InterPlanet Computer Networking: Self-Replicating Networks in Managing a Fleet of Robots within Planetary Environment.

Poondru Prithvinath Reddy

April 2, 2022

InterPlanet Computer Networking: Self-Replicating Networks in Managing a Fleet of Robots within Planetary Environment.

Poondru Prithvinath Reddy

ABSTRACT

The interplanet internet is a conceived computer network in space, consisting of a set of network nodes that can communicate with each other. These nodes are the planet's orbiters (satellites) and landers (e.g. robots, autonomous machines, etc.) and the earth ground stations, and the data can be routed through Earth's internal internet. As resource depletion on Earth becomes real, the idea of extracting valuable elements from asteroids or using space-based resources to build space habitats becomes more attractive, one of the key technologies for harvesting resources is robotic space mining(minerals, metals, etc.,) or robotic building of space settlement. Robots have been designed to carryout various activities and also to work together in a site either to construct a habitat or unearth/mine the surface for minerals. Here in this paper, first we present an evolutionary algorithm which is an addressable information network based on natural selection/ selection of the fittest to implement interactive collaborative process which is a process of selecting the shape of robots body and its distribution of legs and structure that are automatically designed in simulation to perform a specific task. Also, we present a self-replicating robot which is a trained neural network program that is able to take in, react to the environment and create a copy of itself and form a pattern, much like our DNA. Here, we look at self-replicating program which is an intelligent program and produces a copy of its own code in similar capable machine/robot for execution of particular task. As it is planetary based architecture, the results could not be tested due to unavailability of large wireless networks over long distances however, the proposed implementation was tested with small population of sample candidates and the results suggest that we can achieve replication in robots by reproducing work

pattern in networked machines/resources to achieve desired objective of inter-planetary exploration/exploitation.

INTRODUCTION

Inter-planetary exploration, be it Lunar habitation, asteroid mining, Mars colonization or planetary science/mapping missions of the solar system, will increase demands for inter-planetary communications. The movement of people and material throughout the solar system will create the economic necessity for an information highway to move data throughout the solar system in support of inter-planetary exploration and exploitation. The communication capabilities of this interplanet information highway need to be designed to offer; 1) continuous data, 2) reliable communications, 3) high bandwidth and 4) accommodate data, voice and video.

The interplanetary Internet is a conceived computer network in space, consisting of a set of network nodes that can communicate with each other. These nodes are the planet's orbiters (satellites) and landers (e.g., robots), and the earth ground stations. For example, the orbiters collect the scientific data from the Landers on Mars through near-Mars communication links, transmit the data to Earth through direct links from the Mars orbiters to the Earth ground stations, and finally the data can be routed through Earth's internal internet. Interplanetary communication is greatly delayed by interplanetary distances, so a new set of protocols and technology that are tolerant to large delays and errors are required. The interplanetary Internet is a store and forward network of internets that is often disconnected, has a wireless backbone fraught with error-prone links and delays ranging from tens of minutes to even hours, even when there is a connection. In the core implementation of Interplanetary Internet, satellites orbiting a planet communicate to other planet's satellites. Simultaneously, these planets revolve around the Sun with long distances, and thus many challenges face the communications. The reasons and the resultant challenges are: The interplanetary communication is greatly delayed due to the interplanet distances and the motion of the planets. The interplanetary communication also suspends due to the solar conjunction, when the sun's radiation hinders the direct communication between the planets. As such, the communication characterizes lossy links and intermittent link connectivity.

The graph of participating nodes in a specific planet to a specific planet communication, keeps changing over time, due to the constant motion. The routes of the planet-to-planet communication are planned and scheduled rather than being fluctuating. The Interplanetary Internet design must address these challenges to operate successfully and achieve good communication with other planets. It also must use the few available resources efficiently in the system.

A region is an area where the characteristics of communication are the same. The Interplanetary Internet is a "network of regional internets". Examples of regions might include the terrestrial Internet as a region, a region on the surface of the Moon or Mars, or a ground-to-orbit region.

