

Application of Stochastic Optimization to Collision Avoidance

Rebecca Burnett

Johns Hopkins University Applied Physics Laboratory
rebecca.burnett@jhuapl.edu

Abstract

This paper applies simulation-based optimization to the problem of vessel traffic management in a high vessel density environment. Specifically, a Monte Carlo simulation has been developed that models a relatively small craft operating in a high vessel density environment under poor visibility conditions. In this simulation, the small vessel maneuvers to keep all other vessels outside some acceptable range (one of the objectives of the optimization) subject to the requirements on the other vessels to obey pre-established traffic management rules. Several stochastic optimization algorithms- Blind Random Search, Simultaneous Perturbation Stochastic Approximation, and Simulated Annealing- are applied to this problem with conclusions drawn regarding their relative applicability and performance, as well as the practical implications of the results.

1. Introduction

Vessels must operate every day in stressing situations and it is of paramount importance that these operations be completed without incurring collisions with other vessels. In response to this concern, a Monte Carlo simulation has been developed that models a relatively small craft (OwnShip) operating in a high vessel density environment with poor visibility conditions. OwnShip maneuvers to keep all vessels outside some acceptable range. Its ability to do so is dependent on a number of parameters θ . Some of these parameters are constrained by things outside of OwnShip's control, like the ranges at which other vessels are detected, and others are driven by its tactics. Thus far, the effects of these parameters on OwnShips' ability to avoid collisions have been studied by running the model a number of times for different values of θ to obtain a measure of mean performance for each. Note that because of the randomness in the simulation, the effect of a particular value of θ

on collision avoidance cannot be obtained from only one evaluation of the model at that θ . While averaging over multiple runs is a valid approach, it is time consuming and so it will be impossible to consider all possible values of θ . This results in a significant risk that valuable information will not be uncovered. Algorithms designed for optimization in the presence of noise, i.e. stochastic optimization algorithms, should be used to find the optimal set of parameters θ^* for this problem.

This paper will first discuss the simulation mentioned above and highlight some of the simplifying assumptions that were made in order to model reality. Next, the problem will be placed in a stochastic approximation framework, where explicit definitions of the loss function, the constraints of the problem, and the parameters of interest will be given. Although the deterministic form of the loss function is unknown, this section will include some analysis of its expected characteristics. The following section will provide some insight into the properties of the measurement noise. These analyses are important to understanding which stochastic optimization algorithms can be expected to converge to the optimal solution. Finally, the results of applying several algorithms, Blind Random Search, Simultaneous Perturbation Stochastic Approximation, and Simulated Annealing will be presented and compared.

2. Model Assumptions

In each run of the Monte Carlo simulation OwnShip will be a small craft, traveling at a constant speed of 5 knots, with a radar that is not functioning. As a result of the non-functioning radar, other vessels in the area can only be observed visually. Further, the visibility conditions are assumed to be poor, either simply because it is night or as a result of inclement weather, such as fog. There are a number of other vessels in each simulation, the majority of which are fishing crafts with speeds ranging

from 0 to 10 knots, but also including a small number of large merchant ships traveling at a constant speed of 15 knots.

OwnShip maneuvers to attempt to maintain a buffer zone of R kyds (1 kyd = 1000 yards) between itself and all other vessels. These maneuvers will be course changes under the assumption of a full rudder. All other vessels change course to avoid imminent collisions with each other based on the navigational rules of the road. However, since OwnShip is a small craft and visibility is severely restricted the other vessels in the simulation will not detect Ownship and will only react to each other. Trawling fishing vessels, those traveling at 5 knots, will also periodically change course.

The simulation is defined over a circular region with a radius of 20 kyds and each run captures 90 minutes of interaction between OwnShip and the other vessels. The number of other vessels in the simulation region N remains constant throughout each run and is taken as 25 in this study. This is roughly equivalent to one vessel in each circle of radius 4 kyds. Each time a vessel leaves the region a new vessel of that type (speed) will be entered into the simulation at a random bearing on the boundary of the simulation circle and will be assigned a random course.

