CASE STUDY

Scheduling Shipments of Goods and Fuels to Oil Rigs

MemComputing

Learn how MemComputing provides one of the world's largest Oil and Gas companies an optimal scheduling solution for the maritime delivery of goods and fuel to, from, and between offshore facilities. Our solution addresses the problem at scale and significantly improves operational efficiencies and cost savings.

Abstract

This case study was a joint effort between one of the world's largest oil and gas companies and MemComputing. It examines the optimization of a complex scheduling problem for the maritime transportation of cargo to offshore facilities, such as offshore platforms and drilling rigs. The problem involves balancing the delivery of goods and fuel from an onshore dock location to various offshore facilities within a 30-day period. Using Integer Linear Programming (ILP) and data from the customers benchmark, a mathematical model was built to represent the problem. The model was run on a leading commercial solver and the MEMCPU Platform. The problem proved to be intractable for the commercial solver, while the MEMCPU Platform delivered highly optimized solutions in under an hour. The results of the optimization showed a 16% increase in the delivery of goods, a 42% reduction in the number of required ships, and a 48% reduction in the number of overall transits. These optimized routes demonstrated the potential for $1.5M in monthly cost savings and a 18 kt reduction in carbon emissions. This paper highlights the potential of the MEMCPU Platform in addressing complex scheduling problems while delivering tangible benefits.

Table of Contents

1. Introduction

1.1 Customer

The customer is a large Integrated Energy Company with a significant hydrocarbon production portfolio. The company has operations and projects in several countries around the world, including the United States, the United Kingdom, Azerbaijan, and Trinidad, among others.

They operate several offshore production platforms and drilling rigs and uses advanced technologies to ensure the safe and efficient extraction of hydrocarbons. The company also has a strong focus on safety and environmental concerns and is committed to minimizing the impact of its offshore hydrocarbon operations on the ocean and coastal environments.

1.2 The Memcomputing Technology

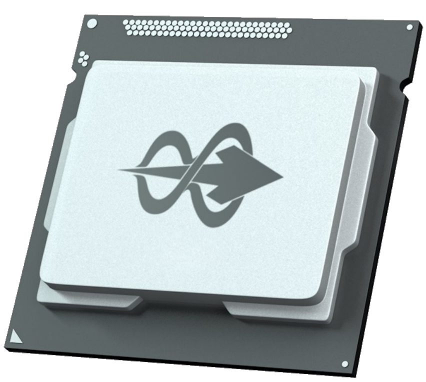
Memcomputing Self-Organizing Gates (SOGs) are a patented computing paradigm that represents a novel approach to performing information processing and computation. It was invented by Dr. Fabio L. Traversa and Dr. Max Di Ventra at UC San Diego in 2015 [1].

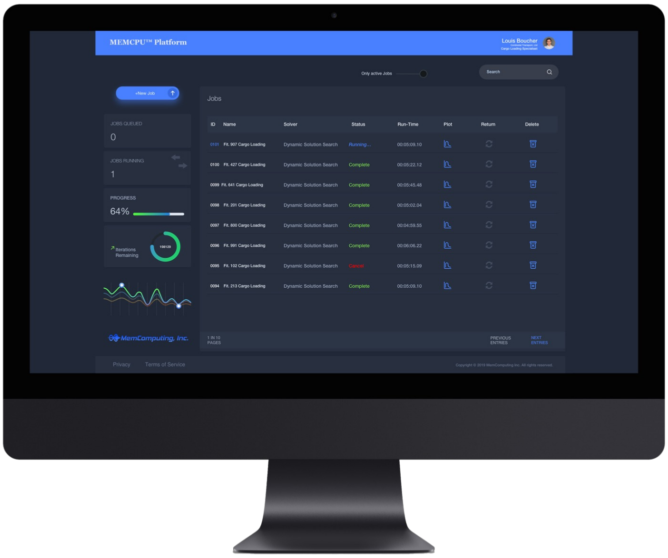
The SOGs are a collection of discrete physical elements that store and manipulate data in a manner similar to that of conventional computer memory, but they also possess the ability to perform computation. The paradigm shift introduced is that the SOGs are connected and form a network to communicate among themselves while computing therefore working in an ultra-high parallelized fashion to solve the problem. The outcome depends on both the information stored in the SOGs and the information exchanged with other SOGs in the circuit. This enables a collective type of computation that is somewhat similar to how the human brain operates. Similar to neurons in the human brain each SOG responds to changes in other SOGs while processing because they are inter-connected. This connectivity creates correlations among SOGs that can span across the entire network. The memcomputing circuit can establish correlations between SOGs at scale (easily 100s of millions of SOGs in a circuit).

1.3 The MEMCPU™ Platform

The cloud-based MEMCPU Platform, released in 2019, is a virtual computing solution for solving complex optimization problems in areas such as logistics, scheduling, resource allocation, and more. It is designed to deliver faster, more efficient solutions compared to traditional optimization methods such as mathematical programming and heuristics.

The MEMCPU Platform uses a combination of hardware and software technologies, including memcomputing, to solve complex problems in real-time. The technology is designed to work with existing computing systems. Problems are defined in a mathematical model formulated using Integer Linear Programming (ILP). The ILP model is sent to the MEMCPU Platform where the circuit configures itself, similar to an FPGA, to exactly match the ILP. Reconfiguration takes nanoseconds, and then the SOGs begin simulating the dynamics of the ILP model. What this means is that the MEMCPU Platform leverages the collective nature of the SOGs, solving the problem in a single MEMCPU Cycle. The MEMCPU Platform is a cutting-edge technology for solving complex optimization problems in a variety of industries. Its ability to deliver faster and more efficient solutions compared to traditional methods makes it a valuable tool for organizations looking to improve their decision-making processes and optimize their operations.





