

Automatisk ruteplanlegging for fritidsbåter i norsk skjærgård: problemkarakterisering, målinger og forskningsagenda

Ola Jacob Mjølven Iversen

Fellag AS (BoatNav), Kongsberg, Norge · olajacob@fellag.as

BoatNav Research Note 2026-01 · versjon 1.2, 26. juli 2026 · preprint — ikke fagfelleurdert

Abstract. Automatic route suggestion for recreational craft in the Norwegian archipelago differs from road-network routing, robot motion planning, and ocean weather routing in ways that break standard assumptions: no graph is given — the navigable space must be discretized from chart polygons and depth data at query time; the graph is vessel-specific (draft, beam, air draft); the output is advisory, so it must be human-editable (few vertices) and justifiable per leg; errors are asymmetric — a false “clear” can ground a vessel, so discretization must be conservative; and the computation must run interactively, offline, on a single JavaScript thread on a phone. We report measurements from a working prototype planner (uniform grid, A^* , geometry-aware post-simplification) over Kartverket vector tiles: grid *construction*, not search, dominates runtime (17 250 cells classified; 326 expanded); a conservative corner rule sealed a 250 m fairway at 120 m cells; and no single resolution serves both harbour and passage scales within a fixed cell budget. We map these problems to established theory (configuration-space erosion, Voronoi retraction, minimum-link paths, relaxation-based two-level oracles), identify gaps in the literature, and present a pre-registered factorial experiment (confirmatory hypotheses H1–H3, exploratory questions E1–E3) designed to select the production architecture.

Keywords: maritime route planning; grid discretization; any-angle path planning; configuration space; minimum-link paths; experimental algorithmics; mobile computing.

Sammendrag. Automatiske rutforslag for fritidsbåter i norsk skjærgård bryter med standardantakelsene i tre etablerte forskningsfelt: ingen graf er gitt (det seilbare rommet må diskretiseres fra kartpolygoner og dybde-data i spørreøyeblikket); grafen er fartøys-spesifikk (dypgang, bredde, høyde); resultatet er rådgivende og må kunne redigeres og begrunnes av et menneske; feilene er asymmetriske (en falsk «klar» kan sette båten på grunn); og alt skal kjøre interaktivt, uten nett, på én JavaScript-tråd på en telefon. Vi rapporterer målinger fra en fungerende prototyp over Kartverkets vektorfliser: *konstruksjonen* av rutenettet — ikke søket — dominerer kjøretiden; en konservativ hjørneregulering forseglet et 250 m bredt løp ved 120 m celler; og én uniform oppløsning kan ikke dekke både havne- og overfartsskala innenfor et fast cellebudsjett. Vi navngir problemene i etablert teori, identifiserer hull i litteraturen, og presenterer et forhåndsregistrert eksperimentdesign som skal avgjøre produksjonsarkitekturen.

1 Innledning

Kartplottere for fritidsbåter har gjort elektroniske sjøkart allment tilgjengelige, og automatisk rutforslag mellom to punkter — fra egen plass til en gjestehavn («dock-to-dock») — er i ferd med å bli en forventet funksjon. BoatNav er en norsk kartplotter-app bygget på Kartverkets sjøkartdata, og denne artikkelen springer ut av arbeidet med nettopp en slik funksjon. Forslaget skal være *rådgivende*: et menneske skal lese ruta, forstå hvorfor hvert knekkpunkt ligger der det ligger, justere den for hånd før avreise, og deretter styre etter den. Beregningen skal skje interaktivt (om lag ett sekund), uten nettverkstilgang, på en telefon.

Tre forskningsfelt ligger nær, men ingen av dem dekker problemet. *Veinettstrutning* [2, 8] forutsetter en gitt, statisk og fartøysuavhengig graf; her må grafen konstrueres fra kartgeometri i spørreøyeblikket, og den avhenger av båtens dypgang, bredde og høyde. *Robot- og spillplanlegging* [5, 13] opererer typisk på småskala rutenett med symmetriske feilkostnader og uten krav om at resultatet skal begrunnes overfor og redigeres av et menneske. *Maritim weather routing* [10, 29] gjelder storskala optimering i åpent hav, ikke geometrisk hindringsunngåelse i hjørnetett skjærgård på mobilbudsjett.

