CHALMERS
UNIVERSITY OF TECHNOLOGY

Evolutionarily Emergent Foraging Strategies for Active Agents

Examining Levy Flights, Locally Optimal Strategies, and Genetically Generated Strategies for Foraging in a Changing Environment

Master's thesis in Complex Adaptive Systems

EMIL JANSSON

DEPARTMENT OF PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

Master's thesis 2022

Evolutionarily Emergent Foraging Strategies for Active Agents

Examining Levy Flights, Locally Optimal Strategies, and Genetically Generated Strategies for Foraging in a Changing Environment

Emil Jansson

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2022

Evolutionarily Emergent Foraging Strategies for Active Agents
Examining Levy Flights, Locally Optimal Strategies, and Genetically Generated Strategies for Foraging in a Changing Environment
EMIL JANSSON
Department of Physics
Chalmers University of Technology
Supervisor: Giovanni Volpe, Department of Physics, Gothenburg University
Examiner: Giovanni Volpe, Department of Physics, Gothenburg University

Abstract

Microbes, insects, birds, and mammals. Many forms of life depend on the search for food to survive. One search strategy that has been observed in nature is a levy flight, where an animal moves from area to area in long stretches to then explore the local environment. Levy flights can be described as statistical mathematical phenomena where the steps lengths of the agent's movement follow a heavy tailed distribution. Earlier studies have shown that in certain environments, a middle ground between ballistic Levy flights and Brownian motion is more efficient than the outlier strategies. This thesis expands on those results by investigating which strategies perform best in an environment where local conditions change as one moves through space. We find that using a strategy that adapts to local conditions is not necessarily efficient if it does not consider the changing nature of the environment. We also let a neural network evolve using a genetic algorithm and let it optimize the movement of an agent which leads to efficient searches.

Keywords: search strategies, active agents, changing environment, genetic algorithm.

Acknowledgements

Many thanks to Giovanni Volpe who have supervised my thesis. His experience in designing experiments and understanding of the wider scientific field surrounding my work have been very valuable.
I also thank all the kind people at the Soft Matter Lab at Gothenburg University. They have all been very welcoming.

Emil Jansson, Gothenburg, June 2022

Contents

List of Figures

List of Tables

1

Introduction

To explore and to find is a basic behaviour in most living things. Finding nutrients might be the most universal need of living organisms and something that their survival depends on. From the scale of single-cell bacteria to insects, birds, large mammals, and even humans, we all explore and depend on finding resources for our survival.

The strategies used and the patterns that appear when foraging have been widely studied in the field of foraging theory [1, 2]. One phenomenon that has often been observed in natural movement patterns is a tendency to move ballistically for a time, turn, and then move ballistically again. The distribution of the length of these ballistic motions often takes the form of a power law with a decreasing likelihood of longer movement. [3] A popular way to model and generate this type of movement is a Levy flight, where the duration of the ballistic motion is sampled from a heavy-tail distribution [4]. Levy flights are emergent in a multitude of systems, and understanding this movement pattern better can help us both to understand natural systems [5] and to create artificial systems that use the Levy movement pattern [6, 7, 8].

The theory of emergent Levy flights in animal movement patterns is a controversial subject, with many articles observing the phenomenon [9, 10, 11, 12] and others arguing against it being scientifically important or even real [13, 14, 15, 16]. One hypothesis from the proponents of Levy flights is the Levy flight foraging hypothesis. "The Lévy flight foraging hypothesis states that since Lévy flights optimize random searches, biological organisms must have therefore evolved to exploit Lévy flights" [17].

In 2017, Volpe and Volpe showed that the optimal distribution for high search efficiency with a Levy flight varies depending on the topography of the environment being explored [18]. This thesis will replicate these results and expand on them by examining how agents moving in a Levy flight pattern perform when the characteristics of the environment change between different regions. It will also test different strategies for efficient exploration in such changing environments. Finally, it will allow a genetic algorithm to optimise the movement of agents and analyse the distributions that emerge.

2

Theory

This chapter covers the theory required to understand the method, results, and reasoning of this thesis.

2.1 Anomalous diffusion

Diffusion is the general movement of a number of objects towards an area of lower concentration [19]. The most common model of a particle that shows diffusive behaviour is Einstein's model of Brownian Motion [20] where the mean square displacement (MSD) can be described by the equation:

$$\langle r^2 \rangle = 2dD\tau,$$

where r is the displacement from the source, d is the number of dimensions, D is the diffusion constant, and τ is dimensionless time.

Anomalous diffusion processes are those statistical movement patterns that produce MSDs that do not have a linear relation to time. Instead, anomolous diffusion is described by the equation:

$$\langle r^2 \rangle = K_\alpha \tau^\alpha,$$

where K_α is a generalised diffusion coefficient [21]. This relation is a power law, and depending on the value of α we can get superdiffusion, normal diffusion, or subdiffusion, as seen in figure 2.1.

An example of a superdiffusive process is a Cauchy flight, a continuous-time random walk, where the step lengths are drawn from a Cauchy distribution.

2.2 Stable Distribution

A stable distribution or a Lévy alpha-stable distribution is a category of distributions that include normal distributions and Cauchy distributions.

There are four parameters that define the characteristics of the stable distribution:

1. μ, the location of the distribution.

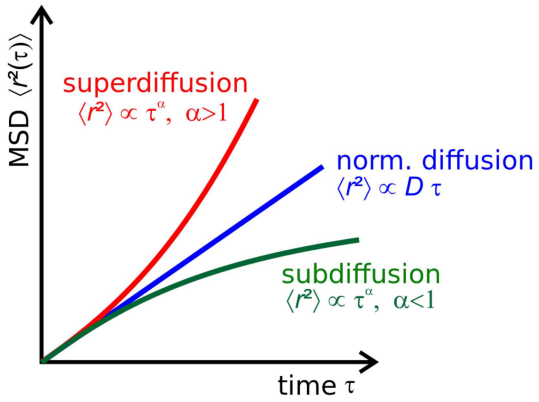


Figure 2.1: The mean square deviation of superdiffusive, normal diffusive, and subdiffusive processes. Img. src: [22]

2. c, the scaling parameter.

3. α, a parameter that controls the concentration of the distribution.

4. β, a parameter that controls the symmetry of the distribution.

For the purpose of this thesis, we always consider a stable distribution with $\mu = 0$, $c = 1$, and $\beta = 0$. This leaves open the possibility of different values of α and some examples of this can be seen in figure 2.2a. We are going to consider stable distributions with α in the range $[1, 2]$.

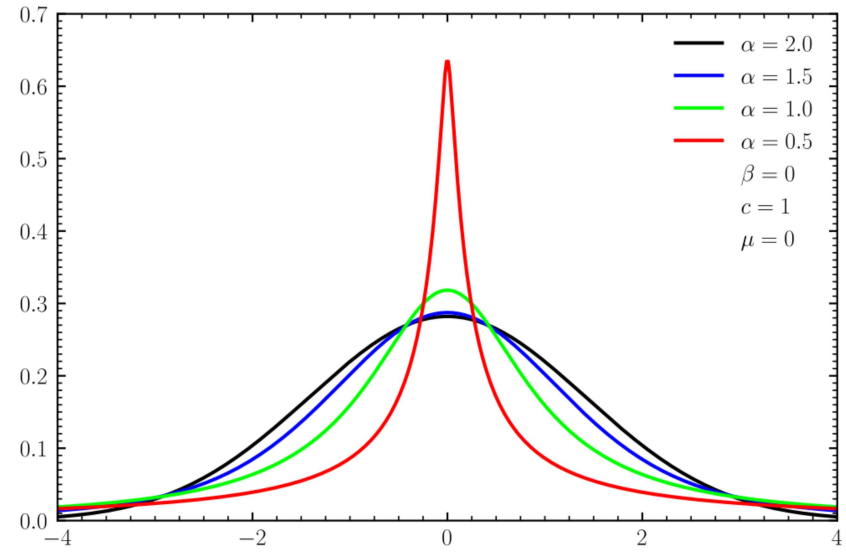
When α is 2, the stable distribution will take the form of the normal distribution. When alpha is 1, the stable distribution takes the form of the Cauchy distribution, also called the Lorentz distribution.

All stable distributions except the normal distribution are heavy-tailed. Having a heavy tail means that the distribution has a relatively high probability of producing outlier values. In the most common definition of heavy-tailed distributions, they have to be unbounded by the exponential distribution. The closer α gets to 1 the heavier the tail of the distribution is. When α reaches 1 the probability spreads to the degree that the expected value of the distribution is undefined, this is the Cauchy distribution.

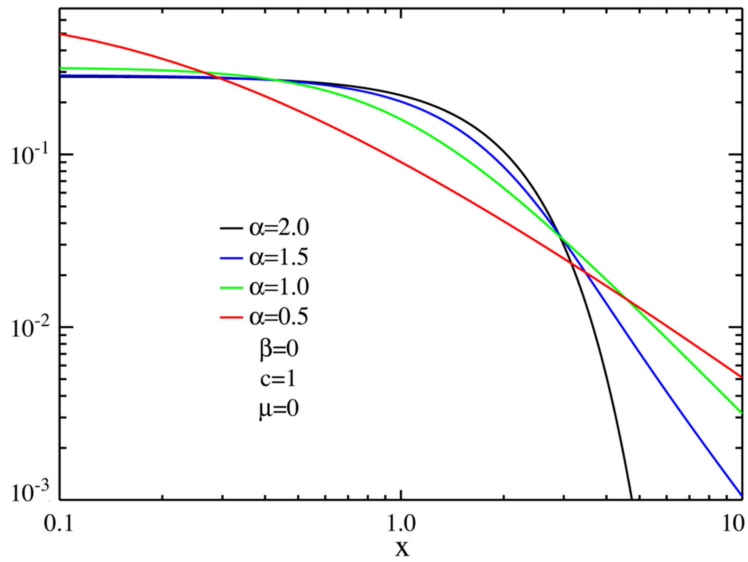
The difference in the tails of the distribution at different α can be more clearly seen in figure 2.2b. Also worth noting from this figure is that stable distributions for

2.3 Levy Flight

There are varying definitions for how to define a Levy walk, but in this thesis we are going to define it as follows: A Levy flight is a random walk whose step lengths are distributed according to the absolute value of a stable distribution [25].



