FIRBUSH NEWS 7, UNIV. EDINBURGH

# 1976 Canadian Computer Chess Workshop

by

T. A. Marsland
University of Alberta
Edmonton, Canada

In June 1976 a specialized workshop designed to measure the computational efficiency of a variety of computer-chess programs was held at the University of Alberta. Programs were submitted from six universities, McGill (OSTRICH), Toronto (CHUTE), Waterloo (TREEFROG, formerly RIBBIT), Alberta (WITA), Michigan (CHAOS), Colorado (SHRDLU),* and two from industry, BELLE (Bell Lab.) and COKO (Western Electric). Five of these programs were installed on an Amdahl 470V6, making the event unique in that this is the largest number of chess programs that have executed simultaneously on a single computer. The others used small machines in the Computing Science Department.

The bulk of the measurements were carried out by Ken Thompson of Bell Labs (en route from a year's leave at Berkeley), Steve Soule of Calgary and myself. Measurements were made of memory requirements, move generation speeds in different execution modes, and for the programs on the Amdahl instruction traces were taken. From the traces a cumulative graph of the (% usage frequency) × (instruction time) was drawn for each of the five programs. For these programs it was deduced that the effective CPU time per instruction tended to limits which were within 25% of each other. It was agreed that the differences attributable to the code generated by the compilers was probably small, and so stylistic factors such as use of constants, multiple index arrays, and rate of procedure invocation were dominant.

The technical details of the programs and their performance in the competition are given in Table 1. Tournament results are summarized in Figure 2. In the previous workshop (see Sigart Dec. 1975) in Calgary an attempt was made to compensate for differences in the effective speed of the machines being used by requiring that programs complete 40 moves within $7.5 \times 10^9$ memory references. By this means the influence of background load on the host machine could be dramatically reduced, and the effect of different CPU speeds equalized. As a practical matter this limit proved generous.

At the 1976 workshop, since most of the programs were using a common machine, a simple CPU limit was used. Modes of operation were chosen to ensure that the programs played, where possible, at an average rate of 40 secs. per move, while competing with each other for the system's resources. This method proved quite satisfactory, since no time limits were necessary, but a few of the games in the first round were abandoned as draws when two of the programs consumed excessive amounts of CPU and exceeded preliminary limits. Not all programs were really controllable;

(Continued on page 40)

*Now E.S.

| Program | Author and Institution | Machine Language | Memory size. Avg. CPU/move | OSTRICH | CHUTE | WITA | CHAOS | BELLE | COKO | SHRDLU | TREE-FROG | Total points |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OSTRICH | Newborn et al. (McGill) | QM1 (Nova) Assembler | 32 Kb - | | ½ (B) | 1 (W) | | 1 (W) | | | 1 (B) | 3½ |
| CHUTE | Valenti (Toronto) | Amdahl XPL | 392 Kb 39.8 secs | ½ (W) | | ½ (W) | 1 (B) | | | | 1 (B) | 3 |
| WITA | Marsland (Alberta) | Amdahl Algol W | 376 Kb 9.6 secs | 0 (B) | ½ (B) | | ½ (B) | | 1 (W) | | | 2 |
| CHAOS | Swartz et al. (Michigan) | Amdahl Fortran | 376 Kb 52.2 secs | | 0 (W) | ½ (W) | | | ½ (B) | 1 (B) | | 2 |
| BELLE | Thompson (Bell Labs.) | PDP 11/45 C + Assemb. | 42 Kb (67.7) secs | 0 (B) | | | | | ½ (W) | ½ (W) | 1 (W) | 2 |
| COKO | Kozdrowicki (Western Elec.) | Amdahl Fortran | 420 Kb 33.7 secs | | | 0 (B) | ½ (W) | ½ (W) | | 1 (W) | | 2 |
| SHRDLU (E.S.) | Courtois (Colorado) | QM1 (Nova) Assembler | 32 Kb - | | | | 0 (W) | ½ (B) | 0 (B) | | ½ (W) | 1 |
| TREEFROG | Hansen et al. (Waterloo) | Amdahl Fortran | 180 Kb 10.9 secs | 0 (W) | 0 (W) | | | 0 (B) | | ½ (W) | | ½ |

Table 1. Details of programs, and results.

