Optimal Analyses for $3 \times n$ AB Games in the Worst Case

Li-Te Huang and Shun-Shii Lin

Department of Computer Science and Information Engineering,
National Taiwan Normal University, No. 88, Sec. 4,
Ting-Chow Rd., Taipei, Taiwan, ROC
linss@csie.ntnu.edu.tw

Abstract. The past decades have witnessed a growing interest in research on deductive games such as Mastermind and AB game. Because of the complicated behavior of deductive games, tree-search approaches are often adopted to find their optimal strategies. In this paper, a generalized version of deductive games, called $3 \times n$ AB games, is introduced. However, traditional tree-search approaches are not appropriate for solving this problem since it can only solve instances with smaller n. For larger values of n, a systematic approach is necessary. Therefore, intensive analyses of playing $3 \times n$ AB games in the worst case optimally are conducted and a sophisticated method, called *structural reduction*, which aims at explaining the worst situation in this game is developed in the study. Furthermore, a worthwhile formula for calculating the optimal numbers of guesses required for arbitrary values of n is derived and proven to be final.

1 Introduction

With the rapid increase in the need of encryption, it becomes urgent to develop an efficient mechanism of cryptanalysis. A kind of cryptanalysis, *differential cryptanalysis*, however, bears resemblance to deductive games in accordance with the analysis by Merelo-Guervos *et al.* [1]. In other words, deductive games can be regraded as abstract models of cryptanalysis problems and any results of deductive games may be applied to cryptography or related combinatorial optimization problems.

There are two players invloved in a deductive game. They are called the *codemaker* and the *codebreaker*, respectively. In the beginning of the game, the codemaker thinks of a secret code in mind and afterwards, the codebreaker tries to identify this secret code by guessing continuously. As long as the codebreaker makes a guess, the codemaker will give him[1] a response to describe the similarity between this guess and the secret code. The mission of the codebreaker is to obtain the code and minimize the number of guesses required at the same time.

More precisely, an $m \times n$ deductive game means that each possible secret code in the game is composed of m digits while every digit has n possibilities

[1] For brevity, we use 'he' and 'his' whenever 'he or she' and 'his or her' are meant.

J. van den Herik and P. Spronck (Eds.): ACG 2009, LNCS 6048, pp. 170–181, 2010.

(symbols). Without loss of generality, the set of these n symbols is defined as $S = \{0, 1, 2, ..., n-1\}$. Assumee that the codemaker has a secret code $c = c_1c_2...c_m$ in mind and the codebreaker makes a guess $g = g_1g_2...g_m$, where $s_i, g_j \in S, \forall i, j$. Then, the codemaker will give a response $[A, B]$, where A and B are defined as follows.

– $A = |\{i : c_i = g_i\}|, \forall i = 1, ..., m$. Thus, A means the number of symbols which appear in both c and g and meanwhile, every symbol occupies the same position in both c and g.
– $B = \sum_{j=0}^{n} \min(p_j, q_j) - A$, where $p_j = |\{i : c_i = j\}|$ and $q_j = |\{i : g_i = j\}|$. In other words, B represents the number of symbols which occur in both c and g but the positions of these symbols in c and g do not match.

Besides the above definitions, there is one additional characteristic to distinguish two families of deductive games. One of the two is Mastermind, in which repeated symbols are allowed in a secret code. The other is AB game, in which all symbols within a code are distinct. Note that the numbers of all possible responses given by the codemaker and all possible guesses the codebreaker can make are calculated as follows.

– For an $m \times n$ deductive game, the codemaker may give one of these responses which are $[m, 0], [m-1, 0], [m-2, 2], [m-2, 1], [m-2, 0], ..., [m-i, i], ..., [m-i, 0], ..., [0, m], ..., [0, 0]$. So, the total number of responses is $1 + 1 + 3 + 4 + 5 + ... + (m+1) = m(m+3)/2$.
– Obviously, there are n^m secret codes in Mastermind and $n!/(n-m)!$ codes in AB game.

In this study, $3 \times n$ AB games are investigated and analyzed, i.e., the case of $m = 3$ is discussed here. Hence, the number of all possible responses is 9 and these responses are [3, 0], [2, 0], [1, 2], [1, 1], [1, 0], [0, 3], [0, 2], [0, 1], and [0, 0] respectively. The number of all possible secret codes equals to $n(n-1)(n-2)$ as well.

For example, assume that the codemaker choose $c = 215$ as a secrete code and meanwhile, the codebreaker makes a guess $g = 012$. Then, the codemaker will offer a response $[1, 1]$.

This paper consists of six major parts. In Section 2, previous surveys are conducted. Section 3 provides some terminologies and notations. The optimal guess for the codebreaker in each turn is analyzed in Section 4. The worst situation caused by the codemaker is discussed in Section 5. Section 6 derives a theorem, which can calculate the optimal number of guesses for $3 \times n$ AB games in the worst case and some conclusions are also given.