(a)



(b)

Figure 2.2: The probability density function of the stable distribution for different alphas. Plotted in linear (a) and logarithmic (b) space. Img. src: [23, 24]

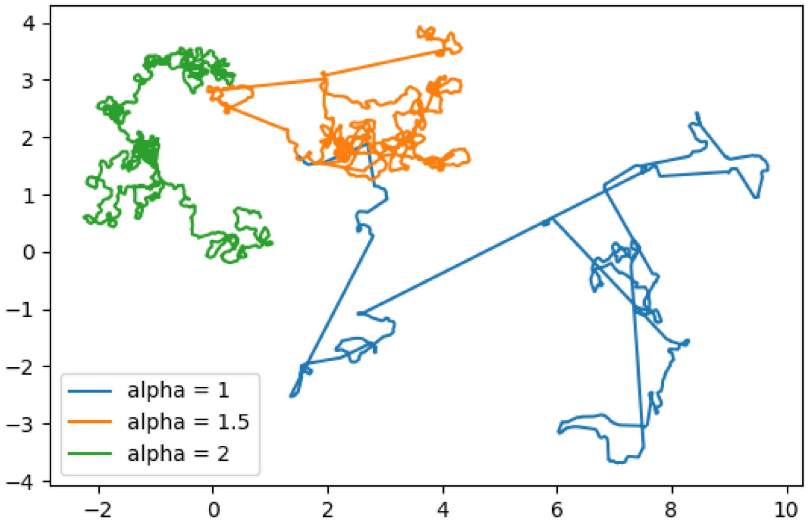


Figure 2.3: An example of Levy flights in 2 dimensions.

Levy flights are a normal diffusive process in the range $\alpha \in (1, 2]$ but when $\alpha = 1$, the Cauchy distribution, the process is superdiffusive. As is exemplified in figure 2.3, when one changes the alpha of the stable distribution, the movement trajectory goes from more ballistic movement at lower alphas to more Brownian movement at higher alphas.

2.4 Genetic Algorithm

A genetic algorithm is an optimization algorithm that is inspired by biological evolution. It contains analogies for many biological concepts such as chromosomes, genes, mutation, reproduction and survival. The genetic algorithm described in this thesis is based on the description of a genetic algorithm in Wadhe's book *Biologically Inspired Optimization Methods* [26].

A basic genetic algorithm follows these steps:

```
population = initialize()

for generation in range(num_generations):

    fitness = evaluate(population)

    parents = select(population, fitness)

    children = cross(parents)
```

```
mutated_children = mutate(children)

population = mutated_children
```

Each step will be explained in the following sections.

2.4.1 Population

The population in a genetic algorithm is made up of individuals represented as chromosomes. These chromosomes consist of arrays of data. Each entry in these arrays is referred to as a gene, and it can take many forms depending on the gene encoding of the algorithm. Common encodings are binary values and floating point numbers.

2.4.2 Evaluation

How the fitness of a chromosome is evaluated is problem-specific. The evaluation function should produce a fitness value for each chromosome that is high if the decoded chromosome fits the optimisation criteria.

An example of this is a genetic algorithm that optimises the speed of cars. Let us say that the blueprint of the car is encoded in the chromosome. The evaluation function might then be to build the car and measure its speed. The fitness returned from the evaluation would be the speed, as this is what is optimised for.

2.4.3 Selection

The selection process of a genetic algorithm is similar to the natural selection in nature. It can be any process that randomly selects individuals, giving a higher probability of being selected if the individual has higher fitness.

A common selection process is tournament selection, which randomly chooses n_{tour} tournament participants from the population of n_{pop} and let the participants compete against each other by random chance where the more fit individual has a chance $p_{tour} >= 0.5$ to win the encounter. The least fit agents start to compete and then the winner faces the one who is one step higher in the ranking until a champion has been chosen out of the n$_t$*our* contestants. This process is then repeated until n_{pop} individuals have been selected. It is possible to select the same individual multiple times.

2.4.4 Crossover

The biological analogy for crossover would be mating. Here the individuals who have been selected are paired up, or grouped up in cases of $n_{cross} > 2$, and their genetic material is interchanged with each other. This exchange can be done in many different ways. Some ways always preserve the length of the chromosome while others do not.

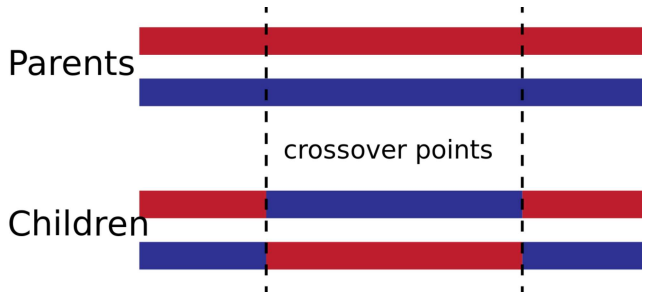


Figure 2.4: An illustration of 2-point crossover. Img. src. [27]

One of the most common ways to combine two chromosomes is n-point crossover. This means that n_{point} indices are selected along the length of the chromosome. The areas between every second of these crossover points are then interchanged with the other individual. An example of a 2-point crossover can be seen in figure 2.4.

2.4.5 Mutation

After a new generation has been formed from the selected parents using crossover some random mutations are introduced. This means that random genes in the individual are changed to create the possibility of new traits appearing.

If the genome is encoded in binary a mutation could be a bit flip from a 0 to a 1 or vice versa. In case the genome is encoded as a floating point number the mutation can be a random change in this number.

After mutation we have a new population ready to be evaluated and the process starts again. As there is always a selective pressure for more fit individuals the population tends to move towards increasing fitness over time.

3

Simulations

This chapter contains a description of the models and simulations that were examined in this thesis, as well as the results from those simulations. First, the element of the model that was shared between all simulations is presented. After that, the specific simulations are described. The process of producing these results has been iterative, and therefore the simulations will be presented one after another in chronological order.

3.1 Model

This thesis is based on agent-based simulations, where agents' movement is tracked as they move in 2-dimensional space.

The agents are point-shaped, with constant speed $v = 0.01$. Three variables are tracked for the agent. x-position, y-position, and the angle at which the agent is currently facing, θ.

The movement of the agent is based on a stable Levy-distribution with $\mu = 0$, $c = 0$, $\beta = 0$, and α varying between 1 and 2. A value e is sampled from the levy distribution and this value is referred to as the agent's effort. Every timestep, of length $\delta t = 0.5$, the effort decreases until it reaches zero. When e reaches zero, the agent's direction θ is modified by adding a random variable drawn from a normal distribution with standard deviation $\sigma = \frac{\pi}{6}$. A new effort-value is then drawn from the Levy-distribution and the algorithm starts again. If the effort value decreases linearly with time, this creates a levy flight.

Changing α changes the characteristics of the agent's movement. At $\alpha = 1$ the agent moves in a superdiffusive levy flight and at $\alpha = 2$ the movement is Brownian [18]. All values in between gradually changes between these extremes.

The starting position of the agent is a random position selected from a uniform distribution over $x \wedge y \in [0, 2U)$. The starting rotation is selected uniformly from $\theta \in [0, \pi)$.

3.1.1 Resistance

All space within the environment has a modifier for how quickly an agent's effort is depleted. I call this value resistance, r.

The effort is randomly drawn from a levy distribution and then decreases every timestep according to this formula:

$$e_{k+1} = e_k - 2^r \delta t$$

where e_k is the agent effort at timestep k and δt is the length of a timestep.

This means that in areas of high resistance, the same effort takes an agent fewer steps and a shorter distance than in a low resistance area. In the limit of low resistance, the agent would move in a single straight path, never turning. In the limit of high resistance, the agent would turn as often as it gets the chance, which in a simulation would be every timestep.

3.1.1.1 Why Resistance?

A very natural way to vary the environment would be to vary the length scale of the obstacles within it. What resistance does instead is that it varies the length scale of the movement of the agents, as with lower resistance you move longer with the same effort and with higher resistance you move shorter distances with a given effort.

Varying the length scale of the movement instead of the environment has some advantages in that it allows the obstacles to stay regular, which speeds up simulation.

Changing the speed of the agent in different regions of the environment would achieve similar results, but would weight the results to the higher time spent in regions of low speed, as they would take more time to get out of. It would also create a problem with the step size of an agent changing in different regions, and this could change the interaction agents have with obstacles.

We therefore settled on using this model of effort and resistance.

3.1.2 Area Units

The world through which the agent moves is divided into a grid, with spacing $u = \frac{1}{20}$. Each square in this grid is referred to as an area unit, and agent performance is measured by considering how many area units can be found over a period of time. More area units found is equated to higher performance.

Finding an area unit consists of the agent stepping inside the square, marking it as discovered. If an agent revisits this area unit at a later time step, the area is already considered to be discovered and does not count as a new discovery. Thus agents are rewarded for moving in new areas where they have never been before.

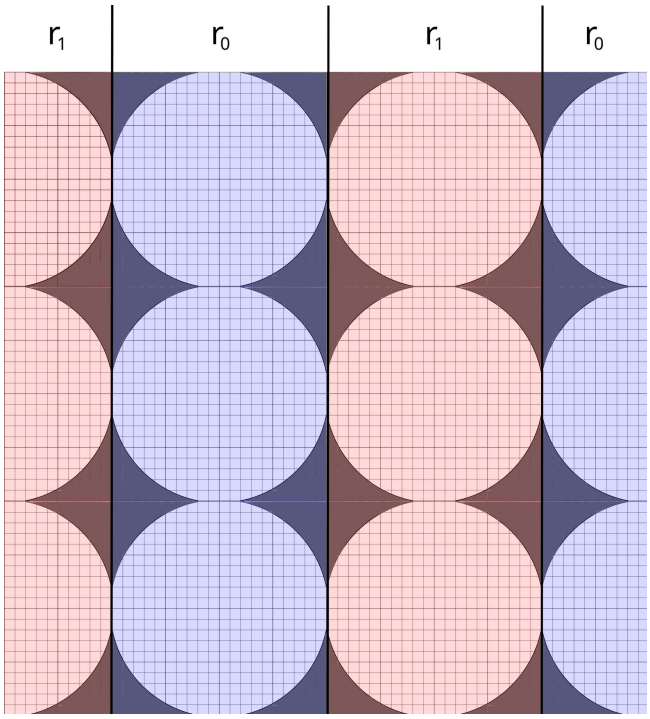


Figure 3.1: Example of an environment made up of cells with obstacles tiling the plane. Even columns are experiencing $r = r_0$ while odd columns are experiencing $r = r_1$.

3.1.3 Cells and Obstacles

To be able to insert obstacles in an infinite environment, we introduce the concept of cells. A cell is a square region in the plane that can be tiled to build the plane. We can insert an obstacle into a cell and it will be repeated in the environment, as seen in figure 3.1. Each cell has side length $U = 1$.

