A Scalable Matheuristic for Routing Capacity-Constrained Groundfish Surveys

Helga Ingimundardóttir[1][0000−0002−2780−3546], Margrét Vala Þórisdóttir[1,2][0009−0006−1468−4003], Bjarki Þór Elvarsson[2][0000−0001−5855−1188], and Tómas Philip Rúnarsson[1][0000−0002−1067−7694]

[1] University of Iceland, Reykjavík, Iceland
{helgaingim,tpr}@hi.is
[2] Marine and Freshwater Research Institute, Hafnarfjörður, Iceland
{margret.vala.thorisdottir,bjarki.elvarsson}@hafogvatn.is

Abstract. Bottom trawl groundfish surveys are planned around a fixed set of sampling stations, but the order of visits and the timing of port returns strongly affect total travel distance and operational feasibility. Although mixed-integer programming (MIP) models can encode capacity and port-call constraints directly, solving the full model to optimality is often impractical at realistic scales. We propose a matheuristic that combines fast tour construction with repeated calls to a time-limited capacity MIP subproblem. Starting from an initial segmented tour, the algorithm iteratively examines boundaries between adjacent segments separated by a port visit and solves a restricted two-segment capacity MIP on the union of stations in the two segments, with fixed endpoints and a single intermediate port call. The MIP reallocates stations between the two segments subject to capacity, and the best incumbent found within a time limit is accepted if it improves total distance. We evaluate the approach under four initialization strategies and a range of MIP time limits. Across scenarios, the method consistently improves baseline plans and provides a fast, reproducible tool for strategic planning and scenario comparison. These results suggest that restricted-MIP local improvement is an effective mechanism for producing high-quality survey routes under realistic operational constraints.

Keywords: Groundfish Survey Problem · Multi-trip VRP · Capacitated Routing with Replenishment · Segment-Based Decomposition · Matheuristics

1 Introduction

Bottom trawl groundfish surveys require visiting a set of sampling stations while respecting operational constraints such as limited onboard capacity, time, and cost. Even when station locations are fixed, the order of visits and the timing of port returns directly influence survey feasibility and efficiency. A method that can generate good survey routes quickly is therefore valuable for both survey planning and survey design.

From an optimization perspective, these decisions couple routing and capacity in a way that makes exact mixed-integer programming (MIP) approaches computationally challenging at realistic scales. Matheuristics address this difficulty by combining mathematical programming with heuristic search [1], often using MIP subproblems to improve a larger solution [2].

This idea appears in several forms in the literature. Fan et al. use a MIP model as a local search operator, repeatedly re-optimizing a selected subset of decisions [5]. Dell'Amico et al. follow a similar strategy, solving MIP subproblems within a heuristic loop and, when needed, imposing a time limit and accepting the best incumbent returned by the solver [4]. This use of time-limited MIP calls–leveraging the best solution found rather than requiring full optimality–is a key ingredient shared with our approach.

Building on these ideas, we introduce a scalable matheuristic for routing capacity-constrained groundfish surveys. Starting from a feasible segmented route, the method applies short, targeted MIP refinements to adjust segment boundaries under vessel capacity. Improvements are accepted whenever found, and the process repeats until no further gains arise. The approach provides a fast and practical baseline planning tool for survey design and scenario evaluation.

Section 2 describes the survey-routing problem, and section 3 presents the capacity-aware MIP model. Because solving this full end-to-end MIP is computationally prohibitive at operational scale, section 4 introduces the scalable matheuristic that effectively addresses the problem. Finally, sections 5 and 6 report the experimental setup and results, followed by discussion and conclusions.

2 Groundfish Survey Problem

Groundfish surveys provide critical information on the abundance, distribution, and biological characteristics of fish stocks and serve as the empirical foundation for sustainable harvest management. In Iceland, these surveys have followed a fixed-station sampling design since 1985, with the same set of trawl stations revisited annually in the spring to maintain a consistent long-term time series [7]. This design aligns with established statistical principles for monitoring population trends [10], while also reflecting the operational continuity of the Icelandic survey program.

For a survey with a given set of stations, the survey leader must determine an efficient routing plan for one or more vessels tasked with visiting and towing at each station. We refer to this routing task as the Groundfish Survey Problem (GSP). The problem is shaped by regulatory and operational constraints. During the survey, some of the catch is retained in order to be sold upon landing. As a result, vessels must periodically return to designated ports to unload because vessel hold capacity is limited. In practice, unloading occurs at regular intervals (typically within five days of operations), embedding mandatory port calls into the routing plan.

The amount of fish caught at each station is uncertain and can vary substantially from year to year. Therefore, routing plans must be robust to variability

in catch volumes, ensuring that capacity constraints are not violated while still maintaining survey coverage. When multiple vessels operate simultaneously, the stations must be partitioned between them in a way that respects their capacities and schedules.

Each trawl station is defined by two tow endpoints, and feasible tow directions may be limited by local bathymetry, prevailing weather, or vessel-specific operational constraints. Consequently, travel between stations is directional: distances from station A to station B may differ from those from B to A. Naïve point-to-point paths can also cross land; to prevent this, we compute waypoint-aware distances that reroute any invalid land-crossing arcs through intermediate waypoints. These corrected, direction-aware distances serve as inputs to both the MIP model and the matheuristic developed for large-scale routing.

