Serdica J. Computing **7** (2013), No 3, 215–226

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

TEST, TEACHERS, QUORUM (PURE POPULATIONS)

Slavian Radev

Abstract. The "trial and error" method is fundamental for Master Mind decision algorithms. On the basis of Master Mind games and strategies we consider some data mining methods for tests using students as teachers. Voting, twins, opposite, simulate and observer methods are investigated. For a pure data base these combinatorial algorithms are faster then many AI and Master Mind methods. The complexities of these algorithms are compared with basic combinatorial methods in AI.

1. Introduction. The praxeological method of "trial and error" is well known to animals. Some of its modifications are known by Homo. In the game "Master Mind" (we suppose that the reader knows it well and use MM for the game and algorithms [2]) there are some optimal strategies based on that method. In this paper we consider tests [3] for knowledge verification [4] instead of the game MM. The difference is that we use a data base of evaluated students instead of generated answers. These tests are well known in the scholar practice by both

ACM Computing Classification System (1998): F.3.2, G.2.1, H.2.1, H.2.8, I.2.6.
Key words: voting, game, observer, data mining, test, master mind.

students and teachers. To be more clear we choose a typical example of such a test, maybe the most popular one: 6 questions, 3 answers, only one of which is correct. In that test we use some populations of equally evaluated students as teachers. The tests are within the scope of the corresponding games. It is easy to generalize these games and to calculate the complexity of the strategies more precisely, but this is not the aim of the paper. Here we concentrate on the main ideas of some combinatorial algorithms and their elements. For this we consider only pure populations and simple forms of the strategies. In a following paper we shall consider game strategies for arbitrary populations.

1.1 The problem. In recent years it is trendy to conduct examinations with tests and for teachers to train the students for these tests. Perhaps the most popular test consists of 6 questions with 3 answers proposed for every question, one of which is correct. This test may be considered in 3-valued logic as well as in rough set theory [9].

In the terminology of MM this is a $\langle 6, 3\rangle$ game [10]. The value is equal to the sum of correct answers. Hence for the student the game is to find which of these answers are true. A stupid strategy is to try all 3^6 combinations. The probability of guessing the true answers seems small and teachers believe that their teaching strategy based on lectures and tests is optimal. Let us consider some games and some student strategies (typical students find an almost optimal strategy within 0.3 sec.). The strategies will be written in a classical inductive schema, but for students of informatics I use a C-program form. This is a game similar to Master Mind (MM), hence we start with short information about known strategies and games. The *space complexity* (sc) of the game is 3^6 and the *tree depth* is 7. We assume that the space is in a table with 7 columns (the last one is for the evaluation). Simultaneously we compare our methods with Artificial Intelligence methodology.

1.2. Tables and logical properties. First we need some general methodological notions. In [8] we show how to use finite mathematics in table form. Every table T is an ordered set (list) of elements t_{ij}. But it is also a list of columns and a list of rows. The columns and rows are vectors In this table form we have one empty place – t_{00}. To fill the table sometimes we use that place for the name of the table and to have a unique name for every table we use [5, 7] the HTML form of the same table with an empty place t_{00}. A typical table is the Pascal triangle with a 0-named first row and a first column filled with 1s. In every remaining cell the number is the sum of the left-hand and the upper cell. These cells contain the binomial coefficients and we may use the notation $e(i, j)$ to sign that the coefficient is calculated in parallel, which is a faster operation then factorial function ("!") used in Newton's formulas.

If we consider T as an *information system* then it has an initial row with names for attributes and an initial column with names of elements. If one column (not the initial one) is called "decision" then it is a *decision system.*

The logical characterization of tables depends on the initial postulates. If we consider the table as an information system with "objects" o_i, "attributes" a_j and "valuations" v_{ij} then the characteristic formula is

$$H_I \stackrel{\text{def}}{=} \wedge_{i\leq n}(\wedge_{j\leq k}(o_i(a_j) = v_{ij})).$$

It is clear that the formula H_I is consistent.

