An Ant System Algorithm for the Neutralization Problem*

Ramazan Algin[1], Ali Fuat Alkaya[1], Vural Aksakalli[2], and Dindar Öz[1]

[1] Marmara University, Department of Computer Engineering, Istanbul, Turkey
[2] Istanbul Sehir University, Department of Industrial Engineering, Istanbul, Turkey
{algin.ramazan,ozdindar}@gmail.com, falkaya@marmara.edu.tr,
aksakalli@sehir.edu.tr

Abstract. We consider a path planning problem wherein an agent needs to safely and swiftly navigate from a given source location to a destination through an arrangement of disk-shaped obstacles. The agent possesses a limited neutralization capability in the sense that it can neutralize a certain number of obstacles enroute and pass through them safely upon neutralization. Optimal utilization of such a capability is called the neutralization problem. This problem is essentially a shortest path problem with resource constraints, which has been shown to be NP-Hard except for some trivial variants. In this study, we propose an ant system algorithm for the neutralization problem. In the proposed algorithm, the state transition rule makes use of certain problem-specific information to guide the ants. We show how the parameters of the algorithm can be fine-tuned for enhanced performance and we present limited computational experiments including a real-world naval minefield dataset. Our experiments suggest that the proposed algorithm finds high quality solutions in general with reasonable computational resources.

Keywords: ant system, path planning, metaheuristics, optimization.

1 Introduction

In this study, we tackle a path planning problem where the goal is to safely and swiftly navigate an agent from a given source location to a destination through an arrangement of disk-shaped obstacles in the plane. The agent is given a neutralization capability with which the agent can safely traverse through the disc after neutralizing it for a cost added to its traversal length. However, number of allowed neutralizations is limited, say by K, due to a particular reason such as the payload capacity of the agent or vehicle. The central issue here is how to direct the agent to optimally utilize this neutralization capability. This problem is called as the neutralization problem [1] which is NP-Hard.

Ant colony optimization (ACO) [2] is one of the most commonly used metaheuristics to solve path planning problems. The ACO is an umbrella concept that

* This work is supported by Marmara University Scientific Research Committee under the project ID FEN-D-100413-0135.

I. Rojas, G. Joya, and J. Cabestany (Eds.): IWANN 2013, Part II, LNCS 7903, pp. 53–61, 2013.

has some variations including ant system (AS) [3], ant colony system (ACS) [3], elitist ant system [4], max-min ant system (MMAS) [5] and some more [2]. The neutralization problem is also a path planning problem and the purpose of this study is to utilize the AS to solve this problem.

The rest of this paper is organized as follows. Section 2 presents an overview of the AS. Section 3 gives the definition of the neutralization problem. In Section 4, we describe our AS algorithm and how it is applied to solve the neutralization problem. Section 5 reports the results of experiments and their discussion. Section 6 concludes the paper with some future work.

2 Ant System

Ant system (AS) algorithm was first introduced by Marco Dorigo in 1992 and applied for solving the Travelling Salesman Problem (TSP) [4,6]. The AS is inspired by real ants life. In real life, ants can find the shortest path between their nest and food with the help of the pheromone [3]. Pheromone is a kind of chemical that is used by some kind of animals in different ways. Ants use it to communicate each other while finding the shortest path between the nest and the food. When an ant passes from a path, it lays some pheromone on this path which causes this path to become more desirable by other ants. At the same time, there is pheromone evaporation on all paths. Then paths which are not selected by ants become less desirable because of the pheromone evaporation. For a better understanding, consider the example given in Fig. 1.

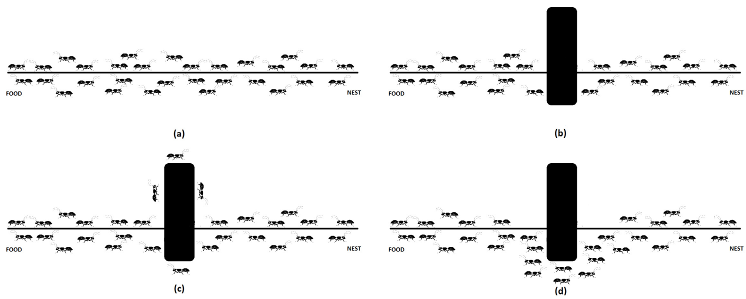

Fig. 1. (*a*) Ants are walking between food and nest without any obstacles on their path. (*b*) An obstacle occurs. (*c*) About half of the ants choose the upper path, the other half chooses the lower path. (*d*) Since ants walking on the shorter lower path reach the other side more quickly, more pheromone accumulates on the shorter path. Consequently, more and more ants start to choose this lower path over time.

