Investigation of a Classifier System for Robot Movement Control on the Plane

Igors Nikitenko and Arkady Borisov
Decision Support Systems Group
Technical University of Riga
1 Kalkyu St., Riga LV-1658, Latvia
Phone: +371 7089 530, Fax:+371 782 00 94
E-mail:aborisov@egle.cs.rtu.lv

ABSTRACT: By using a classifier system, the authors examine a self-learning ability of a moving agent (robot) to learn to implement searching for one or several moving objects, *food*, on the plane. Labour intensity of tasks with various number of objects on the plane is compared.

KEYWORDS: classifier systems, search problem, bucket brigade algorithm

1. INTRODUCTION

A number of classifier systems attempt to use the bucket brigade algorithm to efficiently determine which classifiers are useful and which are not. In the classic Goldberg's statement (Goldberg, 1989), the strength $S_i(t+1)$ of the i-th classifier at step t+1 is calculated as follows

$$S_i(t+1)= S_i (t)- B_i (t)- T_i (t)+R(t)$$

where B_i is the bid of classifier i, $B_i = k_{bid}* S_i$, T_i is the tax $T_i= k_{tax} * S_i$, but R(t) is the reward from the environment . Riolo (1985) proposes his own method to determine the classifier's strength. It is calculated as

$$S_i(t+1)= S_i (t)- B_i (t)+P_i (t)+R(t)$$

where P_i (t) is the sum total of all payments to classifier i from those classifiers which posted similar messages at the preceding step i, $B_i=k*Si* BidRatio_i$ where k is a risk factor, $BidRatio_i$ is a number between 0 and 1, a measure of the classifier's specificity, that shows a number of classifiers with similar input message. If the classifier's message is unique, BidRatio equals 1. If all the classifiers match the input message, BidRatio is 0.

The bucket brigade algorithm is based on the classifier strength's distribution among other classifiers. The pure classifier strength is reached at a fixed point when the amount of strength it receives is equal to the amount it pays. Thus, in the long run a classifier strength at the fixed point, S_{fp}, approaches the value

$$S_{fp} = (R+P)/(k* BidRatio_i)$$

where R and P are the average *activation* amounts the classifier receives as rewards from the environment and payments from other classifiers, respectively. The aim of this paper is to examine the bucket brigade algorithm's ability to quickly train classifiers given various risk factors, k.

For those tasks in which the output classifier message is a vector (Dorigo and Schnepf, 1993), it is necessary to change a formula for the classifier's strength determination. In theory, there are only outlined those cases when the classifier's output message foresees arriving at a certain known situation. For example, in classical two- player games of Crosses and Naughts a classifier system is set as follows XXXXXXXXX :: XXXXXXXXX where the left-hand side is the environment state before a move, but the right-hand side is the state after the move (Dorigo and Schnepf, 1993). In such a case, the value of P_i (t) can easily be determined since all possible classifiers, which have the same right-hand side, can always be found. All these classifiers always unambiguously describe that one can easily move from the state described in the left-hand side to the situation described in the right-hand side. The bucket brigade algorithm classifier chains implement distribution of the strength in the direction of the problem goal.

The form of a classifier in search tasks is the following XXXXXXXX :: YYYYYYYY where the left-hand side describes the environment state, whereas the right-hand side describes a possible action. One cannot know the next environment situation after a move since it dynamically changes as the number of participants increases and every time is different. The number of environment states is practically endless. The strength distribution should be proceeded in the direction from the problem's goal. Calculations are made by formula that follows

$$S_i(t+1)= S_i (t)- B_i (t)+ P_i (t +1)-T (t)$$

where P_i (t) is the total sum of all payments to the i-th classifier from those classifiers which can be activated at the next step. P_i (t) = S_i (t+1)*k+ where k+ is the classifier payment factor.

When the classifier's action yields changes in the environment evaluation V, P_i (t) is equal to the environment reward provided the environment evaluation increases, and P_i (t) is equal to the environment tax if the environment evaluation decreases.

The main advantage is the possibility to gradually construct classifiers, after the direct application of which the goal is not achieved but only a transition to another classifier with greater strength takes place. The closer the goal is, the bigger classifier strength values will be. The latter is illustrated in Fig. 1.

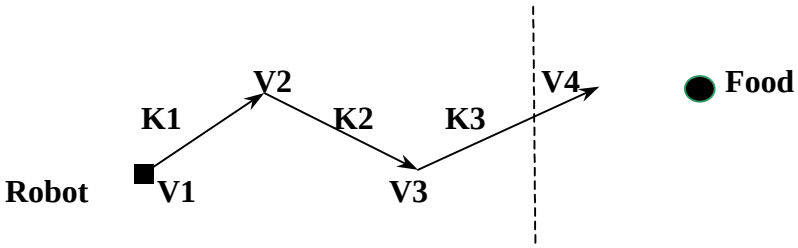

Figure 1: An example of a classifier chain.