If we consider the table as a decision system with a distinguished attribute a_k then the characteristic formula is

$$H_D \stackrel{\text{def}}{=} \wedge_{i\leq n}(\wedge_{j<k}(o_i(a_j) = v_{ij}) \to (o_i(a_k) = v_{ik})).$$

This formula is also consistent.

Also the following formula is consistent:

$$H_{DD} \stackrel{\text{def}}{=} \vee_{i\leq n}(\wedge_{j<k}(o_i(a_j) = v_{ij}) \to (o_i(a_k) = v_{ik})).$$

In our games the evaluation is the number of true answers, hence the rows with equal elements but different decisions are forbidden. Otherwise the formula H_D is inconsistent, while the formula H_{DD} is consistent. Sometimes these formulas are not categorical. The formula H_{DD} is consistent but not categorical. In the tests categoricity is a very important property because the teacher verifies exactly one knowledge. In the decision tables of the test we shall use the following postulate:

$H_T \stackrel{\text{def}}{=}$ "in every row the number of true answers is equal to the decision value".

This postulate is not always categorical. The categoricity is guaranteed with the postulate:

$H_{TC} \stackrel{\text{def}}{=}$ "there is exactly one row with maximal evaluation. That row appears in every extension of the table. If any other row has evaluation n then exactly n elements are equal to the corresponding elements of that row."

Note that this property is in text form not in the form of logical formula because the complexity of that formula would match the volume of this article.

A *model* is any maximal table with all possible combinations of values (except for the decision). Hence categoricity guarantees sufficient information in the table to have exactly one model.

The complexity of recognizing the consistency and categoricity of table depends on the number of variables. Hence the formula H_{TC} is EXP-EXP-TIME. But the following method returns it to P-time:

Candidate method: Consider an arbitrary combination c (except the decision) and call it a "candidate" for model. Evaluate it by every element (evaluated). If the evaluations are always equal, then c is a model. If there is no such c, then the table is inconsistent. If there are two different c and c' models then the formula is not categorical.

The complexity is less then 3^{12} hence it is of polynomial computational complexity (the population is not more than 3^6).

We suppose that every table is normalized. A table is normal if there are no two identical rows. For decision tables we assume that there are no rows that differ only on the decision position (postulate H_{TC}). This is equivalent to the game postulate that the evaluation is the number of true answers. The table is not the best form for information. In our $\langle 6, 3\rangle$ game as well as in the MM games it is better to consider a 6-dimensional cube. In some cases it is better to consider the topology of space as a 6-dimensional torus. Whenever the tables are ordered structures and the MM methods use search algorithms the topology is important in the complexity calculations. For instance the ends of the main diagonals of the cube are neighbourhoods in the torus topology. The difference is in the length between nodes: in torus topology every two points are neighbourhoods.

Note that 3 or 4 possible answers correspond to the combinations of two propositions: A and B. The 4 possible answers are the lattice $A \wedge B$, $A \wedge \neg B$, $\neg A \wedge B$, $\neg A \wedge \neg B$. That lattice suggests that the top is true. The last 3 answers are less intuitive. If the student adds a wrong virtual element C which is in the scope of A and not in the scope B then he/she/it will choose the second possibility. That element changes the structure of the logical lattice. This is the main element in the verified corresponding knowledge.

If we consider a $\langle 6, 3\rangle$ multi-test then the corresponding MM game is $\langle 6, 2^3\rangle$ and the game space is 2^{18}. This is greater then the space of our game.

Note that any logical method is based on the number of variables and these variables in the formulas H are more then 3^7, hence the computational complexity is EXP-EXP-TIME. I think that the logical analysis of the weather equations is less complicated.

2. MM games and strategies. Let us consider two of the methods for the classical MM game. The first game is the following:

Game 1. The student may repeat the same test many times. After every repetition the student receives his evaluation.

