

VIDEOPAC

C7401 < Videopac Hardware > C7420

C7010 Chess Module

Title: Chess Module
Number: C7010
Released: 1982
In: Europe
By: Philips
US Title: -
Euro Rarity: ER
US Rarity: NR
Special: Only plays on Videopac/+ and Jopac consoles!

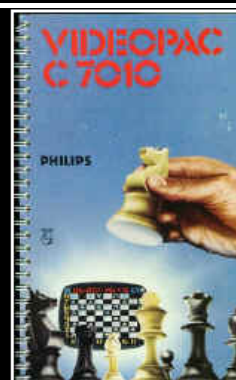


The C7010 is one of the harder to find pieces of Philips Videopac hardware, especially complete with box and manual. There are at least two different box and manual versions, one for the main European market, one for the North-European market.



Since the G7000 didn't have enough RAM memory and computing power for a chess program, Philips decided to equip the chess cartridge with an add-on computer module that sits on top of the G7000 console. The module gets its 5V and 300mA from the main console. The chess program itself resides in an 8244 EPROM chip

on the cartridge and is connected via input/output latches to the C7010 CPU's databus. The extra processing power is provided by an NSC800 8Bit CMOS microprocessor running at 4.43 MHz. The C7010 also has 2K of RAM and 8K of ROM (OS or part of the Chess program itself?).

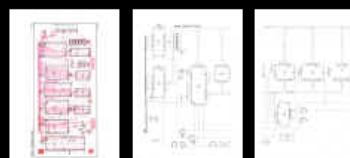


Besides some logic chips there are two MM82PC12N (output latch and address latch) and an empty socket. This socket was unused on the boards of all the four Chess units I opened so far - maybe there were some other versions planned? The empty socket has all the address lines for another 8K.

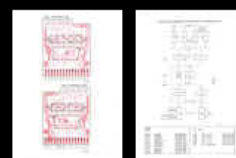
From the looks of it the C7010 could have been adapted to run other programs - an interesting possibility. All the C7010s I've seen so far have been of type C7010/05 but the service manual mentions 00, 05, 16, 18, 19 as possible versions - not sure what the differences are. If you have a different version please let me know and send me a scan - thanks! And if you're really brave please open up your C7010 and send me a chip listing - thanks!!!

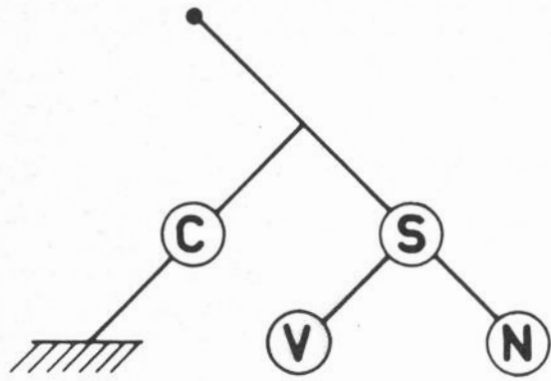


In ads the C7010 was touted as the 'serious' chess cartridge among the various videogame consoles, but it was quite pricey and sold at DM 300 - 400, about 4 - 5 times the price of the Atari 2600 chess cartridge (the famous cheating one) and thus putting it well out of reach of young videogame players and targeting the adults instead.



The C7010 manual is very detailed and provides a good introduction to the game of Chess.





Computerschaak

3^e jaargang nr.3

juni 1983

Zoeklicht op het Philips schaakprogramma

Het is eindelijk zover. Philips kwam uit met het schaakmoduul C 7000 voor zijn G 7000 videospelcomputer. Het eerste Nederlandse commerciële programma en wel van Wim Rens (Gambiet '80). Niet al te sterk spelende schakers kunnen in kleur op hun TV-scherm schaakplezier beleven. Wij zullen deze nieuwe Philips-telg eens aan een nadere beschouwing onderwerpen en ook laten spelen in een vierrondig toernooi.

Bezit u een Philips video spelcomputer? (Prijs f 279,00-f 299,00) Of wilt u die aanschaffen? Nou dan kunt u voor circa f 449,00 (adviesprijs) een schaakmoduul kopen en daarmee op uw TV-scherm in kleur schaak spelen. Het is het Gambiet '80 programma van ons lid Wim Rens, één van onze betere Nederlandse schaakprogrammeurs. Helaas met zijn oudere Gambiet '80 programma, want de nieuwere versies Gambiet '81, Gambiet '82 (Nederland-kampioen) en inmiddels Gambiet '83 spelen natuurlijk sterker dan de '80 versie. Philips had om begrijpelijke technische redenen echter veel tijd nodig dit schaakprogramma aan de videospelcomputer aan te passen. Dat was ook niet zo eenvoudig. Het programma kent zes spelniveau's, toont zijn analyses, geeft veel informatie op uw TV-scherm. De letters en cijfers zijn groot en duidelijk afgebeeld, de schaakstukken wat minder



J. Louwman introduceert het Philips-programma op de middag van de jaarvergadering.

duidelijk, doch dat went snel. Velen vinden het prettig op het TV-scherm te schaken. Vast staat, dat geen enkele videospelcomputer van een ander merk een schaakprogramma bezit van een ook maar te benaderen speelsterkte als bij dit Philips-programma het geval is. Daar moet u niet uit afleiden, dat dit programma zo goed speelt als de huidige gemiddelde schaakcomputer. Dat is n.l. niet het geval. De Elorating ligt bij toernooilevel (level 1, 2 minuten per zet) ongeveer op 1600 à 1650 Elopunten. Voor zwakkere en middelmatige schakers is dat echter voldoende. Het Philips-programma speelt ongeveer even sterk als het Sargon 2,0 programma. Toch ontbreken in dit moduul enkele zaken, welke niet zouden mogen ontbreken.

Spelniveau-verandering tijdens de partij is niet mogelijk en het is eveneens onmogelijk willekeurige schaakstellingen in te voeren. Ook is er geen 'play back' (terugspeelmogelijkheid), zaken die praktisch elke schaakcomputer, hoe goedkoop ook, wel bezit. Dat geen spelniveau-verandering tijdens de partij mogelijk is, betekent extra nadeel, omdat de computer bij een geringer aantal schaakstukken op het bord steeds sneller (dus slechter) gaat spelen. Dat komt door de vrij vaste plydiepte-koppeling aan het gekozen spelniveau. Aan te bevelen is dus in een hoog niveau te spelen, ondertussen kijkt u tijdens zijn nadenken op een ander kanaal naar een TV programma!

Toch kunnen wij trots zijn op dit eerste Nederlandse schaakcomputerprodukt. Philips stond ook niet voor ogen, uit te komen met een topprogramma. Men maakte een bescheiden begin en het is uiteraard vaststaand, dat onze elektronische gigant in de toekomst tot heel wat meer in staat zal blijken te zijn.

Dit Philips-programma speelde in een vierrondige competitie tegen Milton Robot (winnaar), Mephisto Junior en het Sargon 2,0 programma, ditmaal spelend in de VIC 20 personal computer.

Uit de resultaten blijkt duidelijk, dat Philips niet is opgewassen tegen de gemiddelde schaakcomputer, maar wel tegen één der sterkste schaakprogramma's dat commercieel voor personal computers beschikbaar is. Fabrikant: Philips Nederland te Eindhoven. Prijs inclusief videospelcomputer ± f 700,00 of iets lager. De toernooitabel treft u onderstaand aan.

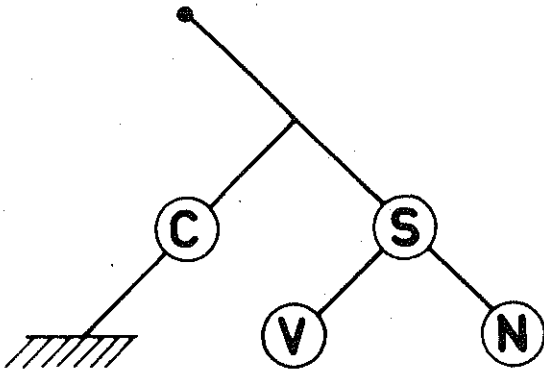
Schaakcomputer	1	2	3	4	Tot. punten	%	Gem. bed.tijd per zet	Prijs in gulden	Opmerkingen
1. Milton Robot	□	2½	3½	3½	9½	79.2	1'50"	± 1750	robot-uitvoering
2. Mephisto Junior	1½	□	2½	3	7	58.3	1'40"	399	zakformaat
3. Philips C 7000	½	1½	□	2	4	33.3	1'35"	449	+ computer à 279
4. Sargon 2.0 (Vic.20)	½	1	2	□	3½	29.2	1'54"	89	+ computer à ± 650

Remise-percentage 33⅓%. Gemiddeld aantal zetten per partij: 69. Levels: toernooi 2 min. per zet. Totaal aantal gespeelde partijen per computer: 12.

Conclusies

Philips levert met dit schaakmoduul voor zijn spelcomputer veruit het beste schaakprogramma ten opzichte van de concurrentie. Schaakprogramma's voor andere videospelcomputers (b.v. Atari e.d.) spelen aanmerkelijk zwakker dan die van Philips en verdienen het predicaat schaak nauwelijks. Het Philips-programma is uniek en interessant (op TV-spelen), ook al omdat, ondanks de foutjes voorheen opgesomd, zorg aan de uitvoering is besteed. Het is zeker geen aanbieding van nul en gener waarde. De nadelen dient men voor lief te nemen. Ik hoop oprecht, dat Philips op de ingeslagen weg voortgaat. In ons land met de verreweg grootste computerschaakorganisatie ter wereld, dient zo'n grote belangrijke industrie als Philips is, eenvoudig met computerschaak niet achter te blijven! En tot Philips zeg ik: die schoonheidsfoutjes waren niet nodig geweest.

J. LOUWMAN



Computerschaak

1^e Jaargang nr. 2

Juni 1981

voortreffelijke reparatiewerkplaats met zeer deskundig technisch personeel. Deze importeur blijkt trouwens toch een bonafide groothandel te zijn. Vooral degenen die in het oosten en zuid-oosten van ons land wonen, kunnen zonder bezwaar een schaakcomputer in de Bondsrepubliek kopen; omdat daar ook service gegeven wordt. Men betaalt soms (niet altijd) ook aldaar minder dan in Nederland. Het is jammer, dat ik dit advies moet geven, want het is eigenlijk een anti-Nederlandse zaak in deze tijd van werkloosheid. Importeurs en detailhandel dienen zich echter te matigen in de winstmarges en prijzen te hanteren die in elk geval niet aanzienlijk uitgaan boven de prijzen in het buitenland. Wij als CSVN wensen ook een vuist te maken.

De probleemschakers maak ik erop attent, dat mat in 4-problemen slechts kunnen worden opgelost door het Supersysteem 3, de Intelligent Chess en de Mephisto. De Mephisto kan geen minor-promotie problemen oplossen, de beide andere computers wel.

Tot besluit van deze bijdrage wil ik u erop attent maken dat ik slechts aan hen die lid zijn van de CSVN telefonische vragen kan beantwoorden over schaakcomputers en dan uitsluitend tussen 19.00 en 20.00 uur op maandag t/m donderdag van elke week. Buiten deze tijden kan en wil ik geen inlichtingen geven, omdat in de afgelopen weken de hele dag door de telefoon rinkelde om informatie en dat voor mij te belastend is. Mijn telefoonnummer, met inachtneming van het bovenvermelde, is 010 — 207343. Bij het verschijnen van dit nummer ben ik in het buitenland. Na 6 juli ben ik weer tot uw dienst. Vraagt u svp geen overbodige zaken, die bijv. reeds duidelijk uit mijn-publicaties blijken.

Tot de volgende keer dus en..... mopper niet te veel als de schaakcomputer u eens in de val laat lopen wanneer u tegen hem schaakt. Troost u dan, het overkwam diverse malen

JAN LOUWMAN

COMPUTERSCHAAK EN ARTIFICIAL INTELLIGENCE

Een veel gehoorde vraag is: Wat is de betekenis van Computerschaak in breder verband. Het is niet gemakkelijk daar in kort bestek een antwoord op te geven. Dat komt omdat Computerschaak op zich behoort tot het gebied van de Artificial Intelligence. Eenmaal op dat gebied aangekomen rollen de belangrijke vragen in veelvoud over de mens heen. We zullen er enkele opnoemen:

- 1 Wat is intelligentie?
- 2 Kunnen computers intelligent zijn?
- 3 Wat is denken?
- 4 Kunnen computers denken?
- 5 Wat is leren?
- 6 Kunnen computers leren?
- 7 Wat is kennis?
- 8 Is kennis eigenlijk principieel mogelijk? Zo ja, dan:
- 9 Wat is structuur, de samenhang van kennis?
- 10 Wat is de relatie tussen de samenhang van de kennis en de neerslag van het gekende in recepten van handelingen?

Het is duidelijk dat de hier opgesomde vragen van belang zijn voor het maken van computerprogramma's, maar daarnaast zijn het ook kernvragen uit de psychologie en/of uit de

filosofie. Er zijn daarom ook mensen die menen dat Artificial Intelligence een onderdeel is van de filosofie.

