Automated Workforce
an Approach to Colonizing Space

Russell Stephenson
Florida State University, Tallahassee, FL
ras09g@my.fsu.edu

Abstract

My work comes from the idea that robots can be programmed to work like humans, or even better than humans. Visually linking robots together can help them work as a single unit since it eliminates the common problems such as a person holding a beam while another bolts it into place but the person holding the beam has to move it around to get it into the right spot. If the person holding the beam could see where the beam needs to be then it would make the job a little quicker. At this point I am trying to make the program the robots to delegate the work needed to be done efficiently given known materials. This is assuming the robots can visualize the materials, place them into a size category, and perform the necessary work. The main issue is making the robots work together as a team in an ever changing environment. There has been work done to program robots to form coalitions to accomplish tasks that the robots could not complete on their own. The computations are simply too complex to program compared to a simple assembly line like those used in the automobile industry.

Introduction

Over the years the Earth has become polluted and eventually humans will need to migrate to space due to overpopulation or simply to help the earth recover from the years of misuse. To do this we would need to build colonies that would orbit the Earth. It would be impractical to build it on earth because the amount of thrust needed to propel something as massive as a colony would be tremendous. It would also be impractical to use humans to build them due to problematic things like air supply, time, and lack of manpower able to be in space at one time. The most rational thing would to develop a robotic workforce for space that would have a certain level of common sense when making decisions about whether something would be a multi-robot job or a single robot would suffice. This would eliminate the need for air, allow for easier navigation since space has few obstacles, and eliminate fatigue, aside from recharging time. The robots would work together as a single cohesive unit linked electronically and make decisions together based on priority of what is being done and delegate work based on the number of units available. I am assuming that a robot can identify parts, correctly pick the parts up and set them into place, and do any work needed to assemble the part. The issues to point out are the lack of the ability to make complex computations, programming common sense in a work related environment, a robot's ability to learn from extensive use and improving on methods to accomplish tasks, and making the robots work together without interfering with another robot or the environment. Previous work has been done to program common sense and learning as well as cooperative working to accomplish tasks as a single unit.

Previous Work

Coalition resource game

Work has been done to program robots to work together as a unit to accomplish tasks. The CRG model was created and improved upon by (Chitnis Hajiaghayi, and Liaghat 2011). The original model contained a nonempty set Ag = $\{a_1, a_2, \ldots, a_n\}$ of agents, a coalition C consisting of multiple agents, and a grand coalition consisting of all agents. The model also contains a finite set of goals, denoted G, with each agent i being a subset of G_i. An agent is satisfied if at least one goal is accomplished within the set of goals. To achieve these goals a resources are expended, much like building. The total amount of resources are denoted by R. There exist a required number of resources to accomplish each goal. Each resource has a finite quantity.

CRG, denoted Γ, is a (n + 5) tuple denoted:

$$\Gamma = \{A_g, G, R, G_1, G_2, \ldots, G_n, en, req\}$$

Where,
A_g = the set of agents
G = the set of goals
R = the set of resources
En = the endowment function
Req = the requirement function

The endowment function makes the sum of all endowments of the members known. Likewise the requirement function sums up the amount of requirements needed.

Learning from repeated games

Naturally it would be good to allow a robot some form of reasoning such that it could accomplish repeated goals quickly without making unnecessary calculations. To solve this problem multi-agent learning could be used in the form repeated scenarios that could cause the robot to learn from repeated events. This is far from a complete method but it has been improved upon over many years. Multi-agent learning works but "setting assumptions, representations, and rules that algorithms encode to determine a strategy from experience." Doing this allows you to form a learning bias that could produce an optimal outcome. A strategy space must be created to form the learning bias. This space consists of a probability distribution for each state of the world. The probability distributions made by the creators of the multi-agent learning algorithm had three distributions. The first distribution consisted of a set of all strategies to accomplish a given task. The second consisted of just the "pure" strategies. The last consisted of filler "mixed" strategies to aid the pure strategies if the pure strategies could not accomplish the task. The pure strategies are quicker in terms of reacting to a changing environment, but are sometimes limited in their execution. The states of the world are usually redundant states known as "stateless". However what I am proposing will always have a changing outer environment, but the tasks are not always completely changing since many tasks share similarities to other tasks. This concept is similar to "recurrent state" which uses information from a previous task and uses it to define the current state.