Vi har implementert en fungerende planlegger (uniformt rutenett, A^* , eksakt geometribevisst etterforenkling) og målt dens atferd i ekte kartdata. Målingene peker et annet sted enn klassisk algoritmeanalyse ville lett: nesten hele kjøretiden går med til å bygge rutenettet, ikke til å søke i det; den konservative diskretiseringen som sikkerheten krever, forsegler smale løp; og én uniform oppløsning kan ikke betjene både havneområdet og overfarten i samme spørring.

Artikkelens bidrag er: (i) en problemkarakterisering med målinger fra et fungerende system i reelle norske kartdata (§3); (ii) en kobling av de målte problemene til etablert teori, som viser at flere av dem har navn og delvise svar (§4); (iii) identifikasjon av konkrete hull i litteraturen, blant annet sammenlikningen mellom any-angle-søk og geometribevisst etterforenkling (§4); (iv) et forhåndsregistrert eksperimentdesign med konfirmatoriske hypoteser og kvantifiserte løsningsmål som skal avgjøre produksjonsarkitekturen (§5).¹

¹ Målingene stammer fra et internt problemnotat (26. juli 2026); det fullstendige forskningsopplegget (DSRM-design, hypoteser, målemetodikk) er nedfelt i et internt designdokument av samme dato. Begge er tilgjengelige på forespørsel.

2 System, krav og datagrunnlag

2.1 Krav fra praksis

Tre trekk ved anvendelsen former alle tekniske valg.

Forslaget er rådgivende, ikke autonomt. Det gir to krav som sjelden stilles i planleggingslitteraturen. For det første *få knekkpunkter*: en optimal sti med 62 punkter langs cellevegger er ubrukelig å redigere; ønsket er en kort sti med få vertekser — nærmere minimum-link-path enn shortest-path. For det andre *begrunnbarhet per legg*: systemet sier aldri «trygt», det sier «ingen funn innenfor 50 m i dataene» og lister hva som ble funnet. En algoritme hvis svar ikke kan dekomponeres i per-legg-begrunnelser er vanskelig å anvende, uansett optimalitet.

Feilene er asymmetriske. En falsk «klar» kan sette en båt på et skjær; en falsk «sperrert» er bare irriterende. All diskretisering må derfor være konservativ, og manglende data må telles som fare: et dybdepunkt uten verdi behandles som farlig, og kartområde som ikke er lest, behandles som ufarbart. Konservatismen har imidlertid en pris som viste seg å være langt høyere enn ventet (§3.1).

Kjøremiljøet er en telefon. JavaScript (Hermes), én tråd, ingen native kode, ingen skyberegning i drift. Interaksjonsbudsjettet er om lag ett sekund med spinner, med et absolutt tak på få sekunder.

2.2 Datagrunnlag

Kartverkets sjøkartdata er forhåndsprosessert til vektorfliser (PMTiles, zoomnivå 14) og leses direkte på klienten. Per utsnitt foreligger *flater* (land, tørrfall, gruntområder med dybdeattributt) og *punkter* (skjær, grunner, dybdepunkter — dybde ofte, men ikke alltid, angitt). Ringene er rene geometrier uten topologi eller naboskapsstruktur, og de er klippet per flis: samme øy opptrer som flere polygoner. Målt i et havneområde (Langesund, boks $\pm 2,5$ km) ga 49 fliser 2 804 punkter og 3 352 flater; ett enkelt landpolygon hadde 841 noder. Tettheten er svært ujevn: åpen fjord er tom, skjærgård er tett.