De bovengenoemde vragen kunnen niet zo maar beantwoord worden. Zelfs als we ons gebied verkleinen — en daar kleven weer andere filosofische bezwaren aan (het holistische argument!) —, dan kunnen we nog niet met de vragen uit de voeten. Laten we het schaakspel als voorbeeld nemen om te laten zien hoe moeilijk een en ander wel niet is: Wat is schaakkennis? etc. En als we van al die zaken al een beetje weten wat het voorstelt, dan blijkt het nog ontzettend moeilijk deze zaken adequaat in een computerprogramma te representeren. Op dat gebied wordt juist tegenwoordig zeer veel onderzoek verricht.

Het schaakspel is niet het enige deelgebied waarop systemen ontwikkeld worden die tot expert-niveau zullen moeten reiken. Op het gebied van automatisch vertalen werkt men eveneens hard aan vooruitgang en bij de medische diagnostiek heeft men al systemen ontworpen die in het stellen van de diagnose in bekwaamheid de arts evenaren.

Onderzoekers op computerschaakgebied mogen dan op een heel klein gebied hun activiteiten ontplooiën en de directe toepassing mag ver te zoeken zijn, zeker is dat als het hen gelukt om een schaakmachine te creëren, die sterker speelt dan de wereldkampioen, we een doorbraak hebben in Artificial Intelligence:

een machine die betere beslissingen neemt dan de mens. Dat betekent dat zo'n machine beter is dan de mens in het schaakdenken. De grote vraag is: hoelang zal het nog duren aler het zover is.

Jaap van den Herik

GAMBIET — Een schaakprogramma voor een Personal Computer

INLEIDING

Gambiet/80 is een schaakprogramma geschreven in Z80-assembler voor de TRS-80 en als zodanig in de handel verkrijgbaar. De auteur van Gambiet, tevens auteur van dit artikel, werkt momenteel aan een wat grotere broer: Gambiet/81. Is Gambiet/80 nog 9 Kbytes groot, Gambiet/81 wordt minimaal 20 Kbytes. Of de prestaties evenredig met de grootte toenemen, zal de toekomst nog moeten leren.

In een serie artikelen zal getracht worden op luchtige wijze te vertellen hoe Gambiet/81 tot schaken in staat is. Deze eerste aflevering bespreekt een aantal grondslagen die gelden voor nagenoeg alle schaakprogramma's op microcomputers. Generatie, evaluatie en selectie van zetten spelen daarbij een belangrijke rol. Maar we beginnen met iets waar het allemaal om begonnen is: het bord met de stukken.

DE STELLING

Om te kunnen schaken moet een programma op een bord kunnen kijken. De realisatie daarvan is eigenlijk nog het eenvoudigste van alles. Gambiet's schaakbord bestaat uit een serie computer-geheugen-lokaties, waarvan elke lokatie overeenkomt met een veld van een echt bord.

Op elke lokatie kan je een getal opbergen en verschillende getallen symboliseren verschillende getallen symboliseren verschillende stukken.

Bijvoorbeeld: het getal 0 op een lokatie betekent dat het bijbehorende veld leeg is; het getal 3 betekent dat er een witte loper staat. Aldus is op elk moment de stelling nauwkeurig vastgelegd op 64 geheugenplaatsen.

DE ZET-GENERATOR

Van een schaakprogramma dat alleen maar op een bord kan kijken raakt niemand erg opgewonden. Men wil er ook nog eens een zet uit zien komen. Derhalve dient het programma in staat te zijn een lijst met alle mogelijke zetten te produceren. Een populaire techniek is dat die zetten worden berekend. Niet zo'n vreselijk briljante gedachte, want computers zijn van huis uit nu eenmaal betere rekenwonders dan schaakwonders. Het concept is simpel.

Een voorbeeld: voorzie alle velden van het bord in de volgorde a1, b1,, h8 van de nummers 1 t/m 64 en je kunt iedere verplaatsing uittellen. Zo ligt het veld schuin-rechts-voor altijd 9 velden verder.

Ook Gambiet gebruikt deze techniek, waarvan de hoofdlijnen zijn:

- Het bord wordt afgezocht naar hetzij witte hetzij zwarte stukken.
- Voor elk gevonden stuk wordt een bij dat stuk behorende tabel met bewegings-richtingen opgezocht.
- Alle mogelijke velden waarnaar het stuk zich in principe zou kunnen verplaatsen worden berekend.
- Alle berekende verplaatsingen worden verder getest en bij gebleken legaliteit in een tabel met gegenereerde zetten geplaatst.

DE ZET-EVALUATOR

Een programma dat legale zetten kan genereren kan al reglementair schaken. Aan zet wordt naar willekeur één van de zetten gekozen. Wij vrezen echter dat zo'n programma nogal eens zal verliezen. Het zal tenminste de goede van de slechte zetten moeten kunnen onderscheiden. Derhalve verzinnen we wederom iets met getallen: voor elke zet wordt een puntenwaardering afgeleid. De wijze waarop, de evaluatie-functie, daar is veel discussie over mogelijk. Gambiet's manier is ongeveer als volgt:

1 punt = zet leidt tot konings winst
255 punten = zet leidt tot konings verlies

De tussenliggende punten zijn voornamelijk gebaseerd op materiaal winst/verlies en in mindere mate op positionele voor/nadelen. We zullen er nog uitgebreid op terugkomen.

DE ZET-SELECTOR

Een beginnend schaker wordt al direct geconfronteerd met "vooruitdenken". Ook een menselijke schaker heeft een soort evaluatie-functie; althans hij gebruikt het één of andere inzicht om een zet als goed of slecht te beoordelen. We weten allemaal dat zo'n inzicht nogal eens wordt bijgesteld als we de zet eventjes daadwerkelijk op het bord uitvoeren. Maar dat mag niet en we moeten ons beperken tot het uit het hoofd doorrekenen van varianten. Voor een schaakprogramma is dat allemaal geen punt. Stukken worden over het geheugen-bord verplaatst zoals de computer het uitkomt, of liever gezegd, zoals de zet-selector het uitkomt. Er is geen reglement dat dat verbiedt.

Voor de ontwikkeling van efficiëntere methoden om varianten door te rekenen heeft bijgedragen tot de vooruitgang van schaakprogramma's in het algemeen. De kracht van een computer is namelijk zijn tempo. Het programma "Belle" bekijkt 160.000 stellingen per seconde. Gambiet doet het met een paar minder: 75. (niet 75 minder maar 75 stellingen/sec.).

Dus per zet van 3 minuten bekijkt Gambiet zo'n 13.500 stellingen. Dat zou prachtig kunnen zijn, ware het niet, dat misschien 13.400 van al die zetten door een normaal mens nooit in overweging zouden worden genomen. Hier hebben we dan meteen de zwakte van schaakprogramma's te pakken; hoog tempo prima, maar dan wel op een zinvolle wijze. In de volgende sectie verklaren we twee principes, minimax en alfabet, waarmee de zet-selector er toch nog het beste van tracht te maken.

MINIMAX

Stel de computer speelt met wit en kan drie zetten doen. We noteren die zetten als A, B en C..

Stel wit doet zet A, waarna zwart de keuze heeft uit drie zetten, genoteerd A1, A2, A3.

Stel ook dat op zet B de antwoorden B1 t/m B3 mogelijk zijn en, u raadt het al, op zet C de antwoorden C1 t/m C3.

Op deze manier hebben we een soort boom van zetten geconstrueerd. Een boom met 3 takken en aan elke tak weer 3 uitlopers.

Op elk van die 9 uitlopers laten we de evaluatie-functie los.

Stel A1 = 50 punten, A2 = 100, A3 = 20 punten. Welke zet zou zwart dus doen als wit zet A zou spelen? Juist, zet A2. Althans, als zwart verstandig speelt, maar met het uitgangspunt dat zwart erg dom speelt kom je waarschijnlijk niet zo ver.

Stel B1 = 80, B2 = 80, B3 = 90. Dus als wit zet B speelt dan speelt zwart B3 met 90 punten.

Welke zet is nu beter voor wit, A of B? Zet B natuurlijk, want dan haalt zwart ten hoogste 90 punten i.p.v. 100 na zet A.

Voor wie dit niet allemaal direct gelooft, echte wetenschappers hebben dit kunnen bevestigen en hebben het bijbehorende taaltje bedacht: minimax. Wit wil het maximale en gunt derhalve zwart slechts het minimale. C'est la vie, n'est ce pas?

In de praktijk wordt natuurlijk een veel grotere boom geconstrueerd en aan elke uitloper weer nieuwe uitlopers geknoopt, en zo gaan we door tot we het welletjes vinden. Een gigantische taak, maar de computer is er goed voor. Alle uiteindelijke uitlopers gaan we evalueren en dan klauteren we weer terug al minimaximaliserend.

MOEILIKHEIDJE

Aspirant schaakprogrammeurs die al direct met de minimax-methode aan de slag willen, zullen het volgende moeilijkheidsje tegen komen. Laten we er eens van uitgaan dat Gambiet twee zetten diep moet denken. Dat is helemaal geen zware eis, net genoeg bijvoorbeeld om mat in twee zetten te zien. Twee hele zetten (vier halve zetten) dat is een zettboom met vier lagen (de vakterm voor "laag" is "ply"): 1e zet Gambiet, 1e zet tegenstander, 2e zet Gambiet, 2e zet tegenstander.

Laten we er vervolgens eens van uit gaan dat er gemiddeld 32 zetten per beurt zijn te vinden. Dat betekent een zettenboom met $32 \times 32 \times 32 \times 32 =$ ongeveer 1.000.000 uitlopers. Die moeten allemaal geëvalueerd worden. Gambiet doet er 75 per seconde; dat maakt dan $1.000.000 / 75 = 13.333 \text{ sec} = 222 \text{ minuten}$. We praten hier dus over één zet.

We vinden dat wel wat lang voor een "four ply depth search" om er maar eens wat Engels tegen aan te gooien. In de volgende alinea zullen we eens bekijken of er nog wat aan te doen is.

ALFABETA

Duidelijk is dat we het snoeimes in de boom moeten gaan zetten; "pruning" heet dat in het Engels. Eén van de technieken daartoe heet alfabeta, die we weer illustreren aan de hand van onze zettenboom met de 9 uitlopers.

We weten al dat zet B betere resultaten oplevert dan zet A. Nu gaan we verder met de uitlopers van C. We evalueren eerst zet C1, en wat blijkt, zwart krijgt hiervoor 110 punten. En nu komt het: het is verder zinloos om de zetten C2 en C3 nog te evalueren. Wit zal namelijk zet C nooit doen, immers alleen al met zet C1 krijgt zwart meer punten dan hij ooit op zet B zou krijgen. Dit is de gedachte achter A, B-pruning. Het aantal te onderzoeken varianten wordt aanzienlijk gereduceerd waarbij de computer tot dezelfde conclusie komt als wanneer hij alle varianten had doorgerekend.

Voorwaarde voor enige reductie is wel dat de goede zetten het eerst worden bekeken. Stel eens dat we eerst zet C2 = 40 punten, dan C3 = 40 punten, en dan pas zet C1 = 110 punten hadden bekeken. Dan hadden we nog bij de zetten C2 en C3 gedacht: dat ziet er gezond uit en rustig doorgeëvalueerd met variant C. Pas bij C3 hadden we de variant verworpen, maar ondertussen waren wel alle uitlopers bekeken. We staan met deze conclusie voor een aardig dilemma:

We klauteren het snelst door de zettenboom als we eerst de beste en dan de slechtste zetten bekijken.

We weten pas zeker welke de goede en welke de slechte zetten zijn als we zijn uitgeklaard.

Door dit soort problemen wordt een serieus schaakprogrammeur op de rand van waanzin gebracht. Maar om u gerust te stellen; er zijn vele van dit soort problemen.

Wim Rens

P.S. Leden van de CSVN die de beschikking hebben over een TRS-80/ 32K met floppy disk, en die sterk geïnteresseerd zijn in Gambiet/81, worden verzocht contact met mij op te nemen.

W.J. Rens, Haringvlietstraat 213, 3313 ED DORDRECHT, 078 - 162760

COMPUTERSCHAAK IN DE PROBLEMEN

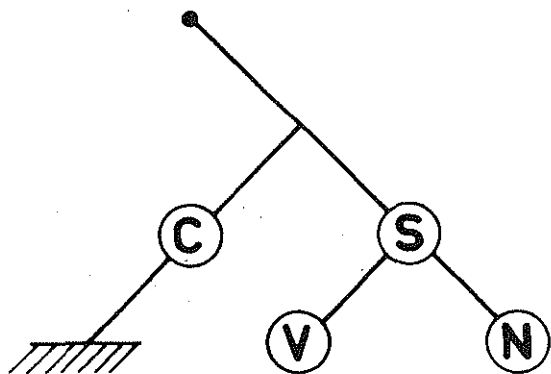
De titel van dit stukje is wat misleidend. Het zal inmiddels genoegzaam bekend zijn dat computers partijschaak kunnen spelen waarbij de sterkste schaakcomputers overigens al snel in de problemen komen wanneer zij met meesters en grootmeesters te maken krijgen. Veel is hierover reeds geschreven.