Self-replication is any behavior of a dynamical system that yields construction of an identical copy of itself. Biological cells, given suitable environments, reproduce by cell division. During cell division, DNA is replicated and can be transmitted to offspring during reproduction.

The ability to pass on successful traits is a defining characteristic of biological organisms. Researchers from Columbia University found a way to apply this principle to artificially intelligent systems — creating self-replicating neural networks called "quines." The idea of self-replicating, self-evolving AI that can automatically take on the most successful traits of previous generations is a pretty tantalizing one, with lots of potentially useful applications.

While self-replication has been studied in many automata, it is notably absent in neural network research, despite the fact that neural networks appear to be the most powerful form of AI known to date. In this paper, we identify and attempt in building and training a self-replicating neural network that integrates new networks. Specifically, we propose to view a neural network as a differentiable computer program composed of a sequence of tensor operations. Our objective then is to construct a neural network that integrates with other neural networks to replicate components and also a neural network that optimizes its own weights. The best solution for a self-replicating network was found by alternating optimization steps.

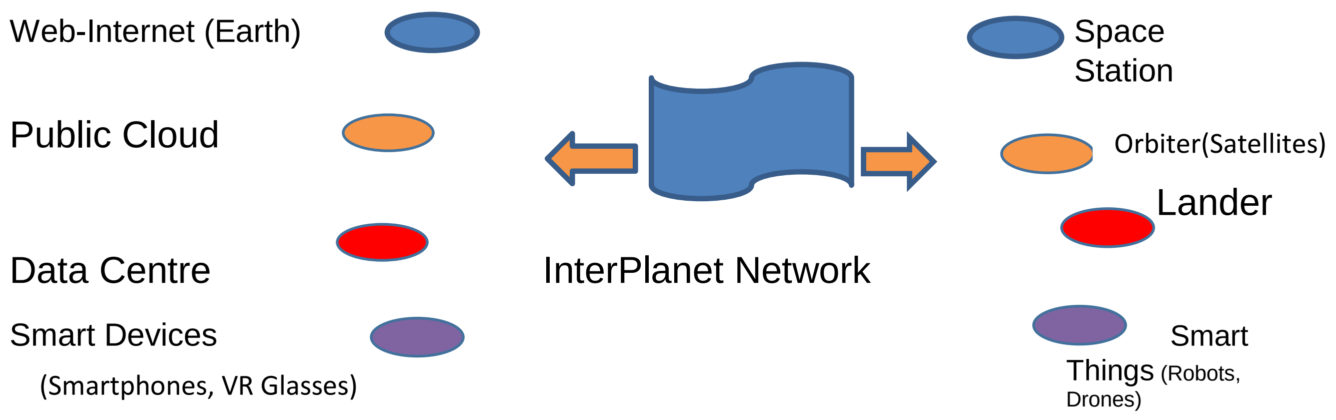
NETWORK ARCHITECTURE

A **Computer Network Architecture** is a design in which all computers in a computer network are organized. An architecture defines how the computers should get connected to get the maximum advantages of a computer network such as better response time, security, scalability, etc.

Network architecture refers to the way network devices and services are structured to serve the connectivity needs of client devices.

- Network devices typically include switches and routers.
- Types of services include DHCP and DNS.
- Client devices comprise end-user devices, servers, and smart things.