2 Preliminaries

Two well-known deductive games are Mastermind and AB game, of which the dimensions are 4×6 and 4×10, respectively. The former is popular in America while the later is widespread in England and Asia. AB game is called “Bulls and Cows” in some places as well.

There have been much research on Mastermind and AB game over past several decades since Knuth [2] first investigated them. Knuth also proposed a worst-case optimal strategy of Mastermind, where the maximum number of guesses is 5. Meanwhile, its expected-case number of guesses is 4.478. Later, many studies for finding better strategies of Mastermind in the expected case were conducted. For example, Irving [3], Neuwirth [4], and Norvig [5] improved the expected-case strategies, in which the required guesses are 4.369, 4.364, and 4.47 in average, respectively. Koyama and Lai [6] demonstrated an optimal strategy in the expected case for Mastermind eventually while the expected number of guesses is about 4.34. Rosu [7] subsequently proposed an alternative algorithm to obtain its optimal strategy as well. Afterwards, a new heuristics for Mastermind was suggested by Barteld [8] and outperformed the conventional heuristics. Recently, an advanced framework to seek the optimal strategy for Mastermind was suggested by Huang *et al.* [9] as well, and this algorithm is innately superior to traditional ones since branch-and-bound pruning is adopted. There were also other approaches that emphasized the efficiency of determining good strategies such as Shapiro [10] and Rosu [7]. However, the qualities of solutions may be often worse than those of sophisticated methods because the simple approaches may not take sufficient time to consider all possible strategies carefully. In research on 4×10 AB game, Chen *et al.* [11] first obtained an optimal strategy in the worst case and showed that the maximum number of guesses is 7.

For the deductive games with higher dimensions, meta-heuristic algorithms are usually developed to solve them. For instance, Kalisker and Camens [12], Singley [13], Chen *et al.* [14], and Berghman *et al.* [15] proposed several meta-heuristic approaches to deal with Mastermind with different dimensions. One crucial research worthy of mention is Chen *et al.* [14]. This might be a promising result as it is the first approximate approach to achieve a near-optimal result for 4×6 Mastermind in the expected case. Although these methods often operate efficiently and effectively, they are not able to guarantee to attain optimal strategies.

For those deductive games with smaller dimensions, tree-search approaches or heuristics are often adopted to find their optimal strategies. However, these common methods are not appropriate for solving deductive games with higher dimensions. Hence, a systematic theoretical analysis, called *graph-partition approach*, was used by Chen *et al.* to investigate $2 \times n$ AB games [16] and $2 \times n$ Mastermind [17]. Optimal results of the two games in the worst and expected case were obtained eventually. Goddard [18] offered some basic discussions of $m \times n$ Mastermind and also attained the optimal numbers of guesses for $2 \times n$ Mastermind in both the worst and expected case, which were basically the same as those in Chen *et al.* [17].

3 Terminologies

Before $3 \times n$ AB games is discussed formally, some terminologies and notations have to be explained first in order to describe the analyses precisely. Thus, some terms are defined as follows.

Definition 1. *A secret code is* ***eligible*** *if it is compatible with all guesses and the corresponding responses given so far.*

Definition 2. *A set, which contains some eligible codes, is referred to as a* ***state****.*

Definition 3. *The state with only one eligible code, which has also been guessed by the codebreaker now, is defined as a* ***final state****. That is to say that the secret code has been identified and the game is over.*

Definition 4. *Let C_1 and C_2 denote two states. We say that C_1 is* ***harder*** *than C_2 if identifying a secret code in C_1 requires more guesses than that in C_2. In other words, the* ***difficulty*** *of a state means how many guesses the codebreaker requires to identify a secret code.*

Definition 5. *A strategy of responses taken by the codemaker is called a* ***devil's strategy*** *or an* ***adversary response*** *if this strategy maximizes the number of guesses required by the codebreaker.*

Definition 6. *Assume that there are two states, which are C_1 and C_2 respectively. If there exists a one-to-one function r such that each secret code in C_1 maps another one in C_2 and preserves the structure of C_1, then we say that C_2* ***dominates*** *C_1. Furthermore, r is called a* ***structural reduction****. In symbols, we write $C_1 \leq C_2$.*

Now, 3 × 5 AB game is taken into account as an illustrative example. Assumee that the set of five symbols in this game is $S = \{0, 1, 2, 3, 4\}$. If the codebreaker makes a guess, 012, and the codemaker responses [2, 0] in the first turn, the eligible codes are therefore 013, 014, 032, 042, 312, and 412 after the first turn. The set $C_{[2,0]} = \{013, 014, 032, 042, 312, 412\}$ forms a state. From the result of the later experiment, which conducts an exhaustive search to 3×5 AB game, the number of guesses required is maximum if the codemaker implements a devil's strategy to provide the response, [0, 2], at the first response. In contrast, $C_{[2,0]}$ and the state, $C_{[1,0]} = \{043, 034, 432, 342, 314, 413\}$, which is produced when the codemaker responses [1, 0] at the first response, are then considered. Notice that the elements in $C_{[2,0]}$ are of the forms, $01b$, $0b2$, or $b12$, where $b \in B = \{3, 4\}$. Thus, we define a structural reduction of r as

$$r : \begin{cases} 01b \mapsto 0zb \\ 0b2 \mapsto zb2, \text{ where } b \in B \text{ and } z \in B - \{b\} \,. \\ b12 \mapsto b1z \end{cases}$$