Er is echter een andere belangrijke en toch vrij onbekende tak van het fraaie schaakspel, die zich bezig houdt met het componeren (zo heet dat daar !) en oplossen van schaakproblemen. Hierover nu gaat dit artikel, waarbij ook de computer met zijn haren (?) erbij wordt gesleept.

Mijn belangstelling voor computers is reeds meer dan 25 jaar oud. Toen bestond reeds de wens ooit eens een schaakcomputer te construeren. Maar helaas mijn financiële middelen en kennis schoten verregaand tekort. Het was mij bijv. een raadsel hoe je aan zo'n verzameling elektronische onderdelen (al of niet intelligent gerangschikt) kon wijsmaken dat bijv. een toren de ene keer van c1 naar c8 mag springen maar bij een andere stand wanneer een stuk op de c-lijn staat dit niet mag doen, om nog te zwijgen van zulke moeilijke zaken als penning, rochade, en passant en promotie. Veel hoop dat dit ooit gerealiseerd zou worden had ik dan ook niet. Wie schetst echter mijn verbazing toen opeens eind 1974 in een Amerikaans tijdschrift de eerste betaalbare hobbycomputer werd aangekondigd en wel de Altair van MITS. Bestellen was snel gedaan, maar 9 maanden later was de nieuwe boreling nog steeds niet aan mijn adres verstuurd. Na nog diverse mislukte pogingen lukte het mij uiteindelijk een KIM computer te veroveren (ca 5 jaar geleden). De kenner weet dat deze micro bestaat uit een printplaat ter grootte van een blad papier (A4), voorzien van een eenvoudig toetsenbordje met display van 6 cijfers. Dit blote geheel moet dan echter nog verbonden worden met diverse trafo's, cassetterecorder en in mijn geval bovendien nog met een extra geheugen van 8.000 bytes. Met twee linkerhanden en een gezond gebrek aan elektronische kennis (mijn belangstelling is nl. alleen gericht op het programmeren) heb ik dit geheel toen met elkaar verbonden en ingebouwd in een schilderskastje. Zo'n klus hoop ik nooit meer te doen, maar het was toen de enige weg tot verwezenlijking van mijn jongensdroom, nl. eens een computer zo ver te krijgen dat hij schaakproblemen kon oplossen.

Inmiddels had ik het Jaarboek 1974 van de Nederlandse Bond van Schaakprobleemvrienden gelezen, dat zeer snelle grote computers in staat waren schaakproblemen, met als opgave mat in 2 zetten, in enkele seconden op te lossen (nog niet zo lang daarvoor was men al blij wanneer dat in 1 à 5 minuten kon). Het snelste in Europa was waarschijnlijk Dr. Zagler, die op de Telefunken Tr440 van de Technische Hogeschool in München 5 problemen (mat in twee zetten) oploste in 19,9 seconden. Drs. H.Je Grand vermeldde in genoemd jaarboek: "Daar de TR 440 een zeer snelle machine is, mag aangenomen worden, dat op korte termijn geen significant hogere snelheden bereikt zullen worden".

De moed zonk mij in de schoenen. Allereerst had mijn machientje in vergelijking met bovengenoemde reus maar een bescheiden snelheid van 300.000 instructies per seconde, maar bovendien moest ik nog vertrouwd proberen te raken met een zeer moeilijk vak t.w. programmeren in machinetaal. Pas na een half jaar voelde ik mij voldoende vertrouwd met dit programmeren - ten koste van de nodige frustraties - om een poging te wagen een programma te maken voor het oplossen van schaakproblemen.



Computerschaak

1^e Jaargang nr.3

September 1981

f 75,- en f 100,-. De **Executive** gaat f 499,- kosten. De **SciSys MK4** is de opvolger van het Super System 3, verkrijgbaar met LCD schaakbord (f 470,-) en thermische printer (f 470,-). De prijs van de MK4 zonder accessoires is f 680,-.

Tenslotte de **MK5**, die over een 24k.Rom-programma beschikt, met vele snufjes (f 1349,-). Het ARB hierbij kost f 899,-. Afgevraagd moet worden, of de PTT hiervoor toestemming geeft, want het Auto Response Board werkt met zender-signalen (frequenties).

Challenger brengt de nieuwe **Champion Challenger Sensory Voice**, met een volkomen nieuw, sterk programma, veel beter dan de oude Sensory Voice, die nogal eens faalt. Volgens berichten uit Amerika speelt deze computer ongeveer zo sterk als de Morphy Master. Met terugzetsmogelijkheid tot 400 ply (f 1400,- à f 1500,-). Challenger brengt ook een **Mini Sensory Challenger**, een reiscomputer voor f 260,-.

Alle genoemde computers zal ik voor u testen en daarover dan weer in dit blad berichten, in de eerstkomende nummers. De MK5 komt echter pas in november a.s., zodat ik daarover pas in februari kan publiceren.

Parijs 28-31 mei 1981

Een schaak-microcomputer-toernooi, gesponsord door l'Ordinateur International. Er waren 18 deelnemers in dit toernooi. Zeven rondes volgens Zwitsers systeem. Enkele van de hierboven beschreven computers speelden reeds mee. Voor de eerste t/m derde plaats is de uitslag wel betrouwbaar, maar door het Zwitsers systeem zijn de overige klasseringen minder betrouwbaar. Er werd gespeeld met een bedenktijd van drie minuten per zet. De Mephisto Y verloor ongelukkig van de Partner 2000. Hij stond een DAME voor, toen hij met 2 seconden zijn bedenktijd overschreed!

De Mephisto Y was een nieuw Mephisto-programma, meer gericht op positioneel spel. De Mephisto X-versie was meer op tactisch spel gericht.

Onbetwist kampioen werd de **Mephisto X**. Het nieuwe Mephisto-programma, dat binnenkort verkrijgbaar is, is een samenvoeging van de X en Y-versie, met een openingsboek met 300 varianten.

Het Grünfeld (openings-)moduul voor de Morphy Master is inmiddels verkrijgbaar. Het Capablanca-moduul (eindspel) voor de Morphy komt in oktober op de markt. De Mephisto X bleef in dit toernooi met 5 winstpartijen en 2 remises (tegen Morphy Master/Grünfeld en MK5) ongeslagen. Hier volgt de eindrangschikking:

Plaats	Deelnemer	Punten (uit 7)
1	Mephisto X	6
2-3	Morphy Master (met Grünfeld-moduul) en SciSys MK5	5
4-5	Morphy Encore en Mephisto Y	4½
6-8	Sargon ARB, Mephisto en Sfinks	4
9-12	Sargon 2.5, Challenger Sensory Voice (aangepast), Boris Diplomat (verbeterde versie) en Challenger Voice	3½
13	SciSys MK4	3
14-16	Partner 2000, Challenger 7 en Challenger Sensory Mini	2
17-18	Novag Micro Chess en Pocket Chess	1½

J. LOUWMAN

GAMBIET - Een schaakprogramma voor een Personal Computer (2)

Inleiding

De vorige keer bespraken we een algemene filosofie achter schaakprogramma's: generatie, evaluatie, en selectie van zetten. Iets dergelijks vindt plaats als mensen schaken, maar we hebben gezien, voor computers wordt het accent verschoven naar het vermogen om getallen te verwerken. Computers kunnen namelijk alleen maar dat en niets anders dan dat. Deze keer zullen we nogal benadrukken dat computerprogramma's uit algoritmen bestaan. Uitgangspunt daarvoor zal zijn Gambiet's meest uitgebreide programma-onderdeel: de evaluatiefunctie.

Een evaluatiefunctie kan variëren van zeer eenvoudig tot zeer gecompliceerd. Waarop de keuze valt moet in overeenstemming zijn met de methodiek voor de zettenselectie. We bespreken eerst de twee meest bekende methodieken: Shannon-A en Shannon-B.

Shannon

Shannon, wetenschapper en grondlegger van het moderne computerschaak, voorzag twee praktische methoden van aanpak:

A. Onderzoek alle zetten van de zettenboom.

(Tot een bepaalde diepte, eventueel daarna nog enkele zetten ter completering van bijvoorbeeld een afruilproces.)

B. Onderzoek alleen de plausibele zetten.

(Dat wil zeggen, onderken de slechte zetten en laat die weg.)

Shannon-B komt de schaker misschien als het meest voor de hand liggende voor, stelt echter de programmeur voor enorme problemen. Voor Shannon-A zou je een zeer eenvoudige evaluatiefunctie kunnen gebruiken. Bijvoorbeeld niet meer dan het tellen van de materiaalpunten. (Mat en pat worden dan herkend aan de generatie van nul zetten.) Een zo'n simpele evaluatie vraagt weinig computertijd, zeker bij knap programmeren of bij geavanceerde computersystemen. Daardoor kunnen er binnen een bepaald tijdsbestek zeer veel evaluaties gepleegd worden, en kan de boom misschien diep genoeg worden onderzocht om tot geducht spel te komen. Echter voor Shannon-B werkt een dergelijke evaluatie nimmer. Er is veel meer voor nodig dan een simpele materiaalrekening om te kunnen beslissen of een bepaalde zet nu wel of niet de moeite van verder onderzoek waard is. Shannon-A vraagt aan de programmeur voornamelijk programmeer-talent, meer dan de spelregels behoeft hij van schaken eigenlijk niet te weten. Voor Shannon-B is wat meer inzicht vereist.

A of B, of gedeeltelijk A, of iets heel anders. De discussie is nog niet gesloten, en de computerschaak-toernooien zullen het ons leren. Een feit is evenwel, dat tot op heden de meest succesvolle programma's: Belle, Chess, Sargon, van het type A zijn.

Eigenwijs

Zijn de meest succesvolle schaakprogramma's van het Shannon-A type, Gambiet/81 is zoiets als Shannon-B. Bij de beslissende keuze hebben de volgende overwegingen een rol gespeeld:

a. Alhoewel voornamelijk Shannon-A, is de evaluatiefunctie van de voorloper van Gambiet-/81, Gambiet/80 dus, toch al tamelijk uitgebreid. En wel om twee redenen. Ten eerste kan bij nauwkeuriger evalueren een meer grondige alfabetasnoeiing gerealiseerd worden door eerst de 'betere' zetten te bestuderen. Ten tweede beperkt beter evalueren het horizon-effect. Dat wil zeggen, dat de computer niet 'ziet' wat voorbij zijn ingestelde maximale ply-diepte, voorbij zijn horizon, geschiedt.

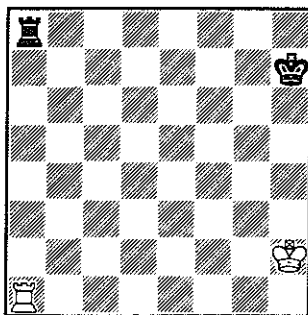
- Van Gambiet/80 naar /81 zien we ongeveer een verdubbeling van de evaluatietijd optreden.
- De mogelijkheden om een Shannon-A programma op eenzelfde type computer drastisch te verbeteren zijn beperkt. Naar eigen ervaring betekent één ply dieper, bij gemiddelde alphabeta-pruning, ongeveer vier keer zoveel denktijd. Anders gezegd, om één zet dieper te combineren moet het programma vier keer zo snel worden. Geen geringe opgave voor de programmeur. Maar al zou het lukken, slechts één ply dieper leidt geenszins tot spectaculair beter schaken.
 - Met Shannon-B, die een smallere en diepere boom onderzoekt, zijn veel leukere combinaties te vinden. Bovendien kan men iets op de diepte laten vallen en de vrijgekomen computertijd gebruiken om wat meer schaak-theorie in de evaluatiefunctie te stoppen.
 - Ik heb nachten liggen jammeren na letterlijk gezien te hebben hoe Gambiet/80 zijn kostbare tijd verknoeit met allerlei onzinnige varianten. Daar is een eind aan gekomen. Ik jammer nu ook gedurende de dag. (Dit is één van de betere in dit nummer. Red.)

Materiaal

Hoe dan ook, het leeuwendeel der evaluatie-punten zal moeten komen van de materiaal-situatie. Een zet, die tot materiaalwinst leidt, zal duidelijk goed gewaardeerd moeten worden. Eén manier is het optellen van al het materiaal op het bord: dame 9, toren 5, loper 3, paard 3, pion 1. Tenminste voor wit. Voor zwart dezelfde getallen, maar dan negatief. Is de uitkomst van zo'n optelling, +1, juist, dan staat wit een pionnetje voor. Is de uitkomst -5, ook juist, dan staat wit een toren achter. Met dit systeem doet de waarde van de koning er niets toe, want er zijn er altijd twee, samen nul dus. Natuurlijk kan het allemaal ook nog nauwkeuriger. Bijvoorbeeld de loper iets hoger dan het paard, of expliciet het loperpaar. Zo niet Gambiet, dat is maar een eenvoudig schaakprogramma.