ROUND I

1. CHUTE(A) V WITA(A): 1/2-1/2 (12 moves)
   172.0 secs/move 9.8 secs/move

   Game abandoned as a draw after 12 moves. Not only was CHUTE using an excessive average of 172 seconds for each move but also a FAKE-OS in compatability caused a premature termination and we were unable to restart properly.

2. COKO(A) V CHAOS(A): 1/2-1/2 (26 moves)
   58.0 secs/move 126.0 secs/move

   Also abandoned as a draw: both programs using excessive amounts of CPU time.

3. TREEFROG(A) V OSTRICH(N): 0-1 (23 moves)

4. BELLE(11) V E.S.(N): 1/2-1/2 (10 moves)

ROUND II

1. TREEFROG(A) V CHUTE(A): 0-1 (57 moves)
   7.4 secs/move 25.6 secs/move

2. WITA(A) V COKO(A): 1-0 (50 moves)
   10.6 secs/move 27.6 secs/move

3. E.S.(N) V CHAOS(A): 9-1 (62 moves)

4. OSTRICH(N) V BELLE(11): 1-0 (54 moves)

   TREEFROG seemed to use its CPU time very erratically, either less than 2 secs/move, or an average of 17.6 secs/move.

ROUND III

1. CHAOS(A) V WITA(A): 1/2-1/2 (23 moves)
   47.6 secs/move 10.6 secs/move

2. CHUTE(A) V OSTRICH(N): 1/2-1/2 (34 moves)

3. E.S.(N) V TREEFROG(A): 1/2-1/2 (57 moves)

4. BELLE(11) V COKO(A): 1/2-1/2 (48 moves)
   27.5 secs/move

   A remarkable round of repetion draws.

   Table 2: Results, round by round, with average move times where available (see text). The machine used is indicated in parenthesis after the name of each program, i.e., A = Amdahl 470/V6, N = NOVA (Nanodata QM-1), 11 = PDP 11/45. See following page for results for Round IV.

Table 2 cont'd

ROUND IV

| | | | |
|---|---|---|---|
| 1. | COKO(A)<br>58.7 secs/move | V | E.S.(N): 1-0 (39 moves) |
| 2. | CHAOS(A)<br>22.35 secs/move | V | CHUTE(A): 0-1 (38 moves)<br>27.52 secs/move |
| 3. | OSTRICH(N) | V | WITA(A): 1-0 (23 moves)<br>23.6 secs/move |
| 4. | BELLE(11)<br>65.1 secs/move | V | TREEFROG(A): 1-0 (39 moves)<br>15.75 secs/move |

In this round, TREEFROG made better use of its time. Note the dramatically increased time by WITA, caused by exposed king.

---

Continued from page 37

for example, WITA generated its moves with little regard to time, while COKO tried to use all its time in the first few moves. TREEFROG was most unpredictable, but good control was eventually exercised over CHUTE and CHAOS.

Two other points of technical interest are that CHUTE was executed under a special monitor (FAKE-OS written by C. Benet), which allows object programs developed for IBM OS environments to execute under the Michigan Terminal System (MTS) without recompilation. The overhead associated with this feature is less than 10% and saved us from first installing the XPL language on our machine. Special arrangements were also necessary for OSTRICH and SHRDLU (E.S.). These programs require a NOVA computer, but were executed in emulation mode on a Nanodata QM1 (part of our microprogramming research facility). Because the emulator speed is approximately half that of a Nova 1200 no time limits were forced on these programs. Both were made to play slightly faster than usual, and this may have affected their performance. Certainly OSTRICH dominated all its games, even though it had lost to both E.S. and TREEFROG in recent ACM tournaments. The simplest program to use was BELLE which performed creditably and without special attention. There were no software or hardware malfunctions throughout the duration of the tournament.

The experience gained in this workshop is providing insight into ways of measuring the intrinsic efficiency of the algorithms employed by these programs.

-ooOoo-