The common simulation parameters include two variables r_0 and r_1. They represent resistance in even and odd indexed columns of cells in the environment as seen in figure 3.1. In the current simulation setup we can therefore have two different resistances in a single environment at the same time.

3.2 Initial Simulations

The initial simulations are done to verify the results found in [18] which will act as a starting point for further exploration.

To test the performance of the agents, a simulation of 10^5 timesteps is run. The performance of the agent is given by the number of area units discovered in that time. The alpha of the agent varies between 1.0 and 2.0 in 0.1 intervals, resulting in 11 different values of α. For each value, the simulation is run 1000 times, and a mean of discovered cells is calculated. The resistance in the environment is set to 0

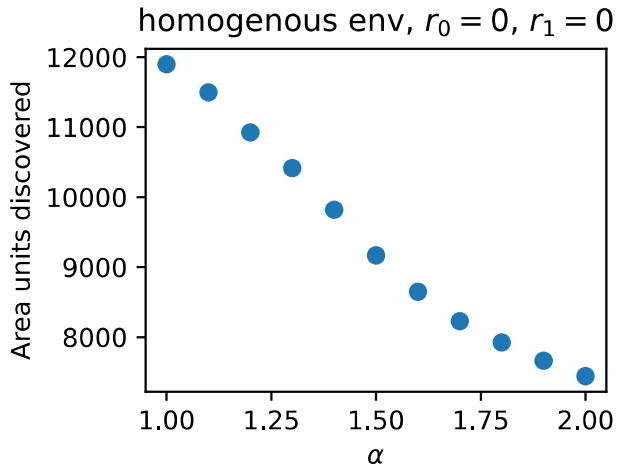


Figure 3.2: Performance of agents with varying alpha in homogeneous environment.

which means $(r_0, r_1) = (0, 0)$.

3.2.1 Homogeneous Environment

The first simulation takes place in a homogeneous infinite environment. The results can be seen in figure 3.2. We can also see some example trajectories in figure 3.3.

Here we can see an increased efficiency as alpha gets lower, meaning approaching superdiffusivity, with longer movement before turning. This is in line with what can be seen in previous studies [18][28].

3.2.2 Convex Environment

The second simulation features obstacles in the environment. The obstacles take the shape of circles with diameter 0.8 that the agent cannot enter; see figure 3.4a. If the agent tries to enter a circle, it is pushed out as shown in figure 3.5a. This leads to a behaviour that is very similar to the agent sliding with its component velocity along the circle circumference.

In figure 3.4b we can see a very similar trend to the homogeneous case, with lower alphas leading to more discoveries. This can be understood by considering that an agent travelling along a straight path will never get stuck in a single place given these obstacles and boundary conditions, only be pushed to the side. One can get an idea of a typical trajectory by looking at figure 3.4c. These results are consistent with what has been observed in previous studies [18].

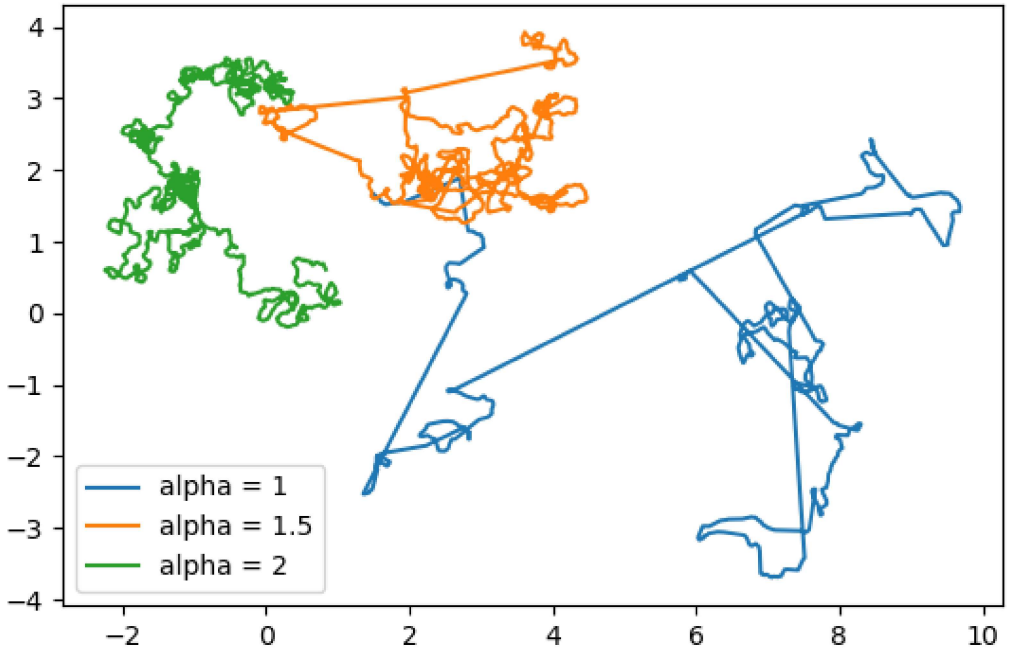


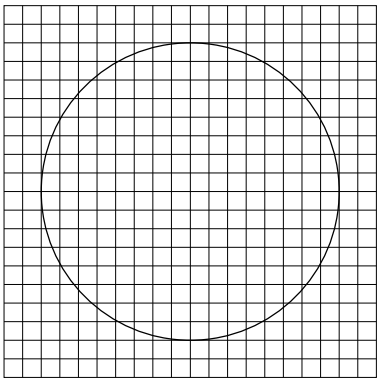
Figure 3.3: Example trajectory in homogeneous environment.

3.2.3 Concave Environment

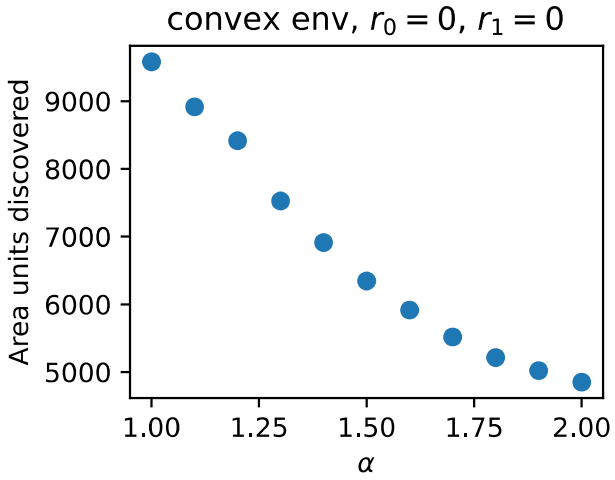
The third simulation also features obstacles, but this time with a concave shape. The obstacles are generated by creating a circle for each cell and not allowing the agent to exit this circle while in the cell. Since the diameter of the circle is larger than that of the cell side, there will be regions in which the agent can leave the cell without interacting with the circle. The boundary conditions are similar to the convex case, only that the agent is pushed into the circle toward the middle instead of out of it; as shown in figure 3.5b. Some example trajectories can be seen in figure 3.6c. The diameter of the circle is selected so that the obstacles cover 0.8 of the cell side as seen in figure 3.6a.

In figure 3.6b we can see the results of simulations in this environment. We see a local maximum for discoveries over α, meaning that a balance of diffusive and ballistic movement is advantageous in this environment. These results are consistent with what has been observed in previous studies [18].

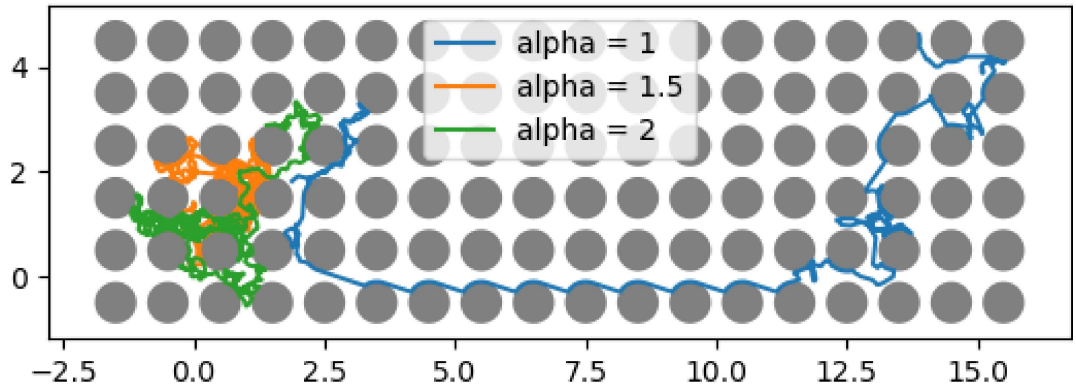
When the system moves toward low α, agents travel along straight lines to a higher degree, which leads them to get stuck in the concave obstacles. If the movement pattern is too diffusive, however, we get a decrease in efficiency regardless of the obstacle; as seen in the homogeneous case.



(a) Single convex cell. Grid in background is area units.

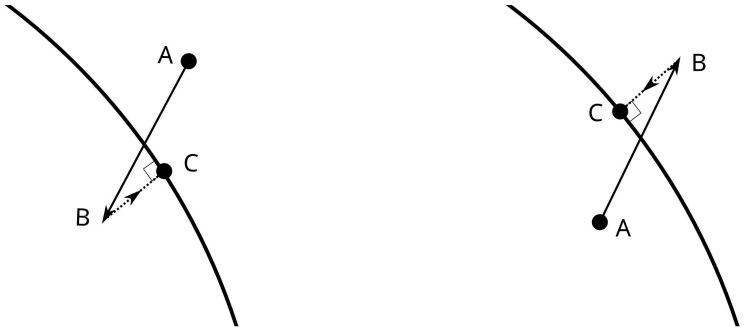


(b) Performance of agents with varying alpha in environment with convex obstacle.



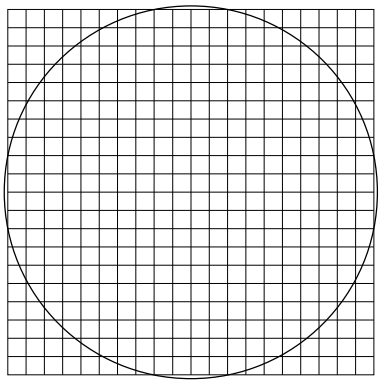
(c) Example trajectory in convex environment.

Figure 3.4: Summary of initial simulation with convex obstacles.