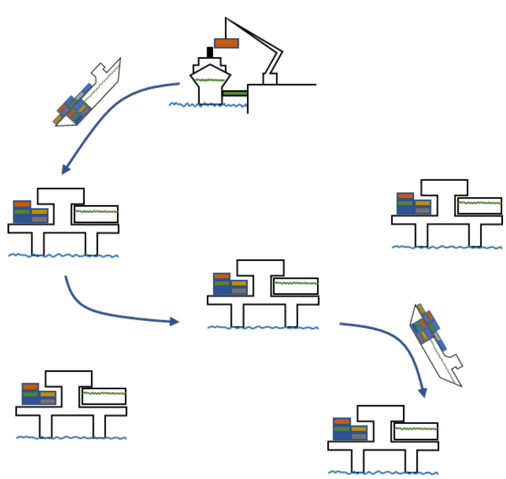
2. The Problem Defined

2.1 Overview

Enhancing the efficiency of logistics operations leads to improved safety, reduced operating costs, and a decrease in carbon emissions. As part of their commitment to becoming carbon neutral, the company's logistics team is continuously working to optimize schedules and minimize risk.

In this case study, we examine the maritime transportation of cargo to offshore platforms and rigs in the Gulf of Mexico, which require a constant supply of goods and resources to sustain their operations. These deliveries are scheduled in advance to ensure that the offshore facilities have the necessary supplies at the right time. However, the complexity and large scale of these operations, coupled with critical time windows, make scheduling a challenging task for which no automation exists today. This process is currently performed manually by experienced scheduling personnel.

- These offshore platforms require a variety of consumable goods and services. These include;
- Large quantities of fuel to power the offshore facility and its systems.
- Lubricants to maintain various machinery and equipment.
- A variety of chemicals, including muds, inhibitors, polymers, surfactants, and biocides.
- Large quantities of water for drilling, cooling, and injecting into wells to maintain pressure.
- Large quantities of cement used to stabilize wells and prevent leaks.
- Drilling supplies like drill bits and drilling pipes.
- Spare parts, and other supplies.



2.2 Manual Scheduling Process

Optimize cargo delivery to offshore facilities in the Gulf of Mexico and pick up and return cargo to the port. Vendors provide delivery dates when certain shipments will be available at the port and identify the intended offshore facility. Schedulers work with the offshore facilities to determine the desired delivery date of the cargo required by the various offshore facilities. From this, the schedulers assign which ships, from a fleet of "slow" and "fast" ships (see sidebar), visit which offshore facilities and then assign the cargo accordingly. The use case for this problem was based on actual schedules from a previous month. The most challenging part of this problem is scheduling the cargo shipment. There are 3500 cargo items which must go to 15 offshore facilities, and each item has a specific delivery time limit. The goal is to find the right combination of ships to minimize the number of transits; thus, the complexity of this problem grows exponentially with the number of cargo items.

Slow Ships are the workhorses that deliver the vast majority of the cargo. Fast Ships are used for ad hoc deliveries when something might be needed as soon as possible, on emergency, or just additional coverage of cargo.

2.3 Constraints

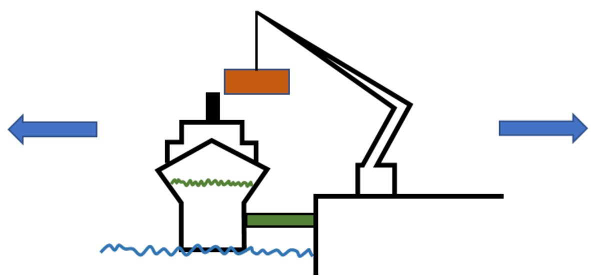
Constraints represent restrictions that the scheduler must adhere to for a successful schedule.

These constraints fall into the five following categories.

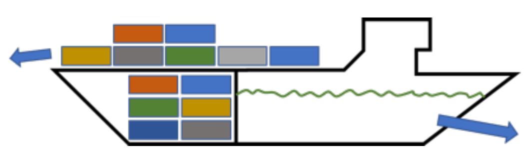
- At the dock
- On the ships
- At offshore facilities
- Route constraints
- Dynamic constraints

The following subsections detail the constraints.

2.3.1 Constraints at the dock

<u>Cargo</u> Cargo is available within a specified window of time. (Resolution of a few hours)		<u>Fuel</u> Available fuel at the dock is "infinite." Fuel is uploaded to the ship, never offloaded.

2.3.2 Constraints on the ships

<u>Cargo</u> Total allowed weight and volume are independent of the fuel load. Total cargo weight must be less than or equal to the Max cargo weight. Total cargo volume must be less than or equal to the Max cargo volume. Cargo will ship from the dock to the offshore facilities and vice-versa. Cargo will also ship between the offshore facilities. Not more than 50% of the deck space can be used on any of the vessels		<u>Fuel</u> The total allowed fuel weight and volume are independent of the cargo. Total fuel weight must be less than or equal to the Max fuel weight. Total fuel volume must be less than or equal to the Max fuel volume. Fuel only ships from the dock to the offshore facilities.

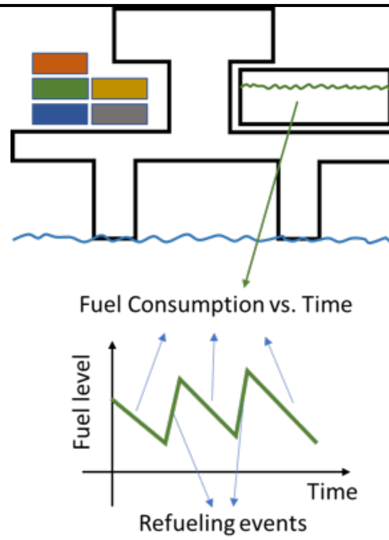
2.3.3 Constraints at the offshore facilities

Cargo

Cargo must arrive within a prescheduled time window.

(Resolution to a few hours)

Offshore facilities can only accept shipments when the weather is suitable, and the dock is within a reasonable transport distance.



Fuel

Fuel consumption occurs at a constant rate.

The fuel level must remain above the Min fuel level.

The fuel level cannot exceed the Max fuel level.

Fuel consumption can be affected by the weather.

2.3.4 Constraints on the routes

Some ships cannot reach all offshore facilities.

