Algorithms for Computation of Blackjack Strategies

by

David Heath
Department of Mathematics
University of Minnesota
(~1973)

I. Introduction.

The purpose of this paper is to present a collection of algorithms which together constitute an efficient and apparently new procedure for calculating optimal strategies for the game of blackjack when the remaining pack is "large". Blackjack strategies published so far for use in actual play have been simplified strategies which use only a small fraction of the information available to the player. The major reason for this is that far more information is available to the player than he could store (or use) efficiently.

Unfortunately, the procedures necessary for exact calculation involve quite a lot of computation — too much, for example, to contemplate using them to simulate play with optimal strategy on a computer so that one could obtain accurate estimates of how favorable a game blackjack is with optimal play — and certainly too complicated for programming on a micro-computer for use in actual play.

This paper presents algorithms which, under the additional assumption (which is of course not really satisfied in actual play) that the pack of cards remaining to be dealt is infinite (so that its composition is not further modified as cards are drawn from it), calculate exactly the optimal strategies and do so with very few computations and using very little storage. These algorithms are so simple, in fact, that I was able to calculate by hand the optimal strategies for dealer's up-card of ten or ace for a full pack under the rule (common in European casinos) that the dealer does not receive his second card until the players have played out their hands (so that if a player splits or doubles he loses his increased bet to a dealer's blackjack.) These computations required only a few hours to perform.

These algorithms are presented in Sections II, III, IV and V; Section VI presents a powerful strategy computed using these techniques and Sections VII and VIII give two additional applications.

II. Basic Model.

Since the algorithms presented can be used to calculate optimal strategies under a variety of different rules of play, we shall not specify the rules of play. We do presume (merely to be specific) that the dealer declares his blackjack before the player considers doubling or splitting.

We presume that the dealer has received a card (whose value we know) and that we know the composition of the pack from which further cards will be selected. Let $F(1), F(2), \ldots, F(10)$ denote the fractions of aces, twos,..., tens (including face cards) in the remaining pack. The import of our assumption about an infinite pack is that we shall assume in our calculations that $F(1), \ldots, F(10)$ remain fixed throughout the play of this hand.

The following observation is essential for the existence of the algorithms: there is an ordering for the possible totals of a blackjack hand having the property that if a card is added to a blackjack hand it changes its total to one appearing later in the ordering. We can thus number the possible totals (from, say, 1 to 32) as follows:

STATE	1	2	3	4	5	6
TOTAL	2H	3H	4H	5H	6H	7H

STATE	7	8	9	10	11	12
TOTAL	8H	9H	10H	11S	12S	13S

STATE	13	14	15	16	17	18
TOTAL	14S	15S	16S	17S	18S	19S

STATE	19	20	21	22	23	24
TOTAL	20S	21S	11H	12H	13H	14H

STATE	25	26	27	28	29	30
TOTAL	15H	16H	17H	18H	19H	20H

STATE	31	32
TOTAL	21H	OVER 21 .

(By the total of a hand we mean the number obtained as follows: Count 2's, 3's,...,10's at their face values; Jacks, Queens and Kings as 10 , and if there are any aces in the hand, count one of them as 11 and the others as 1 unless this results in a total over 21 , in which case count them all as 1 . If there are no aces being counted as 11 the total is "hard", denoted "H" ; if there is an ace being counted as 11 the total is "soft", denoted "S" .)