Figure 1 exhibits the mapping of each code in $C_{[2,0]}$ in detail. Note that the mapped codes in $C_{[1,0]}$ preserve the structures of those in $C_{[2,0]}$. This implies that finding a secret code in $C_{[1,0]}$ is *as hard as* or *harder* than that in $C_{[2,0]}$. Intuitively, this is also obvious since there is one more identified symbol in $C_{[2,0]}$ than in $C_{[1,0]}$. Hence, we say that $C_{[1,0]}$ dominates $C_{[2,0]}$. Furthermore, the structural reduction has the property of the transitive relation obviously. That is to say that $C_1 \leq C_3$ if $C_1 \leq C_2$ and $C_2 \leq C_3$.

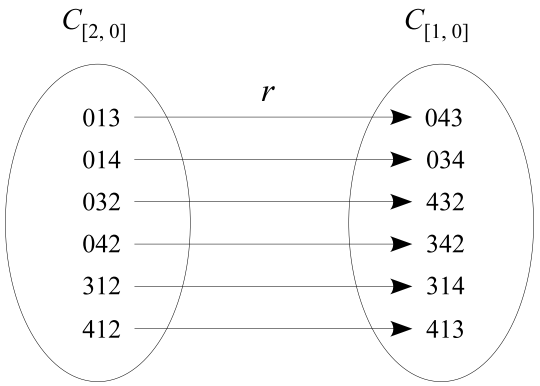

Fig. 1. Mapping from codes in $C_{[2,0]}$ to those in $C_{[1,0]}$ for 3 × 5 AB game

4 Analyses of the Optimal Guesses for the Codebreaker

In this section, the analyses of the optimal guess in each turn for the codebreaker are provided. First, a special kind of states C^* will be analyzed to determine the best guess for the codebreaker when he encounters this kind of states. Then, the discussion in the next section will reveal that the special states that are discussed here just match the attribution of states resulting from the devil's strategy for the codemaker. Consequently, our conclusions are attained finally.

Before the formal discussion, a critical concept should be clarified first. Intuitively, the more secret codes a state has, the harder the codebreaker identifies a secret code in it. However, the rule is not absolutely correct especially when the size of one state is very close to that of the other. Hence, the structural reduction is adopted to determine the difficulties of two states instead of simply comparing their sizes in the following discussion.

Assume that $S = \{0, 1, 2, ..., n-1\}$ represents the set of symbols appearing in $3 \times n$ AB games. The set, $B = \{b_0, b_1, ..., b_{h-1}\}$, is a subset of S, where $b_i \in S$ and $|B| = h, 3 \leq h \leq n-3$. Moreover, another set, A, is defined as $A = S - B = \{a_0, a_1, ..., a_{n-h-1}\}$, of which the cardinality is $(n-h)$.

Assume that there is a special state, called C^*, which consists of the secret codes that are all possible permutations of h symbols in B. In other words, the special state has $h\,(h-1)\,(h-2)$ secret codes in it. This state may be regarded as a subproblem of a $3 \times n$ AB game, i.e., a $3 \times h$ AB game. Notice that the symbols in A do not appear in the codes of the special state because of the definition of C^*. We can intuitively treat the symbols in A as those eliminated from previous responses made by the codemaker.

Now, imagine a scenario where C^* is encountered by the codebreaker during the process of playing a $3 \times n$ AB game. Since any symbols in S may be used in a guess made by the codebreaker for a $3 \times n$ AB game, all possible guesses for the codebreaker can be classified into four types according to the numbers of symbols that belong to A and B. Thus, the four types of guesses for the codebreaker are listed and discussed as follows. Here we assume that $a_i, a_j, a_k \in A$ and $b_i, b_j, b_k \in B$.