2.3 Dagens planlegger

Prototypen arbeider i fem steg: (1) romlig indeks over utsnittet; (2) uniformt boolsk rutenett der en celle sperrer hvis senteret eller noe av de fire hjørnene ligger i sperret flate, eller en punktfare ligger nærmere enn halve cellediagonalen; (3) A^* med 8-naboskap og euklidisk heuristikk; (4) grådig etterforenkling («hopp så langt fram som mulig») der hvert kandidathopp valideres med eksakt segment-mot-polygon-sjekk mot originaldataene; (5) endelig verifisering av hvert gjenværende legg, med per-legg-begrunnelse. Arbeidsdelingen er bevisst: *rutenettet avgjør om det finnes vei; den eksakte sjekken avgjør hvordan ruta ser ut* — slik arves ikke diskretiseringsfeilen inn i sluttsvaret. En tidligere variant (rekursiv halvering med informerte sidesprang rundt hindringer) ble forkastet: den løser én hindring elegant, men konvergerer ikke over en rekke av dem — og skjærgård er nettopp en rekke hindringer.

3 Målte problemer

3.1 Konservativ diskretisering forseglar smale løp

Hjørneregelen er nødvendig: med ren senteresting smalt diagonalsteg mellom to «farbare» celler tvers gjennom et nes, og planleggeren leverte stier som dens egen eksakte sjekk etterpå avviste — forslag som ikke holdt sitt eget løfte. Men samme regel sperrer ethvert løp smalere enn omtrent to celler (figur 1). Tabell 1 viser målingen: i Langesund havn eksisterte ikke en ~ 250 m bred, hurtigbåttrafikkert renne ved 120 m celler — og merk at nettet da brukte 1 932 av 60 000 tillatte celler. Det var ikke cellebudsjettet som tvang grovheten, men en dårlig valgt konstant.

3.2 Én uniform oppløsning dekker ikke begge skalaer

Dette er hovedproblemet, og det er uløst i prototypen. Cellesørrelsen bestemmes av søkeboksens areal under et celletak på 60 000 (tabell 2). På 223 m celler er forseglingsproblemet tilbake i full styrke: sund smalere enn ~ 450 m forsvinner. Planleggeren virker altså i havneområdet man starter i, og degraderer til «åpen sjø»-oppløsning nettopp når ruta blir lang nok til å passere gjennom skjærgård underveis. Å heve celletaket løser det ikke innenfor kjøretidsbudsjettet: klassifiseringskostnaden er lineær i celleantallet, og en 40 km boks med 40 m celler er $\sim 10^6$ celler.

3.3 Endepunkter i hindringsflater er normalt ille

En båt ved kai ligger nesten alltid *innenfor* havnas kystkontur i kartdataene; den faktiske GPS-posisjonen ved kai i Langesund ligger inne i det nevnte 841-noders landpolygonet. Første versjon returnerte «startpunktet ligger på land» og feilet dermed i 100 % av tilfellene der funksjonen er mest nyttig — planlegging fra egen plass. Nåværende håndtering snapper begge ender til nærmeste farbare celle (ring-søk, 600 m) og merker strekningen ut fra plassen eksplisitt som *uverifisert* i stedet for å skjule den. Problemet — start og mål i det formelt ikke-farbare området, der snapping må være både geometrisk fornuftig og semantisk ærlig — framstår underbeskrevet i litteraturen.

3.4 To orakler som er uenige

Systemet har to farevurderinger: en *grov* (celleklassifisering) som avgjør om vei finnes, og en *eksakt* (segment-mot-polygon med 50 m korridor) som godkjenner hvert legg. Den grove viste seg å være *strengere* enn den eksakte: punktfarer sperrer celler ubetinget, mens den eksakte policyen godtar passering 40 m fra en grunne. Resultatet var at planleggeren svarte «ingen vei» på strekninger verifikatoren gladelig ville godkjent — en stille inkonsistens, ikke en synlig feil. §4 viser at dette bryter et formulerbart designkrav.

