International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 10 (2018) pp. 7842-7847

'NeuroG': An Adaptive Neuro-Engine for 'Dynamic Space-Junk Removal' Mission

[1]Vijay A. Kanade

[1]Research Associate, Evalueserve SEZ (Gurgaon) Private Limited, Sector–48, Gurugram – 122001, Haryana, India.

Abstract

The space around the planet Earth is littered with vast amount of 'Space Debris'. Space debris, junk, rubbish, remains, waste, or trash is a cluster of obsolete man-designed articles in space – old dilapidated satellites, wastes from multi-stage rockets, and flakes derived from erosion, collisions caused by debris itself or malfunctioning space units. This orbiting debris is growing at an alarming rate and posing a serious threat to the operational orbits above the planet. There is enough scientific evidence to suggest that the debris density at the Low Earth Orbit (LEO) has already reached a sufficient level to trigger a chain reaction of debris-collisions. To avoid any damage to the essential orbiting satellites and to keep the planet's space sustainable, we ought to soon embark on a mission of cleaning-up the space debris. Space organizations and scientific communities around the world are well-informed of the threat space debris possesses and have developed solutions on how to tackle the problem for a better future. Prominent solution for the above problem includes launching a spacecraft for actively removing the dangerous plethora of space debris pieces. Such a missioner-spacecraft would isolate / de-orbit or clean several targets (i.e. debris) during one single mission. This raises a significantly relevant question of "what kind of targets and in what order should the spacecraft visit them?" This paper answers the above question. The paper presents a novel method of employing 'NeuroG' (i.e. Neuro-engine) on a satellite platform for removing the space junk in a cost-effective & coherent manner.

Keywords: Low Earth Orbit (LEO), Geostationary Earth Orbit (GEO), Evolutionary Algorithm (EA), Debris Removal Entity (DRE), Neuro-Engine (NeuroG), Travelling Salesman Problem (TSP), Genetic Algorithm (GA), Debris object (D.O)

INTRODUCTION

First artificial satellite successfully launched into space was Sputnik-1, in the year 1957. After the first mission, a flurry of such man-made objects has been launched that has created enormous amount of space junk around the planet earth. In fact, exactly after four years of the launch of Sputnik, an explosion was recorded in Space – the explosion generated a huge amount of orbiting debris, out of which 300 pieces of debris were identified and tracked. The history repeats itself - in over a decade 200 such similar catastrophic accidents were recorded, creating more than 1000 debris fragments. The major concern relating to such events is the chain reaction that a single event leads to. This is having a serious impact on the utilization of operational orbits. As per the scientists at NASA, satellite launches are only adding to the already denser debris population at the LEO region. So for us to exploit the near-Earth atmospheric space, cleaning up the selective pieces seems the only viable option. Significant work and development is done to address the mentioned space junk problem. Prominently, an 'Active Space Debris Removal' method is designed for de-orbiting the selected debris object with the aid of tethers, robot manipulators, thrusters, etc. which are attached to a spacecraft. The spacecraft is launched in space for traversing the entire geographic area by moving to each piece of debris one at a time, and thereby cleaning the space junk. These traditional approaches highlight the importance of choosing the sequence of debris to be removed during such missions. A lot of trust is being shown on the Travelling Salesperson Problem (TSP) for tackling the problem [1], [5]. But one of the main differences is that the debris, unlike cities, is dynamic and is moving at high orbiting speeds. Thus the value of a visiting sequence depends on the visiting schedule used while traversing the dynamic debris. The visiting sequence is radically different from the one used in a standard TSP framework wherein only static cities are under consideration. The proposed paper takes into account the dynamism shown by the space junk, while keeping the cost function of the mission in check.

Figure: Space-debris around the Planet Earth

METHODOLOGY

The methodology employs usage of three notable modules, positioned on a 'Satellite Platform'. The three modules are as discussed below:

A. *Object recognition*

Object recognition is a process for recognition of a specific 'object' within a digital image or a real-time video. The algorithms developed rely on matching, learning, and/or pattern recognition based on feature extraction or appearance identification. Standard methods include edges, gradients, Histogram of Oriented Gradients (HOG), Haar wavelets, and linear binary patterns implemented within MATLAB environment.