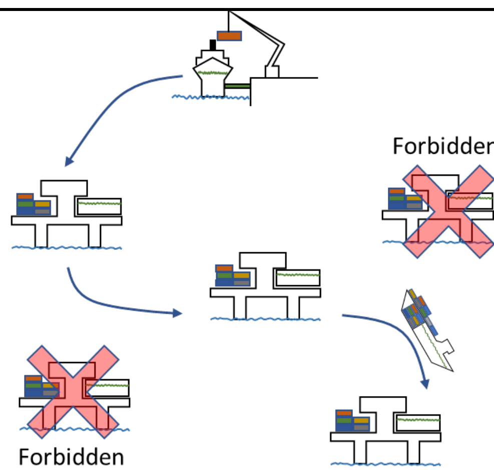
Some offshore facilities can be forbidden for a window of time due to the weather.

The time to go from one offshore facility to another is independent of the weather.

Ships can remain at the dock indefinitely.

Ships at the docks remain for a minimum time (fixed operations and maintenance) plus the time proportional to the lifts.

Ships can remain at an offshore facility for a minimum time (fixed operations) plus the time proportional to the lifts.



There is only one port/dock for this case study.

(More docks can be added.)

A ship at a dock has a cost per day.

A ship at an offshore facility has no associated cost.

2.3.5 Dynamic constraints

Unexpected events and emergencies provide additional challenges, for example, weather, maintenance, crew availability, etc. MemComputing analyzed one year's worth of data to understand the average impact of these events. Since the impact of these events during the 30-day period studied was less than the average, the stricter percentages in the table below were included in the model.

Average Wait Times Per Ship	At Fourchon	% at Fourchon	Offshore Facilities	% at Offshore Facilities
Maintenance	2.25	0%	-	
Maneuvering	216.03	9%	221.32	25%
Standby	-		108.50	12%
Tank Cleaning	155.20	7%	-	
Waiting	1,888.64	83%	477.07	53%
Crew Not Avail	17.56	1%	4.12	0%
Weather	-		91.54	10%
Avg. Wait Times Per Ship (mins)	**2,279.68**	**100%**	**902.55**	**100%**
Avg. Wait Times Per Ship (Hours)	**37.99**		**15.04**	

3. Combinatorial Nature

3.1 Relevant NP-Hard Problems

3.1.1 The Travelling Salesperson Problem

The Traveling Salesperson Problem (TSP) is a classical combinatorial optimization problem that involves finding the shortest possible route that visits a given set of cities and returns to the starting city. The problem can be formalized as follows: given a set of cities and the distances between each pair of cities, find the shortest possible route that visits each city exactly once and returns to the starting city.

TSP is one of the most well-known and well-studied problems in the field of combinatorial optimization. It is a NP-hard problem, meaning that finding an exact solution in a reasonable amount of time is computationally intractable for large instances of the problem. As a result, approximate solutions or heuristics are often used to find good solutions in practice.

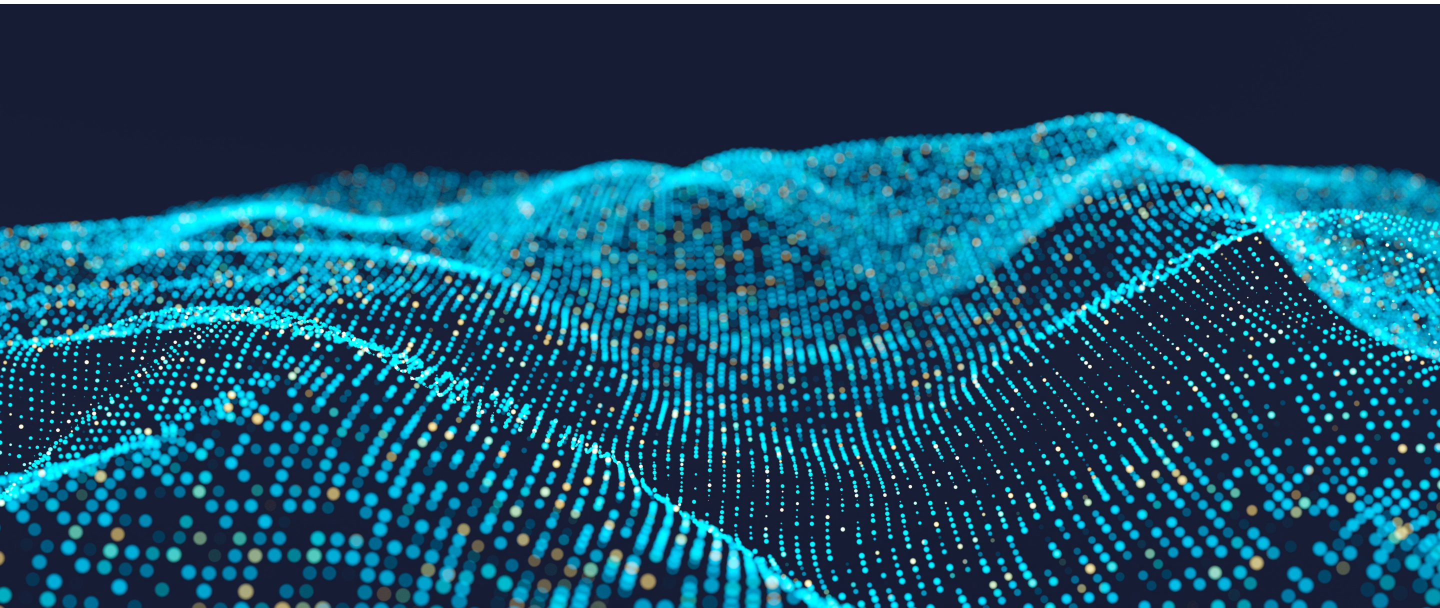
3.1.2 The Bin Packing Problem

The bin packing problem is another classic combinatorial NP-hard optimization problem. The problem consists of allocating items of different sizes and weights into bins where the number of bins used must be minimized and cannot exceed the capacity of each of them. Many variants of this problem are deeply studied, and the literature provides a large choice of exact and approximation algorithms specialized for this problem.

