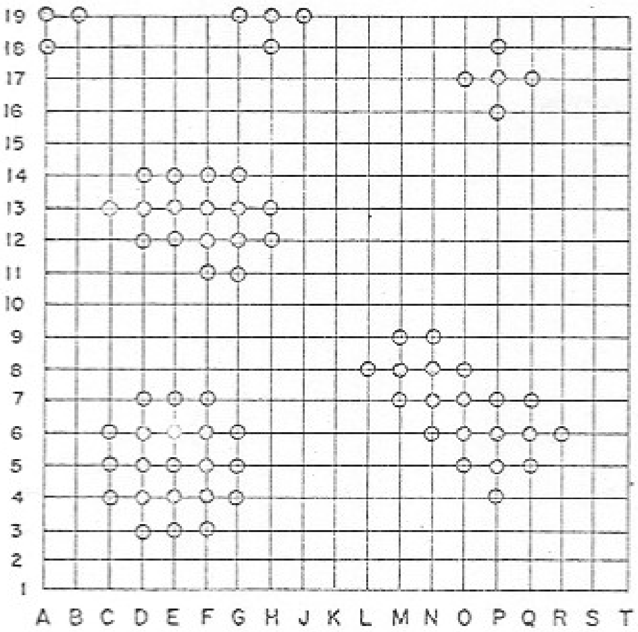

Fig. 1.—Groups of captured black stones

The rules above are clear enough, although they are not stated in precise mathematical terms. There are two situations where the rules are not precise. A situation can arise where both players can agree that one or the other has the advantage, and they stop making moves relative to this situation. The other case that may arise is one where the players are repeating a long sequence of moves without realizing it. In a case like this, the game may last so long that the players come to an agreement as to the outcome.

In formulating rules for the game of Go we have eliminated the situations above. For this reason, the game which we analyse will be called Computer Go. We now give the rules for this game.

Computer Go is a game between two players whom we will refer to as Black and White. Black plays on the odd turns and White plays on the even turns. We generalize the game to $N \times M$ rectangular boards, where N and M are arbitrary positive integers. For $n = 1, 2, \ldots, N$ and $m = 1, 2, \ldots, M$ consider the pairs (n, m) corresponding to the $N \times M$ matrix of points of intersection. For each turn t and each pair (n, m) there is an assigned occupancy value $a^t_{n,m}$. When play is ready to begin on turn t these values are assigned by the computer as follows:

$$a^1_{n,m} = 0 \text{ and } a^t_{n,m} = a^{t-1}_{n,m} \text{ for } t > 1.$$

A Black [White] move to (n, m) on turn t means set $a^t_{n,m} = 1[2]$. We note there that it is not always possible to move to any pair (n, m). The restrictions will be discussed later.

A Black [White] pass on turn t means Black [White] does not move on turn t.

The pair (n, m) is adjacent to the pair (i, j), denoted $(n, m)\ A\ (i, j)$, if one of the following holds:

(1) $n = i + 1$ and $m = j$
(2) $n = i - 1$ and $m = j$
(3) $n = i$ and $m = j + 1$
(4) $n = i$ and $m = j - 1$.

The pair (n, m) is equivalent to the pair (i, j), denoted $(n, m) \sim (i, j)$, if there exists a positive integer p and pairs (r_w, s_w), $w = 1, 2, \ldots, p$ such that $(n, m)\ A\ (r_1, s_1)\ A(r_2, s_2)\ A \ldots A(r_p, s_p)$ and $a^t_{n,m} = a^t_{r_1,s_1} = a^t_{r_2,s_2} = \ldots = a^t_{r_p,s_p} = a^t_{i,j}$. Note that $\sim$ is equivalence relation in the mathematical sense, and that there are three types of equivalence classes so defined. They are (1) Black connected groups, (2) White connected groups, and (3) connected groups of vacancies.