The twitched line denotes a boarder after which a robot can meet *food*. Initially, there is no one useful classifier. During training the robot will sooner or later arrive at the environment state V3, construct classifier K3 and arrive at the environment state V4 in which it yet sees *food*. Classifier K3 receives payment from the environment, assume it equals 100, and is placed on the list of useful classifiers. S(K3)=100.

Later on, while training the robot can arrive at the environment state V2, construct classifier K2 and arrive at yet known situation V3. In this case the classifier pertains compensation P (t) from classifier K3 since it can be activated at the next step. Provided that risk coefficient k is equal to 0.1 and the tax is equal to 1, the following holds

$$S(K2)=S(K3)*k=10.$$

After that, classifier K3 is naturally chosen which yields improvement of the environment evaluation.

$$S(K1)=100-(100*0.1)+100-1=189.$$

By similarly continuing the task, the robot can arrive at the environment state V1, construct classifier K1 and come to yet known environment state V2 from which it is yet able to arrive at *food* along the chain.

$$S(K1)=S(K2)*k=1$$
$$S(K2) = 10-(10*0.1)+(189*0.1)-1=27$$
$$S(K3)=189-(189*0.1)+100-1=269$$

In the case, when state V1 is again entered and *food* is reached while moving along the chain, classifier strength values are changed as follows:

$$S(K1)=1-(1*0.1)+(27*0.1)-1=3$$
$$S(K2) = 27-(27*0.1)+(269*0.1)-1=50$$
$$S(K3)= 269-(269*0.1)+100-1=341.$$

If a task of robot movement control is considered, it is worth studying and analysing possibilities of the bucket brigade algorithm by using a simplified situation variant. Hence, the situation is being examined on the constrained plane without internal obstacles and with a certain quantity of *food* objects (see Fig. 2).

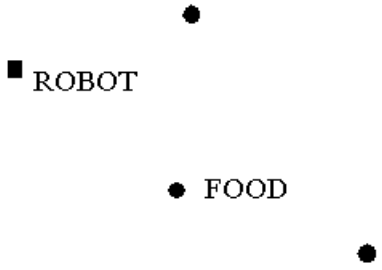

Figure 2: Representation of the environment.

Testing algorithm is executed as follows:

1. Choose initial risk factor, say, 0.1. The system operates until the robot learns to catch the food.
2. Choose 10-15 classifiers with the largest strength so that BidRatio $_i$=1, in order there would be no classifiers in this sample that describe one environment message.
3. Choose a certain risk factor value.
4. Assign the initial strength value to all the sample's classifiers.
5. Operate the system until all classifiers of the sample reach the maximum value.
6. Choose another value of factor k and go to step 4 to repeat the procedure.

The algorithm is executed until there are obtained results for all chosen k $_{bid}$. All the results are summarised in one aggregated graph which represents the number of necessary cycles for all classifiers strength to increase up to 90% starting from the fixed point for all k.

The results obtained enable one to access the most efficient k for a quicker system's training. While executing the task, genetic algorithms have been employed to generate new classifiers in situations when no one of the existing classifiers is able to improve the environment evaluation. In this case a certain number of classifiers with the greatest evaluation is chosen and a genetic operation of crossover is performed with them that yields generation of new classifiers.

2. EXPERIMENTAL PART

The main aim of the practical part was to examine the learning ability of the system by using the modified bucket brigade algorithm. In order to objectively evaluate the results, the experiments have been performed in a simple, average complicated and complicated environment. A computer program to implement the algorithm was written in Delphi. The following initial parameters were chosen:

- number of moving food objects=3;
- environment's reward = 200;
- environment's tax = 60,
- non-chosen classifiers tax = 2,
- Kbid=0.1,
- K+=0.1.

The task consists of three parts:

(1) Training with three objects followed by searching for the optimum Kbid value (simple environment).
(2) Implementation of the process of training with six food objects (average complicated environment).
(3) Training provided nine food objects (complicated environment).

Initially, the execution cycle is repeatedly run to train the classifiers until the number of successful moves reaches 95% 55 times in succession. Within each cycle's iteration there are examined 100 moves. The results are given in Fig.3. In 220 iterations there were achieved very good results. The robot can always choose a correct action with a very high probability.

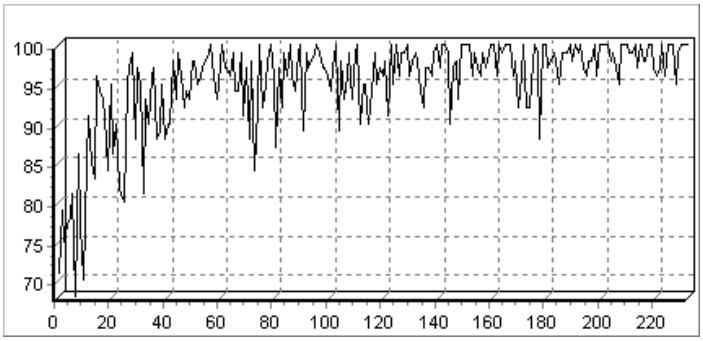

Figure 3: Representation of the process of training given three various food objects. The number of classifiers is 1342.

