

Μελέτη και Υλοποίηση Μηχανής Σκακιού

Παρουσίαση Διπλωματικής Εργασίας

Κωνσταντίνος Δεσποινίδης 2021035

2026-06-23

Διεθνές Πανεπιστήμιο της Ελλάδος

1. Εισαγωγή

1. Εισαγωγή

1.1. Πρόβλημα

1.2. Στόχοι

2. Υλοποίηση

3. Πειράματα

4. Επίλογος

- Το σκάκι είναι ένα περίπλοκο παιχνίδι με αποτέλεσμα ο χώρος αναζήτησης να είναι τεράστιος
- Τα κλαδιά του δέντρου αυξάνονται εκθετικά (μέσος συντελεστής διακλάδωσης περίπου 30–35)
- Η εξαντλητική αναζήτηση είναι πρακτικά ανέφικτη, άρα απαιτούνται αποδοτικές ευρετικές και κλαδέματα
- Ο στόχος δεν είναι η ‘τέλεια’ επίλυση του παιχνιδιού, αλλά ισχυρές αποφάσεις εντός αυστηρού χρονικού ορίου

Σχεδιασμός και υλοποίηση μηχανής σκακιού σε **C**, συμβατή με το **UCI** πρωτόκολλο. Αυτό περιλαμβάνει:

- Άρτια και αποδοτική παραγωγή νόμιμων κινήσεων
- Αξιολόγηση της σκακιέρας μέσω ευρετικών μεθόδων
- Αναζήτηση της βέλτηστης κίνησης (minimax + τεχνικές κλαδέματος)
- Υλοποίηση του πρωτοκόλλου επικοινωνίας UCI

- Ρυθμός παραγωγής νόμιμων κινήσεων
- Μελέτες αφαίρεσης (ablation studies)
- Συντελεστής διακλάδωσης
- Εκτίμηση Elo

2. Υλοποίηση

1. Εισαγωγή

2. Υλοποίηση

2.1. Castro

2.2. Θεωρία Αλγορίθμων Αναζήτησης

2.3. Irida

3. Πειράματα

4. Επίλογος

Η **Castro** συνδυάζει δύο αναπαραστάσεις της ίδιας θέσης:

- **Bitboards:** αποδοτικοί έλεγχοι επιθέσεων και παραγωγής κινήσεων
- **Πίνακας 8×8 :** άμεση πρόσβαση στο περιεχόμενο κάθε τετραγώνου
- Σύνολα τετραγώνων ανά χρώμα και κατάσταση (white, black, empty bitboards)
- Η Irida προσπελάζει τη θέση **αποκλειστικά** μέσω του API της βιβλιοθήκης Castro

Zobrist hashing για κάθε θέση:

- Ανακόμιση hash ανά κίνηση/αναίρεση, χωρίς πλήρη επαναϋπολογισμό
- Κοινό κλειδί για TT, βιβλία ανοίγματος και αναγνώριση επαναλήψεων

Δημιουργία **pseudo-legal** κινήσεων και φιλτράρισμα σε νόμιμες:

- **Pseudo-legal:** Κινήσεις που ακολουθούν μόνο τη γεωμετρία των κομματιών, αγνοώντας αν ο βασιλιάς είναι σε шах.
- **Legal (Νόμιμες):** Το τελικό φιλτράρισμα των pseudo-legal κινήσεων με βάση τους κανόνες του σκακιού.
- **Έλεγχος νομιμότητας:** Απορρίπτονται κινήσεις που δεν επιλύουν τσεκ ή παραβιάζουν κάρφωμα (pins).
- **Perft:** Έλεγχος βάθους (move generation testing) για την επαλήθευση ότι το φιλτράρισμα είναι 100% ακριβές.

1. Negamax & Alpha-Beta Pruning

Πρόβλημα: Η minimax αναζήτηση εξερευνά ολόκληρο το δέντρο

Βασική ιδέα (Negamax): Ενιαία ρουτίνα για max/min: η επιστροφή αλλάζει πρόσημο στη γονική κλήση

$$V(s) = \max_{a \in A(s)} (-V(s_a))$$

Βασική ιδέα (Alpha-Beta): Παράθυρο (α, β) : απόρριψη κλάδων που δεν επηρεάζουν την απόφαση

- **Fail-low** ($\leq \alpha$): η γραμμή δεν βελτιώνει την τιμή
- **Fail-high** ($\geq \beta$): η γραμμή αρκετά καλή = διακοπή αδερφών

2. Iterative Deepening (ID)

Πρόβλημα: Άγνωστο βάθος υπό χρονικό περιορισμό — η σταθερή τιμή είναι υποβέλτιστη

Βασική ιδέα: Σταδιακή αύξηση βάθους, εκμετάλλευση TT πληροφορίας από προηγούμενη επανάληψη