Further, to enhance the effectiveness of 'object recognition' in space, a machine-learning technique needs to be exploited known as "deep learning" — which is a re-establishment of the traditional artificial-intelligence technique of neural networks — which are used to train scene-classifier. This implies that, scene-recognition and object-recognition work in close-relative-association [3],[7].

Array of sensors are utilized for applying the above mentioned object recognition algorithm for identifying / tracking the object (i.e. identifying if the object is actually a form of debris or a useful satellite, etc) within the observed scene & plotting the location of the debris objects (D.Os) on the map of the targeted region. The map developed is further used by the NeuroG for junk clean-up task.

B. NeuroG

A neuro-engine runs an evolutionary algorithm for cleaning-up the identified debris from the space ecosystem in a specific sequence that yields the most optimal solution set.

The evolutionary algorithm employed in the neuro-engine is a modified 'Genetic Algorithm'. The algorithm draws a metaphor from human biology and genetics to iteratively evolve a population of initial individuals to a population of high quality individuals, where each individual represents a solution of the problem to be solved and is composed of a fixed number of genes. The population of individuals refers to the set of identified orbital debris that needs to be traversed for clean-up purpose.

Figure: Satellite Platform for Space-Junk Removal

Implemented Genetic Algorithm:

In order to implement GA, we need to define the fitness function to be used. As 'distance' is the main criteria on which we are going to decide the fitness of a solution, the fitness function is given by:

$$f(x)=\min(distance\_1, distance\_2. . . . distance\_n)$$

We have used GA twice– in a successive manner. First GA is used to find out the single shortest path. Second GA is used to find the optimal division of that very path amongst the available debris removal entities (i.e. DREs) for cleaning up the junk. This method helps in deciding the optimal DRE set required for the task. The implementation details of these two GA's are as given below:

In the first GA, we generate the initial population set for visiting and cleaning the plotted debris objects (i.e. sequence for visiting each debris object). For this purpose we use array whose length is equal to the no. of target D.Os. Further, we utilize Roulette-wheel selection method for selecting any two solution sets from the initial population set for further operation.

Then we perform crossover operation - In this phase a crossover point is randomly selected for 'one solution set' and the sequence of D.O is kept unchanged up to the crossover point for the same solution set. Now, the D.Os following the crossover point are interchanged with the D.O sequence occurring in the 'second solution set' without repeating the D.Os already placed up to the crossover point. Then the distance metric is computed for the newly generated solution set (i.e. child set). If the computed distance is less than their parent sets (i.e. two selected solution sets) then we kill the parent sets and keep the child set. Similarly, if the distance metric of the newer off-spring is more than parent then we kill the child set and go for next crossover with newer crossover point.

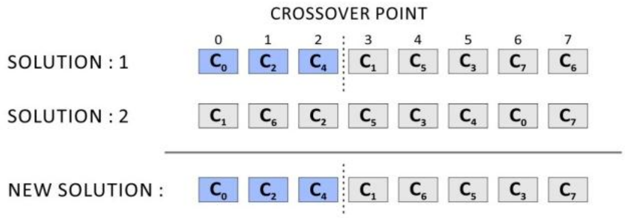

Figure: Crossover

After single crossover, we perform mutation operation on the newly generated child sets. We select any two D.Os from the newly created solution set and interchange or swap them in order to change the sequence of path. Then the distance metric is again calculated for the mutated solution. If the distance metric output is lesser, we retain the changed solution set and kill the old solution set; else we kill the new solution set and retain the solution set existing before the mutation operation. This process is repeated – [count (D.O*D.O)] no. of times until the best optimal solution is generated [4].

