Danmarks
Tekniske
Universitet

DTU

42117 Transport Optimization
Assignment 1: ERCSP for Fast Bunk

Authors:	*Student Numbers:*
Malthe Jelstrup	s184291
Vera Nijmeijer	s213011

September 21, 2022

Introduction

In this assignment we made a proof of concept for bulk shipping company FastBulk A/S. The goal is to choose a set of orders which maximizes the company's profit. The profit is the chartering revenue minus the bunker consumption.

Labelling Algorithm Modifications

To identify the best set of orders we modified the label correcting algorithm of the ERCSP. The label contains the following information: $< v, d[v], [K_v \forall v \in V], [r_i \forall i \in I], \pi >$. The changes we made affect the "extend" function and "dominance" function.

In the "extend" function we check that the bunker fuel used up until the previous node plus the bunker fuel that would be used by adding this edge, will not exceed the maximum of 60. If it does lead to a bunker fuel usage of more than 60, no label will be created. We also check if the vessel is able to do this order in the right time window. To ensure that, we check three things:

- If the current time we have used plus the time used by this vertex is smaller than the end of the time window of the order.
- If the start time of the time window is bigger than the current time, the time is set to the start time of the time window. Therefore the time after completing the order is the start time window plus service and sailing time of the order.
- If vessel arrives in the port after the start of the time window and is able to complete the service before the end of the time window. The sailing and service time are added to the resources used.

A label is created for the new visited vertex, the new profit and updated used resources (time and bunker fuel).

The "dominance" function also has changes, because it was based on minimizing cost and the goal now is to maximize profit. The difference is that label λ now dominates label λ^* if $d[\lambda] \geq d[\lambda^*]$. The other rules stay the same, where the label also has to have visited a subset of the nodes ($K_\lambda \leq K_{\lambda^*} \forall v \in V$) and used less resources ($r_i[\lambda] \leq r_i[\lambda^*] \forall i \in I$).

1 Visualizations

The ports accessible in this problem are laid out in the Mediterranean sea as can be seen in figure 1, where the colors indicate the following: green is land, blue is water, orange are ports and white is the initial port Bari.

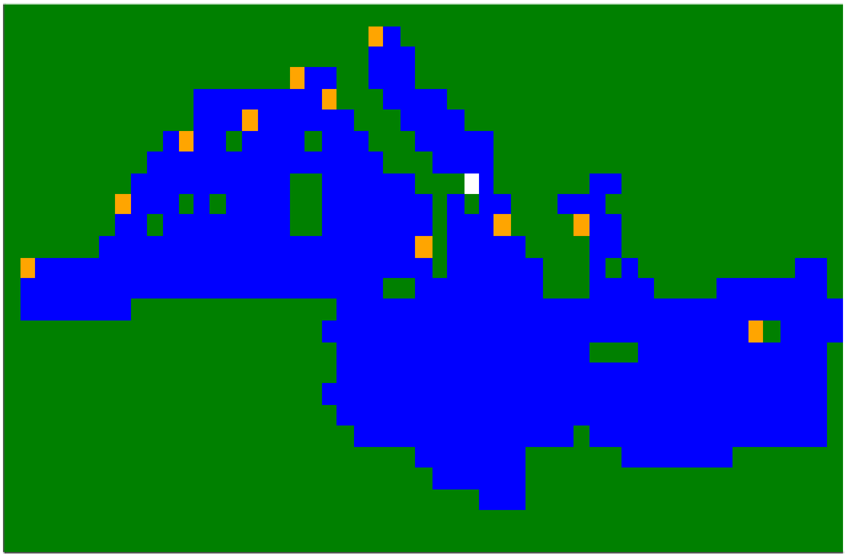
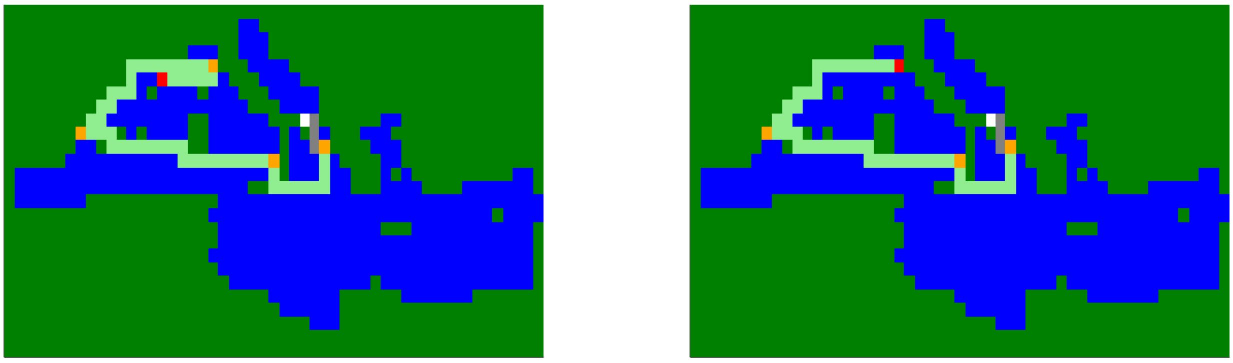