Prior research has examined this routing task through both early genetic-algorithm formulations of the GSP [9] and more recent interactive evolutionary reoptimization approaches for dynamic survey planning [13].

In general, the problem involves the following. (i) a large number of geographically distributed stations, (ii) direction-dependent travel and towing constraints and (iii) regulatory requirements that impose periodic port returns due to mandatory retention and finite vessel capacity.

3 Capacity-Aware MIP Model

The survey-routing problem shares features with multi-trip vehicle routing problems (VRP), since capacity limitations force repeated returns to port. At the same time, it differs from classical VRP formulations because all stations must be visited exactly once and the solution forms a single directed tour with capacity-induced segment breaks. In effect, the problem is a hybrid between a multi-trip VRP and a directed traveling salesman problem (TSP) with mandatory unloads.

Node representation. We represent the GSP as a directed tour over a unified paired-node graph. Each physical entity (vessel, station, or port) is modeled by paired nodes $(2i, 2i+1)$ connected by a mandatory internal arc, where each node corresponds to a geographic coordinate $(\mathrm{lat}_i, \mathrm{lon}_i)$. Let n' denote the total number of paired entities. The vessel pair $(0, 1)$ defines the start and end of the tour, respectively. Stations occupy paired nodes $\{2, \ldots, 2n+1\}$, and port nodes occupy the remaining paired nodes $\{2n+2, \ldots, 2n'-1\}$.

For stations, the paired nodes correspond to distinct tow endpoints, allowing the model to select towing direction explicitly. For ports and the vessel, the paired nodes are dummy entry/exit nodes sharing identical coordinates. This preserves a uniform directed routing structure across all node types.

Parameters. Let S denote the set of survey stations, with $n = |S|$, and let P denote the physical set of candidate ports. Since each paired port node in the MIP graph can appear only once in the tour, repeated port visits are represented by duplicating nodes in the fixed port-call set P_{fix}. The total number of paired entities is therefore given by $n' = 1 + |S| + |P_{\text{fix}}|$.

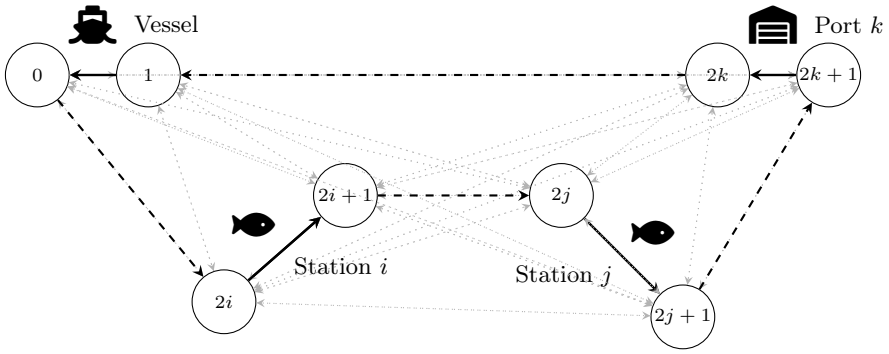


Fig. 1: Indexed node notation used in the MIP model. The underlying structure is a fully bidirectional graph: all node pairs are connected by dashed gray edges to indicate every feasible arc (a, b) in the waypoint-aware distance matrix. Solid black arrows represent mandatory internal arcs that enforce movement from each start node to its corresponding paired end node (e.g., $2i \to 2i{+}1$), capturing the directed towing or port-reset structure. Dashed black arrows show an example tour demonstrating how the vessel may travel from the boat start, through a sequence of stations, then to a port, and finally back to the boat end node.

Let C denote the catch capacity of the vessel, and let $d_{i,j}$ denote the waypoint-aware transit distance between nodes i and j. Since haul distances are fixed across all feasible solutions, internal paired arcs are assigned zero cost, i.e., $d_{2i,2i+1} = d_{2i+1,2i} = 0$ for all paired nodes.

Decision variables. The binary decision variable $x_{i,j} \in \{0, 1\}$ indicates whether arc (i, j) is used in the tour.

Degree constraints. To ensure the solution forms a single tour that visits every node exactly once, each node has one incoming and one outgoing arc:

$$\sum_{i \in \{0,\ldots,2n'-1\}} x_{i,j} = 1 \qquad \forall j \in \{0, \ldots, 2n' - 1\} \qquad \text{(in-degree)}$$

$$\sum_{i \in \{0,\ldots,2n'-1\}} x_{j,i} = 1 \qquad \forall j \in \{0, \ldots, 2n' - 1\} \qquad \text{(out-degree)}$$

These constraints imply that each instantiated port-call node in P_{fix} is visited exactly once.