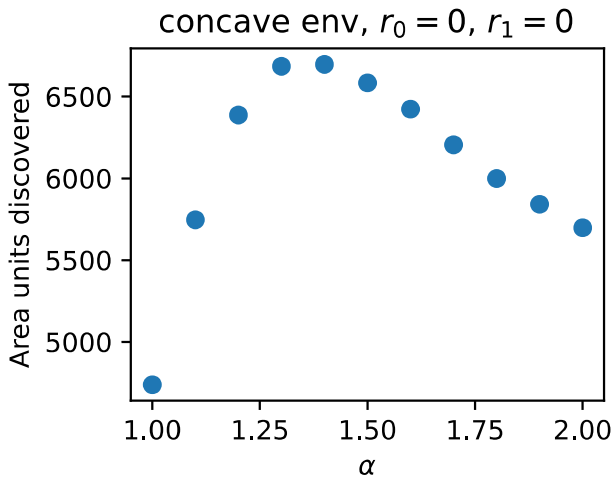


(a) Convex **(b)** Concave

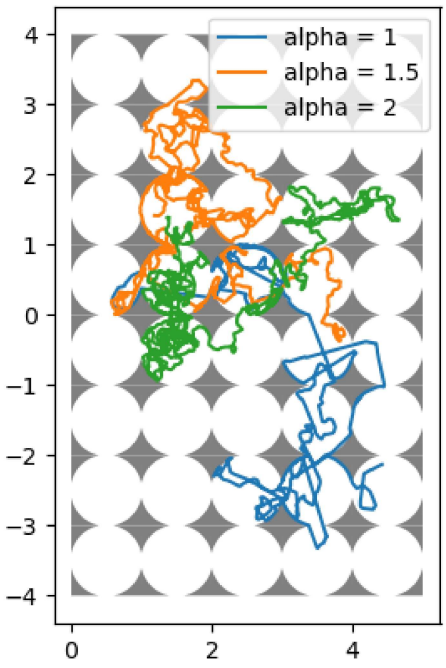
Figure 3.5: Illustration of collision with (a) convex obstacle and (b) concave obstacle. AB is attempted movement. BC is change in position due to obstacle. AC is full movement over timestep.



(a) Single concave cell. Grid in background is area units.



(b) Performance of agents with varying alpha in environment with concave obstacles.



(c) Example trajectory in concave environment.

Figure 3.6: Summary of initial simulation with concave obstacles.

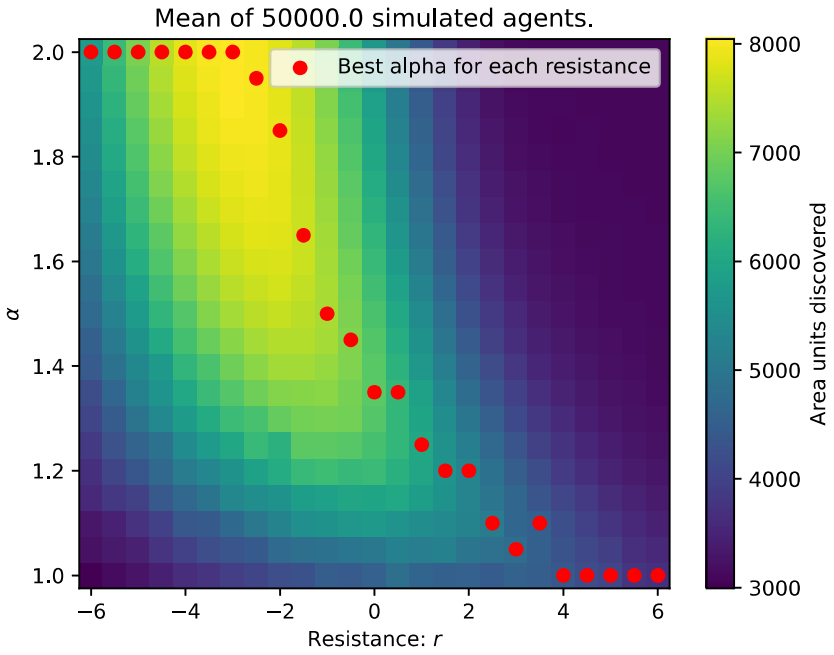


Figure 3.7: The optimal α for each resistance marked with a red dot in each column. The background colour gradient indicates the performance of each alpha at a given resistance.

3.3 Local Optimization

In this section we will introduce new strategies that try to maximize search efficiency in environments with spatially varying resistances, meaning $r_0 \neq r_1$.

Before doing that, we will try to find the optimal alpha for a range of resistances in environments whee $r_0 = r_1$. These optimal values will then be used in the new strategies.

3.3.1 Optimal Alphas

To find the optimal alpha for each resistance level, we run simulations with the same parameters as in section 3.2, but with changing resistances between -6 and 6 in 0.5 intervals, leading to 25 values of $r_0 = r_1$. For each resistance we will test α between 1 and 2 in 0.05 intervals, leading to 21 values.

This simulation gives us an optimal alpha for each resistance and the results can be seen in figure 3.7.

3.3.2 Locally Optimal Strategy

With knowledge of the optimal *alpha* for different resistances, we can create a movement strategy that is adjusted to the local environment. To create a spatially varied

environment, we will set $r_0 \neq r_1$.

This movement strategy, which is going to be referred to as "local" when labeled in figures, works similarly to the standard levy movement used for the initial simulations. The difference is that when a new effort value is drawn, it is not always drawn from the same distribution. Instead, the agent draws the new effort from the stable distribution with the α that was deemed optimal in section 3.3.1 for the resistance it is currently experiencing.

We can see in figure 3.8a that the local strategy slightly outperforms the best single alpha strategy in an environment with $(r_0, r_1) = (6, 4)$. In figure 3.8b however, with $(r_0, r_1) = (0, -6)$, the local strategy does not outperform the best alpha but is instead closer in performance to the worst alpha. Looking at table 3.1 we can see that the local strategy outperforms the best alpha in only 2 of 11 environments tested. Hypotheses as to why the performance is so low will be discussed in Chapter 4.

3.3.3 Improved Local Strategy

In an effort to improve upon the local strategy, we introduce a strategy that in figures will be referred to as "local$_s$". It is based on the local strategy with the difference that when the agent crosses over to an area with a new resistance, it will immediately set its effort to zero, causing a direction change and a new effort to be generated with the optimal distribution of the new local environment.

In table 3.1 We can see that the local$_s$ strategy outperforms the local strategy in all 9 of 11 tested environments and that it outperforms the best static alpha in 8 out of 11 environments. Two examples are displayed in figure 3.8.

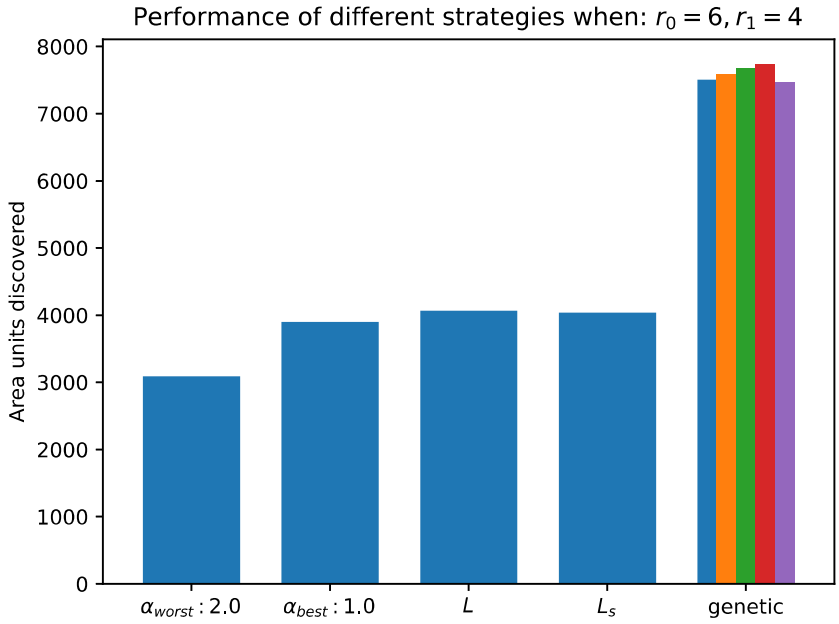
3.4 Genetic strategy

To further improve over the local$_s$ strategy, we would like a strategy that is not dependant on information from a parameter search, that is similar to a blind search, and that produces a high search efficiency. Ideally, the performance should be higher than that of local$_s$.

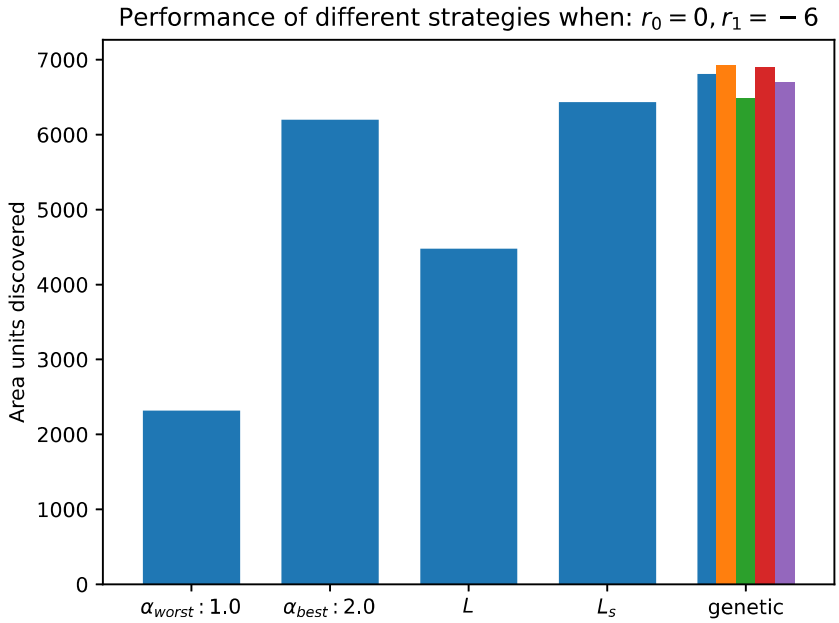
In order to fulfill these criteria, we propose a strategy where the agent is controlled by a neural network that takes as input the recent discoveries the agent has made. The network will act in a similar way to the levy distribution in earlier simulations in that it will generate an effort value whenever the agent turns.

The network will be trained using a genetic algorithm that sets the weights and biases through the chromosome. The population size is 100 and the training lasts 300 generations.

An interesting aspect in letting this strategy develop through a genetic algorithm is that we can study what kind of distributions of effort values emerge when the



(a)



(b)

Figure 3.8: Overview of the performance of different strategies when $(r_0, r_1) = (6, 4)$ and $(r_0, r_1) = (0, -6)$. Each bar in the genetic category represents a separate instance of the genetic strategy that has been trained independently.

network generating the distribution is exposed to the pressure of natural selection.