After a move or a pass has been made on Black [White] turn t, the pair (n, m) is said to be captured by Black [White] on turn t if all the following hold:

(1) $a^t_{n,m} = 2[1]$
(2) there exists a pair (i, j) such that $a^t_{i,j} \neq 2[1]$
(3) if, for all (p, q) and (k, l) such that $(p, q)\ A\ (k, l)$ and $(k, l) \sim (n, m)$, $a^t_{p,q} \neq 0$.

If (n, m) is captured, we set $a^t_{n,m} = 0$.

A move to the pair (n, m) by Black [White] is illegal in each of the following situations:

(1) $a^t_{n,m} \neq 0$
(2) (n, m) can be captured by White [Black] after a pass on turn $t + 1$
(3) a move to (n, m) would result in $a^t_{n,m} = a^{t-2}_{n,m}$ for $n = 1, 2, \ldots, N$ and $m = 1, 2, \ldots, M$.

For each turn t there is a number C_t, called the *capture count*, that is assigned as follows:

(1) if t is odd, then $C_t = C_{t-1} + Q$, where Q is the number of pairs captured by Black on turn t for $t > 1$ and $C_1 = 0$
(2) if t is even, then $C_t = C_{t-1} - Q$, where Q is the number of pairs captured by White on turn t.

The pair (n,m) is said to belong to Black [White] if all the following hold:

(1) $a^t_{n,m} = 0$
(2) there exists a pair (i, j) such that $a^t_{i,j} = 1[2]$
(3) if, for all (p, q) and (k, l) such that $(p, q)\ A\ (k, l)$ and $(k, l) \sim (n, m)$, $a^t_{p,q} \neq 2[1]$.

The game is terminated if two consecutive passes occur, where a pass may be by choice or because no move is available. If t was the last turn of play, then the game total S is given by $S = C_t +$ (number of pairs belonging to Black) $-$ (number of pairs belonging to White). If $S > 0$, then Black wins. If $S < 0$, then White wins. If $S = 0$, then the game is a draw.

The game is also terminated after turn t if there exists s such that $s < t$, where t and s are both odd or both even, and $a^t_{n,m} = a^s_{n,m}$, $a^{t-1}_{n,m} = a^{s-1}_{n,m}$ for $n = 1, 2, \ldots, N$ and $m = 1, 2, \ldots, M$. The condition $a^t_{n,m} = a^s_{n,m}$ is not sufficient because a move to a pair of one configuration may be illegal, but at the same time be legal in the other configuration. If $C_t > C_s$, then Black wins. If $C_t < C_s$ then White wins. If $C_t = C_s$, then the game is a draw. Note that this rule guarantees that the game tree for Computer Go is finite, because it prevents the game from going through endless cycles, in which the players run again and again through the same configuration, by assigning a reasonable value to this configuration.

3. General results

After writing an initial computer program to analyse Computer Go, we found that the number of possible moves was an even greater restriction than we had imagined. For example, one might think that the games of Tic Tac Toe and 3 × 3 Computer Go are of comparable complexity. This is not the case, for a Go player is allowed to pass on any given turn, adding an additional sub-branch to each branch of the game tree. Also, stones are captured in Go, creating vacant board positions which in turn increase the number of possible moves.

As a consequence most of our general results (all general results are for $N \times M$ rectangular Go unless otherwise stipulated) were motivated by an attempt to eliminate possible moves and increase the efficiency of the computer program. However, one theorem was motivated by the results of the initial program, so we state and prove it first.

Theorem 1: Black never has a forced loss in Computer Go.

PROOF: Assume that Black does have a forced loss, and White moves so as to force a Black loss. Then if Black passes on his first move, White will not pass, for this would make the game a draw. Hence White moves. But this means that Black could win by making the same first move, a contradiction which completes the proof.

Several players to whom we have talked have the erroneous impression that if Black begins by moving to the centre (assume $N \times N$ Go and that N is odd) and then continues to move opposite to White, then Black has at least a draw. This is false for odd $N \geq 5$, as shown by the idea indicated in Fig. 2.

A first step in eliminating possible moves was to consider a modified form of Computer Go which we now define. By T-truncated Go we mean Computer Go restricted to exactly T turns. If neither player has won when T turns have been completed, then the game is a draw. A corollary of the next theorem gives the basic relationship between Computer Go and T-truncated Go.