Direction choice. For each paired-node structure $(2i, 2i + 1)$ we enforce

$$x_{2i,2i+1} + x_{2i+1,2i} = 1 \qquad \forall i \in \{1, \ldots, n' - 1\}. \qquad \text{(pair)}$$

For station pairs, this constraint selects the towing direction. For port and vessel pairs, it enforces traversal through the dummy entry/exit pair.

Objective. We minimize total distance over the selected arcs:

$$\min \sum_{\substack{i\in\{0,\dots,2n'-1\}\\ j\in\{0,\dots,2n'-1\}}} d_{i,j}\, x_{i,j}. \qquad \text{(objective)}$$

Because intra-entity arcs are assigned zero cost, the objective reflects only inter-entity transit distance.

Subtour elimination. Without connectivity constraints, the tour can split into subtours. We add cuts via callbacks: for any subtour arc set $\mathcal{P}$,

$$\sum_{(i,j)\in\mathcal{P}} x_{i,j} \le |\mathcal{P}| - 1. \qquad \text{(subtour elimination)}$$

This prevents disconnected cycles and yields a single connected tour.

Capacity and catch flow. To enforce capacity and represent cumulative catch carried along the tour, we use directed flow variables $w_{i,j}$ for the load on arc (i,j), and $v_{i,j}$ for catch injected at station endpoints on the internal arc. Before presenting constraints, note that $w_{i,j}$ must respect both direction and capacity and propagates the load forward along selected arcs.

Flow conservation at stations. For each station $j \in \{1,\dots,n\}$, we enforce conservation at both endpoints:

$$\begin{aligned} \sum_{k\in\{0,\dots,2n'-1\}\setminus\{2j+1\}} w_{k,2j} + v_{2j,2j+1} &= w_{2j,2j+1},\\ \sum_{k\in\{0,\dots,2n'-1\}\setminus\{2j\}} w_{k,2j+1} + v_{2j+1,2j} &= w_{2j+1,2j}. \end{aligned} \qquad \text{(flow conservation)}$$

The station catch injected on the internal arc satisfies

$$v_{2j,2j+1} + v_{2j+1,2j} = a_j. \qquad \text{(station expected catch)}$$

Flow transfer after towing. After completing the tow, the accumulated load must transfer to the next arc in the tour:

$$\begin{aligned} w_{2j,2j+1} &= \sum_{k\in\{0,\dots,2n'-1\}\setminus\{2j,2j+1\}} w_{2j+1,k},\\ w_{2j+1,2j} &= \sum_{k\in\{0,\dots,2n'-1\}\setminus\{2j,2j+1\}} w_{2j,k}. \end{aligned} \qquad \text{(flow transfer)}$$

These equalities propagate the accumulated catch from a station to the next arc.

Port unload. At ports, the vessel unloads before leaving. For each port $k \in \{n+1,\dots,n'-1\}$ and each of its nodes $p \in \{2k, 2k+1\}$:

$$\sum_{j\in\{0,\dots,2n'-1\}} w_{p,j} = 0. \qquad \text{(port reset)}$$

Algorithm 1: General capacity-aware MIP (**C-MIP**) procedure for computing a feasible end-to-end survey tour.

Input: *Problem instance*: stations S, fixed port-call set P_{fix}, vessel capacity C, distances $d_{i,j}$. *Solver controls*: time limit $L_{N\text{seg}}$
Output: Best feasible tour found within $L_{N\text{seg}}$
1 Build the directed graph with paired nodes for the vessel, stations S, and instantiated port calls P_{fix}, using waypoint-aware transit distances $d_{i,j}$ for inter-entity arcs and zero cost for intra-entity arcs;
2 Create the MIP with degree, pair, flow, and capacity constraints;
3 Enable subtour elimination via callback;
4 Set solver time limit to $L_{X\text{seg}}$;
5 Optimize and record the best feasible incumbent;
6 **return** incumbent tour and objective, if any;

Capacity bounds. Capacity is enforced by

$$w_{i,j} \le C\, x_{i,j}, \qquad v_{i,j} \le C\, x_{i,j}. \tag{capacity}$$

Together with the port reset constraints, this forces a return to port once the accumulated catch approaches capacity.

Discussion and complexity. The full formulation presented here, which we refer to as the C-MIP (capacity-aware MIP), extends classical flow-based subtour-elimination models for the TSP [3] by incorporating directed towing, load propagation, mandatory port resets, and a vessel capacity constraint. These additions substantially increase computational difficulty, especially when port choices and the number of required port calls are not fixed in advance. The no-port variant (NP-MIP) removes all capacity constraints, reducing the problem to the classical directed TSP. Solving the full C-MIP over the complete station set corresponds to the full-segment case in Algorithm 1, with time limit $L_{N\text{seg}}$.

4 Matheuristic Framework

We model a tour as an ordered list of capacity-feasible segments separated by ports. Let C denote vessel capacity. Each segment is a directed path that begins at a port (or the vessel's start location), visits a subset of stations whose accumulated expected catch remains below C, and ends at the next port where the vessel unloads.

The proposed matheuristic (MH) constructs such a tour by first producing an initial station ordering, then forming an initial capacity-feasible segmentation, and finally iteratively improving the solution by re-optimizing *adjacent* segment pairs using local C-MIP subproblems. This yields a scalable approach that exploits the problem structure while avoiding the combinatorial difficulty of solving the full C-MIP.