1. $a_i a_j a_k$
   All symbols of this type of guesses belong to A. If the codebreaker makes this kind of guesses, all eligible codes are then classified into the substate, $C_{[0,0]}$, trivially. So, the guesses of Type 1 are redundant and non-optimal results will be obtained if the codebreaker chooses this kind of guesses.
2. $b_k a_i a_j, a_i b_k a_j,$ and $a_i a_j b_k$
   The guesses of Type 2 contain two symbols in A and one symbol in B. This type of guesses can be further divided into three kinds of guesses such as $b_k a_i a_j, a_i b_k a_j,$ and $a_i a_j b_k$ in accordance with their positions of symbols. Without loss of generality, $g = b_k a_i a_j$ is taken to conduct the following analyses. The discussions of the other two can be undertaken in a similar way. Three nonempty substates, which are $C_{[1,0]}, C_{[0,1]},$ and $C_{[0,0]}$, are produced as the codebreaker makes the guess, g. Note that their cardinality are $(h-1)(h-2), 2(h-1)(h-2),$ and $(h-1)(h-2)(h-3)$, respectively. Now, we can show that $C_{[0,1]} \leq C_{[0,0]}$ and $C_{[1,0]} \leq C_{[0,0]}$ if $h \geq 5$.

   Lemma 1. *If the codebreaker encounters the state, C^*, and then makes the guess, $g = b_k a_i a_j, a_i b_k a_j$, or $a_i a_j b_k$, where $a_i, a_j \in A$ and $b_k \in B$, then $C_{[0,0]}$ dominates $C_{[0,1]}$ and $C_{[1,0]}$ if $h \geq 5$.*

   Proof. In order to prove that $C_{[0,1]} \leq C_{[0,0]}$, a structural reduction, r_1, is defined as

$$r_1 : \begin{cases} b_p b_k b_q \mapsto b_p z_1 b_q, \text{ where } b_p, b_q \in B' = B - \{b_k\} \\ b_p b_q b_k \mapsto b_p b_q z_2 \quad \text{and } z_1, z_2 \in B' - \{b_p, b_q\}. \end{cases}$$

   From r_1, it reveals that the structures of the secret codes, which are $b_p?b_q$ and $b_p b_q?$, are preserved after mapping. Note that $b_p z_1 b_q$ and $b_p b_q z_2$ should be distinct to reserve the property of one-to-one mapping. On the one hand, we can achieve this by assigning the symbols of z_1 and z_2 carefully while mapping is conducted. On the other hand, there should be two symbols left for the assignments of z_1 and z_2 once b_p and b_q have been fixed during the mapping. The proof is therefore correct if $h \geq 5$. The proof of $C_{[0,1]} \leq C_{[0,0]}$ is finished now. Afterwards, another structural reduction, r_2, is defined as

$$r_2 : b_k b_p b_q \mapsto z_1 b_p b_q, \text{where } b_p, b_q \in B' = B - \{b_k\} \text{ and } z_1 \in B' - \{b_p, b_q\}.$$

   There should be one symbol left for the assignment of z_1 once b_p and b_q have been assigned. Hence, the proof is right if $h \geq 4$. In other words, $C_{[1,0]} \leq C_{[0,0]}$. From the results of r_1 and r_2, we know that $C_{[0,0]}$ dominates $C_{[0,1]}$ and $C_{[1,0]}$ when $h \geq 5$. This completes the proof of Lemma 1. □

3. $a_i b_j b_k, b_j a_i b_k,$ and $b_j b_k a_i$
   The guesses of this type are composed of a symbol in A and two symbols in B. These guesses can also be further classified into three kinds of guesses, i.e.,

$a_ib_jb_k, b_ja_ib_k,$ and $b_jb_ka_i$. Without loss of generality, $g = a_ib_jb_k$ is choosen to undertake the following discussions. Besides, the analyses of $b_ja_ib_k$ and $b_jb_ka_i$ can be derived in a similar way and so, they are ommited here. There are six nonempty substates after the codebreaker makes the guess, g. They are $C_{[2,0]}, C_{[1,1]}, C_{[0,2]}, C_{[1,0]}, C_{[0,1]},$ and $C_{[0,0]}$, respectively. Note that their corresponding cardinality are $(h-2), 2(h-2), (h-2), 2(h-2)(h-3), 4(h-2)(h-3),$ and $(h-2)(h-3)(h-4)$. Now, we show that $C_{[0,0]}$ dominates the other five substates if $h \geq 8$.

Lemma 2. *If the codebreaker encounters C^*, and then makes the guess, $g = a_ib_jb_k, b_ja_ib_k,$ or $b_jb_ka_i$, where $a_i \in A$ and $b_j, b_k \in B$, then $C_{[0,0]}$ dominates $C_{[0,1]}$, $C_{[1,0]}$, $C_{[0,2]}$, $C_{[1,1]}$, and $C_{[2,0]}$ when $h \geq 8$.*

Proof. Five structural reductions, called $r_3, r_4, r_5, r_6,$ and r_7, are defined as follows to certify that $C_{[0,1]} \leq C_{[0,0]}$, $C_{[1,0]} \leq C_{[0,0]}$, $C_{[0,2]} \leq C_{[0,1]}$, $C_{[1,1]} \leq C_{[1,0]}$, and $C_{[2,0]} \leq C_{[1,0]}$ respectively.

