Discrete Applied Mathematics 202 (2016) 8–18

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Rényi–Berlekamp–Ulam searching game with bi-interval queries and two lies

Shu Min Xing [a,*], Wen An Liu [b], Kun Meng [b]

[a] *Department of Mathematics and Physics, Anyang Institute of Technology, Anyang 455000, People's Republic of China*

[b] *College of Mathematics and Information Science, Henan Normal University, Xinxiang 453007, People's Republic of China*

A R T I C L E I N F O

Article history:
Received 20 February 2014
Received in revised form 26 November 2015
Accepted 8 December 2015
Available online 9 January 2016

Keywords:
Rényi–Ulam game
Search
Lie
Bi-interval queries
Worst-case

A B S T R A C T

We consider the following searching game: there are two players, say Questioner and Responder. Responder chooses a number $x \in S_n = \{1, 2, \ldots, n\}$, Questioner has to find out the number x by asking *bi-interval queries* and Responder is allowed to lie at most two times throughout the game. The minimal number $q^*(n)$ of bi-interval queries sufficient to find the unknown integer x is determined for all integers n. This solves completely Rényi–Berlekamp–Ulam searching game with bi-interval queries and two lies, partially solved by Mundici and Trombetta. Their solution applied only to the case when n is a power of 2.

1. Introduction

Rényi–Berlekamp–Ulam searching game, abbreviated by RBU, was investigated by Rényi [22], Berlekamp [2] and Ulam [24], respectively. There are two players, called *Questioner* and *Responder*. Given a search space $S_n = \{1, 2, \ldots, n\}$, Responder thinks of a "target" integer $x \in S_n$, and Questioner is required to find out x by asking a series of queries "$x \in Q$?", where Q is a subset of S_n. Each query can be answered by "yes" or "no", and Responder is permitted to lie up to e times. The aim of RBU is to determine the smallest possible number needed to find the unknown number x. Pelc [19], Guzicki [12] and Deppe [9] have solved completely the cases $e = 1$, $e = 2$ and $e = 3$, respectively, based on previous work by Czyzowicz and Mundici [7,8], Negro and Sereno [17,18], Pelc [20], Hill [13], Cicalese [5] and Deppe [10].

RBU has many variants. In the terminology of search theory, RBU mentioned above belongs to the framework of binary adaptive search. Pelc [21] and Liu [14] solved completely 3-ary adaptive search with one lie and two lies, respectively. Aigner [1] and Malinowski [15] investigated the q-ary adaptive search with one lie. Cicalese and Vaccaro [6] studied the q-ary adaptive search with two lies. For more details, the reader can refer to Pelc [20], Hill [13] and Cicalese [3].

It is of interest to investigate the optimal strategies involving the simplest possible queries. To this purpose, the concepts of *interval queries* and *bi-interval queries* are introduced. Cicalese [4] proved that the optimal strategies exist by using *k-interval queries* for the case $e \geq 1$ and $n = 2^m$, where k depends only on e. Mundici and Trombetta [16] gave the minimal number $q^*(n)$ of bi-interval queries for the special case $n = 2^m$ and $e = 2$. They also showed that this result does not hold for interval queries. In this paper, we will generalize Mundici and Trombetta's result from $n = 2^m$ to arbitrary integer n.

* Corresponding author.
E-mail address: xxxxsssmmm@126.com (S.M. Xing).

http://dx.doi.org/10.1016/j.dam.2015.12.007
0166-218X/

2. RBU with arbitrary queries and two lies

This section is devoted to RBU with arbitrary queries and two lies, i.e., each query is of the form "$x \in Q$?", where Q is a subset of S_n. Let $q(n)$ be the smallest possible number of arbitrary queries sufficient to find the unknown number x, for $e = 2$ and a fixed positive integer n. In [12], Guzicki has given a complete solution to the exact values of $q(n)$, but the representation of $q(n)$ given by Guzicki is very complicated. We try to give a simple formula on the exact values of $q(n)$ for all integers $n \geq 2$. This technique of obtaining the simple formula on the exact values of $q(n)$ is necessary to determine the exact values of $q^*(n)$.

We will follow the notations and terminologies introduced by Guzicki [12]. After some number of queries, an intermediate stage of the game can be represented by a *state* $\sigma = (A_0, A_1, A_2)$, where A_i denotes the set of the elements of S_n associated with i lies, $0 \leq i \leq 2$, and A_0, A_1, A_2 are pairwise disjoint. If Questioner chooses a query $Q \subseteq S_n$ then Responder answers either "yes" or "no". By answering "yes", Responder assigns an additional lie to each element in $S_n - Q$, so that a resulting state $\sigma_y = \sigma_y(Q)$ is obtained from σ by moving the elements corresponding to $S_n - Q$ to the right one position. Similarly, by answering "no", Responder gets a resulting state $\sigma_n = \sigma_n(Q)$ by moving the elements corresponding to Q to the right one position (see Guzicki [12]), i.e.,

$$\begin{aligned}\sigma_y &= \sigma_y(Q) = (A_0 \cap Q, (A_0 - Q) \cup (A_1 \cap Q), (A_1 - Q) \cup (A_2 \cap Q)),\\ \sigma_n &= \sigma_n(Q) = (A_0 - Q, (A_0 \cap Q) \cup (A_1 - Q), (A_1 \cap Q) \cup (A_2 - Q)).\end{aligned} \tag{1}$$

By $\sharp A$ we denote the *cardinality* of set A. Let $a = \sharp A, b = \sharp B, c = \sharp C$, the triple (a, b, c) is called *type* of $\sigma = (A, B, C)$. We will write $\sharp\sigma = (a, b, c)$, if necessary. A state $\sigma = (A_0, A_1, A_2)$ is characterized by its type $\sharp\sigma = (\sharp A_0, \sharp A_1, \sharp A_2)$, i.e., two states $\sigma = (A_0, A_1, A_2)$ and $\sigma' = (A'_0, A'_1, A'_2)$ have the same minimal number of queries sufficient to find the unknown integer x if $\sharp A_0 = \sharp A'_0$, $\sharp A_1 = \sharp A'_1$ and $\sharp A_2 = \sharp A'_2$. This technique has been used by many papers. See Cicalese [3].

