Project 1.B:
Evolving Colonel Blotto Strategies using a Genetic Algorithm)

Mikael Brevik

February 15, 2012

Contents

Chapter 1

Introduction

1.1 About the project

The purpose of this project is to gain experience with genetic algorithms and coevolution.

The EA framework developed in previous project (Part A), will be used to solve a problem in coevolution. The problem that will be solved is a war game, Colonel Blotto, which includes a set of individuals, Colonels, compeating with each other.

1.2 The Colonel Blotto Problem

The problem is for finding the best of resource (soldiers S) distribution across a number of battles, B. There are a fixed number of battles, and the winner of most battles in a war wins the war. A Colonels distributes all resources before a war.

As a added twist, there can be reployment of left over resources in a battle. The remaining resources gets divided evenly across the remaining battles. The reployment is moderated by a fraction R_f.

A strength reduction is also added to the problem. A strength reduction will reduce resources in each battle, if a previous battle is lost. This strength reduction will aggregate through out the battles. Each war begins with strength 1.0 but is decreased by a fraction L_f each time a battle is lost.

Chapter 2

Deliverables

2.1 Description of representation

Since the genome should have B binary genes with a random integer between 0 and 10, we know that the total size of the binary string should be $B * 4$. This is due to the span of the random values all can be represented in 4-bit.

For representing a phenotype we need to find out the original 10-based value for the gene and then normalize it. So in the $toPhenotype()$ method will loop through the genotype string with an increment of 4 and convert each gene to a denary value. After the weight is found again we can normalize these values by dividing all by the sum of the weights. If the sum of the weights equals zero (I.e no forces in battle), the resources get automatically distributed to all battles.

Example: If we have a two battle war, and the random integers selected are 7 and 4, we convert those to binary values: 0111 and 0100. These binary values get concatenated to 01110100. This is our genotype.

To convert this to a phenotype we need to convert from binary to denary again and yet again we have 7 and 4. Now we normalize these values. 7/11 and 4/11: 0.6364 and 0.3636. The vector of these values ([0.6364, 0.3636]) is our phenotype.

2.2 Significance of strategy entropy

The strategy entropy is defined by the following formula:

$$H(s) = -\sum_{i=1}^{B} p_i \log_2(p_i)$$

In the formula, p_i is the fraction of the total resource that is devoted to the ith battle. This is the same as the ith index of the phenotype.

Let's analyze the formula. If the normalized weight is very small or very high, the entropy will get close to 0. This is due to the $\log_2()$. So this means that the strategy isn't a generic "trying to cover it all" strategy, but rather favors certain battles. This sounds more like a strategy.

Let's say that the game uses both the troop redistribution and strength reduction. With those two aspects, the smartest move is to start hard in the first battle and therefor get an advantage in the following battles with redistribution of the troops and full strength. The entropy says that when it is close to zero, there are big differences in the p_i therefor more likely to be a good strategy.

2.3 Settings of key parameters

Population size	20
Mutation probability	0.04
Mutation function	Change random weight
Birth probability (Crossover)	0.35
Crossover type	Uniform
Generations	200
Adult selection	FullReplacement
Parent selection	SigmaScaling
Elitism	2
Truncation	0

Table 2.1: The default values of the key EA parameters

The result of running the application with the defaults value can be viewed at Figure 2.1 and Figure 2.2