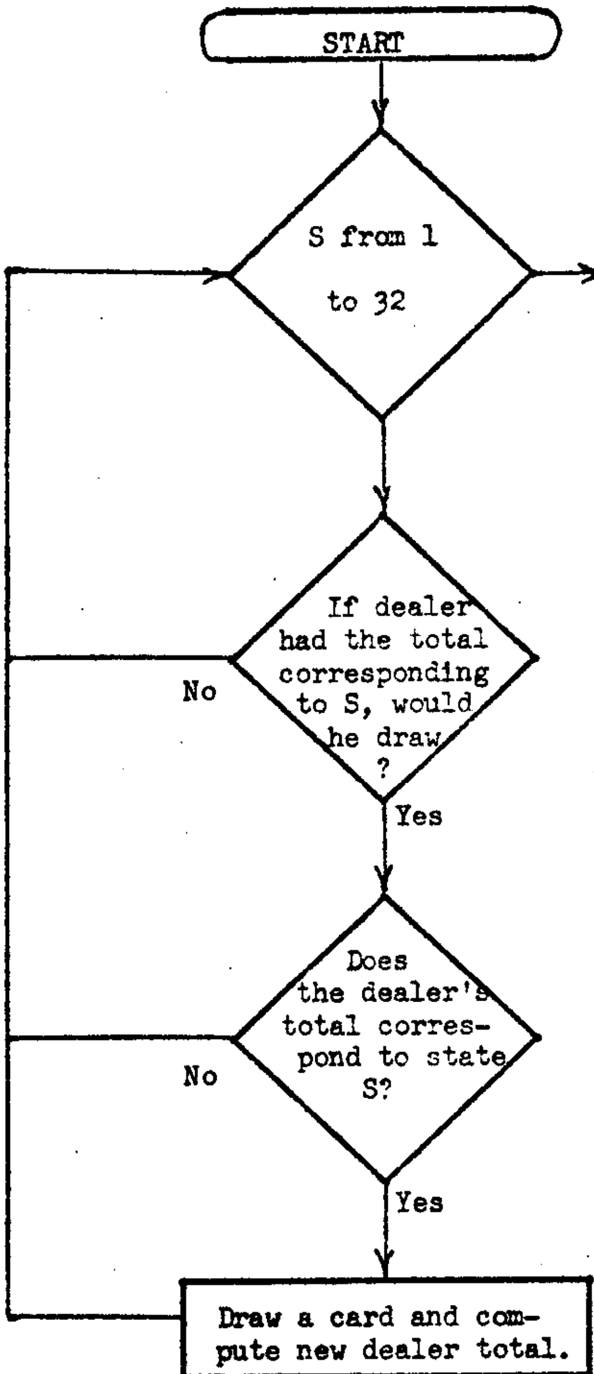
Notice that "blackjack" is not listed in the above ordering; it is thought of as being included in 21S . This is eaily taken care of when necessary.

III. Calculation of Dealer's Probabilities.

We wish now to calculate, for a given value of the dealer's up-card, the probabilities of the dealer's final total being 17(H or S), 18, 19, 20, 21 (but not blackjack), blackjack, and over. As mentioned above, we first lump blackjack with 21S and later correct for that.

The procedure which the dealer follows in drawing additional cards is of course completely specified by the rules and can be described in a very simple fashion; we choose here to consider instead an apparently more complicated description of this procedure. The advantage of this description is that it leads to a very simple and efficient algorithm for calculation of the dealer's probabilities. To present the description we use a flow chart; our symbols are consistent with those of Kemeny and Kurtz [1]. The description appears in Figure 1.

