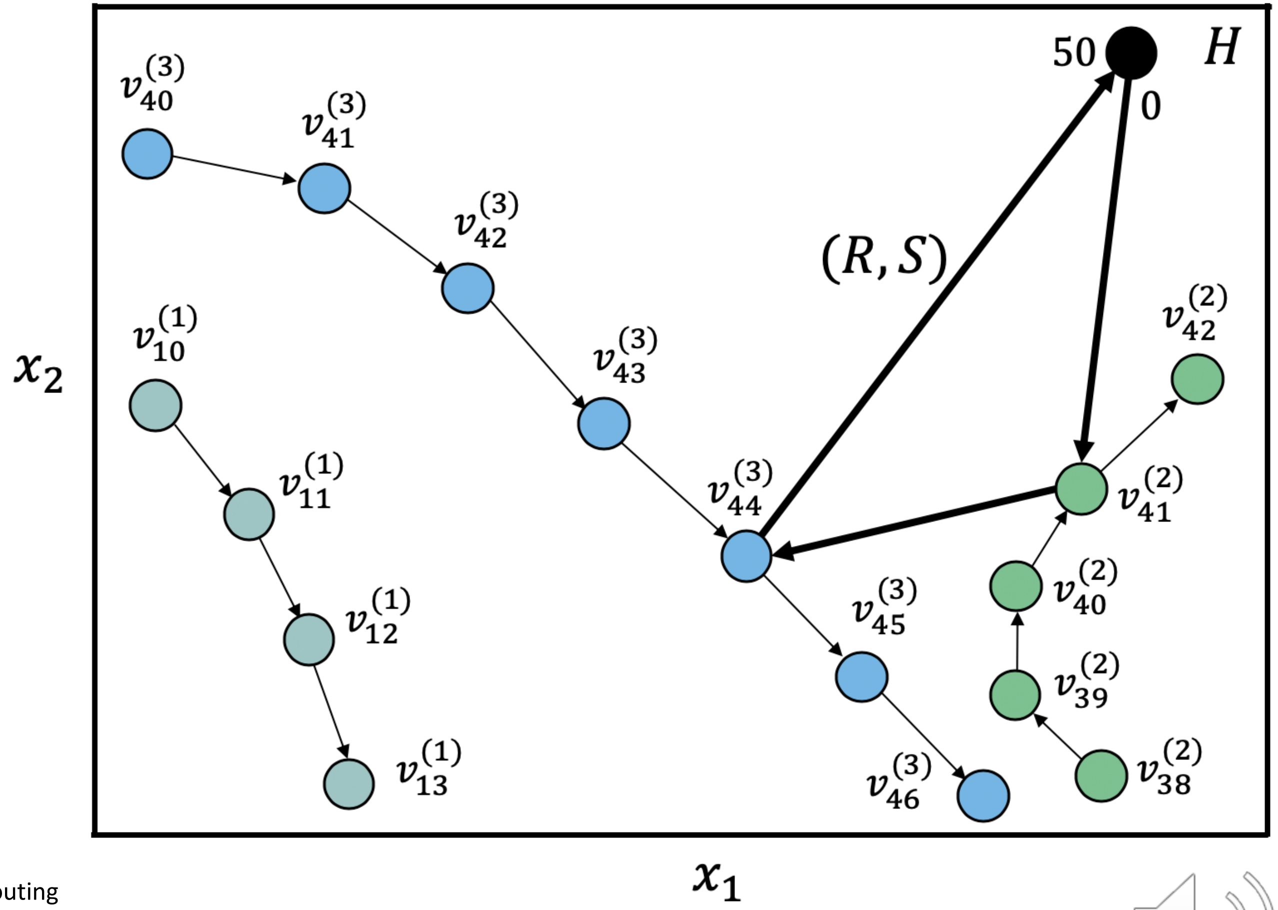
Dynamic Vessel-to-Vessel Routing Using Level-wise Evolutionary Optimization