Theorem 2: In the finite game tree for Computer Go, suppose that draws are inserted at random at various points of the tree. Then the value of each remaining branch and the game is either unaltered or becomes a draw.

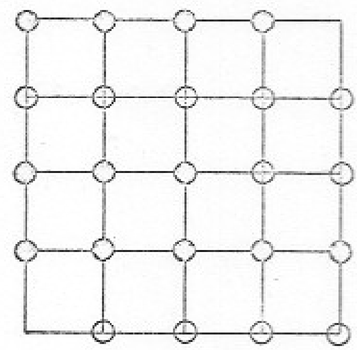

Fig. 2.—White moves to the lower left vertex and then has a winning position

PROOF: Assign the values +1, 0, and −1 to wins, draws and losses, respectively, for Black. To assign values to branches of the tree, we maximize values of Black turns and minimize values of White turns. We compute upwards from the bottom of the tree. This is possible since the game tree is finite. By a "Black Branch" we mean one which arises from a Black turn. Suppose a branch is given, and some of the sub-branches of this branch are reassigned the value 0. Assume the given branch is Black, so we want to minimize the values of the sub-branches. If the original value of the branch is +1, then the new value V of the branch satisfies $0 \leqslant V \leqslant 1$. If the original value of the branch is 0, then $V = 0$. If the original value of the branch is −1, then $-1 \leqslant V \leqslant 0$. Hence the new value V of the branch is either 0 or equal to the original value of the branch. This completes the proof.

Corollary 3: If Black wins in T-truncated Go, then Black wins in Computer Go.

We see from the corollary above that if we can find a winning strategy for Black in T-truncated Go, a much simpler game, then we have found a winning strategy for Black in Computer Go.

A corollary of the next theorem gives a way of eliminating many White moves.

Theorem 4: Suppose we are given a finite game tree. Let branches of the tree be removed at random and let new values be assigned at random to the tips of this pruned tree. Then the following holds:

(1) Suppose the new values $\{w_i\}$ are always greater than or equal to the original values $\{v_i\}$ and that $\min \{w_i\} = Y$, then if the value V of the original game satisfies $V \geqslant Y$, the value of the new game is also V. If the value of the original game satisfies $V < Y$, then the value W of the new game satisfies $Y \geqslant W \geqslant V$.

(2) Suppose the new values $\{w_i\}$ are always less than or equal to the original values $\{v_i\}$ and that $\min \{w_i\} = Y$. Then if the value V of the original game satisfies $V \leqslant Y$, the value of the new game is also V. If the value of the original game satisfies $V > Y$, then the value W of the new game satisfies $Y \leqslant V \leqslant W$.

PROOF of (1): Suppose a given branch of the original tree has value v, and some of the sub-branches of the given branch have had new values assigned. Let the original values be s_j and the new values be t_j. Let w be the new value of the given branch. Let $y = \max \{t_j\}$. Then $y \geqslant w \geqslant v$. If $v \geqslant y$ then $y = w = v$. Since the branch was arbitrary, the result holds for the game, completing the proof.

The proof of (2) is omitted due to its similarity to the proof of (1).

The next corollary follows at once from theorems 1 and 4.

Corollary 5: In the game tree for Computer Go or T-truncated Go, if a branch is a win or draw for Black, then if all sub-branches that are losses for Black are either completely removed or changed to draws, then the value of the branch remains unchanged.

So far the results that have been given are related to the game tree for Computer Go. The next two theorems are related to the actual status of the game. There are many special situations in the game where the players can recognize the outcome of the game. Two general conditions for this are now given.

Theorem 6: Suppose White has just played on turn t. Let B be the number of pairs (i, j) such that $a_{i,j} = 1$ and let W be the number of pairs (i, j) such that $a_{i,j} = 2$. If $NM - W + B < C_t$ then Black wins by passing on the remaining Black turns.