The AS algorithm can easily be explained by considering the TSP as the tackled problem. Consider a graph (V,E) where V represents the set of cities and E is the set of edges between cities. In this graph, edges have travelling costs (δ). In addition to the travelling costs of edges, the AS algorithm needs to use the desirability value (τ) for each edge. The desirability in the algorithm is analogous to the pheromone in the real life and they will be used interchangeably through this manuscript. Different from the constant travelling costs of edges, the level of pheromone on each edge changes during execution of the algorithm based on the ants edge preferences. The goal of the TSP is finding the minimum closed tour length by visiting the each city only once [4].

In the implementation of the AS algorithm for TSP, each ant is initialized and then placed on the cities randomly (there must be at least one ant at each city). The edges are initialized to a predetermined initial pheromone level (τ_0). Then, the ants start their tours. Each ant tries to complete its tour by choosing the next city, if it is not visited before, according to the state transition rule. The state transition rule assigns probabilities based on the pheromone level and travelling cost to the edges. When all ants complete their tours global update rule is applied to all edges. By applying the global update rule, the level of pheromone on edge which is selected by ants increases and also at the same time pheromone evaporation occurs on all edges. The edges which are not selected by ants lose pheromone quickly and have less desirability, which means that the probability for being selected by other ants decrease.

State transition rule is applied according to the formula given by (1). This formula gives the probability of ant k that wants to go from city a to city b.

$$p_k(a,b) = \begin{cases} \frac{[\tau(a,b)].[1/\delta(a,b)]^\beta}{\sum_{u \in J_k(a)} [\tau(a,u)].[1/\delta(a,u)]^\beta} & \text{if } b \in J_k(a) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where τ is the pheromone, $\delta(a,b)$ is the cost of edge that is between city a and city b. β is a parameter which determines the relative importance of pheromone versus distance ($\beta > 0$). $J_k(a)$ is a set that keeps the unvisited cities. In (1), by multiplying the $\tau(a,b)$ and the heuristic value $1/\delta(a,b)$, the edges which have shorter length get higher pheromone.

When all ants complete their tours, global update rule is applied to all edges according to the formula given in (2).

$$\tau(a,b) = (1-\alpha).\tau(a,b) + \sum_{k=1}^{m} \Delta\tau_k(a,b) \tag{2}$$

where $$\Delta\tau_k = \begin{cases} \frac{1}{L_k} & \text{, if } (a,b) \in \text{tour done by ant } k \\ 0, & \text{otherwise} \end{cases}$$

α is pheromone decay parameter between 0 and 1, m is the number of ants and L_k is tour length performed by ant k.

In the next section, definition of the neutralization problem will be given.

3 The Neutralization Problem

Neutralization problem instance is a tuple (s, t, $\mathcal{A}$, c, K), where s is start point and t is terminal point in $\mathbb{R}^2$, $\mathcal{A}$ is a finite set of open discs in $\mathbb{R}^2$, c is a cost function $\mathbb{R}_{\geq 0}$, and K is a given constant in $\mathbb{R}_{\geq 0}$. In this problem we have an agent that wants to go from s to t. This agent cannot enter the discs unless he/she has an option to neutralize discs that can be considered as obstacle or threat like mine. The agent has neutralization capability that is limited with K number of discs where $K \leq |A|$. When discs are neutralized their neutralization cost is added to the cost of the path. In this problem our aim is taking the agent from s to t safely and using the shortest path.

An example for the neutralization problem is given in Fig. 2. In this figure, each disc has radius of 3 and neutralization cost of 0.8. As seen in figure, when our agent has $K = 0$ neutralization, (s)he chooses path 1. Similarly, when $K =$ 1, 2, 3, paths 2, 3 and 4 are our optimum paths, respectively. For $K = 1$, only A_6 neutralized, for $K = 2$, A_5 and A_6 are neutralized and for $K = 3$, A_5, A_6 and A_7 are neutralized.