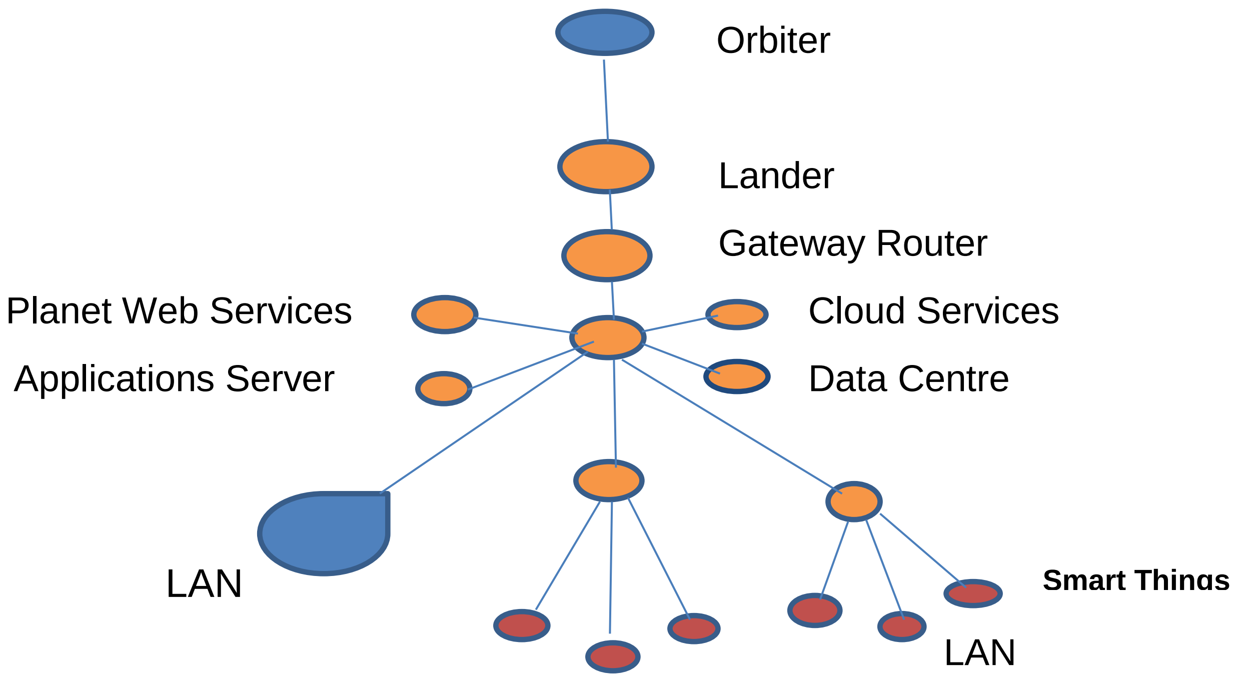
The network architecture for the planet Mars or the Moon is as shown in below figure:-



Computer networks are built to serve the needs of certain functionality and also their clients. Described below are three types of planetary networks:

- Access networks, for campuses and local areas, are built to bring machines and things onboard, such as connecting robots, drones, etc. within a location.
- Networks for data center connect servers that host data and applications and make them available to smart devices.
- Wide-area networks (WANs) connect robots and others to applications, sometimes over long distances, such as connecting robots to cloud applications related to space mining operations.

We give below the architecture of network on the planet Mars or the Earth's Moon is as shown in below figure:-



Local Area Network

An Internet is a "network of networks" in which routers move data among a multiplicity of networks with multiple admin. domains.

The main aim of networks is to connect remote endpoints with end-to-end principle and network should provide only those services that cannot be provided effectively by endpoints.

Since the networks are predominantly wireless, the fundamental impact of distance due to speed-of-light delays and impact on interactive applications – for both data and control is to be considered. Also power consumption of wireless links as a function of distance is to be examined.

The interplanetary internet is a conceived networks of nodes and these nodes are space station, planet's orbiters (satellites), planet's landers, robots (drones, autonomous machines, etc.), earth ground stations and earth's internal internet.

METHODOLOGY

Outer space contains a vast amount of resources that offer virtually unlimited wealth to the humans that can access and use them for commercial purposes. One of the key technologies for harvesting these resources is robotic mining of minerals, metals, etc. The harsh environment and vast distances create challenges that are handled best by robotic machines working in collaboration with human explorers. Humans will visit outposts and mining camps as required for exploration, and scientific research, but a continuous presence is most likely to be provided by robotic mining machines that are remotely controlled by humans either from Earth or from local space habitat.