Μηχανισμός:

- Αναζήτηση $d = 1, 2, 3, \dots$ έως λήξη χρόνου
- TT best move από $d - 1$ βελτιώνει διάταξη στο d

3. Quiescence Search

Πρόβλημα: *Horizon effect*: η στατική αξιολόγηση σε βάθος d διακόπτει τακτικές ακολουθίες πρόωρα

Βασική ιδέα: Συνέχιση αναζήτησης σε τακτικά ασταθείς θέσεις μέχρι σταθεροποίησης

Μηχανισμός:

- **Stand-pat:** στατική αξιολόγηση $\geq \beta$ οδηγεί σε cutoff (η θέση είναι ήδη αρκετά καλή)
- Διαφορετικά, επέκταση με **μόνο συλλήψεις** (όχι checks/ήσυχες κινήσεις)
- Μέγιστο βάθος (max ply) για αποφυγή ατέρμονων αλυσίδων

Η τακτική ακρίβεια υπερτερεί του υπολογιστικού κόστους

4. Transposition Tables (TT)

Πρόβλημα: Η ίδια θέση προσεγγίζεται από διαφορετικές ακολουθίες κάτι που οδηγεί σε επαναλαμβανόμενη αναζήτηση

Βασική ιδέα: Caching αποτελεσμάτων ανά Zobrist hash, επαναχρησιμοποίηση σε κλάδους και επαναλήψεις ID

Μηχανισμός:

- **Κλειδί:** 64-bit Zobrist hash
- **Καταχώρηση:** βάθος, τύπος ορίου (exact/lower/upper), best move, generation tag
- Cutoff μόνο όταν $\text{stored depth} \geq \text{remaining depth}$

Πολιτική αντικατάστασης: κενή > ίδιο κλειδί > βαθύτερο > stale generation

5. Move Ordering

Πρόβλημα: Η απόδοση alpha-beta εξαρτάται κρίσιμα από το πόσο νωρίς βρίσκεται η καλύτερη κίνηση

Βασική ιδέα: Πολλαπλά στρώματα ευρετικών: από ακριβή (TT) έως στατιστική (history)

Ιεραρχία ταξινόμησης:

1. **TT best move:** αποτέλεσμα προηγούμενης αναζήτησης
2. **MVV-LVA** (Most Valuable Victim / Least Valuable Attacker) για συλλήψεις
3. **Killer moves:** 2 ανά ply, ήσυχες cutoffs στο ίδιο βάθος
4. **History heuristic:** πίνακας (παίκτης, from, to) μάθηση από cutoffs

6. Null-Move Pruning (NMP)

Πρόβλημα: Η αναζήτηση όλων των κινήσεων είναι ακριβή. Μπορούμε να παραλείψουμε με ασφάλεια;

Βασική ιδέα: Στις περισσότερες θέσεις, η παράλειψη σειράς είναι χαμένη, ο αντίπαλος κερδίζει «δωρεάν»

Μηχανισμός:

- Μείωση βάθους κατά R , null-window search $(\beta - 1, \beta)$
- Αν $\geq \beta$ τότε cutoff (η θέση αντέχει)
- Προαιρετικό verification search

Προϋποθέσεις ασφαλείας: Μη root, επαρκές βάθος, όχι σε τσεκ, μη-μηδενικό υλικό

7. Late Move Reductions (LMR)

Πρόβλημα: Οι τελευταίες κινήσεις σε καλά ταξινομημένη λίστα είναι σπάνια βέλτιστες

Βασική ιδέα: Μείωση βάθους για απίθανες κινήσεις + re-search αν ξεπεράσουν το α

Μηχανισμός:

- Μόνο για ήσυχες κινήσεις (όχι captures/checks), όχι TT move, όχι PV node
- Μείωση $R(d, i)$ συναρτήσει βάθους και σειράς
- Αν reduced score $> \alpha$ τότε re-search σε πλήρες βάθος

8. Aspiration Windows

Πρόβλημα: Το αρχικό παράθυρο $(-\infty, +\infty)$ είναι υπερβολικά ευρύ, οδηγώντας σε λιγότερα cutoffs και αργή σύγκλιση.

Βασική ιδέα: Στένεμα του παραθύρου αναζήτησης γύρω από το αναμενόμενο σκορ (από προηγούμενη επανάληψη ID) για αύξηση των cutoffs.