Let $\sigma = (A, B, C)$ be a state with type $\pi = (a, b, c)$. By $\overrightarrow{q} = [x, y, z]$ we denote a query of $\pi = (a, b, c)$, i.e., a query Q consists of x, y and z elements of A, B and C, respectively. This query $\overrightarrow{q} = [x, y, z]$ of $\pi = (a, b, c)$ yields two resulting states $\pi_y = \pi_y(\overrightarrow{q})$ and $\pi_n = \pi_n(\overrightarrow{q})$:

$$\begin{aligned}\pi_y &= \pi_y(\overrightarrow{q}) = (x, a - x + b, b - y + x),\\ \pi_n &= \pi_n(\overrightarrow{q}) = (a - x, x + b - y, y + c - z).\end{aligned} \tag{2}$$

We use the notation $\binom{k}{\leq m} = \sum_{i=0}^{m} \binom{k}{i}$. The *q-weight* of σ with type $\pi = (a, b, c)$ is defined by

$$w_q(\sigma) = w_q(\pi) = \binom{q}{\leq 2} a + \binom{q}{\leq 1} b + c. \tag{3}$$

Berlekamp [2] showed that if a state σ has a winning strategy with q queries then

$$w_q(\sigma) \leq 2^q. \tag{4}$$

The number $\mathrm{ch}(\sigma) = \mathrm{ch}(\pi) = \min\{k \mid w_k(\sigma) \leq 2^k\}$ is called the *character* of σ (or π). It is easy to see that

$$\begin{aligned}w_k(\sigma) &= w_{k-1}(\sigma_y) + w_{k-1}(\sigma_n),\\ w_k(\pi) &= w_{k-1}(\pi_y) + w_{k-1}(\pi_n).\end{aligned} \tag{5}$$

A query $Q = (A', B', C')$ of state $\sigma = (A, B, C)$ is called $[i, j, k]$*-like* if $i = \sharp A', j = \sharp B'$ and $k = \sharp C'$. A state $\sigma = (A, B, C)$ with type (a, b, c) is called *typical* if $b \geq a - 1$ and $c \geq k = \mathrm{ch}(a, b, c)$.

By the definitions of $q(n)$ and character, we have $q(n) \geq \mathrm{ch}(S_n, \emptyset, \emptyset)$. It is possible that $q(n)$ does not equal to the character of the starting state, as $q(n)$ depends on the state achieved after the first two queries. See Spencer [23]. Given an initial state $\sigma = (S_n, \emptyset, \emptyset)$ with type $\pi = (n, 0, 0)$, and $k = \mathrm{ch}(\pi)$.

Let $\pi_y = \pi_y(\overrightarrow{q}_1)$ and $\pi_n = \pi_n(\overrightarrow{q}_1)$ be the resulting states yielded by any first query $\overrightarrow{q}_1$; π_{yy} and σ_{yn} (resp. π_{ny} and π_{nn}) be the resulting states yielded by any second query $\overrightarrow{q}^{\,y}_2$ of π_y (resp. $\overrightarrow{q}^{\,n}_2$ of π_n), i.e.,

$$\pi_{yy} = (\pi_y)_y(\overrightarrow{q}^{\,y}_2), \qquad \pi_{yn} = (\pi_y)_n(\overrightarrow{q}^{\,y}_2), \qquad \pi_{ny} = (\pi_n)_y(\overrightarrow{q}^{\,n}_2), \qquad \pi_{nn} = (\pi_n)_n(\overrightarrow{q}^{\,n}_2).$$

After the first two queries $(\overrightarrow{q}_1, \overrightarrow{q}^{\,y}_2)$ or $(\overrightarrow{q}_1, \overrightarrow{q}^{\,n}_2)$, we obtain one of four states $\pi_{yy}, \pi_{yn}, \pi_{ny}$ and π_{nn}. We should try to choose the first two queries so that the biggest weight $Z_{max} = \max\{w_{k-2}(u) \mid u \in \{\pi_{yy}, \pi_{yn}, \pi_{ny}, \pi_{nn}\}\}$ can be as small as possible.

We choose the first two queries in the following way (this strategy coincides with Guzicki's strategy, see [12]): the first query $\overrightarrow{q}_1 = (t, 0, 0)$, where $t = \lceil \frac{n}{2} \rceil$ ($\lceil x \rceil$ denotes the smallest integer $\geq x$) and let π^o_y and π^o_n be the resulting states yielded by $\overrightarrow{q}_1$; the second queries $\overrightarrow{q}^{\,y}_2$ for π^o_y and $\overrightarrow{q}^{\,n}_2$ for π^o_n are given in Table 1.

Let $\pi^o_{yy}, \pi^o_{yn}, \pi^o_{ny}, \pi^o_{nn}$ be the four resulting states yielded by the first two queries stated above, and $Z^o_{max} = \max\{w_{k-2}(u) \mid u \in \{\pi^o_{yy}, \pi^o_{yn}, \pi^o_{ny}, \pi^o_{nn}\}\}$. The Lemma 1 shows that $Z_{max} \geq Z^o_{max}$, i.e., the first two queries stated above are weight-optimal.