Influencing the Task Distribution in Evolutionary Robotics

Kim Bosman
VU University Amsterdam
De Boelelaan 1081
1081 HV Amsterdam
k.bosman@student.vu.nl
2524547

Abstract

In 2013, Haasdijk and Bredeche [5] performed an experiment focusing on evolution of a robot swarm. For this experiment, an algorithm called MONEE (Multi-Objective aNd open-Ended Evolution) was used. MONEE allows for both task-driven and environment-driven evolution, meaning that both decisions and environment can influence the way the swarm evolves. MONEE allows the user to influence the task-driven evolution by adding a premium value. This premium value changes the importance of a task in comparison with another. Additionally, the original algorithm incorporates a market mechanism, which normally makes sure that the robots will not only focus on the more valuable tasks. This paper further explores the possibilities of the premium, but will exclude the market mechanism for the simulations. By excluding the market mechanism, the hypothesis can be made that there will be a clear distinction in the task performance between the different premiums.

Our results show that the market mechanism does influence the task ratio, however not significantly.

1 Introduction

Evolution is a given fact for both animals and plants. Animals and plants have a need to evolve in order to keep up with their ever changing environment, but humans have a need for robots to evolve as well. As Eaton [1] describes in his book "Evolutionary Humanoid Robotics", the most important reason we need robots to evolve is so that they can come up with useful engineering artefacts that might be difficult or even impossible to create by other means.

Evolution is defined as the slow process of adapting to an ever changing environment. Within the Computer Science field, evolution is performed through Evolutionary Computing [2]. Evolutionary Computing is the application of the Darwinian principles of natural selection to robots. Evolution can be employed for two goals: to provide a force for adaptation to the environment and to provide a force for optimization. The first goal makes sure that a population survives and the second goal makes sure the population makes itself useful [3]. With the recent increase in both interest in and successes of artificial life, a question has risen to mind: can robots experience evolution the same way plants and animals do? To research this subject, a paradigm named MONEE (Multi-Objective aNd open-Ended Evolution) was created by Haasdijk et al. [3]. MONEE allows a combination of both task-driven and environment-driven adaptation, which allows a whole new spectrum of testing and simulating hypotheses.

This paper is an extension on the work of Haasdijk et al. and will focus on the market mechanism that is in place in his paradigm. The Business Dictionary [4] defines a market mechanism as 'a mean by which the forces of demand and supply determine prices and quantities of goods and services offered for sale in a free market'. In MONEE's case, the market mechanism applies to the tasks that can be performed. The mechanism regulates the corresponding value of each task based on their scarcity. To influence these values, a premium can be set. This premium can both increase and decrease the worth of one of the tasks and thus change the value. An increase in worth will make them more valuable while a decrease in worth will make them less

valuable, and thus influence the priority of the tasks.
Haasdijk and Bredeche [5] performed an experiment in 2013 researching the influence of different premiums while the market mechanism was activated. Our research aims to examine the results of using different premiums without a market mechanism. The corresponding research question will be: “How would the swarm react to different valued tasks without a working market mechanism?” By making a comparison between a simulation with working market mechanism and a simulation without one, a better insight in the swarm’s evolutionary needs and priorities can be formed. It could be verified that the swarm will indeed aim for the more valuable task.

This paper will introduce the subject with an overview of work that has already been conducted on this topic. After that, the method used to perform the experiments will be explained and the results of these experiments will be analyzed. Finally, a conclusion will be made and possible future work will be discussed.