Algorithm 2: Matheuristic boundary-sweep framework (**MH**)

Input: Stations S, ports P, capacity C, waypoint-aware distances $d_{i,j}$, start/end vessel nodes, initialization mode InitMode, single-segment TSP limit $L_{1\text{seg}}$, two-segment C-MIP limit $L_{2\text{seg}}$
Output: Best segmented tour found

Phase 0: Initialization and baseline;
Obtain S_{ord} and initial partition $\mathcal{V}$ using InitMode;
Evaluate each segment in $\mathcal{V}$ using the directed segment-TSP with limit $L_{1\text{seg}}$;
Mark all boundaries as *active*;
Phase 1: Boundary-sweep improvement;
`// A sweep is one full traversal of all boundaries in circular order.`
repeat
  foreach *boundary b between adjacent segments in tour order* **do**
    if *b is not active* **then**
      continue
    end
    Mark b as inactive;
    Call algorithm 3 on boundary b using limits $L_{1\text{seg}}$ and $L_{2\text{seg}}$;
    if *boundary changed* **then**
      Update $\mathcal{V}$;
      Mark the modified boundary and its neighboring boundaries as *active*;
    end
  end
until *no boundary changed in a full sweep*;

Algorithm overview. The full matheuristic consists of two phases: an initialization phase that produces an initial ordering construction and segmentation, and an iterative boundary improvement phase that repeatedly refines adjacent segment pairs. The overall procedure is given in algorithm 2. Whenever a boundary is selected during the sweep, the algorithm invokes the local two-segment optimization routine in algorithm 3. Accepted changes update the segmentation and trigger local re-optimization of the affected segments. The process terminates once a full sweep yields no improvements.

Initial ordering. The framework requires an initial ordering of the survey stations from which a capacity-feasible tour can be constructed. This ordering may be produced by any heuristic or optimization-based procedure, and the choice can substantially influence the resulting segmentation and solution quality. A wide variety of initialization strategies are available in practice, offering different balances between computational effort and ordering quality. In our experimental study (see section 5), we examine four representative initialization variants ranging from lightweight greedy methods to a more computationally demanding NP optimum based baseline.

Algorithm 3: Optimize Boundary Capacity (C-MIP subroutine in MH)

Input: Boundary b between segments S_L and S_R, boundary port p, capacity C, single-segment TSP limit $L_{1\text{seg}}$, two-segment C-MIP limit $L_{2\text{seg}}$
Output: Updated segments S'_L, S'_R if improved; otherwise unchanged

Boundary setup: Set $S_{2\text{seg}} = S_L \cup S_R$;
Fix start and end nodes around $S_{2\text{seg}}$ and enforce exactly one visit to p;
Solve the two-segment C-MIP (algorithm 1) on $S_{2\text{seg}}$ with time limit $L_{2\text{seg}}$;
Extract tentative segments S'_L and S'_R from the resulting C-MIP tour;
Re-optimize each tentative segment using a directed segment-TSP with $L_{1\text{seg}}$;
if *either segment-TSP fails* **then**
  return no change;
end
if $d(S'_L) + d(S'_R) < d(S_L) + d(S_R)$ **then** improvement
  return updated segments S'_L, S'_R;
end
return no change;

Initial segmentation. Let S_{ord} denote the ordered list of stations. From the total expected catch, we compute the minimum number of required segments $N_{\min} = \lceil (\sum_{j \in S} a_j)/C \rceil$. The algorithm then scans the ordering, accumulating load and inserting the nearest feasible port (with respect to $d_{i,j}$) either *before* a station if adding it would exceed capacity, or *after* a station once the accumulated load reaches C and additional stations remain. Each port insertion resets the load.

The resulting visit list $\mathcal{V} = \{S_1, \ldots, S_N\}$ defines a capacity-feasible segmentation with $N \geq N_{\min}$, forming a partition of the station set. Each segment is then solved as a directed segment-TSP (time limit $L_{1\text{seg}}$), producing a baseline segmented tour. All segment boundaries are marked as eligible for improvement.

Note, unlike the full C-MIP, the MH does not require a fixed port-call set P_{fix} to be specified in advance. Instead, it operates directly on the physical candidate port set P, inserting port visits dynamically during segmentation.

Two-segment subproblem. When the sweep selects a boundary b between adjacent segments S_L and S_R with separating port p, we solve a local C-MIP on $S_L \cup S_R$ with the start node fixed (previous port or vessel start), the end node fixed (next port or vessel end), and a single port visit forced at p. The C-MIP attempts to reassign stations between the two segments subject to capacity. A time limit $L_{2\text{seg}}$ is imposed and the best incumbent is used; if either resulting segment-TSP fails, the change is rejected. The procedure is given in algorithm 3.

Sweep and termination. The algorithm sweeps boundaries in circular tour order. A boundary is defined between adjacent segments S_k and $S_{k+1 \bmod N}$, and whenever such a boundary is modified, its two neighboring boundaries (at $k-1 \bmod N$ and $k+1 \bmod N$) are marked *active* for re-evaluation. A *sweep* consists of visiting all boundaries once in this circular order. The algorithm terminates when a full sweep yields no accepted changes.

