Evolutionary Dynamic Optimization Techniques for Marine Contamination Problem

Lokman Altin
Computer Engineering Dept.
Marmara University
Goztepe, Istanbul, Turkey
lokman.altin@marmara.edu.tr

Haluk Rahmi Topcuoglu
Computer Engineering Dept.
Marmara University
Goztepe, Istanbul, Turkey
haluk@marmara.edu.tr

Murat Ermis
Industrial Engineering Dept.
Turkish Air Force Academy
Yesilyurt, Istanbul, Turkey
m.ermis@hho.edu.tr

ABSTRACT

Marine pollution is the release of by-products that cause harm to natural marine ecosystems and one of the most important sources is the discharge of oil, ballast water from vessels. If the relevant technology is not available, alternative way to monitor environmental pollution is to use unmanned air vehicles (UAVs). Since the navigating vessels move in different directions and speeds, the determination of the tour that should be traveled by a UAV resembles to the dynamic traveling salesman problem (DTSP) in many aspects. This paper addresses a new type of DTSP, where targets can move in different directions with different speeds. The locations of all vessels can change due to changes in velocity that alters the length of all edges. Consequently, this problem has a higher complexity in comparison to classical DTSP presented in the literature. An empirical study is conducted to evaluate performance of selected evolutionary dynamic optimization techniques on solving the problem.

Categories and Subject Descriptors

Computing methodologies [**Artificial Intelligence**]: Search methodologies—*Discrete space search*; Applied computing [**Computers in other domains**]: Military

General Terms

Algorithms, Experimentation

Keywords

Evolutionary Dynamic Optimization; Dynamic Traveling Salesman Problem

1. INTRODUCTION

One of the most significant causes of marine contamination problem is accepted as pollutants, including dumping, oil and exhaust pollution that are discharged from large ships. An alternative way to detect the discharge of such substances is to use unmanned air vehicles (UAVs). In this problem, a couple of traveling vessels at the sea on different positions should be surveilled along the way according to predefined objective function by a UAV. The UAV wants to take a snapshot of these vessels starting from ground station and returning after it visited all the vessels to its original base. The objective is to minimize total distance traveled or total elapsed time.

Assume that at t_0 a UAV takes off and it is on its route, and there are initially eight vessels $(V_1, \cdots, V_8)$ positioned at different locations. By considering the distance between vessels, the route can be generated, and the UAV starts the tour. Most of the vessels move in different directions with different speeds, hence the planned tour should be affected seriously. In this way, both the positions and their links (i.e. distances) should be changed during the execution. After the occurrence of a change, an alternative tour must be generated fast enough in order to complete the mission. In this real world application, travel directions and speeds can change dynamically and hence, are reasonably modeled as stochastic variables.

Most of the real world problems in different domains have various forms of dynamism, including a change(s) in the objective function, the problem constraints, the decision variables or environmental parameters with time, where the main motivation of a given dynamic optimization problem becomes tracking the global optimum value as close as possible [7]. But, there is a few studies focuses on these real world problems and applying evolutionary dynamic optimization (EDO) techniques to solve them.

The main motivation of this paper is to better understand the nature of moving vessels in determination of tours dynamically for the domain of marine contamination, where course directions and speeds of vessels can change dynamically. In order to model this problem, we present a new type of dynamic traveling salesman problem (DTSP). We also conduct an empirical study to evaluate performance of selected EDO techniques on solving the problem.

2. A VARIANT OF THE DYNAMIC TRAVELING SALESMAN PROBLEM

The traveling salesman problem (TSP) is a fundamental and intensively studied NP-hard combinatorial optimization problem. Given a set of cities and the distance between each pair of the cities, it aims to find the shortest path that visits each city exactly once and returns to the original city. There

GECCO'15 Companion, July 11–15, 2015, Madrid, Spain.

http://dx.doi.org/10.1145/2739482.2768433.

are various formulations of the traveling salesman problem; and one of them is the binary integer programming formulation [6].

