Przeanalizowane prace naukowe:

* <https://www.researchgate.net/profile/Xiao-Cui-12/publication/267809499_A-based_Pathfinding_in_Modern_Computer_Games/links/54fd73740cf270426d125adc/A-based-Pathfinding-in-Modern-Computer-Games.pdf> - Optymalizacje A\* w grach, HPA\*, search space
* <https://ojs.aaai.org/index.php/AIIDE/article/view/12397/12256> - HPA\*, SHPA\*, DHPA\*
* <https://link.springer.com/chapter/10.1007/978-3-642-31900-6_32> - HPA + decision tree
* [https://en.wikipedia.org/wiki/Lifelong\_Planning\_A\*](https://en.wikipedia.org/wiki/Lifelong_Planning_A*), <https://www.worldscientific.com/doi/abs/10.1142/S0218194020400215> - LPA\*
* <http://journals.rta.lv/index.php/ETR/article/view/868> - analiza wydajności HPA\*, A\*, LPA\*
* <https://dl.acm.org/doi/abs/10.1145/3186728.3164141?casa_token=fU5zgdprubkAAAAA:_tLSCFhWgQtWOXcsHc07rv0WRH2oPdsmL4YbJwDJahnaZJsRsvR8zcoOpYAwl7tQsUPgie1Pvw>, <https://link.springer.com/chapter/10.1007/978-3-319-07959-2_22> - Hub labelling (google maps)
* <https://link.springer.com/chapter/10.1007/978-3-642-39206-1_7>, <https://dl.acm.org/doi/abs/10.1145/3186728.3164141?casa_token=A2QTjfqWLaEAAAAA:EjoxXA20_6X550cY1ivOaNNR3igmzKp_FMr2Tj0_1aaGS_qpDO6_AKLXTdPhw3SMKCY1Bsu4Cw>  - Hub labelling optimizations
* <https://link.springer.com/chapter/10.1007/978-3-540-68552-4_24>, <https://pubsonline.informs.org/doi/abs/10.1287/trsc.1110.0401?casa_token=jmVa0vC4tSoAAAAA:zTDAF7a79JOlMNKh4eE6gWpSp-EAcf9UXMFuRcS_I6BvdYOsMRlppMjp78RYKcVRW_O_Fqej>, <https://dl.acm.org/doi/abs/10.1145/2886843?casa_token=v5ZimgUPWOEAAAAA:JcPoSiKeNLl1eXXaf4LKpO1q7fhZujszGvWoZZ9cBcoWgJ5go9oO0cDS322dNJRGeLtq6ZcyMQ>, <https://epubs.siam.org/doi/abs/10.1137/1.9781611972894.10>, <https://www.sciencedirect.com/science/article/pii/S0304397516303176>, <https://dl.acm.org/doi/abs/10.1145/3186728.3164141?casa_token=A2QTjfqWLaEAAAAA:EjoxXA20_6X550cY1ivOaNNR3igmzKp_FMr2Tj0_1aaGS_qpDO6_AKLXTdPhw3SMKCY1Bsu4Cw>, <https://en.wikipedia.org/wiki/Contraction_hierarchies> - Contraction hierarchies (takie HPA\* tylko bardziej ogólne)
* <https://en.wikipedia.org/wiki/Transit_node_routing>, <https://link.springer.com/chapter/10.1007/978-3-642-38527-8_7>, <https://link.springer.com/article/10.1007/s10732-008-9089-8>, <https://link.springer.com/chapter/10.1007/978-3-540-72845-0_6>, <https://www.science.org/doi/abs/10.1126/science.1137521> - Transit node routing (niby wolniejsze od Hub labelling ale szybszy preprocessing)

Wnioski:

Temat szybkiego znajdywania ścieżek w grafie jest tematem zakrawającym o pare dziedzin. Jedną z nich są ogólnie rzecz biorąc gry komputerowe, potrzebują one algorytmu który będzie w stanie szybko znaleźć trasę pomiędzy dwoma punktami w grafie np. w celu zaprojektowania dynamicznego AI przeciwników. Algorytm HPA\* i jego odmiany służą właśnie rozwiązaniu tego problemu, nastawione są one bardziej pod optymalizację pojedynczego zapytania. W dużym uproszczeniu algorytm polega na zbudowaniu hierarchicznego grafu na podstawie już istniejącego i wykonywanie zwykłego A\* w stylu „top-down”. Wg. prac jest on wystarczająco dokładny aby gracz nie zauważył, że przeciwnicy obierają bardzo dziwne trasy. Algorytm wymaga preprocessing aby był funkcjonalny (stąd podejrzewany narzut pamięci). Drugą dziedziną jest dziedzina bardziej związana z tematem pracy – czyli znajdywanie tras drogowych. Tutaj mamy 2 algorytmy: Transit Node Routing oraz Hub Labelling + Contraction Hierarchies. Co do transit node routing - wymaga on dość sporego preprocessingu pod katem wstępnego obliczenia części odległości pomiędzy punktami, z drugiej strony jest on ciekawy pod kątem dróg ze względu, że bierze pod uwagę „hierarchię” dróg tzn. będzie starał się chętniej wybrać drogę „szerszą” niż prowadzić nas po faktycznie szybszej trasie (co w prawdzie i tak okaże się w 90% przypadków wolniejszą trasą). Hub Labelling + Contraction Hierarchies jest szybszym algorytmem od Transit Node Routing, polega na zhierarchizowaniu danych drogowych do postaci „labelki” dzięki której algorytm potrzebuje jedynie 2 zapytań (dostępów do labelek) w celu znalezienia trasy pomiędzy dwoma node’ami. Z minusów jest on bardzo pamięciożerny.