Yash Vesikar, Julian Blank, Kalyanmoy Deb, Markku Kalio, Alaleh Maskooki

COIN Laboratory, Michigan State University

DV2VRP Problem Formulation

- The service ship must simultaneously optimize the following objectives:
  1. Maximize the number of different the target ships visited (**α**) within a specified time period T
  2. Minimize the total distance traveled (***d***)
- Depart and return to the Harbor before a pre-defined time limit T_w is exceeded
- Is a generalized traveling salesman problem with an incorporation of time dependencies
- Variable Encoding:
  - R = (H, 2, 3, H)
  - S = (0, 41, 44, 50)

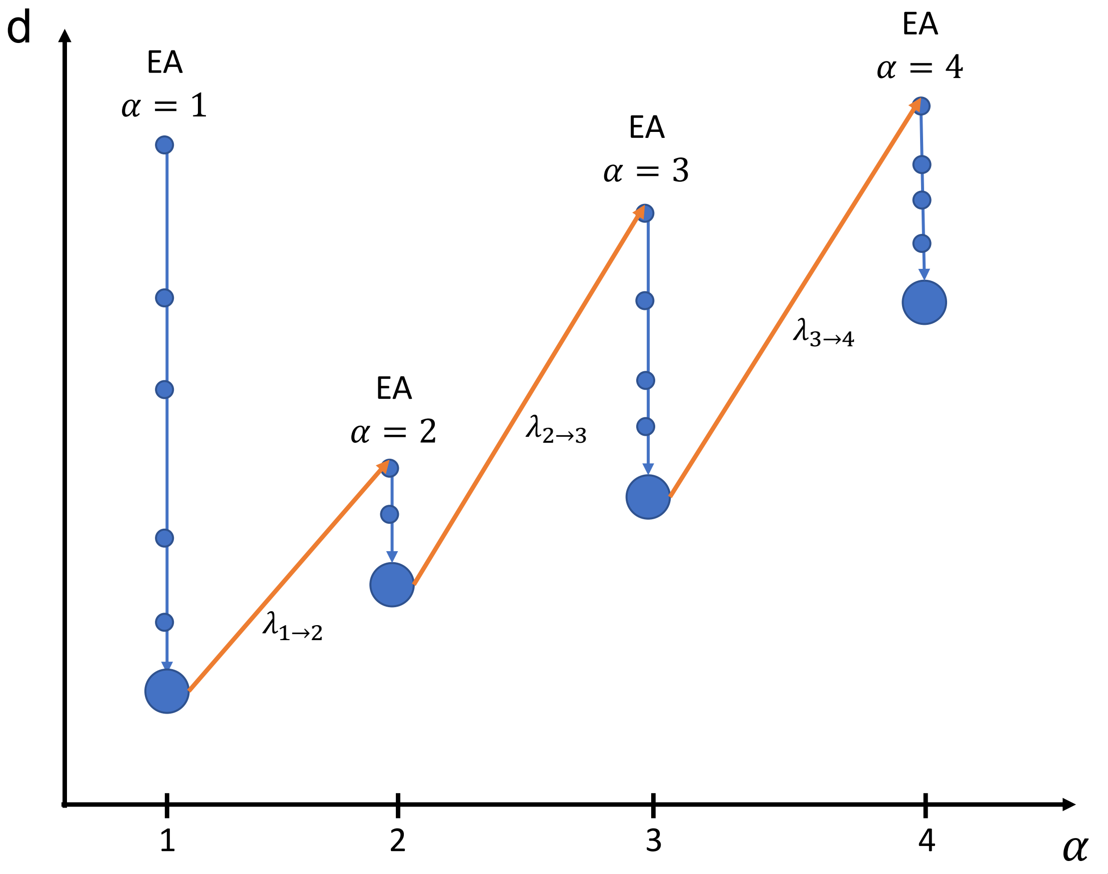


More details about the problem can be found in [1]: A. Maskooki and Y. Nikulin. 2018.Multiobjective Efficient Routing In a Dynamic Network. Technical Report 1198, Turku Center for Comp. Sc. (TUCS), Finland

Proposed Level-Wise GA

Multi-Level Approach

1. **α-level**: Subproblem (α=k) and make the transition from α=k to α=k+1 through a heuristic-based initial population
2. **Upper level:** Genetic Algorithm optimizing routes given an α
3. **Lower level:** Optimizing schedules using dynamic programming given a route



We have used the multi-objective optimization framework pymoo [2] as a basis for our customizations.