3.5 Kostnaden ligger i konstruksjonen, ikke i søket

Etter optimalisering bruker hele forslaget (nettbygging + A^* + forenkling + verifisering) ~ 750 ms på havneområdet, målt i Node på en arbeidsstasjon; Hermes på telefon er ennå ikke målt og forventes vesentlig tregere. Fordelingen er det mest



Figur 1: Skjematisk illustrasjon av forseglingsmekanismen (§3.1). En celle sperres når et hjørne berører en hindringsflate (skravert: sperrede celler). (a) Når celledsiden nærmer seg løpsbredden, får alle celler i løpet hjørner i land — løpet forsegles, og planleggeren svarer «ingen vei» selv om løpet er farbart. (b) Med tilstrekkelig fin oppløsning består en fri korridor (grønn). Oppløsningen er dermed ikke en smakssak, men en funksjon av smaleste passasje som skal garanteres (§4).

Tabell 1: Samme havneområde (Langesund, boks $\pm 2,5$ km), samme data, to celledørrelser. Renna i løpet er ~ 250 m bred.

Celledørrelse	Rutenett	Farbare celler	Utfall
120 m	$46 \times 42 = 1\,932$	1 019 (53 %)	Ingen sti. A* åpnet 4 celler — start lå i isolert lomme
40 m	$138 \times 125 = 17\,250$	10 794 (63 %)	Sti funnet: 62 rutenettpunkter; 326 celler åpnet

Tabell 2: Celledørrelse som funksjon av strekning under fast celledørrelse (60 000).

Strekning	Boks	Celle	Celler
0,3 nm (havn)	$\pm 2,5$ km	40 m	17 250
8,4 nm (overfart)	$\pm 12,6$ km	223 m	60 180

interessante: nesten alt går med til å bygge nettet. A* åpnet 326 av 17 250 celler; klassifiseringen dominerer fordi hver celledørrelse kjører punkt-i-polygon mot lokale flater, og et 841-noders polygon gjør hver test dyr. Én enkel gevinst er tatt (hjørnegitteret klassifiseres én gang, delt mellom fire naboceller: 1 162 ms $\rightarrow$ 749 ms); scanline-rasterisering — $O(\text{kanter} + \text{fylte celler})$ i stedet for $O(\text{celler} \times \text{kanter})$ — er identifisert, men ikke gjort. Søkedelen er tilsvarende uoptimalisert (lineært minimumsuttak, ingen binærheap), og det har ikke målbar betydning ved dagens celledørrelser. Klassisk profilering av A*-implementasjonen er altså feil sted å lete i dette domenet.

3.6 Etterforenklingens grense

Den grådige etterforenklingen gir få punkter, men den er post hoc: den kan bare kollapse den stien A* allerede har valgt, og A* valgte etter rutenettavstand, ikke sjøavstand. Målte omveier rundt øyer: +140 % i ett tilfelle, +55 % i et annet der ruta er geografisk riktig. Any-angle-søk [5] påvirker *hvilke noder som utvides*, ikke bare hvordan resultatet ser ut etterpå, og er det nærliggende svaret — men litteraturen sammenlikner any-angle mot naiv strengtrekking i nettet, ikke mot etterforenkling med tilgang til originalgeometrien (§4).

3.7 Binær farbarhet og «urimelig omvei»

En båt vil ikke gå så nær et skjær som geometrien tillater; ønsket klaring avhenger av sjøgang, mørke, dypgang og skipperens nerver. Dagens modell er boolsk med én global 50 m-korridor, og to utslag er observert: ruter legges gjer-

ne helt i kanten av det farbare (kostnaden er ren lengde), og terskelen for hva som sperrer en celle blir et hardt valg der virkeligheten er glidende. Beslektet: «urimelig omvei» lar seg ikke uttrykke som forholdstall alene. Et tidlig forslag gikk 280 km rundt enden av en kunstig vegg; men et rent forholdstak straffer korte strekninger urimelig — veien ut av Langesund havn er 1,35 nm mot 0,28 nm rett linje (+387 %) og helt normal. Nåværende regel (forkast bare hvis omveien er stor både relativt, $2,5\times$, og absolutt, 3 km) er en innrømmelse av at fenomenet er en egenskap ved *konnektiviteten* — nesten-adskilte komponenter forbundet gjennom en omvei ingen ville valgt — ikke ved stien.