3.4.1 Genetic algorithm

In order to use a genetic algorithm, we need to encode the network into a chromosome. This is done by creating an array where the network weights and biases are stored as floating-point values.

Selection is carried out through tournament selection with a tournament size of 2 and a 100% probability that the most fit individual is selected. The 100% probability is due to there already being randomness in agent performance generated from the random movement characteristics of the agents. There is therefore no need to introduce further randomness.

Crossover is done using two-point crossover with a crossover probability of 90%.

Mutation is done using Gauss shifting, meaning that if a gene is selected for mutation, the value will be shifted by adding a value from a normal distribution. In this case, we have used a normal distribution with mean 0 and standard deviation 0.1. The mutation rate is $\frac{1}{n}$ where n is the length of the chromosome, meaning we have on average 1 mutation per chromosome per generation.

The training is carried out over 300 generations with a population of 100 individuals. Each individual is evaluated by running a simulation for 10^4 timesteps using the genetic strategy. There are no repeats and averages over many runs in the evaluation of each agent.

3.4.2 Shape of the Network

A genetic algorithm generally requires the dimensionality of the parameter space to be low to perform well. Therefore, we use a network of minimal size to allow faster convergence. The network used to generate the results in this report has 4 inputs, 2 hidden nodes, and one output node, resulting in 13 total parameters to optimize. The hidden layer uses relu activation and the output node has no activation function. Negative values from the output have the same effect as zero values, however, so it is effectively the same as a relu activation.

3.4.3 Memory

To generate an input for the network that is based on recent success of the agent some sort of memory is needed. The way we chose to represent that memory is as an array of 256 1:s and 0:s. Each number in the array corresponds to a recent timestep. If the agent took its first step in a new area unit in that timestep, it will register as a 1. If the agent takes a step in an area unit that has already been visited before, it will register a 0. This memory is then split into 4 sections of 64 digits, where the numbers in each section are summed up, and the result of these sums are what is being fed to the network as input.

(r_0, r_1)	α_{worst}	α_{best}	L	L_s	genetic
(-4, -6)	3136	6612	6617	6595	7915
(-2, -6)	3054	6838	6689	6781	7628
(0, -6)	2317	6200	4474	6436	6765
(2, -6)	1854	4932	2612	5776	5160
(3, -4)	2482	5163	2825	6193	5950
(4, -3)	2788	5129	2929	5899	6015
(6, -6)	1946	4699	2004	4978	5056
(6, -2)	3205	5020	3406	5369	5105
(6, 0)	3797	4621	4060	4922	6303
(6, 2)	3259	4264	4158	4321	7267
(6, 4)	3092	3898	4065	4035	7589
SUM	30930	57376	43839	61305	70753

best
2:nd
3:rd
4:th
worst

Table 3.1: Overview of performance (number of area units discovered) for different strategies in convex environments with varying resistance. Detailed plots on each simulation can be found in appendix B.

3.4.4 Genetic Results

When pre-trained agents are tested under the same conditions as the agents with a static or locally optimized alpha we can see a high overall performance in table 3.1. The genetic strategy outperforms the local strategy in all cases. It outperforms the local_s strategy in 8 of 11 environments. We can also see the high performance of the genetic strategy in figure 3.8.

The distributions of effort values produced by the genetically trained networks during simulation can be seen in appendix B. It can be seen from the difference in efforts generated in the even and odd columns that the agent notices a difference between the two areas and adjusts its strategy accordingly.

Genetic training also seems to be very consistent in converging to the same performance when several agents are trained in the same conditions. This was tested by running five genetic training processes for each environment. In 10 out of 11 environments, the independent processes converged to similar performance and distribution of effort values. In 1 of the 11 environments, one of the genetic training processes produced a noticeably worse agent. The median area units discovered by the populations after each generation for the case with a diverging process can be seen in figure B.44 while an example where all training processes converged to similar values can be seen in figure B.14.

3.5 Many agents

When creating the simulation to generate figure 3.7 I at one point had a bug in my code that resulted in releasing 100 agents at a time into the environment instead of one. The combined search efficiency of these 100 agents was then measured.

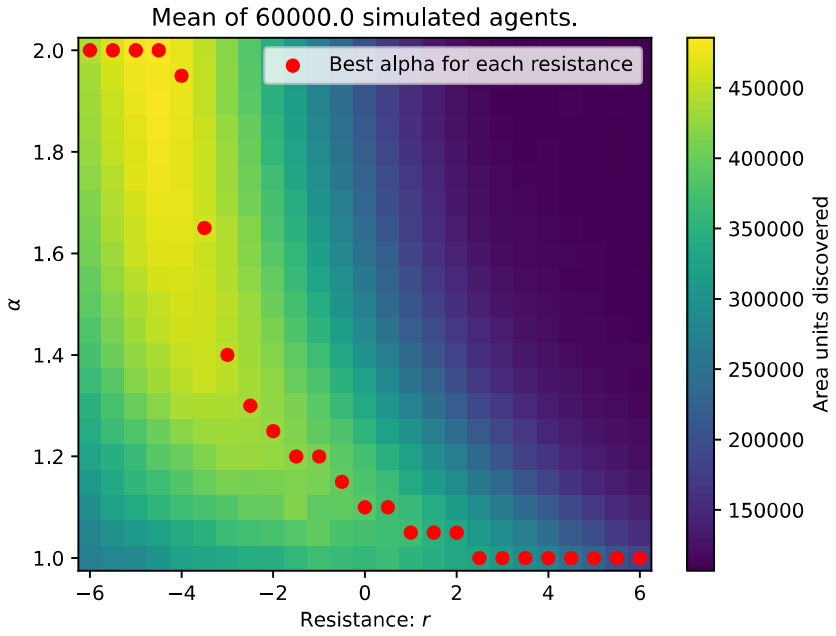


Figure 3.9: The optimal α for each resistance marked with a red dot in each column. The background colour gradient indicates the performance of each alpha at a given resistance. The performance measures the combined search area of 100 agents released in close proximity to each other.

I thought the results were interesting, so they are presented in figure 3.9 and a comparison to the case with one agent can be seen in figure 3.10.

We can see that ballistic movement becomes more favorable if we consider the combined search area when many agents are released at the same time.

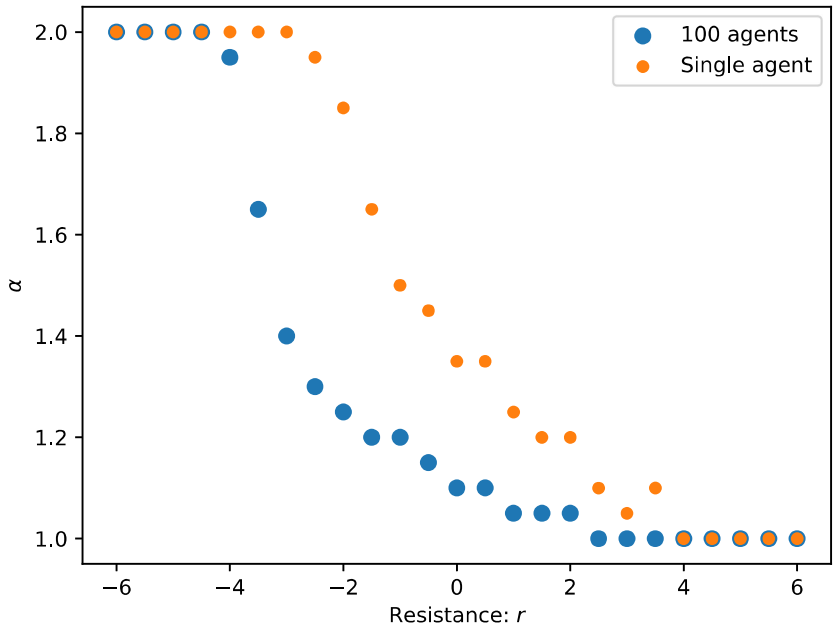


Figure 3.10: The optimal α for each resistance in the case of a single agent and 100 agents.

4

Discussion

In this chapter we will analyse the results in more depth, discuss some hypotheses of what is the underlying reason for the behaviours we are seeing, and list some things that would be interesting to explore further.

I parts of the discussion we will refer to appendix B so it might be helpful to keep a copy at hand in a parallell pdf reader or as printed paper on the side.

4.1 Initial Simulations

More ballistic agents will tend to find new area more often as they are diffusing further into the environment. I did not measure the MSD in this thesis but it has been shown that levy flights with a lower α will have a higher diffusion coefficient and therefore tend to find new ground more often as they disperse further away from the initial position [].

This dynamic leads to higher performance for low α agents in the homogeneous environment but also in the convex environment. In the concave environment however, low α agents are not the best performers. This can be understood by considering a high effort movement, which becomes more common as α aproaches 1.

When an agent performs a high effort move it is travelling along a straight line for an extended period. All straight trajectories excepth those where

$$\theta \in \{\frac{\pi k}{2} : k \in \mathbb{Z}\}$$

will eventually hit an obstacle. When they do they will be stuck moving against the obstacle perimeter as long as their direction is in the range

$$\theta \in \{\frac{\pi k e}{2} : k \in \mathbb{Z}\}, e \in \left(-\frac{\theta_f}{2}, \frac{\theta_f}{2}\right)$$

where θ_f is the angular size of each opening in the circle that makes up the concave obstacle.

As the probability distribution of θ is uniform one can make the conclusion that when moving in a straight line with a random θ, the probability of being stuck in

place is

$$p_s = 1 - \frac{2\theta_f}{\pi}.$$

In the simulations in this thesis the openings in the concave obstacles are 0.2 l.u. long which gives a radial opening of:

$$\theta_f = cos^{-1}\left(\frac{14}{15}\right) \approx 0.367 \text{ rad}.$$

Which meanst that

$$p_s \approx 0.766$$

and that most ballistic movement gets stuck on an obstacle in the concave environment.

4.2 Diffusive better than ballistic overall

Going back to the results seen in figure 3.7, we can see that the most efficient seaches happening in the concave environment are when the movement pattern is diffusive and the resistance is low. This indicates that for these regular concave obstacles, it is efficient to move with normally distributed step sizes given the right scaling parameter of the distribution. In contrast, it is not as efficient to move with stable distributions of lower α and optimal rescaling. This is interesting as ballistic Levy flights are generally considered more efficient.

