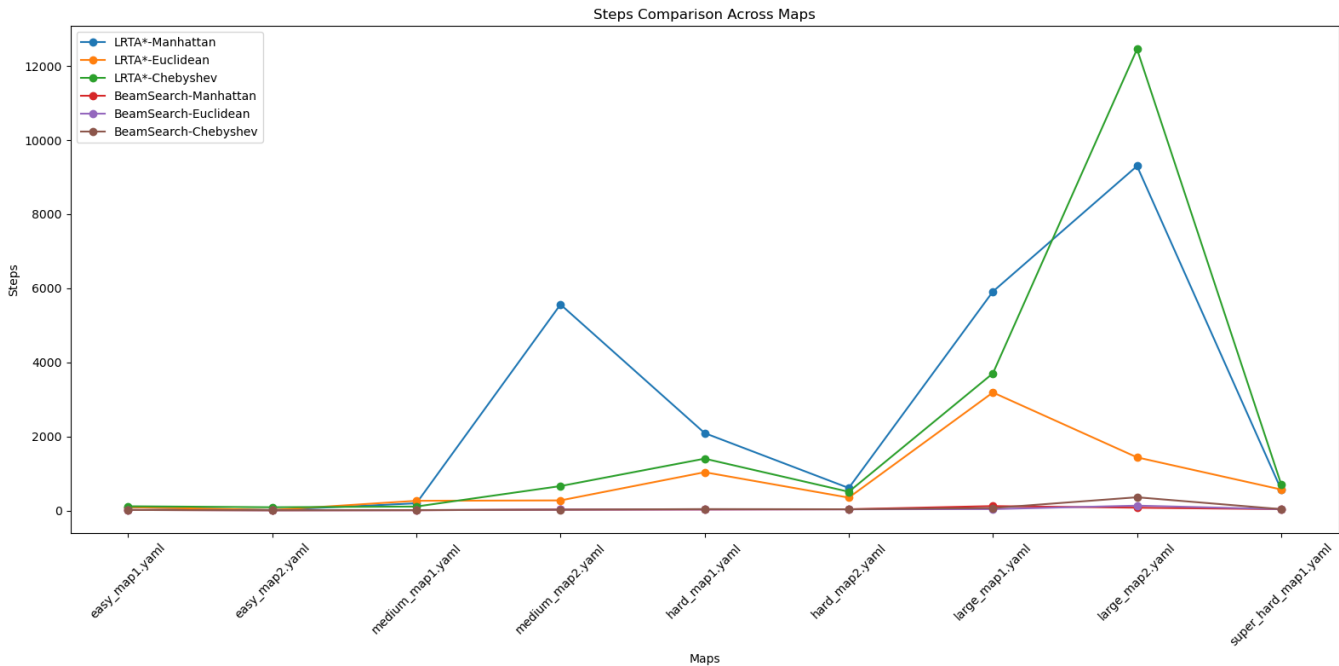
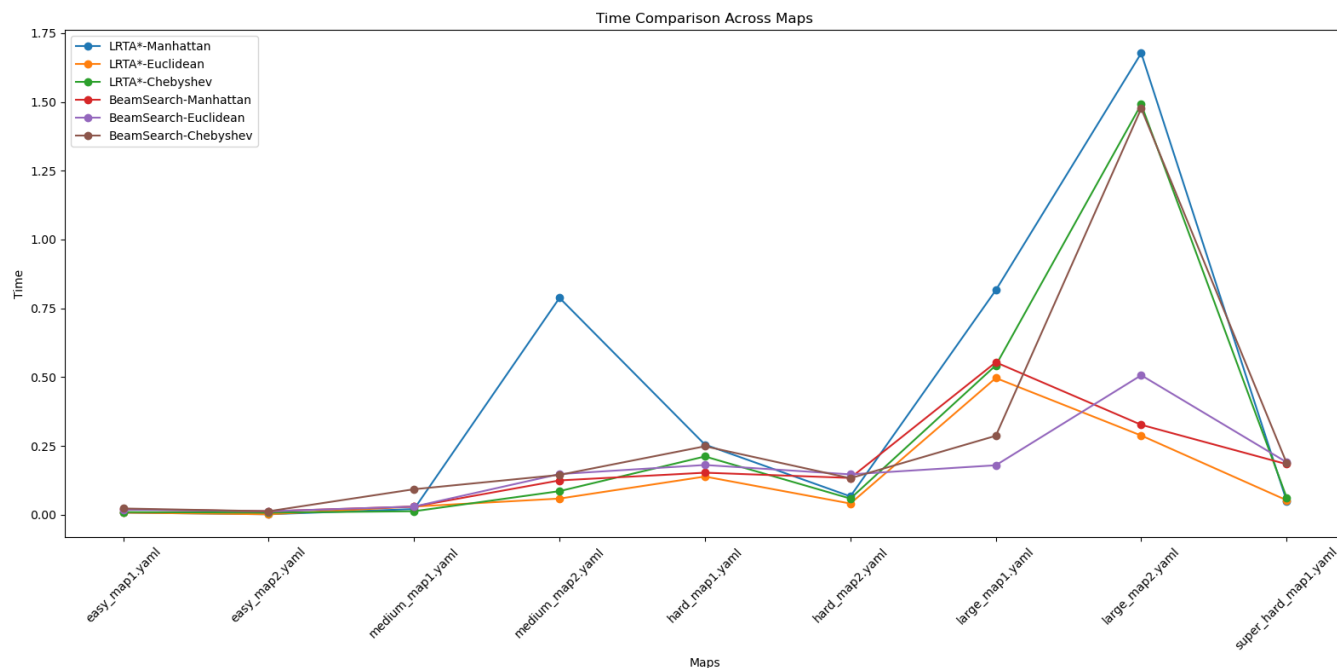