2 Related Work

The predecessor of MONEE is mEDEA [6], which is a minimal Embodied Distributed Evolutionary Algorithm. This algorithm describes how evolution is handled on a local basis. Within this algorithm, autonomous robots move around freely while exchanging their genome with other robots they come across with. When the robot’s lifetime comes to an end, a random genome of all the received genomes will be selected and altered. In MONEE, this algorithm is reused and modified by adding the possibility to assign certain tasks the robots should perform. Adding specifically defined tasks ensures that not only the environment-driven but also the task-driven adaptation will be evaluated.
In a MONEE simulation, autonomous robots will move around in an arena which contains both obstacles and pucks. The given task is to collect as many pucks as possible before the lifecycle of a robot comes to an end. When the lifecycle of a robot ends, it will change into an egg. While the robot is in this egg phase, it will not move or perform any actions except receiving genomes from robots that are passing by within a specific range, also defined as the communication distance. This egg phase lasts a predetermined time, and when time runs out the egg will choose one of the robots that have passed by as a parent by comparing the final puck counts of all the genes the robot has collected in the egg phase. Once a satisfactory parent is selected, this genome will be used to start a new lifecycle.
In order to influence this parent selection, the user is able to define a premium. This premium changes the value of one of the tasks, shifting the interest of the swarm to collect the task that will generate the highest score. The standard premium has a value of 1 and suggests that task 0 is as valuable as task 1, taking into account that the task assigned to the robots is collecting both puck type 0 and puck type 1. Increasing or decreasing this premium will change the task value and thus the ratio. The parent selection is performed by multiplying the premium with the number of type 1 pucks collected. So, a premium set to -1 would hinder a robot collecting type 0 pucks, but a premium set to 100 would increase the chances of being selected as a parent considerably. By calculating this exchange rate, the egg makes sure that the easier tasks will not outweigh the harder tasks. After calculating the final puck score of each robot the egg had an encounter with, the egg will compare the results and pick the best robot to use the genome for a new lifecycle.
Haasdijk and Bredeche [5] extended their research in 2013 to see to what extent the premium has an effect on the swarm. In this experiment, puck type 0 corresponds with green pucks and puck type 1 with red pucks. As predicted, a premium of -1 results in a drastic decrease of the collected green pucks. Setting the premium to 0, which minimizes the value of the green puck to 0, decreases the number of green pucks collected to a ratio of 0.05 compared with the red pucks. This means that the swarm keeps collecting some green pucks even though these have no value. A premium of 1 generates an equally distributed collection of both green and red pucks. Strangely enough, increasing the premium did not generate the expected results. Using premium values larger than 1results in more green pucks collected until a certain premium is reached. The increase flattens and the amount of collected pucks stays about the same for each premium thereafter. In other words: the number of green pucks collected is higher than the number of red pucks collected, but this ratio is equal for a premium of 10 and a premium of 100. These results can be reviewed in figure 1.
One could argue that the market mechanism holds a great part in these results, since the pre-

mium pucks are not any less common than the pucks with a normal value. This would mean that the market mechanism corrects the value of the premium pucks, making their value comparable to the other pucks on the field.

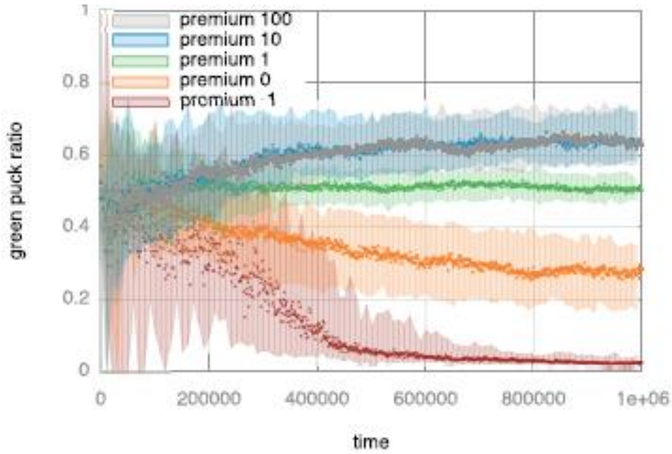

Figure 1: The results of Haasdijk and Bredeche in 2013, showing the green puck ratio of the corresponding 5 different premiums set before the experiments. The points stand for the median green puck ratio of 1000 time steps and the shades represent the lower and upper quartile

As Barabási [7] mentioned in his paper "The origin of bursts and heavy tails in human dynamics", human task selection is based on a priority list. The tasks with the highest priority will be performed first and when this task is done the task with the second highest priority will be executed. This cycle continuous until the task with the lowest priority is reached. If humans would perform the puck collection task, the pucks with the highest value will be the first ones to be collected since they will be assigned the highest priority.

The pending question is the manner in which robots perform their tasks. Within MONEE, the learning cycle is purely based on evolution. Robots with a higher puck count will be more desirable and thus will have more descendants. In order to live on and evolve along with the swarm, the individual robot has to keep up with the mean puck count and preferably even aim for a higher puck count in order to increase their chances. Eventually, the number of robots that cannot keep up will reduce or disappear entirely due to the lack of descendants. The swarm evolves and the puck ratio will change along with it.