The dynamic traveling salesman problem (DTSP) in the literature can be constructed in three ways/modes [8]. In the first mode, the *edge change mode*, DTSP is generated by introducing traffic factor [5] on one or more edges, which can be simulated by the increase or decrease in the distance between cities. An increase in the traveling distance on an edge due to traffic factor does not change the locations of the cities that are connected by the edge; and all other remaining edges are not affected. On the other hand, the *insert/delete mode*, adds or deletes one or more cities, which is more difficult for EDO techniques due to variable length chromosome representation. In order to simplify representation, the number of cities are kept constant by keeping half of all cities in the initial solution and using the remaining half as the spare set for changes [4]. The last mode, the *vertex swap mode*, swaps the locations of two cities, which only changes the solution but keeps the length of the tour constant.

Although the problem presented in this paper looks similar to the DTSP, there are some important differences. In our problem, the locations of all vessels can change due to changes in velocity that alters the length of all edges, which is not possible in the *edge change mode*. Furthermore, after a long testing interval with a high frequency of change (say in every 10-20 generations), the *insert/delete mode* may start to repeat previous instances if a fixed length chromosome is considered [4]. Additionally, our search space (the number of different locations of vessels) gradually increases according to frequency of change, and they are not predefined as in the DTSP instances in the literature. Finally, the *vertex swap mode* does not cause a change in the length of optimal tour, which is irrelevant with our version of the DTSP.

3. EVOLUTIONARY DYNAMIC OPTIMIZATION TECHNIQUES

In our experimental study, we consider four algorithms in order to measure their performance on the given DTSP, where a brief explanation of each algorithm is given below.

Steady-State Genetic Algorithm (SSGA).

It is basically a simple genetic algorithm where it checks whether there is a change in the environment at each iteration. In case of a change, whole population is randomly re-initialized; otherwise, normal GA cycle is repeated. It should be noted that changes in the environment are known a priori for most of the dynamic optimization problems where EDO techniques do not perform an extra work.

Random Immigrants (RI) Method.

Random Immigrant algorithm [3] is a genetic algorithm that targets to maintain diversity by adding a number of generated random individuals to the population without detecting the changes explicitly. A control parameter p_r, which is the *replacement rate*, is used to generate $r \times |P|$ number of individuals where $|P|$ is the population size.

Hyper-Mutation (HM) Method.

The hyper-mutation approach [7] targets to increase the diversity of a genetic algorithm by increasing the current mutation rate. In case of a change, mutation rate, p_m, is increased to a high mutation rate, p_{high}. After a new population is reproduced with the high mutation rate, it is set to low mutation rate, p_m.

Memory/Search (MS) Algorithm.

The *memory/search algorithm* [2] divides the population into two sub-populations each having n individuals, a *memory population* and a *search population*. Additionally, a third population, a separate explicit *memory* of k individuals, is also considered in this method. While the search population explores new areas of the search space and it submits new peaks to the memory, the memory population exploits the memory and it maintains minimum jump [2].

Memory update frequency term determines when the best individual selected from the memory population and the search population is stored into the memory. When the memory becomes full, the *mindist* (minimum distance) replacement strategy [2] is considered to replace the best individual with another one from the population. Search population is evolved as the memory population. When there is a change, the memory population and the explicit memory are merged, and the best n individuals constitute the memory population. The individuals in both populations are reevaluated when the environment is changed. The search population is reinitialized in case of a change.

4. RESULTS AND DISCUSSION

In order to investigate the performance of four algorithms for the given DTSP, a set of experiments are conducted in this study. For each algorithm on a DTSP instance, a total of 30 independent runs were carried out by using the same random environmental changes with the stopping criteria of 5000 generations. Number of vessels in our experiments are set to 100 for all tests, and the coordinates of vessels are randomly assigned from the range [20..80], unless otherwise stated. The initial heading angles (θ values) are randomly assigned from the range [0°..360°], and the initial velocity of the vessels are randomly assigned from the range [2..12].

The population size for each algorithm (other than the MS algorithm) is set to 50. On the other hand, the size of both memory and search population of Memory Search (MS) algorithm is set to 25; and explicit memory size is set to 10. The mutation rate is set to 0.02; and high mutation rate for hyper-mutation is equal to 0.5. The population replacement rate for the RI algorithm is set to 0.3.