$$r_3 : \begin{cases} b_jb_pb_q \mapsto z_1b_pb_q \\ b_pb_qb_j \mapsto b_pb_qz_2, \text{ where } b_p, b_q \in B' = B - \{b_j, b_k\}, \\ b_kb_pb_q \mapsto z_3b_pb_q \quad \text{and } z_1, z_2, z_3, z_4 \in B' - \{b_p, b_q\}. \\ b_pb_kb_q \mapsto b_pz_4b_q \end{cases}$$

$$r_4 : \begin{cases} b_pb_jb_q \mapsto b_pz_1b_q, \text{ where } b_p, b_q \in B' = B - \{b_j, b_k\} \\ b_pb_qb_k \mapsto b_pb_qz_2 \quad \text{and } z_1, z_2 \in B' - \{b_p, b_q\}. \end{cases}$$

$$r_5 : \begin{cases} b_jb_kb_p \mapsto b_jz_1b_p \\ b_pb_kb_j \mapsto b_pz_2b_j, \text{ where } b_p \in B' = B - \{b_j, b_k\} \\ b_kb_pb_j \mapsto b_kb_pz_3 \quad \text{and } z_1, z_2, z_3 \in B' - \{b_p\}. \end{cases}$$

$$r_6 : \begin{cases} b_kb_jb_p \mapsto z_1b_jb_p, \text{ where } b_p \in B' = B - \{b_j, b_k\} \\ b_jb_pb_k \mapsto z_2b_pb_k \quad \text{and } z_1, z_2 \in B' - \{b_p\}. \end{cases}$$

$$r_7 : b_pb_jb_k \mapsto b_pb_jz_1, \text{ where } b_p \in B' = B - \{b_j, b_k\} \text{ and } z_1 \in B' - \{b_p\}.$$

Note that $z_1b_pb_q$, $b_pb_qz_2$, $z_3b_pb_q$, and $b_pz_4b_q$ in r_3 should be distinct to reserve the one-to-one mapping property. Likewise, $b_pz_1b_q$ and $b_pb_qz_2$ in r_4 should be distinct and $b_jz_1b_p$, $b_pz_2b_j$, and $b_kb_pz_3$ in r_5 should also be distinct while $z_1b_jb_p$ and $z_2b_pb_k$ in r_6 have to be distinct as well. We can attain this with assigning these symbols of z_1, z_2, z_3, and z_4 carefully when mapping is undertaken. In order to meet requirements of the assignments of z_i in $r_3, r_4, r_5, r_6,$ and r_7, the following conditions should be maintained respectively: $h \geq 8, h \geq 6, h \geq 6, h \geq 5,$ and $h \geq 4$. Consequently, it is true that $C_{[0,0]}$ dominates $C_{[0,1]}$, $C_{[1,0]}$, $C_{[0,2]}$, $C_{[1,1]}$, and $C_{[2,0]}$ while $h \geq 8$. Hence, the proof of Lemma 2 is completed. □

4. $b_ib_jb_k$
All symbols of this kind of guesses belong to B entirely. There are totally nine nonempty substates, which are $C_{[3,0]}$, $C_{[1,2]}$, $C_{[0,3]}$, $C_{[2,0]}$, $C_{[1,1]}$, $C_{[0,2]}$,

$C_{[1,0]}$, $C_{[0,1]}$, and $C_{[0,0]}$ respectively, as the codebreaker makes the guess, $g = b_i b_j b_k$. Notice that their cardinality are $1, 3, 2, 3(h-3), 6(h-3), 9(h-3), 3(h-3)(h-4), 6(h-3)(h-4)$, and $(h-3)(h-4)(h-5)$, respectively. In the following statements, we would certify that $C_{[0,1]} \leq C_{[0,0]}$, $C_{[1,0]} \leq C_{[0,0]}$, $C_{[0,2]} \leq C_{[0,1]}$, $C_{[1,1]} \leq C_{[1,0]}$, $C_{[2,0]} \leq C_{[1,0]}$, $C_{[0,3]} \leq C_{[0,0]}$, $C_{[1,2]} \leq C_{[0,0]}$, and $C_{[3,0]} \leq C_{[0,0]}$.

Lemma 3. *As the codebreaker encounters C^*, and then makes the guess, $g = b_i b_j b_k$, where $b_i, b_j, b_k \in B$, then $C_{[0,0]}$ dominates $C_{[0,1]}$, $C_{[1,0]}$, $C_{[0,2]}$, $C_{[1,1]}$, $C_{[2,0]}$, $C_{[0,3]}$, $C_{[1,2]}$, and $C_{[3,0]}$ when $h \geq 11$.*

Proof. Since the cardinalities of $C_{[3,0]}$, $C_{[1,2]}$, and $C_{[0,3]}$ are fixed numbers, then $C_{[0,0]}$ trivially dominates $C_{[3,0]}$, $C_{[1,2]}$, and $C_{[0,3]}$ as long as there are at least three symbols in B and thus, the three symbols can be permuted appropriately to map the three substates.