5 Experimental Setup

Data and instance summary. The data come from the 2023 spring groundfish survey conducted by the Icelandic Marine and Freshwater Research Institute [8]. The instance contains $n = 580$ trawling stations and $|P| = 23$ ports, with 7 repeated port visits. The full dataset also includes 133 waypoint nodes and 4 survey vessels. For simplicity, all experiments consider a single vessel, *Árni Friðriksson*, based in Hafnarfjörður with capacity $C = 45$ tonnes.

In total, the instance contains $n' = 604$ entities (stations, ports, and vessel), corresponding to 1208 nodes in the paired representation. Station catch values a_j are expected catches derived from historical time series. The total catch is 528 tonnes, implying a minimum of $N_{\min} = 12$ capacity-feasible segments.

Distances $d_{i,j}$ are waypoint-aware shortest-path distances computed with Dijkstra's algorithm, rerouting land-crossing arcs via waypoint nodes. The cumulative haul distance across all stations is fixed at 2242.85 nm and does not depend on routing decisions. Consequently, the optimization focuses on minimizing transit distance between stations and ports.

Port configuration. The C-MIP uses a fixed port list P_{fix} is set here to match the port structure of the 2023 survey, and requires $N = 13$ segments. In contrast, the matheuristic selects ports dynamically from the unique candidate set P.

Experimental scenarios. We evaluate three model classes. The NP-MIP is a no-port directed TSP over all stations and provides both an optimal port-free routing and a theoretical lower bound on achievable distance. The C-MIP is the full capacitated MIP using the fixed port set P_{fix}. Finally, the MH variants implement the proposed matheuristic, which applies the C-MIP only to small two-segment subproblems and differs solely in the initialization strategy used to construct the initial station ordering. All experimental configurations are summarized in table 1.

Initialization strategies. The MH experiments consider four alternative constructions of the initial station ordering. The NN (nearest-neighbor) strategy repeatedly selects the closest feasible unvisited station, inserting ports when capacity is exceeded; it is extremely fast but can produce noisy segmentations. The CI (cheapest insertion) strategy grows a tour by inserting at each step the station that causes the smallest marginal increase in the length of tour that results in smoother orderings while remaining lightweight [12]. The GE (greedy edge) strategy follows a Kruskal-like process, adding the shortest admissible edges while avoiding premature subtours before inserting ports [11]; it is often competitive but sometimes less stable than CI. Finally, the OPT initialization derives an optimal no-port directed TSP (via NP-MIP) and then inserts ports to satisfy capacity, giving high-quality orderings at higher computational cost.

MIP and solver parameters. All models are solved with Gurobi 13 [6]. Even with a long time limit ($L_{N\text{seg}} = 7$ days), the full C-MIP on the complete dataset

produces no feasible solution. In contrast, the baseline NP-MIP solves in under 4 hours. The single-segment directed TSP subproblems are solved without explicit time limits (i.e., $L_{1\text{seg}} = \infty$). In practice, these solves are computationally inexpensive, with mean runtime 8.5 s and maximum runtime 88.2 s across 52 solves. The two-segment C-MIP refinement subproblems constitute the primary computational bottleneck of the MH framework. We evaluate a range of limits

$$L_{2\text{seg}} \in \{10, 30, 60, 120, 180, 240, 300, 600\} \text{ seconds},$$

ranging from very short refinement budgets to more computationally intensive runs, to evaluate trade-offs between runtime, solution quality, and robustness.

Matheuristic parameters. Boundary updates consider only the immediate neighbors, although larger neighborhoods would also be possible. After each accepted change, the two affected segments are re-solved via the directed segment-TSP.

6 Results

Baseline initialization quality. Before evaluating MH performance, table 1 reports the baseline distances produced by each initialization strategy. The no-port column reports the underlying unconstrained ordering, while the with-port column reports the corresponding capacity-feasible segmentation. The four strategies yield markedly different starting qualities: NN produces the weakest port-feasible route (5500.29 nm), CI and GE give considerably stronger orderings, and the OPT initialization (obtained by solving the no-port directed TSP) achieves the strongest baseline (4335.18 nm) at the expense of a 3.5 hour preprocessing step. All strategies produce $N = 13$ capacity-feasible segments. These differences frame the improvement potential under MH refinement.

MH performance under two-segment refinement. Table 2 summarizes MH performance across all four initialization strategies and multiple values of $L_{2\text{seg}}$. For each configuration, we report the number of refinement sweeps, total runtime, attempted and accepted boundary changes, station redistribution statistics (total, median, and maximum number of stations moved per pass), and the resulting

Table 1: Variant notation, baseline initialization distances, and runtime.