3.1.3 The Scheduling Problem

Scheduling is a general combinatorial problem that includes several NP-hard problems such as interval scheduling, job shop scheduling, etc. A generic scheduling problem in industrial applications typically represents a challenge where even approximations are hard to find in the available time. For example, rescheduling flights due to weather conditions is a huge challenge for airline companies.

These problems have many real-world applications, including vehicle routing, logistics, scheduling, planning, and resource allocation. These problems are extensively studied, and research fields are devoted to working on algorithms and heuristics developed to solve them.



3.2 Relating the Case Study To The NP-Hard Problems

Consider the problem addressed here. Multiple ships must visit multiple offshore facilities, which is a multi-TSP problem. Add that a collection of goods must be delivered to these facilities, with a specific availability date on the port and "no-later than" delivery date for goods to an offshore facility. The goods and fuel are destined for different offshore facilities. For fuel, the starting fuel level, the minimum, and maximum fuel levels, and the consumption rate for each offshore facility must accounted for in the model. Additionally, the cargo and fuel must be packed onto several ships; therefore, the bin packing problem is also represented.

All these problems need to be solved at once since they share variables. This makes this cargo delivery problem far more complex than an individual NP-hard problem.

A solution to the problem at the scale required by the customer is intractable for even the best-in-class commercial optimization problem solver (as shown later). To solve this problem using traditional methods, the problem must be broken up, and some constraints or variables must be ignored. A solution can be provided, but it will be an approximation. In many cases, such an approximation may not be better than a random solution.

The MEMCPU Platform can address the entire problem at the required scale. To understand how and why MemComputing is different than these other methods and why it is not affected by the exponential nature of combinatorial optimization problems, view our video that compares MemComputing to Branch and Bound [2]. Further reference material can be found at the end of this document [3-7].

3.3 Conventional Approach: Branch and Bound

Leading edge commercial optimization solvers use Branch and Bound, a general algorithmic framework for solving combinatorial optimization problems. It is also used for problems where finding an exact solution is computationally intractable but finding an approximation or a good upper bound on the optimal solution is possible.

The basic idea of Branch and Bound is to divide the problem into smaller subproblems and then solve each subproblem separately. The algorithm works by exploring the solution space in a systematic way and eliminating parts of the solution space that are not feasible. At each step, the algorithm decides which branch to explore next, and it keeps track of the best solution found so far.

The "bound" in Branch and Bound refers to the best solution that has been found so far, and it provides an upper bound on the optimal solution. This bound is used to prune the search space and to eliminate branches that are unlikely to contain the optimal solution. This makes the algorithm more efficient and helps to avoid exploring parts of the solution space that are not relevant.

Branch and Bound is widely used in practice, and it is especially well suited for problems where the solution space is large and complex. It is also used for problems with multiple constraints, as the algorithm can be easily adapted to handle a wide range of different types of constraints. However, the efficiency of the algorithm depends on the quality of the bound and the efficiency of the branching strategy, and it can be difficult to find a good solution in some cases.

3.4 Solutions: Approximate vs Optimal

An optimal solution is the best possible solution to a problem, while an approximate solution is any feasible solution (i.e., satisfying all constraints) that can be close to, but not, the optimal solution. In combinatorial optimization problems, finding an optimal solution is often computationally intractable for current computers, due to the exponential growth of the number of possible combinations as the size of the problem increases. As a result, finding an approximate solution, which can be far from the optimal, is a common outcome for these solvers.

The quality of an approximate solution is usually measured by how close it is to the optimal solution, either in terms of the objective function being optimized or in terms of other performance metrics. A good approximate solution should provide a solution that is close to the optimal solution, with a bounded error or with a high probability of being close to the optimal solution.

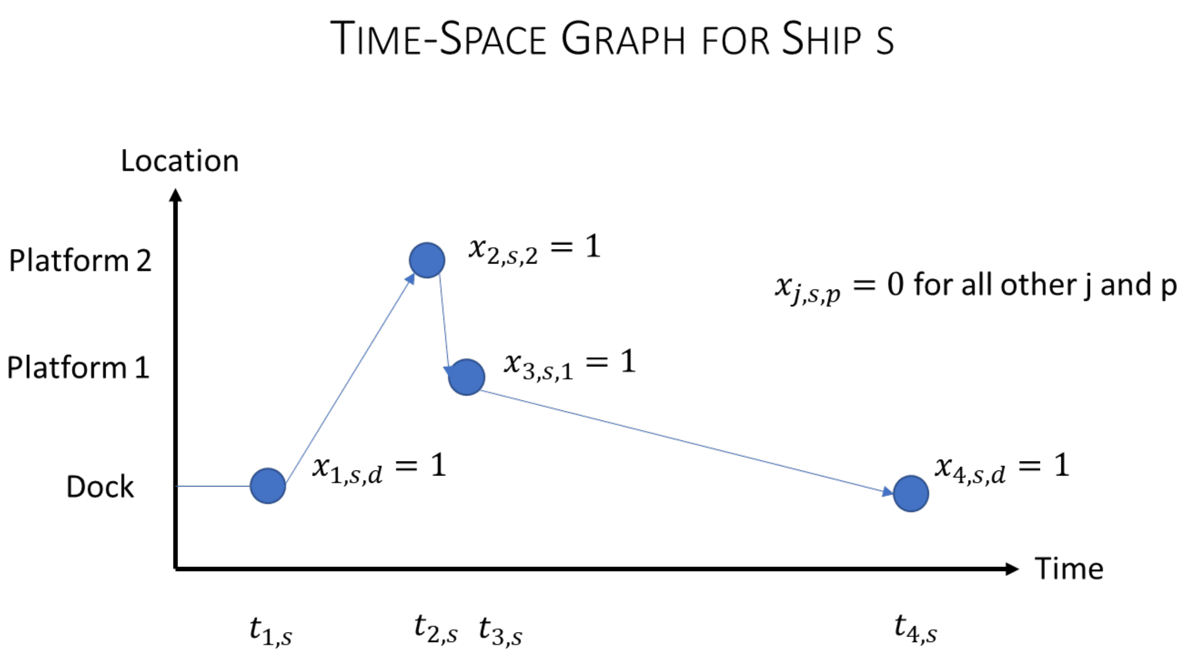
Poor approximate solutions lead to inefficiencies in effort, use of resources, and increased costs. Attempts to solve logistics problems are well documented. For example, Parabas conducted a study for optimizing helicopter transport to, from, and between oil rigs [8] and MemComputing studied the same problem showing how MemComputing provides a near optimal solution [9]. (Additional examples can be found online [7].)