To perform the next part of the task, 15 classifiers with good strength values are chosen for further examination and the rate of their strength increase is tested under the following five Kbid values: 0.1, 0.3, 0.5, 0.7 and 0.9. Testing is finished after 500 iterations or when 90% of the common strength value is reached. The results obtained are shown in Fig. 4.

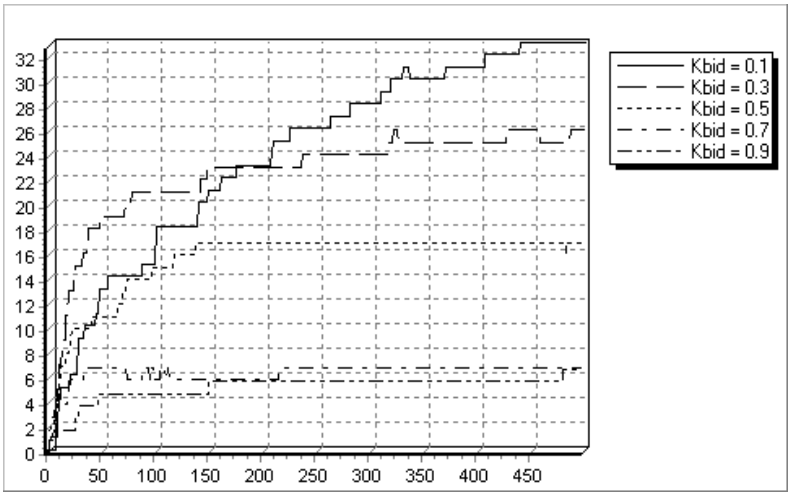

Figure 4:The rate of training under various Kbid values.

The next experiments have been executed in the same way with only difference that the number of food objects was first increased to 6 and then up to 9. The results are shown in Fig. 5 and Fig.6.

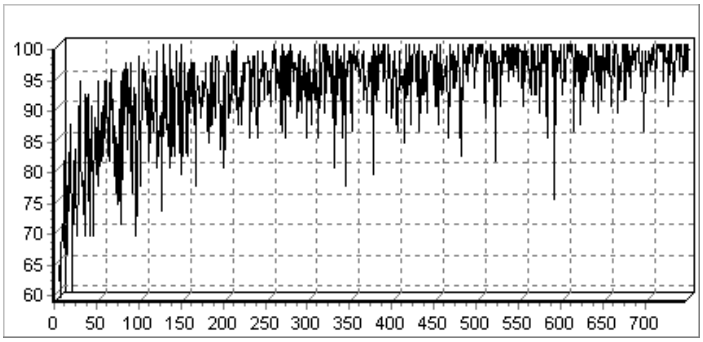

Figure 5: Training given six food objects. The number of classifiers is 6214.

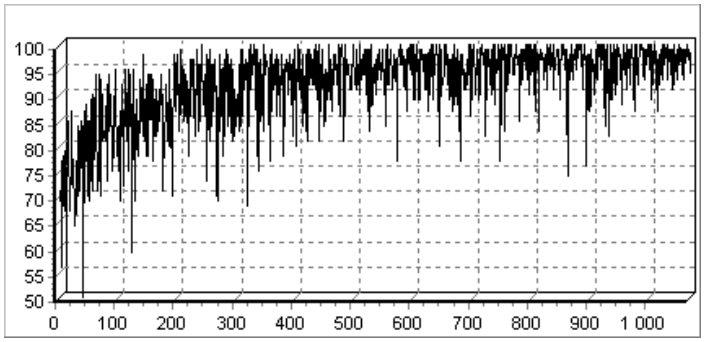

Figure 6: Training given nine food objects. The number of classifiers is 10427.

3. CONCLUSIONS

On examining the results, one can conclude the following:
(1) The rate of training depends on the Kbid value. The less it is, the quicker training will be.
(2) The number of iterations is large, however, the aggregated value of the classifier strength is far from 100%. This indicates that when the task is repeated, there exist alternative good classifiers with the help of which the goal can be achieved.

By considering different environment training options, one can easily be convinced of the training time dependence on the number of objects in the environment.

REFERENCES

Dorigo, Marko ; Schnepf, Uwe, 1993, "Genetics- Based Machine Learning and Behavior-Based Robotics: A New Synthesis" , IEEE Transactions on Systems, Man and Cybernnetics. Vol..23, No1, January/February, pp.141-153.

Goldberg, D.E.,1989, " Genetic Algorithms in Search, Optimization and Machine Learning". New York:Addison-Wesley.

Riolo, Rick L., 1985, "Bucket Brigade Performance: I. Long Sequences of Classifiers", Proc. of the Intern.Confer. on Genetic Algorithms and their Applications, John J.Grefenstette (Ed.) pp.184-195.