Future **Moon(or Mars)** bases will likely be constructed using resources mined from the surface of the Moon/Mars. The difficulty of maintaining a human workforce on the Moon(or Mars) and communications lag with Earth means that mining will need to be conducted using **collaborative robots** with a high degree of autonomy. Therefore, the utility of autonomous collaborative robotics(with thousands of robots in operation) towards addressing several major challenges in autonomous mining in the lunar(Martian) environment with lack of satellite communication systems, navigation in hazardous terrain, and delicate robot interactions to achieve effective collaboration between robots and long-lasting operation.

Collaborative Robotics

Robots can be shaped to perform specific tasks. Robots have been designed and shaped in such a way that they can walk, swim, push pellets, carry payloads, carry shoveling and work together in a group to aggregate debris scattered along the surface into neat piles or possibly, to build a space settlement. They can survive for long-time without recharge and heal themselves after any damage/confusion. The shape of a robot's body, and its distribution of legs and structure, are automatically designed in simulation to perform a specific task, using a process of trial and error.

Robots are **collections of task executors** and have no brain system of their own. But in a real sense they can be programmed — to work autonomously and collaborate with other robots, as in this study, or eventually to do other things. tackling everything from space mining to deep space exploration.

Robots were able to move on their own. And using artificial intelligence, these robots can be programmed as specific executor of an assigned task for a number of situations and also using artificial intelligence to figure out the best shape for the Robots to perform in group on a more consistent basis to have better control over performance of assigned work.

Using a computational model that simulates the nature of work and everything of the Robot Capability, the process yields the robotic shape best suited to ensure the shape of the actual Robots into more efficient form suitable to a particular situation/task and accordingly enables robots to gather together in their environment forming them into groups with the same capability.

The methodology essentially consists of two parts :-

1. Optimization steps –The best solution for a self-replicating network by using evolutionary algorithm.
2. Integrating new neural networks

ARCHITECTURE

Optimizing Population of Solutions using Evolutionary Algorithm

Evolutionary Algorithm

Evolutionary Algorithms' (EA) constitute a **collection of methods** that originally have been developed to solve combinatorial optimization problems. Nowadays, Evolutionary Algorithms is a subset of Evolutionary Computation that itself is a subfield of Artificial Intelligence / Computational Intelligence.

An evolutionary algorithm (EA) is an algorithm that uses mechanisms inspired by nature and solves problems through processes that emulate the behaviors of living organisms. In EAs, the solutions play the role of individual organisms in a population.

The main classes of EA in contemporary usage are (in order of popularity) **genetic algorithms (GAs), evolution strategies (ESs), differential evolution (DE) and estimation of distribution algorithms (EDAs).**

Evolutionary algorithms are based on **concepts of biological evolution**. A 'population' of possible solutions to the problem is first created with each solution being scored using a 'fitness function' that indicates how good they are. The population evolves over time and (hopefully) identifies better solutions.

Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive. In other words, they simulate "survival of the fittest" among individual of consecutive generation for solving a problem. Although genetic algorithms are the most frequently encountered type of evolutionary algorithm, there are other types, such as Evolution Strategy and we have chosen evolution strategy as preferred option as it suits our requirement of collaborative management.

Evolution Strategy

In Evolution Strategy, we do not care much about the function and its relationship with the inputs or parameters. Some large population of numbers (parameters of the model) go into the algorithm and it squeeze out one value. We try to find the best set of such numbers which returns good values for our optimization and search problem.

The implementation process is as below:-

1. Randomly generate parameters(population species)
2. Generate a population with different parameter vectors by sampling with Gaussian distribution
3. Evaluate each candidate by running the model and based on the output value evaluate the loss or the objective function by binding it to Evolution Strategy.
4. We then select top/best performing elite parameters and take the mean of these parameters for finding our best parameter so far
5. We then repeat the above process by again generating different parameters by adding Gaussian distribution to our best parameter obtained so far and repeat the whole process till best solution is found

