

FACTS › SINTEF Ocean › May 2026

# Offshore Maritime Optimization Logistics Workbench

*Cloud-based decision support for offshore supply vessel routes and schedules.*

[Open the Workbench](#)

## Periodic and operational planning, on the same platform

The **Offshore Maritime Optimization Logistics Workbench** is a SINTEF Ocean web application - funded by the [LowEmission Research Centre](#) and developed through a long collaboration with Equinor - that turns periodic and operational supply-vessel planning into a single workflow managed from browser. The tactical engine, MOLO (Maritime Offshore Logistics Optimisation), was developed jointly with Equinor; the operational engine, LogOps (Logistics Optimisation for Operational planning), was scoped from requirements gathered during the same collaboration. Two solver engines run side by side under one shell, each with its own scenario template - similar in structure but tuned to different planning horizons:

- **Tactical (MOLO engine)** – periodic supply vessel planning. Construct a weekly route plan that minimises fleet cost while satisfying installation visit frequencies, voyage duration limits and PSV-installation compatibility, using the Adaptive Large Neighbourhood Search (ALNS) metaheuristic of [Kisialiou et al. \(2018\)](#).
- **Operational (LogOps engine)** – short horizon, weather-aware daily routing. Mandatory and optional cargo, recursive per-leg speed optimisation, fuel curves per operating mode, live met.no weather.

**Roadmap:** In addition, the workbench is set up for facilitating resource sharing across operators. This sits alongside the two production engines under the same shell, one module for tactical and one for operational planning, each with its own scenario template tuned to what the module needs.



*Fig. 1 — Dashboard. Two production workflows (Tactical / MOLO and Operational / LogOps) and two roadmap modules, all sharing the same shell.*

## Cloud architecture

A permanent API service on Azure App Service is the single-entry point for the browser - it authenticates requests, persists scenarios, vessels, installations and solutions in a managed database, reads vessel power model files from an Azure Files share, and orchestrates an optimisation worker created individually for each user. Authentication is handled by ASP.NET Identity with three roles: Owner, Admin and User.

All heavy optimisation runs in a user worker - an Azure Container Instance with 4 vCPU and 16 GB RAM, provisioned the moment the user opens the platform and torn down automatically after idle. The worker shares the same database and Azure Files mount as the API, runs the MOLO and LogOps engines, and writes results back where the API can serve them - all while the browser polls live status. Cost is therefore proportional to actual use, not to the number of logged-in users.

Two external services close the loop. Live weather forecasts (waves, wind, currents) are pulled from the [met.no](#) ocean-forecast API per installation. Vessel power and fuel are evaluated on the fly against a local surrogate model stored on Azure Files -





Fig. 4 — Tactical-engine pipeline.

## Operational engine — LogOps

LogOps targets the daily question: "given the fleet that is actually available tomorrow, the cargo waiting on the quay, and the weather expected over the next 48–72 hours, what is the cheapest, safest set of voyages?". The planning horizon is a single departure day, but the engine is materially different from the tactical one: leg speed becomes a decision variable rather than a constant, and fuel consumption is computed by integrating ship power across each leg rather than from per-time rates.

- **Energy from a ShipX-derived power model.** Required power at any (speed, heading, sea-state) is read at runtime from a surrogate model in the form of interpolation file built from [ShipX](#) evaluations;
- **Per-leg energy and fuel.** Leg cost is the integral of power over leg time (kWh) and is then converted to fuel cost via per-fuel-type energy coefficients - MGO, LNG, dual-fuel and hybrid are all dispatched separately. Cost evolves smoothly with leg speed and the local sea state.
- **Speed optimisation.** When the OptimizeSpeeds flag is on, leg speeds are tuned by a recursive procedure that aims for the latest legal arrival at

the base and accelerates only when an installation time-window forces an earlier ETA; speeds are clamped to vessel min/max.

- **Mandatory + optional cargo.** Voyages take all mandatory deliveries and pickups first, then fill remaining capacity with optional cargo. Two objective modes are exposed: cost-only, or lexicographic - maximise optional cargo first, then minimise cost.
- **Weather.** Wave height, wave period, wind and ocean current vectors are pulled from met.no per installation and interpolated along each leg by the vessel's relative position. The power model is re-evaluated whenever the local sea state changes, so leg cost reflects the actual sea conditions encountered.