Particle Filters in Robotics

The last problem faced in this proposal is keeping the robots from running into each other and other things. Recently particle filters have made leaps in solving robotic problems. Two of the most important problems were global localization and the kidnapped robot. The kidnapped robot is the ability of a robot to figure out its location supposing it were disoriented or lost. This allows a robot to be more massive without needing to worry about it bumping into things. Particle filters consist of sequential Monte Carlo algorithms. For space a robot would need a full 360° sphere to be scanned which will likely cause problems. This is because the probabilistic model will always be wrong when dealing with the state. One example is a robot probing underground tunnels will never know exactly where it is and will not be able to accurately show a diagram of the tunnels until it makes a full circuit.

Method

To accomplish most of this in a program I needed to make the robots separate entities. To do this I just make a structure consisting of one variable: is the robot busy? This allows me to divide the work among robots that are busy versus those that are not. Using the coalition resource game equation I made several data structures to store the goals and the requirements of each goal. The agents are the robots. The resources are a vector of pairs consisting of the resource name and the amount of the resource. If the amount of the resource is required but not available then the job cannot be accomplished. While the structure being constructed is not complete a loop is iterated through. First a robot finds a goal and checks how many robots are required to accomplish the goal. Then the robot checks the resources required to accomplish the goal and gathers those resources. Then the robot enters another while loop to figure out how to put the part together each time checking to see if it performed the correct method.

Machine Learning

Once the robot figures out the correct method it stores that method into its memory under a certain category such as electrical or structural. It then has completed the task and removes that task from the set of goals and finds another task. If the new task is the same or similar as the first task then it searches its memory for a task that is similar to the new goal. The robot then compares the materials needed for the task in its memory with the new goal and it starts performing the steps of the old task until it reaches an unknown part. When it reaches this part it stores the method so far in a short-term memory so it can easily get back to where it was. Then it performs combinations again over a smaller scale to figure out how to accomplish the new goal and saving it into memory once that is accomplished. This allows for a quicker assembly time after the first assembly of similar items in the same category.

Localization

As of right now I will be unable to perform tests on localization without cameras and actual robots. As such I will save that for future improvements in my experiments. As for the problem of localization I would normally suggest using sound combined with the particle filter method as sonar. Comparing the two methods would yield a relatively accurate estimation of distance. The problem lies in the fact that there is no sound in space so sonar wouldn't work. The method of particle filtering would have to be improved since it is inaccurate to a degree. Likely two 360° cameras would have to be developed and used to calculate distance much like the human eyes. From this the distance could be measured from all parts of the robot to the obstacles of the environment several times a second. The calculations to do this would likely be difficult, somewhat inaccurate, and require a great deal of processing power. However in space, options are limited to measure locality. It's similar to scuba diving at night, because when a person orients their self a certain way it will disorient them and cause them to lose their sense of direction i.e. down will seem like up.

Theoretically this is a relatively easy process to program. However many assumptions are made; robots successfully being able to identify a part by scanning over it with a camera would be hard to program. For instance a structural beam could be different sizes. How can a robot determine that a 10ft. beam is a beam just like the previously identified 5ft. beam? Furthermore, how will the robot know when to use the 5ft. beam over the 10ft. beam? Likely another program would have to be built to measure the distance from the robot to the structural component and then have the robot determine the size, but this size could just be a rough estimate. Another problem would be the robot determining if a room is air tight while welding the structure together. Is the structure strong enough to hold air? How accurate can the robot weld the materials? Then there is electricity; electricity is even more complicated to understand than building for a human. A robot would likely not know where to begin to assemble a simple light switch. Many things would have to be preprogrammed into the robot's memory before it could sufficiently perform such actions as these. Even after being programmed extensive testing would be performed in a near weightless state such as underwater before it would even be considered for space adaptation. While this experiment is being done many of these questions are being asked and assumptions are being made to narrow the experiment down to fundamental ideas. At present I can only demonstrate machine learning and cooperative work in a goal oriented environment with limited resources and no knowledge of the goals other than what is being presented to the agents.

Results

As demonstrated in the attached program a single robot operates to complete goals. I would like to note that at present each goal is a single robot job. During the first cycle the robot doesn't know what it is doing and must complete a goal before gaining knowledge. Then it is able to complete similar jobs in less time. While these jobs are similar in a programming environment, they may not be so similar in a real world environment. Nevertheless the robot completes jobs quicker after gaining proficiency in a certain area. Naturally if a robot has installed a light switch once already then it can install another in just one cycle without the need to figure out how it the correct method to install it. Another thing I would like to point out is like humans, the robot in question learns independently from other robots; if one learns how to install a light switch then the other must learn in its own. However in an actual adaptation it would be preferable to allow the robots to share their memory with other robots to optimize the system. The program also demonstrates the coalition resource game since the robot fetches the materials and cannot use the same resource for another project and it shows a world of scarce resources and requirements to finish a goal. Overall the results were as expected in demonstrating machine learning through repeated trial and error. It also functioned properly to show how this could theoretically be implemented given the assumptions were true.