Vessels are each assigned an initial position and course randomly, and a speed according to the proportions stated previously. In order to simulate the scenario in which OwnShip has just found itself in a stressing environment, i.e. radar was just lost, it is assumed that moving vessels start at least M_0 kyds from OwnShip. Those vessels traveling at a speed of 5 knots are assigned an initial (first in the run) time since last course change at random from the interval 0 to 60 minutes and without loss of generality, the initial OwnShip course is due North.

To simulate the poor visibility conditions caused by either night or fog, OwnShip detects other vessels at half of the range corresponding to perfect visual detection on the horizon. Under the assumptions of mast height used in this study, detection ranges for fishing vessels and merchants are 6.5 and 10.5 kyds, respectively. Upon detection, perfect knowledge of each vessel's position and velocity will be assumed. The Closest Point of Approach (CPA) will be projected between OwnShip and each of the other vessels. These projected CPAs will determine the action taken by the OwnShip. The projected CPA between OwnShip and each of the N vessels, $C_1 \dots C_n$ is based on the current

motion of each. If all vessels have projected CPAs outside of the R kyd buffer in the next 30 minutes, then OwnShip will remain steady on course. It will only implement its maneuvering tactics when this condition is not satisfied.

3. Definition of Theta and the Loss Function

The set of input parameters to be varied will be given by $\theta = [R, M_0]$ where R is the buffer range that OwnShip is trying to maintain and M_0 is the minimum initial range to all moving vessels. The domain of θ will be defined as the hypercube $\Theta = [2, 8]^2$. In practice, the lower bounds of R and M_0 will be defined by the "cavalierness" of the crew, whereas the upper bounds on both are dictated by visibility conditions.

After each run of the model, one scalar valued loss function measurement will be collected - the negative of the minimum range between OwnShip and any vessel during the run. Safety of OwnShip (and the other vessels) depends upon this range being sufficiently large. Thus the goal of this analysis will be to find the set of input parameters θ that minimize the loss function $L(\theta)$, subject to $\theta \in \Theta$. Note that the deterministic form of $L(\theta)$ is unknown and no gradient information is available. There are only noisy loss function measurements, $y(\theta) = L(\theta, \varepsilon(\theta))$, with the noise $\varepsilon(\theta)$ resulting from the randomness in each run of the simulation. In addition, it is possible that $L(\theta)$ may have several local minima in the domain Θ in addition to one global minimum.

4. Characteristics of the Loss Function

Some analysis of $L(\theta)$ is necessary. Consider a particular run and the Euclidean distances between OwnShip and each of the vessels in that run. Let the distance function for the vessel that came closest to OwnShip during that run be given by $d(t)$. Note that the number of distance functions resulting from one run will in general be some number $\tilde{N} > N$ since each time a vessel leaves the simulation area a new one is entered. Further, note that each distance function is continuous and differentiable, regardless of any changes to θ . The following paragraphs discuss the effects of changes in θ on $L(\theta)$.

The parameter R governs the behavior of OwnShip. If any vessels are projected to come

within the buffer of R kyds then OwnShip considers all possible course changes and chooses the one that maximizes the minimum CPA, i.e. it attempts to maximize the loss function with each maneuver. For values of R that are sufficiently small, OwnShip should be fairly successful and so $E[y(\theta)] \approx R$. However, recall that the vessels are distributed uniformly in position, not in distance from OwnShip. Let D' be a random variable representing the distance from OwnShip to a particular vessel at time t . Consider a particular distance r .

$$P(D' \leq R) = F(r) = \pi r^2 / \pi 20^2$$

$$P(D' = R) = f(r) = F'(r) = 2r / 20^2$$

For the moment, disregard the effect of M_0 on vessel density. Note that this is a relevant simplification since as time progresses the randomness in the simulation should eliminate the effect of the M_0 and the positions of the vessels will reach a “steady-state” where they are following a uniform distribution. Let x'_R represent the number of vessels within R kyds of OwnShip at time t . Then $E[x'_R] = N \cdot F(R) = N(R^2/20^2)$ during the “steady-state” portion of each run. Thus as R increases, the vessel density will increasingly impede OwnShip’s ability to maintain this buffer. Note further that although adjustments to R affect OwnShip’s tactics, they have no effect on the randomness in the behavior of the other vessels. The other vessels are completely unaware of the presence of OwnShip and only react to each other.