We consider *offline performance* as the main metric for the comparisons, which is defined as the average of the best solution found within the same period over a given number of periods until termination, where a period is the time interval between two landscape changes. Formally, the offline performance is defined by $x^* = \frac{1}{T}\sum_{t=1}^{T} e_t^*$ where $e_t^* = \max\{e_\tau, e_{\tau+1}, \ldots, e_t\}$, e_t is the t^{th} evaluation, τ is the time of the last change. It should be noted that the offline performance values given in tables and figures are the average of 30 independent runs, unless otherwise stated.

Table 1 presents offline performance values of the SSGA, the HM, the RI and the MS algorithms. The severity of changes in this paper is represented with three parameters: (α, θ, V_{moving}). The first parameter, α, is the the percentage of changing (i.e. moving) vessels in the environment, where different α values are given at the first row of the table. If α

is equal to 100%, all vessels move by changing their velocities and the heading angles so that their positions are updated. The θ parameter is the change severity in the heading angle, where a vessel may change its heading angle randomly up to θ angle in both radial directions. The V_{change} parameter is the severity of change in velocity of vessels, where the velocity of a moving vessel is updated using the uniform distribution $U(-V_{change}, V_{change})$. When a vessel is selected for the change, its both velocity and moving direction (i.e. the heading angle) will be updated.

Based on the results observed in Table 1, the HM algorithm outperforms all other algorithms for environments with different characteristics. Increasing V_{change} and α values causes performance degradation of algorithms. An increase in θ value leads to better performance for the cases of low V_{change} and α values. The SSGA algorithm is the worst performing algorithm for all cases due to suffering from random initialization and beginning from scratch to the convergence process for all change periods.

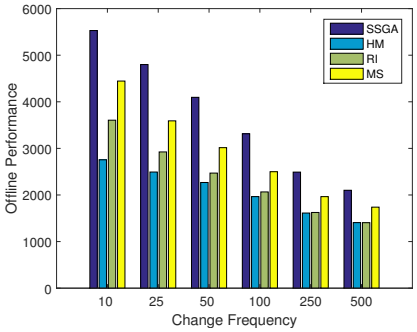

Figure 1: Offline performance by varying the change frequency

The second experiment shows the effect of frequency of change (f) on solution quality of each algorithm where V_{change} is set to 5.0, θ is set to 40 and α is set to 0.2. Figure 1 presents the average offline performance values of the four algorithms for six different change frequencies. As the the number of generations between successive changes increases, algorithms have longer time to come up with a solution which decrease the offline performance values for all algorithms. When $f = 10$, the HM algorithm significantly outperforms the other algorithms; and it is better than SSGA algorithm by 50%. Both the HM and the RI algorithms give the best results and they outperform the SSGA by 33% when $f = 500$.

In another experiment, initial coordinates of the vessels are updated based on different coordinate ranges, where a coordinate range of [0..100] indicates that vessels have a higher dispersion area for setting their initial coordinates when compared with the case of range [40..60] in a 2-D space (see Figure 2). All algorithms have better results for environments when vessels are initialized in a closer range; and the HM algorithm outperforms all other algorithms for different initial conditions.

We consider three different severity of change levels, *low severity*, *medium severity* and *high severity*, in order to measure performance of algorithms on environments with different characteristics of our variant of the DTSP. For each level, the three parameters for specifying the severity of changes,

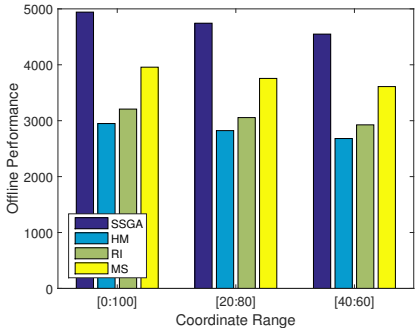

Figure 2: Offline performance by varying the initial range of the coordinates for all vessels

which are α, θ, V_{moving}, are set based on the values given in Table 2.