4.3 Locally Optimized

A hypothesis for the low performance of the local strategy in many environments was at an early stage that the low performance is due to agents crossing over from a high-resistance region to a low-resistance region. When this happens, the agent is likely to have a high effort value drawn from a distribution that is efficient in the high resistance area. This effort will then be misaligned with the new resistance, and might require many simulation steps in the new resistance to decrease the effort to zero. This leads to more ballistic movement than what was intended, and the performance is lower due to the disadvantages of ballistic movement in a concave environment.

In order to examine this hypothesis we created the local_s strategy as resetting the effort when moving into areas of new resistance would eliminate the above mentioned problem. As the performance of the local_s strategy was much better than that of the local strategy we have assumed our earlier hypothesis to be true.

4.4 Genetic Strategy

The genetic strategy is the most successful one tested in this thesis, but the exact reason is not known. In this section, we will discuss some of the possibilities as to why the genetic strategy is performing better than the others.

4.4.1 Differentiating Resistances

Even though the genetic agents are only given the recent discoveries in a very low resolution it is able to differentiate between being in a low resistance region and being in a high resistance region. This can be seen by examining the figures displaying the distribution of effort values for trained agents in appendix B.

Though this is most likely an important part of the strategy, it is not enough to explain the performance, as the local and local$_s$ strategies have direct access to the local resistance.

4.4.2 More efficient distribution

What is most likely also happening is that the distribution of values generated by the network is better adapted to the convex environment than a stable distribution.

None of the genetic networks generate the large outliers that are a distinct feature of heavy-tail distributions. This could lead to better efficiency and more stable performance in a concave environment that potentially punishes outlier efforts by getting agents stuck in obstacles.

In some cases, we see distributions that select between a few values suggesting flights that consist of a few separate step lengths. This can be seen when $(r_0, r_1) \in \{(-4,-6),(-2,-6),(0,-6),(2,-6)(6,-6),(6,-2)\}$, in figures B.5, B.10, B.15, B.20, B.35, and B.40.

In other cases, the distribution looks more continuous, with a preference for lower values with frequency sloping downwards for higher efforts. This can be seen when $(r_0, r_1) \in \{(3,-4),(4,-3),(6,0),(6,2)(6,4)\}$, in figures B.25, B.30, B.45, B.50, and B.55.

A trend that can be noticed is that of the 6 cases where only a few values are selected, 5 of them have a column with resistance -6 while none of the environments generating more continuous distributions features that resistance. It might be that this extreme low resistance is what causes this type of distribution. However, that does not explain the $(r_0, r_1) = (6,-2)$ case.

It is also interesting that the genetic algorithm converges to similar distributions for each environment, even though the learning process is separate for each instance. This indicates that it is more advantageous to use discrete distributions in some environments and continuous distributions in others.

4.5 Performance with Different Resistances

Examining table 4.1 we can see the performance of each strategy, relative to the best strategy in each environment.

(r_0, r_1)	α_{worst}	α_{best}	L	L_s	genetic
(-4, -6)	0.3961	0.8353	0.8360	0.8332	1.0000
(-2, -6)	0.4004	0.8965	0.8768	0.8889	1.0000
(0, -6)	0.3425	0.9165	0.6614	0.9513	1.0000
(2, -6)	0.3209	0.8539	0.4523	1.0000	0.8933
(3, -4)	0.4008	0.8336	0.4562	1.0000	0.9607
(4, -3)	0.4636	0.8527	0.4870	0.9808	1.0000
(6, -6)	0.3849	0.9294	0.3964	0.9845	1.0000
(6, -2)	0.5969	0.9349	0.6343	1.0000	0.9508
(6, 0)	0.6023	0.7330	0.6441	0.7808	1.0000
(6, 2)	0.4485	0.5868	0.5722	0.5947	1.0000
(6, 4)	0.4075	0.5136	0.5357	0.5317	1.0000
SUM	4.7644	8.8862	6.5524	9.5459	10.8048

best
2:nd
3:rd
4:th
worst

Table 4.1: Overview of performance (number of area units discovered) for different strategies in convex environments with varying resistance. Same as table 3.1 but values are normalized to the best performer in each environment.

It is notable that the local strategy outperforms the local$_s$ strategy in two cases, $(r_0, r_1) = (-4, -6)$ and $= (6, 4)$. What is special about these environments is that they have such extreme resistances that both resistances have the same optimal alpha. That means that the best alpha and the local strategy is the exact same and it is only random variation that makes a difference between the strategies. The local$_s$ strategy behaves slightly different from the others as it will still reset it's effort when crossing between columns but it should still be very similar to the best alpha and the local strategy.

Another thing worth noting is that the genetic strategy performs best in relation to the second best strategy when resistance is very high. With $(r_0, r_1) = (6, 2)$ and $= (6, 4)$ we see a 41% and 44% drop in performance from the genetic strategy to the next best strategy.

5

Conclusion

The results of our initial simulations confirm earlier results that movement in between Brownian and superdiffusive motion is optimal for certain environmental topologies.

We also conclude that a strategy that is efficient for all local conditions separately does not have to be efficient in the wider environment, as it does not necessarily take into account the changing nature of its surroundings.

A third conclusion is that genetically trained neural networks can adapt to local conditions with minimal input from the environment and that this can be efficient even with minimal neural networks.

A final conclusion is that there might be other distributions which are more efficient than the heavy tailed stable distriburions. This can be argued since the genetic agent outperforms the agents moving in traditional Levy flights and the distribution of efforts generated by the networks are not stable distributions.

Bibliography

[1] G H Pyke. "Optimal Foraging Theory: A Critical Review". In: *Annual Review of Ecology and Systematics* 15.1 (Nov. 1984), pp. 523–575. DOI: 10.1146/annurev.es.15.110184.002515. URL: https://doi.org/10.1146/annurev.es.15.110184.002515.

[2] Graham H. Pyke. "Optimal Foraging Theory: An Introduction". In: *Encyclopedia of Animal Behavior*. Elsevier, 2019, pp. 111–117. DOI: 10.1016/b978-0-12-809633-8.01156-0. URL: https://doi.org/10.1016/b978-0-12-809633-8.01156-0.

[3] AM Reynolds. "On the intermittent behaviour of foraging animals". In: *EPL (Europhysics Letters)* 75.4 (2006), p. 517.

[4] Anis Farhan Kamaruzaman et al. "Levy Flight Algorithm for Optimization Problems - A Literature Review". In: *Applied Mechanics and Materials* 421 (Sept. 2013), pp. 496–501. DOI: 10.4028/www.scientific.net/amm.421.496. URL: https://doi.org/10.4028/www.scientific.net/amm.421.496.

[5] Blaine J. Cole. "Fractal time in animal behaviour: the movement activity of Drosophila". In: *Animal Behaviour* 50.5 (1995), pp. 1317–1324. DOI: 10.1016/0003-3472(95)80047-6. URL: https://doi.org/10.1016/0003-3472(95)80047-6.

[6] Christoph Tholen et al. "On the Robustness of Self-Adaptive Levy-Flight". In: *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*. IEEE, May 2018. DOI: 10.1109/oceanskobe.2018.8559485. URL: https://doi.org/10.1109/oceanskobe.2018.8559485.

[7] Wei Kaidi, Mohammad Khishe, and Mokhtar Mohammadi. "Dynamic Levy Flight Chimp Optimization". In: *Knowledge-Based Systems* 235 (Jan. 2022), p. 107625. DOI: 10.1016/j.knosys.2021.107625. URL: https://doi.org/10.1016/j.knosys.2021.107625.

[8] Hüseyin Haklı and Harun Uğuz. "A novel particle swarm optimization algorithm with Levy flight". In: *Applied Soft Computing* 23 (Oct. 2014), pp. 333–345. DOI: 10.1016/j.asoc.2014.06.034. URL: https://doi.org/10.1016/j.asoc.2014.06.034.

[9] David W. Sims et al. "Scaling laws of marine predator search behaviour". In: *Nature* 451.7182 (Feb. 2008), pp. 1098–1102. DOI: 10.1038/nature06518. URL: https://doi.org/10.1038/nature06518.

[10] Nicolas E. Humphries et al. "Environmental context explains Lévy and Brownian movement patterns of marine predators". In: *Nature* 465.7301 (June 2010),

pp. 1066–1069. DOI: 10.1038/nature09116. URL: https://doi.org/10.1038/nature09116.

[11] David A. Raichlen et al. "Evidence of Lévy walk foraging patterns in human hunter–gatherers". In: *Proceedings of the National Academy of Sciences* 111.2 (Dec. 2013), pp. 728–733. DOI: 10.1073/pnas.1318616111. URL: https://doi.org/10.1073/pnas.1318616111.

[12] David W. Sims et al. "Hierarchical random walks in trace fossils and the origin of optimal search behavior". In: *Proceedings of the National Academy of Sciences* 111.30 (July 2014), pp. 11073–11078. DOI: 10.1073/pnas.1405966111. URL: https://doi.org/10.1073/pnas.1405966111.

[13] G. J. Pierce and J. G. Ollason. "Eight Reasons Why Optimal Foraging Theory Is a Complete Waste of Time". In: *Oikos* 49.1 (May 1987), p. 111. DOI: 10.2307/3565560. URL: https://doi.org/10.2307/3565560.

[14] Andy Reynolds. "HOW MANY ANIMALS REALLY DO THE LÉVY WALK?" In: *Ecology* 89.8 (Aug. 2008), pp. 2347–2351. DOI: 10.1890/07-1688.1. URL: https://doi.org/10.1890/07-1688.1.

[15] Andrew M. Edwards. "Overturning conclusions of Lévy flight movement patterns by fishing boats and foraging animals". In: *Ecology* 92.6 (June 2011), pp. 1247–1257. DOI: 10.1890/10-1182.1. URL: https://doi.org/10.1890/10-1182.1.

[16] Graham H. Pyke. "Understanding movements of organisms: it's time to abandon the Lévy foraging hypothesis". In: *Methods in Ecology and Evolution* 6.1 (Nov. 2014). Ed. by Luca Giuggioli, pp. 1–16. DOI: 10.1111/2041-210x.12298. URL: https://doi.org/10.1111/2041-210x.12298.

[17] G.M. Viswanathan, E.P. Raposo, and M.G.E. da Luz. "Lévy flights and superdiffusion in the context of biological encounters and random searches". In: *Physics of Life Reviews* 5.3 (Sept. 2008), pp. 133–150. DOI: 10.1016/j.plrev.2008.03.002. URL: https://doi.org/10.1016/j.plrev.2008.03.002.