Integrating New Neural Networks

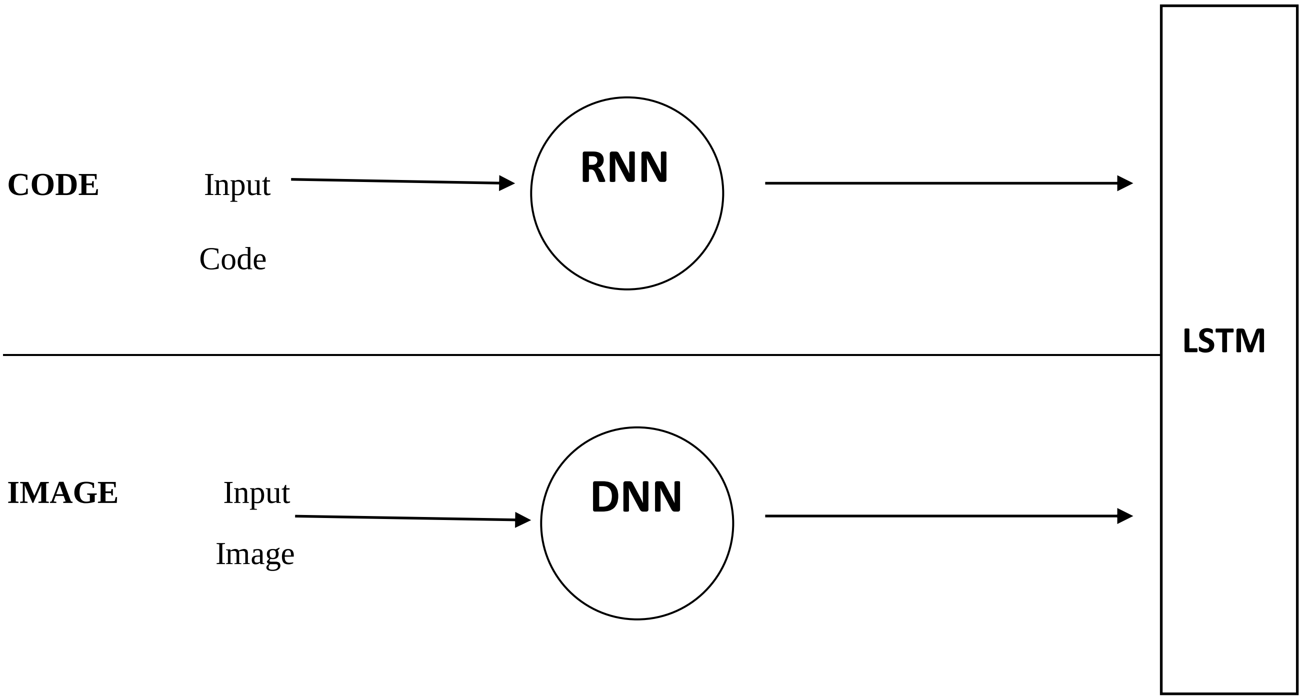


Figure 1: The INNN(Integrating New Neural Networks) architecture. It contains three subsystems that are trained separately. In the image subsystem, the encoder can transfer an input (or predicted) image into a population representation vector I at the DNN layer (mimicking the Deep Neural Network for high-level image representation), and the decoder can reconstruct a vector output from LSTM to a predicted image, which can be fed into the encoder to form the guided loop. In the code subsystem, The coding system which consists of a mapping to transfer symbol texts into respective numeric and a RNN to extract the sequence dependencies from the input texts, and an output encoder to convert numeric values into text symbols. There is a memory layer implemented by a RNN to extract sequence information from the vector C. The LSTM layer serves as working memory, that takes the concatenated input [C, I] from both code and image subsystems, and output the predicted next element representation that could be fed back into both subsystems to form a guided loop.