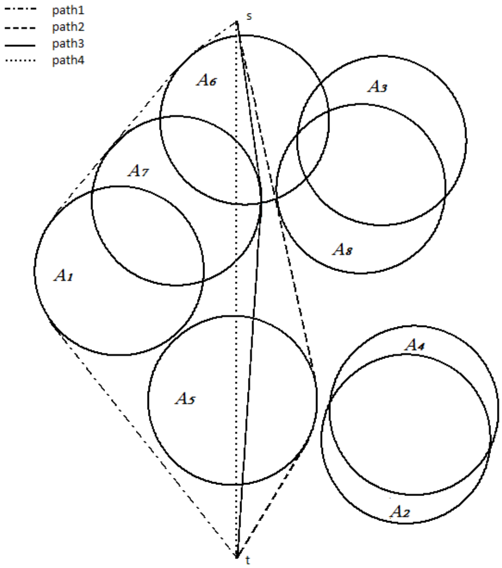

Fig. 2. An example to the neutralization problem and optimal paths for K=0, 1, 2 and 3

Similar to the TSP setting, an instance of the neutralization problem is represented by graph (V,E). In this graph there is also a set of discs and the edges intersecting these discs have additional travelling cost which is calculated proportionally by the number of discs intersected. Actually, these additional costs represent the neutralization cost of the discs. Another property of the edges is their weight values. Simply, weight of an edge represents the intersection status with a disc. So, the number of intersecting discs determines the weight value for an edge. Cost and weight of a path on the graph are sum of costs and weight of the edges on the path. Therefore the weight property of a path is used for checking its feasibility, i.e. satisfying the maximum number of neutralization (K) constraint.

To our knowledge, the neutralization problem defined above is studied in [1] where the authors develop a two phase algorithm for solving the neutralization problem. However, their proposed algorithm is based on the assumption that every disc has the same radius and same neutralization cost. In this paper, that constraint is released and we provide a solution method for more realistic scenarios. On the other hand, neutralization problem is closely related to the problems undertaken in real world applications in diverse fields such as telecommunications routing [7], curve approximations [8], scheduling and minimum-risk routing of military vehicles and aircraft [9]. Therefore, the techniques developed in this study may also be applied to other application domains.

In the next section, we provide the details of the ant system algorithm proposed for solving the neutralization problem.

4 Ant System Algorithm Proposed for Solving the Neutralization Problem

In the neutralization problem, we have an agent and this agent wants to go from the start point (s) to the target (t) point. From now on ants are our agents and our aim is still the same, we want to take these agents from s to t safely using the shortest path algorithm.

In our AS design, we made use of costs of the shortest paths from each node to destination. The cost of shortest path on each vertex guides the ants on their decisions. However the shortest path information is not enough to guide an ant, because a shortest path may be infeasible due to its weight. Therefore, the ant should also be aware of the weight information of the shortest paths. The pheromone level on each edge is definitely another guiding tool for the ants. With all these information at hand, the ant will choose the vertex to go next at each step. Our AS algorithm developed for the neutralization problem is given in Fig. 3.

We modified the original state transition rule (1) to apply to our problem. The probability of ant k that wants to go from vertex a to vertex b, in state transition rule is found according to the formula given in (3).

$$p_k(a,b) = \begin{cases} \frac{[\tau(a,b)].[1/(\omega(a,b,t).\delta(b,t))]^{\beta}}{\sum_{u \in J_k(a)}[\tau(a,u)].[1/(\omega(a,u,t).\delta(u,t))]^{\beta}} & \text{if } b \in J_k(a) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $\tau(a,b)$ is the pheromone level of edge which is between vertex a and vertex b, t is our terminal or destination point, $\delta(b,t)$ is total cost of path that is between vertex b and vertex t (terminal). β is a parameter which determines the relative importance of pheromone versus distance ($\beta > 0$). $J_k(a)$ is a set that keeps possible vertex to go. ω is the weight function which is found by

```
for each edge (a, b)
    τ(a, b) = τ_0
end for
/*Main loop*/
for n iterations
    for each ant k
        μ(k) = K
        place ant k on s
    end for
    repeat
        for each ant k
            /*suppose that ant k is at vertex v*/
            if (μ(k) ≤ η(λ(v)))
                return the shortest path found by Dijkstra's algorithm
            end if
            ant k chooses next vertex according to (3)
            if next vertex == null /*this means ant is jammed*/
                restart ant k by placing it at s and setting μ(k) = K
            end if
        end for
    until all ants reach t
    for each edge (a, b)
        apply global update rule given in (2)
    end for
end for
return best path found
```