Simpelweg tellen van het materiaal moet tot de onnauwkeurige evaluatiemethoden gerekend worden. We bekijken diagram 1. Na telling van het materiaal komen we uit op nul. Maar een beginneling ziet in, dat als wit de laatste zet heeft gedaan, dit toch wel een hopeloos slechte is geweest. Hij verliest namelijk onmiddellijk zijn toren, ziet u het ook? De materiaalbalans is misschien formeel wel nul, maar -5 voor een toren verlies brengt veel beter de slechte waardering voor de witte stelling tot uitdrukking.

Een computer kijkt niet tegen die twee torens aan zoals wij dat doen, geenszins. Een rekenautomaat kan alleen maar een beetje rekenen, een beetje vergelijken, een beetje onthouden. Voor ons schaakprogramma, niets anders dan een gecompliceerde goocheltruc, zullen we een nauwkeurig voorschrift moeten verzinnen, een algoritme, willen we een correctie op de materiaaltootaaltelling toepassen. Een algoritme, heel belangrijk, dat werkt voor het algemene geval, voor bijna alle stellingen op het schaakbord. Zo'n algoritme zit er wel eens naast. Maar goed ook. Gambiet heeft niet de ambitie schaken te degraderen tot een spelletje boter, kaas en eieren.



1

De materiaal-correctie

Het algoritme, waarmee Gambiet aangevallen en onvoldoende gedekt staande stukken opspoor, is ongeveer als volgt.

- voor elk stuk
- zoek alle aanvallers op dat stuk
- sorteer die aanvallers van laag naar hoog
- doe evenzo met de verdedigers
- maak een lijst van het afruilverloop:
 - 1 aanvaller 1 neemt stuk
resultaat = waarde stuk
 - 2 verdediger 1 neemt aanvaller 1
resultaat = waarde(stuk - aanvaller 1)
 - 3 enz. enz.

De lijst stopt eenvoudig als alle aanvallers of verdedigers verbruikt zijn.

- Bij nul of negatief resultaat staat het stuk voldoende gedekt. Een positief resultaat kan waarschijnlijk metterdaad aan materiaalpunten behaald worden.

Passen we dit algoritme toe op diagram 1.

Ta1 heeft 1 aanvaller, geen verdedigers. Winst 5 punten.

Ta8 heeft ook 1 aanvaller en geen verdedigers. Winst 5 punten.

Het algoritme is dus nog niet compleet. Corrigeren we voor zowel het verlies van de witte toren als van de zwarte, dan zijn we nog even ver, en dat is natuurlijk onzin. Als zwart aan zet is, dan gaat de witte toren verloren en niet de zwarte.

We vullen derhalve het algoritme aan:

- Corrigeer het totaal aan materiaalpunten met het verlies van de partij, die de laatste zet heeft gedaan. (Moet zijn tegenstander kiezen uit twee of meer winstmogelijkheden, neem dan de hoogste.)

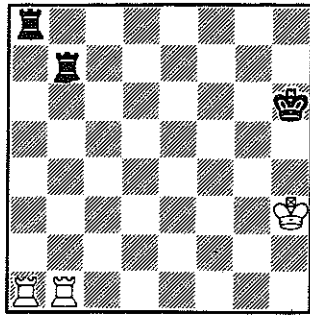
De laatste opdracht tussen de haakjes niet omdat computers zo bescheiden zijn, maar omdat ze niets weten.

Diplomat

Maar is het nu wel zo verstandig om de winst van de partij, die niet aan zet is, maar te vergeten? In het geval van diagram 1 wel, maar in dat van diagram 2? Volgens ons algoritme wint zwart aan zet een toren en wordt de materiaalbalans gelijk -5 punten. Het algoritme weet niet beter. Wij wel, wit wint direct weer een toren terug.

We hebben hier weer zo'n aardig dilemma, waar schaakprogrammeurs zo'n last van hebben. Soms is het wijs alleen voor de winst van de aan zet zijnde partij te corrigeren, soms kan de winst van de tegenpartij beter ook worden meegenomen. We hebben hier tevens een van de meest sprekende problemen bij het 'evalueren zonder vooruitkijken' te pakken.

Gambiet heeft de volgende diplomatieke oplossing gevonden: het evalueert twee waarden. Die met alleen het verlies van de niet aan zet zijnde partij, de minimum-evaluatie genaamd, en die met ook nog de winst van die partij, de maximum-evaluatie. Misschien een wat verwarrende definitie, maar bedenk dat Gambiet altijd zetten evalueert, nooit stellingen. Laten we er eens van uitgaan, dat er een witte zet wordt geëvalueerd, dus dat in diagram 1 of 2 zwart aan zet is. Vanuit wit gezien is dan het minimum een toren verlies (-5 punten) en het maximum ook nog een toren winst (0 punten).



2

Pennen

Het gegeven algoritme voor de materiaalcorrectie zou een Shannon-A programma misschien niet misstaan, maar is voor Shannon-B nog volstrekt onvoldoende. Uitgangspunt was namelijk dat alle aanvallers/verdedigers beschikbaar zijn voor een taak in een eventueel afruilproces. Maar wat denkt u van stukken, die gepend staan tegen de koning? Natuurlijk moet het programma die opsporen en niet mee laten doen met het algoritme. Andere wijzen van pennen zijn echter minder triviaal.

Bekijken we diagram 3. Stel we evalueren de witte zet Ta1xh1, die duidelijk tot torenwinst leidt. Maar het algoritme berekent:

5 punten winst vanwege Ta1

5 punten verlies vanwege Tf4

totaal nul = minimum evaluatie = maximum evaluatie

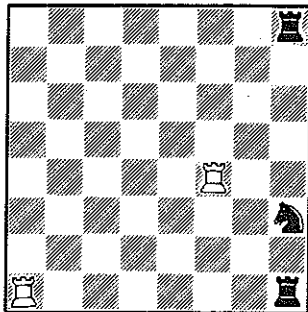
Voor een Shannon-A programma is dit geen ramp. Eén zet dieper in de boom wordt de torenwinst op h8 gezien. Maar voor een Shannon-B programma kunnen de gevolgen jammerlijk zijn. Met materiaalwinst nul en misschien andere positionele nadelen, kan Ta1xh1 wel eens niet tot de plausibele zetten gerekend worden. Dat wil zeggen, geheel vergeten, en dat is natuurlijk slecht schaken. Gambiet's oplossing voor dit probleem is dat de formatie h1-h3-h8 als een potentiële penning wordt herkend en dat derhalve de zet Ta1xh1 de torenwinst op h8 in de maximum-evaluatie meekrijgt.

Iets anders, waar Gambiet terdege rekening mee moet houden, is een situatie als in diagram 4. Het algoritme zal Te1xe2 op verlies evalueren:

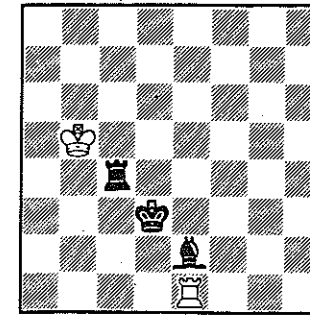
3 punten winst vanwege Le2

5 punten verlies vanwege Te2

De toren op c4 staat volgens het algoritme prima gedekt: één aanvaller en één verdediger. De oplossing in het Gambiet-programma is dat, omdat Kd3 een beslissende rol speelt in de afruil op e2, hij verder niet in aanmerking komt voor een verdedigende rol in de bepaling van de maximum-evaluatie.



3



4

Ruilen

Al met al is de routine voor het bepalen van de correctie op de materiaaltotaaltelling één van Gambiet's meest gecompliceerde programma-onderdelen geworden. Waar we het nog niet over gehad hebben, is dat een der partijen altijd nog een afruilproces zal afbreken, wanneer dat beter uitkomt. Om dit te detecteren laat Gambiet er een soort minimaxprocedure op los.

Iets anders waar we voorzichtig mee moeten zijn is bij het sorteren van aanvallers/verdedigers van laag naar hoog. Een dame schuin achter een loper, prima. Maar een loper achter een dame? Juist, kan tot vreselijke misinterpretaties leiden. Een oplossing is, dat zo'n loper tijdelijk een hogere waarde dan de dame krijgt foebedeeld.

Tot zover de materiaal-evaluatie. Hierna zullen we eens kijken wat er nog aan positie spel wordt gedaan.

Gulzig

Gambiet vindt belangrijk mobiliteit, centrumbeheersing en pionformatie. Maar ook zaken als ontwikkeling, rokeren, afruiltactiek en open lijnen krijgen aandacht. Echter, stelt u zich er allemaal niet te veel van voor. Voor elk criterium worden wat positiepuntjes gegeven. Weer positief, indien gunstig voor wit, anders negatief. Bijvoorbeeld een witte dubbelpion telt voor -2 punten, een witte korte rokade, onder gunstige voorwaarden, +2 punten.

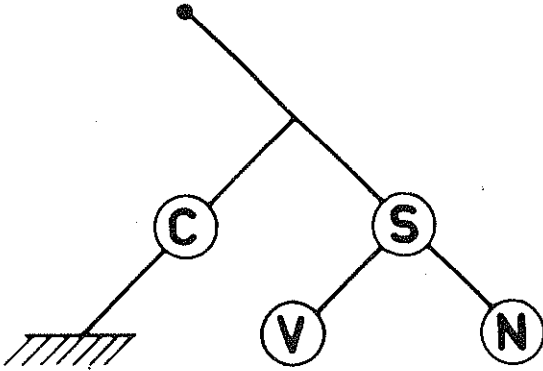
Het totaal aan positiepunten wordt gecombineerd met de materiaalpunten tot één waarde, waarbij het positie-aandeel nooit groter wordt dan een materiële pion-waarde. Met dit laatste begrijpt u dan meteen waarom computers zo materialistisch zijn, en, vooral op de lagere niveau's, zo'n ziekelijke belangstelling voor pionnetjes hebben. Gambiet niet te letterlijk nemen dus.

(Wordt vervolgd.)

W.J. RENS

Leden van de toernooicommissie voor het kampioenschap 1981 zijn:

Ir. J. Barentsz, P. van Diepen, J. Louwman, Drs. J.J. van Oosterwijk Bruyn, H. Schiet en Ir. B. Swets.



Computerschaak

2^e jaargang nr. 3

juni 1982

is een must voor iedereen die in computerschaak geïnteresseerd is. Het is te hopen dat er na het 2e toernooi weer zo'n mooi en informatief bulletin uitkomt van dezelfde kwaliteit (misschien wat sneller).

- 4) **The Chess Computer Book**. Door **T.D. Harding**, uitgegeven door **Pergamon Press** (Oxford-New York-Toronto-Sydney-Paris-Frankfurt) (1982) (215 blz., prijs: f 31,45).

Het is een uitermate populair boekje met veel gegevens over de huisschaakcomputers, zoals de Morphy, de Challengers, de Sargon etc. Voorts zijn een aantal teststellingen opgenomen met resultaten. De (vele) partijen zijn in de algebraïsche notatie (figurines) gegeven. In de tekst staan op veel plaatsen onderhoudende verhalen over de geschiedenis van het computerschaak en soms wordt er gerefereerd naar de schaakwereld. De schrijver doet een bescheiden poging iets van de werking van een programma uit te leggen; hij geeft ook enig inzicht in de computerschaakwereld. Het is jammer dat dit zo onsystematisch gebeurt. Het boek is geschikt voor de in computerschaak geïnteresseerde leek, testers zullen de inhoud met hun eigen resultaten kunnen vergelijken en programmeurs zullen er bekende (soms reeds overwonnen) problemen in tegen komen. Aanbevelen!

- 5) **Chess Computer News**. Een uitgave van **Competence, The Chess Computer Specialists**. Het is een tijdschrift, waarvan het eerste nummer op 27 maart 1982 verschenen is (30 blz., prijs £ 0,95). Adres: 263A Eversholt Street, London NW1 1BA, United Kingdom.

De uitgave bevat testrapporten, marktoverzichten, schaakcomputer-nieuws en commerciële nieuwtjes. Ook de resultaten van Louwman worden genoemd. Per jaar zullen er twee grote uitgaven verschijnen (april en november) en drie tussentijdse testrapporteringen (januari, juni en september). [Abonnementsprijs voor Europa: £ 7,75]. Voor testers en voor hen die op de hoogte willen blijven (consumentengids) is deze uitgave zeer nuttig.

- 6) **Thought and Choice in Chess, 1981**. Door **A.D. de Groot, Heymans Bulletin HB81-497-EX**. Een uitgave van de Rijksuniversiteit te Groningen (niet in de handel).