Below five definitions of structural reductions, which are named as r_8, r_9, r_{10}, r_{11}, and r_{12}, are provided as follows to confirm that $C_{[0,1]} \leq C_{[0,0]}$, $C_{[1,0]} \leq C_{[0,0]}$, $C_{[0,2]} \leq C_{[0,1]}$, $C_{[1,1]} \leq C_{[1,0]}$, and $C_{[2,0]} \leq C_{[1,0]}$ respectively.

$$r_8 : \begin{cases} b_p b_i b_q \mapsto b_p z_1 b_q \\ b_p b_q b_i \mapsto b_p b_q z_2 \\ b_j b_p b_q \mapsto z_3 b_p b_q, \text{ where } b_p, b_q \in B^{'} = B - \{b_i, b_j, b_k\} \\ b_p b_q b_j \mapsto b_p b_q z_4 \quad \text{and } z_1, z_2, z_3, z_4, z_5, z_6 \in B^{'} - \{b_p, b_q\}. \\ b_k b_p b_q \mapsto z_5 b_p b_q \\ b_p b_k b_q \mapsto b_p z_6 b_q \end{cases}$$

$$r_9 : \begin{cases} b_i b_p b_q \mapsto z_1 b_p b_q \\ b_p b_j b_q \mapsto b_p z_2 b_q, \text{ where } b_p, b_q \in B^{'} = B - \{b_i, b_j, b_k\} \\ b_p b_q b_k \mapsto b_p b_q z_3 \quad \text{and } z_1, z_2, z_3 \in B^{'} - \{b_p, b_q\}. \end{cases}$$

$$r_{10} : \begin{cases} b_j b_i b_p \mapsto z_1 b_i b_p \\ b_p b_i b_j \mapsto b_p z_1 b_j \\ b_j b_p b_i \mapsto z_1 b_p b_i \\ b_j b_k b_p \mapsto b_j z_1 b_p \\ b_p b_k b_j \mapsto b_p b_k z_1, \text{ where } b_p \in B^{'} = B - \{b_i, b_j, b_k\} \\ b_k b_p b_j \mapsto b_k b_p z_1 \quad \text{and } z_1, z_2 \in B^{'} - \{b_p\}. \\ b_k b_i b_p \mapsto z_2 b_i b_p \\ b_k b_p b_i \mapsto b_k b_p z_2 \\ b_p b_k b_i \mapsto b_p z_2 b_i \end{cases}$$

$$r_{11} : \begin{cases} b_i b_p b_j \mapsto b_i b_p z_1 \\ b_i b_k b_p \mapsto b_i z_2 b_p \\ b_p b_j b_i \mapsto b_p b_j z_1, \text{ where } b_p \in B^{'} = B - \{b_i, b_j, b_k\} \\ b_k b_j b_p \mapsto z_2 b_j b_p \quad \text{and } z_1, z_2 \in B^{'} - \{b_p\}. \\ b_p b_i b_k \mapsto b_p z_1 b_k \\ b_j b_p b_k \mapsto z_2 b_p b_k \end{cases}$$

$$r_{12}:\begin{cases} b_ib_jb_p \mapsto b_iz_1b_p \\ b_ib_pb_k \mapsto z_1b_pb_k, \text{ where } b_p \in B^{'} = B - \{b_i, b_j, b_k\} \\ b_pb_jb_k \mapsto b_pb_jz_1 \quad \text{and } z_1 \in B^{'} - \{b_p\}\,. \end{cases}$$

Note that each secret code in each structural reduction, i.e., r_8, r_9, r_{10}, r_{11}, and r_{12}, should be distinct from each other to reserve the one-to-one mapping property. This can be attained by assigning these symbols of z_1, z_2, z_3, z_4, z_5, and z_6 carefully. To satisfy each assignment of z_i in r_8, r_9, r_{10}, r_{11}, and r_{12}, the following constraints have to be kept in correspondence with the order given: $h \geq 11, h \geq 8, h \geq 6, h \geq 6$, and $h \geq 5$. So, it is therefore correct that $C_{[0,0]}$ dominates $C_{[0,1]}, C_{[1,0]}, C_{[0,2]}, C_{[1,1]}, C_{[2,0]}, C_{[0,3]}, C_{[1,2]}$, and $C_{[3,0]}$ when $h \geq 11$. Hence, the proof of Lemma 3 is completed. □

After four kinds of guesses for the codebreaker are discussed, only three kinds of guesses among them are useful since the first one causes non-optimal results trivially. In order to simplify the notations, let $C^{(2)}$, $C^{(3)}$, and $C^{(4)}$ denote the hardest states caused by guesses of Type 2, Type 3, and Type 4, respectively. Hence, the difficulties of these three states have to be determined to choose the best guess for the codebreaker. The following lemma therefore describes the phenomena.