PROOF: Suppose Black passes on turn $t + 1$ and then White passes on turn $t + 2$. Then the game is over and the number of pairs belonging to White is at most $NM - W - B$. But by hypothesis we have $C_t + W > NM + B$ so $C_t - (NM - W - B) = C_t + W - (NM - B) > (NM + B) - (NM - B) = 2B \geqslant 0$. Thus Black wins.

Suppose Black passes but White continues to move. White would have a maximum number of pairs by moving once, capturing all of the pairs occupied by Black, and then passing on subsequent moves. If this is the case, the number of pairs belonging to White is at most $NM - W - 1$ and $C_{t+2} = C_t - B$. But by hypothesis $C_t-(NM-W+B)>0$, so $C_{t+2}-NM+W>0$ so $C_{t+2}-NM+W+1>0$. Hence $C_{t+2}-(NM-W-1)>0$, so Black wins. This completes the proof.

Theorem 7: Suppose Black has just played on turn t. Let B be the number of pairs (i, j) such that $a_{i,j} = 1$ and let W be the number of pairs (i, j) such that $a_{i,j} = 2$. If $-NM - W + B > C_t$ then White wins by passing on the remaining White turns.

PROOF: Suppose White passes on turn $t + 1$ and then Black passes on turn $t + 2$. Then the game is over and the number of pairs belonging to Black is at most $NM - B - W$. But
$C_t + NM - B - W < -NM - W + B + NM - B - W = -2W \leqslant 0$ so White wins.

Suppose White passes but Black continues to move. Black would have a maximum number of pairs by moving once, capturing all of the pairs occupied by White, and then passing on subsequent moves. If this is the case, the number of pairs belonging to Black is at most $NM - B - 1$, and $C_{t+2} = C_t + W$.

But $C_{t+2} + (NM - B - 1) = C_t + NM + W - B - 1 < -NM - W + B + NM + W - B - 1 = -1 < 0$. Hence White wins, completing the proof.

5. Strategies

We have found best Black strategies for 1×1 through 1×8, 2×2 through 2×4 and 3×3 Computer Go. For the $1 \times n$ games, where $1 \leqslant n \leqslant 8$ we found that the following are best Black results.

$1 \times 4n$:	Win
$1 \times (4n - 1)$:	Win
$1 \times (4n - 2)$:	Draw
$1 \times (4n - 3)$:	Draw.

For $3 \leqslant n \leqslant 8$, (1, 2) is a best first move for Black.

The best Black strategy for 2×2 Computer Go is a draw. The best Black strategy for 2×3 Computer Go is also a draw. In this game the best first play for Black is a pass. The best Black strategy for 2×4 Computer Go is a win. In this game Black should try to occupy opposite corners of the centre square. If White does not allow this, Black should fill one vertical side of the centre square. If White fills the other side of the centre square, a Black move to the upper right-hand corner (1, 4) will lead to a win for Black. If White does not fill the other side of the centre square, a Black move to one of the other corners of the centre square will lead to a win.

The best Black strategy in 3×3 Computer Go is a win for Black. One can show, by writing out cases, that Black wins if he occupies the entire middle row or middle column, and White occupies at most two positions. If we combine this principle and the fact that a connected group with one eye automatically has two eyes in 3×3 Go, we find that a simple winning strategy can be described as follows. Black begins with a move to the centre. After this move, Black has four ways in which to connect. White can only block two of these in his first two moves, so, after Black's third move, he will have always connected up and gained a winning position.

We have determined the complete set of moves for each of the strategies above, but they are too lengthy to be presented here.

We plan to try and find best strategies for larger versions of Computer Go in the future. Even 3×3 Go was too large for the computer when the "brute force" approach of completely evaluating the game tree was applied. After the general results of Section 3 were introduced, the computer became an aid in the solution of the problem. We expect to be able to solve the 3×4 case, but we believe that additional general results and a corresponding reduction of the problem will be needed in order to solve the 4 × 4 case on present-day computing machines.

The computer that was used to obtain these results was Maniac II at Los Alamos. This computer has 16,384 words of storage, a word being 48 bits in length. This computer has an add time of 19 microseconds, a multiply time of 160 microseconds, a divide time of 425 microseconds, and a shift time of 10 + 1·8 times the number of shifts in microseconds.