4 Problemene i lys av litteraturen

Tabell 3 oppsummerer koblingen. Tre punkter fortjener utdyping.

Diskretisering har en teori. Med båten modellert som disk i konfigurasjonsrommet [18] er erosjon av det frie rommet med båtradius pluss margin den konservative operasjonen, og disc-retraksjon langs Voronoi-diagrammet [21] er *komplett*: den finner vei hvis og bare hvis en finnes med krevd klaring. For rasterrepresentasjoner følger at oppløsningen må ligge under smaleste passasje som skal garanteres (minus fartøysbredden) — figur 1 er altså ikke en kuriositet, men et brudd på en formaliserbar betingelse. For uniforme rutenett er dessuten diskretiserings-suboptimaliteten kjent: verstefall $\sim 8,2$ % for 8-naboskap og $\sim 2,8$ % for 16-naboskap, hvilket setter en nedre grense for hva ren nett-oppløsning kan levere uten etterbehandling.

Orakelkravet kan formuleres presist. Et grovt nivå som skal avkrefte eksistens trygt («ingen vei») må være en *relaksasjon* av det eksakte problemet — optimistisk, med samme logikk som at admissible heuristikker aldri overestimerer [23]. Et konservativt grovt nivå kan bare *bekreft* vei. Prototypen har rollene byttet om, og det forklarer de falske «ingen vei» i §3.4 som et arkitekturbrudd, ikke en parameterfeil. Fundamentet er klassisk: A* med konsistent heu-

Tabell 3: Målte problemer, etablert teori og identifiserte hull.

Målt problem	Etablert ramme («problemet har navn»)	Åpent hull
Forsegling av smale løp (§3.1)	Konfigurasjonsrom-erosjon [18]; komplett disc-planlegging via Voronoi-retraksjon [21]	Konservativ og konnektivitetsbevarende diskretisering bygget i sanntid fra flisdata på telefon
Skala-spenntet (§3.2)	Hierarkisk/adaptiv planlegging [3, 25, 6]	Grov-til-fin der grovnivået er <i>pessimistisk</i> : kan ikke beskjære eksistens uten å risikere eneste farbare vei
Orakel-uenighet (§3.4)	Relaksasjon/admissibilitet [23]; potensialfunksjoner og reweighting [4]	To-nivå-arkitektur som er konsistent i begge retninger (optimistisk eksistens + eksakt konstruksjon)
Konstruksjonen dominerer (§3.5)	Implisitte grafrepresentasjoner [4]; scanline-rasterisering	Lat konstruksjon kobler byggekostnad til søkefronten — og dermed til heuristikkstyrken; lite belyst når konstruksjonen dominerer
Få vertekser som mål (§3.6)	Minimum-link-paths [26]; bikriterie-stier [1, 20]; skranket korteste vei [11]	Any-angle-søk [5, 14] og lærte heuristikker [28] mot <i>geometribevisst</i> etterforenkling — sammenlikningen finnes ikke i litteraturen
Klaring og «urimelig omvei» (§3.7)	Avstandstransform [7]; GVD-skjelett; flerkriterie korteste vei [12, 16]	Flaskehals-/omveisdeteksjon i skjelettgrafen som prinsipiell erstatning for forholdstall-heuristikk; presentasjon av avveiningen for én beslutningstaker
Delvis ukjent kart (§2)	Canadian Traveller [22]; inkrementell replanlegging [17]	Omvendt fortegn: ukjent = <i>ufarbart</i> , og utvidelse av det kjente koster nettverk og tid

ristikk h er nøyaktig Dijkstra på en reweightet graf med $w_h(u, v) = w(u, v) - h(u) + h(v)$, der konsistens er ekvivalent med $w_h \geq 0$ [4, 15] — heuristikker, potensialer og relaksasjoner er samme mekanisme sett fra tre kanter.