Lemma 4. *When the codebreaker encounters C^*, the hardest states caused by guesses of Type 2, Type 3, and Type 4, i.e., $C^{(2)}$, $C^{(3)}$, and $C^{(4)}$, are produced. Thus, we have $C^{(4)} \leq C^{(3)} \leq C^{(2)}$.*

Proof. From the meanings of $C^{(2)}$, $C^{(3)}$, and $C^{(4)}$, it reveals that $C^{(2)}$ is composed of secret codes that are permutations of $(h-1)$ symbols, and $C^{(3)}$ consists of what are permutations of $(h-2)$ symbols while the codes in $C^{(4)}$ are permutations of $(h-3)$ symbols. Let $S^{(2)}$, $S^{(3)}$, and $S^{(4)}$ denote the sets of symbols appearing in $C^{(2)}$, $C^{(3)}$, and $C^{(4)}$, respectively. Then, let the symbols in $S^{(2)}$, $S^{(3)}$, and $S^{(4)}$ be sorted separately according to the lexicographical order. A mapping is generated naturally if we map each symbol in $S^{(4)}$ to that in $S^{(3)}$ one by one in sorted order. So does the mapping between $S^{(3)}$ and $S^{(2)}$. Obviously, we have $C^{(4)} \leq C^{(3)} \leq C^{(2)}$. This completes the proof. □

After Lemma 1, Lemma 2, Lemma 3, and Lemma 4, we may conclude with the following lemma.

Lemma 5. *For a special state, C^*, which also represents a $3 \times h$ AB game $(11 \leq h \leq n)$, the optimal guess for the codebreaker now is $b_ib_jb_k$, where b_i, b_j, $b_k \in B$.*

Proof. From Lemma 4, $C^{(4)}$ is the easiest state to identify a secret code compared to $C^{(2)}$ and $C^{(3)}$. The goal of the codebreaker is to minimize the number of guesses required and so, the codebreaker has to choose the guess which results in $C^{(4)}$ in the worst situation. The optimal guess for the codebreaker is therefore $b_ib_jb_k$. □

5 The Devil's Strategy for the Codemaker

Since the mission of the codebreaker aims to minimize the number of guesses to acquire a secret code, the codemaker tries to maximize the number of guesses for the codebreaker if he decides to implement a devil's strategy. Hence, the worst case for the codebreaker means that his opponent conducts a devil's strategy (or called a worst response for the codebreaker) in each turn during the gaming process in order to maximize the number of guesses. In the follow-up, a lemma is exhibited to demonstrate what is the worst response for the codebreaker if he encounters a $3 \times h$ AB game, where $h \leq n$.

Lemma 6. *For a* $3 \times h$ *AB game, where* $11 \leq h \leq n$*, the codebreaker will require a maximum number of guesses to obtain the code while the codemaker answers* [0, 0] *after the codebreaker's guess.*

Proof. From Lemma 5, it is obvious that the codebreaker must choose $b_i b_j b_k$ as a guess for a $3 \times h$ AB game. After the codebreaker takes the optimal guess, nine substates will be formed. These substates are $C_{[0,0]}$, $C_{[0,1]}$, $C_{[1,0]}$, $C_{[0,2]}$, $C_{[1,1]}$, $C_{[2,0]}$, $C_{[0,3]}$, $C_{[1,2]}$, and $C_{[3,0]}$, respectively. $C_{[0,0]}$ dominates $C_{[0,1]}$, $C_{[1,0]}$, $C_{[0,2]}$, $C_{[1,1]}$, $C_{[2,0]}$, $C_{[0,3]}$, $C_{[1,2]}$, and $C_{[3,0]}$ in accordance with the result of Lemma 3. In other words, $C_{[0,0]}$ is the hardest substate among the nine ones. Conclusively, the codemaker must response [0, 0] as his worst response and this will result in the worst case for the codebreaker because of the maximum number of guesses. The proof is thus finished. □

6 Conclusions

From the above discussions, the optimal guess for the codebreaker and the adversary response for the codemaker, which refers to the worst case for the codebreaker as well, are eventually obtained with the consideration of the special state C^*. In the follow-up, all results mentioned above will be concluded to derive a theorem.

Theorem 1. *For a* $3 \times n$ *AB game, the minimum number of guesses for the codebreaker in the worst case is*

$$\begin{cases} \lfloor n/3 \rfloor + 3, & \text{if } 3 \leq n \leq 7 \\ \lfloor (n+1)/3 \rfloor + 3, & \text{if } n \geq 8. \end{cases}$$

Proof. At the beginning of a $3 \times n$ AB game, the n symbols are not used and then all secret codes are all equivalent. As a result, a secret code is chosen randomly as the first guess for the codebreaker. Nine substates are therefore produced and [0, 0] is taken as an adversary response according to Lemma 6. Afterwards, $C_{[0,0]}$, which results from the first response, matches the attribution of the special state C^* described in Lemma 5. Thus, Lemma 5 can be applied to this state. We find that the situations mentioned in Lemma 5 and Lemma 6 will appear alternately in the following gaming process. So we have the following recurrence.

