Optimizing Group Transit in the Gulf of Aden

Ondřej Hrstka[1], Ondřej Vaněk[1]

[1]Dept. of Cybernetics, FEE Czech Technical University, Technická 2, 166 27 Praha, Czech Republic

{hrstka,vanek}@agents.felk.cvut.cz

Abstract. *The situation around the Horn of Africa is critical. The pirates attack transiting vessels on daily basis and even though various counter-piracy measures have been employed, there is still a great potential for the optimization of these navigation techniques. We focus on optimization of a specific transit scheme – combination of the International Recommended Transit Corridor with Group Transits, temporal schedules for ships sailing through the Gulf of Aden.*

Having a large dataset of ships' speeds, we employ dynamic programming together with branch and bound technique to optimize the Group Transit schedules, proposing a variety of alternatives. We evaluate the results both statistically, taking into account only distribution of speeds, and dynamically, using a mature multi-agent simulation to verify the superiority of proposed schedules in a dynamic and rich simulated environment. The results of this work show, that even small changes in the current schedules can save days of transit times and millions of US dollars per year.

Keywords

group transit, maritime security, piracy, dynamic programming, optimization

1. Introduction

Over the last years, maritime areas around the Horn of Africa have experienced a steep rise in piracy. For approximately 20 thousand ships that annually transit the area, insurance rates have increased more than tenfold and the costs of piracy were estimated at up to US$16 billion in 2009.

Even though the pirates are able to attack ships up to 1500 nm[1] from the Somali coast, they are mostly concentrated in the narrow corridor in the Gulf of Aden (GoA), where all the transiting ships have to pass. This makes the area – together with a small distance from the shore and high traffic density – a perfect ambush place.

Two adjustments to increase the security of the GoA transit were made: establishment of the International Recommended Transit Corridor (IRTC), a direct shipping lane through the GoA and Group Transits, transit schedules for ships of various speeds. Groups of ships are defined by a set of speed levels. Each ship has to reduce her speed to the closest group speed level to transit the corridor together with ships of similar speed, having an advantage of better protection from present naval forces. However, the disadvantage of Group Transits is the reduction of speed of basically every ship, which results in significant delays in shipping times.

Our main goal is to optimize the Group Transit schedule and propose an optimal grouping scheme to minimize the delay without significantly decreasing security of each ship. We formalize the situation as an optimization problem and we use dynamic programming together with branch and bound method to solve the problem for varying number of speed levels. We evaluate the results in a multi-agent simulation of the maritime domain (AgentC) and we show that by modifying the current Group Transit schedule, we are able to save hundreds of days of the aggregated shipping time and millions of US dollars per year without increasing number of hijacked ships.

2. Domain Description

In recent years, the *Gulf of Aden* (GoA) became the central area of pirate activity. To be able to effectively protect transiting ships, the International Recommended Transit Corridor (IRTC), together with Group Transit schedules, was established.

IRTC is a straight transit lane in GoA, introduced in February 2009. IRTC is formed of two corridors, one for east bound transit and one for west bound transit. Each corridor is 5 nm wide with 2 nm buffer zone between and it is defined by an entry point and an exit point[2]. IRTC is successor of *Maritime Security Patrol Area* (MSPA). Unlike MSPA, it avoids national waters, international navy forces are thus able to protect merchant ships along all their route through GoA. IRTC is also designed to avoid main fishing areas, which results in a decrease of false piracy alerts.

Group Transits were introduced in August 2010. The main idea is to form ships into groups that sail through the

[1] 1 nautical mile = 1.852 kilometers

[2] http://redfoursecurity.com/clientdocs/doc_download/4-gulf-of-aden-irtc.html

Speed	Entry point A – time	Entry point B – time
10 kn	04:00 GMT+3	18:00 GMT+3
12 kn	08:30 GMT+3	00:01 GMT+3
14 kn	11:30 GMT+3	04:00 GMT+3
16 kn	14:00 GMT+3	08:30 GMT+3
18 kn	16:00 GMT+3	10:00 GMT+3

Tab. 1. Gulf of Aden group transit schedule for vessels traveling at different (maximum) speed.