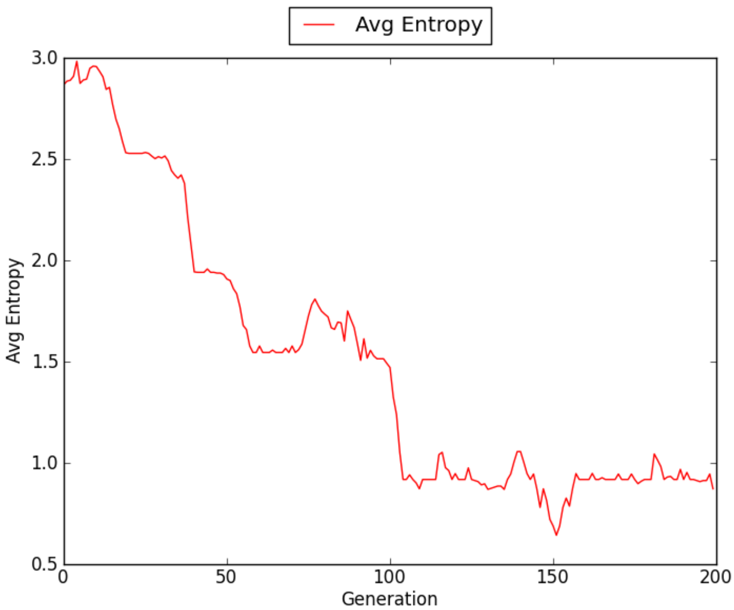

Figure 2.1: Showing average entropy for run with default EA parameter values as given by Table 2.1

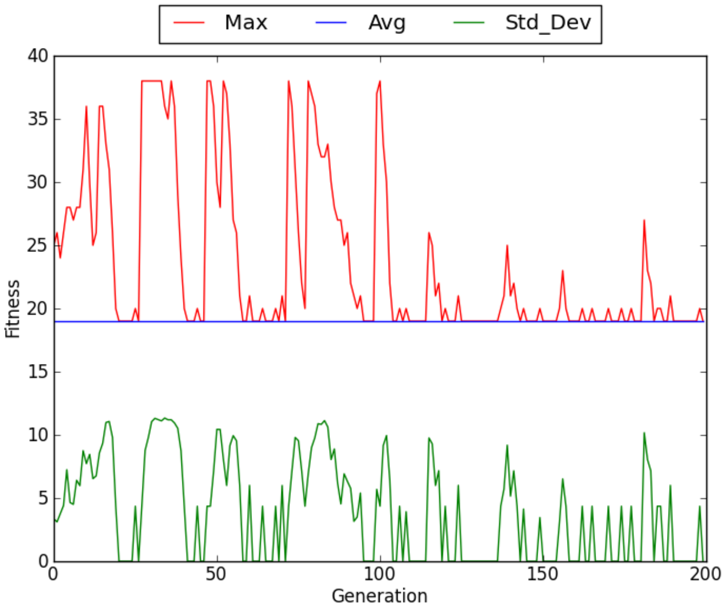

Figure 2.2: Showing fitness for run with default EA parameter values as given by Table 2.1

2.4 Summarizing base runs

Run	B	R_f	L_f	Description
1	5	0.0	0.0	Convergence to one stable strategy [0.25, 0.75, 0.0, 0.0, 0.0]. Best 90 times of the last 91 generations.
2	5	0.3	0.0	Convergence to one stable strategy [0.0, 0.1538, 0.0, 0.0, 0.8461]. Best the 17 last generations.
3	5	1.0	0.0	Continuous shifting between strategies where the second first and two last battles looks most important.
4	5	0.0	0.4	One strategy won the last 10 wars. Focus on the first battles.
5	5	0.0	1.0	Strong convergence to the strategy [1.0, 0.0, 0.0, 0.0, 0.0]. Won the last 140 generations.
6	5	0.3	0.4	Convergence to [0.625, 0.125, 0.0, 0.0, 0.25] (with small alterations).
7	5	1.0	0.4	Converges to [0.8461, 0.0769, 0.0, 0.0769, 0.0]. Heavy on the first p_i.
8	5	1.0	1.0	Strong convergence to the strategy [1.0, 0.0, 0.0, 0.0, 0.0]. Best the last 150 generations.
9	10	0.0	0.0	Convergence to one stable strategy [0.0, 0.0, 0.5, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0]. Best the last 20 times.
10	10	0.3	0.0	Strong convergence to a strategy with the last element has all troops.
11	10	1.0	0.0	Convergence to a strategy with it main focus on the last battle. Synonym strategies was best the 30 last time.
12	10	0.0	0.4	Convergence to a strategy with it main focus on the first battle. Strategy best the last 50 times
13	10	0.0	1.0	Convergence to the strategy with 1.0 distribution in the first battle.
14	10	0.3	0.4	Convergence to a strategy with .4 as first battle and .27 as the next. Other battles have small troop allocation. Fairly high entropy.
15	10	1.0	0.4	Convergence to [0.55, 0.1, 0.0, 0.2, 0.05, 0.0, 0.05, 0.0, 0.05, 0.0]. Best the last 40 times
16	10	1.0	1.0	Convergence to a strategy focusing on the first battle. Fairly low average entropy.
17	20	0.0	0.0	Convergence to one strategy. Fairly high entropy; pretty evenly distributed troops.
18	20	0.3	0.0	Continuous shifting between two strategies. Similar strategies but do not appear to be equivalent.
19	20	1.0	0.0	Convergence to a strategy with it main focus on the last battle but also smaller values regularly in earlier battles.
20	20	0.0	0.4	Convergence to a strategy with it main focus on the first battle, but also smaller values throughout the battles.
21	20	0.0	1.0	Convergence to the strategy with focus on the first battle. About .5 in the first battle, and about .04 on battle 4, 6, 8, 11, 13, 14.
22	20	0.3	1.0	Battle between two strong strategies. One with .45 in the first battle and the other with .35. High entropy.
23	20	1.0	0.4	Convergence to a strategy with evenly distributed troops. Entropy in the >3 interval.
24	20	1.0	1.0	Shifting between two strategies. Both high focus on the first 4 battles.