Konstruksjonskostnaden endrer hva som er verdt å forske på. Når klassifiseringen dominerer og søket åpner under 2 % av cellene (§3.5), er standardmetrikken «ekspanderte noder» misvisende alene. Men koblingen består: med *lat* klassifisering (celler klassifiseres først når søket berører dem) blir konstruksjonskostnaden proporsjonal med søkefronten i stedet for boksarealet — og da blir heuristikkstyrken totalkostnadskritisk likevel. Sterkere admissible heuristikker enn euklidisk avstand kan forhåndsregnes: ALT-landemerker [9] kan produseres i samme pipeline som kartflisene og distribueres i offline-kartpakker, uten skyberegning i drift.

5 Forhåndsregistrert forskningsopplegg

Eksperimentene er utformet, men ikke kjørt; denne seksjonen fungerer dermed som offentlig forhåndsregistrering av hypotesene. Rammen er Design Science Research [24]: artefakten er en modulær planlegger-harness der dagens prototyp fryses som baseline K0, og kunnskapsbidraget er de empiriske svarene. Metodikken følger etablert praksis for eksperimentell algoritmikk [19], med validitetsanalyse etter Wohlin *mf.* [27].

5.1 Design

Faktorene er *diskretisering* (uniformt grid 8/16-naboskap, adaptiv quadtree, korridoravgrenset synlighetsgraf, klarringsfelt med GVD-skjelett), *konstruksjonsstrategi* (upfront scanline mot lat on-demand), *søk* (Dijkstra, A^* med euk-

lidisk heuristikk, ALT med flislagrede landemerker, Theta^{*}), *etterbehandling* (geometribevisst etterforenkling av/på) og *båtprofil* (dypgang 0,5/1,0/2,0 m med bredde- og høydekrav). Korpuset er 270 instanser stratifisert over farvannstype (indre skjærgård, fjord, åpen kyst) og distanse (<5, 5–30, >30 nm), med Langesund-strekningene fra §3 som forankringsinstanser og en kvote instanser med endepunkt innenfor kystkonturen (§3.3). Fasit er eksakt korteste vei i synlighetsgrafen over erodert geometri, verifisert mot euklidisk nedre og finmasket øvre skranke. Kvantifiserte løsningsmål: aldri falsk «klar»; ingen falsk «ingen vei» for garantert passasjebredde; lengde-suboptimalitet $\varepsilon \leq 5\%$ (p95); ≤ 15 vertekser med per-legg-begrunnelse; p95-responstid ≤ 1 s under 5 nm og ≤ 3 s opp til 30 nm på fysisk målenhet (Hermes); alt uten nett.

5.2 Hypoteser

Tabell 4 oppsummerer. Konfirmatoriske hypoteser testes parvis per instans med Wilcoxon signed-rank og Holm-korreksjon ($\alpha = 0,05$); eksplorative spørsmål rapporteres som effektestimater med bootstrap-konfidensintervaller. Alle tidstall rapporteres faseoppdelt (konstruksjon/søk/etterbehandling/verifisering), siden §3.5 viser at aggregerte tall skjuler det som betyr noe.

6 Trusler mot validitet

Indre validitet. Alle publiserte tidstall er målt i Node på arbeidsstasjon; Hermes på telefon er umålt og forventes vesentlig tregere. Alle stokastiske målinger i eksperimentet gjøres derfor på fysisk målenhet, og Node–Hermes-forholdet etableres i en pilot før hovedkjøringen. Implementasjonsbias motvirkes med felles datastrukturer, frosset ba-

Tabell 4: Forhåndsregistrerte konfirmatoriske hypoteser (H) og eksplorative spørsmål (E). $N(\cdot)$ er antall åpne celler.