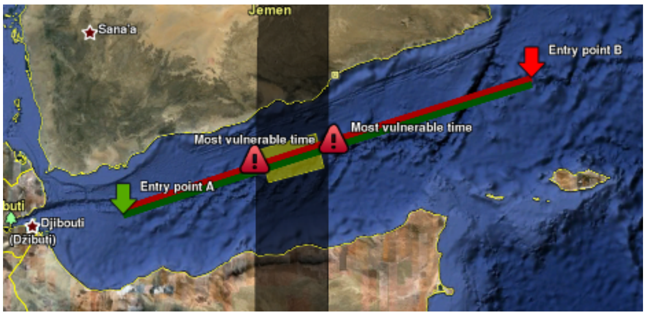

Fig. 1. Situation in the GoA. Vessels enter the gulf area from both sides. The most dangerous area between 47E and 49E (denoted by the yellow rectangle) is best to be crossed at night. Entering and exiting the area is the most dangerous part of transit as most attacks take place at dawn/dusk.

Gulf together. Every ship with travel speed under 18 knots[3] is strongly recommended to participate in the Group Transit. For each group, there is a schedule with an entry time and a group speed (see Table 1). There are ten scheduled groups, five for east bound and five for west bound traffic. A ship sailing in respective direction should adjust her speed to pass through the relevant entry point at the recommended time and then continue with the group speed through GoA. Ships should not stop and wait near entry points for other ships in group to avoid risk of being attacked by pirates. The schedule is designed to synchronize every group with the 24 hour cycle, passing the high risk area at night, when it is difficult for pirates to attack. The situation is depicted on the Figure 1.

Even though there are many naval forces present in the area, the groups do not have guaranteed protection along the corridor. However, the warships are aware of the schedule and modify their routes as to stay close to most of the transiting groups. The effectiveness of this mechanism is high, resulting in zero successful attacks in the IRTC for the last year.

2.1. Available Data

To be able to optimize the Group Transit schedules, we have collected over 2500 samples of speeds of transiting vessels from the Vessel Tracker website[4], which stores AIS records of most ships around the world. AIS (*Automated Identification System*) records are data samples from an automated tracking system used for identifying and locating ships. AIS data contain information about ships' trajectory tagged with a time stamp. From these data we were able to determine the average speed in the corridor by averaging the ratios between two spatial samples in the area and the attached time interval. The data are visualized in form of a histogram on the Figure 2.

3. Problem Definition

Our solution is based on finding a combination of N speed levels with minimal sum of ships' delays. This can be also viewed as partitioning histogram of ships speed into N parts.

We have given histogram of ships' speeds H. The histogram is computed from test dataset with the bin width 0.1 kn. Bins positions form a set $\mathbf{V}$ where $v \in V$ is the ship's speed and $H(v)$ is the number of ships with travel speed v.

We seek to find a set of N groups $\mathbf{G} = \{g_1, ..., g_N\}$. Group g_i is defined as discrete interval $g_i = < \underline{v_i}, \overline{v_i} >$, where $\underline{v_i}$ is the transit speed of the group g_i. This means, that each ship with a speed $v \in < \underline{v_i}, \overline{v_i} >$ must reduce her speed to $\underline{v_i}$. Groups are disjunct i.e. $\bigcap \mathbf{G} = \emptyset$, and their conjunction is equal to $\bigcup \mathbf{G} = \mathbf{V}$.