α-level Optimization

All sequences in α-level subproblem have a sequence length of α

To advance to the next α-level we need to define a transition function to increase α

Transition Function

Sequence Schedule

0,	32,	63,	4,	0
0,	15,	35,	50,	72

Window

$f1 = 3$
$f2 = 42.864$

$k = 2$
$n = 1$

→

Sequence Schedule

0,	32,	30,	63,	4,	0
0,	15,	25,	35,	50,	72

$f1 = 4$
$f2 = 43.474$

Parameters:
k := new permutation size
n := number of ships to replace

Upper Level Optimization

Upper Level optimization is a custom GA that searches for routes with the following operators:

Selection - Random Selection

Crossover - Single-point crossover

Parent 1:	0,	32,	4,	63,
Parent 2:	0,	15,	6,	12,

→

Parent 1:	0,	32,	6,	12,
Parent 2:	0,	15,	4,	63,

Mutation - Modified Transition function

k = n, no new ships are inserted, the existing sequence is mutated

0,	32,	6,	12,

→

0,	32,	5,	12,

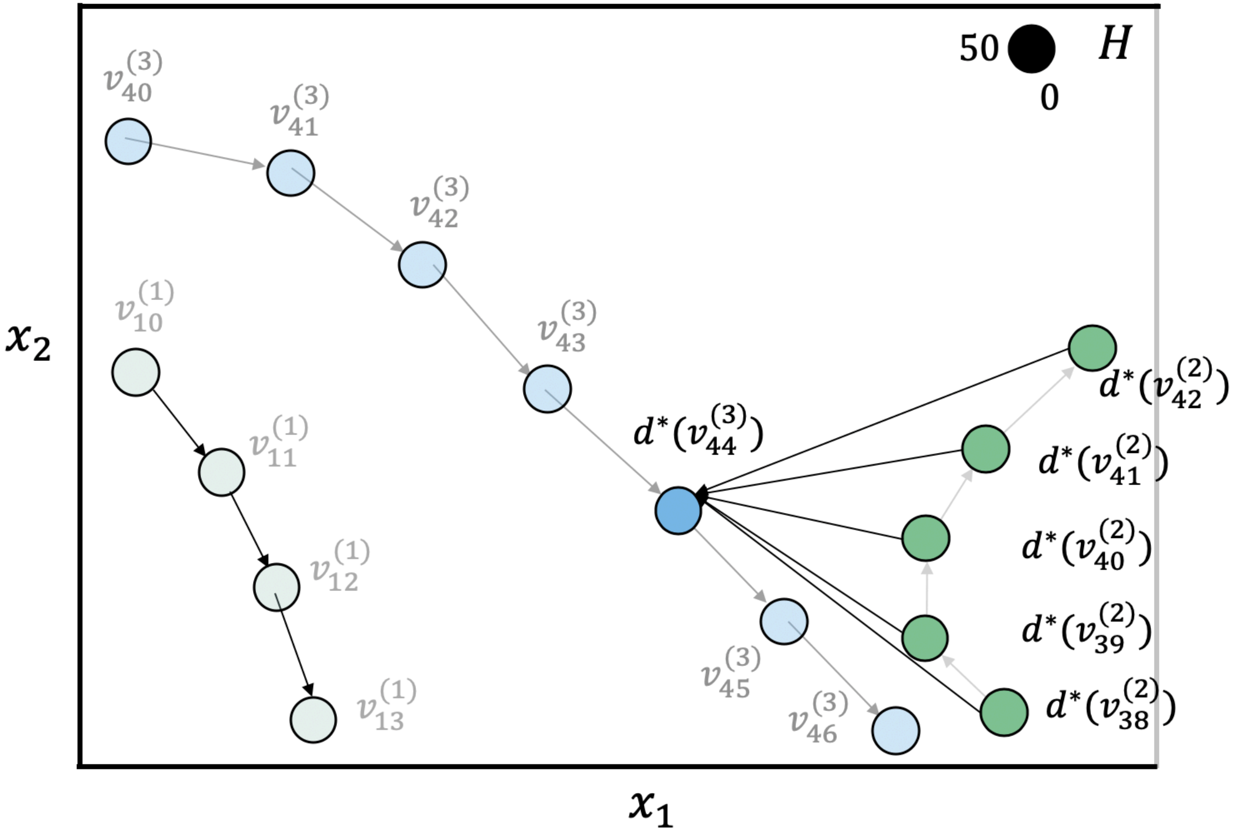
Lower Level Optimization

Given a sequence of target ships the lower level optimizer returns schedule and total distance for the sequence.