De Groot geeft een overzicht van de huidige stand van zaken over het schaakdenken gebaseerd op de denkbeelden van Otto Selz. In het rapport borduurt De Groot voort op zijn publikaties van 1946 (proefschrift) en van 1965 resp. 1978 (de vertaling en aanvulling van zijn onderzoekswerk: *Thought and Choice in Chess*). Computerschaak komt regelmatig aan de orde, maar er worden geen directe aanwijzingen gegeven met betrekking tot de programmering en/of de implementatie. Het rapport is opgenomen als hoofdstuk 6 in **'Otto Selz. His contribution to psychology'** (ed. Nico H. Frijda & Adriaan D. de Groot).

- 7) **Planmatig beslissen. Een onderzoek naar planning in het schaakspel**. Een afstudeerverslag van **Gerard van der Born** aan de interfaculteit bedrijfskunde te Delft (mei 1981).

Het onderzoek en het verslag ervan werd gecoached door David Brée. Ook hier vinden we geen aanwijzingen met betrekking tot de directe schaakprogrammering. De conclusie die uit het onderzoek getrokken kan worden is dat de planningsruimte gelijk is aan de spelruimte (in feite een verrassend resultaat, dat 'bewerkstelligd' wordt door de flexibiliteit in het denken van de schaakspeler). Het experiment tussen Joan Baart en Wim Heemskerk is zeer leuk beschreven.

- 8) **Structures for knowledge-based chess programs**. Een artikel van **K. v. Luck en B. Ownicki**. Onze Belgische lezer Albrecht Heeffter attendeerde me op dit artikel, dat gepubliceerd is in **GWAI-81** (German Workshop on Artificial Intelligence, Bad Honnef, January 1981), blz. 296-306).

Het systeem dat beschreven wordt is de neerslag van een aantal ideeën. Of het werkt weten de schrijvers zelf niet, om nog maar helemaal niet te spreken van de speelsterkte die een (middenspel)-programma, gebouwd volgens dit systeem, zal hebben. Een publikatie die uiter-

aard nog op vervolgonderzoek wacht; hopelijk zijn de schrijvers er zelf mee bezig.

- 9) **Basic en AI, AI-literatuur**. Twee artikelen van **Peter van Diepen** in de **HCC Nieuwsbrief 39**, blz. 35-39. Het gaat niet over schaken, maar geeft wel een aardig overzicht van de belangrijkste boeken op AI-gebied. Leuk om te lezen voor iemand die zich in het gebied van de kunstmatige intelligentie wil oriënteren.

- 10) **De achtergrond van een artificiële bridgespeler**. Het blad **Bridge** schenkt in een serie van 4 artikelen (voorlopig, want daarna komen er reacties) aandacht aan het fenomeen Computer-bridge. Het eerste artikel van de serie, die door mij verzorgd wordt, stond in het april-nummer (blz. 120-122). In juni, juli/aug. en sept. volgen de andere afleveringen: Het plan in een AI-bridgeprogramma, Speltechniek in een AI-bridgeprogramma, De Complexiteit van een artificiële bridgespeler. Er waren al reacties in de trant van: 'Moet de computer nu ook al leren briden?'

H.J. VAN DEN HERIK

Gambiet/82

Organisatie en structuur

In 1979 ging ik een schaakprogramma schrijven om programmeren te leren. Vandaag de dag kan ik al aardig programmeren, maar helaas nog steeds geen programma maken dat kan schaken. Aldus verluidd een van mijn gebruikelijke grapjes.

Een ieder die de lagere school heeft doorlopen kan binnen een paar dagen leren hoe hij een computerprogramma moet schrijven. Met voldoende vrije tijd ook wel een schaakprogramma, Gambiet/80 bijvoorbeeld. Zo'n programma, geschreven door een beunhaas, kan misschien best redelijk schaken, maar leent zich veelal slecht voor onderhoud. Dat laatste is vakjargon voor het niet of nauwelijks geschikt zijn voor verbeteringen, c.q. uitbreidingen. Slecht gestructureerd, zo wordt dat ook wel genoemd. Derhalve begonnen we voor Gambiet/81 maar helemaal opnieuw, en voor Gambiet/82, jawel.

Gambiet/82 heeft dus een beetje structuur en daarover gaan we vertellen. Hoe is het programma georganiseerd; welke modules zijn er met welke wisselwerkingen?

Gecompileerde computerprogramma's worden gewoonlijk opgedeeld in kleinere eenheden met elk een nauwkeurig gedefiniëerde opdracht. Zo'n moduul kan dan weer uit andere modules bestaan, of andere modules activeren. Elk moduul krijgt de een of andere suggestieve naam, zoals bijvoorbeeld GOGAMB.

Nu moet u het zich maar niet aantrekken als u dat in het geheel niet suggestief kunt vinden. U zult merken dat de namen steeds tot maximaal zes letters beperkt blijven, welke letters geplukt worden uit een of meer Engelse woorden. Bovendien verzijn ik ze zelf en dat verklaart veel.

We beginnen de bespreking bij moduul MATCH, die tot taak heeft het coördineren van een partijtje schaak. We duiken er midden in en zien dan dat MATCH in een keurige rij de volgende modules aanroept: ANALYS - MOVOUT - STATUS - PLAYER - MOVOUT - STATUS - SWEEP.

Dat 'aanroepen' hoeft u overigens niet letterlijk te nemen. Met 'MATCH roept ANALYS' bedoelen we dat het programma verder gaat met moduul ANALYS en, wanneer klaar, weer terugkomt bij MATCH.

Voelt u de organisatie al een beetje? MATCH coördineert de partij, ANALYS doet wat anders. En wel een zet voor Gambiet opzoeken. Daar hangt dus een hele tros met modules aan, maar dat zal MATCH een zorg wezen. Zijn zorg is het coördineren van de partij. Daartoe roept hij na ANALYS moduul MOVOUT, die de zet wereldkundig maakt, dan STATUS, die wat administratie bijhoudt, gevolgd door PLAYER, die een ander ook eens een zetje laat doen. Dan weer MOVOUT en STATUS, en dan nog SWEEP, die bepaalde huishoudelijke karweitjes opknaapt waarover straks meer.

En dan begint het feest weer bij ANALYS, want om beurten een zet doen is wel zo gezellig. Zolang de partij voortduurt, worden bovengenoemde modules dus in een kringetje doorlopen. Redelijk overzichtelijk, en daar houden ervaren programmeurs van. Onervaren programmeurs vinden dat niet intelligent genoeg, zo valt me telkens weer op.

MATCH doet nog wat meer dan alleen het op volgorde afroepen van een aantal routines. ANALYS maakt voor MATCH nog een statusbyte (een code) klaar, waarvan MATCH het een en ander kan aflezen. Bijvoorbeeld dat ANALYS tot zijn grote droefenis geen enkele zet kan vinden, hetgeen mat of pat voor Gambiet betekent. MATCH roept dan niet stomweg MOVOUT om een zet wereldkundig te maken, maar handelt zoals van een redelijke computer verwacht mag worden. Ook kan het zijn, dat ANALYS tevoren op eigen houtje MOVOUT reeds heeft geroepen. Gewoonlijk is dat ook inderdaad het geval, want Gambiet denkt in de tijd van de tegenstander, en dat 'denken' geschiedt dus door ANALYS. Heeft ANALYS een vooringesteld aantal stellingen geëvalueerd, zegt 12.000, dan is zijn denktijd verstreken en roept hij moduul MOVOUT om een zet af te geven. Daarna analyseert hij weer lustig verder. Totdat hij of zij aan het toetsenbord te kennen geeft een zetje te willen doen. Zulks interrumpeert het gemijmer van ANALYS, waarna deze alsnog als de weerlicht controle over het programma teruggeeft aan MATCH. En wel met het juiste bitje in de statusbyte, zodat MATCH weet, dat hij MOVOUT kan overslaan.

MOVOUT is verantwoordelijk voor het definitief uitvoeren van een zet. Een stuk wordt over het bord verplaatst met de nodige extra's in geval van en passant, rokade of promotie. Ook wordt de zet keurig netjes bijgeschreven in de zettenlijst op het scherm en wordt de grafische voorstelling van het schaakbord aangepast aan de nieuwe situatie. Verder wordt de schaakklok 'ingedrukt'. Kortom, MOVOUT doet alles wat de zet voor de buitenwereld zo definitief maakt.

Zoals we reeds zagen, kan MOVOUT vanuit drie verschillende plaatsen in het programma geroepen worden. Het enige wat MOVOUT moet weten is welke zet, en die vindt hij in een buffertje. De rest zoekt hij allemaal zelf uit.

STATUS, u begint het al te begrijpen, houdt de status bij. Onder andere de rokade-status. Is er een koning of een toren verplaatst, dan zet STATUS in een bepaalde statusbyte een bitje. Deze code zal later door de zetten-generatie-routines worden bekeken in verband met de legaliteit van de een of andere rokadezet.

PLAYER maakt eerst een lijst van alle toegestane zetten. Althans dat doet hij niet zelf, maar roept daartoe een moduul genaamd GEN. Als GEN geen zet kan vinden, mat of pat voor Gambiet's tegenstander dus, dan geeft PLAYER direct controle terug aan MATCH. Anders krijgt de speler gelegenheid een zet in te tikken. Komt deze zet niet in het lijstje van PLAYER voor, dan wordt de opponent aan het toetsenbord beleefd, maar niettemin dwingend, verzocht het nog eens over te doen.

Tijdens de zetten-selectie, het afzoeken van de varianten-boom, legt Gambiet een gedeelte van de boom in zijn geheugen vast. Laat ik dit verder aanduiden met 'het varianten-bestand'. Dat varianten-bestand, kennis over bepaalde voortzettingen dus, wordt meegenomen naar de volgende zet.

Bijvoorbeeld, heeft Gambiet gezien dat hij in vijf halve zetten een stuk kan winnen, dan is hij dat op de volgende zet niet vergeten en ziet dan direct de stukwinst in drie halve zetten. Dit is een aardigheidje van de programmeur, waarvan de mogelijkheden echter sterk beperkt blijven, omdat Gambiet's geheugen nou eenmaal erg klein is ten opzichte van die varianten-boom.

Na het uitvoeren van een zet kan de materiaalverhouding op het bord zich gewijzigd hebben, en kan de puntenwaardering in het variantenbestand moeten worden aangepast. Daarvoor hebben we moduul SWEEP. Dit moduul geeft tevens een flink stuk van het geheugen weer vrij voor succesievelijk gebruik, omdat na een definitieve zet een hele reeks varianten komt te vervallen.

Tot zover de moduul-bespreking. In een volgend verhaal zal ik ingaan op ANALYS en verdergaan met mijn betoog over modulair programmeren.

Gecomplieerde programma's kan je beter niet in een keer schrijven en vooral niet in een keer testen. Je schrijft in modules, die je afzonderlijk van alle kanten uitprobeert. Interacties met andere modules, die er nog niet zijn, die simuleer je zolang.

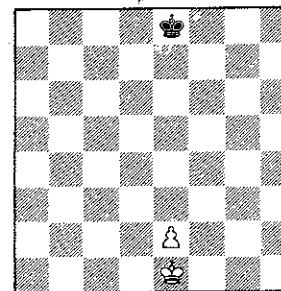
Altijd gestructureerd zijn omvangrijke programma's waaraan wordt gewerkt door meerdere programmeurs (U kunt dit navragen bij de heren van PION.)

Verder wordt 'evolutionair' programmeren mogelijk gemaakt, en dat acht ik voor schaakprogramma's erg belangrijk. Laat mij dit verduidelijken. Voor Gambiet/82 werd eerst een minimum aan modules ontworpen, geprogrammeerd en ten slotte op fouten gecorrigeerd. Die modules werden samengevoegd en Gambiet kon legaal schaken, zij het minimaal. Een voor een werden toen de verbeteringen en uitbreidingen ingevoegd en getest, zonder het voorgaande werk te veel aan te tasten.

Zo dachten de eerste versies nog niet in de tijd van de tegenstander. Maar het bijbouwen van deze eigenschap was betrekkelijk simpel. In ANALYS werden op de juiste plaatsen enkele controlerende routines ingeschoven. MOVOUT, het afgeven van de zet, kon zonder enige aanpassing gebruikt worden. MATCH moest worden uitgebreid met slechts een enkele test en wel op het overslaan van MOVOUT.

W.J. RENS

De oppositie-regel



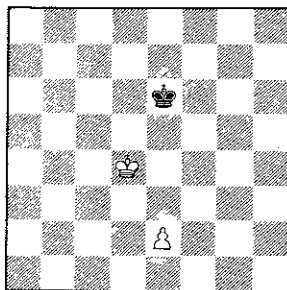
Wit speelt en wint

Met als eerste zet Ke1-d2 spelen de meeste nieuwe huis-schaakcomputers op hoog niveau dit eindspel goed. De pion kan promoveren en wit wint.

Echter in onderstaande, moeilijker stand gaan de meeste computers in de fout en spelen op zet 4. e2-e4? remise.

1. Ke1-d2 Ke8-d7
2. Kd2-e3 Kd7-d6
3. Ke3-d4 Kd6-e6
4. Kd4-e4! Zwart geeft op. Deze stand is voor wit gewonnen.

Gaan wij dus uit van het onderstaande diagram, dan is de sleutelzet 4. Kd4-e4!