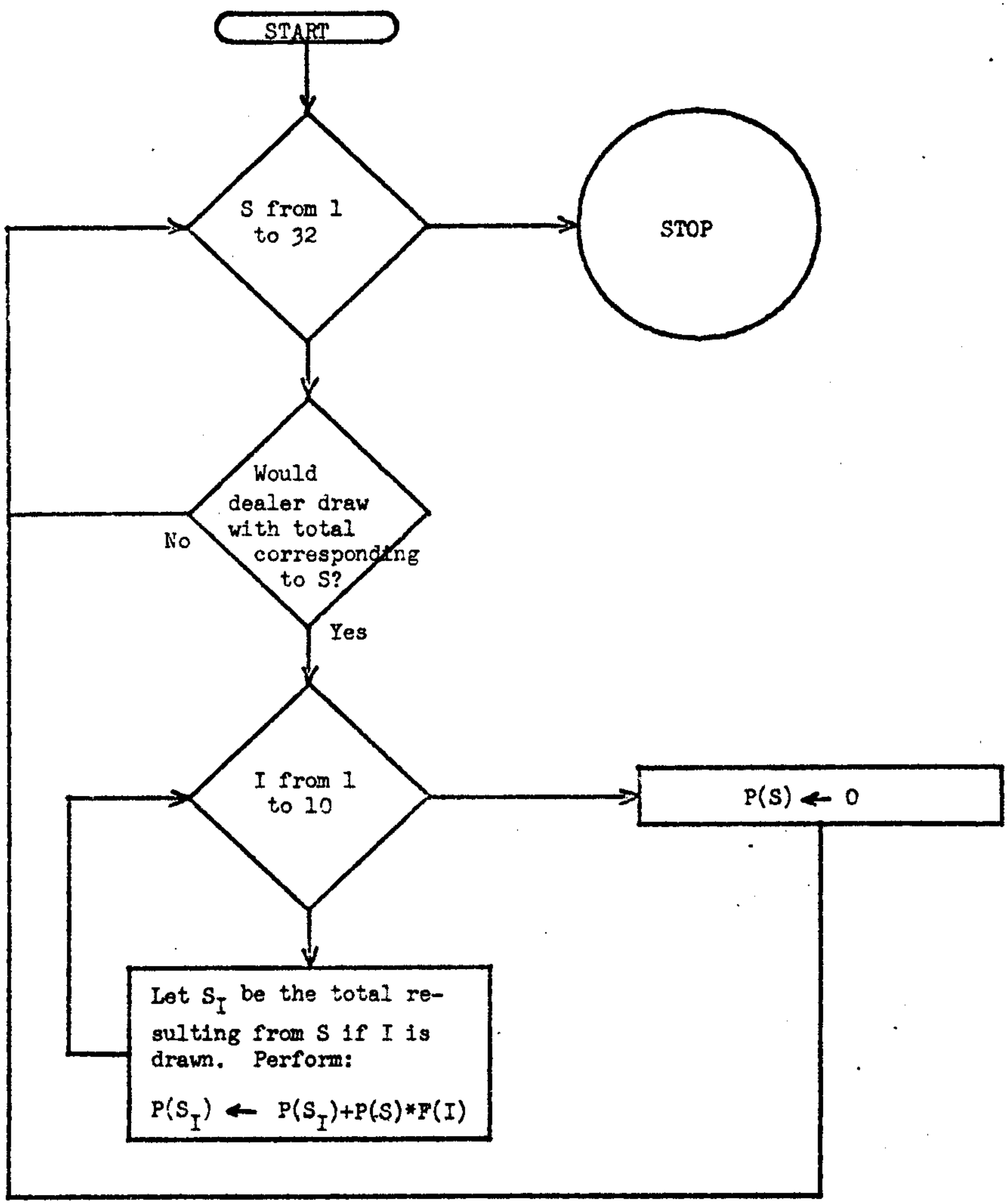
Figure 1: Algorithm for Dealer Drawing



This procedure is clearly equivalent to the procedure actually followed, in the sense that it yields the same behavior. Notice that we can let the dealer "decide" whether or not to draw his hole card too, since according to this algorithm he will always decide to draw it.

For the following procedure, we let P(1),P(2),...,P(32) be the probabilities that (at some particular stage in the procedure) the dealer's totals corresponds to state 1,2,...,32 . We begin the procedure with the dealer having only his up-card, so all this P(S)'s but one are zero. The P(S) corresponding to the total he has with only one card is of course equal to one. Fugure 2 gives a procedure of keeping track of these probabilities as the dealer carries out his drawing.

Figure 2: Algorithm for Calculation of Dealer's Probabilities



Of course there are obvious possibilities for speeding up these algorithms; for example instead of letting S run from 1 to 32 one can let S run from the state corresponding to the dealer's up-card to 26 . On the average there will be 21.5 values of S ; since for some of these values the dealer does not draw it is easily seen that this procedure requires approximately 165 multiplications and several hundred additions (including those to increment I and to obtain new totals resulting from drawing.) Even on a relatively slow machine this does not take very long to carry out.

IV. Calculation of Optimal Standing - Drawing Decisions.

We now proceed to calculate the correct decisions for various situations. We suppose now that the dealer has checked for blackjack and does not have it. We therefore modify the probabilities calculated in the previous section in the following way:

Based on the preceeding calculations we know the probabilities that the dealer will end up with 17 (H or S), 18, 19, 20, 21 (including blackjack) and OVER 21 . It is trivial to calculate the probability that he has blackjack: if his up-card is not an ace or a ten the probability of blackjack is 0 ; if he has a ten up the probability of blackjack is F(1) ; and if he has an ace up the probability of blackjack is F(10) .