- R = (H, 2, 3, H)
- Transition from 2 to vessel 3

$$d^*\left(v_t^{(R+1)}\right) = \min_{q \in \Omega(v^{(R_k)})} \left[d^*\left(v_q^{(R_k)}\right) + c(v_q^{(R_k)}, v_t^{(R_{k+1})}) \right]$$

- Repeat this for all $v^{(R_k)}$



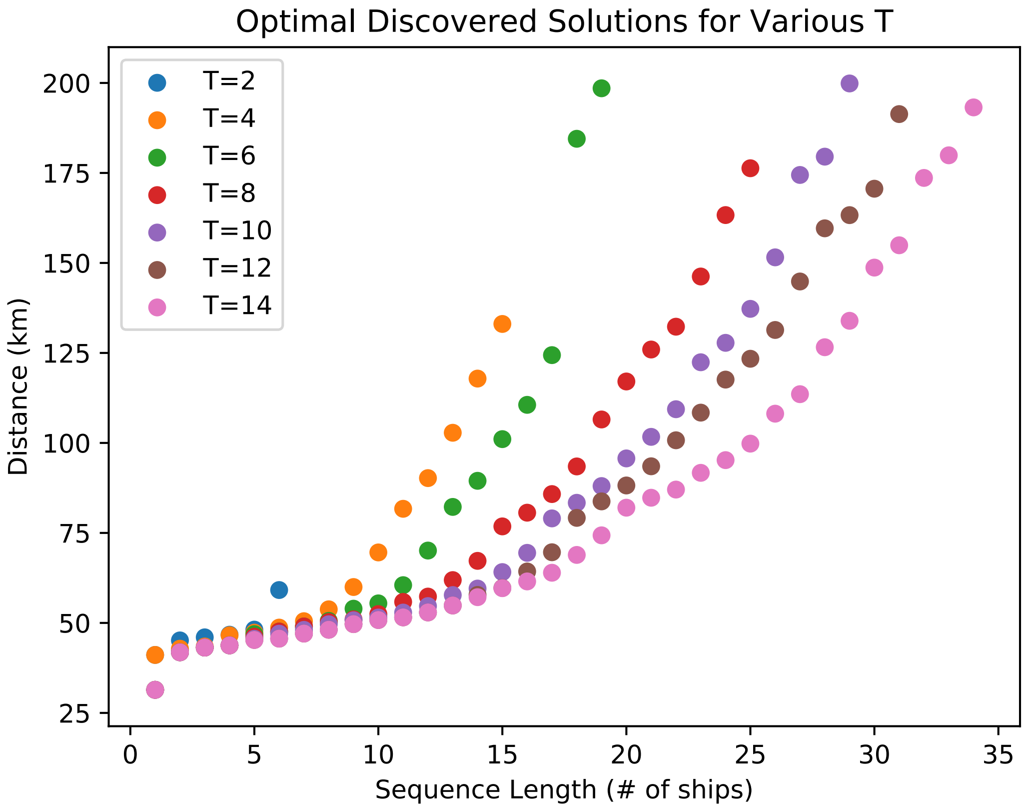
Experimental Results

Execution Times Comparison(s)