Als enige huisschaakcomputer van de big Five speelde de Chess Challenger Champion dit correct! Niveau H in 40 minuten.

P.C. WITTEMAN



G. Vriend sprak op de jaarvergadering over schaakprogrammering, n. a. v. zijn programma GVNAA.

Gambiet afgeslacht

Onlangs speelde het nieuwe Gambietprogramma zijn debuut en dat was weer een waar feestgebeuren. De eer viel te beurt aan Peter van Grijfland, die een ruime ervaring met schaakprogramma's heeft en goed aanvoelt hoe je een computer moet afslachten. Tengevolge van twee klassieke problemen bij het computerschaak wordt Gambiet's spelpeil een paar honderd ELO-punten omlaag gebracht: horizonblindheid en gebrek aan offervaardigheid. Hieronder volgt de partij.

Wit: Peter van Grijfland

Zwart: Gambiet - versie 8212

1. e2-e4 e7-e5
2. f2-f4 e5xf4
3. Pg1-f3 Lf8-e7
4. Lf1-c4 Pg8-f6
5. e4-e5 Pf6-g4
6. Pb1-c3 Le7-h4+

5 ... Pg4 was de laatste zet uit het openingsboek. Ik ben geen echte schaker, maar heb het gevoel dat 6. ... Lh4+ wel moet kunnen.

7. Ke1-f1 Pg4-f2

Gambiet evalueert deze laatste zet op kwaliteitswinst en dat is natuurlijk niet goed.

8. Dd1-e1 Pf2xh1
9. De1xh4 Dd8xh4
10. Pf3xh4

Nogmaals, ik ben geen schaker, maar volgens mij staat zwart erg slecht, echter nog niet helemaal verloren.

Voor de computer is helaas een dodelijke stelling ontstaan. Hij kan geen afstand doen van dat paard, en schuift het verlies ervan voortdurend met allerlei dubieuze aanvalszetten achter zijn horizon.

10. Pb8-c6
11. Pc3-d5 Ke8-d8
12. Ph4-f3 Pc6-a5

Een heel duidelijk geval van horizonblindheid. Ik geef hieronder de varianten zoals Gambiet ze evalueerde. De eerste vier worden op twee pion-equivalenten verlies gezet, de laatste op een pion (na onderzoek tot en met ply 3 of 5).

De varianten zijn gegeven in Gambiet's aanvankelijke voorkeur.

- | | | | | |
|------|------|------|------|------|
| h8e8 | f7f6 | c6e7 | d7d6 | c6a5 |
| flg1 | flg1 | f3g5 | flg1 | c4e2 |
| -2 | -2 | h8f8 | -2 | c7c6 |
| | | g5h7 | | d5f4 |
| | | -2 | | -1 |

13. Pf3-g5 Th8-f8

14. Lc4-e2 f4-f3
15. Le2xf3 h7-h6
16. Pg5-e4 c7-c6
17. Pd5-c3 d7-d5
18. Pe4-d6 f7-f6
19. b2-b4 d5-d4

Tragisch. Je krijgt nu een situatie, waarbij Gambiet het niet kan schelen of hij het ene of het andere paard verliest. Hij weet niet dat na Pa5 het andere paard onherroepelijk volgt.

Weer de varianten:

- | | | | |
|------|------|------|------|
| a5c4 | f6e5 | d5d4 | d5d4 |
| d6c4 | flg1 | b4a5 | b4a5 |
| d5c4 | a5c4 | f6e5 | d4c3 |
| flg1 | d6c4 | d6c8 | glh1 |
| | | -2 | -2 |

20. Pc3-e4 b7-b6
21. b4xa5 f6-f5

Opwinding in het Gambiet-kamp. Hij forceert nu een afruil van Ph1! Briljante zet, echter helaas ongehinderd door enigerlei kennis van zaken. (Waar overigens de meeste schaakprogramma's hun faam aan te danken hebben.)

Dat blijkt wel op de volgende zet.

22. Pe4-f2 Lc8-a6+

De varianten:

- | | |
|------|------|
| h1f2 | c8a6 |
| f3c6 | f2d3 |
| c8a6 | a6d3 |
| flf2 | c2d3 |
| -1 | -1 |

Het afslachten kan nu zijn aanvang nemen.

23. Pf2-d3 Ta8-b8
24. Tal-b1 La6-b7
25. Lc1-a3 Lb7-a6
26. Lf3xc6 La6xd3+
27. c2xd3 Kd8-c7
28. Lc6-d5 Tf8-d8
29. Tbl-cl+ Kc7-d7

30. e5-e6+ Kd7-e7
 31. Tc1-c7+ Ke7-f6
 32. Tc7-f7+ Kf6-g5
 33. Tf7xg7+ Kg5-f6

34. Tg7-f7+ Kf6-g6
 35. a5-a6 Td8xd6
 36. La3xd6 Tb8-d8
 Zwart geeft op

Twee klassieke problemen: horizonblindheid en het onvermogen een stuk als verloren te beschouwen. Wie mij kan adviseren met een algemene oplossing, de nadruk op algemeen, wordt verzocht onmiddellijk contact met me op te nemen.

Maar ik heb een troost. Elk computerprogramma als tegenstander was vroeg of laat, maar eerder vroeg, de winst op h1 gaan halen. En dat had Gambiet uit zijn moeilijke situatie gered. Jan Louwman, wat denk je van de volgende stelling: Computers, die goed tegen andere computers spelen, behoeven nog niet goed tegen mensen te spelen.

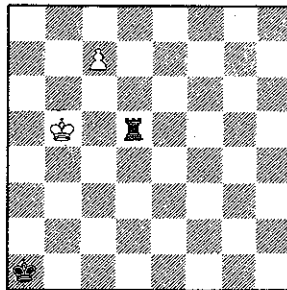
W.J. RENS

De stelling van Saavedra

Berekening of toeval?

Waarom speelt een schaakcomputer de stelling van Saavedra foutloos af? Dit is de vraag, die in het vorig nummer werd gesteld.

Het programma CHESS-0.5X hoefde slechts op een diepte van 5 halve zetten te worden ingesteld, om het probleem te kunnen oplossen. *Het probleem bleek zelfs te kunnen worden opgelost, wanneer het hele positionele deel van de evaluatiefunctie werd weggelaten.* Om te zien hoe de computer 'denkt' werd de derde zet van wit nader bekeken (zie diagram 1).



De stelling van Saavedra na 2... Td5+.

De computer speelde Kb4. Deze zet was ook wel te begrijpen. Toch had de computer op dezelfde gronden net zo goed Kb6 kunnen spelen. Waarom werd Kb4 gespeeld en niet Kb6?

We zetten de stelling van Saavedra weer op, *maar nu met verwisselde kleuren.* We plaatsen de witte koning en de witte toren op respectievelijk h8 en e4. De zwarte koning en de zwarte pion komen op respectievelijk g3 en f3 te staan. De opgave luidt dus nu: zwart speelt en wint. We stellen het programma weer in op een vaste diepte van 5 halve zetten en laten de computer weer zonder de positionele evaluatie spelen. De computer speelt nu:

1. ... f3-f2 2. Te4-e3+, Kg3-g4 3. Te3-e4+ (zie diagram 2)

Tot nu toe gaat het nog net zo als in de stelling zoals die voor wit was opgezet. De computer vervolgt: 3. ... Kg4-g3 4. Te4-e3+, Kg3-g4 5. Te3-e4+, Kg4-g3 enz.

De computer lost het probleem niet meer op! Er ontstaat remise door herhaling van zetten.

Hoe is dit nu mogelijk? Zou de computer ook zo spelen wanneer we de positionele evaluatie weer mee laten doen? We proberen het en de computer speelt de stelling van Saavedra weer zonder fouten af. Met de stelling voor wit opgezet kon de positionele evaluatie worden gemist, nu is deze blijkbaar hard nodig!

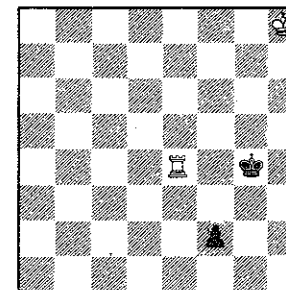
Twee vragen dringen zich op: waarom is de materiële evaluatie nu niet meer voldoende en welke factor in de positionele evaluatie zorgt ervoor dat het weer goed gaat?

Het antwoord op de eerste vraag hangt samen met de volgorde waarin het programma de verschillende zetten onderzoekt. Eerst wordt bekeken welke zetten kunnen worden gedaan. Dit gebeurt door het schaakbord rij voor rij, van links onder naar rechts boven langs te lopen. Wanneer een stuk is gevonden wordt bekeken naar welke velden dit stuk kan bewegen. Ook dit gebeurt in dezelfde volgorde, rij voor rij, van links onder naar rechts boven.

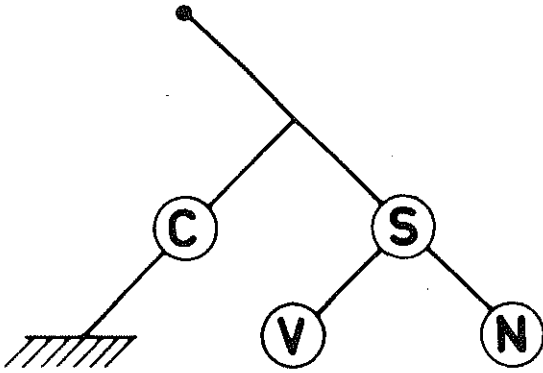
Deze procedure wordt eerst voor de zware stukken uitgevoerd en daarna op een iets andere wijze voor de pionnen. Tenslotte wordt nagegaan of rokadezetten mogelijk zijn.

Voor elk van de mogelijke zetten wordt nu de waarde bepaald van de stelling die ontstaat, wanneer die zet wordt uitgevoerd. Daarna worden de zetten zó gerangschikt, dat de zet die de stelling met de grootste waarde oplevert vooraan komt en de zet die de stelling met de kleinste waarde oplevert achteraan. Dit maakt de kans om snel de beste zet te vinden groter. Wanneer de positionele evaluatie is uitgeschakeld hebben alle zetten dezelfde waarde, zolang er niet een stuk van de tegenstander wordt geslagen en dat is hier bij geen van de zetten het geval.

Met de stelling voor wit opgezet betekent dit, dat de zetten worden bekeken in de volgorde Ka4, Kb4, Kc4, Ka6, Kb6 en Kc6. Daarom speelde de computer twee halve zetten diep Kb5-a4. Met de stelling voor zwart opgezet is de volgorde echter Kf3, Kg3, Kh3, Kf5, Kg5 en Kh5. Tijdens het rekenproces kan de volgorde nog iets worden gewijzigd. Elke keer wanneer een zet wordt gevonden, die beter is dan de tot op dat moment gevonden beste zet, dan wordt die zet uit de rij gelicht en vooraan geplaatst. Dit maakt de kans om snel de beste zet te vinden wanneer een halve zet dieper wordt 'geanalyseerd' groter. Tot vijf halve zetten diep is Kb6 niet als beste zet voorgekomen. Op dat moment staat Kb4 dus nog vóór Kb6 in de rij. Voor de computer zijn vijf halve zetten diep in de stelling voor wit Kb4 en Kb6 beide even goed. Aangezien Kb4 vóór Kb6 in de rij staat, zal eerst Kb4 als beste zet worden uitgekozen. De zet Kb6 wordt daarna wel bekeken, maar terzijde gelegd, omdat deze zet geen beter resultaat geeft. De computer besluit dus tot Kb4.



De stelling van Saavedra met verwisselde kleuren na 3. Te4+.



Computerschaak

2^e jaargang nr. 4

augustus 1982

Nawoord van de redactie

Overtuigd van het belang van dit artikel heeft de redactie besloten het in één keer, ongesplitst te publiceren, wat uiteraard consequenties had voor andere artikelen. Toen de auteurs dan ook 'op grond van nieuwe gegevens' het reeds persklaar gemaakte stuk voorlopig wilden terugnemen, kon de redactie dit niet toestaan. Maar we wilden dat ook welbewust niet, toen bleek wat die nieuwe gegevens waren. In een brief, eind juli, van Ken Thompson (van Belle!) bleek dat deze over een **niet gepubliceerde** database beschikte die leerde: '... its a 32-move game, not 33'. Eén van de verdiensten van het artikel van Dekker en Van den Herik is dat het falsificeerbaar geschreven is, d.w.z. dat andere auteurs duidelijk kunnen aangeven waar een eventuele fout zit. De redactie hoopt t.z.t. te mogen publiceren hoe de gedachtenwisseling met Thompson is verlopen. Doorgaan tot de fout gevonden is, stelt Thompson voor; dat is een gezonde wetenschappelijke opvatting!



Eind mei 1982 speelden twee Nederlandse programma's mee in het open kampioenschap van Utrecht, het Rabo-toernooi: Pion en PK'83. In het oktobernummer volgt een verslag.