4. Developing the Mathematical Model

Building the mathematical model begins with identifying the "backbone" or core of the problem. This is done by analyzing the combinatorial nature of the problem and determining the set of decision variables, objectives, and related constraints involved. This was a collaborative process between the MemComputing and customer teams.

4.1 Space-Time Graph

We start by developing a Space-Time Graph for a given ship, S. This helps us understand the position of the vessels over time. All ships begin at the dock, visit one or more facilities and return to the dock.



4.2 Variables

We then identify the main variables for the mathematical model.

Offshore Facility variables

- $t_{j,p}$ time of event j of offshore facility p
- $v_{j,p}$ level of fuel at instant j for the offshore facility p
- $\Delta v_{j,p}$ fuel uploaded at instant j to the offshore facility p
- $\Delta g^{-}_{p(i),j,s}$ binary variable defining the unload of the good i on the event time j from the ship s to the offshore facility p
- $\Delta g^{+}_{p(i),j,s}$ binary variable defining the load of the good i on the event time j from the offshore facility p to the ship s

Ship variables

- $t_{j,s}$ time of event j for ship s
- $v_{j,s}$ level of fuel at instant j for the ship s
- $g_{i,j,s}$ binary variable allocating the good i at the event time j on the ship s
- $x_{j,s,p}$ binary variable indicating at event time j the ship s is at the offshore facility p or the dock $d = p$

Dock variables

- $t_{j,s,d}$ time of event j of dock d for the ship s
- $\Delta g^{-}_{d(i),j,s}$ binary variable defining the unload of the good i on the event time j from the ship s to the dock d
- $\Delta g^{+}_{d(i),j,s}$ binary variable defining the load of the good i on the event time j from the offshore facility p to the dock d

4.3 Constraints

4.3.1 Loading and Unloading of Fuel

4.3.1.1 Loading and Unloading of Fuel From a Ship

This section details the constraints that must be satisfied when loading and unloading fuel to or from any cargo ship.

(1) Upper bound of the fuel level at the time j for the ship s.

$$v_{j,s} \leq v_{j-1,s} - \sum_{p} x_{j,s,p} \Delta v_{j,p} + \sum_{d} x_{j,s,d} v_{s\,max}$$

(2) Lower bound of the fuel level at the time j for the ship s.

$$v_{j,s} \geq v_{j-1,s} - \sum_p x_{j,s,p} \Delta v_{j,p}$$

(3) Reset the fuel to the max fuel if the ship s stops at the dock d.

$$v_{j,s} \geq \sum_d x_{j,s,d} v_{s\,max}$$

(4) Initial condition for the fuel level of the ship s

$$v_{0,s} = \bar{v}_{0,s}$$

(5) Fuel limits for the ship s

$$v_{s\,min} \leq v_{j,s} \leq v_{s\,max}$$

4.3.1.2 Unloading fuel to an offshore facility and fuel consumption

The cargo ships carry enough fuel to refuel the offshore facilities it visits. We must track the fuel consumption of the facility. We must ensure that the facility's fuel level does not fall below the minimum and stays within the maximum.

(6) Continuity equation for the fuel of the offshore facility p at time t with consumption rate k_p

$$v_{j,p} = v_{j-1,p} - k_p\left(t_{j,p} - t_{j-1,p}\right) + \sum_s x_{j,s,p} \Delta v_{j,p}$$

(7) Consistency of consumption and minimum fuel level of the offshore facility p at time t with consumption rate k_p

$$v_{j-1,p} - k_p\left(t_{j,p} - t_{j-1,p}\right) \geq v_{p\,min}$$

(8) Initial condition for the fuel level at the offshore facility p

$$v_{0,p} = \bar{v}_{0,p}$$

(9) Condition for the fuel level at the offshore facility p at the end of the scheduling period

$$v_{end,p} \geq \bar{v}_{end,p}$$

(10) Fuel limits for the offshore facility p

$$v_{p\,min} \leq v_{j,p} \leq v_{p\,max}$$

(11) First time stamp must be consistent with the initial condition (7)

$$t_{0,p} = 0$$

(12) Last time stamp must be consistent with the final condition (8)

$$t_{max} - 1\text{day} \leq t_{end,p} \leq t_{max}$$

4.3.2 Loading and Unloading of Goods

There are many complexities involving the loading and unloading of the goods. The schedule includes the earliest date that a good may be available at the dock and the latest date for delivery to the offshore facility. The max weight and volume allowed for all freight (collection of goods) can be set. Varying the right hand side of constraints 8 and 9 we can lower the load of the maximum ship capacity by 50% as requested by the customer.