Strategy 1.2.1. "Try and learn"

Base: choose an arbitrary vector t of 6 answers and receive evaluation z.

Step: change in t the answer of the ith question. For the new evaluation z' there are 3 possibilities: greater then z, hence the new answer is correct; less then z, then the old answer is correct; equal to z, then the third answer is correct.

Complexity 1.2.1. Within $6+1 < \log(3^6)+1$ steps the algorithm reaches the excellent answer of the test. Hence the game complexity is LOG-time and the computation complexity is LOG-time. Note that in the mentioned multi-test the same strategy is also LOG-time.

Strategy 1.2.2. MM method (Knuth)

Base: as before

Step: using the information from the previous evaluations to calculate the probability of the evaluations $z'(t') = x$ $(x = 0..6)$. Choose t' such that $\max(p(z'(t') = x \mid x = 0..6)$ is minimal.

This algorithm is well known [2] and proved almost optimal for the game MM. The game complexity gc of that algorithm is 5. Here by "game complexity" we understand the number of moves sufficient to obtain evaluation 6. Hence it is LOG-time. The computational complexity cc depends on the calculations of probabilities. Hence it is at least linear-time, in practice it is EXP-time. It is a little faster (5 steps in the game complexity) and slower (Exp-time computation complexity) than the previous one.

Strategy 1.2.3. Observers (check-points method in MM)

Instead of a calculation of probabilities the student may use vectors $t_1..t_m$ that "observe" the 6-dimension space of the game. Suppose that these "maximally independent" vectors are calculated before and given in a table [1]. Hence the gc complexity is again LOG-time. Note that cc complexity depends on the logical analysis of the table which is about 3^6 long. Using the lexicographical order the search is computable in one step in a 3^7 table (there is great redundancy – the rows: "if $v(t_1) = 6$ and $v(t_2) = 2\dots$ " are impossible in the game and search is finished at the first step when $v(t_1) = 6$). After elimination of impossible combinations the space is still large enough and the computational complexity grows. On the other hand the table is calculated once before all games, hence the complexity is restricted to search in the table. This search is about half of the table (Exp-time complexity) in an unordered table and 1 step (linear complexity) in an ordered table.

These algorithms are hard for the greatest number of students. Note that typical student finds almost optimal strategy within 0,3 sec. In [3, 6] we consider some democratic methods based on a noise calculator.

3. Test methods. The test game we consider is a little different. In some sense it correspond to the knowledge data discovery game:

Game 2. There is only one possibility for the student, but there are many other students that have been examined and are evaluated. This situation is also natural. Some examinations are in different countries at the same local (not absolute) time.

In this game and in this paper we consider only pure cases: all students in the population are equally evaluated.

For simplicity we suppose that the data base of the population is in reverse alphabetical order with respect to the decision column $(6, 5..0)$.

Case 2.1. There is a 6-point (excellent) student.

Strategy 2.1. Copy her answers.

This algorithm is at the top of computation and game complexity: the cost is the constant 1. The only problem is to find such a student in the data base (if there is one in it).

3.1 Voting methods. For the other cases the voting strategy is the same: ask the populations of x-point students for the answers of the corresponding questions: if $x = 5, 4, 3$ the true answer sounds stronger, if $x = 1$ it sounds weaker, if $x = 0$ it is not mentioned [6]. Hence voting is positive in 6, 5, 4, 3 point populations and negative in 1, 0 point populations.

Moreover there is redundancy in the mentioned populations and let $q(x)$ denotes the *quorum* of the population x – the minimal number that allows making a voting decision. For 4, 3, 1, 0 the quorum is $1/2 + 1$ and for 5 it is $1/4 + 1$.

Example 5-valued population:

There are 12 5-valued students.