Sokoban Search Benchmark



În privin•a euristiciilor folosite, putem observa c•a euristica Manhattan func•ioneaz• în general foarte bine în Sokoban, deoarece spa•iul de mi•care este pe o gril• și obiectele trebuie mutate doar pe direc•ii ortogonale (sus, jos, stînga, dreapta). Manhattan distance m•soar• exact acest lucru — suma distan•elor pe axele X și Y — f•r• a permite deplas•ri pe diagonal•, ceea ce se potrive•te perfect regulilor jocului. De aceea, algoritmi care folosesc euristica Manhattan tind s• g•seasc• solu•ii mai scurte și mai eficiente. Euristica Euclidian•, de•i în teorie mai "precis•" pentru distan•e fizice, nu este foarte potriv•it• aici pentru c• Sokoban nu permite mi•c•ri diagonale. Astfel, folosind distan•a euclidian•, agentul poate avea o estimare nerealist• a c•t de "aproape" este o cutie de destina•ie, ceea ce poate duce la explor•ri inutile. Chebyshev distance poate fi rezonabil• doar dac• jocul ar permite mi•c•ri diagonale (ca în •ah pentru rege), ceea ce nu este cazul aici, deci și aceast• euristic• poate introduce erori de ghidare în c•utare, fiind în general mai pu•in eficient• pentru Sokoban dec•t Manhattan. Referitor la algoritmi, LRTA* și Beam Search sunt destul de diferi•i ca filozofie de func•ionare. LRTA* este un algoritm de c•utare reactiv• care înva•• pe m•sur• ce exploreaz•, adapt•ndu-și func•ia de cost estimat în timp real. Avantajul major al LRTA* este c• poate func•iona f•r• a avea o memorie imens•: nu are nevoie s• construiasc• tot spa•iul de c•utare dinainte, ceea ce îl face potrivit pentru h•r•i foarte mari sau c•nd nu ai suficient• memorie. Dezavantajul este îns• c• poate ajunge s• "se plimbe" foarte mult, mai ales dac• euristica este slab• sau neinformat•, ceea ce duce la un num•r mare de pa•i (cum se vede în unele rezultate de peste 9000 sau chiar 12000 pa•i). Beam Search, în schimb, este un algoritm orientat pe vitez• și pragmatism: la fiecare pas, men•ine doar un num•r limitat de st•ri cele mai promi•toare, t•ind agresiv ramurile mai slabe. Asta înseamn• c• este rapid și economise•te memorie, dar exist• riscul s• piard• solu•ia optim• sau chiar s• nu g•seasc• nicio solu•ie dac• beam-ul (num•rul de st•ri p•strate) este prea îngust. Totu•i, pentru h•r•i mai simple sau de complexitate medie, Beam Search cu o euristic• bun• cum este Manhattan poate performa excep•ional de bine și rapid, oferind solu•ii în c•teva sute de pa•i. Pe tipuri de h•r•i, LRTA* se potrive•te mai bine acolo unde spa•iul este foarte mare, labirintic, și agentul trebuie s• "descopere" drumul în timp real f•r• a epuiza memoria. De exemplu, pe h•r•i de tip "large_map" sau "super_hard_map", unde spa•iul de stare este imens și greu de preprocesat, LRTA* este util, chiar dac• face multe mi•c•ri. Beam Search este mai potrivit pentru h•r•i compacte, unde solu•ia nu este foarte ad•nc• și exist• multe c•i posibile, deoarece acolo poate rapid selecta cea mai bun• ramur• și ajunge la solu•ie f•r• s• piard• timp în explor•ri inutile. De asemenea, Beam Search, dac• este configurat bine (beam width suficient de mare), poate rezolva eficient și unele h•r•i medii sau chiar mai complexe, dar risc• s• nu func•ioneze bine în h•r•i extrem de complicate, unde este nevoie de explorare mult mai ampl• pentru a g•si solu•ia. Per ansamblu, în Sokoban, Manhattan distance este cea mai eficient• euristic•, LRTA* este mai sigur dar mai lent, iar Beam Search este mai rapid dar mai riscant. Alegerea trebuie f•cut• în func•ie de m•rimea și dificultatea h•r•ii și de resursele disponibile (memorie și timp).

Time Comparison:



În primul rând, Beam Search este în mod consistent mult mai rapid decât LRTA* pentru toate tipurile de hărți, indiferent de euristica folosită. Chiar dacă LRTA* reușește să rezolve problemele, timpul de execuție este semnificativ mai mare, mai ales pe hărțile mai complexe, cum sunt `large_map1.yaml`, `large_map2.yaml` și `super_hard_map1.yaml`, unde diferența devine foarte evidentă. Beam Search menține timpi de rezolvare destul de mici și relativ constanți chiar și pe aceste hărți dificile, ceea ce arată că este mult mai eficient la nivel de viteză. În al doilea rând, LRTA* suferă de o creștere mare a timpului de execuție pe măsură ce hărțile devin mai mari și mai complicate. Motivul este specific modului în care funcționează LRTA*: el explorează în mod reactiv și învață pe parcurs, ceea ce înseamnă că poate face foarte multe mișcări "inutile" înainte să găsească soluția. În contrast, Beam Search taie rapid opțiunile slabe și păstrează doar cele mai promițătoare stări, ceea ce îi permite să ajungă mai repede la soluție. Mai trebuie spus că alegerea euristicii influențează puțin timpul, dar diferențele majore între timpi sunt datorate în primul rând algoritmului, nu euristicii. De exemplu, LRTA* cu Manhattan, Euclidean sau Chebyshev are mereu timpi mai mari decât Beam Search cu oricare dintre aceste euristici.