Notation	Variant	Distance (nm)		Runtime
		no port	with port	
NP-MIP	No-port directed TSP	3552.03	—	3.5 h
C-MIP	Capacity-aware MIP	—	timeout†	1 week
MH-NN	MH with nearest-neighbor	—	5500.29	0.9 min
MH-CI	MH with cheapest-insertion	4848.69	5172.83	1.1 min
MH-GE	MH with greedy-edge	5138.75	4858.97	2.3 min
MH-OPT	MH with NP-based initialization	3552.03*	4335.18	1.5 min

— variant does not entail a corresponding no-port or with-port solution.
* Uses the NP-MIP ordering as initialization.
† No feasible solution found before timeout.

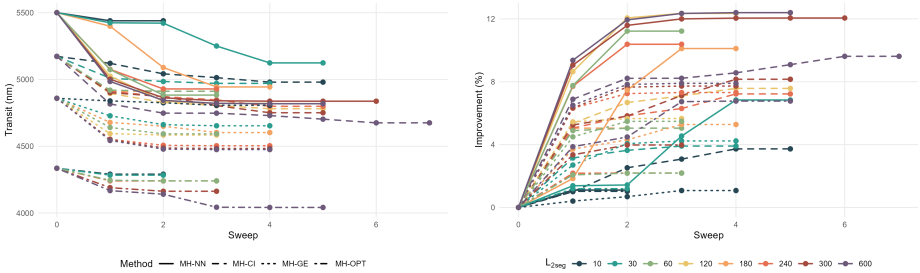
final transit distance together with the absolute and relative improvement from the initial capacity-feasible solution.

Repeated sweeps with segment re-optimization consistently improve the global route beyond the initial segmentation. Increasing $L_{2\text{seg}}$ generally improves solution quality, particularly when moving from very small limits (10–30 s) to moderate limits around 60–180 s. Larger neighborhood limits can occasionally unlock substantially improved solutions, especially for stronger initializations such as OPT, but at considerably higher computational cost.

Convergence and refinement behavior. Figure 2 illustrates refinement progress across successive sweeps for different MH configurations. Across all initialization strategies, most improvement occurs within the first few refinement sweeps, after which the solutions tend to stabilize. Larger values of $L_{2\text{seg}}$ generally produce stronger refinements, particularly for weaker initialization strategies such as NN and CI, although the gains are not strictly monotonic. This suggests that most large-scale structural improvements are identified early, while later sweeps primarily refine segment boundaries locally.

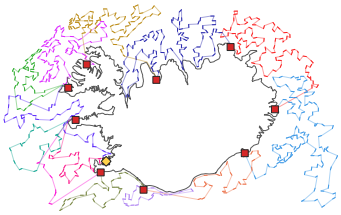
Table 2: MH performance by initialization strategy.

Method	$L_{2\text{seg}}$	Sweeps	Runtime (hr)	Boundary attempts	Boundary changes	$\lvert S\rvert$ moved total	$\lvert S\rvert$ moved med	$\lvert S\rvert$ moved max	Transit nm	Transit Δnm	Transit Δ%
MH-NN	10	2	0.06	17	2	3	2	3	5439.4	61	1.1
MH-NN	30	5	0.29	28	7	125	14	92	5124.1	376	6.8
MH-NN	60	3	0.70	38	17	248	57	191	4883.0	617	11.2
MH-NN	120	4	1.22	37	16	262	28	206	4821.0	679	12.4
MH-NN	180	4	2.03	37	15	254	49	156	4944.0	556	10.1
MH-NN	240	3	2.18	33	11	239	49	190	4929.0	571	10.4
MH-NN	300	6	2.81	41	16	249	14	190	4837.2	663	12.1
MH-NN	600	5	5.63	42	17	261	17	203	4818.4	682	12.4
MH-CI	10	5	0.09	30	7	134	30	51	4980.0	193	3.7
MH-CI	30	4	0.24	27	6	75	14	47	4970.7	202	3.9
MH-CI	60	3	0.51	27	7	50	3	47	4911.8	261	5.0
MH-CI	120	5	1.39	39	16	209	30	90	4780.9	392	7.6
MH-CI	180	2	1.24	23	6	45	22	45	4911.8	261	5.0
MH-CI	240	5	2.47	37	14	143	34	56	4799.1	374	7.2
MH-CI	300	5	3.55	43	19	171	45	58	4750.8	422	8.2
MH-CI	600	7	6.43	43	18	220	12	128	4675.2	498	9.6
MH-GE	10	4	0.08	25	6	88	14	60	4806.7	52	1.1
MH-GE	30	4	0.35	35	12	158	30	98	4653.3	206	4.2
MH-GE	60	3	0.65	32	12	255	86	169	4592.7	266	5.5
MH-GE	120	3	0.95	28	11	137	8	129	4584.1	275	5.7
MH-GE	180	4	1.95	37	15	278	48	183	4602.5	257	5.3
MH-GE	240	4	2.45	35	18	236	14	207	4503.6	355	7.3
MH-GE	300	4	2.87	36	17	264	26	211	4483.6	375	7.7
MH-GE	600	4	5.58	35	17	303	46	212	4475.1	384	7.9
MH-OPT	10	2	0.04	16	2	8	4	8	4290.7	44	1.0
MH-OPT	30	2	0.15	19	4	14	7	14	4283.6	52	1.2
MH-OPT	60	3	0.47	26	8	62	8	54	4240.5	95	2.2
MH-OPT	120	3	0.81	24	7	57	4	53	4240.2	95	2.2
MH-OPT	180	2	1.07	22	6	53	26	53	4240.9	94	2.2
MH-OPT	240	3	1.60	24	7	57	4	53	4240.2	95	2.2
MH-OPT	300	3	2.18	29	12	83	9	74	4162.9	172	4.0
MH-OPT	600	5	6.69	46	21	236	42	117	4041.7	294	6.8