It is expected that, when there is no market mechanism in place, a robot will always choose the puck which holds to most value since a higher puck count will lead to a more desirable robot and eventually to more descendants, keeping the bloodline alive. Using the same premium settings as Haasdijk and Bredeche [5] used in their paper, a few hypotheses can be made.

First of all, using a premium of 0 or -1 will have a negative influence on the collection of type 0 pucks. Collecting type 0 pucks with a premium of 0 would be a waste of the robot's time and effort since this puck will hold no value. A type 0 puck with a premium of -1 would even decrease the total puck count, which is a negative consequence. Secondly, a premium of 1 will result in a 1:1 ratio. Both type 0 and type 1 pucks will hold an equal value, leading to the robots not having to make a distinction between the two. Thirdly, setting the premium to 10 should increase the number of type 0 pucks collected and decrease the number of type 1 pucks collected. It will be 10 times more rewarding to collect type 0 pucks. Finally, a premium with the value of 100 will create an even bigger increase of type 0 puck collections and an even bigger decrease of the collected type 1 pucks.

These hypotheses can be summarized as follows: the swarm will adjust its behavior according to the premium value assigned, making sure that it will perform the task that generates the highest points.

3 Method

The MONEE algorithm mentioned and explained above will be implemented in a simulator named RoboRobo [7]. RoboRobo is a multi-platform robot simulator for large-scale collective robotics experiments. Each experiment will be repeated 32 times, which is the exact half of repetitions Haasdijk and Bredeche [5] used in their experimental setup. Since this experiment will be an extension on the work of Haasdijk and Bredeche, most of the settings will be kept the same. This results in a more accurate comparison of the final findings.

The experiment will feature 100 agents and 300 pucks. These pucks are divided into two kinds: type 0 and type 1. The pucks represent two different collection tasks the swarm can perform, meaning that they either collect puck type 0 or puck type 1. Just as in the experiment Haasdijk and Bredeche performed, a premium value can be set. By changing this premium, the value of puck type 0 will be changed. This can either increase or decrease the value and thus the interest in performing the corresponding task. The tasks in this experiment are simple: collect the pucks on the board. Each time a puck is collected it will disappear and immediately reappear in a random location, so the total number of pucks on the board will be constant at all times. With a constant number of pucks on the board, the experiment will not be affected by a shortage or a surplus of pucks.

As mentioned before, the MONEE algorithm is based on evolution. The swarm evolves through gene selection. The robots move around in the arena, collecting pucks as they go. The more positive premium pucks a robot collects, the more likely the genes will be selected. The more negative valued premium pucks a robot collects, the less likely the genes will be selected. After the lifetime of a robot has ended, which is after 2000 time steps, the robot will change into an egg, entering the egg phase that will last 200 time steps. This egg will receive genomes and puck statistics from bypassing robots. After the egg phase is finished, the genes of one robot will be used to reproduce. This genome will be selected by multiplying the final puck count with the premium and comparing these results. The higher the puck count of the task with the higher premium, the higher the chance their genome will be selected. Each simulation will have a total of 1,000,000 time steps.

The premium is set to influence the value of a task. As in the work of Haasdijk and Bredeche, the standard setting is a premium value of 1, which implies that task 0 is as valuable as task 1, making them equal to each other. In the experiment Haasdijk and Bredeche performed in 2013, 5 different premiums were set: -1, 0, 1, 10 and 100. Where their experiment involved making use of the market mechanism, this experiment will not. In order to make the best comparison as possible between the two experiments, the same premiums will be used. The summarized experimental setup can be found in table 1.

Experimental Setup	
Robot group size	100
Number of pucks	300
Puck types	2
Number of repeats	32
Simulation length	1,000,000 time steps
Premium settings	-1, 0, 1, 10, 100
Robot lifetime	2000 time steps
Egg lifetime	200 time steps

Table 1: An overview of the experimental setup

After the experiment is repeated 32 times, the analysis can be executed. The analysis summarizes all the repeats and combines them into 1 document, showing the total pucks collected per 1000 time steps for both the combined tasks as each task apart. The analysis will also analyze the inseminations so that the descendants can be traced back to the original parents.

4 Results and Analysis

Figure 2 summarizes the results of the experiments performed for this research. It shows the mean number of pucks collected of each type per premium.

