Olympiads in Informatics, 2007, Vol. 1, 165–174

Development and Exploration of Chinese National Olympiad in Informatics (CNOI)

Hong WANG
Tsinghua University
100084, Beijing, China
e-mail: wanghong@tsinghua.edu.cn

Baolin YIN
Beijing University of Aeronautics and Astronautics
100083, Beijing, China
e-mail: yin@nlsde.buaa.edu.cn

Wenxin LI
Peking University
100871, Beijing, China
e-mail: lwx@pku.edu.cn

Abstract. This article presents a general overview of the historic development, exploration and practice of CNOI during the past 23 years. It includes: 1) some historical data recording the development of CNOI; 2) main contest activities organized by the Scientific Committee and Competition Committee of NOI of CCF, and some relevant management experiences; 3) the selection mechanism for the best contestants of CNOI; 4) the development and characteristics of a testing and evaluation system; 5) the development and characteristics of a visible team competition; 6) training of contestants and teachers, and the improvement and perfection of competition rules.

Key words: contest organization, contestant selection, evaluation system, visible contest.

1. Introduction

“The popularization of knowledge of computers should begin from children”, said in 1984 the former leader Deng Xiaoping, General designer of China’s reform and opening policy.

The China Computer Federation organized the 1st China Computer Programming Contest for Youth and Children in 1984, afterwards named the National Olympiad in Informatics (NOI). By the year 2006, a total of 23 NOI contests had been held successfully. As one of the earliest countries to participate the International Olympiad in Informatics (IOI), China has been present at every IOI contest since the first one, IOI1989. The effects of informatics olympiads have been demonstrated in promoting the popularization and raise of information technology in China’s middle and high schools, which has and will play an important role in the cultivation and selection of talents in informatics. The following data records the development of CNOI:

- in 2000, the 12th International Olympiad in Informatics (IOI2000) was successfully held in Beijing;
- during the past 18 IOI contests (to the end of the year 2006), Chinese contestants won a total of 42 Gold Medals, 17 Silver, and 11 Bronze;
- during the IOI contests held in the consecutive three years 2004-2006, all the Chinese contestants that participated in the contests were awarded Gold Medals;
- the year 2004 was the 20th Anniversary of China National Olympiad in Informatics, and the China National Symposium on Computer Education was held with the official proceeding published;
- 2006 NOI Contest included the first Visible Team Competition;
- contestants enrolled to join the CNOI Contests in 2006 reached nearly 80,000;
- the first Yearbook of Chinese National Olympiad in Informatics (CNOI2006) was officially published in 2006.

2. Main Contest Activities and Selection Mechanism of CNOI

2.1. *Main Contests and Activities of CNOI*

The NOI Scientific Committee and Competition Committee, under the guidance of the China Computer Federation, is responsible for the technical organization and management of NOI contest. A number of activities are run every year which are aimed at middle and high school students for enrichment and competition in computer programming. The main contests and activities are given in the Table 1.

2.2. *Selection Mechanism for the Best Contestants*

The CNOI establishes a strict selection mechanism and rules for the excellent contestants. It is based on multi-contests and paper defence. It guarantees that the best contestants can be selected by the mechanism.

Take the selection of IOI2007 China team as example. The process begins from NOI2006 during July of 2006. The top 20 contestants of NOI2006 become the members of the National Training Team (NTT) for the IOI2007. This is the first contest that forms part of China's formal IOI team selection process. Then we arrange 4 contests and structures in order to select the best 4 contestants from these 20. Each part has a different weight or score. These structures and their proportions are respectively:

Homework: 5%

NOI Winter Camp Testing and Competiton: 25%

Paper presentation and oral defence: 10%

China Team Selection Competition (CTSC): 60% (two contests, 30% each time).

Besides the score, we also check the comprehentive character and English proficiency of contestants. A personal statement and letter of commitment is requested for the contestants.