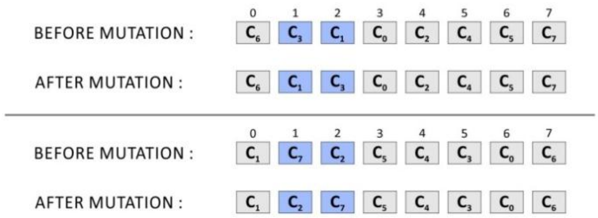

Figure: Mutation

Second GA is quite complicated than the first GA. Up to second GA we have the shortest path for a single DRE to travel and clear selected debris. In the second GA we divide this path into 'm' no. of paths so that the total addition of distance metric of all paths accounts to a minimum value. Though we have the shortest path, it is complicated to divide that into 'm' no. of paths, because the shortest path before division may yield costlier distance metric after division. So while performing crossover and mutation we have to make the necessary changes to the shortest path. After division of path sequence, the 2D array might look like:

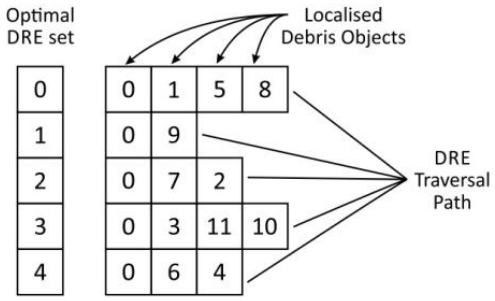

Figure: Solution after division of DRE Traversal Path

Once the intermediate solution for visiting the debris in a specific order by multiple DREs is identified, the very next task addressed is of handling the dynamically active space debris. This is taken care of by the following module – 'Vision-based navigation'

C. Vision-based Navigation

This navigation method is incorporated in the devices / tools used for capturing the identified debris. The technique aids in dealing with the dynamic debris, according to whether the debris orbits are considered as fixed in time or subject to orbital perturbations.

In the proposed vision based navigation approach, images of the space environment are sampled, stored and organized as a set of ordered or sequenced images (visual path) which provide a visual memory of the environment. Further, the DRE navigation task is defined as a concatenation of visual path subsets (called visual route) linking the current observed image and a target image stored in the visual memory. Thus, the DREs are controlled by using a vision-based control law adapted to its dynamic environment. The DREs are subject to initial trigger movement once the trajectory is plotted on the map of the target area [2]. After the trigger movement, the DREs navigate the path on their own. This navigation continues until the path plotted on the geographic map is traversed entirely by respective DREs. Further, once the DREs clean the identified space debris as per the plotted map, they return to the satellite platform.

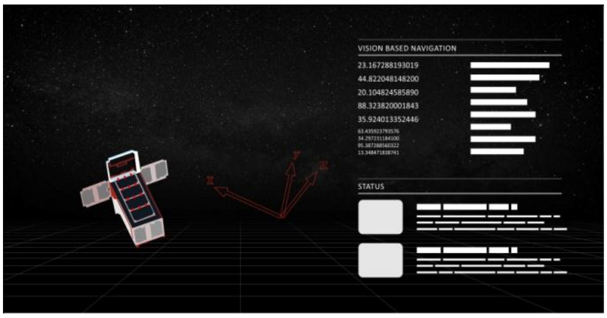

Figure: Vision-based Navigation

[Note: The tools used for cleaning up the junk may be chosen from plurality of solutions like net capture, magnetic nets, harpoon based solution, de-orbiting junk by using dragsnail architecture, etc [6]. All the above solutions have NeuroG developed 'map' incorporated in them which act as a reference for their individual navigation process. Thus, the three necessary modules are carefully embedded on the satellite platform used for the mission]

ALGORITHM – NEUROG

'NeuroG' works as follows:

Step 1: Start with a randomly generated population of 'n' chromosomes acting as candidate solution set to a problem.

Step 2: Calculate the fitness f(x) of each chromosome 'x' in the population.

Step 3: Repeat the following steps until 'n' off-springs have been created

Step 3(a): Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness (i.e. Roulette wheel selection). Selection is done "with replacement," meaning that the same chromosome can be selected more than once to become a parent.

Step 3(b): With probability pc ("crossover probability" or "crossover rate"), cross over the pair at a randomly chosen crossover point (chosen with uniform probability) to form newer offspring. If no crossover takes place, form two off springs that are exact copies of their respective parents.

Step 3(c): Mutate the two off-springs at each locus with probability pm ("mutation probability" or "mutation rate"), and place the resulting chromosomes in the new population.

Step 4: Replace the current population with the newer population.

Step 5: Go to step 2.