Gambiet/82

Het Search-blok

Het schaakprogramma Gambiet/82 is te verdelen in twee hoofdblokken. Een voor de communicatie met de buitenwereld en een met de eigenlijke schaaklogica. Deze schaaklogica valt weer uiteen in drie subprogramma's, een met de zettengeneratie/evaluatie, een met positionele analyses, en een met het zetten-selectie-algoritme.

Over het laatste blok, SEARCH geheten, ga ik het hebben. In drie voorafgaande artikelen heb ik hier een beetje naar toe geschreven. Het eerste gaf toe dat mijn computers helemaal niet kunnen schaken en zich moeten behelpen met wat rekenen. Het vervolg legde uit dat Gambiet een uitgebreidere afruilevaluatie gebruikt. Zonder 'vooruitkijken' worden zowel de nabije materiaalwinst van wit als van zwart ingeschat. Als wit dan aan zet is, dan wordt de zwarte winst als verlies aangemerkt en verwerkt in de minimum-evaluatie. Vervolgens wordt er de witte winst bij opgeteld, hetgeen de maximum-evaluatie oplevert. Deze keer zal ik met een derde evaluatie op de proppen komen, die uiteindelijk bepalend is voor de zetkeuze.

Wanneer Gambiet ziet dat er een zet bedacht moet worden, dan geschiedt dat door een call naar ANALYS. Dat is het entry-moduul van een groep modules, samengevat onder de toepasselijke naam SEARCH. De meeste SEARCH-modules worden honderden keren per zet aangeroepen, echter ANALYS altijd slechts een keer.

Eerst laat ANALYS nagaan of Gambiet nog in een zettenreeks uit zijn openingsboek zit. Zo ja, dan is de search gauw bekeken, en komt het antwoord bijna ogenblikkelijk. Zoniet, dan wordt er een call gedaan naar de positionele routines. Dit programma-blok, dat evenals ANALYS eenmaal per zet wordt aangeroepen, heeft tot voornaamste taak het samenstellen van de tabellen met positionele straf/bonus-punten, welke gebruikt gaan worden door de evaluatie-routines.

Een voorbeeld. Pionnen krijgen punten voor het opmarcheren. Centrum-pionnen meer dan rand-pionnen: vrijpionnen meer dan niet-vrijpionnen. Vlak voor een gerokeerde koning kunnen ze strafpunten krijgen voor het opspelen. De analyse van deze simpele zaken vraagt vrij veel software. Bijvoorbeeld, om te weten of we op e5 met een vrijpion te maken hebben, moeten we gaan testen op de velden d6, e6, f6, d7, e7, f7, waar dan geen vijandelijke pion mag staan. En bij elk testje tikt de klok door. Maar hoe weet het programma, dat hij op d6 t/m f7 moet kijken? Ook dat vraagt een onderzoek met een bepaalde tijdsduur.

De kracht van een computer is zijn snelheid. De meest gecompliceerde analyse van een schaakstelling duurt zelden meer dan enkele milliseconden. Maar, als er per zet zo'n 10.000 stellingen onderzocht moeten worden, dan praten we ineens wel over enkele tientallen hele seconden, op een niveau van 180 seconden voor een zet. Liever beperken we ons bij het eigenlijke evalueren tot testjes, die zich laten meten in microseconden. De meer gecompliceerde analyses, die doen we dan maar eenmalig vooraf.

Natuurlijk schuilt hierin een zwakte. Gaan we dieper in de zettenboom, dan zal de informatie in de positionele tabellen verouderen. Jammer dan! Een schaakprogrammeur moet nu eenmaal kiezen. Om de korte combinaties redelijk goed te onderkennen, en dat is een eerste vereiste voor enigszins redelijk schaak, moeten er minimaal zo'n 15.000 stellingen per zet bekeken worden. Dit geldt voor Gambiet, maar bij andere programma's zal het om dezelfde orde van grootte gaan. Dat betekent op een zet van 150 seconden, 100 evaluaties per seconde, of 10 milliseconden voor een evaluatie, aannemende dat de evaluatie de bulk van de tijd voor zijn rekening neemt. In 10 milliseconden doe je op micro of minicomputer pakweg 1000 elementaire computerbewerkingen als het optellen van twee simpele getalletjes.

Zo, en nu hebben we u eens haarfijn voorgerekend waarom de schaakprogrammeurs zo bescheiden over hun programma's doen.

Maar we hadden het over ANALYS. Wel, na het positionele onderzoek kan de eerste zettenlijst gegeneerd worden. Dit geschiedt door het roepen van GEN. Deze routine genereert alle legale zetten, en roept voor elke zet de evaluatie routine EVAL. Wat ANALYS dus terugkrijgt is een keurige lijst compleet met minimum- en maximum-evaluaties.

Na afhandeling van wat bijzaken (we zullen alleen de hoofdlijnen blijven volgen) is het voor ANALYS bekeken, en geeft hij de controle over het programma door aan moduul DEPTH.

DEPTH wordt geroepen als SEARCH dieper wil gaan in de zettenboom. Op DEPTH drukken enkele verantwoordelijkheden, waarover straks meer, maar bij deze eerste aanroep is de opdracht simpel: zoek in de zojuist gegeneerde zettenlijst de beste zet, dat is de zet met de hoogste minimum-evaluatie.

En dan geeft DEPTH controle aan EXMOVE, welke die beste zet 'in gedachten' op het bord uitvoert, waarmede dan een nieuwe stelling is ontstaan. Derhalve kunnen nu de zetten voor de tweede ply gegeneerd worden. Daartoe geeft EXMOVE controle aan NODE welke een nieuwe call naar GEN voorbereidt en uitvoert, en daarna weer teruggaat naar DEPTH. Van DEPTH gaat het dan weer naar EXMOVE en van EXMOVE weer naar NODE.

Heeft u 'm nog, 't zal wel niet:

ANALYS - DEPTH - EXMOVE - NODE - DEPTH - EXMOVE - NODE
ANALYS en NODE riepen GEN; DEPTH zocht de beste zet, die werd uitgevoerd door EXMOVE. Nemen we even de beginstelling in gedachten, dan voerde EXMOVE de eerste keer misschien wel e2-e4 uit, en de tweede keer misschien wel e7-e5.

Ja, en nu is de verwachting misschien dat het van NODE weer naar DEPTH gaat, maar liever niet. In dit stadium gaat SEARCH niet dieper dan ply 3. De beste zet in de laatste zettenlijst behoeft ook niet uitgevoerd te worden. Wel belangrijk is diens minimum puntenwaardering. Was deze x punten, dan is dit tevens het beste resultaat van de uitgevoerde ply 2 zet (e7-e5 namen we als voorbeeld). Aan deze zet kennen we nu een 'eindevaluatie' toe van -x punten. Negatief, want wit en zwart hebben tegengestelde belangen. De eindevaluatie zal meestal liggen tussen de minimum- en de maximum-evaluatie, en, wat belangrijker is, de eindevaluatie heeft een grotere betrouwbaarheid, omdat alle mogelijke antwoorden nu in overweging zijn genomen.

Het is nu zaak ook voor de overige zetten op ply2 de eindevaluatie te bepalen. Daartoe moet eerst e7-e5 worden teruggezet, hetgeen wordt geregeld door moduul CLIMB, die verantwoordelijk is voor het terugklimmen in de boom. CLIMB geeft weer controle aan DEPTH, die nu misschien wel e7-e6 naar voren schuift. En zo krijgen we dus de lus:

DEPTH - EXMOVE - NODE - CLIMB.

Dat gaat door tot alle eindevaluaties op ply 2 bekend zijn, en de beste daarvan wordt weer tegengesteld toegekend aan de ply 1 zet (e2-e4). Merk op, dat deze eindevaluatie een betrouwbaarheid heeft van twee halve zetten diep.

Voor de overige ply 1 zetten halen we nu hetzelfde spelletje uit, met dat verschil, dat, waar mogelijk, het principe van de alfabetische snoeiing zal worden toegepast. Het controlerende moduul hiervoor is SLIMB, die dus het liefst weer zo gauw mogelijk boven is.

Wanneer alle zetten bekeken zijn dan komt moduul NEXPAS, die de nodige boekhouding verzorgt, opdat SEARCH hetzelfde spelletje kan herhalen, echter nu tot ply 5 in plaats van tot ply 3. Belangrijk is daarbij, dat nu, waar mogelijk, DEPTH de zetten selecteert op de eindevaluatie van de voorgaande search. Dit is daarom van belang omdat de eindevaluatie betrouwbaarder is, en de alfabetische techniek alleen maar effect heeft als eerst de goede zetten worden bekeken. Derhalve ook

duft SEARCH ply 5 pas aan, nadat hij wat meer zekerheid over de evaluaties heeft verkregen. En nu zou het prachtig zijn als er ook nog een search tot ply 7, 9, 11, en 13 zou volgen, maar ergens zitten er ook nog hohouwertjes, die de tijd in de gaten houden en Gambiet zijn zet laten afgeven.

Deze techniek van herhaald doorzoeken noemt men nou een iteratieve search. Een van de mankementen aan Gambiet/81 was, dat deze direct op ply 5 tekeer ging. Vaak gaat dat goed, maar vroeg of laat valt de search-efficiency totaal verkeerd uit, passeert het programma zijn bovenste tijdslimiet, en geeft een slechte zet af, zonder ooit aan de voor de hand liggende goede zet te zijn toegekomen.

Moduul DEPTH komt in actie wanneer in een knoop een keuze moet worden gemaakt uit een aantal mogelijke zetten. Er doen zich nu twee situaties voor. De eerste is dat DEPTH in actie komt via bijvoorbeeld NODE, wanneer er zojuist een kersverse zettenlijst is geproduceerd. De tweede situatie is dat DEPTH in actie komt via CLIMB, wanneer zojuist een zet uit de onderhavige lijst is teruggezet, en er dus tenminste 1 eindevaluatie voor de onderhavige knoop bekend is. DEPTH haalt dan een aardigheidje uit. Hij schrapt uit de zettenlijst alle zetten met een lagere maximumevaluatie dan de beste knoop-eindevaluatie.

Wat door DEPTH wordt toegepast is voorwaartse snoeiing. Dit in tegenstelling tot CLIMB, welke de terugwaartse snoeiing voor zijn rekening neemt. CLIMB maakt gebruik van de alfabetische techniek, DEPTH van een eigen gedachtenkronkel. De terugwaartse snoeiing van CLIMB is risicool, de voorwaartse snoeiing van DEPTH zit vol voetangels en klemmen. Wanneer de programmeur schuimbekkend rondtolt, dan denkt zijn omgeving: 'Ach, die voorwaartse snoeiing zeker weer'.

Is dit nu Shannon-B? Niet helemaal. Wat Shannon voorstelde was: onderken de plausibele zetten en bouw met een beperkt aantal daarvan de boom op. Wel voorwaartse snoeiing dus. Maar Gambiet werkt niet met plausibele zetten. Hij volgt meer een redenering van: ik heb al een goede zet, en deze zet levert toch niet meer op, dus waarom zou ik er nog aandacht aan besteden. Bij kommer en kwel zal Gambiet een volledige boom onderzoeken. Bij klare dame-winst hooguit nog een paar schaakjes. Gambiet lijkt soms wel een mens!

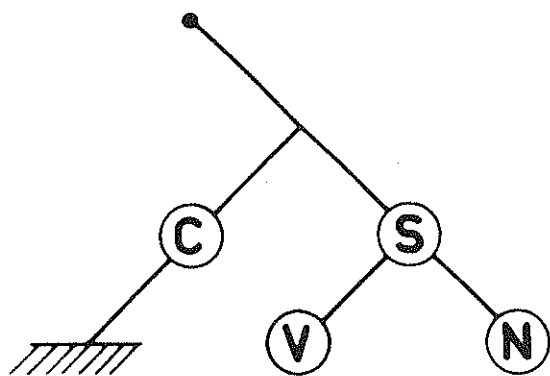
W.J. RENS

Elorating voor schaakcomputers

In de volgende aflevering van Computerschaak worden van ± 8 schaakcomputers de berekende Elo-getallen gepubliceerd.

Deze rating is vrij betrouwbaar, maar kost veel tijd (± 36 uur per schaakcomputer). De gekozen methodiek verloopt via een aantal teststellingen tot een totaal van 184 per computer! Na controle op dezelfde methodiek, toegepast bij schakers, bleek deze methode redelijk betrouwbaar.

J. LOUWMAN



Computerschaak

2^e jaargang nr. 5

oktober 1982

Gambiet 82, de nieuwe Nederlandse computerschaak-kampioen

Het Computerschaak kampioenschap 1982 is weer achter de rug. Drie dagen lang was de 'Leeuwenborch' van de Landbouwhogeschool in Wageningen het toneel van een spannende strijd. Tot het laatst bestond de mogelijkheid, dat drie programma's met een gelijk aantal punten bovenaan zouden eindigen. Uiteindelijk besliste Gambiet 82 van Wim Rens de strijd in zijn voordeel door YNCT 2.0 (Luuk de Vries) te verslaan en daarmee revanche te nemen voor de nederlaag van vorig jaar. Weliswaar verloor Gambiet 82 van de verrassend sterk spelende Rebel (E. Schröder), maar deze verloor op zijn beurt weer van YNCT 2.0. Op de tweede tot en met vierde plaats eindigden met een gelijk aantal punten, een vol punt achter de kampioen, YNCT 2.0, Rebel en een andere nieuwkomer, Chess 0.5X van W.H. Elsenaar. Hierbij lieten zij o.a. PION en Dappet, die reeds vorig jaar tot de voorste geleerden behoorden, achter zich.