Table 1

Main contests and activities of CNOI

Activity	**When**	**What**	**Participantes and Size**
* China National Olympiads in Informatics (NOI)	July–August	Two day competition (5 hours for 3 tasks/each day) and one week activity similar to IOI. A team competition was added in NOI2006. The top 20 contestants from NOI form National Training Team (NTT) for the IOI of next year (candidates for the China team)	5 contestants each province. Totally 150 pers.
* NOI Summer Camp.	At the same time with NOI	The competition and activity is the same as NOI.	4 contestants each province. Totally 120 pers.
* National Olympiad in Informatics in Province (NOIP)	October–November	Preliminary competition: a multiple-choice / short-answer competition. Final competition: 3 hours for 3 tasks. Contestants are divided into middle and high school group	Nearly 80000 contestants participanted in NOIP2006.
Homework practicing and training for NTT	August–Jan.	Training and practicing by homework. Contestants will submit their solutions, and someone will give instant feedback.	20 contestants of NTT
* NOI Winter Camp	Jan–Feb	An intense one week training and a five-hour competition. The formal contestants will also participate in an oral paper presentation and defence $(10' + 5')$	Formal: 20 contestants of NTT Informal: 4 contestants each province
* China Team Selection Competition (CTSC) for the IOI	May	The final China Team Selection Competition. Two day competition (5 hours for 3 tasks each day) is similar to the IOI. The top six contestants will participante in an oral defence. The best four contestants will form China team for the IOI.	Formal: 20 contestants of NTT Informal: 3–4 contestants each province
* APIO (Asia Pacific Informatics Olympiad)	The 2nd Satuday of May	China Regional Competition of APIO held in one place organized by CCF. It is in parall with the CTSC from the year 2007. We took China Regional Competition of APIO as the first day competition of the CTSC in 2007.	20 contestants of NTT; top 50 of the last NOIP; and 1 contestant each province
Training before the IOI	August	Two training competitions with ACM/ICPC contestants for one week before the IOI	4 contestants of China team
Teacher and coach training	1–2 times each year	One week training course for algorithm design and programming skills, also including competition organization rules and evaluation system, etc.	Teachers, coaches and officials of provinces

3. The Development and Characteristics of an Evaluation System (Arbiter)

In order to guarantee the correctness and efficiency of evaluating the contestants' programs, the Scientific Committee of the Chinese National Olympiad in Informatics (SC of NOI) authorized the Group of Advanced Information Technology, Beijing University of Aeronautics & Astronautics (GAIT, BUAA) to develop an official evaluation tool, the Arbiter, seven years ago. After six years use in various competitions organized by the SC of NOI and similar events, the system has been proven to be a stable, comprehensive and trustworthy tool. Arbiter provides support in every phase of a competition, including task design, contest environment preparation, program evaluation, scoring and ranking, statistics and data backup.

Arbiter runs on various versions of the Linux operating system, including Redhat, Debian and Ubuntu. The system is based on a LAN with C/S architecture. The server is an independent computer communicating via a LAN with the clients of the contestants' PCs through out the contest. A contest can be divided into three stages: preparation, contesting and evaluation. In the preparation stage, the server cooperates with the clients to establish and update the contest settings, construct the language environments and contestants' accounts on the client PCs, and issue the contest data. During the contesting stage, the Arbiter clients on the contestants' PCs monitor and control the network communications of the PCs, and filter out forbidden packets according to the preset regulations. In the evaluation stage, the server commands the clients to evaluate the contestants' programs locally and then collects the results to form the score list, ranking list, and various analyzing forms as required.

Arbiter has the following characteristics and advantages:

- Flexibility and Efficiency

Arbiter enables the administrator to configure and control the contest in an easy manner. A contest is composed of a number of tests, and each test contains a number of tasks. A task can be of several types, such as a standard program, a program interacting with a library, or a results only task. A task can be evaluated with a number of items of evaluating data, each with configurable weight. The contestants are allowed to develop their programs. The administrator is able to modify the configuration of the system at any stage in case some client PCs are not working. Since the programs are evaluated in parallel on the client PCs, the evaluation will be finished very fast. Arbiter is highly adaptable to the hardware environment. In case there are not enough client PCs with a unique hardware configuration, the administrator is able to appoint a number of PCs with the same configuration as the evaluation machines so that the timing will be unique and impartial for all the contestants. Furthermore, as the evaluation is done on the contestants' PCs, there is no need for the contestants to submit their programs to the server. This avoids the network traffic jam, as happens on most of the Web base evaluation systems.