References

FALKENER, EDWARD (1961). *Games Ancient and Oriental and How to Play Them*, Dover, New York.
GOODELL, JOHN D. (1957). *The Game of Ki*, Riverside Research Press, St. Paul, Minnesota.
LASKER, EDWARD (1960). *Go and Go-Moku, the Oriental Board Games*, Dover, New York.
MORRIS, LESTER and ELISABETH (1951). *The Game of Go*, American Go Association, New York.
REMUS, H. (1962). "Simulation of a Learning Machine for Playing Go," Proc. IFIP Congress 62, Munich, North Holland Publishing Company, Amsterdam, pp. 192–194.
ROSENTHAL, GILBERT (1954). 13 *Line Go*, published by author, 4009 Liberty Heights Avenue, Baltimore, Md.
SMITH, ARTHUR (1956). *The Game of Go*, Charles E. Tuttle Co., Rutland, Vermont.
TAKAGAWA, KAKU (1958). *How to Play Go*, American Go Association.
—— (1958). *The Vital Points of Go*, American Go Association.

UNWIN BROTHERS LIMITED, WOKING AND LONDON

Addendum to A Partial Analysis of Go

In our recent paper "A partial analysis of Go" (this journal), there was a flaw in our formulation of the rules for computer Go. This flaw was pointed out to us by Dr. Joseph A. Schatz of the Sandia Corporation. It was our intention to make precise the somewhat colloquial rules for Go. We wished to formulate a set of rules that would be suitable for use in a high-speed computer. When the game was evaluated by these rules we wished to arrive at the same scoring (perhaps plus or minus a point) as did human players in actual games.

In particular, we introduced a "pass" rule that is different from what is actually done in the game. In actual games, players must both make active moves until, at some point, they both believe that further moves will serve no purpose. Then they both pass and the game is scored. We could see no way to arrange for the computer to directly imitate the human and arbitrary decision to end the game. Our pass rule was intended to give an alternative but game--theoretically equivalent procedure. According to our rule, a player can either pass or make an active move at any point in the game.

The flaw that Dr. Schatz pointed out arises via this pass rule. It might best be first illustrated by an example. Suppose that the simple configuration indicated in figure 1 has arisen. Suppose that it is now White's turn. Suppose also that the net capture count is 20 in favor of White. Our rules assign all 20 points of territory to Black, since he "surrounds" it. Thus, if the game were stopped now it would be scored a tie by the machine, using our rules.

Human players would also score it a tie. They might argue as follows. There is no point in White continuing to move because it is obvious that he cannot hope to build a living group against any *reasonable* opponent. (Black *could* kill his own group, in effect, by helpfully filling in the board until his group became vulnerable, but it is hardly to be expected that he *would* do this.) There is no reason for Black to continue moving, for to do so would merely fill in his territory and lose him points. So both players would pass and the game would end. (In an actual game, White would probably fill up Black's territory with the 20 Black prisoners which he had. This would make it easier to arrive at the final score of "tie" or "0".)

The machine and the human scorings agree. So far, so good. But now suppose that White moves so that the configuration becomes as in figure 2. Now Black has temporarily lost 10 units of territory. He must make an active move or lose (by 10 points). The only worthwhile active moves are ones directed at the capture of the White stone. But it will take at least three moves to do so. If White continues to pass, a configuration like that in figure 3 might arise. The White man has now been removed but Black has lost the game. This deviates from the results of an actual game.

To remedy this defect in our rules, we now modify the pass rule as follows. A player who passes must now pay one point for

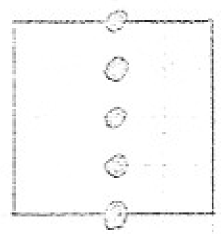

figure 1. The Black group has two eyes and can never be destroyed. White cannot build a living group.

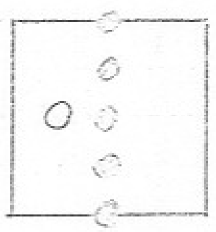

figure 2. One white move later.