(1) Conservation equation for the good i at time j for the ship s.

$$g_{i,j,s} = g_{i,j-1,s} + \Delta g^{+}_{p(i),j,s} - \Delta g^{-}_{p'(i),j,s}$$

(2) The good i must be picked up before $\bar{t}_{i,p,max}$

$$t_{j,s,p} \leq \bar{t}_{i,p,max} + \left(1 - \Delta g^{+}_{p(i),j,s}\right)T$$

(3) The good i must be picked up after $\bar{t}_{i,d,min}$

$$t_{j,s,p} \geq \bar{t}_{i,p,min} - \left(1 - \Delta g^{+}_{p(i),j,s}\right)T$$

(4) The good i must be delivered before $\bar{t}_{i,p,max}$

$$t_{j,s,p} \leq \bar{t}_{i,p,max} + \left(1 - \Delta g^{-}_{p(i),j,s}\right)T$$

(5) The good i should be picked up, g_i will go in the objective function times a large penalty forcing optimal solutions to deliver the maximum amount of goods as possible.

$$\sum_{j,s} \Delta g^{+}_{p(i),j,s} = 1 - g_i$$

(6) initial condition for $g_{i,j,s}$

$$g_{i,0,s} = \Delta g^{+}_{p(i),j,s}$$

(7) The good i must be delivered if previously pickup

$$g_{i,end,s} = 0$$

(8) The freight cannot exceed the max weight for the ship s

$$\sum_i w_i g_{i,j,s} \leq Maxweight_s$$

(9) The freight cannot exceed the max volume for the ship s

$$\sum_i \mathrm{v}_i g_{i,j,s} \leq Max\mathrm{vol}_s$$

4.4 Forcing Consistency On The Routes

The routes, or transits, that a ship makes between the dock and to, from, and between offshore facilities must be managed carefully. The complexity is fairly obvious when you review the constraints we developed below.

(1) Upper bound for the time required to go from offshore facility p to offshore facility p' or remain at the dock indefinitely. $\Delta_{p,p'}$is the travel time.

$t_{j+1,s} \leq t_{j,s} + \sum_{p,p'} \Delta_{p,p'}\, x_{j,s,p} x_{j+1,s,p'} + T \sum_d x_{j,s,d} x_{j+1,s,d}$

(2) Lower bound for the time required to go from offshore facility p to offshore facility p' or remain to at the dock indefinitely.

$t_{j+1,s} \geq t_{j,s} + \sum_{p,p'} \Delta_{p,p'}\, x_{j,s,p} x_{j+1,s,p'}$

(3) Consistency of the offshore facility time and ship time.

$t_{j,p} = \sum_s x_{j,s,p}\, t_{j,s}$

(4) Consistency of the dock time and ship time.

$t_{j,s,d} = x_{j,s,d} t_{j,s}$

(5) Enforce ordering of the ship stops

$t_{j+1,s} \geq t_{j,s}$

(6) Enforce ordering of the offshore facility times.

$t_{j+1,p} \geq t_{j,p} + \Delta\bar{t}_{min}$

(7) Load from the dock if the ship is present

$\Delta g^{+}_{p(i),j,s} \leq x_{j,s,p}$

(8) Load or unload at the offshore facility only if the ship has stopped there

$\Delta g^{-}_{p(i),j,s} \leq x_{j,s,p}$

(9) Ships must stop either at a offshore facility or at dock for a given time j.

$\sum_p x_{j,s,p} = 1$

(10) offshore facilitys accept at most one ship at the same time.

$\sum_s x_{j,s,p} \leq 1$

(11) Ships cannot stop two consecutive times at the same offshore facility

$x_{j,s,p} + x_{j+1,s,p} \leq 1$

(12) Initial location of the ship s.

$x_{0,s,p} = \bar{x}_{0,s,p}$

(13) Final location of the ship s.

$x_{end,s,p} = \bar{x}_{end,s,p}$

4.5 The Cost/Objective Function

The objective function represents the desired optimization goals. For this problem, we want to minimize the time required for the ships to complete their routes. To ensure that the delivery of goods is enforced, there is a penalty in the objective function for any undelivered good.

Variable Coefficients:

(1) $x_{j,s,p,p'}$ coefficient is $\Delta_{p,p'} + C_{s,p'}$ which is the time to transit and perform cargo ops.

Variable $x_{j,s,p,p'} = x_{j,s,p}\, x_{j+1,s,p'}$ indicates ship s transits from p to p' at event j.

(2) g_{i} coefficient is $10 \max_{p,p'}(\Delta_{p,p'} + C_{s,p'})$ which is the penalty for not delivering cargo.

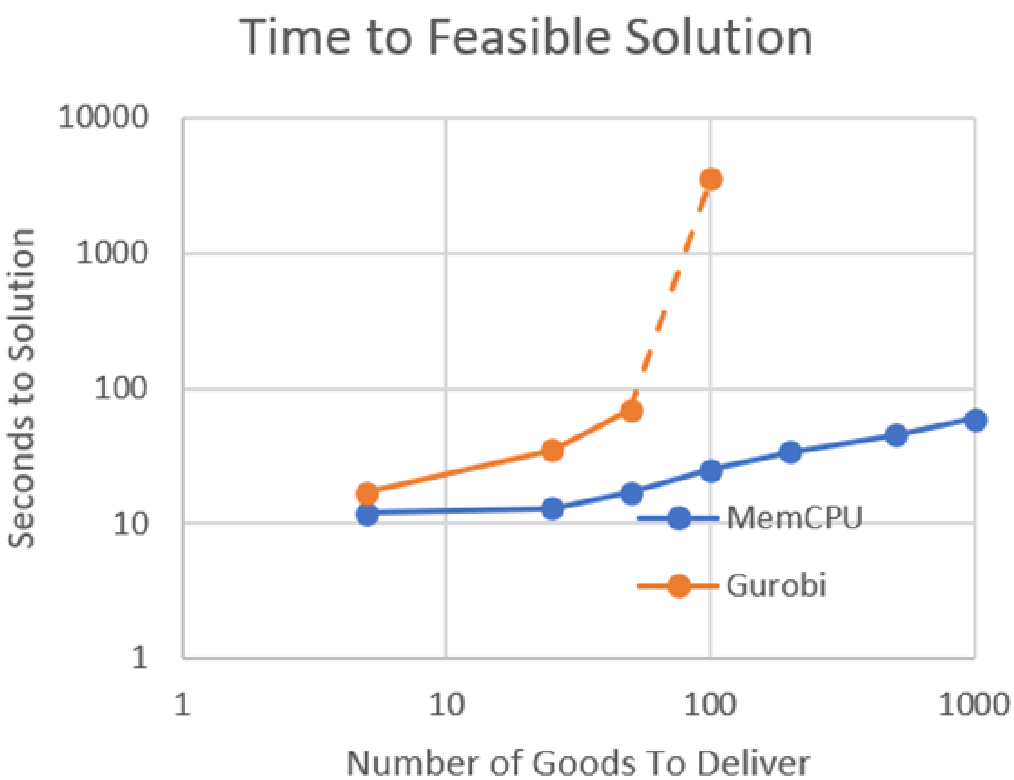
Variable g_{i} indicates good i has been delivered consistent with the request.