As is shown in Figure 1, the INNN network contains three main subsystems including the code, image and LSTM subsystems. The image encoder network was trained separately. After training, the encoder is separated into two parts: the encoder (or recognition) part ranges from the image entry point to the final encoding layer, to provide the high-level abstract representation of the input image; the decoder part ranges up to image prediction point. The activity vector of the encoding layer are concatenated with code activity vector as input signals to the LSTM. Finally, the predicted image is fed back to the encoder network for the next iteration. The code processing component first converts the input text symbol into a sequence of binary vectors [C(t = 0), . . . ,C(T)], where T is the text length. To improve the code recognition, we added one RNN layer to generate the sequence dependencies of the text. The LSTM training based on the next component prediction (NCP). The LSTM is trained by the NCP principle, where the goal of the LSTM is to output the representation vectors (including both code and image) of the next component which required the understanding of the previous text code and observed images. The LSTM subsystem contains a LSTM and a full connected layer. It receives inputs from both code and image subsystems in a concatenated form of c(t) = [C(t),I(t)] at time t, and gives a prediction output a a'(t) = [C'(t), I'(t)], which is expected to be identical to a(t + 1) = [C(t + 1), I(t + 1)] at time t+1. This has been achieved with a next component prediction (NCP). So given an input image, the LSTM can predict the corresponding code description. The strategy of learning by predicting its own next component is essentially an unsupervised learning. Our LSTM subsystem was trained separately after code and image components had completed their functionalities. Finally, we demonstrate how the network forms a thinking loop with text code and predicted images.

DATASET: User interface components / elements along with programming language scripts / code associated with each visual component has been selected as dataset. And we used images of different visual components / elements from User Interface elements dataset.

The Training is based on the next component prediction (NCP). The LSTM-FC is trained by the NCP principle, where the goal of the LSTM-FC is to output the representation vectors (including both code and image) of the next component / element. At time T, the LSTM of INNN generated the guided digit instance, which required the understanding of the previous code language and observed images.

The LSTM subsystem was trained separately after vision and code components had completed their functionalities. We have trained the network to accumulatively learn different components, and the related code results. Finally, it is demonstrated how the network forms a thinking loop with code language and observed images.

The LSTM layer serves as working memory, that takes the concatenated input [**C**,**I**] from both code and image subsystems, and output the predicted next component representation that could be fed back into both subsystems to form a guided loop.

RESULTS

Based on population of sample candidate solutions, and using visualization library that shows how the accuracy changes across each population. It is observed that after few iterations, on the MNIST dataset, we are able to find good enough accuracy . Therefore, evolutionary model as an optimization and search approach that mimics the concept of natural evolution / natural selection is encouraging.

Also, it is observed that after limited iterations on the visual components dataset, we are able to find an accuracy that is satisfactory. However, using different values for the parameters such as more number of iterations might increase the accuracy.

CONCLUSION

The interplanetary computer network in space is a set of computer nodes that can communicate with each other. We proposed a network architecture with planet's orbiters, landers (robots, etc.), as well as the earth ground stations and linked through Earth's internal internet, and consisted of complex information routing through relay satellites to address direct planet-to-planet communication. This paper presents an addressable information network based on the evolutionary process which is based on natural selection, or the selection of the fittest and the experimental implementation is based on the evolutionary strategy for selecting fittest robots for the desired task of either space settlement or mining the space. Also, we presented a self-replicating neural network program that is able to react to the environment and create a copy of itself and form a pattern. The paper demonstrated a collaborative network that can be operated by reproducing the evolutionary intelligence and self-replication in managing a fleet of robots in both space colonies as well as at space mining site. As it is planetary based architecture outline, the proposed approach would be effective in addressing replication by reproducing work pattern in networked machines/ resources at the planetary level to achieve desired objective of inter-planetary exploration/exploitation.

REFERENCE

1. Poondru Prithvinath Reddy: “**InterPlanet Computer Networking: An Evolution-enabled Reconfigurable Robotic System for Space Applications”,** Google Scholar

2. Adam Slowik & Halina Kwasnicka: **“Evolutionary algorithms and their applications to engineering problems”,** Springer Link

3. Poondru Prithvinath Reddy “Artificial Intelligence that Learn to Write Code : Memory Guided Programming”, Google Scholar