As defined, the criterion for optimization is the minimization of ships' delays. The delay for one ship at speed v is defined as

$$F(v, \underline{v}) = l(\frac{1}{\underline{v}} - \frac{1}{v}), \tag{1}$$

where $\underline{v}$ is the speed group, to which ship belongs and l is the length of GoA[5]. The delay of all ships grouped into groups in $\mathbf{G}$ can be expressed as a criterion function

$$C(\mathbf{G}) = \sum_{i=1}^{|\mathbf{G}|} \sum_{v \in < \underline{v_i}, \overline{v_i} >} H(v) F(v, \underline{v_i}). \tag{2}$$

We aim to minimize this function:

$$\mathbf{G} = \arg\min C(\mathbf{G}) \tag{3}$$

4. Implementation

First, we solved the optimization problem given in (3) by a complete enumeration. The number of all possible solutions is equal to $\binom{|\mathbf{V}|-1}{|\mathbf{G}|-1}$. This naive implementation of the complete enumeration algorithm simply iterates over all possible combinations of groups $\mathbf{G}$ and picks the best variant. This is described by the algorithm 1.

The naive implementation can be improved by dynamic programming techniques [2]. Computation of the function

[3] 1 kn = 1.852 km/h

[4] http://www.vesseltracker.com

[5] Because l is an independent constant, it can be removed from the function F.

Algorithm 1 Naive solution

```
bestValue ← ∞
bestGroup ← ∅

for all G ∈ allCombinations do
    value ← C(G)
    if value < bestValue then
        bestGroup ← G
        bestValue ← value
    end if
end for
    return bestGroup
```

$C(\mathbf{G})$ has a linear time complexity and this computation must be executed for each combination. However, a part of the computation of the function $C(\mathbf{G})$ can be used for more groups.

Let's have two combinations $\mathbf{G}_a, \mathbf{G}_b$. Each combination is proposed set of groups, with common k last groups: $\mathbf{G}_a = \{g_{1,a}, g_{2,a}, ..., g_{N-k}, g_{N-k+1}, ...g_N\}, \mathbf{G}_b = \{g_{1,b}, g_{2,b}, ..., g_{N-k}, g_{N-k+1}, ...g_N\}$. For these combinations the criterion function $C(\mathbf{G})$ can be split into common part and different part $C(\mathbf{G}) = C_{comm}(\mathbf{G}) + C_{diff}(\mathbf{G})$, where

$$C_{comm}(\mathbf{G}) = C(\{g_N - k, ..., g_N\}) \tag{4}$$

$$C_{diff}(\mathbf{G}) = C(\{g_1, ..., g_{N-k-1}\}) \tag{5}$$

For given $\mathbf{G}_a, \mathbf{G}_b$ it holds $C_{comm}(\mathbf{G_a}) = C_{comm}(\mathbf{G_b})$. This fact can be exploited in dynamic programming implementation which is described by algorithm 2.

Algorithm 2 Solution using dynamic programming

```
1:  bestValue ← ∞
2:  bestGroup ← ∅

3:  COMPUTE(N, V, 0, ∅)

4:  procedure COMPUTE(n, V_s, G, value)
5:
6:      if n = 1 then
7:          g_1 ← < min V_s, max V_s >
8:          value ← value + C({g_1})
9:          if value < bestValue then
10:             bestGroup ← G ∪ g_1
11:             bestValue ← value
12:         end if
13:     else
14:         for all v̲ ∈ possible(V_s, n) do
15:             g_n ← < v̲, max V_s >
16:             value_n ← value + C({g_n})
17:             COMPUTE(n−1, V_s \ {g_n}, G ∪ g_n, value_n)
18:         end for
19:     end if
20: end procedure
        return bestGroup
```

Algorithm	$N = 4$	$N = 5$	$N = 6$
Naive	528 ms	12 565 ms	242 960 ms
Dynamic	238 ms	4 498 ms	72 643 ms
B&B	176 ms	2 693 ms	34 436 ms

Tab. 2. Comparison of performance for different number of groups.

This algorithm builds the solution recursively by the procedure COMPUTE. The argument n is the number of groups to be created, $\mathbf{V_s}$ is a set of speeds from which groups are created, $\mathbf{G}$ is a solution created so far and $value$ is a value of the criterion function $C(\mathbf{G})$ for this partial solution.