Fig. 3. Ant system algorithm proposed for solving the neutralization problem

$$\omega(a, b, t) = \eta(\lambda(a)) - \mu(k) + \vartheta(a, b) + 1 \tag{4}$$

where $\lambda(a)$ is the shortest path from vertex a to t, $\eta(\lambda(a))$ is weight of shortest path from vertex a to t, $\mu(k)$ is the number of remaining neutralizations for ant k, $\vartheta(a, b)$ is weight of edge which is between vertex a and vertex b. In this formulation, we favor the edges which have greater amount of pheromone, and which have smaller cost and smaller weight on the shortest path to t. Global update rule that we use in our problem is exactly same as in (2).

In the next section there will be some computational experiments on real and synthetic data.

5 Experimental Work and Discussion

After designing the ant system algorithm for solving the neutralization problem, computational tests are carried out to assess its performance. The computational tests are conducted both on real and random discrete minefields. The real minefield has 39 discs on a [0, 92]x[0, 80] rectangle [10]. On the other hand there are seven random minefields each having 100 discs on [0, 100]x[0, 100] rectangles. The real minefield and one of the random minefields are given in Fig. 4.

The performance of AS is compared with the exact solutions that are obtained by an exact algorithm having a high order of run-time complexity. The exact method used in this study is a novel approach which is currently being developed and based on applying $k-th$ shortest path algorithm starting from a lower bound solution. On the average, our proposed AS obtains the solution in 290 seconds for ten ants and ten iterations whereas the exact algorithm requires 69 hours.

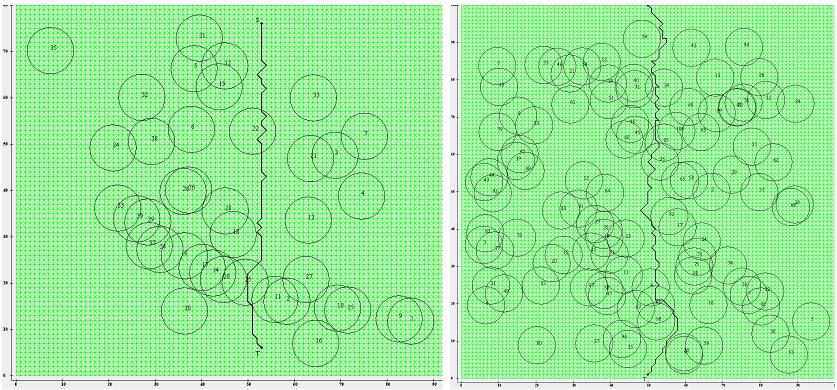

Fig. 4. Real minefield (left) and a random minefield (right)

For the AS proposed in this study, we followed the following work plan to find out the best performing parameter values from the set given in Table 1. Before that, we should note that when the modified AS is run with a parameter set, the best path of the ants is recorded as the output of that run.

To find out the best parameter values for a parameter of modified AS, say β, the parameter is fixed to a value, say 2, and all results taken with various combinations of m, α and n are averaged to find out average results for that value of the parameter. In this way, the average performance of the parameter value on all circumstances is presented. This analysis is repeated on all minefields. AS has four parameters and their values make a total of 270 different combinations. The performance analysis for the parameters is given in Table 1 where both cost of best path and its percentage deviation from the exact solution are displayed.

We may reach the following conclusions after analyzing Table 1. We firstly observe that increasing the value of the β parameter gives better paths. Based on the state transition rule, we can say that the increase for β means the ants are guided to the shortest but feasible paths. It is interesting to note that the value of the α parameter does not have any significant effect on the performance. On the other hand, increasing the number of ants (m) and number of iterations