## Bases

Each scenario starts from one onshore supply base. The Base settings page captures names, latitude/longitude, the offshore point at which the PSV reaches open water, daily opening / closing times, the turnaround at the quay, and any extra-location sailing time. A modal dialog sets the per-weekday minimum and maximum number of departures from the base.

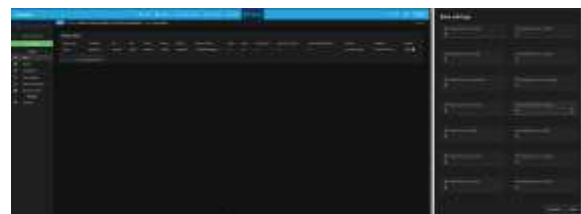


Fig. 7 - Base settings. Left: the chosen base with its offshore point and turnaround time. Right: the per-weekday min/max departure-capacity dialog.

## Vessels

Each scenario draws on a vessel model/type library. A vessel type carries the capacity, design speed (or sailing-speed bounds when speed optimisation is enabled), time-charter cost per hour, fuel type (MGO, LNG, ...), consumption rates for each fuel, and the number of available vessels. Fuel costs and per-fuel emission factors (CO<sub>2</sub>, NO<sub>x</sub>, SO<sub>x</sub>, PM) are configured globally on the Fuel-settings page.

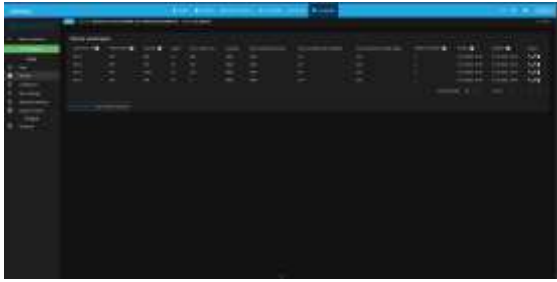


Fig. 8 - Vessel settings.

## Installations

Installations are added to a scenario from the central Installations archive (see below). Each scenario record carries names, coordinates, daily opening hours, weekly visit frequency, laytime per cargo unit, weekly demand, demand-intensive flag and use-in-optimisation toggle. The LogOps view extends the same record with mandatory and optional cargo in/out and a service date window.

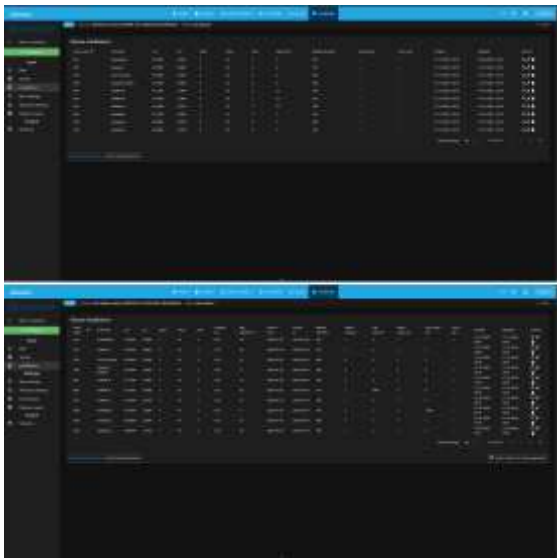


Fig. 9 - Installations. Top: the tactical (MOLO) installations table — opening hours, frequency, weekly demand, demand-intensive flag. Bottom: the operational (LogOps) view - mandatory and optional cargo in/out, service date window, daily laytime.

## Scenario settings

Solver knobs are grouped into four sub-pages to manage settings on different levels:

- **Basic.** Maximum running time and the choice between auto-generated and manual distance matrix.
- **Voyage.** Per-voyage min/max installations, max demand-intensive installations, and maximum voyage durations split between weekdays and weekends.
- **Fuel.** Per-fuel cost (NOK/ton) and per-fuel emission factors for CO<sub>2</sub>, NO<sub>x</sub>, SO<sub>x</sub> and PM.

- **Advanced.** Constraints on evenly spread departures, min/max days between visits, discrete departure times, vessel–installation compatibility matrix, fixed installation departure-day requirement, voyage slack.