The initial vessel positions also affect $L(\theta)$. Previous analyses of the model have shown that at the start of each simulation the probability of a close encounter is higher than in the later, “steady-state” portion. Thus, large portions of the loss function measurements result from the first part of each run. One reason for this is that the initial positions of the dead-in-the-water (DIW) vessels can be arbitrarily close to OwnShip. However, since only 5% of all vessels in each run are of this type the expected number of DIW vessels in each simulation N_{DIW} is only 1.25. It follows that the effect of these vessels on the loss function should be quite negligible. The primary cause of the spike in close encounters during the first portion of each run is that OwnShip has not yet had the chance to position itself optimally with respect to the moving vessels. At the start of each simulation the constraint on moving vessels results in a higher vessel density outside M_0 and only DIW vessels

inside M_0 . Let $x_R^0(M_0)$ be the number of vessels inside R kyds at time 0, given M_0 . Then $E[x_R^0(M_0)]$ is given by

$$N_m \frac{F(R) - F(M_0)}{F(20) - F(M_0)} + N_{DIW} F(R) \text{ for } R > M_0$$

$$\text{and } N_{DIW} F(R) \text{ for } R \leq M_0$$

where N_m represents the expected number of moving vessels in the simulation. This increase in vessel density outside M_0 in turn affects the interaction between the vessels, as they must change course more often to avoid colliding with one another. So the combined affect of M_0 is not straightforward. As M_0 increases, the initial positions of the vessels move farther away from OwnShip, giving it more time to position itself optimally. But the randomness in the vessel motion also increases as M_0 increases, which could make it more difficult to maintain a safe distance.

Clearly $L(\theta)$ must either come from the “steady-state” portion or the first part of each run. Let the prior discussion regarding the level of difficulty in keeping all vessels outside of R be represented by the function Δ , where $\Delta(x_R^0(M_0)) < 0$ for all θ in Θ . Thus under the assumption that the effect of vessel density on the level of difficulty is continuous, i.e. Δ is continuous, $L(\theta)$ must also be continuous.

If $M_0 \geq R$, then $L(\theta)$ will most likely result from the steady-state portion of the run. However if $M_0 < R$ and if there are other vessels within R at time 0, then these vessels will most likely result in loss function measurements. Let the expected value of the distance from OwnShip to vessels that are between M_0 and R kyds from OwnShip be $E[D_{M_0,R}]$. Then the expected value of the loss function is given by $E[y(\theta)] = R + E[\Delta(x_R(M_0))]$ if $M_0 \geq R$ and if $M_0 < R$, it is simply represented by $E[y(\theta)] = P(\text{at least } 1 d_i(0) < R)E[D_{M_0,R}]$

$$+ P(\text{all } d_i(0) \geq R)(R + E[\Delta(x_R(M_0))])$$

$$= \left[1 - \frac{1 - F(R)}{1 - F(M_0)} \right]^{N_m} \int_{M_0}^R r f(r) dr$$

$$+ \left[\frac{1 - F(R)}{1 - F(M_0)} \right]^{N_m} (R + E[\Delta(x_R(M_0))])$$

Thus it seems that $L(\theta)$ is continuous which provides some hope that the stochastic optimization algorithms will converge to an optimal θ^* . But θ^* may not be unique and there

is still the complicating factor of noisy loss function measurements.

5. Properties of Measurement Noise

As stated previously, each loss function measurement $y(\theta)$ contains noise $\varepsilon(\theta)$. Convergence of many stochastic optimization algorithms requires that this measurement noise be sufficiently well-behaved. The properties of $\varepsilon(\theta)$ were investigated by collecting the loss function values $y(\theta)$ for 1000 runs of the simulation at each of 16 different values of θ , where $\theta = [2i, 2j]$ $i, j = 1, \dots, 4$. The sample mean $\bar{y}(\theta)$ and standard deviation were then calculated for each value of θ . Note that for a particular θ , the distribution $\eta(\theta)$ resulting from the difference between each of the observed values and the sample mean should approximately represent the properties of the measurement noise at that θ .