SIMULATION RESULTS

The TSP algorithm utilized in the prior art was studied & the algorithm was further modified and re-designed in a manner that would provide the most optimal solution for debris removal.

The metric considered for deriving the optimal solution is the 'distance metric'.

Let's consider a case study for cleaning-up '15' identified debris materials plotted on the geographic map of the target area. The standard TSP algorithm was run for the above case – wherein only one debris removal entity (DRE) was used for traversing and cleaning up all the 15 spots in space. The distance thus covered for cleaning up the debris turned out to be '1076.285 units' for one DRE.

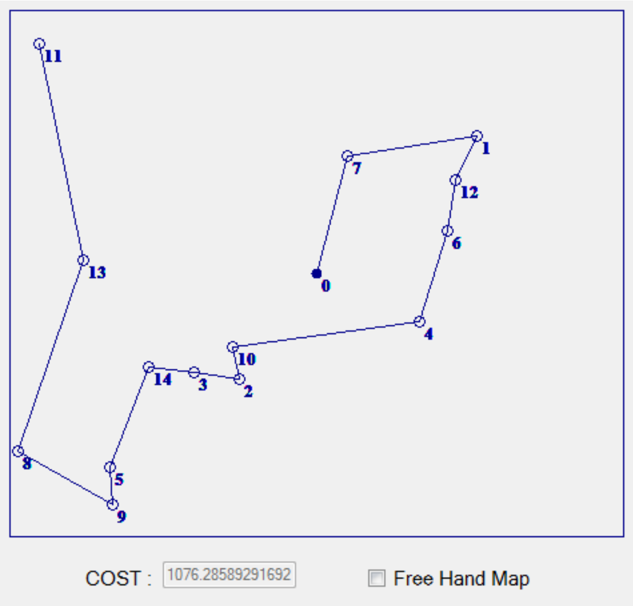

Figure: Prior-art – Single DRE cost = 1076.285 units

Now, NeuroG was run on the same sample dataset of '15' debris materials. NeuroG applies standard 'Genetic Algorithm' (GA) twice on the sample dataset. Since GA is applied twice on the same dataset, it is observed that the solution set obtained evolves from the previously obtained solution set to provide the most optimal solution in comparison to the case when GA is applied just once (e.g. TSP). Thus, the algorithm is termed as an 'adaptive evolutionary algorithm'. Further, the devised algorithm lets you input number of DREs available for debris removal task, which is not observed in the standard TSP case applied in the prior art work. The algorithm runs iterations for each set of entities. Once the iterations are completed, the algorithm intelligently figures out the most optimal number of DREs needed for accomplishing the mission in a cost-efficient manner. This ensures proper utilization of resources (i.e. DREs) & avoids the unnecessary expense overhead in the debris removal mission.

In the above discussed case, '5' DREs were available for the clean-up task of '15' identified debris materials. The algorithm calculates values for all five entities internally and outputs the most optimal solution. The results are as shown below:

1) For 'one' DRE: distance metric = 1076.285 units (prior-art)
2) For 'two' DREs: distance metric = 1012.676 units