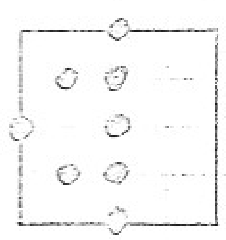

figure 3. After three additional Black moves and three White passes, the White stone is captured but Black has lost.

for the privilege. In other words, the capture count is adjusted against him by one point. In a human game, the player who passes could simply add one of his stones to his opponents group of prisoners. Two of these modified passes end the game, as before.

So far as we now know, the game thus modified is precisely equivalent in results to human Go, except perhaps for occasional one point discrepancies in scoring.

Edward O. Thorp
Mathematics Department
New Mexico State University
University Park, New Mexico

William E. Walden
Los Alamos Scientific Laboratory
University of California,
Los Alamos, New Mexico

For M x N Go, with M and N both odd, we define "symmetric play by black", or, briefly, "symmetric play", as follows. Black moves to the center on his first move. Thereafter, black attempts to move opposite the center from white, i.e. on every turn, Black trys to move in such a way as to insure that the center of gravity of the configuration of men (each assigned unit mass) is at the center of the board after Black completes his move. If White passes, Black passes. Of course White may eventually move in such a way that Black cannot do this, as when White captures a group of Black men and the loss of the group so changes things that the corresponding group of White men cannot now be captured.

Assuming best play by White, for which M and N does the strategy of symmetric play give Black at least a draw? If M = N = 1, the strategy cannot be followed since no one has a legal move. If M = 1, N = 3, the strategy wins for Black. I haven't yet checked out numerous additional small M or N cases. However, if M and N are both at least 5, White wins under best play. The idea is as in figure 2, page 205.

Figure 3, attached, shows how the idea extends readily to 5 x 7. To extend to 5 x N, N ≥ 7, just add pairs of columns, one on each side of the Y-axis, until a figure of the desired size is generated.

Figure 4 shows how to extend the figure in the M direction. The two extensions are independent of each other, so any desired M x N example, M and N ≥ 5, can be obtained.

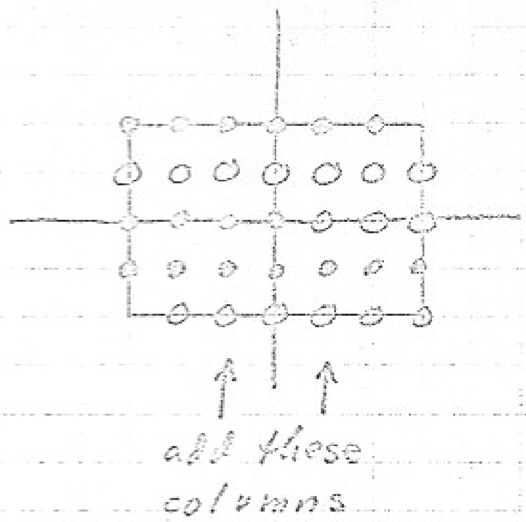

Fig. 3

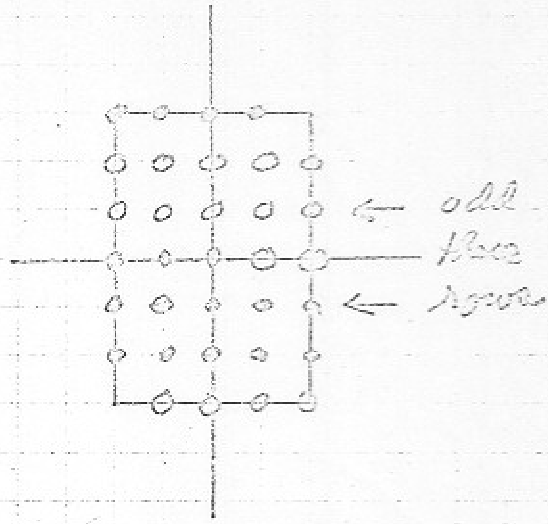

Fig. 4