ID	Påstand
H1a	I indre skjærgård gir euklidisk heuristikk liten gevinst: $\text{median } N(\text{Dijkstra})/N(A^*\text{-euklid}) < 2$.
H1b	Flislagrede ALT-landemerker gir stor gevinst i samme farvann: $\text{median } N(\text{Dijkstra})/N(\text{ALT}) > 10$.
H1c	For strekninger ≥ 5 nm slår lat konstruksjon + ALT upfront-konstruksjon + ALT på total kjøretid (byggekostnad $\propto$ søkefront, ikke boksareal).
H2	Med samme cellebudsjett (60 000) gir adaptiv quadtree null falske «ingen vei» på instansene der uniformt grid taper passasjer (sund 200–450 m), uten å være signifikant dårligere på suboptimalitet, tid eller minne.
H3	For strekninger > 30 nm holdes responstidsbudsjettet bare med forhåndsberegning distribuert i kartpakkene.
E1	Effekten av any-angle-søk (Theta*) på lengde, vertekstall og tid når <i>begge alternativer får geometribevisst etterforenkling</i> — sammenlikningen litteraturen mangler.
E2	Formen på avveiningen lengde $\leftrightarrow$ minimumsklaring under klaringsbasert kostnad, og dens følsomhet for vektparameteren.
E3	Presisjon/dekning for flaskehals- og omveisdeteksjon i GVD-skjelettgrafen mot dagens forholdstall+meter-regel.

seline og lik optimeringsinnsats per algoritme. *Konstruktiv validitet.* «Rutekvalitet» er flerdimensjonal (lengde, vertekstall, klaring, begrunnbarhet); dimensjonene rapporteres separat og veies ikke sammen i én indeks. AIS-data brukes kun til plausibilisering av klaringskostnaden — fritidsbåter under 15 m har ikke AIS-plikt, så sporene overrepresenterer større fartøy. *Ytre validitet.* Tre farvannstyper uttømmer ikke norskekysten, men er stratifisert for topologisk spennvidde; kartgrunnlagets standardformat gjør overføring til nabolands farvann plausibel. Endelig en avgrensning som også er et etisk standpunkt: algoritmen kan gi *korteste lovlige* vei, ikke *riktig* vei i sjøgang og mørke. Systemet uttaler seg deretter — det finnes ingen «trygg»-verdi i koden, og føreren har alltid ansvaret.

7 Konklusjon

Vi har karakterisert automatisk ruteplanlegging i norsk skjærgård på mobilklient som et eget problem, målt hvor kostnadene og feilene faktisk oppstår i en fungerende prototyp, vist at flere av problemene har navn og delvise svar i etablert teori — og at minst ett (any-angle mot geometribevisst etterforenkling) mangler svar i litteraturen. Det forhåndsregistrerte eksperimentopplegget skal avgjøre produksjonsarkitekturen; resultatene vil bli rapportert i en oppfølgende artikkel, uansett utfall. Vi tar gjerne imot innspill fra fagmiljøene innen kombinatorisk søk, beregningsgeometri og maritim informatikk — særlig pekere til arbeid vi kan ha oversett på punktene i tabell 3.

Erklæringer

Interessekonflikt: Forfatteren eier Fellag AS, som utvikler BoatNav. *KI-assistanse:* Forskningsopplegget og manuskriptet er utarbeidet med assistanse fra Claude (Anthropic); alle målinger stammer fra BoatNav-implementasjonen, og forfatteren står ansvarlig for innholdet. *Bidragsroller (CRediT):* Conceptualization, Methodology, Software, Investigation, Formal analysis, Validation, Visualization og Writing er utført av forfatteren med KI-assistanse som beskrevet over. *Validation* er holdt uavhengig av implementeringen. *Data og kode:* Målingene og eksperiment-harnessen planlegges åpnet sammen med resultatartikkelen; inntil da tilgjengelig på forespørsel. *Ansvarsavgrensning:* BoatNav gir veiledende ruteslag for fritidsbruk og er ikke et godkjent navigasjonshjelpemiddel.

Versjon 1.1 (30. juli 2026): erklæringer om interessekonflikt, KI-assistanse og bidragsroller lagt til etter fagfellevurderingen.

Versjon 1.1 (30. juli 2026): referanser som sto ubrukt i litteraturlista er knyttet til teksten der innholdet hviler på dem. Ingen påstander, tall eller konklusjoner er endret i noen av versjonene.