The distribution of answers is 10-1-1 for every question. Hence it is sufficient to ask only 4 students. This is the quorum. The optimal practical algorithm is to sit in a place from which 4 students are visible. The computational complexity is 3 steps of the operation “count if”. The other cases are in the following table. The number of the populations is (i, j) binomial coefficient times 2^{k-j} hence is typical combinatorial formula.

All these calculations are well-known from the ancient democratic world. Note that here are only direct voting cases. For a multi-level choice of decision the quorum is greater.

Unfortunately the greatest population of 2-point students has equal answers for every possibility (the ideal coin produces such a population but no coin has a diploma). Hence we cannot make a decision and the voting method is not universal in that case. In mixed populations the probability that the situation is

Table 1. voting method

val	sc	qr	vm	vc
6	1	1	+	1
5	12	4	+	1/4sc+1
4	60	31	+	1/2sc+1
3	160	81	+	1/2sc+1
2	240	no	no	no
1	192	97	–	1/2sc+1
0	64	33	–	1/2sc+1

analogous cannot be ignored. Between these populations is the maximal population, hence the voting method is not good because in the maximal population is the full information, hence strategy 2.1 may be used.

Let us consider some other combinatorial methods.

3.2. Opposition methods. In the family of MM methods is the algorithm with randomly chosen students. This algorithm has proved a little bit faster then the method of Knuth [1, 2]. More of these students are in opposite positions (diagonals) of the table (if the population is in a cube form).

Case 3.2.7. There are two 0-point students with different answers $\forall i(t_1(i) \neq t_2(i))$. Call them "oppositionists".

Wilde lemma: Whenever people agree with me, I always think I must be wrong. (Oscar Wilde)

Strategy 2.7. Using the previous lemma in its symmetrical form, choose for every question the third (not mentioned by the oppositionists) answer.

This algorithm is good but is based on the search of pairs of oppositions between 0-star students. The population is 2^6 and the quorum $2^5 + 1$ is sufficient. The game complexity gc is 1 and equivalent to the game complexity of excellent students. The computation complexity cc is linear and depends on search algorithms.

Note that the table form is not adequate to the method. If we consider the space as a 6-dimension torus, then the opposition are in the main diagonals. This space is similar to the 6-dimensional cube but has the property: every two points are neighbourhoods.

Cases 2.1, 2.2, 2.3 There are no 6, 5, 4-point opposition.

Hence the method is not universal.

3.3 Method twins. For the cases 5, 4, 3, 2, 1, 0 we look for "twins". *Twins* are every two vectors that are different in exactly one position. *Cousins* are two vectors that are different in exactly two positions. Case 6 is one-element but it is simultaneously the answer.

The general strategy (even in mixed populations) is: If their evaluations are equal, then the third answer is true, else the better evaluated one has the true answer in that position. Put the true answer instead the false answers in all students. Then we have a mixed population even if we start from a pure population. Choose only those that are better evaluated. Hence the game in the next step is $\langle 5, 3\rangle$ Repeat the search for twins. Continue to game $\langle 2, 3\rangle$. And so on.

The quorum for twins is the sum of the quorums of corresponding games. Hence the quorum is less then the initial population. The search algorithms in full populations 5, 4, 3, 2, 1, 0 are 1-step, while for the quorum is greater then 2 steps, hence the quorum complicates the complexity. Nevertheless the cc is linear.

Note that in the form of cube (not torus) the twins are in the nearest planes.

Strategy 3.3.1. Instead of the class of all different 3-star students look for a set of pairs of complementary students $\forall i(t_a(i) \neq t_b(i))$. The search algorithms determine the number of pairs.

The complexity is also about P-time. Unfortunately the search algorithms works slowly when the population is near quorum. That result is not elegant and is not included here. But the method is good when it is hard to find greater populations.

Case 2-evaluated students

This case is the most interesting one because the voting strategy for that population fails: the answers 1, 2, 3 are given by the same number of students.

