

混雑海域における衝突回避問題へのアニーリングアルゴリズムの適用

Applying an annealing algorithm to the collision avoidance problem in a congested straits

白井菖太郎*
Shotaro Shirai

八木武尊*
Hotaka Yagi

新保潤*
Jun Shimbo

Abstract

海運における大型タンカーが絡む衝突事故は大きな経済的損失を生むだけでなく、燃料の重油の流出に伴う環境汚染も引き起こす。特に、マラッカ・シンガポール海峡などの混雑海域においては多くの人的コストを支払って衝突回避のオペレーションが行われているため、衝突回避問題を効率的に解くことは需要があると考えられる。本論文では、衝突回避問題を組合せ最適化問題として定式化し、アニーリングマシンを用いて近似解を求める。また、既存の整数計画問題ソルバとの解の質、計算時間の比較からアニーリングマシンを用いてこの種の問題を解くことに一定の優位性があることを示す。

1 はじめに

海上輸送は衣食住やエネルギーにまつわる物資を海をまたいで運ぶ輸送手段であり、特に日本は島国であることから、貿易に占める海上貨物の割合はトン数ベースで、輸出入合計の 99.6% を占めており、海上輸送は貿易に不可欠である [1]。世界各地には海上輸送において利点のある海峡や大都市に隣接した港など、多数の船舶が集中する海域が存在し、そのような海域では船舶同士の衝突事故が度々発生している [2]。また、衝突事故による原油や燃料の流出による環境汚染も大きな問題となっている。現在、このような衝突事故を未然に防ぐため、経験を積んだ航海士がレーダーや AIS (Automatic Identification System: 船舶自動識別装置) 情報を元に航路を決定し、衝突回避を行っている。しかし、マラッカ・シンガポール海峡などの同時に多数の船舶が通過し、地理的にも狭く水深の浅い部分が存在する海峡では船舶の密度が上昇し、衝突回避は非常に複雑な問題となる [3]。

本稿ではこの問題を混雑海域に進入するタイミングを調節して、船舶の希望する日程からの遅れ時間を最小にしつつ船舶の衝突を回避する組合せ最適化問題として定式化を行った。巡回セールスマン問題などの組合せ最適化問題は NP 困難問題と呼ばれ、従来の計算機では効率的に解を求めることができるアルゴリズムは発見されていない。しかし、1998 年に提案された量子アニーリング [4] は組合せ最適化問題とイジングモデルといわれる物理モデルに対応付ける

ことにより、効率的に解くことができる。この理論に触発され、量子的な効果を用いた D-Wave 社のマシンや、シミュレーテッドアニーリングアルゴリズムを実行することに最適化された、富士通株式会社、株式会社日立製作所、株式会社フィックスターズなどの量子性を用いない専用機を利用可能なサービスが登場している。

そこで、マラッカ・シンガポール海峡における衝突回避問題を定式化し、アニーリングマシン、もしくは既存の線形計画問題ソルバ CPLEX(IBM)[5] を用いて解き、得られた実行可能解の品質の比較を行った。

2 問題設定

マラッカ・シンガポール海峡に注目することによって、他の海峡や港で本手法が適用可能かどうかの判断が下せると考え、この海峡で問題を作成する。

各船舶の進入予定時刻から、どのくらい時間を遅らせれば良いかを、管制から各船舶に伝えるという想定で問題を作成し、海峡を通過する船舶の船速・サイズ・進入予定時刻・進入地点は、AIS や無線などによって事前に全てわかることを仮定した。また、マラッカ・シンガポール海峡の特に混雑する東側では、通行レーンが設定されており、その上を船舶は航行することになっている。そこで私たちは、それを参考にルートを決定した。[3]

海峡へは船舶が絶えず入ってくることを踏まえて考えると、最適化対象である海峡へ入るタイミングを操作できるのは、新しく海峡に入ってくる船舶のみである。すでに海峡に入っている船舶は最適化済みなので衝突はしない。そ