5. Comparison

As part of the evaluation, the teams ran the ILP formulations on the MEMCPU Platform and a best-in-class commercial solver, Gurobi.

Gurobi was accessed through the freely available NEOS Server for Optimization. Gurobi is one of the world's best-in-class commercial solvers [10]. Gurobi is extremely efficient for many problems. However, Gurobi is subject to the limitations of the branch and bound and cutting plane algorithm. Thus, because the problem is the combination of multiple NP-Hard problems, the company's scheduling is expected to scale exponentially.

The team evaluated the scaling of the MEMCPU Platform and Gurobi by starting with a small number of goods and then building up from there. The figure below shows this scaling comparison and the rapid exponential increase in the time to solve on the Gurobi solver.

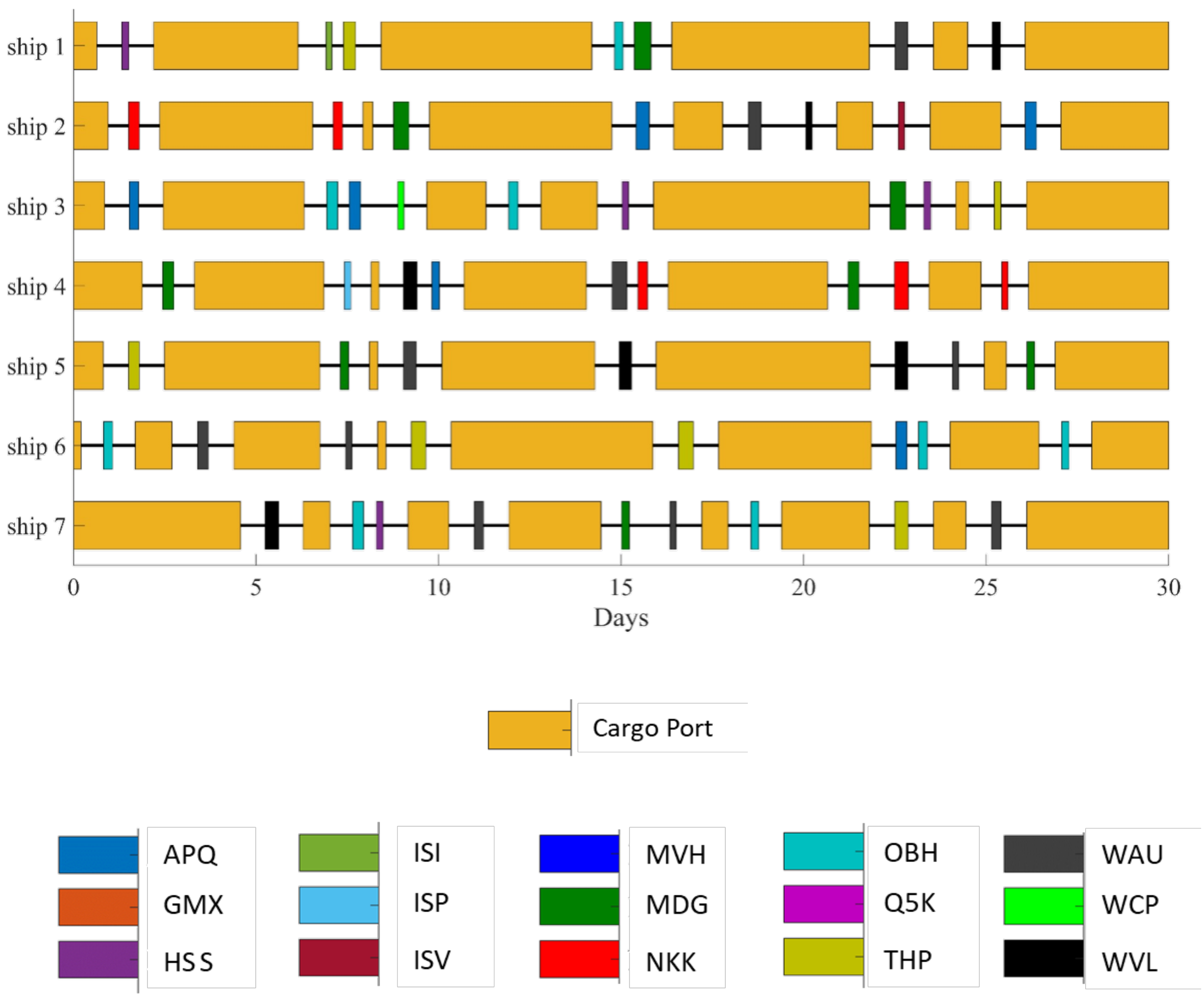


At scale, 3,500 goods must be delivered over a 30-day scheduling period. The results in the table above show that Gurobi can solve instances of the problem that reach up to 80 goods. However, at 100 pieces of Cargo, Gurobi times out after 8 hours without finding a solution. On the other hand, the MEMCPU Platform converges to a solution in seconds for all instances regardless of size, and scales linearly. Therefore, the full evaluation could only be completed on the MEMCPU Platform.

5.1 Results

5.1.1 Scheduling of Vessels (stops at offshore facilities)

The following graphic visually represents the schedule for each ship, showing which days it will be in port (Yellow), which days it will be in transit (line), and which days it will be at each offshore facility (facilities denoted by color).



5.1.2 Percent of Cargo Delivered

In this table, note the "# ship" column under Actual and MemComputing. The data shows that the optimal schedule generated by MemComputing only requires 7 of the original 12 ships. Further, the number of cargo elements was increased by 15.8% over the original schedule, leaving only 4.2% of the cargo to be delivered by the Fast Ships.