The function $possible(\mathbf{V_s}, n)$ on line 14 returns a set $\mathbf{V_s}$ reduced of first $n-1$ minimal elements. This function ensures that for each $\underline{v}$ at step n there is still $n-1$ elements $v < \underline{v}$ that can create groups $g_1, ..., g_{n-1}$, even if that groups are at size $|g_i| = 1$.

The variable $value$, which is handed to each call of procedure COMPUTE, stores the value of the function $C(\mathbf{G})$ for groups created so far; i.e. only the criterion for a group g_n created at the step n needs to be computed.

The algorithm 2 can be enhanced by the branch and bound method [5]. The improvement is straightforward – in the beginning the procedure COMPUTE (line 5) a test is added, whether $value \geq bestValue$. If this condition is true, the partial solution created so far cannot outperform the best solution found so far and there is no point in exploring current branch of the solution. In this case, the algorithm simply returns from the procedure. If the condition is false, there is still chance that the current explored branch could contain better solution and the procedure continues its execution.

5. Theoretical Evaluation

In this section we examine results of the algorithms for given dataset. In section 6 we evaluate the computed optimal schedules in the multi-agent testbed AgentC.

The first part of the evaluation is a performance comparison of naive, dynamic and dynamic with branch and bound method (B&B) algorithms. Each algorithm was executed 300 times with the same dataset. The results are in the Table 2. Times are mean values of all run times for given configuration. Although each implementation has worse than exponential time complexity, the dynamic and B&B algorithms are clearly faster than the naive algorithm. From comparison of dynamic and B&B algorithms it is obvious, that B&B reduces computation time approximately twice. This allows us to compute optimal group transit schedules up to 15 groups.

The second part of evaluation is a delay comparison of optimal group schedules with the current schedule. Algorithms were executed with the dataset presented in sec-

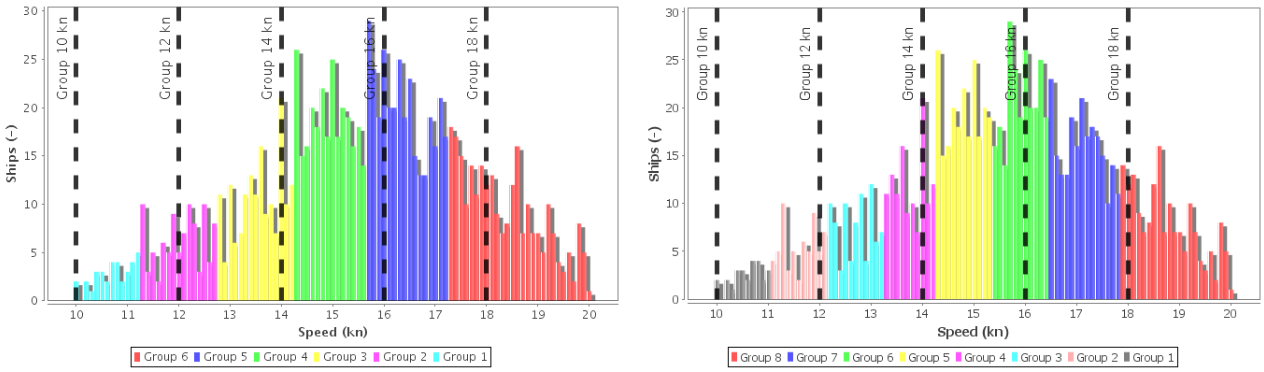

Fig.2. Optimal groupings for six (left) and eight (right) speed levels. Current speed group levels are marked by dashed lines.