- Safety and Security

Much attention was paid during the design of Arbiter to processing the contestants' results safely and securely. Before the evaluation starts, the contestants' programs and data are uploaded to the server. A backup copy is also stored locally in a system directory. The backup will be used during the self check by the contestants. Arbiter safeguards the data with encryption and access controls, even the legal users are not permitted to access the data files directly. All the data operations must be done by using a system operation tool with authorization control.

In some contests, the contestants are allowed to access some strictly specified websites, while other network communications are forbidden. In order to control the network communications, the Arbiter clients monitor and filter the network communications on contestants' PCs. The filtering is based on the source and destination IPs and the type of the protocol of the packets. Only the pre-specified network communications are allowed.

- Independent to the Language Environments

Arbiter is independent of the programming language environments. The compliers and relevant tools are specified by the administrator during the preparation stage when the contest is under configuration. The command line options can also be specified at the same time for each operation. This enables Arbiter to meet new requirements in the future by supporting various programming languages and different types of the tasks, provided the relevant compilers and tools are available.

- Accuracy in Timing

The system can account and control the execution time of the program being evaluated in the 5ms time slice. The results show that Arbiter performs very stable evaluation and accurate timing with an error rate of less than 0.01% and timing error of less than 10ms.

- Self-Adaptive and Efficient Network Communications

In order to work in different network environments, Arbiter automatically explores and analyzes the topology of local networks. Consequently the best schemes are adopted to transfer data between the server and the clients. If the network supports broadcasting with a low packet drop rate, broadcasting mode will be used to send the data from the server to the contestants' PCs. This mode will provide high speed for data transfer. A typical test shows that 100MB data can be sent to over 100 PCs in less than 20s over a 100M Ethernet. If the PCs are cascaded in the network and broadcasting is forbidden, or the packet drop rate of the network is above a threshold, the system adopts a cascaded mode P2P data transfer when sending data from the server to the PCs, in order to transfer the data correctly with relatively high speed. In both modes, application layer checking will be done, and data will be re-transferred if there is any mistake.

- Data Import and Export

In order to meet the requirements by users for using other tools, such as MS Office, to process and display the contest data, Arbiter stores its data in the standard formats of CSV and PostScript. The data import and export functions are provided, and therefore all configuration files, contestants' information files, and score files can be easily imported and exported.

- Easy to Use

  The server of Arbiter is GUI based and running in an interactive mode, while the clients are running in a daemon mode without any direct interaction with the administrator. All functions are shown in the user area of the GUI of the Arbiter server. Any principal function can be selected within 3 clicks of the mouse buttons with clear guiding information. As both the server and the client are statically linked with the libraries, they are independent of the library versions on the target system.

- Comprehensive Functions

  The Arbiter system is accompanied by a series of supporting tools: a task verifying tool for checking the correctness of the testing data and the score evaluating plug-ins, a seat appointing tool for deploying the contestants over the PCs, and a user account and password generating tool. While both the seat appointing tool and the account and password generating tool are for the contest administrator to set a contest, the task verifying tool is for the task creators to test the evaluating points, the standard programs and time limits.

An aggregative evaluating tool has also been developed in order to meet requirements where a LAN based contest environment cannot be set. In this case, all programs are collected via other media, such as USB disks and email. The aggregative evaluating tool will run after the contestants' programs are collected and stored in a specified file hierarchy, and the evaluation and statistics will be performed in the same way as the Arbiter server.

4. The Development and Characteristics of a Visible Team Competition

In order to make the programming contest more interesting, attractive, and more understandable to the public, the Scientific Committee of the Chinese National Olympiad in Informatics (SC of NOI) authorized the Group of ACM/ICPC (International Collegiate Programming Contest) team of Peking University to develop a new style of programming contest where the running steps of the programs can be viewed on the screen. In 2006, we extended an open source software "Dominate Continent" and developed a contest named "risk" which is similar to the Java Challenge in ACM/ICPC. The difference is that "risk" accepts programs written in any programming language other than just Java in the Java Challenge. Details of "risk" are given below.

4.1. *What is the Meaning of "Visible"*

In the contest, we have several programs competing with each other to dominate as much land as they can. When the programs are running, the map is shown on the screen, see Fig. 1. There are four teams competing on the land. Each team is represented by a unique color. The teams play in turns. In each turn, teams play in a certain sequence. The play result of each step is shown on the map. In Fig. 1, the black circle with number "6" represents the army of team "PNJ_Y" and it is attacking and winning one of its neighbor land. Fig. 2 show the four teams in Fig. 1 competing on other maps.