Fig. 10 — Basic and Voyage settings.

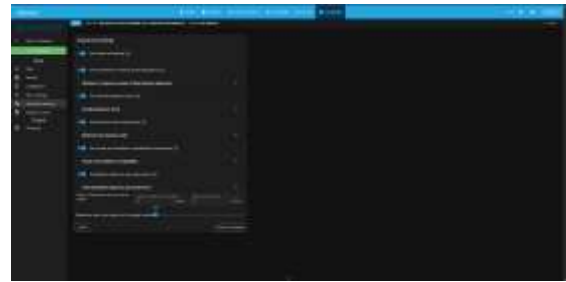


Fig. 11 — Advanced settings.

## Offshore installations archive

Across scenarios the platform exposes an installations archive that planners populate manually - adding new offshore installations one by one with their coordinates, opening hours, demand profile and other attributes. The archive is rendered on an interactive Leaflet map of the North and Norwegian Seas; planners pick the relevant subset into each scenario.



Fig. 12 — Installations archive. Installations rendered on an interactive Leaflet map of the North and Norwegian Seas.

## Outputs and KPIs

Every optimisation run lands in the scenario's Solutions page, which lists currently-running and finished runs side by side with the full KPI vector: created timestamp, status, run time, used vessels, number of voyages, total / voyage / vessel-charter cost (NOK), deviation from the input plan, weighted objective, sailing distance (nm), slack and waiting hours, and CO<sub>2</sub> / NO<sub>x</sub> / SO<sub>x</sub> / PM emissions in kilograms. Every row carries a quick-action toolbar

— view, clone, export to JSON, import to manual mode, delete — so the planner can pick a recent run, copy it as the seed for a what-if, or hand it to colleagues.

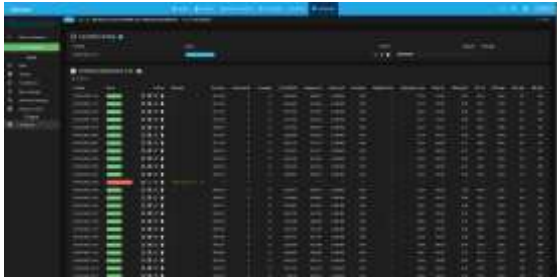


Fig. 13 - Solutions page. Currently running and finished runs with the full KPI vector and per-row quick actions.

Clicking a finished run from the Solutions page opens its output view. Both engines deliver this view in the same shape — a route map, a Gantt schedule of every PSV across the planning horizon, and per-day or per-leg detail. The screenshot below is from the tactical engine; the operational view is structurally identical.



Fig. 14 - Engine output (drilled down from a Solutions-page row). Route map (top), Gantt schedule of every PSV across the planning horizon (middle), and per-day vessel cards listing departures and installation arrivals (bottom). The operational engine renders the same three panels with per-leg detail in place of the daily cards.

## Vessel power model

Required power and energy originate in [ShipX](#), SINTEF Ocean's hydrodynamic workbench. To make exhaustive search feasible, the platform uses an interpolation surrogate generated from many [ShipX](#) evaluations across combinations of wind, wave and current state, vessel speed and heading. At runtime, leg power is computed on the fly by binary recursive interpolation against this look-up table — orders of magnitude faster than calling [ShipX](#) directly. The interpolation file lives on Azure Files and is loaded once per worker;

## Open the Workbench

The platform is live at [Workbench](#). Sign in with the credentials provided by your administrator.

## Authentication and administration

ASP.NET Identity backs every request with three roles — Owner, Admin and User. Admin-only views expose the archived bases, installations and vessels, and a vessel–installation compatibility matrix per scenario.

## Roadmap

Two further modules are on the roadmap: Resource sharing — identifying opportunities to share vessel capacity across operators to reduce idle time and cost — and Emission estimation — fleet-wide CO<sub>2</sub> / NO<sub>x</sub> / SO<sub>x</sub> / PM accounting for planned voyages.

## Contact

**Yauheni Kisialiou, PhD** — Research Scientist, SINTEF Ocean (Bergen)

[yauheni.kisialiou@sintef.no](mailto:yauheni.kisialiou@sintef.no) · Phone: +47-90-25-06 10 · [www.sintef.no/programvare](http://www.sintef.no/programvare)