Μηχανισμός:

- Ορισμός στενού παραθύρου $(\text{score}_{d-1} - W, \text{score}_{d-1} + W)$
- Αναζήτηση εντός παραθύρου: περισσότερα cutoffs λόγω στενότητας
- Αν fail-low $(\leq \alpha)$ ή fail-high $(\geq \beta)$, διεύρυνση παραθύρου και re-search
- Επανάληψη έως ότου το σκορ πέσει εντός παραθύρου

9. Principal Variation Search (PVS)

Πρόβλημα: Η πλήρης αναζήτηση κάθε κίνησης με το ίδιο παράθυρο (α, β) είναι υπολογιστικά ακριβή, όταν οι περισσότερες κινήσεις είναι χειρότερες από την πρώτη.

Βασική ιδέα: Αξιοποίηση της υπόθεσης ότι η πρώτη κίνηση (από TT / move ordering) είναι συνήθως βέλτιστη, οι υπόλοιπες ελέγχονται μόνο με zero-window scout.

Μηχανισμός:

- **Πρώτη κίνηση:** Πλήρης αναζήτηση με (α, β) για ακριβές σκορ
- **Υπόλοιπες κινήσεις:** Zero-window search $(\alpha, \alpha + 1)$: γρήγορη απόρριψη
- **Re-search:** Αν scout ξεπεράσει το α , επαναληπτική αναζήτηση με πλήρες παράθυρο για επιβεβαίωση

Το **UCI (Universal Chess Interface)** είναι το πρότυπο επικοινωνίας μεταξύ μηχανής και GUI:

- **Βασικές εντολές:** **uci**, **isready**, **position**, **go**, **stop**, **quit**
- **Ανεξάρτητο module:** module γενικής χρήσης, όχι φτιαγμένο συγκεκριμένα για την Ίριδα
- **Συνεργασία:** δοκιμές με Arena, Cute Chess, αυτοματοποιημένα scripts

Στατική βαθμολογία σε **centipawns** από την οπτική του παίκτη που κινεί. Βασικό **hand-crafted** μοντέλο (**PeSTO**):

- Χωριστά βάρη και **piece-square tables** για μέση και τελική φάση
- **Tapered evaluation**: η φάση παιχνιδιού (από το υλικό) ανάμεικνει τις δύο συνεισφορές
- Πρόσθετοι όροι HCE πάνω στον πυρήνα

Πρόσθετοι όροι που συμπληρώνουν το υλικό–PST πυρήνα:

- **Δομή πονιών:** αλυσίδες, περασμένα, απομονωμένα και αδύναμα πόνια
- **Κινητικότητα:** νόμιμες κινήσεις ανά κομμάτι (άμεση δραστηριότητα)
- **Ασφάλεια βασιλιά:** έκθεση, ασπίδες πιονιών, ανοιχτές στήλες
- **Δραστηριότητα / απειλές:** ενοχλητικά πόνια, χωρική πίεση στον αντίπαλο
- **Εξειδικευμένα endgames** και **δραστηριότητα πύργων** όταν το υλικό το δικαιολογεί

3. Πειράματα

1. Εισαγωγή

2. Υλοποίηση

3. Πειράματα

3.1. Perft

3.2. Ablations

3.3. Effective Branching Factor (EBF)

3.4. Εκτίμηση Elo

4. Επίλογος

Perft: μέτρηση όλων των **νόμιμων** κινήσεων σε σταθερό βάθος d .

- Επαλήθευση παραγωγής κινήσεων της **Castro** έναντι γνωστών αναφορών
- Ξεχωριστή μέτρηση από την πλήρη αναζήτηση (χωρίς αξιολόγηση / TT / κλαδέματα)

Βάθος	Κόμβοι
1	20
2	400
3	8,902
4	197,281
5	4,865,609
6	119,060,324

Απόδοση παραγωγής κινήσεων (ανά θέση, βάθος d):

Θέση	Βάθος	Κόμβοι	MN/s
Αρχική	4	197.281	2,33
Αρχική	5	4.865.609	2,68
Kiwipete	3	97.862	6,15
Kiwipete	4	4.085.603	5,47

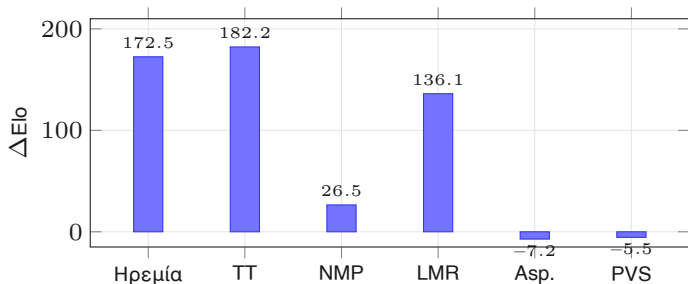
- Σταθερά **>2 MN/s**, έως **6,15 MN/s** στην απαιτητική θέση Kiwipete

Κάθε στάδιο προσθέτει **ένα** χαρακτηριστικό αναζήτησης στην αλυσίδα:

1. Alpha–Beta + iterative deepening
2. Quiescence search
3. Transposition table (TT)
4. Null-move pruning (NMP)
5. Late move reductions (LMR)
6. Aspiration windows
7. Principal variation search (PVS)

Head-to-head ανά στάδιο, αποτέλεσμα: ΔElo και SPRT (H_0 / H_1)

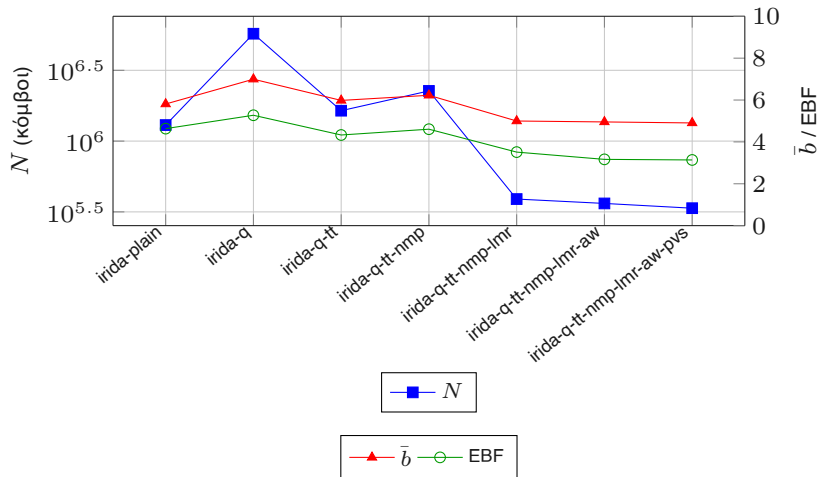
Πλεονέκτημα E_2 έναντι E_1 (λογιστικό ΔElo , 5.000 παιχνίδια ανά ζεύγος):



- **Μεγαλύτερα κέρδη:** ηρεμία (+172) και TT (+182), στη συνέχεια LMR (+136)
- **Aspiration / PVS:** αρνητικά ΔElo , φάνηκε ευαίσθησία στον περιορισμό χρόνου

Effective Branching Factor (EBF)

Κόμβοι N στο βάθος 8 (λογικός άξονας) και $\bar{b} / \text{EBF}_{\text{geom}}$ (γραμμικός δεξιός άξονας), αρχική θέση.



Η Irida δοκιμάστηκε έναντι του **Stockfish 18** περιορισμένο στα **2000 Elo**.

- **Αγώνες:** 2.000 ζευγαρωτοί αγώνες
- **Εκτιμώμενη διαφορά:** +110,94 Elo
- **Διάστημα εμπιστοσύνης 95%:** [+79,48, +142,40]
- **Εκτιμώμενο rating:** περίπου 2110 Elo

4. Επίλογος

1. Εισαγωγή
2. Υλοποίηση
3. Πειράματα
4. Επίλογος
 - 4.1. Κύρια Ευρήματα
 - 4.2. Συζήτηση και επόμενα βήματα

Μέγιστη Απόδοση (σε μικρό χρόνο σκέψης)

- **Κορυφαίες τεχνικές:** Quiescence (Αναζήτηση ηρεμίας), Transposition Table (Πίνακας μεταθέσεων) και LMR (Μειώσεις όψιμων κινήσεων).
- **Aspiration / PVS:** Αναποτελεσματικά σε πολύ μικρό χρόνο (π.χ. 0.1 δευτερόλεπτα) λόγω του κόστους των επαναλήψεων (re-search).

Συμπέρασμα

Η ισχύς του αλγορίθμου πηγάζει από την **αλληλεπίδραση** των τεχνικών και τη σωστή διαχείριση χρόνου, όχι από μεμονωμένες μεθόδους.

- **Προφίλ ανά time control** (aspiration, LMR, NMP) με δυναμική επιλογή
- **Βελτίωση HCE** και πιο ισχυρό NNUE (πιθανόν **incremental accumulator**)
- **Βελτίωση move ordering** για πλήρη αξιοποίηση των aspiration windows και PVS
- **Παράλληλη αναζήτηση** (π.χ. Lazy SMP), διαφορετική δυναμική ΤΤ/ταξινόμησης

- Επιπλέον **time controls**, σουίτες ανοίγματος και αντιπάλους
- **Αυτόματο tuning** (black-box + SPRT gating) για αποδοχή/απόρριψη ρυθμίσεων
- **Αναπαραγώγιμα manifests** και scripts για επανάληψη πειραμάτων

Ευχαριστώ για την προσοχή σας

Κωνσταντίνος Δεσποινίδης <**iee2021035@ihu.gr**>

- **github.com/KDesp73/irida**
- **github.com/KDesp73/castro**