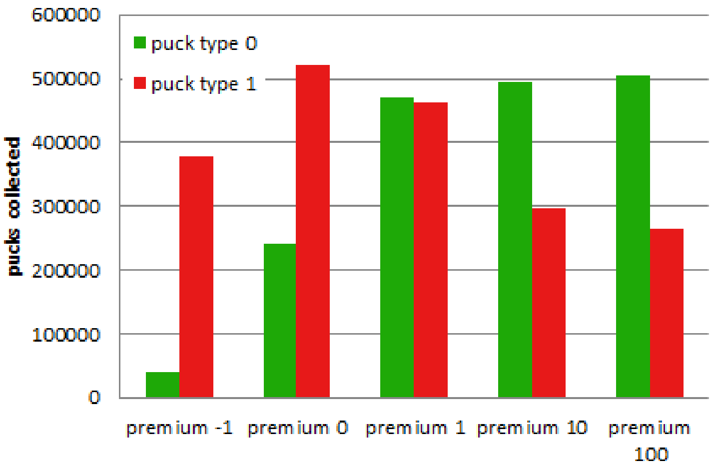

Figure 2: Mean total pucks collected over the 32 runs, divided in the different premiums

As one can clearly see, the collected puck ratio differs for each premium. Setting a premium to the value of -1 leads to a lower collection of puck type 0 than when setting the premium value to 0. As predicted, a premium value of 1 generates a close to equal value of collected pucks. For premium 10 and 100, the collected pucks of type 0 are almost equal, where the collected pucks of type 1 differ with 30000 pucks. Notable is that the total number of pucks collected differs for each premium, which has a logical explanation. In each run, the total number of pucks of each type will stay the same, namely 150. If one of these types causes a negative reaction when being collected, this type will be avoided in order to keep the evolution as strong as possible. This leads to an extra desire of the other puck kind, which has 150 of them on the field as well. Instead of being able to choose between 300 pucks, the robots now have to fight over the 150 desired pucks on the field. Not wanting to collect one half of the pucks and having to fight over the other half of the pucks will lead to a decrease in the total number of collected pucks. A premium of 0 will have the same effect, only less severe. Collecting pucks with a premium of 0 will be undesirable, but not as undesirable as pucks with a premium of -1. This leads to an increase of total pucks of type 0 collected. This increase also means that the focus is not only on puck type 1 anymore, making it easier to collect type 1 and thus leading to an increase of both puck type 1 and puck type 0. This is also why the total pucks collected by a premium value of 1 are the highest. These pucks have an equal value, making the desired puck versus robots on the field a 3:1 ratio. The pucks will be easy to collect, making the total puck count higher. When the premium is set to a value of 10, the opposite happens. The type 0 pucks become more desirable and thus will have a higher collection count, leading to a lower count of type 1 pucks since the focus shifts to puck type 0. This trend can also be found with a premium value of 100, although the difference is not as tremendous as one would expect. The precise numbers can be found in table 2.

	Premium -1	Premium 0	Premium 1	Premium 10	Premium 100
Type 0	40093	239955	470400	494727	504800
Type 1	378714	521171	463947	296128	263593
Total	418807	761126	934347	790855	768393

Table 2: The mean total pucks collected for each type of each premium and the total number of pucks collected per premium

Figure 3 shows the development of the puck type 0 ratio over the 1.000.000 time steps of a simulation. This ratio is being used to assess the usefulness of the premium values. Each premium always start out with a 1:1 collection ratio, since this is the first generation of robots and the rules still need to be figured out. After roughly 125.000 time steps have passed, the distinction between the different premiums can be seen clearly. At this point, the puck type 0 ratio for the premium value of -1 has already decreased to 30% and will continue dropping until it reaches 5%, indicating that only 5% of the collected pucks are type 0 pucks. The statistics for the premium value of 0 also show a decrease in the puck type 0 ratio, but this decrease is shallower than the decrease that premium value -1 shows. Premium 0 ends at a 30% ratio for puck type 0. As figure 3 shows, the premium value 1 generates a 1:1 ratio, showing as a straight line in the figure. A premium value of 10 increases the