scheme	ΔT/ship	ΔT/year	Savings/year
current	0 min	0 days	0 USD
opt. 2	-3 h 40 min	-3061 days	-76 525k USD
opt. 3	-1 h 26 min	-1207 days	-30 175k USD
opt. 4	-27 min	-388 days	-9 700k USD
opt. 5	7 min	111 days	2 775k USD
opt. 6	32 min	457 days	11 425k USD
opt. 7	48 min	667 days	16 675k USD
opt. 8	59 min	825 days	20 625k USD

Tab. 3. Time and money savings of transit groups for selected grouping schemes. **ΔT/ship** is the average transit time reduction compared to the current scheme for one ship, **ΔT/year** is the aggregated average time reduction for all the ships for one year and **Savings/year** express the amount of US dollars saved by reducing the Group Transit delays. Note that the optimal schedules for 2 and 3 speed levels are impractical and are presented only for the completeness of results.

tion 2.1. We have computed optimal solutions for 2 to 8 groups. The optimal solutions for six and eight groups are depicted on the Figure 2.

The results are summarized in the Table 3. All data are relative to the current group transit scheme. **ΔT/ship** is the average time reduction per one transit, whereas **ΔT/year** is time reduction per year. We assume, that 20,000 ships sail through GoA per year [6]. **Savings/year** data were calculated considering cost 25,000 USD for a day of sail [6].

From the Table 3 we can see that for 5 groups, the optimum scheme is close to the current scheme, saving 7 minutes per ship in average. However even a small change – i.e. adding one speed group level – can reduce the average time for half an hour and save more than 10 million US dollars while keeping the number of attacks constant (see Section 6).

Average delays for one ship in minutes for different transit schemes are depicted on the Figure 3 (on the left). Delays are the difference between ships' time in the Group Transit and the time when sailing according to her individual schedule. These delays do not consider a delay while the ship approaches the entry point and needs to reduce her speed in order to arrive in the scheduled time.

6. Experimental Evaluation

The experimental evaluation was performed in *AgentC* – a multi-agent simulation of the maritime domain [7]. AgentC is a software testbed that allows simulating maritime traffic using the multi-agent approach. The testbed includes models of various types of vessels such as merchant ships, pirate boats and navy warships with their helicopters. These vessels are able to interact with each other in order to simulate the real world domain. The testbed uses a Finite State Machine-based behavioral models [1] to express the rich behavior of each vessel, which allows us to combine different transit schedules and consider different types of pirates. For the overview and description of the AgentC testbed, see [3, 4].

The simulation was run for the current transit scheme and for optimal transit schemes with number of 2 to 8 groups. For each transit scheme, we run 100 of simulation runs, each at length of 70 days of simulation time. The number of transport ships was set to correlate with real world numbers – every day about 50 ships enter GoA from each side. Three naval warships were present to protect the merchant ships.

For each simulation run, we have calculated the number of hijack attempts and the number of successful hijacks. We considered only events in the observed area of Gulf of Aden. The mean values of the total number of attacks and successful hijacks for each transit are depicted on Figure 3 in the middle and on the right respectively.

It is apparent from charts on the Figure 3, that while increasing the number of speed levels, the number of hijacks remains approximately constant, whereas the number of attempts varies. These trends can be explained by two counter-effects: (1) as the number of speed levels increases, there are more groups present in GoA. The pirates have thus higher probability to localize a ship and attack it. However, (2) with the increasing number of speed levels, the average speed of the transiting ships rises (to their optimal, non-reduced value) thus making the attack less probable to succeed. These two trends can be roughly observed in the Figures, however, the interaction between all agents is non-