30 day schedule	Actual			MemComputing		
	# ship	Cargo	percent	# ship	Cargo	percent
Slow Ships	12	2899	80%	7	3472	95.8%
Fast Ship	2	725	20%	2	152	4.2%
Totals	14	3631	100%	9	3355	100.0%

5.1.3 Comparison of Number of Transits

This table shows another important measurement in optimizing cargo delivery, the number of transits. Note that the original solution required 176 transits total by the 12 ships. However, our optimized solution only requires 92 transits by the 7 ships, representing a 48% reduction in the number of transits.

30 day schedule	Actual		MemComputing		
	# ships	# Transits	# ships	# Transits	% reduction
Slow Ships	12	176	7	92	48%
Fast Ship	2	32	TBD	TBD	
Totals	14	162	7	102	

6. Conclusion

6.1 General Conclusion

This case study involved developing an optimal 30-day schedule for delivering 3,500+ goods from a port using a variety of ships to various offshore facilities. The goal was to provide a far more optimized shipping schedule while utilizing a minimal number of ships.

The team evaluated a state-of-the-art commercial solver, the Gurobi Solver. However, Gurobi showed rapid exponential scaling demonstrating the NP-Hardness of the problem. At 100 goods, Gurobi's computational time exceeded 8 hours to find any feasible solution.

The MEMCPU Platform showed linear scaling as the number of goods increased. The resulting schedule over 30 days showed the following optimizations.

- An increase of 15.8% in the number of goods delivered by the Slow Ships
    - This also means a 15.8% reduction in the goods delivered by the Fast Ships, which might justify eliminating 1 Fast Ship.
- A decrease in the number of Slow Ships from 12 to 7 (42% reduction)
- The total number of transits decreased from 176 down to 92 (48% reduction).

These optimizations should then deliver the following benefits.

- $1.5M monthly($18M annual) reduction in ship leasing payments.
- 18kt reduction in carbon output based upon the reduction in the number of transits.
- Possibly other hard-dollar and certainly additional soft-dollar savings.
- The ability to multiply the savings by the number of situations around the world where they have ocean-based oil rigs and exploration.

This work has demonstrated that the MEMCPU Platform can efficiently solve NP-Hard optimization problems that are intractable for today's best-in-class solutions, driving significant improvements in efficiency for various applications in the energy sector and beyond.

6.2 Recommendations for Future Enhancements

The following recommendations will further aid the scheduler.

- Initially include Fast Ships in the optimization with the ability to override and repurpose by the scheduler. Overriding will cause the schedule to be re-optimized based on the (temporary) removal of a Fast Ship.
- Real-time updating: The system will monitor the logistics database to provide the following functionality.
    - Automatically rerun the 30-day schedule and notify the scheduler of any changes for approval.
    - Provide a dynamic schedule for the current day. As cargo is being loaded or unloaded, real-time issues can be plugged in, and the schedule adjusted for the remainder of the day (or until the next issue).
    - From this, the scheduler will be able to provide updates to the workers on the offshore facilities as to when they can expect certain cargo.
- What if scenario builder: The scheduler can adjust priorities, cargo availability and offshore facility required dates and then run multiple scenarios in parallel to choose the best one.
- Emergency evacuations: In the event of a hurricane or other situation that would require evacuating workers from offshore facilities, the system could be halted and immediately repurpose all ships to evacuate all personnel optimally.
- Include weather forecast expectations in 30-day schedule to refine the dynamic weather constraint.
- Optimizing the loading of cargo on the deck. In this case study no more than 50% of the deck space could be occupied. However, the placement of cargo could be optimized such that a greater percentage of deck space would be used. This would potentially reduce the number of transits and consequently the number of required ships [11].
- Lastly, forecasting could be run out many months vs. just 30 days. If the amount of cargo is going up or down over time, the system could identify the optimal times to add or remove ships. Here the leasing contract duration/end date for each ship should be included to enhance the accuracy of these calculations.

References

[1] Massimiliano Di Ventra, Fabio L. Traversa, (2018), *SELF-ORGANIZING LOGIC GATES AND CIRCUITS AND COMPLEX PROBLEM SOLVING WITH SELF_ORGANIZING CURCUITS,* US 9,911,080 B, USPTO, the title of the patent (in italics), the patent number, https://patents.google.com/patent/US9911080B2/en?oq=US+9%2c911%2c080+B

[2] "Memcomputing vs Branch and Bound: Memcomputing." MemComputing | Breakthrough Computing Performance, 25 July 2022, www.memcpu.com/videos/memcomputing-vs-branch-and-bound/

[3] M. Di Ventra, F. L. Traversa, Perspective: Memcomputing: Leveraging memory and physics to compute efficiently, Journal of Applied Physics, **123**, 180901, 2018

[4] F. L. Traversa, M. Di Ventra, Polynomial-time solution of prime factorization and NP-complete problems with digital memcomputing machines, Chaos: An Interdisciplinary Journal of Nonlinear Science, **27**, 023107, 2017

[5] F. L. Traversa, M. Di Ventra, Memcomputing integer linear programming, arXiv preprint arXiv:1808.09999, 2018

[6] F. L. Traversa, M. Di Ventra, Self-Organizable Logic Gates and Circuits and Complex Problem Solving with Self-Organizing Circuits, US patent US 9911080 B2 filed on July 12, 2015, published 2018

[7] All peer reviewed articles on memcomputing can found at https://www.memcpu.com/publications/

[8] Fernanda Menezes, Oscar Porto, Marcelo L. Reis, Lorenza Moreno, Marcus Poggi de Aragão, Eduardo Uchoa, Hernán Abeledo, Nelci Carvalho do Nascimento (2010). Optimizing Helicopter Transport of Oil Rig Crews at Petrobras. Informs

[9] F.L. Traversa (2021). Optimizing Helicopter Transportation to Oil Rigs.

[10] https://www.gurobi.com/

[11] Aircraft Cargo Loading Optimization Case Study (https://www.memcpu.com/media/2019/04/AirCargo-UseCase-V2.pdf)