$$T(n) = T(n-3) + 1, \text{ when } n > 11. \tag{1}$$

The minimum number of guesses cannot be obtained easily with the use of analyses when $n \leq 11$ because of the irregular behavior. So, a refined exhaustive search, which originates from Huang *et al.* [9], is adopted to acquire the results. After the use of computer programs written with this approach, the minimum numbers of guesses required for the codebreaker in the worst case are obtained in several hours and they are 4, 4, 4, 5, 5, 6, 6, 6, and 7, respectively when $n = 3, 4, 5, 6, 7, 8, 9, 10,$ and 11. For example, an optimal strategy for 3×7 AB game is considered with $S = \{0, 1, 2, 3, 4, 5, 6\}$. If the codemaker takes 165 as a secret code, the gaming process will be as follows: 012, $[0, 1]$, 023, $[0, 0]$, 041, $[0, 1]$, 156, $[1, 2]$, 165, $[3, 0]$. In other words, the codebreaker requires 5 guesses to identify 165 while playing the worst-case optimal strategy.

We derive the above recurrence (1) and conclude with the results of smaller values of n. Hence, the closed form of the formula is exhibited as follows.

$$\begin{cases} \lfloor n/3 \rfloor + 3, & \text{if } 3 \leq n \leq 7 \\ \lfloor (n+1)/3 \rfloor + 3, & \text{if } n \geq 8. \end{cases}$$

This completes the proof. □

Partial results of $3 \times n$ AB games, $3 \leq n \leq 16$, are summarized in Table 1. As $3 \times n$ AB games have been solved successfully, a natural generalization is to explore the techniques for $m \times n$ AB games, where $m \geq 4$. This problem remains open. We hope that the methods proposed here could help other related research in the future.

Table 1. The minimum number of guesses for $3 \times n$ AB games in the worst case

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16
# of guesses	4	4	4	5	5	6	6	6	7	7	7	8	8	8

Acknowledgements

We thank anonymous referees for their valuable comments. This research was supported in part by a grant NSC 97-2221-E-003-008 from National Science Council, ROC.

References

1. Merelo-Guervos, J.J., Castillo, P., Rivas, V.M.: Finding a needle in a haystack using hints and evolutionary computation: the case of evolutionary MasterMind. Applied Soft Computing 6(2), 170–179 (2006)
2. Knuth, D.E.: The computer as Mastermind. Journal of Recreational Mathematics 9, 1–6 (1976)
3. Irving, R.W.: Towards an optimum Mastermind strategy. Journal of Recreational Mathematics 11(2), 81–87 (1978)

4. Neuwirth, E.: Some strategies for Mastermind. Mathematical Methods of Operations Research 26, 257–278 (1982)
5. Norvig, P.: Playing Mastermind optimally. ACM SIGART Bulletin 90, 33–34 (1984)
6. Koyama, K., Lai, T.W.: An optimal Mastermind strategy. Journal of Recreational Mathematics 25, 251–256 (1993)
7. Rosu, R.: Mastermind. Master's thesis, North Carolina State University, Raleigh, North Carolina (1999)
8. Barteld, K.: Yet another Mastermind strategy. ICGA Journal 28(1), 13–20 (2005)
9. Huang, L.T., Chen, S.T., Huang, S.J., Lin, S.S.: An efficient approach to solve Mastermind optimally. ICGA Journal 30(3), 143–149 (2007)
10. Shapiro, E.: Playing Mastermind logically. ACM SIGART Bulletin 85, 28–29 (1983)
11. Chen, S.T., Lin, S.S., Huang, L.T., Hsu, S.H.: Strategy optimization for deductive games. European Journal of Operational Research 183(2), 757–766 (2007)
12. Kalisker, T., Camens, D.: Solving Mastermind using genetic algorithms. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 1590–1591. Springer, Heidelberg (2003)
13. Singley, A.: Heuristic solution methods for the 1-dimensional and 2-dimensional Mastermind problem. Master's thesis, University of Florida (2005)
14. Chen, S.T., Lin, S.S., Huang, L.T.: A two-phase optimization algorithm for Mastermind. The Computer Journal 50(4), 435–443 (2007)
15. Berghman, L., Goossens, D., Leus, R.: Efficient solutions for Mastermind using genetic algorithms. Computers and Operations Research 36(6), 1880–1885 (2009)
16. Chen, S.T., Lin, S.S.: Optimal algorithms for $2 \times n$ AB games - a graph-partition approach. Journal of Information Science and Engineering 20(1), 105–126 (2004)
17. Chen, S.T., Lin, S.S.: Optimal algorithms for $2 \times n$ Mastermind games - a graph-partition approach. The Computer Journal 47(5), 602–611 (2004)
18. Goddard, W.: Mastermind revisited. Journal of Combinatorial Mathematics and Combinatorial Computing 51, 215–220 (2004)