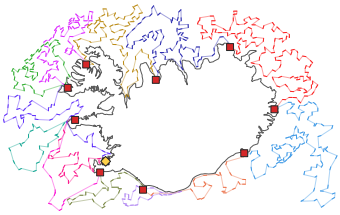


(a) Transit distance (b) Relative improvement

Fig. 2: Refinement progress across sweeps for all MH configurations.



(a) Initial, 4335.18 nm



(b) Refined, 4041.68 nm

	Before			After			Change	
#	$\|S\|$	t	nm	$\|S'\|$	t	nm	$\|S^-\|$	$\|S^+\|$
1	40	41	323	42	44	332	2	4
2	35	44	254	42	45	239	4	11
3	66	44	340	62	41	264	26	22
4	64	45	393	66	44	406	29	31
5	83	39	570	61	41	423	30	8
6	68	45	466	101	45	611	1	34
7	102	45	697	87	43	587	20	5
8	25	43	275	22	44	250	5	2
9	21	34	164	19	31	135	2	0
10	27	39	212	23	33	188	4	0
11	6	40	203	11	45	188	1	6
12	19	45	228	18	44	212	3	2
13	24	25	210	26	29	206	0	2
Σ	580	528	4335	580	528	4042	127	127

(c) Redistribution

Fig. 3: Representative refinement example: (a) segmented capacity-feasible route after port insertion; (b) optimized route; (c) segment-level summary.

Representative refinement behavior. Figure 3 illustrates how relatively localized station redistribution between neighboring segments can produce meaningful global route improvements. In this MH-OPT example, only 127 stations (22% of all stations) are reassigned during refinement, primarily near influential segment boundaries induced by capacity constraints and large catch accumulations.

Best observed performance. Across all runs, the best final plan achieves a final transit distance of 4041.68 nm. Relative to its initial segmented route, this corresponds to a reduction of 293.5 nm (6.8%), equivalent to approximately 24–37 hours saved at realistic steaming speeds (8–12 kn).

7 Discussion

The MH acts primarily as a local improvement operator: it consistently improves all initialization strategies, yet the strongest initializations also produce the best final routes. Weak initializations such as NN admit larger relative gains because they contain more opportunities for beneficial boundary redistribution, whereas stronger initializations require comparatively limited station movement to reach

high-quality final solutions. This behavior is consistent with the restricted neighborhood structure of the method, which focuses on localized segment-boundary refinement rather than large-scale reordering of the global station sequence. Consequently, the search may converge to high-quality local optima rather than globally optimal solutions.

Computationally, the two-segment C-MIP subproblems remain tractable at moderate time limits and consistently produce meaningful improvements under repeated local refinement. The results suggest that very small limits are typically insufficient, while limits around 60–180 s provide a practical balance between runtime and solution quality. Increasing the limit beyond that often yields only marginal additional improvement while substantially increasing runtime.

Operationally, the approach produces fast, reproducible plans suitable for annual survey planning and scenario analysis, improving current deterministic baselines without requiring intractable end-to-end optimization.

However, the experimental scope is intentionally restricted. All experiments are conducted on a single vessel and under deterministic expected catches. This choice reflects the computational difficulty of the problem: even in this reduced setting with $n = 580$ stations, the full C-MIP fails to produce a feasible solution within extended time limits. The single-vessel case therefore serves as the simplest nontrivial setting in which the proposed local refinement mechanism can be isolated and evaluated.

While the problem is related to multi-trip vehicle routing, it does not align directly with standard VRP formulations, and methods from that literature are not immediately applicable in this setting. The goal of this work is therefore not to compete with state-of-the-art VRP solvers, but to assess whether restricted MIP-based local refinement can provide consistent improvements in a setting where exact optimization is computationally intractable. The framework is agnostic to the choice of initial solution: while we consider several simple construction methods, more sophisticated initialization procedures could equally be used as inputs and subsequently improved through the same refinement mechanism.

8 Conclusions

This paper introduced a matheuristic for capacitated groundfish survey planning based on repeated two-segment MIP refinements of an initial segmented route. Mandatory port returns and direction-dependent distances substantially increase the computational difficulty of large-scale survey planning, and the full capacitated MIP remains computationally intractable at operational scale.

The proposed MH approach consistently improves capacity-feasible routes across multiple initialization strategies while maintaining practical runtimes. Overall, the results demonstrate that restricted local MIP refinement provides a practical and scalable mechanism for constructing high-quality, reproducible survey plans suitable for operational use, strategic planning, and scenario comparison.