Table 2: Values of parameters for different severity levels

Setting	Low Severity	Medium Severity	High Severity
Velocity(V_{length})	2.0	5.0	10.0
Angle(θ)	20	40	60
Changing Vessel Ratio	0.05	0.5	1.0

In this set of experiments, we study both average performance over the given population and the best performance of all algorithms for different levels of severity changes based on the parameter values given in Table 2, where frequency of change is set to 100. If there is a small change in the environment, both the HM and RI algorithms outperforms other algorithms, and the populations of the HM algorithm converge better than the populations of the RI algorithm. A similar behavior is observed for the medium severity case and the HM algorithm outperforms the other algorithms, where the changes in each period is not as smooth as the low severity case.

On the other hand, performance degradation is observed for the high severity case, which is the result of distribution of vessels in search space beside the high level of change severity in the environment. Positions of vessels are initialized to the coordinate ranges of [20..80]. Vessels can move all directions significantly for the high severity case by leaving the initial search space. After a few change cycles are performed, positions of vessels become more sparse, which cause performance drops of the algorithms. As in the other cases, the HM algorithms gives the best offline performance values for the high severity case, which is followed by the RI algorithm. The previous set of experiments are repeated for a low frequency of change value, i.e., when a change occurs in every 250 generations. The HM and RI algorithms outperforms the other algorithms for all severity levels. Although HM algorithm is better than the RI algorithm, the differences in best performance curves are not as significant as the case of higher frequency of change values.

As our last experiment, we consider the *area between curves (ABC)* metric [1], which is a new metric that quantifies the distance between the performance curves of each pair of algorithms. It is the difference of area below performance curves, which is calculated with the trapezoidal method for consecu-

Table 1: Average offline performance values of the algorithms for different severity of changes by varying the percentage of modified vessels, velocities and heading angles

	$\alpha = \%5$			$\alpha = \%10$			$\alpha = \%20$			$\alpha = \%50$			$\alpha = \%100$		
	θ=20	θ=40	θ=60	θ=20	θ=40	θ=60	θ=20	θ=40	θ=60	θ=20	θ=40	θ=60	θ=20	θ=40	θ=60
V_{change}:2															
SSGA	2913.16	2892.77	2846.88	3347.39	3256.18	3154.79	3842.28	3671.03	3491.74	4586.11	4206.15	3840.25	4929.90	4445.44	4055.48
HM	**1559.62**	**1533.11**	**1495.84**	**1788.39**	**1765.93**	**1715.58**	**2004.22**	**2029.97**	**1918.96**	**2345.07**	**2373.50**	**2246.88**	**2667.37**	**2722.25**	**2546.89**
RI	1703.30	1655.75	1625.86	1932.99	1898.64	1844.38	2182.50	2155.30	2085.31	2547.70	2598.28	2439.14	2923.10	2950.43	2763.20
MS	2054.55	2000.40	1977.01	2356.96	2326.74	2225.55	2710.64	2646.88	2521.20	3190.10	3118.26	2921.54	3626.13	3533.98	3225.66
V_{change}:5															
SSGA	2969.14	2989.79	2896.57	3533.42	3476.16	3307.19	4210.55	4096.04	3860.94	5002.16	4743.67	4389.52	5300.15	4975.72	4614.46
HM	**1617.41**	**1610.56**	**1554.03**	**1938.54**	**1916.56**	**1902.90**	**2195.82**	**2268.22**	**2272.00**	**2552.58**	**2822.80**	**2811.90**	**2949.12**	**3320.12**	**3282.45**
RI	1742.42	1722.88	1743.08	2127.76	2062.07	2054.44	2447.68	2470.01	2446.52	2837.14	3055.71	3065.60	3222.67	3594.75	3545.51
MS	2078.31	2090.28	2036.06	2543.36	2535.62	2482.71	3009.82	3015.54	2962.21	3564.97	3756.02	3663.56	4141.48	4350.90	4197.06
V_{change}:10															
SSGA	3124.61	3080.07	3021.69	3781.94	3712.46	3576.79	4503.87	4384.06	4196.79	5115.20	4942.46	4671.19	5315.66	5122.77	4836.29
HM	**1731.21**	**1704.93**	**1720.88**	**2098.15**	**2100.01**	**2108.41**	**2368.42**	**2498.81**	**2523.79**	**2698.34**	**3114.55**	**3217.05**	**3136.53**	**3664.33**	**3781.74**
RI	1866.72	1877.64	1841.16	2277.76	2281.68	2275.07	2574.15	2742.02	2751.31	2930.51	3377.11	3455.13	3474.38	4004.65	4069.27
MS	2258.76	2234.31	2234.23	2770.48	2750.02	2714.62	3245.46	3330.53	3336.43	3766.39	4186.31	4144.30	4355.88	4799.04	4764.61