Figure 1: Map of instance highlighting port locations

Each instance of the problem was solved twice, firstly not accounting for the time-windows in which orders could be picked up and secondly factoring in the time-windows. The qualitative results as well as the order routes for the three instances, both constrained by time and not, can be found in the table 1 below. In the following figures 2-4 the routes resulting from accounting for time is shown and the colors indicate the following: green is land, blue is water, orange are intermediate ports, white is the initial port Bari, red is the final port, light green is the path the ship should travel while carrying cargo and grey is the path the ship should travel while empty of cargo (to get to the next order).

	Type	Profit	Fuel Consumption	Order Route
Instance 1	Without time	$71100	59 Bunker	18, 1, 6, 2
	With time	$64350	53 Bunker	18, 1, 6
Instance 2	Without time	$50300	60 Bunker	12, 13, 17, 2
	With time	$35350	51 Bunker	8
Instance 3	Without time	$51375	50 Bunker	7, 5
	With time	$51375	50 Bunker	7, 5

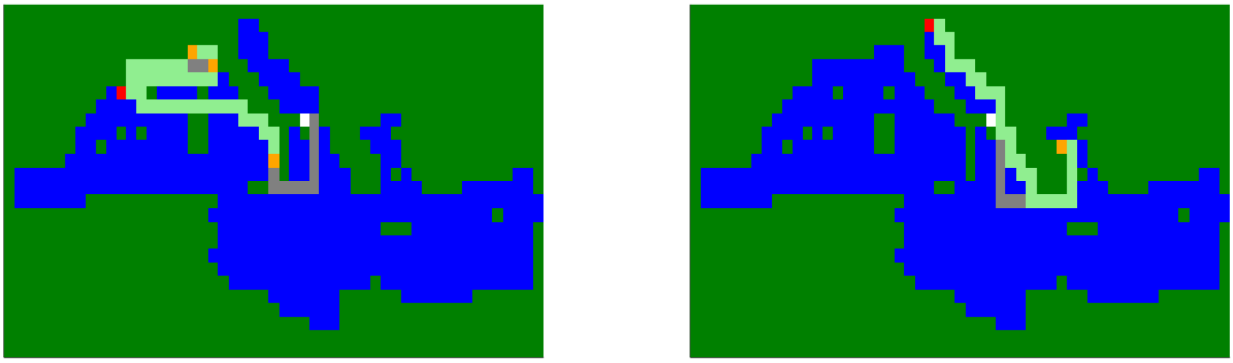
Table 1: Instance route information



(a) Without time, orders; [18, 1, 6, 2]

(b) With time, orders; [18, 1, 6]

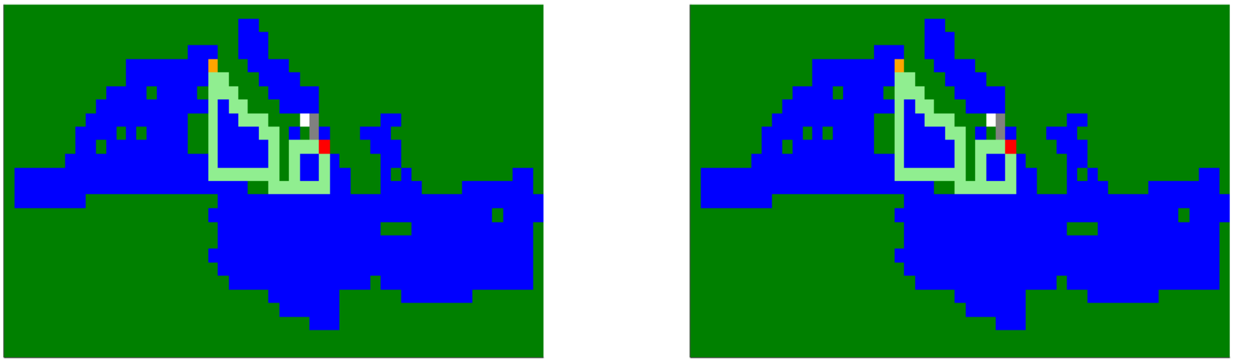
Figure 2: Vessel routes for instance 1



(a) Without time, orders; [12, 13, 17, 2]

(b) With time, orders; [8]

Figure 3: Vessel routes for instance 2



(a) Without time, orders; [7, 5]

(b) With time, orders; [7, 5]

Figure 4: Vessel routes for instance 3