Strategy 3.3.1. Use 2 oppositioners. The restricted table is for $\langle 5, 3\rangle$ game. Repeat and reduce the problem to $\langle 2, 3\rangle$ game.

In the other cases the strategies are based on the nearest (or symmetrical) strategies. For 5-star students one may use an inverse modification of 2.0., for 1-star students modification of 4-star students strategies.

Note that these cases depend on the size of the data base, and on the search algorithms. In the examples here we need no search methods, because the data base is a torus.

3.4 Simulating teacher. In this method we try to simulate the teacher behaviour. This method is also a modification of “trial and error”.

Teacher postulate: If teacher t evaluates student s with n then the student s evaluates the teacher also with n.

Strategy: consider a modification of “candidate strategy”

Table 2. Twins

val	sc	nnq	nnc	gaq	gac
6	1	1	+	1	1
5	12	8	+	3	LIN
4	60	40	+	4	LIN
3	160	110	+	6	P
2	240	160	no	10	P
1	192	130	–	15	P
0	64	45	–	20	P

Base: choose vector t. Choose an arbitrary student s from the population. Let $t(s)$ be the t-evaluation of s.

Step: If $t(s) = v(s)$ then t is candidate: choose s'; return. If $t(s) \neq v(s)$ then choose t' that differs from t in one position. If $t'(s) = v(s)$ then it is the new candidate, else if $t'(s)$ is near $v(s)$ then $t(s)$ and continue with t' else with t; choose s'; return.

Complexity: the computational complexity of this algorithm is linear, the game complexity is 1.

4. Mixed strategies. More important is the universality of a method then the complexity. Hence consider some new strategies in which more than one method is used.

Observation 1. Twins is an universal method and may be used in a 2-point population instead of voting. After twins is performed we may extract two subpopulations from the population – 3-point and 1-point. Hence we may continue with voting methods.

Also voting and twins work in the quorum calculation. For instance in 5-point population in every combination of 3 students there are either twins or all positions have 2-1 evaluations. Hence the quorum is 3. Simultaneously the computation complexity grows to 5.

Note that this quorum is minimal because if 111112 and 111121 are evaluations of two 5-star students then 111111 and 111122 are possible candidates for 6.

Observation 2. Twins may be combined with genetics. As before every two 5-point students are either twins (of the form 11111a and 11111b) or cousins (of the form 11111a and 1111b1) Hence a genetic combination is either 111111 or 1111ba. Every other 5-point student is good to decide which of these to choose.

Complexity: quorum is 3, game complexity is 3, computational complexity is 5.

Hypothesis: twins with voting method is complexity equivalent to twins with genetics.

5. Experiments. The genetic algorithms in 2-star case are several times slower. If an algorithm is based on the full population then it is polynomial time cc, if it is based on quorum for twins, then it is again polynomial cc. If it is based on a small initial population then it waits for "mutation". Their complexity is at least linear (for instance in 5-star populations) with respect to the space of events. In the case of a 0-star population the mutation operation is the only instrument that allows finding the true answer. Hence at least 6 mutations (approximately 20^6 moves) have to be performed to reach the result, while the opposition method needs 1 step.

The artificial neural network (ANN) after many repetitions continues to be around the ideal coin, a simple ANN after 10^6 steps is in the initial position.

That's why we concentrate on comparison with the MM methods with some combinatorial methods in the cases of pure populations. In real data bases it is possible to obtain a result faster even from the populations smaller then absolute quorum, but the quality of such a result is not good.

The next table is a part of the full table of 50 professional ANN applied to the same pure 4-star population:

Table 3. ANN results

RBF 6-6-2	66.80384	RBFT
RBF 6-15-2	77.50343	RBFT
RBF 6-6-2	70.37037	RBFT
MLP 6-4-2	77.36626	BFGS 8
MLP 6-9-2	76.95473	BFGS 5
RBF 6-6-2	76.81756	RBFT
MLP 6-4-2	81.89300	BFGS 44
MLP 6-4-2	68.99863	BFGS 5
RBF 6-6-2	75.03429	RBFT
MLP 6-4-2	85.45953	BFGS 61
MLP 6-4-2	77.09191	BFGS 2
MLP 6-8-2	77.22908	BFGS 5
MLP 6-4-2	85.04801	BFGS 33

Observation: the mistake (near 1/3) is great for such a small data base.