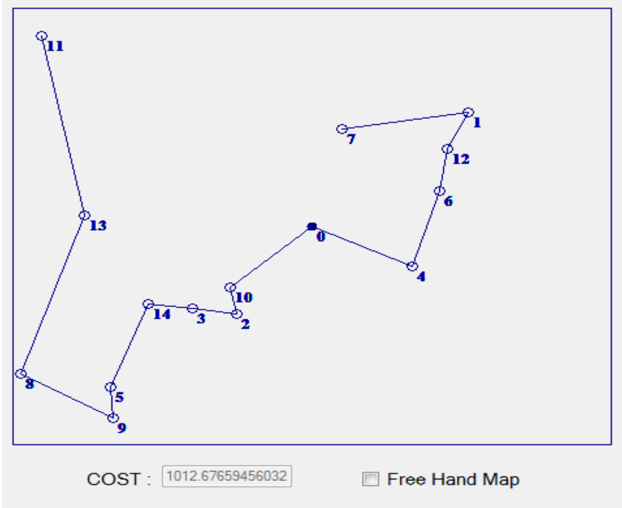

Figure: Two DREs cost = 1012.676 units

3) For 'three' DREs: distance metric = 1036.799 units

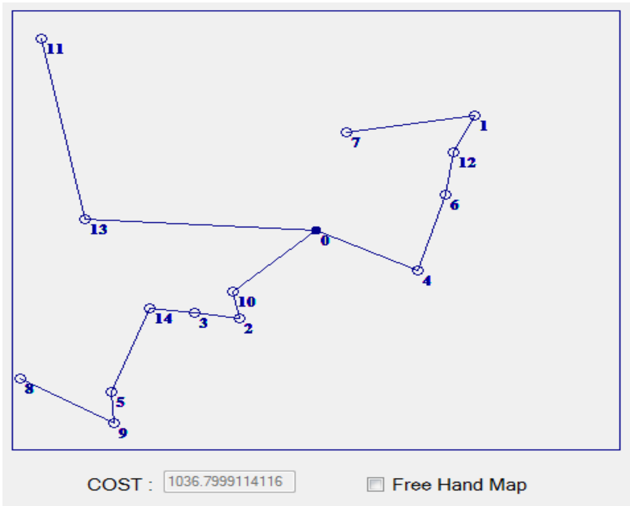

Figure: Three DREs cost = 1036.799 units

4) For 'four' DREs: distance metric = 1110.289 units

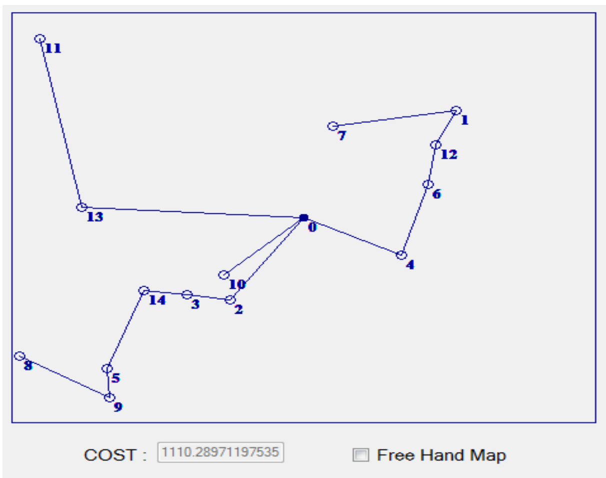

Figure: Four DREs cost = 1110.289 units

5) For ‘five’ DREs: distance metric = 1193.991 units

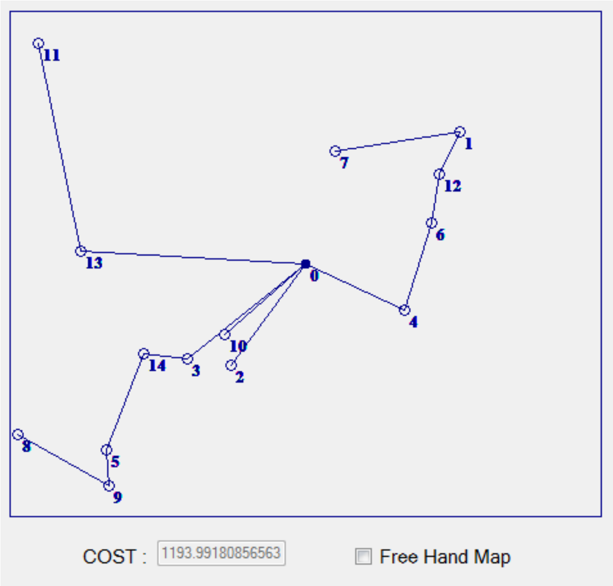

Figure: Five DREs cost = 1193.991 units

[Note: The calculated internal values cannot be viewed individually for each DRE set, but for validating the output of the algorithm we have run each case separately as well as in a combinatorial manner]

In the above example, the computed optimal set of DREs for debris removal turns out to be ‘two’ – wherein distance metric appears to be the lowest (i.e. 1012.676 units) amongst the available solution set. Once the computation is completed, the identified entity set (i.e. DRE set) is fired in appropriate directions (i.e. as evaluated by the NeuroG) from the satellite platform to perform the actual clean up task.