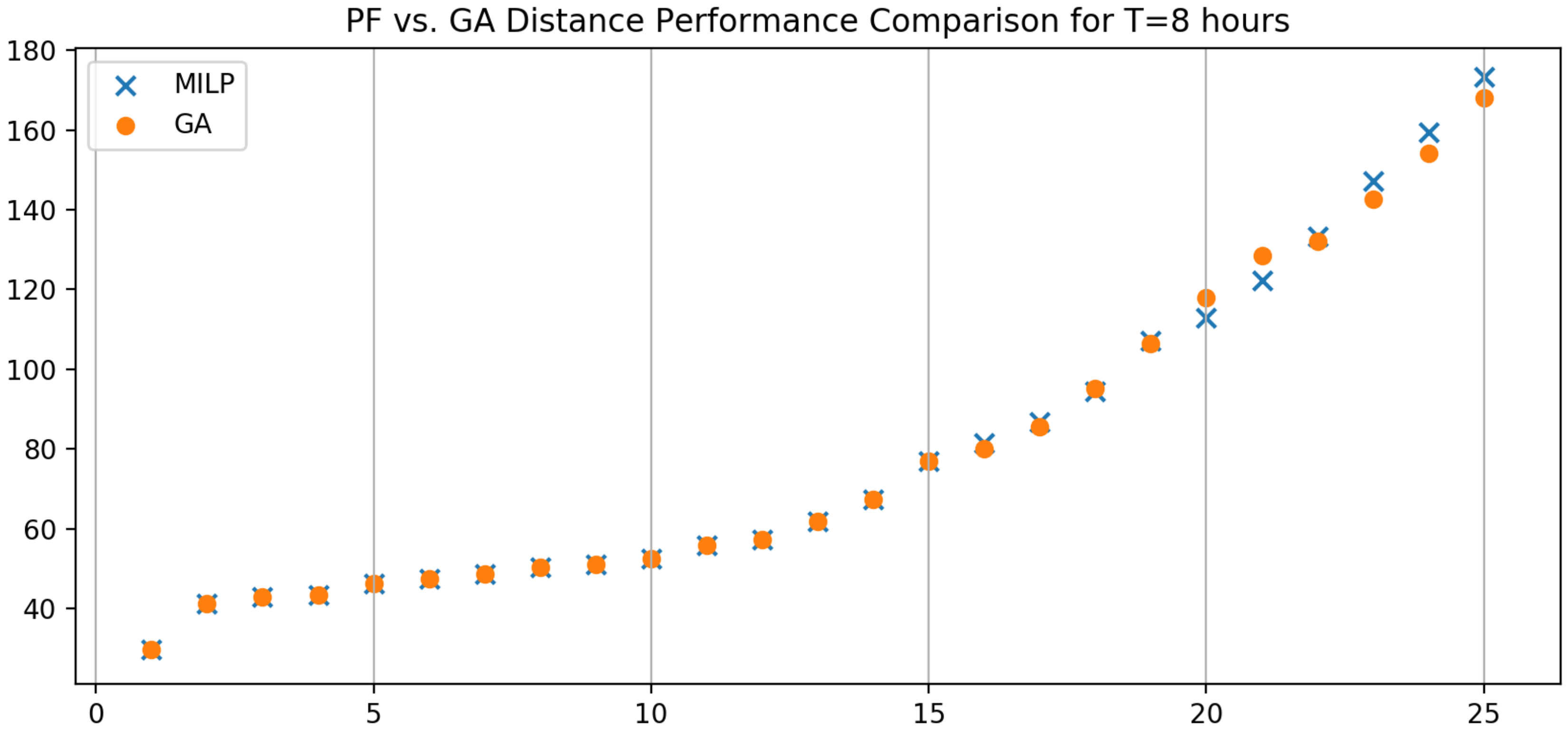
T	GA	MILP
4	217	30
6	416	404
8	425	1214
10	1832	7285

Max alpha Comparison

T	GA	MILP
4	15	15
6	20	20
8	25	25
10	30	32



Experimental Results - Pareto-Front Comparison



T	GA (s)	MILP (s)
8	425	1214

Due to slight differences in problem formulation, the GA is occasionally able to outperform the MILP optimal solution. Throughout the course of our study we have found these differences to be insignificant.

Future Work

Design a framework for solving large scale dynamic routing problems that are otherwise intractable using standard MILP techniques.

Going forward we are investigating:

1. Dense networks with many ships and many available positions
2. More sophisticated transitioning techniques, escaping local optima

References

[1] A. Maskooki and Y. Nikulin. 2018. Multiobjective Efficient Routing In a Dynamic Network. Technical Report 1198, Turku Center for Comp. Sc. (TUCS), Finland.

[2] J. Blank and K. Deb, "pymoo: Multi-Objective Optimization in Python," in IEEE Access, vol. 8, pp. 89497-89509, 2020, DOI: 10.1109/ACCESS.2020.2990567.

Questions?