Kampioen Wim Rens (met zwarte das) op het toernooi in Wageningen.

De volledige uitslag luidt als volgt:

Programma	Auteur	Computer	1	2	3	4	5	6	7	8	9	10	11	12	Pnt
1. Gambiet 82	W.J. Rens	Northstar/Xerox	x	1	0	1	1	1/2	1	-	-	1	1	1	7 1/2
2. YNCT 2.0	L.C. de Vries	Sharp MZ 80B	0	x	1	1	0	-	1/2	1	1	1	-	1	6 1/2
3. Rebel	E. Schröder	TRS 80	1	0	x	0	1	-	1/2	1	-	1	1	1	6 1/2
4. Chess 0.5X	W.H. Elsenaar	PDP 11/70	0	0	1	x	1	1/2	-	-	1	1	1	1	6 1/2
5. PION	Team TH Delft	PDP 11/70	0	1	0	0	x	1	1	1	1	-	1	-	6
6. Dappet	D. Hartman														
	P. Kouwenhoven	PDP 11/70	1/2	-	-	1/2	0	x	0	1	1	1	1	1	6
7. PK 83	A.R.D. v. Bergen	PDP 11/70	0	1/2	1/2	-	0	1	x	1/2	1	1/2	1	-	5
8. Storm 82	T. v.d. Storm	PDP 11/70	-	0	0	-	0	0	1/2	x	0	1/2	1	1	3
9. Usurpator	H.G. Muller	AIM 65	-	0	-	0	0	0	0	1	x	1	0	1	3
10. GVNAC	G. Vriend	DEC 10	0	0	0	0	-	0	1/2	1/2	0	x	-	1	2
11. Schaker	E.F.R. Knoop	OSI Challenger	0	-	0	0	0	0	0	0	1	-	x	1	2
12. Loper 82	L.H. Verberkt	Video Genie	0	0	0	0	-	0	-	0	0	0	0	x	0

In het team van de TH Delft werkten mee: S.T. Dekker, J. Derksen, H.J. v.d. Herik, J. Huisman, R. Hünen en H. Nefkens.

Bij een gelijk aantal punten werd de volgorde bepaald door het systeem Sonneborn-Berger. Een extra prijs werd toegekend aan H.G. Muller (Usurpator II), die een drievoudig programma had gerealiseerd in één micro-computer.

Hieronder volgt van elk der 4 hoogst geplaatsten een winstpartij zonder commentaar. In het volgende nummer komen wij meer uitvoerig terug op dit toernooi.

J.J. VAN OOSTERWIJK BRUYN

Wit: YNCT 2.0

Zwart: Gambiet 82

1. e4 e5, 2. Pf3 Pc6, 3. Lb5 a6, 4. La4 Pf6, 5. 0-0 Le7, 6. Te1 b5, 7. Lb3 d6, 8. c3 0-0, 9. h3 Pa5, 10. Lc2 Dd7, 11. d4 Pc6, 12. dxe5 dxe5, 13. De2 Td8, 14. a4 b4, 15. Le3 bxc3, 16. Pxc3 Lb4, 17. Tad1 De7, 18. Tf1 Le6, 19. Txd8+ Dxd8, 20. Td1 Dc8, 21. Pd5 Lxd5, 22. exd5 Pe7, 23. Lg5 Pexd5, 24. Pxe5 De6, 25. Lxf6 Pxf6, 26. Ld3 Te8, 27. f4 Db6+, 28. Kh1 a5, 29. Lb5 c6, 30. Lxc6 Dxc6, 31. Tal Pd7, 32. Dh5 Pxe5, 33. fxe5 De4, 34. Tc1 Dxe5, 35. Dxe5 Txe5, 36. Tc8+ Lf8, 37. Kh2 Te4, 38. b3 Te3, 39. Tb8 f6, 40. Kg1 Kf7, 41. Tb7+ Le7, 42. Kf2 Tc3, 43. h4 h5, 44. Tb5 Lc5+, 45. Ke2 Te3+, 46. Kd2 Te5, 47. Kd3 Lf2, 48. Txe5 fxe5, 49. Ke4 Kf6, 50. Kf3 Lxh4, 51. g4 hxg4+, 52. Kxg4 Le1, 53. Kf3 g5, 54. Ke2 Lb4, 55. 0-1.

Wit: Chess 0.5X

Zwart: Pion

1. e4 d6, 2. Pc3 Pf6, 3. Pf3 g6, 4. Lc4 Lg7, 5. 0-0 Pbd7, 6. d3 c6, 7. Lg5 b5, 8. Lb3 e5, 9. De2 Lb7, 10. a4 a6, 11. axb5 cxb5, 12. d4 Db6, 13. Dxe5 dxe5, 14. Pd5 Pxd5, 15. exd5 0-0, 16. Le7 Tfe8, 17. d6 Tac8, 18. Pg5 Txe7, 19. dxe7 Pf6, 20. Dxe5 Ta8, 21. De3 Dxe3, 22. Lxf7+ Kh8, 23. fxe3 Pe8, 24. Ld5 Lxd5, 25. Tf8+ Lxf8, 26. exf8D+ Lg8, 27. Pf7+ 1-0.

Wit: Gambiet 82

Zwart: Rebel

1. e4 e5, 2. Pf3 Pc6, 3. Lb5 a6, 4. La4 Pf6, 5. 0-0 Le7, 6. Te1 b5, 7. Lb3 d6, 8. c3 0-0, 9. h3 Pa5, 10. Lc2 c5, 11. d4 Dc7, 12. Pbd2 Ld7, 13. d5 Tfc8, 14. De2 Tab8, 15. b3 Db6, 16. De3 Ta8, 17. Lb2

Tc7, 18. Ld3 Ph5, 19. Lf1 Pf4, 20. c4 Tb7, 21. cxb5 Lxb5, 22. Tac1 Lxf1, 23. Kxf1 Db5+, 24. Pc4 h6, 25. Dc3 Pxc4, 26. Dxc4 Pd3, 27. Te2 Pxc1, 28. Dxc1 Dd3, 29. Pe1 Db5, 30. De3 Lg5, 31. Dc3 Tbb8, 32. Pd3 De8, 33. Tc2 f6, 34. a4 Dg6, 35. Dc4 Tb7, 36. b4 cxb4, 37. Pxb4 Dh5, 38. Te2 a5, 39. Pd3 Df7, 40. Dc6 Dd7, 41. Tc2 Tab8, 42. Dc4 Tb3, 43. Kgl g6, 44. Kh1 f5, 45. Kgl fxe4, 46. Dxe4 Df5, 47. Dxf5 gxf5, 48. Pxe5 Txb2, 49. Txb2 Txb2, 50. Pc6 Kf7, 51. g3 Kf6, 52. f4 Tb1+, 53. Kf2 Lxf4, 54. gxf4 Ta1, 55. Ke3 Txa4, 56. h4 h5, 57. Pd4 Ta1, 58. Pb5 Td1, 59. Pxd6 Txd5, 60. Pc4 a4, 61. Pb6 Ta5, 62. Pc4 Tc5, 63. Kd4 Tc8, 64. Kd3 0-1.

Wit: Chess 0.5X

Zwart:YNCT 2.0

1. e4 e5, 2. Pf3 Pc6, 3. Lb5 a6, 4. La4 Pf6, 5. 0-0 Le7, 6. Te1 b5, 7. Lb3 0-0, 8. c3 d6, 9. d3 Le6, 10. Pbd2 Lxb3, 11. Pxb3 a5, 12. a4 b4, 13. cxb4 Pxb4, 14. Le3 Pg4, 15. De2 Pxe3, 16. fxe3 f5, 17. exf5 Txf5, 18. Tf1 Dc8, 19. e4 Tf6, 20. Tac1 Kh8, 21. Pbd2 Dd7, 22. d4 exd4, 23. Pxd4 Txf1+, 24. Pxf1 Tf8, 25. Pe3 g6, 26. Pf3 c5, 27. b3 Da7, 28. Tf1 Tf6, 29. Pg4 c4+, 30. Kh1 cxb3, 31. Pxf6 Lxf6, 32. Pd2 Lg7, 33. Db5 Da8, 34. Pxb3 h5, 35. Dxa5 Dxe4, 36. Dd8+ Kh7, 37. Dxd6 Pd3, 38. a5 De2, 39. Kgl De3+, 40. Kh1 Pf2+, 41. Kgl Ph3+, 42. Kh1 Dg1+, 43. Txgl Pf2+, 44. 0-1.

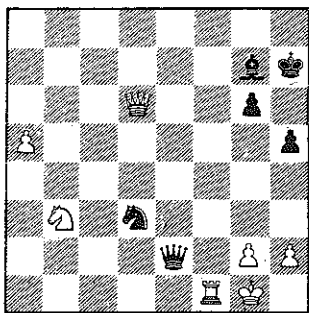
Het mat van Lucena

Toen ik in de eerste ronde van 'Wageningen' het spannende eindspel van Chess 0.5X tegen YNCT 2.0 bij de computers stond te bekijken, kon ik haast niet geloven, dat ons hier 'onbewust' een matcombinatie, met dame-offer en stikmat, op het bord werd voorgetoverd, die sinds 1497 al in de schaakklassieke voortleeft als 'het mat van Lucena'.

Interessant is ook dat beide programma's het mat vier zetten voor het einde al voorzagen, de verliezende partij nota bene het eerst!

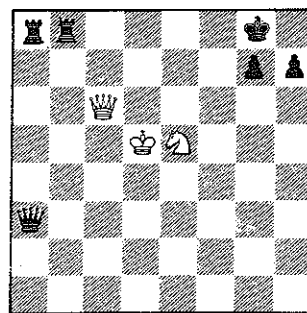
Tenslotte bleek ook dat deze klassieke combinatie van de grote Spaanse theoreticus door beide programma-ontwerpers niet als zodanig in hun programma was opgenomen.

1982



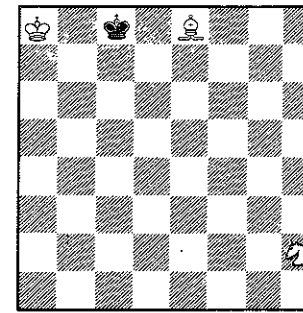
Chess 0,5X-YNCT 2.0

1497



Lucena

Wit aan zet



Mat in 33 (volgens ons)

Zwart geeft mat in 5 zetten.

39. ... De2-e3+, 40. Kgl-h1 Pd3-f2+, 41. Kh1-g1 Pf2-h3++, 42. Kgl-h1 De3-g1+, 43. Tf1xgl Ph3-f2 mat.

Wit geeft mat in 5 zetten.

1. Dc6-e6+ Kg8-h8, 2. Pe5-f7+ Kh8-g8, 3. Pf7-h6++ Kg8-h8, 4. Dc6-g8+ Tb8xg8, 5. Ph6-f7 mat.

J. TEN HAVE

Toch mat in 33

Inleiding

De publikatie 'Mat met looper en paard' (1) is gepaard gegaan met gezonde wetenschappelijke twijfels. Het gevoel van iets gevonden te hebben dat de uiteindelijke waarheid voor een bepaald eindspel betekent, gaf ons reden tot vreugde. Aan de andere kant deed het ons de voorzichtigheid niet uit het oog verliezen en namen wij contact op met Ken Thompson. Gezien onze resultaten achtten wij in juni/juli onze vondst echter al wel geschikt voor publikatie. Groot was onze teleurstelling toen Thompson schreef: '... its a 32-move game, not 33' (2). In hetzelfde schrijven bood Thompson aan: '... send me a 'best-play' example in 33 moves and I will send back where I think white can improve. We can iterate until the error is found.' (2)

Correspondentie

Tevens had Thompson ons een voorbeeld van zijn database gestuurd met een 'mat-in-32'-stelling en een daarbij behorend afspel. Controle van dit voorbeeld door onze database leverde het volgende op:

1. Onze database beschouwde de stelling als een 'mat-in-33'-stelling.
2. Onze database gaf aan dat de 12e zet van Zwart, zoals geproduceerd door de database van Thompson, niet optimaal was.

Dit resultaat zonden wij te zamen met een tweetal andere 'mat-in-33'-stellingen en bijbehorende afwikkeling terug aan Thompson ter verificatie.

Hieronder geven wij de analyse van onze database ten aanzien van de 'mat-in-32'-stelling van Thompson. Uitgangspunt is de variant van Thompsons database. Tussen haakjes vermelden wij de door onze database aangegeven optimale zetten-alsmede het commentaar daarop.