We then subtract the probability of blackjack from the probability he has 21 and divide all the probabilities by (1 - probability he has blackjack). This gives the conditional probabilities of the various totals given that he doesn't have blackjack.

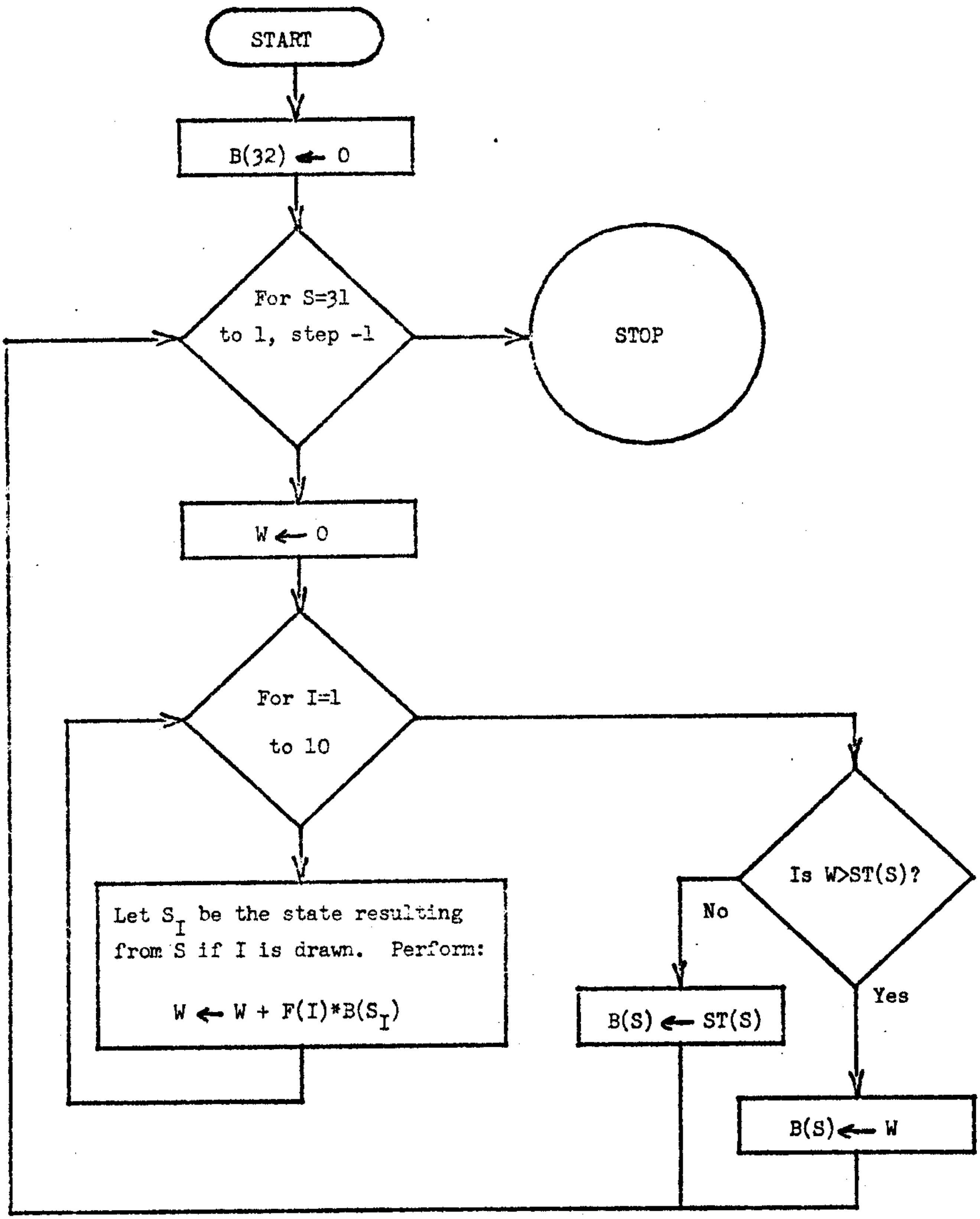
For each state S we define ST(S) to be the probability that the player wins if his total corresponds to state S and he stands. For example, if S = 4 (which corresponds to a total of 5H) ST(S) is the probability that the dealer's total is OVER 21 since this is the only case in which the player wins. To handle the possibility of a stand-off we introduce the artifice of supposing that, instead of calling off the bet, a fair coin is tossed to determine who wins. Thus ST(17) , the probability that the player wins if he stands with a total of 18S , is equal to the probability that the dealer's total is 17 or OVER 21 plus one-half the probability that the dealer's total is 18 . (Since by "optimal" we mean "optimal in the sense of expected value" and this artifice does not change the expected value, it is harmless.) Thus we know $ST(1)$, $ST(2),\ldots,ST(32)$. (Of course $ST(32) = 0$.)