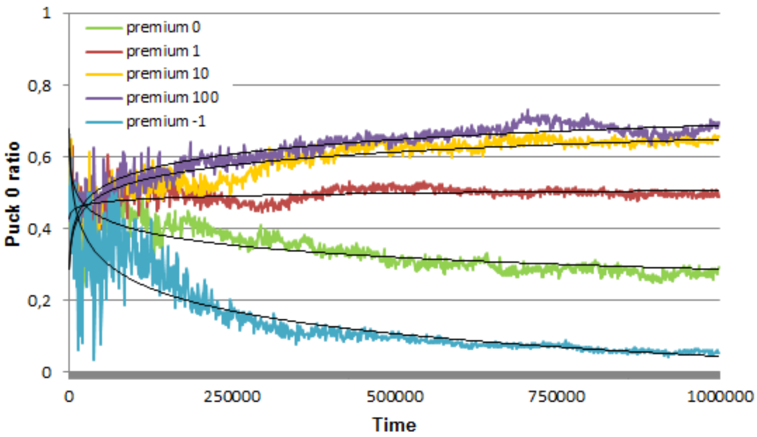

Figure 3: The median puck ratio of puck type 0 shown per premium. The black lines represent the log lines of each premium

puck type 0 ratio which ends at 65%, showing that the swarm clearly has more interest in pucks of type 0 instead of type 1. All these results are as expected and hypothesized. The negative premiums show an obvious decrease in the total puck count of type 0, whereas the premium with a value of 1 has no influence. The premium with a value of 10 generates an increase in the total puck count of puck type 0.

The premium with a value of 100 does not entirely behave as expected. Although the average values are higher than the total puck count of premium value 10, the difference is not as big as was expected. The type 0 ratio eventually ends at 70%, which is still 5% higher than the ratio premium 10 showed. One of the reasons the ratio does not reach 100% entirely might be due to the fact that, even though less, the pucks of type 1 still hold a value. These pucks will be collected along the way, even by robots that aim for a high collection percentage for puck type 0 since these pucks are there for the taking anyways. Each collected puck will increase the total puck count and thus make the robot more desirable and likely to be selected as a parent. This would mean that as long as there are two puck types on the board and both of them have a positive premium value, the pucks will keep on being collected since they only increase the total puck count a robot has possessed in his lifetime.

One should take note that at the end of the simulation, when 1,000,000 time steps have passed, the median lines have not flattened out yet. If the simulation time would be increased, these lines could keep on moving in the same directions and maybe ultimately reach either the 100% or the 0%.

Comparing the found results of this experiment to the results Haasdijk and Bredeche found in 2013, both some differences and some similarities can be found. First of all, the biggest differences are the two extreme premium values. Where a market mechanism makes sure that the negative premiums have a faster drop rate, it also leads to a lower end percentage of the positive premiums. A faster drop rate for the negative premium leads to a faster evolutional improvement within the swarm, making sure that these negative pucks will not be collected anymore. But, a lower ratio for the premium with a value of 100 implies that less premium pucks were collected and more normal pucks. This is not bad, but collecting more premium pucks would lead to a swarm that is evolutionary stronger than a swarm that does not collect these pucks.

As for the similarities, the less extreme valued premiums, being 0, 1 and 10, all have comparable results.

When analyzing the global results, one can see that applying the market mechanism will lead to a faster evolution rate while turning the market mechanism off leads to a higher task performance rate.

5 Conclusion

Based on the research question stated prior to conducting the experiments “How would the swarm react to different valued tasks without a working market mechanism?” and the corresponding hypothesis, some conclusions can be made. As predicted, the experiments showed that the swarm had a different reaction for each premium value set in the experiments. Setting a premium allows users to influence the behavior a swarm shows. The additional value of adding a market mechanism to the swarm is debatable. On the one hand, the market mechanism leads to a faster drop rate of the negative premiums, ensuring a faster evolutional improvement within the swarm. On the other hand, the total puck count without a market mechanism lies higher for a premium value of 100 than that it was with a market mechanism in place. Depending on the results one wishes to achieve, the decision has to be made whether or not the market mechanism should be applied.

Our findings are in line with the hypotheses stated after the literature study was conducted, although the fourth hypothesis regarding premium 100 turned out to have a lesser impact that expected. The different premiums all generated a certain behavioral pattern for the swarm that lies within the predicted outcomes.

The summarized hypothesis “The swarm will adjust its behavior according to the premium value assigned, making sure that it will perform the task that generates the highest points” was met. The swarm did change its behavior when another premium value was provided, making sure that the swarm evolves in a positive manner.