Table 2.2: Summarization of results of base runs

2.5 Plotted data

I have chosen to look more in depth at run 7, 13 and 22.

2.5.1 Fitness and entropy run 7

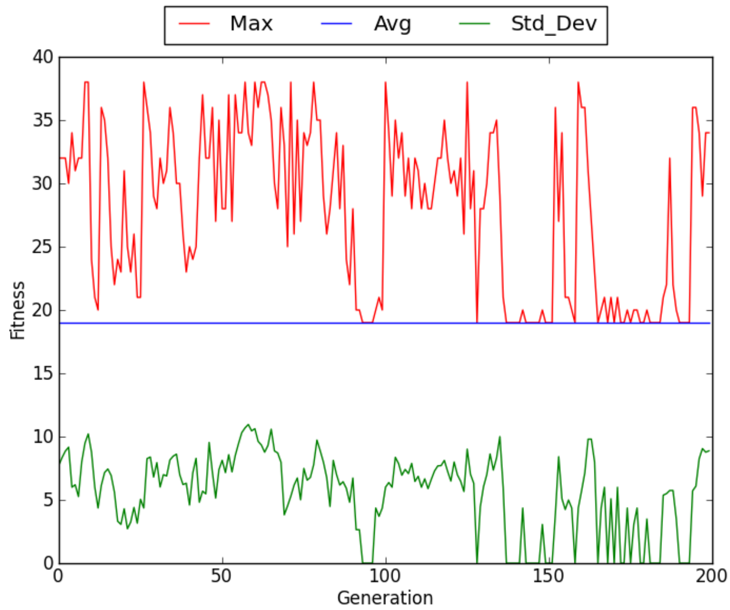

Figure 2.3: Fitness plot with B = 5, R_f = 1.0 and L_f = 0.4

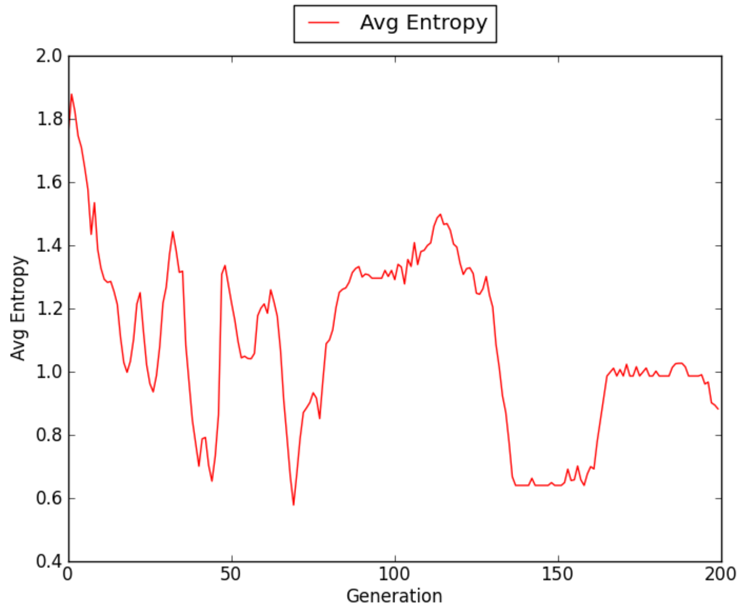

Figure 2.4: Average entropy plot with B = 5, R_f = 1.0 and L_f = 0.4

It is clear that with $R_f = 1.0$ and $L_f = 0.4$, the strategy is to focus the forces in one battle. As the avg entropy in **??** starts high, and gets reduced as generations go. There was a climb in the entropy in the final 50 generations with this run. There was some competition around the generations 150-160. It might be that due to the R_f, after winning multiple battles it tried to move some of the "redundant" forces from the first battle to win more battles.