We now wish to calculate numbers $B(1)$, $B(2),\ldots,B(32)$ where $B(S)$ is the probability that the player, having current state S , wins if he uses the best possible standing - drawing decisions. <u>If we can calculate these</u> -- the standing - drawing decisions are easy to obtain: If the player's total corresponds to state S , he should stand if $B(S) = ST(S)$ and should draw if $B(S) > ST(S)$.

To calculate $B(S)$ we reason as follows: Suppose we knew $B(T)$ for all $T > S$. If we draw and then play "the best way", it is easy to calculate the probability that we win. If this is larger than $ST(S)$ we use this as $B(S)$; otherwise we set $B(S) = ST(S)$.

Thus we need to calculate "backwards". Clearly $B(32) = 0$. Figure 3 gives a procedure for calculating $B(S)$ for $S = 31, 30, 29,\ldots,1$.

Figure 3: Algorithm for Computation of B(S)



V. Calculation of Optimal Splitting and Doubling Decisions.

Once the values of $B(S)$ and $ST(S)$ are known it is easy to calculate the optimal doubling decision: Suppose the current total corresponds to state S. Without doubling, the expected winnings for a unit bet would be $2\,B(S) - 1$. To evaluate the probability of winning with doubling we simply sum $F(I)\,ST(S_I)$ where S_I is the state resulting if card I is drawn. Since the bet is then two units, the expected winnings are $4 \sum_{I=1}^{10} F(I)\,ST(S_I) - 2$, so doubling is good if

$$\sum_{I=1}^{10} F(I)\,ST(S_I) > \frac{1}{2}\,B(S) + \frac{1}{4}.$$

If doubling after splitting is allowed we should build an array $D(S)$ such that $2\,D(S) - 1$ is the expected winnings starting at state S is the player does the best possible of doubling, drawing, and standing. Clearly this is easy to do. If doubling is not allowed after splitting, it is not necessary.

For splitting there are two possibilities: either re-splitting is allowed or it is not. We discuss only the first case here (the second is also easy but leads to a quadratic equation.) Suppose that the player holds two equal cards, each of which correspond to state S, whose total corresponds to state S'. Then splitting is good provided $4\,B(S) - 2 > 2\,B(S') - 1$. (If when splitting aces only one additional card is given the condition for splitting aces is $4\,ST(S) - 2 > 2\,B(S') - 1$). If doubling after splitting is allowed, D should be used in place of B in these formulae.

VI. Use of the Algorithms to Compute a Strategy.

The author first developed algorithms similar to (but more complicated and less efficient than) those presented here is conjunction with Professor Bert Fristedt of the University of Minnesota in 1971. These algorithms were used to calculate a powerful (but hard-to-use strategy based upon the following:

Let A = (# of small cards seen) - (# of tens seen)

B = (# of middle cards seen) - (# of tens seen)

C = # of aces remaining (i.e., unseen)