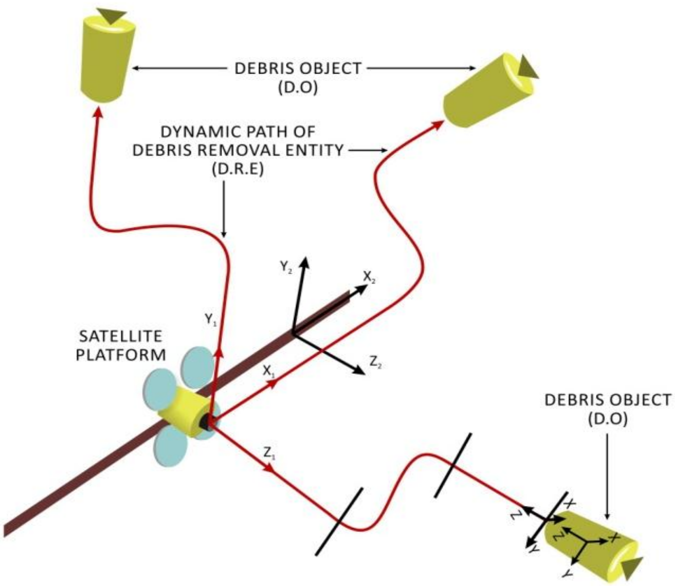

Figure: Satellite platform executing [NeuroG + Vision-based navigation]

Following table shows the main characteristics of the Neuro-Engine (NeuroG). The simulation results for cleaning ‘n’ no. of D.O (Debris Objects) by using optimal DRE (Debris Removal Entity) set = ‘m (Optimal)’ are disclosed in the below table. Further, the table presents comparative results for conventionally applied Genetic Algorithm (G.A) & modified algorithm used in NeuroG".

Table I: Simulation Results

Sr. No.	D.O (n)	DRE (m)	G.A (m=1) (Prior-art)	NeuroG m(Optimal)	Distance metric
1	17	10	1346.79	4	1259.61
2	22	12	1235.99	4	1177.01
3	26	15	1515.61	2	1396.62
4	20	17	1360.05	3	1263.63
5	13	12	1187.22	3	1019.82

CONCLUSION

Thus, the NeuroG and Vision-based navigation technique collaboratively handle static and dynamic part of space debris. We are able, for the first time, to simulate the optimal entity set required for cleaning up space debris in a defined territory by running the evolutionary algorithm. The neuro-engine thus designed shows a better success rate than the previously computed work for ‘space debris removal’.

REFERENCES

[1] Dario Izzo, Ingmar Getzner, Daniel Hennes, Luís F. Simões, “Evolving solutions to TSP variants for active space debris removal,” GECCO ’15, July 11 - 15, 2015, Madrid, Spain.

[2] Antoine Petit, Eric Marchand, Keyvan Kanani, “Vision-based Space Autonomous Rendezvous : A Case Study,” 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25-30 Sept. 2011

[3] Kozo Ohtani and Mitsuru Baba, Hiroshima Institute of Technology, Ibaraki University, Japan“Shape Recognition and Position Measurement of an Object Using an Ultrasonic Sensor Array”, "Sensor Array", *book edited by Wuqiang Yang, ISBN 978-953-51-0613-5, Published: May 23, 2012 under CC BY 3.0 license. © The Author(s)*

[4] Peter Ross, Dave Corne, “Applications of Genetic Algorithms”, AISB Quarterly on Evolutionary Computation, University of Edinburg, Department of Artificial Intelligence, Genetic Algorithms Research Groups, Paper No. 94 – 007.

[5] “Travelling Salesman Problem for Active Space Debris Removal”, http://esa.github.io/pygmo/examples/example7.html#tra

velling-salesman-problem-for-active-space-debris-removal

[6] Jesse Emspak , “How can humans clean up our space junk?”, Dec 30, 2016, https://www.theverge.com/2016/12/30/14116918/space-junk-debris-cleanup-missions-esa-astroscale-removedebris

[7] Larry Hardesty, “System designed to label visual scenes according to type turns out to detect particular objects, too.”, MIT News Office May 8, 2015