To have a reasonable chance of convergence, it is important that $\bar{\eta}(\theta) = \frac{1}{1000} \sum_{i=1}^{1000} \eta_i(\theta)$ be very close to zero for each value of θ . Two-tailed Z-tests with the null hypothesis $H_0: \bar{\eta}(\theta) = 0$ were performed for each value of θ . In each case the data did not provide strong evidence that the sample means of the error differed from zero. It is also important to understand the relationship between $\eta(\theta)$ and θ . Figure 5 indicates that except for a slight dip around [4,6], the standard deviation of the noise increases steadily (and substantially) as θ increases. The magnitude of this standard deviation could hinder convergence to the optimal solution θ^* .

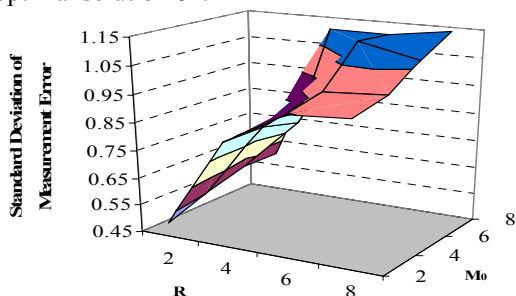


Figure 5. Standard deviation of $\eta(\theta)$

6. Application of Stochastic Optimization Algorithms

This section will provide a brief discussion of Blind Random Search, Simultaneous Perturbation Stochastic Approximation, and Simulated Annealing and then present the results of applying each of these gradient-free algorithms to this problem. Each algorithm was given the same initial condition $\theta_0 = [4, 4]$ and was allowed 40 replications with 1000 evaluations of the loss function per replication.

6.1. Blind Random Search

The first algorithm implemented on this problem was the recursive form of Blind Random Search (BRS). It is especially attractive as a first approach because it does not have a large number of parameters that require tuning, in fact in the noise-free case there are none. This algorithm only requires that measurements of the loss function, possibly noisy, be available. No gradient information is required and in fact it is not necessary that the gradient exist. Further, there is no requirement that $L(\theta)$ have a unique global minimum. The theory behind this algorithm states that if L is sufficiently smooth and there are noise free loss measurements, then this algorithm will converge to $\theta^* \in \Theta^*$ a.s., where Θ^* represents the set of values that minimize $L(\theta)$. The previous analysis of L seems to suggest that it will satisfy the requirements of this algorithm, but there are only noisy inputs available – this may be a problem.

At each iteration, a new value θ_{new} was sampled from a uniform distribution on $[2,8]^2$. Thus in this problem setting the application of this algorithm contains both characteristics of stochastic search and optimization – injected randomness in each θ_{new} and noisy inputs. θ_{new} was constrained to fall within the domain Θ and was assigned the nearest boundary value if it fell outside of Θ . In the noise-free implementation of BRS, $\hat{\theta}_{k+1} = \theta_{new}(k+1)$ if $y(\theta_{new}(k+1)) < y(\hat{\theta}_k)$. However, since each loss function measurement contains noise, this acceptance criterion must be altered to lower the probability that a new value of θ will be accepted when it is really only lower because of noise. A conservative approach is to define $\tau = 2\sigma$, where σ is the standard deviation of the measurement noise at the initial condition and accept $y(\theta_{new})$ if $y(\theta_{new}) < y(\theta_{curr}) - \tau$. It is expected that as the number of iterations k increases the number of accepted θ_{new} values will decrease. But after several initial replications with $\tau = 2\sigma$ there were only a few θ_{new} values

accepted each time, which implied that this bound was too conservative. τ was redefined as $\tau = 0.5\sigma$, which resulted in the acceptance of more candidate values and so a more reasonable rate of decay of L .

Let the region of acceptable error be given by $S(\theta^*) = [-0.1, 0.1]^2$. The probability that a particular iterate of θ is in $S(\theta^*)$ is $P^* = .0011$. Let $1-\rho$ represent the probability that θ lands in $S(\theta^*)$ within 1000 iterations. Using the fact that each value of θ is generated independently for this algorithm, we can obtain $1-\rho = 1 - e^{1000 \log(1-P^*)} = 0.67$. This indicates that in subsequent analyses, when more time is available, BRS should be implemented with more iterations in order to get a higher probability of landing in S .