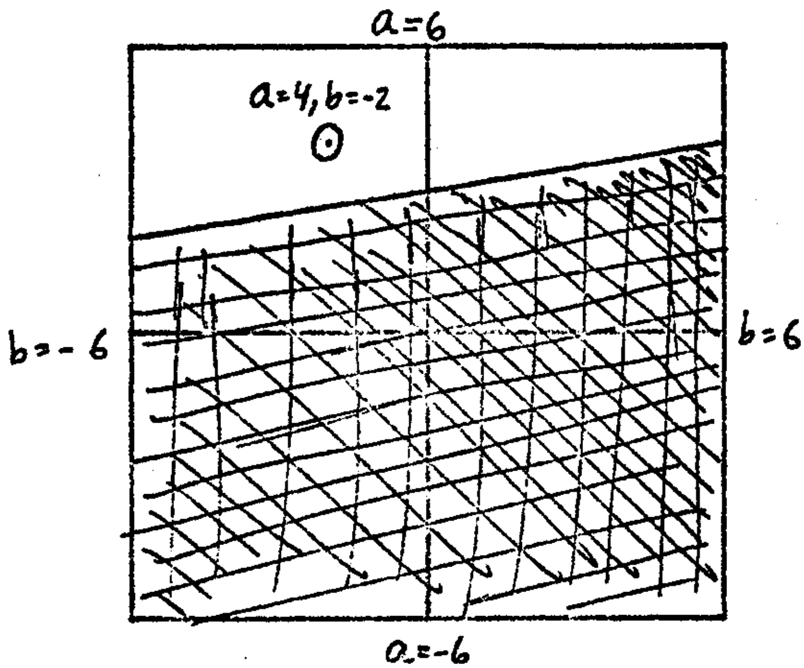
H = # of half-decks remaining (approximate).

Let $a = (A/H)$, $b = (B/H)$, and $c = (C/H) - 2$. The player counts A and B in his head, and C on his unused hand; he estimates H .

Strategy decisions are based on a and b alone. For each possible situation a graph is drawn in which a region is shaded. The following example illustrates the use of the graphs:

Suppose $A = 2$, $B = -1$, $H = 1/2$ (13 cards unseen) and $C = 2$. Then $a = 4$, $b = -2$, and $c = 3$. Suppose the dealer has a 9 up and the player has a 5 and a 10 . Splitting and doubling are out of the question so we consider drawing. The relevant box on the chart is

. (In the chart, dealer's up-cards are listed across the top and player's situations are listed along the sides. If a box is missing then "basic strategy" is correct in this situation.) Here's how to interpret this box:



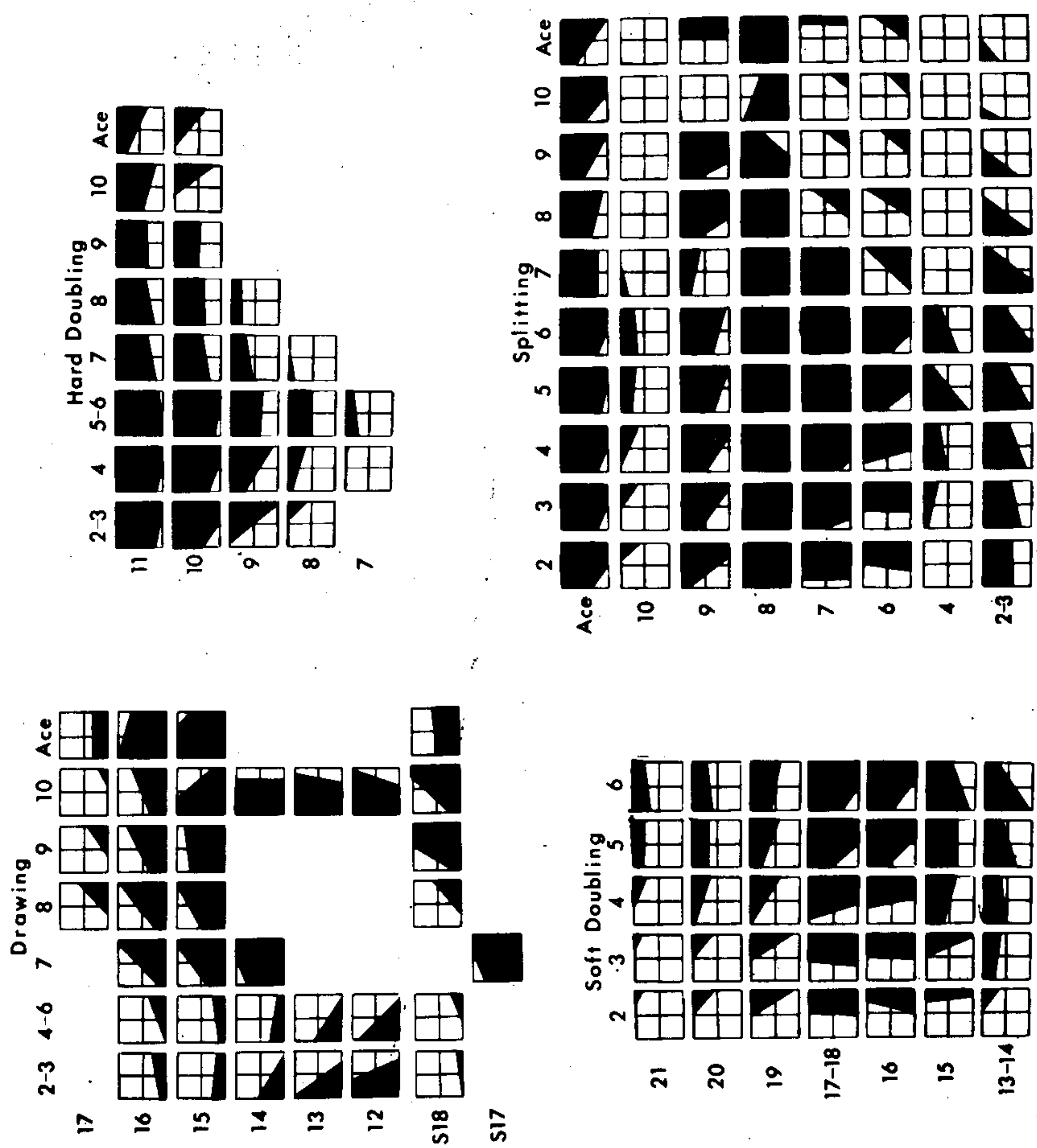
The a-axis runs up-and-down and the b-axis runs side-to-side. The values on the axis run from -6 to $+6$. The point $a = 4$, $b = -2$ is shown.