9 Future Work

Neighborhood limitations. The current MH optimizes repeated two-segment neighborhoods, limiting global exploration and preventing guarantees of global optimality. Future work should investigate larger neighborhoods, including occasional three-segment refinements, periodic resegmentation or "shake" phases, and broader controlled station transfers across segment boundaries.

Time-based objectives. Incorporating time-based considerations, such as station service times and vessel endurance constraints (time at sea), is a natural extension. These factors shift the objective beyond distance minimization toward minimizing completion time (e.g., makespan) or balancing route duration under capacity and endurance limits.

Multi-vessel extension. Extending the method to a multi-vessel setting is a key direction. In this setting, several boats cooperatively service the station set, leading to a two-level structure: a global assignment phase that partitions stations across vessels, followed by local refinement within each vessel's route using the same two-segment MIP improvements. Boundary refinements can be extended to allow controlled transfers of stations between vessels, enabling coordinated improvements across routes and accommodating heterogeneous fleets.

Uncertainty and robustness. Uncertainty is a practical concern. Sailing times, service times, and realized catches vary substantially year to year, and solutions optimized for expected catches may degrade in feasibility or require additional port visits under adverse realizations, increasing total travel distance and operational time. A straightforward next step is to assess feasibility and performance via empirical sampling from historical distributions, enabling sensitivity analysis and favoring solutions that remain feasible with high probability and exhibit robust performance under variability.

Availability. The complete implementation of this study and the 2023 survey data are openly available at `https://github.com/HI-IDN/groundfish-singlevessel-routing`.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Boschetti, M.A., Letchford, A.N., Maniezzo, V.: Matheuristics: survey and synthesis. International Transactions in Operational Research **30**(6), 2840–2866 (Nov 2023). `https://doi.org/10.1111/itor.13301`
2. Boschetti, M.A., Maniezzo, V.: Matheuristics: using mathematics for heuristic design. 4OR **20**(2), 173–208 (Jun 2022). `https://doi.org/10.1007/s10288-022-00510-8`
3. David L. Applegate, Robert E. Bixby, Vašek Chvátal, William J. Cook: The Traveling Salesman Problem: A Computational Study. Princeton University Press (2011)

4. Dell'Amico, M., Montemanni, R., Novellani, S.: Matheuristic algorithms for the parallel drone scheduling traveling salesman problem. Annals of Operations Research **289**(2), 211–226 (Jun 2020). https://doi.org/10.1007/s10479-020-03562-3
5. Fan, J., Zhang, C., Yang, F., Shen, W., Gao, L.: A matheuristic with re-lot-sizing strategies for flexible job-shop rescheduling problem with lot-streaming and machine reconfigurations. European Journal of Operational Research **319**(3), 747–762 (Dec 2024). https://doi.org/10.1016/j.ejor.2024.07.030
6. Gurobi Optimization, LLC: Gurobi Optimization, https://www.gurobi.com/
7. Ingibjörg G. Jónsdóttir, Hjalti Karlsson, Höskuldur Björnsson, Jón Sólmundsson, Klara B. Jakobsdótir, Valur Bogason: A manual for the Icelandic Groundfish Survey in spring 2025. Kver KV 2025-02, Marine and Freshwater Research Institute, Hafnarfjörður, Iceland (Feb 2025), https://www.hafogvatn.is/static/research/files/rallhandbok_2025-enska-1.pdf
8. Jónsdóttir, I.G., Karlsson, H., Pétursson, H., Björnsson, H., Sólmundsson, J., Bogason, V.: Stofnmælingar botnfiska á Íslandsmiðum 2023: Framkvæmd og helstu niðurstöður. Tech. Rep. HV 2023-12, Hafrannsóknastofnun, Hafnarfjörður, Iceland (Mar 2023), https://www.hafogvatn.is/static/research/files/hv_2023_12.pdf
9. Jónsson, M., Rúnarsson, T.P., Steinarsson, B.: Optimizing the Sailing Route for Fixed Groundfish Survey Stations. International Council for the Exploration of the Sea (ICES) (1996)
10. Kimura, Daniel K and Somerton, David A: Review of Statistical Aspects of Survey Sampling for Marine Fisheries. Reviews in Fisheries Science & Aquaculture **14**(3), 245–283 (2006)
11. Kruskal, J.B.: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proceedings of the American Mathematical Society **7**(1), 48–50 (1956). https://doi.org/10.2307/2033241
12. Rosenkrantz, D.J., Stearns, R.E., Lewis, II, P.M.: An Analysis of Several Heuristics for the Traveling Salesman Problem. SIAM Journal on Computing **6**(3), 563–581 (1977). https://doi.org/10.1137/0206041
13. Rúnarsson, T.P.: Interactive Evolutionary Reoptimization for Groundfish Survey Planning. In: Singh, H., Ray, T., Knowles, J., Li, X., Branke, J., Wang, B., Oyama, A. (eds.) Evolutionary Multi-Criterion Optimization (EMO 2025). pp. 275–287. Springer Nature, Singapore (2025). https://doi.org/10.1007/978-981-96-3506-1_19