2.5.2 Fitness and entropy run 13

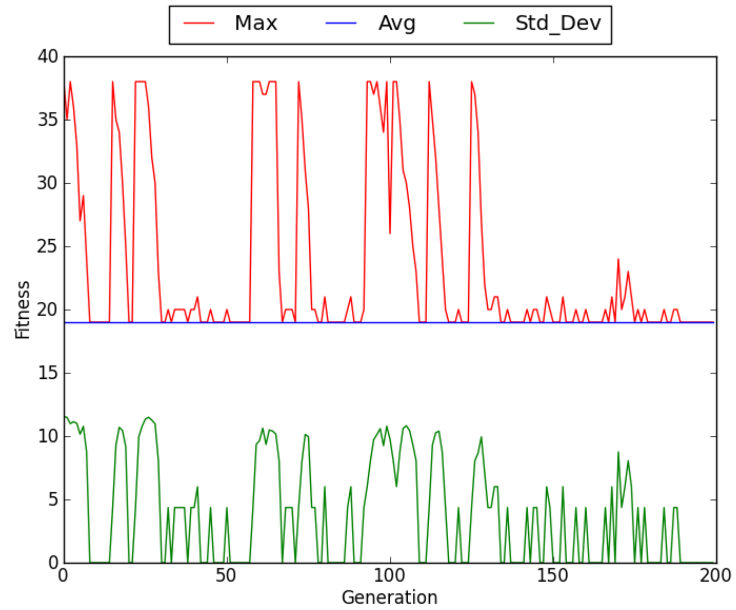

Figure 2.5: Fitness plot with B = 10, R_f = 0.0 and L_f = 1.0

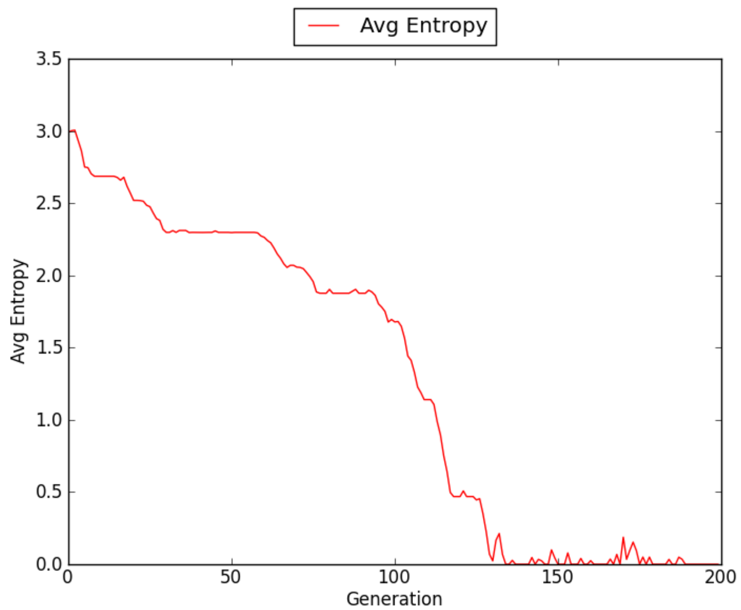

Figure 2.6: Average entropy plot with B = 10, R_f = 0.0 and L_f = 1.0

In this example we can clearly see the effects of high L_f. The L_f will make it so that if you lose the first battle, you will take a massive hit for the next battles. So the best strategy was to move the forces to the first battle. As we can see from the avg entropy and fitness plot, all the Colonels had to do the same. In the final generations the highest and average fitness were equivelent.