* 東京理科大学 Tokyo University of Science

のため、新しく入ってくる船舶が海峡内の船舶と衝突せず、さらに新しく入ってくる船舶同士で衝突しないような海峡へ入るタイミングを求めることとする。

3 定式化

今回この衝突回避問題を 0-1 整数計画問題として捉え、Quadratic Unconstraint Binary Optimization (QUBO) 形式で定式化した。[6] QUBO 形式とは、 Q_{ij} を実数値として以下の式で与えられる目的関数の最適化問題のことである。

$$\mathcal{H} = \sum_{i < j} Q_{ij} x_i x_j, \quad x_i \in \{0, 1\} \quad (1)$$

アニーリングマシンは、 \mathcal{H} を最小化する決定変数 $\{x_i\}$ の組み合わせを探索するように動作するマシンである。今回は変数 $\{x\}$ を各船舶の遅延とし、制約として (1) 求めた遅延が一意にデコードできるための制約、(2) 衝突数を 0 にするための制約の 2 つを用い、目的関数としては遅延の総和を用いた。なお、船舶数は $i = 1, \dots, N_v$ 、遅延時間を $j = 1, \dots, N_d$ で表し、 $\{x_{ij}\}$ は $N_v \times N_d$ 個用意する。この $\{x_{ij}\}$ が qubit に対応する。なお、整数計画問題ソルバを用いるときは制約項、目的関数をそれぞれ与えて使う。

3.1 エンコーディング項

一つ目の制約は、解として得られた遅延が一意にデコードできることを保証するための制約である。これは以下のように表すことができる。

$$f_{\text{encoding}} = \sum_{i=1}^{N_v} \left(\sum_{j=1}^{N_d} d_{i,j} - 1 \right)^2 \quad (2)$$

これは各 i について、ただ一つの j について $d_{i,j} = 1$ となり、それ以外は 0 となるような場合にのみ最小値 0 を取ることがわかる。つまりは、各船について 1 となるビットは一つだけということである。このことから、各船の遅延は一意にデコードができることがわかる。

3.2 衝突項

まず、船舶同士の衝突を定義する。船舶 i の optimal なルートは点列 $\{x_{i,s}\}_s$ で表すこととし、各 $x_{i,s}$ は時間 s に船 i がいる点である。また、 Δ, Δ_t は空間的、時間的に開けるべき間隔のことであり、 δ_i, δ_j はそれぞれ船 i, j の進入予定時刻からの遅れ時間とする。

定義 1 次の二条件を満たす時、二点 $x_{i,s}, x_{j,t}$ は衝突しているという。

1. $\|x_{i,s} - x_{j,t}\| < \Delta$
2. $|(s + \delta_i) - (t + \delta_j)| < \Delta_t$

定義 2 次の二条件を満たす時、二点 $x_{i,s}, x_{j,t}$ は潜在的衝突しているという。

1. $\|x_{i,s} - x_{j,t}\| < \Delta$
2. $\max\{\delta_i^*, \delta_j^*\} + \Delta_t < |s - t|$

ただし、 δ_i^*, δ_j^* はそれぞれ船 i, j の最大可能遅れ時間を表しており、 $\delta_i \leq \delta_i^*$ かつ $\delta_j \leq \delta_j^*$ である。また、潜在的衝突する添字の組の全体の集合を $C = \{(i, s), (j, t) | x_{i,s}, x_{j,t} \text{ は潜在的衝突}\}$ と表す。

一つ目の衝突は実際の衝突を表しており、二つ目の潜在的衝突は、各船舶にどんな遅れを与えても新たな衝突が発生しないようになっている。今回は、潜在的衝突する点を 0 にするように遅れ時間を選ぶ。

定義 3 潜在的衝突する添字の組の全体の集合 C を次のように分割できる。

$$C = \bigcup_k C_k \quad (3)$$

詳しくは [6] を参照されたい。

そして、

$$B_k = \bigcup_{\{(i,s),(j,t)\} \in C_k} (-\Delta_t + t - s, \Delta_t + t - s) \quad (4)$$