[18] Giorgio Volpe and Giovanni Volpe. "The topography of the environment alters the optimal search strategy for active particles". In: *Proceedings of the National Academy of Sciences* 114.43 (Oct. 2017), pp. 11350–11355. DOI: 10.1073/pnas.1711371114. URL: https://doi.org/10.1073/pnas.1711371114.

[19] Fernando A. Oliveira et al. "Anomalous Diffusion: A Basic Mechanism for the Evolution of Inhomogeneous Systems". In: *Frontiers in Physics* 7 (Feb. 2019). DOI: 10.3389/fphy.2019.00018. URL: https://doi.org/10.3389/fphy.2019.00018.

[20] A. Einstein. "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen". In: *Annalen der Physik* 322.8 (1905), pp. 549–560. DOI: 10.1002/andp.19053220806. URL: https://doi.org/10.1002/andp.19053220806.

[21] Ralf Metzler et al. "Anomalous diffusion models and their properties: non-stationarity, non-ergodicity, and ageing at the centenary of single particle tracking". In: *Phys. Chem. Chem. Phys.* 16.44 (2014), pp. 24128–24164. DOI: 10.1039/c4cp03465a. URL: https://doi.org/10.1039/c4cp03465a.

[22] *File:MSD anomalous diffusion.svg*. URL: https://commons.wikimedia.org/wiki/File:Msd_anomalous_diffusion.svg.

[23] *File:levy distributionpdf.svg*. URL: https://en.wikipedia.org/wiki/File:Levy_distributionPDF.svg.

[24] *File:levy0 ldistributionpdf.svg*. URL: https://en.wikipedia.org/wiki/File:Levy0_LdistributionPDF.svg.

[25] Alexei V. Chechkin et al. "Introduction to the Theory of Lévy Flights". In: *Anomalous Transport*. Wiley-VCH Verlag GmbH & Co. KGaA, pp. 129–162. DOI: 10.1002/9783527622979.ch5. URL: https://doi.org/10.1002/9783527622979.ch5.

[26] Mattias Wahde. *Biologically inspired optimization methods: an introduction*. WIT press, 2008.

[27] *File:TwoPointCrossover.svg*. URL: https://commons.wikimedia.org/wiki/File:TwoPointCrossover.svg.

[28] G. M. Viswanathan et al. "Optimizing the success of random searches". In: *Nature* 401.6756 (Oct. 1999), pp. 911–914. DOI: 10.1038/44831. URL: https://doi.org/10.1038/44831.

[29] emiljanQ3. *Emiljanq3/self-optimizing-agents*. URL: https://github.com/emiljanQ3/Self-Optimizing-Agents.

A

Guide to Recreate Results

To recreate the results of this thesis, one can use the code located at the Github repository emiljanQ3/Self-Optimizing-Agents [29]. Rerunning these simulations will take some time, likely several days for each run depending on your setup. The data used to plot the figures in this thesis takes up 30 GB of disk space. There are a few steps required to produce the figures that are displayed in appendix B. A guide for how to do so is also available on the Github-page for this thesis [29].

B

Figures From All Simulations

In the coming sections in this appendix, we will present figures from all the simulations that were done in the final stretch of this thesis work. They are what most of the conclusions in this thesis are based on.

B.1 Figures Resistance (-4, -6)

Here follows figures from a simulations in an environment with $r_0 = -4, r_1 = -6$.

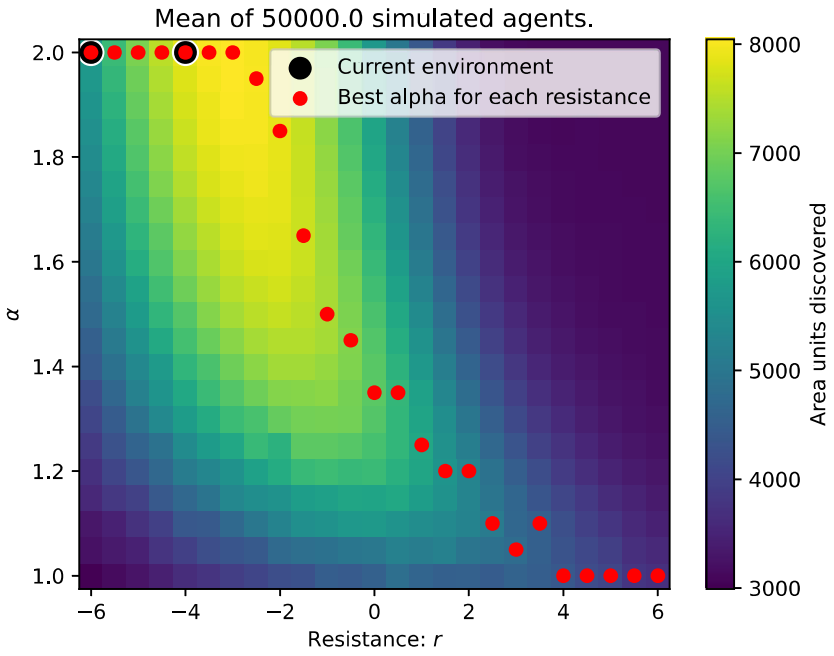


Figure B.1: A plane where color indicates search effectiveness for a given combination of resistance and alpha in the levy distribution. The alpha with the best result for each resistance is marked with red dots. The alphas and resistances that are used in this environment, $r_0 = -4, r_1 = -6$, are marked with a black border around the dot.

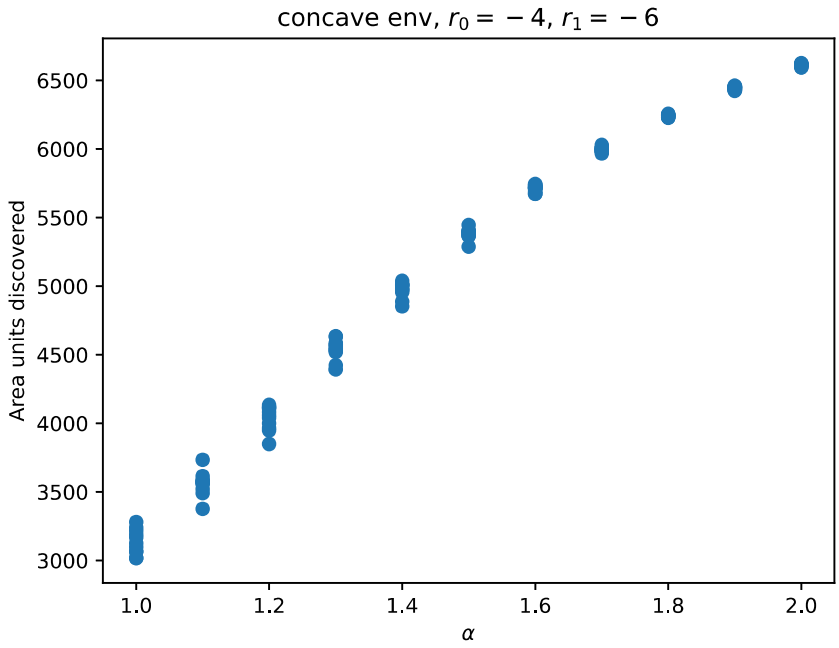


Figure B.2: Performance of each static alpha strategy in environment (-4, -6).

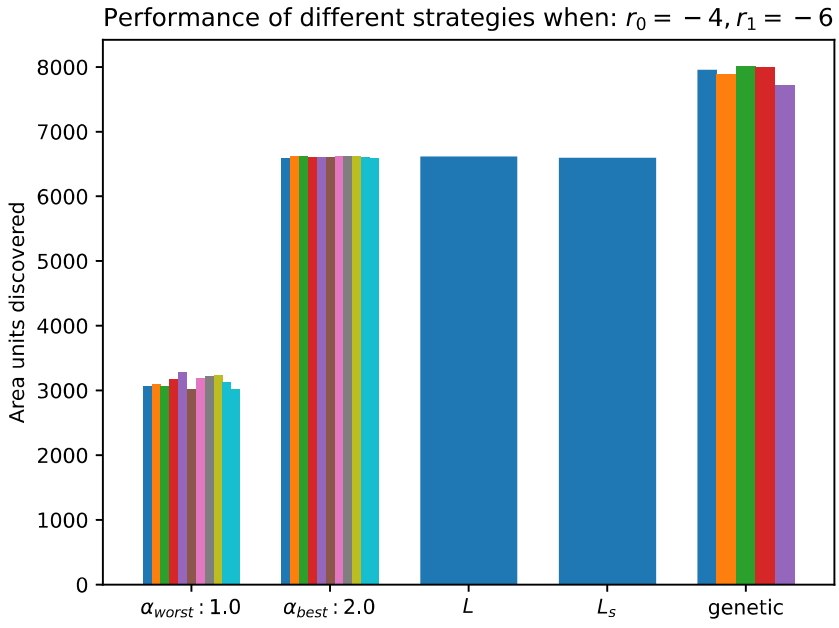


Figure B.3: Effectiveness of different search strategies can be seen in terms of mean area units discovered. α_{worst}: Worst alpha in the simple levy flight strategy. α_{best}: Best alpha in the simple levy flight strategy. local: Using the locally optimal alpha when drawing a new effort. local$_s$: Same as local but resets agent effort when moving into area with new resistance. genetic: Chooses effort through a neural network trained by a genetic algorithm.

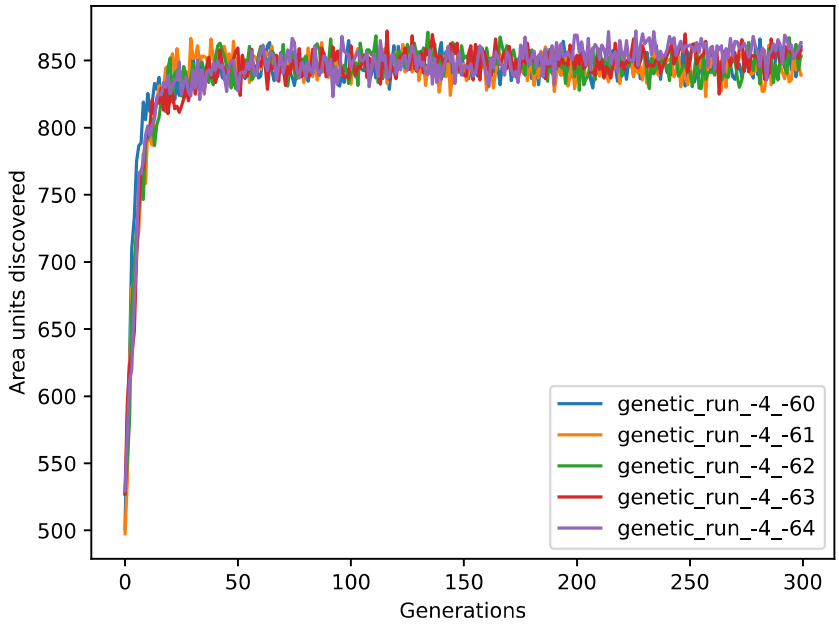


Figure B.4: Plot of the performance of the median agent in the population during genetic training.