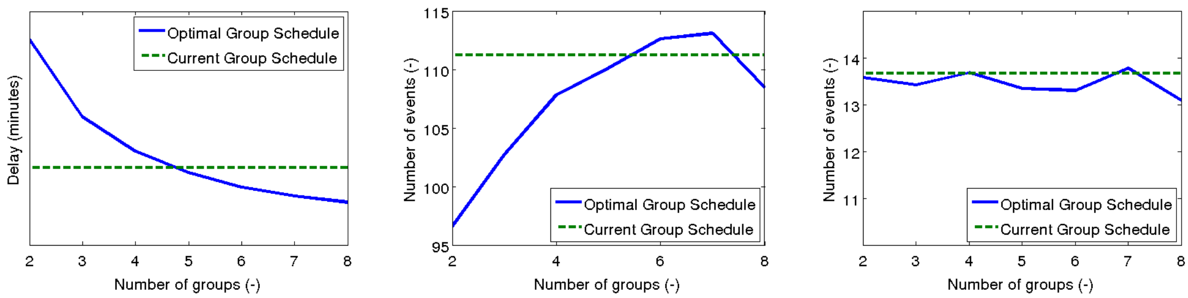

Fig.3. Evaluation of different group transit schemes. The left chart depicts the average delay in minutes. The middle chart depicts the number of pirate attempts and the right chart depicts the number of hijacks.

linear and complex, thus complicating the reasoning. Better understanding of nature of this phenomenon is subject to future research.

7. Conclusion

In this work, we deal with the critical situation in the Gulf of Aden, where the transiting ships – following a narrow corridor and synchronized according to a Group Transit schedule – are frequently attacked by Somali pirates. We aim to optimize the Group Transit schedules by means of dynamic programming and propose a set of alternatives that allow ships to cross the Gulf faster, thus saving time and money. We have evaluated the proposed schedules in a rich simulated maritime environment AgentC and we have shown the superiority of our solution, saving millions of US dollars per year.

In the close future, we will explore more advanced techniques for optimal grouping, taking into account not only speeds of ships, but also their time of arrival. We want to employ cooperative game theory and multi-agent negotiation techniques to search for an optimal solution and evaluate all the results in the simulation of the maritime domain.

Acknowledgements

The work presented is supported by the Office for Naval Research project no. N00014-09-1-0537 and by the Czech Ministry of Education, Youth and Sports under Research Programme no. MSM6840770038: Decision Making and Control for Manufacturing III.

References

[1] P. ADAMCZYK. The Anthology of the Finite State Machine Design Patterns. online, 2003. `http://hillside.net/plop/plop2003/Papers/Adamczyk-State-Machine.pdf`.

[2] R. BELLMAN and R. C. S. M. CALIF. *The theory of dynamic programming*. Defense Technical Information Center, 1954.

[3] M. JAKOB, O. VANEK, and M. PECHOUCEK. Using Agents to Improve International Maritime Transport Security. *IEEE Intelligent Systems*, 26(1):90–96, january 2011.

[4] M. JAKOB, O. VANĚK, B. BOŠANSKÝ, O. HRSTKA, and M. PĚCHOUČEK. Adversarial Modeling and Reasoning in the Maritime Domain Year 2 Report. Technical report, ATG, CTU, Prague, 2010.

[5] E. LAWLER and D. WOOD. Branch-and-bound methods: A survey. *Operations research*, 14(4):699–719, 1966.

[6] B. TSVETKOVA. Securitizing Piracy Off the Coast of Somalia. *Central European Journal of International & Security Studies*, page 44.

[7] O. VANĚK, M. JAKOB, O. HRSTKA, and M. PĚCHOUČEK. Using Multi-Agent Simulation to Improve the Security of Maritime Transit. In *Proceedings of 12th internation Workshop on Multi-Agent-Based Simulation*, 2011.

About Authors...

Ondřej HRSTKA holds bachelor degree in Cybernetics and Measurement from FEE, CTU in 2010 where he now pursuits master's degree in Artificial Intelligence. He also works at Agent Technology Center at the Department of Cybernetics, FEE, CTU, where he focuses on optimization in multi-agent systems.

Ondřej VANĚK graduated in Technical Cybernetics from FEE, CTU in 2008, and now he is a PhD student at the Agent Technology Center at the Department of Cybernetics, FEE, CTU. His main research is focused on machine learning in multi-agent systems, cooperative and non-cooperative game theory and coordination and cooperation in multi-agent systems.