とすれば、 C_k は $d_i - d_j \notin B_k$ で回避できる。

以上より、衝突を 0 にするような制約は次のように表すことができる。

$$f_{\text{conflict}} = \sum_k \sum_{l, l' | \Delta_d(l' - l) \in B_k} d_{i,l} d_{j,l'} \quad (5)$$

ただし、 Δ_d は遅延の単位時間である。例えば、 $\{d_{i,l}\}_l$ のうち、 d_{i,l_0} のみが 1 になっていれば $d_i = \Delta_d l_0$ となる。複数の $d_{i,l}$ が 1 になる場合には、式 2 が 0 にならないので解にならない。よって $f_{\text{encoding}}, f_{\text{conflict}}$ が 0 になれば、全ての k について $d_i - d_j \notin B_k$ となり、全ての衝突が回避される遅延を得ることができる。

3.3 目的関数

目的関数は各船舶に対する遅延の総和であり、以下のよう表すことができる。

$$f_{\text{delay}} = \sum_{i=1}^{N_v} \sum_{l=1}^{N_d} l d_{i,l} \quad (6)$$

l はビットのインデックスに対応しており、 $1, \dots, N_d$ までの値をとる。なお、 f_{delay} は 0 である必要はない。

4 実験条件

実験をする際の問題設定を説明する。船舶の隻数は、海峡の内外含め 10~100 隻とし、船速はすべて 15 knot、船のサイズは衝突判定領域を 1.1 km (7 割)、2.1 km (3 割) と分け、船の大きさの違いを反映した。進入予定時刻は 1 日の中から乱数を用いてランダムに生成し、入場地点・到着地点に関しては予め決めてある航路 18 通りの中から乱数を用いてランダムに選択した。用いたアニーリングマシンは、GPU アニーラ (フィクスターズ)、Digital Annealer (富士通) である。また、比較用として用いる CPLEX は、CPU: Intel core i7-8700 3.4GHz, メモリ: DDR4-2666 16GB のマシン上で動作させる。

評価基準として私たちは Time to feasible solution (TTFS) と Average delay を用いた。以下、それぞれについて説明する。まず、TTFS についてであるが、これは実行可能解が求まるまでに必要なアニーリング時間の目安である。ここで実行可能解とは、3 章で説明した制約式 (2)、(5) を満たし、式 (6) に関しては最小とは限らない解のことを指し、必ずしも最適解ではない。さらに、TTFS の具体的な表式について説明をする。

アニーリング時間 τ で求解を行い、その試行を行った回数を N 、その中で実行可能解が得られた個数を N_s とすると、実行可能解が得られる確率 $p_s(\tau)$ は $p_s(\tau) = N_s/N$ となる。また、求解を R 回繰り返し 1 つ以上実行可能解が得られる確率 p_R は以下の式で表される。

$$p_R = 1 - (1 - p_s(\tau))^R \quad (7)$$

この式を R について解くと以下の式が得られる。

$$\text{TTFS} = \tau R = \tau \frac{\ln(1 - p_R)}{\ln(1 - p_s(\tau))} \quad (8)$$

今回用いた TTFS では $p_R = 0.99$ とした式を用いた。上記で述べた通り、TTFS とは実行可能解を求めるのに要する時間の目安であり、その時間だけ求解を行うと、99% の確率で実行可能解が得られると解釈することができる。また、次章で行った実験では、成功事象として最適解のみをカウントする場合は TTS (Time to solution) の表記を用い、さらにその横に時間が指定されている場合は最適解から指定した時間分の余分な遅れ時間が課されている解までを成功事象とカウントした場合の TTS となっている。次に、Average delay について説明する。一度実行可能解が得られるとそれに応じて各船舶に遅れ時間が割り当てられる。この遅れ時間の総和を問題に含まれる船舶数で割ったものが Average delay である。また、比較対象した CPLEX では決定論的