6 Future Research

Possibilities for future research related to the current experiment can be addressed on two levels. The first level encompasses an extension of the current research line. As mentioned in the analysis, the loglines have not flattened out yet after 1,000,000 time steps, it would be interesting to investigate what would happen to the ratios when the total number of time steps will be increased. Another interesting subject for future research is finding out exactly why premium value 10 and premium value 100 do not reach the 100% line within the 1.000.000 time steps. They should be more favorable to collect than the lower valued pucks. The most intriguing finding of the current study, the lack of difference between premium 10 and premium 100, also warrants a further scientific explanation.

The second possibility for future research is on a more abstract level. Darwin [9] defines “Survival of the Fittest” as the preservation of favorable variations, and the destruction of injurious variations. This theory is applicable to all species occupying the earth. Ever since the first CPU was designed, the Intel 4004 in 1971 [10], interest in technology started booming. Engineers kept on developing better and faster technology, which had an exponential growth rate due to the better technological grounds each development brought. This development is still going on today, and Ray Kurzweil [11] even predicts that we will reach a singularity stage in the near future. He defines singularity as an era in which technology grows so fast that the human body and mind will not be able to keep up unless we merge with nanotechnology. He explains his theory in his book “The Singularity is Near” [11]. Evaluating the evolution rate of technology, one can conclude that technology has come a long way since 1971 and will probably keep evolving for quite some time. An interesting question is whether technology can be seen as part of a species occupying this earth, or just as a creation of the human mind. How far would humankind have to go before the technology will be smart enough to evolve itself, and if the day when technology evolves itself comes, will it then become a species?

If technology would be considered as a species, it would be applicable for “Survival of the Fittest”. This would mean that when the environment changes, the adaptable robots would survive. The effect of an abrupt change in the environment on the swarm’s harmony has not been explored yet, and important questions remain: “Would they stick together”, “Would a dominant robot step up as a leader”, “Will some of the robots be left out due to weaker genes”?

Self evolving technology is not something humankind can only dream of, but it is turning into a reality in a fast pace. It would be both useful and interesting to research pending questions surrounding this topic before the knowledge is really needed.

7 Acknowledgements

I would like to thank my mentor Evert Haasdijk for allowing me to contribute to his research and for guiding me through this process. Without him and the excellent group of students to discuss my findings and problems with, this paper would not have been as it is now.

8 References

1. M. Eaton (2015). Evolutionary Humanoid Robotics. *Springer.*
2. A.E. Eiben and M. Schoenauer (2002). Evolutionary Computing. *Elsevier,* 82(1):1-6.
3. E. Haasdijk, B. Weel and A.E. Eiben (2013). Right on the MONEE: Combining Task- and Environment-Driven Evolution. *Proceedings of the 15th annual conference on Genetic and evolutionary computation,* 207-214.
4. The Business Dictionary. Retrieved from: <http://www.businessdictionary.com/definition/market-mechanisms.html> Page last visited May 5th 2015
5. E. Haasdijk and N. Bredeche (2013), Controlling Task Distribution in MONEE. *Advances in Artificial Life, (ECAL 2013).* 671-678.
6. N. Bredeche, JM. Montanier, W. Liu and A. Winfield (2012). Environment-driven Distributed Evolutionary Adaptation in a Population of Autonomous Robotic Agents. *Mathematical and Computer Modelling of Dynamical Systems*, 18 (1):101-129.
7. A. Barabási (2005). The origin of bursts and heavy tails in human dynamics. *Nature 435*, 207-211.
8. N. Bredeche, J.M. Montanier, B. Weel and E. Haasdijk (2013). Roborobo! A Fast Robot Simulator for Swarm and Collective Robotics. *CoRR*, abs/1304.2888.
9. The complete works of Darwin online, page 92. Retrieved from: <http://Darwin-online.org.uk/content/frameset?viewtype=side&itemID=F387&pageseq=121> Page last visited June 4th 2015
10. Intel’s history page. Retrieved from: <http://www.intel.nl/content/www/nl/nl/history/museum-story-of-intel-4004.html?wapkw=4004&_ga=1.191328872.357062434.1436110577> Page last visited July 2nd 2015
11. R. Kurzweil (2005). The Singularity is Near. *Barnes&Noble.*