6.2 Basic SPSA

The next algorithm considered was the basic, i.e. gradient-free, form of Simultaneous Perturbation Stochastic Approximation (SPSA). Again, this algorithm only requires that noisy loss function measurements be available. In fact, unlike BRS, the convergence theory for SPSA is based on noisy inputs. But there is a price to be paid for this robustness in the presence of noise and it comes in the form of more restriction on the loss function. A sufficient condition for convergence of SPSA requires (among other things) that the loss function must be three-times continuously differentiable. This may or may not be the case and could cause problems with convergence. Also note one of the most attractive features of SPSA is its efficiency in high dimensional problems. In this analysis the dimension of θ is only 2 and so that feature will not really come into play, but in subsequent analyses it could prove to be very beneficial.

The Bernoulli distribution was used for the random perturbations Δ_k . The gain coefficients were taken as $A=50$, $c=0.8$, $a=14$, $\alpha=0.6$, and $\gamma=0.1$ with the gain sequences given by $a_k = a/(k+1+A)^\alpha$ and $c_k = c/(k+1)^\gamma$. These are “semi-automatic” gain coefficients, obtained using the guidance offered in [1] on pages 189-91. Note that the choices of Δ_k , a_k , and c_k all satisfy the requirements on the gains necessary to ensure convergence for this algorithm. Recall from the section on the properties of the measurement noise that $E[\varepsilon(\theta)] = 0$, which is also required for convergence. In fact, the only possible hindrance to convergence should be that L may not be sufficiently well-behaved.

6.3 Simulated Annealing

The final algorithm applied to this problem was Simulated Annealing (SAN), which again can work with only noisy measurements of the loss function available. One other attractive feature of this algorithm is that if the rate of decay of the “temperature” is chosen properly, it has a good chance of not getting stuck in local minima. Since it is possible that there are several local minima for this problem, this may be useful.

The temperature T was given a geometric rate of decay, $T_{new} = 0.9T_{old}$ and 40 iterations were performed at each value of T . Initially a very small value was chosen for the initial temperature, but the algorithm seemed to get stuck at the initial condition of $\theta_0 = [4, 4]$ and so T_0 was increased to 1.0. A form of SAN, in which new values of θ are generated by $\theta_{new} = \theta_{old} + T*r$ where r is a random draw from $U[0,1]^2$, was used in this implementation. Both the randomness in r and the fact that $E[r] \neq 0$ enhance the algorithm’s ability to avoid premature convergence in local minima. New values of θ were then compared using $\delta = y(\theta_{new}) - y(\theta_{curr})$ via an altered form of the Metropolis Criterion, which was modified from $\delta < 0$ to $\delta < \tau$ to account for the noise in the loss function measurements. In other words, θ_{new} was accepted if $y(\theta_{new}) \leq y(\theta_{curr}) + \tau$. The choice of τ depends on the hypothesized characteristics of $L(\theta)$. If $L(\theta)$ has many local minima then τ should be positive, resulting in a more lenient acceptance criterion and a better chance of converging to the global minimum. However, a negative value of τ is appropriate if $L(\theta)$ has only a small number of local minima. This more restrictive criterion lowers the probability of accepting a θ_{new} that actually leads to an increase in the value of the loss function. In this analysis, τ was given a negative value of $\tau = -\sqrt{2\sigma^2}$.

Since the measurement errors have mean zero and are uncorrelated with finite variances the one-sided Chebyshev inequality can be used to get an upper bound on the probability of incorrectly accepting a new value of θ . With the choice above of $c = 1$, it can only be said that this probability will be less than or equal to 0. Note that if the measurement errors were in fact normally distributed then this probability would be bounded by $1 - P(Z < c)$, where Z is standard

normal, and would be 0.159 - a much more meaningful bound.