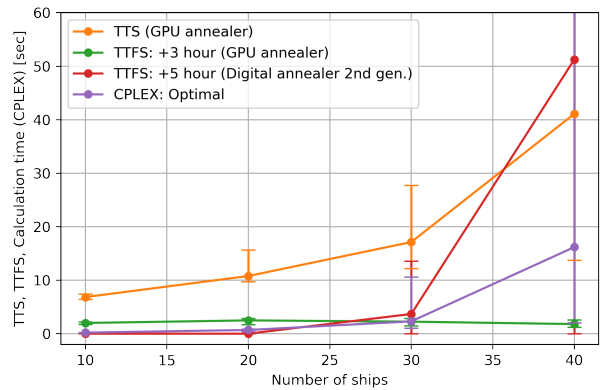


図.1: 10 種類のシードについての TTS または TTFS。横軸は船舶の数、縦軸は TTS または CPLEX の計算時間である。高低線は最大最小値を示す。

に解が求まるため、TTFS のような指標を定義することが難しい。そこで、CPLEX に関しては解を得るまでに要した時間をそのまま TTFS に対する比較対象として用いた。さらに、CPLEX では、デフォルトでは最適解が見つかるまで解を探し続けるが、問題の規模が大きくなると実用的な時間内で最適解を求めることができない。そこで、CPLEX の解の更新回数に制限をかけ 20 回もしくは 100 回、式 (6) の値がより小さい解へと更新が行われたところでその時点での出力を実行可能解として採用することとした。各船舶数の問題は 10 種類のシード値をもとに生成した。

5 実験結果

上記の条件の下で実験を行った結果を示す。まず、10 隻から 40 隻までの、CPLEX を用いて最適解が現実的な時間 (数分以内) で求まる範囲において解を求め TTS または最適解から平均して 3 時間、5 時間の余分な遅れを認めた場合の TTFS と CPLEX の計算時間の比較を行った。その結果を図 1 に示す。GPU アニーラの TTS: 0hour と CPLEX の最適解が求まるまでの計算時間を比較すると CPLEX の計算時間がより小さくなっていることが読み取れる。GPU アニーラについて、3 時間の余分な遅れまでを認めることにすると、短時間で実行可能解が求まっているが数秒程度の差でしかないため、10 隻から 40 隻までの問題規模では CPLEX を用いて最適化を行うことが望ましいと考えられる。

次に、10 隻から 100 隻までの間で船舶数を変化させて、求まった解の平均遅れ時間を 10 種類のシードについて平均を取った結果が図 2 である。ここでは、最適解、実行可能解を区別していないため、統一して TTFS の表記を用いている。この結果から実験を行った環境においては 80 隻程度の問題規模以上では GPU アニーラを用いたほうがより、高精度な (Average delay が小さい) 実行可能解が求まること