2.5.3 Fitness and entropy run 22

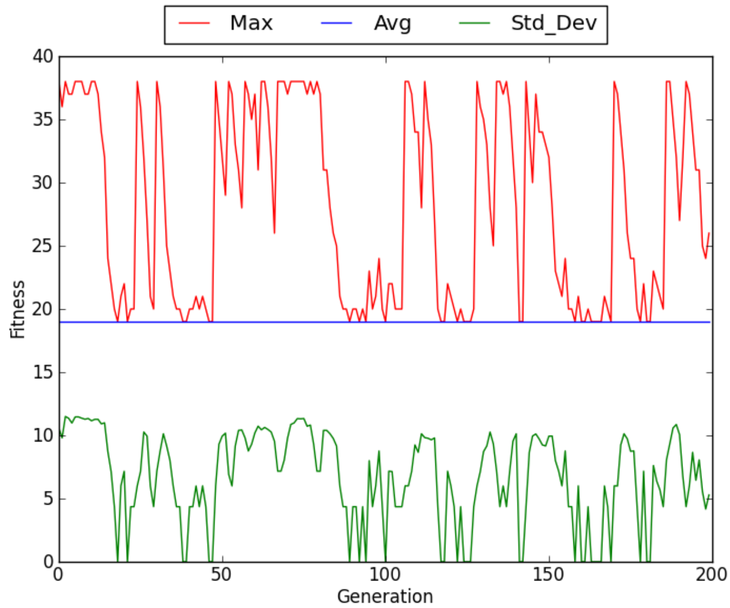

Figure 2.7: Fitness plot with B = 20, R_f = 0.3 and L_f = 1.0

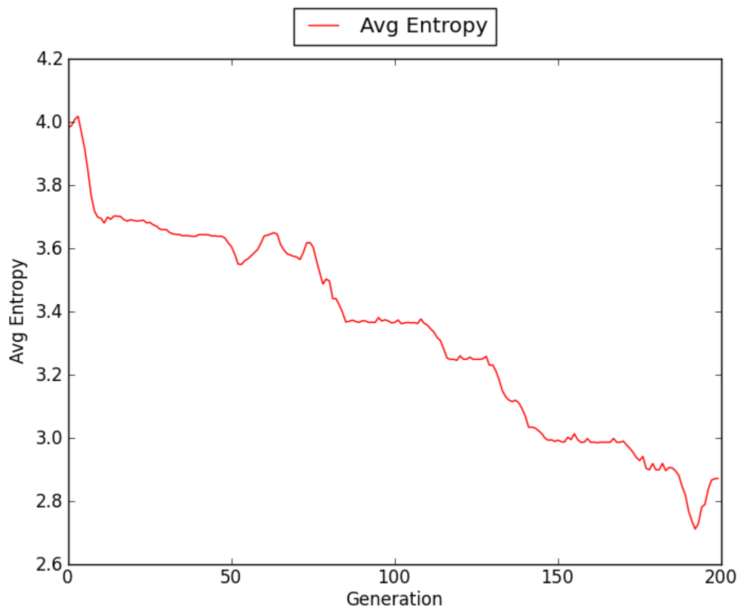

Figure 2.8: Average entropy plot with B = 20, R_f = 0.3 and L_f = 1.0

There was a competition between two strong strategies. With the settings of R_f and L_f, the was to have the main focus on the first battle, but also distribute some of the forces out to the remanding battles.

One strategy has 10% more of the forces in the first battle. This had it's advantages, since the last 5-6 generations, this strategy was the best. As the weaker strategy lost against the dominant in the first round, he took major hits from the strength reduction.

2.6 Coevolution

I've seen, that in a co-evolution, one can't just look at the fitness to assess the strength of the solution. The max fitness value varied radically from generation to generation. In other implementations, non co-evolution, one could se a gradually increase in strength as the generations passed. Typically in a staircase shape. This is not the case for competitive co-evolution.

Instead of looking at the fitness one could other measures to find the strength of a solution. In this case I used the entropy function $H(s)$. This value helped indicate if the strategy was a smart one or not.

List of Figures