Table 1. Performance analysis of parameter values

	parameter values	real minefield tc	real minefield dev	random minefield(avg) tc	random minefield(avg) dev
m	5	78,38	5,2%	117,85	8%
	10	76,97	3,3%	117,45	7%
	20	**76,97**	**3,3%**	**116,50**	**6%**
β	1	113,88	52,9%	150,18	37%
	2	101,40	36,1%	136,50	25%
	5	95,36	28,0%	132,20	21%
	10	87,60	17,6%	128,27	17%
	20	81,11	8,9%	120,16	10%
	100	**76,97**	**3,3%**	**116,42**	**6%**
α	0,01	77,80	4,4%	117,36	7%
	0,02	76,97	3,3%	117,21	7%
	0,05	76,97	3,3%	117,57	7%
	0,10	**76,97**	**3,3%**	**116,77**	**7%**
	0,20	76,97	3,3%	117,34	7%
n	10	76,97	3,3%	117,58	7%
	50	76,97	3,3%	117,02	7%
	100	**76,97**	**3,3%**	**116,89**	**7%**

Table 2. Detailed performance analysis of the β parameter

β	avg	avg dev
50	124,87	14,1%
60	122,48	11,9%
70	121,17	10,7%
80	120,73	10,3%
90	120,49	10,1%
100	120,26	9,9%
110	**118,89**	**8,6%**
120	119,95	9,6%
130	126,71	15,8%
140	125,58	14,7%
150	124,61	13,9%

(n) bring better paths which is expected due to pheromone update. When more ants pass from a path with more iterations, pheromone level on this path is increased. We should note that the best parameter set shows a performance about %6 away from the optimum. Most of this gap is due to the zigzag patterns of the paths that is a result of the stochastic decisions of ants (see Fig. 4).

One wonders about the limit of β where the performance increase stops. To reveal this, we ran the modified AS algorithm with a mesh of β values where α, m and n were set to 0.1, 10 and 100, respectively. The results are given in Table 2. It can be easily observed that the performance with respect to various β values, shows a U shape result. That is, the performance gets better for increasing values of β from 1 up to 110 and the best performance is obtained when $\beta = 110$. The reason why the performance deteriorates after $\beta = 110$ is due to the probability values converging to zero.

6 Summary, Conclusions and Future Work

In this study, an ant system algorithm is developed and tested for solving the neutralization problem. Proposed ant system algorithm differs from the original AS in the state transition rule. In our state transition rule, we favor the edges which have greater amount of pheromone, and which have smaller cost and smaller weight on the shortest path to the destination. The performance of the proposed AS algorithm is compared with exact solution both on real and synthetic data and the results are promising. As a future work, the algorithm can be improved by applying some postprocessing or by trying some other state transition rule rules. In addition, other ACO algorithms may be exploited to solve neutralization problem.

References

1. Alkaya, A.F., Aksakalli, V., Periebe, C.E.: Heuristics for Optimal Neutralization in a Mapped Hazard Field (submitted for publication)
2. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press Cambridge (2004)
3. Dorigo, M., Gambardella, L.M.: Ant Colony system: A Cooperative learning approach to the Travelling Salesman Problem. IEEE Transaction on Evolutionary Computation 1(1) (1997)
4. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst., Man, Cybern. B 26(2), 29–41 (1996)
5. Stutzle, T., Hoos, H.: Improvements on the ant system, introducing the MAX-MIN ant system. In: Proc. ICANNGA97–Third Int. Conf. Artificial Neural Networks and Genetic Algorithms. Springer, Wien (1997)
6. Lawler, E.L., Lenstra, J.K., Rinnooy-Kan, A.H.G., Shmoys, D.B.: The Travelling Salesman Problem. Wiley, New York (1985)
7. Kuipers, F., Korkmaz, T., Krunz, M., Van Mieghem, P.: Performance evaluation of constraintbased path selection algorithms. IEEE Network 18, 16–23 (2004)
8. Dahl, G., Realfsen, B.: Curve approximation and constrained shortest path problems. In: International Symposium on Mathematical Programming, ISMP 1997 (1997)
9. Zabarankin, M., Uryasev, S., Pardalos, P.: Optimal Risk Path Algorithms. In: Murphey, R., Pardalos, P. (eds.) Cooperative Control and Optimization, vol. 66, pp. 271–303. Kluwer Academic, Dordrecht (2002)
10. Fishkind, D.E., Priebe, C.E., Giles, K., Smith, L.N., Aksakalli, V.: Disambiguation protocols based on risk simulation. IEEE Trans Syst. Man Cybernet Part A 37, 814–823 (2007)