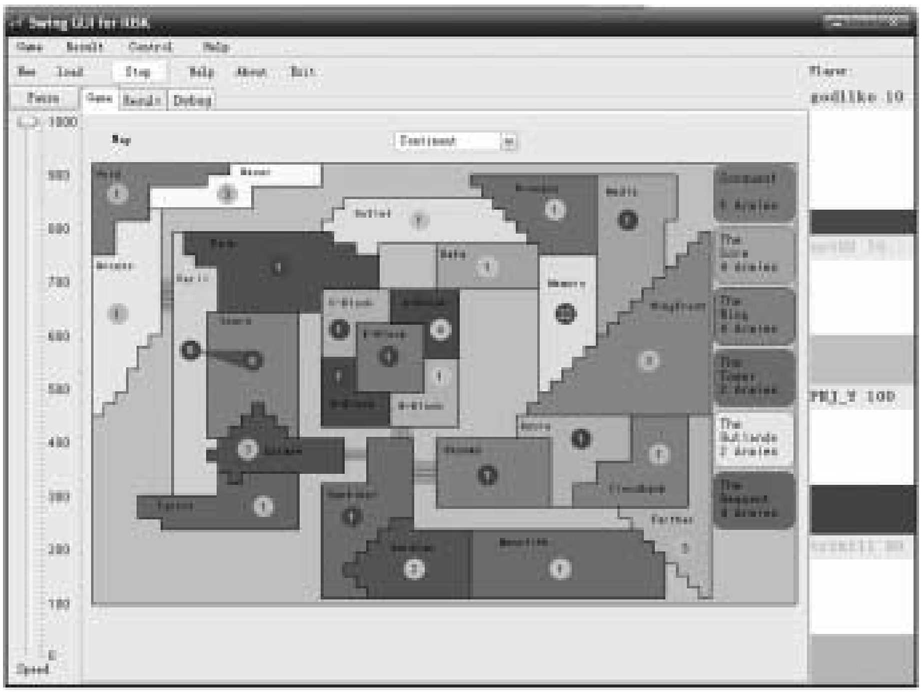

Fig. 1. Four teams (godlike, notHK, PNJ_Y, trikill) are competing on the land.

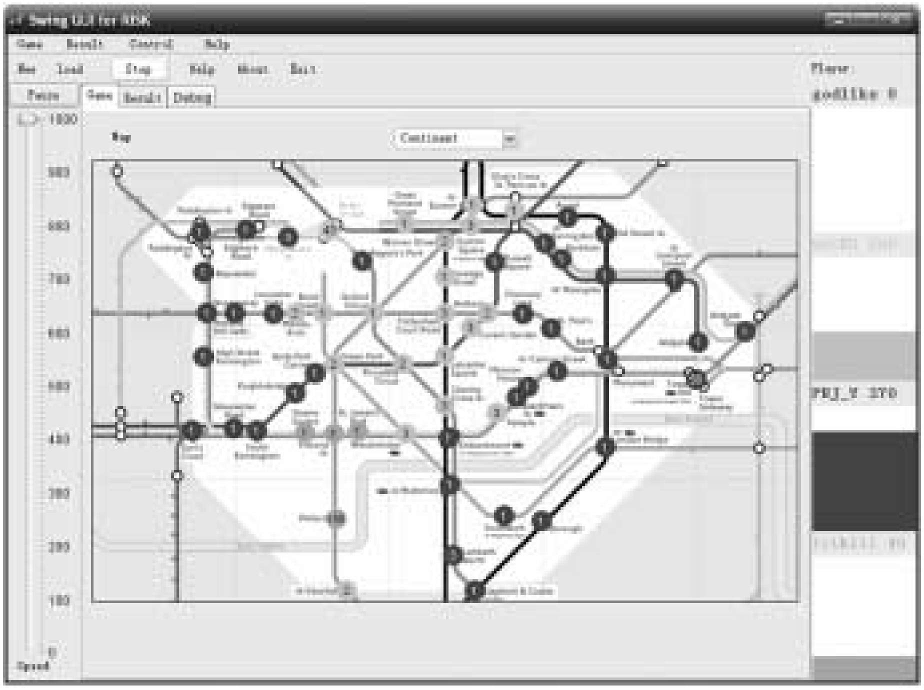

Fig. 2. Four teams competing on a train network.