tive generations. It can be mathematically represented with the following integral in the interval $[1..G]$,

$$ABC_p^{A_1,A_2} = \frac{1}{G}.\int_1^G p_{A_1}(x) - p_{A_2}(x)dx \qquad (1)$$

where G is the total number of generations and the $p_{A_1}(x)$ and $p_{A_2}(x)$ are the functions (for algorithms A_1 and A_2) that are to be replaced with a measure of population quality including average best of generation, offline performance value, offline error value [1]. ABC value can be positive and negative, where a negative value indicates that algorithm A_1 performs better than A_2 for a minimization problem.

Table 3: ABC results for low severity

Algorithms	SSGA	HM	RI	MS
SSGA	-	1353.54	1209.86	858.61
HM	-1353.54	-	-143.68	-494.92
RI	-1209.86	143.68	-	-351.25
MS	-858.61	494.92	351.25	-

Table 4: ABC results for medium severity

Algorithms	SSGA	HM	RI	MS
SSGA	-	1920.88	1687.96	987.66
HM	-1920.88	-	-232.91	-933.22
RI	-1687.96	232.91	-	-700.31
MS	-987.66	933.22	700.31	-

Tables 3, 4 illustrate the ABC results for each pair algorithms in environments with low severity and medium severity, respectively, when the change frequency is set to 50. The rows of the HM algorithm always receive negative ABC values with all other algorithms, since the curves of the HM algorithm always lie under the other three alternatives. An increase in the gap between performance of algorithm pairs is observed when it is moved from low severity (Table 3) to medium severity (Table 4).

Performance evaluation of all algorithms show that the HM algorithm is the best approach to converge each intervals' best solution while keeping its population more diversified compared to other dynamic optimization techniques. Inadequate performance of the RI approach shows that random search, aiming to keep diversified population to tackle changes, is less suitable than the HM algorithm where there are local changes. On the other hand, the MS approach performs worse than other dynamic optimization techniques due to lack of any cyclicity/periodicity in the environment. Use of explicit memory is futile for the problems where the previous best solutions do not visit same or nearby locations in the search space.

5. CONCLUSIONS

In this paper, we model the marine contamination problem with a new version of the dynamic traveling salesman problem (DTSP), assuming that targets change their location with different directions and speeds. It has higher complexity in comparison to classical DTSP presented in the literature. The empirical study validates the applicability of selected evolutionary dynamic optimization techniques for solving a different version of the DTSP.

6. REFERENCES

[1] E. Alba and B. Sarasola. Abc, a new performance tool for algorithms solving dynamic optimization problems. In *Congress on Evolutionary Computation (CEC), IEEE*, pages 1–7. IEEE, 2010.

[2] J. Branke. Memory-enhanced evolutionary algorithms for changing optimization problems. In *Congress on Evolutionary Computation (CEC'99)*, pages 1875–1882. IEEE, 1999.

[3] John J. Grefenstette. Genetic algorithms for changing environments. In *Parallel Problem Solving from Nature 2*, pages 139–146. Elsevier, 1992.

[4] Michalis Mavrovouniotis and Shengxiang Yang. A memetic ant colony optimization algorithm for the dynamic travelling salesman problem. *Soft Comput.*, 15(7):1405–1425, 2011.

[5] Michalis Mavrovouniotis and Shengxiang Yang. Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors. *Appl. Soft Comput.*, 13(10):4023–4037, 2013.

[6] R. Rardin. *Optimization In Operation Research.* Prentice-Hall, 1998.

[7] Shengxiang Yang and Xin Yao. *Evolutionary Computation for Dynamic Optimization Problems.* Springer-Verlag, 2013.

[8] Abdunnaser Younes. *Adapting Evolutionary Approaches for Optimization in Dynamic Environments.* PhD thesis, University of Waterloo, 2006.