6.4. Results and Conclusions

The terminal estimates of θ that were obtained for each algorithm after each of the 40 replications are shown in Figure 6. BRS and SPSA seem to be converging to the same region of Θ . SAN, on the other hand, converged almost exclusively to either $[4, 4]$ or $[8, 8]$ with the percentage of θ_{term} values at each given by 65% and 20% respectively. Consider Figure 7, which shows the mean loss values over 1000 runs of the simulation for $\theta = [2i, 2j]$ where $i, j = 1, \dots, 4$. Note that the loss value at $[8, 8]$ is the minimum over these 16 values of θ . This plot also illustrates that the loss values in the region that BRS and SPSA are converging to are roughly the same. This slows their convergence

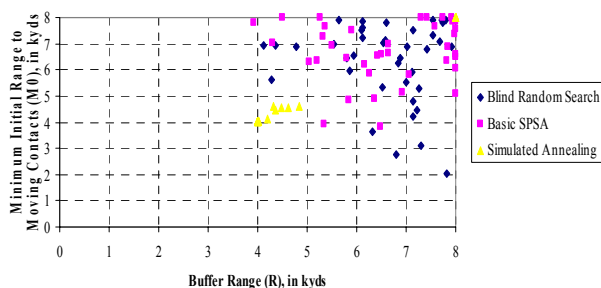


Figure 6. Terminal Estimates of θ

to $[8, 8]$ but SPSA does get there several times, whereas BRS never quite makes it. It is interesting that when SAN gets away from the initial condition $\theta_0 = [4, 4]$ it goes directly to $[8, 8]$.

Since each loss function measurement contains noise, simply taking one loss value at each θ_{term} and then averaging these over the 40 replications will not be an accurate measure of relative performance. Instead, we must first run the simulation a number of times at each θ_{term} . To decide how many runs N are required to average out the noise, recall that the standard deviation of the sample mean will decay at a rate of $1/\sqrt{N}$, provided that the noise is independent. In the area containing the majority of the terminal estimates of θ , the standard deviation of the measurement noise is about 1.0. $N = 100$ runs were performed at each θ_{term} , which yield an error in $\bar{L}(\theta_{\text{term}})$ of about 0.1, which is within an

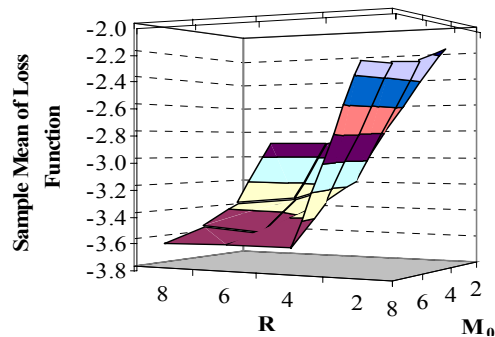


Figure 7. $\bar{L}(\theta)$ over 1000 runs of the simulation.

acceptable range. Note that SPSA has both the lowest value of $\bar{L}(\theta_{\text{term}})$ and the smallest standard deviation associated with the terminal loss. Table 3 presents the sample means of $\bar{L}(\theta_{\text{term}})$ for each algorithm and their corresponding 95% confidence over the 40 replications. However, two sample t-tests with non-identical variances must be used to determine if the performance of the algorithms was significantly different. The results of these tests show that the mean terminal losses for BRS and SPSA are both significantly lower than SAN at the 95% level, but they are not statistically different from each other.

In conclusion, it seems that both BRS and SPSA performed relatively well on this problem. It will be interesting to see how they compare when the dimension of θ is increased. The practical implications of the terminal estimates of θ are also quite interesting. While the terminal values for M_0 seem quite intuitive, those for R do not. The results seem to suggest that in order to maintain a safe distance between itself and all vessels OwnShip should project CPAs to all other vessels it is currently detecting. It was originally hypothesized that considering so many other vessels would result in an unreasonable number of OwnShip maneuvers and so a great deal of close encounters. But this does not seem to be the case. It should be noted, however, that when the vessel density is increased in later analyses these results may change dramatically. Thus, as is usually the case, the results of only one analysis are not the end of the road. Instead they simply spur more questions and the journey continues.

7. References

- [1] J.C. Spall, *Introduction to Stochastic Search and Optimization*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2003.