Referanser

- [1] E. Arkin, J. S. B. Mitchell og C. Piatko. Bicriteria shortest path problems in the plane. *Proc. 3rd Canadian Conf. on Computational Geometry*, 1991.
- [2] H. Bast mfl. Route Planning in Transportation Networks. I *Algorithm Engineering*, LNCS 9220, Springer, 2016.
- [3] A. Botea, M. Müller og J. Schaeffer. Near Optimal Hierarchical Path-Finding. *Journal of Game Development* 1(1), 2004.
- [4] T. H. Cormen, C. E. Leiserson, R. L. Rivest og C. Stein. *Introduction to Algorithms*, 4. utg. MIT Press, 2022.
- [5] K. Daniel, A. Nash, S. Koenig og A. Felner. Theta*: Any-Angle Path Planning on Grids. *Journal of Artificial Intelligence Research* 39, 2010.
- [6] D. Demyen og M. Buro. Efficient Triangulation-Based Pathfinding. *Proc. AAAI*, 2006.
- [7] P. Felzenszwalb og D. Huttenlocher. Distance Transforms of Sampled Functions. *Theory of Computing* 8, 2012.
- [8] R. Geisberger, P. Sanders, D. Schultes og D. Delling. Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks. *Proc. WEA*, 2008.
- [9] A. Goldberg og C. Harrelson. Computing the Shortest Path: A* Search Meets Graph Theory. *Proc. SODA*, 2005.
- [10] H. Hagiwara. *Weather routing of (sail-assisted) motor vessels*. Doktoravhandling, TU Delft, 1989.
- [11] G. Handler og I. Zang. A dual algorithm for the constrained shortest path problem. *Networks* 10(4), 1980.
- [12] P. Hansen. Bicriterion Path Problems. I *Multiple Criteria Decision Making: Theory and Application*, Springer, 1980.
- [13] D. Harabor og A. Grastien. Online Graph Pruning for Pathfinding on Grid Maps. *Proc. AAAI*, 2011.
- [14] D. Harabor, A. Grastien, D. Öz og V. Aksakalli. Optimal Any-Angle Pathfinding in Practice. *Journal of Artificial Intelligence Research* 56, 2016.
- [15] P. Hart, N. Nilsson og B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. on Systems Science and Cybernetics* 4(2), 1968.
- [16] R. Hassin. Approximation Schemes for the Restricted Shortest Path Problem. *Mathematics of Operations Research* 17(1), 1992.
- [17] S. Koenig og M. Likhachev. D* Lite. *Proc. AAAI*, 2002.
- [18] T. Lozano-Pérez. Spatial Planning: A Configuration Space Approach. *IEEE Trans. on Computers* C-32(2), 1983.

- [19] C. C. McGeoch. *A Guide to Experimental Algorithmics*. Cambridge University Press, 2012.
- [20] J. S. B. Mitchell. Geometric shortest paths and network optimization. I *Handbook of Computational Geometry*, Elsevier, 2000.
- [21] C. Ó'Dúnlaing og C. Yap. A “retraction” method for planning the motion of a disc. *Journal of Algorithms* 6(1), 1985.
- [22] C. Papadimitriou og M. Yannakakis. Shortest paths without a map. *Theoretical Computer Science* 84(1), 1991.
- [23] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.
- [24] K. Peffers, T. Tuunanen, M. Rothenberger og S. Chatterjee. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24(3), 2007.
- [25] H. Samet. The Quadtree and Related Hierarchical Data Structures. *ACM Computing Surveys* 16(2), 1984.
- [26] S. Suri. A linear time algorithm for minimum link paths inside a simple polygon. *Computer Vision, Graphics, and Image Processing* 35(1), 1986.
- [27] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell og A. Wesslén. *Experimentation in Software Engineering*. Springer, 2012.
- [28] R. Yonetani, T. Tani, M. Barekatain, M. Nishimura og A. Kanezaki. Path Planning using Neural A* Search. *Proc. ICML*, 2021.
- [29] T. Zis, H. Psaraftis og L. Ding. Ship weather routing: A taxonomy and survey. *Ocean Engineering* 213, 2020.