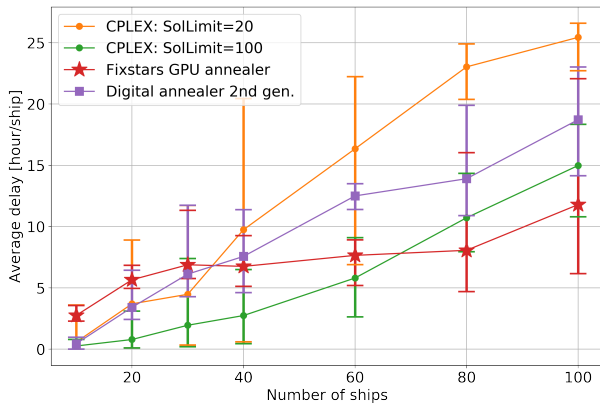


図.2: 10 種類のシードについての平均遅れ時間。高低線は最大最小値を示す。

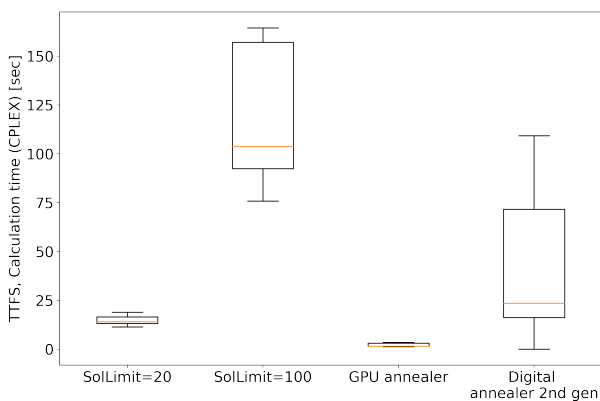


図.3: 100 隻の問題について 10 種類のシードの TTFS または CPLEX の計算時間を箱ひげ図で示したものが図 3 である。この図 3 から 100 隻程度の規模の問題になると、現環境では GPU アニーラが CPLEX より高速に、より精度の高い解を求めることができることが分かった。

分かった。さらに、実行可能解を求めるのにかった時間を比較するため、100 隻の規模の問題について TTFS または CPLEX の計算時間を箱ひげ図で示したものが図 3 である。この図 3 から 100 隻程度の規模の問題になると、現環境では GPU アニーラが CPLEX より高速に、より精度の高い解を求めることができることが分かった。

6 結論

本論文では、衝突を回避しながら混雑海域を通過する時間を最小化するという問題を扱った。私たちは、問題を解く方法として既存の整数計画問題ソルバ (CPLEX) を用いる方法とアニーリングマシンを用いる方法を TTFS と Average delay によって比較した。今回の問題設定では、船の隻数が 40 隻程度の問題では CPLEX がより高速に最適解を求めることが分かった。一方で、100 隻程度の問題規模となると解の質、求解に掛かる時間ともにフィックスターズ社の GPU アニーラの方が CPLEX に比べて勝ることが分かった。

マラッカ海峡では 1 日に 300 隻程度の船が海峡を通過す

ることを考えると、アニーリングマシンを利用して混雑海域における衝突回避問題を解くことにメリットがある。しかし、船の隻数が 300 などさらに多くなる問題では解が求められないことが多い。その状況に対応するため、以下のアプローチが考えられる。まず、混雑海域内での衝突箇所を衝突しやすい船の合流地点に絞ることで、回避する衝突の数を減らすことで解の探索空間を狭めたり、解空間に解が存在するように航路の自由度をさらに増やすなどが挙げられる。また、ある程度の衝突を許容して航路を局所的に変更し、衝突を回避する方法も考えられる。さらに、ハミルトニアン加重が解に大きな影響を与えることが経験的に分かっており、隻数が大きくなると問題ごとにハイパーパラメータをチューニングする必要がある。このチューニングを問題に合わせて素早く求められるような機構を作ることも今後の課題であると考えられる。

また、本論文で議論されていた問題設定を元に作ったアプリケーションを以下の WEB サイトに公開している。
<https://deconfships.dev/mitouapp/>

文献

- [1] 公益財団法人日本海事センター, “Shipping now 2019-2020[データ編]”, (2019).
- [2] 種市雅彦, “世界における日本の海難”, 日本航海学会誌 NAVIGATION **196**, 89 (2016).
- [3] 関根博, “石油輸送の生命線 マラッカ海峡航行: 現状と問題点”, (2006).
- [4] T. Kadowaki and H. Nishimori, “Quantum annealing in the transverse ising model”, Phys. Rev. E **58**, 5355 (1998).
- [5] “IBM ILOG CPLEX”, <https://www.ibm.com/jp-ja/products/ilog-cplex-optimization-studio>.
- [6] T. Stollenwerk, B. OGorman, D. Venturelli, S. Mandra, O. Rodionova, H. Ng, B. Sridhar, E. G. Rieffel, and R. Biswas, “Quantum annealing applied to deconflicting optimal trajectories for air traffic management”, IEEE Transactions on Intelligent Transportation Systems **21**, 285 (2020).

謝辞

本研究に取り組むにあたり、ご助言頂いた田中宗先生、アニーリングマシンを提供して頂いた株式会社フィックスターズ様に厚く感謝を申し上げます。本研究は情報処理推進機構未踏ターゲット事業の助成を受けています。