Hypothesis: The ANN ignores one of the possible answers or half of the test population. If the learn population is about 2/3 then it is within the scope of the quorum, hence the mistake is in the algorithms.

I think that such methods are good only when the ignored information is mentioned. In most papers about ANN this information is not mentioned.

The experiments with genetic algorithms sometimes give the answer in time approximately near the corresponding voting method but sometimes (in 4-point population) their computational complexity was more than an hour (about 10^7 evolutions or 10^{11} operations) and the program was aborted as unusual. We do not consider modifications of genetic algorithms in which the next generation uses twin method or any other method based on forbidden combinations (for instance observers).

6. Summary. All these methods generate different algorithms that may be included in the genetic algorithm paradigm. In this paper we concern LOG and NLOG algorithms. The first class corresponds to reflexes, the second to combination of reflexes with careful observations. Almost the same methods work in the mixed populations. Unfortunately the computational complexity and game complexity are not so elegant, the quorum is not trivial, the search algorithms in data bases are complicated. Hence it is better to continue investigations of Master Mind methods with observers in real data bases. In some of the strategies (2.4.1, 2.5.0, ...) we can use different forms of granulations. Combinations of specialized granulations with complex populations of observators seems fruitful but these ideas are not in the scope of this paper. The evolution programming may be more effective when additional information is included.

In teacher practice I use a multi-test with 10 questions. In that test the 0-information population is between 3 and 4-points (there is no pure population that corresponds to 2-evaluated population in the game $\langle 6, 3\rangle$). Negative points (a chosen false answer gives -1) were used once but after the negative result of more than $1/3$ of the students I use only positive evaluations. There was a student who got 0 points in each of 10 tests of 10 questions, while an arbitrary coin got 33 points. His enormous negative knowledge is sufficient to be vice director (under suggestion of N. Parkinson). From that emerged the hypothesis that two 0-point students are equivalent to an excellent one. Hence the idea is not mine.

REFERENCES

[1] Chvatal V. Mastermind. *Combinatorica,* **3** (1983), 325–329.

[2] Knut D. E. The computer as mastermind. *J. Recreational Math.*, **9** (1976), 1–6.

[3] Kropiwnicki L., S. Radev. Multi-agent Systems for Decisions. Telecotron, Warszawa, 2006 (in Polish).

[4] Radev S. Mixed theories. *Serdica Journal of Computing*, **1** (2007), No 1, 101–114.

[5] Radev S. Discrete Mathematics and Logic. Siedlce, Poland, 2011(in Polish).

[6] Radev S. Test is Voting. In: Proceedings of the Second International Scientific Conference "Education, Science, Innovations 2012", Pernik , 9-10 June 2012. (in print)

[7] Radev S. Classes, Speculations, Ontologies. Telecotron, Warszawa, 2013 (in Polish).

[8] Radev S. Table mathematics. Telecotron, Sofia, 2012 (in Bulgarian).

[9] Skowron A., C. Rauszer, Z. Pawlak. The discernibility matrices and functions in information systems. *System Theory, Knowledge Engineering and Problem Solving*, **11** (1992), 331–362.

[10] Swaszek P. F. The Mastermind Novice. *J. Recr. Math.,* **30** (2000), 193–198.

Slavian Radev
Department of Mathematical Linguistics
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
Acad. G. Bonchev Str. Bl. 8
1113 Sofia, Bulgaria
e-mail: `SlavianRadev@gmail.com`

Received June 3, 2013
Final Accepted February 5, 2014