Since this point lies in the unshaded region the player should not draw. (The shaded region is the region in which the player should do what is indicated in the heading on the chart.) Note that there are compositions of the deck for which one would stand with 14 against a dealer 10 but draw with 15 .

The player's percent advantage is approximately $a + c$. The player should take insurance if $A + B > C$.

This system was sucessfully tested by Bert Fristedt and the author in 9 days (18 man-days) of play in Las Vegas.

Figure 4: A Two-Parameter Strategy



VII. Simulation Using the Algorithms.

One would expect that the strategy calculated using the "infinite deck approximation" is better than the basic strategy (see Thorp [2]) as the deck varies. To check this a (too -) modest simulation was undertaken. Three hundred hands were played (in simulation; one pack, head-on, with shuffle before a deal if ten or fewer cards remained) and whenever the basic strategy and the infinite deck approximation differed the calculated advantage of using the infinite-deck strategy was recorded. The results were not significant since the average gain in expectation per hand was .02 while the standard deviation of the observed data was .08 . In 300 hands with flat bets (1 unit per hand) the winnings were 31.50 , but the standard deviation for this number is 17 . More work is needed to determine the gain from using this strategy. (This may not be too expensive. Simulation of 300 hands using a not-too-efficient program on a CDC 6600 took 30 seconds.)

Note that the standard error of the .02 estimate is thus $\frac{.08}{\sqrt{300}} \cong \frac{.08}{17} \cong .005$, so the result ~~is~~ significant. D. H.

VIII. One Other Use.

In the past few years several semiconductor manufacturers (Intel, Motorola, National, RCA, Signetics, etc.) have introduced microprocessors. Since the algorithms presented here require the storage of very few numbers (fewer than 128 , say) it would be quite possible to design a small micro-computer for use in actual play. With only little ingenuity one can imagine input and output devices suitable for covert use of such a machine. The first people to make such a device might do very well indeed!

REFERENCES

[1] Kemeny, J. G., and Kurtz, T. E., Basic Programming, John Wiley and Sons, Inc., New York, 1967.

[2] Thorp, E. O., Beat the Dealer, Random House, New York, 1966.