4.2. *How to Organize the Contest*

The contest includes three stages, the first stage is round-robin; the second stage is qualifying; the third stage is final.

The round-robin has several rounds. In each round the system divides the teams into some small groups randomly. Each group has a separate game. In the same round, every group uses the same map. Maps in different rounds are different. Each game lasts three minutes. After a game, each attended team gets some points, which is equal to 10 times the number of countries it occupied, when the game ends. Each team's points are accumulated as a total score.

In the qualifying, the system divides the teams into small groups according to their scores in the round-robin. The top teams will not be placed into the same group. Each group plays for several rounds and only the champion is put through to the final.

The final still has several rounds, but only one group. The score in the final stage will be the final score.

The information is announced three months before the contest. All the provinces are invited to attend the contest. Each province may organize one team which may include at most five students. One week before the contest, all teams should submit their programs for the **Round-Robin** stage contest. Thereis a break between round-robin and qualifying, and a break between qualifying and final. Teams are permitted to modify their source code at any time, but only permitted to resubmit their source code during the breaks.

4.3. *Hardware and Software Used in the Competition*

Server: WinXP, JRE1.5, Python2.4.3,Tomcat5.5, FPC(Free Pascal Compiler)2.0.2, Dev C++ 4.0

Client: WinXP,JRE1.5, FPC(Free Pascal Compiler)2.0.2,Dev C++ 4.0

4.4. *Description of the Task*

Initial Army Placement

Every player rolls dice,to decide the order of play. Starting with the first player, everyone in turn places one army onto any unoccupied territory. Continue until all territories have been claimed.

Each player in turn places one additional army onto any territory he or she already occupies. Continue in this way until everyone has run out of armies. There is no limit to the number of armies you may place onto a single territory.

Playing

Whoever placed the first army takes the first turn.

Each player's turn consists of three steps, in this order:

1) getting and placing new armies;
2) attacking, if you choose to, by rolling the dice;
3) fortifying your position.

At the beginning of each turn, new armies you'll add to your territories based on

1) the number of territories you occupy;

2) the value of the continents you control;

3) the value of the matched sets of RISK cards you trade in;

4) the specific territory pictured on a traded-in card.

The above 4 steps are automatically calculated by the game and it will display how many armies you can place for that turn.

The task isribed in http://162.105.81.202/noip/noip_game/game/Risk_1.0.8.5.zip

5. Conclusion

This article gives some historic data recording the development and exploration of CNOI, and main contest activities organized by SCNOI and CCNOI of CCF. The selection mechanism for the best contestants has been proven to be effective. In addition, an evaluation system with some features, and a visible team competition used by NOI2006 are also presented respectively.

References

China Computer Federation (Eds.) (2007). *The Yearbook of Chinese National Olympiad in Informatics 2006* (*CNOI2006*), Henan Publisher Group of China.

A Summary of Chinese NOI Development Forum (2005). The file of SCNOI and CCNOI of CCF. Oct. 22–23, Beijing, China.

The Competition Rules and Measurements of Chinese NOI (2005–2006). The file of SCNOI and CCNOI of CCF.

Wang, H., B. Yin (2006). Visualization, antagonism and opening – towards the future of the IOI Contest. In *1st Workshop on Computer Science Competitions Reform*, Jan., Germany.

H. Wang received his PhD degree from the Department of Computer Science and Technology, Tsinghua University in 1993. He is currently an associate professor at Department of Computer Science and Technology, Tsinghua University. He also serves as the chairman of Scientific Committee of National Olympiad in Informatics of CCF.

B. Yin received his PhD degree from the Department of Artificial Intelligence, the University of Edinburgh in 1984. He is currently a professor at the School of Computer Science and Technology, Beihang University. He also serves as the vice chairman of Scientific Committee of National Olympiad in Informatics.

W. Li received her PhD degree from the Department of Computing, the Hong Kong Polytechnic University in 2004. She is currently a professor at Department of Computer Science and Technology, Peking University. She serves as a member of Scientific Committee of National Olympiad in Informatics of CCF. She is also the coach of ACM/ICPC Peking University team.