I could not figure out a good method for picking a correct assembly method from a combination of strings in an array due to time constraints and the complexity of the algorithm when comparing to data containers. As such if this project were to demonstrate the coalition resource game then it could be called a success.

Due to being unable to experiment with particle filters and localization, I could not imagine how my idea would function in a real world situation given the limitations I placed on the experiment. Likely particle filtering alone would not be effective enough to work without the robot becoming disoriented and resulting in the kidnapping problem without being able to recover. In a tunnel or even underwater would be far easier than figuring out how to implement spatial localization in space. More so since the environment is constantly changing as the robots complete goals. This is likely the most difficult problem and until it is resolved further implementation would be futile.

Conclusion

Artificial intelligence is still a relatively new field but it has experienced much innovation over the years. Many of these methods started out as a problem many years ago but are now resolved or partially resolved. All of these methods still have a great deal of work since problems with the current implementation have been noted. As such people are working to improve each of these for more demanding usage. As noted above spatial locality is far from perfect, however, we now have the ability to allow a robot to move around a museum and dodge obstacles. Machine learning is also a puzzling since we can teach robots to think similar to an ape through repeated games and methods. However there's still something missing to make a robot think like a human. It's possible we are incapable recreating a human brain, thus making machine learning on a grand scale impossible. But it wasn't too long ago that human's flying seemed like it was impossible. However nothing is impossible until it is proven impossible; even then it may not be as impossible as some think. I think space colonies will be possible in the near future, although not likely in our lifetime. It would still be a good idea to plan for in case of a world catastrophe or simple overpopulation. Maybe even excess pollution if we don't shift to a more environmental friendly society.

Future Work

In the future I would like to experiment with particle filtering and try to solve the mystery for the quintessential method for special locality in robots; especially in an open area where the ground is not flat and walls cannot help guide a robot. I feel it would be a tremendous leap in robotics. I would never condone using robots for employment areas except underwater or orbital construction. This is due to robotic labor being free and negatively affecting the job market. I would also like to continue with machine learning with this application. I don't have aspirations to make robots that act and think like humans. Creating a robot to perform certain tasks would be far easier and more beneficial to society. As I noted before there are many challenges faced with trying to accomplish this task. This is mostly due to a robot not having the same awareness and carefulness of a human when constructing buildings. Presently robots are unable to think, care, or even understand safety. There have been tests to help identify human emotion and the severity of each emotion such that a robot could understand it but research still needs to be done to completely implement this. Ultimately in my lifetime I would like to make all of this possible through further research so the human race could establish a home in space with relatively easy access to and from earth.

References

Cohler, Yuga J., and Lai, John K., and Parkes, David C., and Procaccis, Ariel D. 2011. *Envy Free Cake Cutting*. Association for the Advancement of Artificial Intelligence.

Chitnis, Rajesh, and Hajiaghayi, MohammudTaghi, and Liaghat, Vahid. 2011. *Parameterized Complexity of Problems in Coalition Resource Games.* Association for the Advancement of Artificial Intelligence.

Kapoor, Ashish, and Desney Tan, Bonshin Lee, and Horvitz, Eric. 2011. *Learning to Learn: Algorithmic Inspirations from Human Problem Solving.* Association for the Advancement of Artificial Intelligence.

Crandall, Jacob W., and Ahmed, Asad, and Goodrich, Michael A. 2011. *Learning in Repeated Games with Minimal Information: The Effects of Learning Bias.* Association for the Advancement of Artificial Intelligence.

Park, Kyoungup, and Shen, Chunhua, and Hao, Zhihui, and Kim Junae. 2011. *Efficiently Learning a Distance Metric for Large Margin Nearest Neighbor Classification.* Association for the Advancement of Artificial Intelligence.

Shrot, Tammar, and Aumann, Yonatan, and Kraus, Sarit. 2009. *Easy and Hard Coalition Resource Game Formation Problems- A Parameterized Complexity Analysis.* International Foundation for Autonomous Agents and Multiagent Systems.

Thrun, Sebastian. 2002. *Partical Filters in Robotics.* Proceeding of Uncertainty in AI.