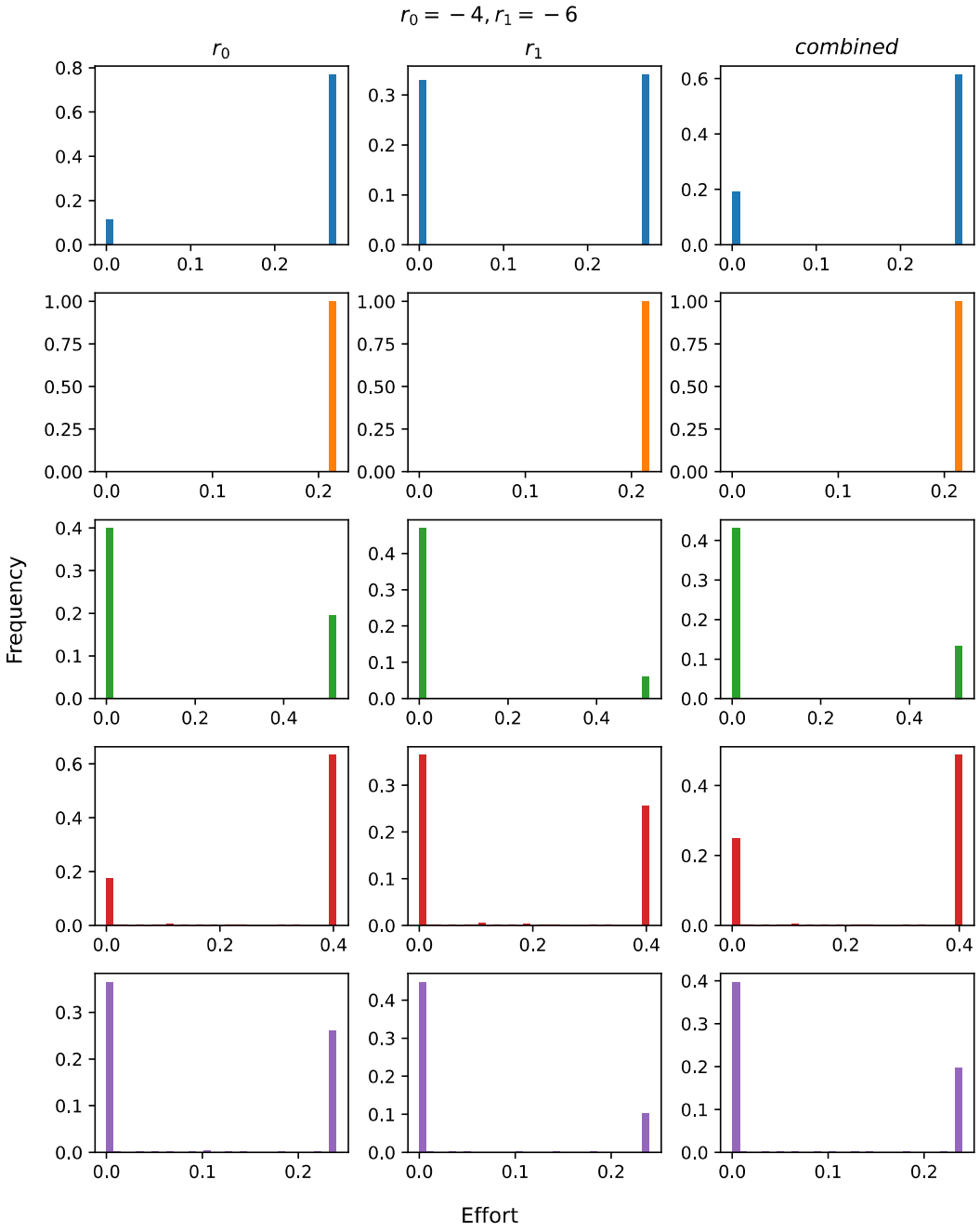


Figure B.5: Distribution of selected effort values for genitically trained agents. Each row contains data from a single agent. THe leftmost column is effort values sampled in resistance $r_0 = 3$, the middle in $r_1 = -4$, and the rightmost is from r_0 and r_1 combined.

B.2 Figures Resistance (-2, -6)

Here follows figures from a simulations in an environment with $r_0 = -2, r_1 = -6$.

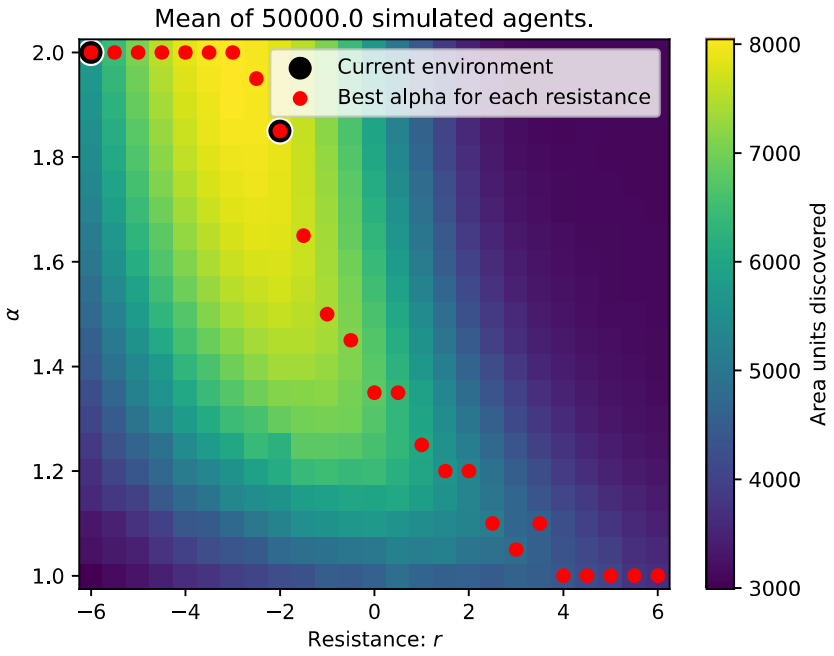


Figure B.6: A plane where color indicates search effectiveness for a given combination of resistance and alpha in the levy distribution. The alpha with the best result for each resistance is marked with red dots. The alphas and resistances that are used in this environment, $r_0 = -2, r_1 = -6$, are marked with a black border around the dot.

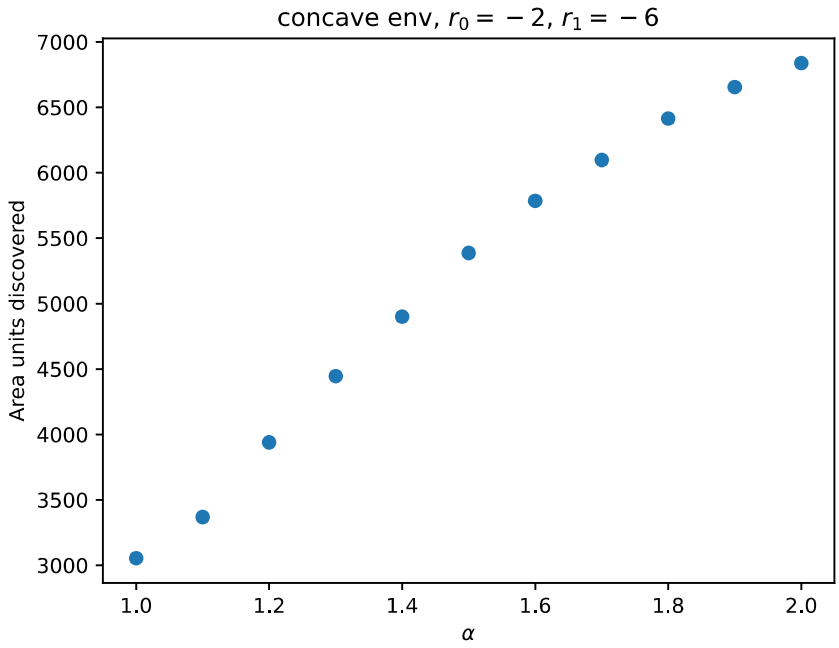


Figure B.7: Performance of each static alpha strategy in environment (-2, -6).

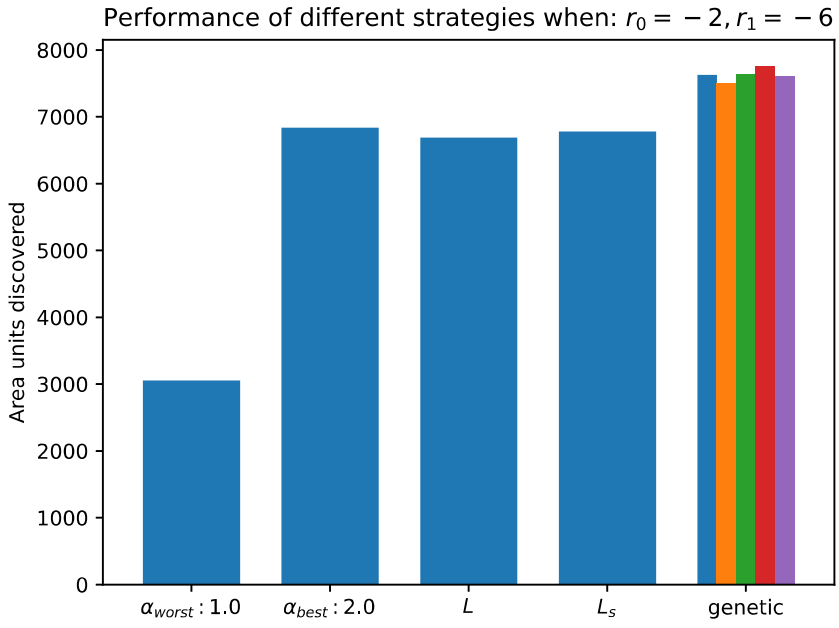


Figure B.8: Effectiveness of different search strategies can be seen in terms of mean area units discovered. α_{worst}: Worst alpha in the simple levy flight strategy. α_{best}: Best alpha in the simple levy flight strategy. local: Using the locally optimal alpha when drawing a new effort. local$_s$: Same as local but resets agent effort when moving into area with new resistance. genetic: Chooses effort through a neural network trained by a genetic algorithm.

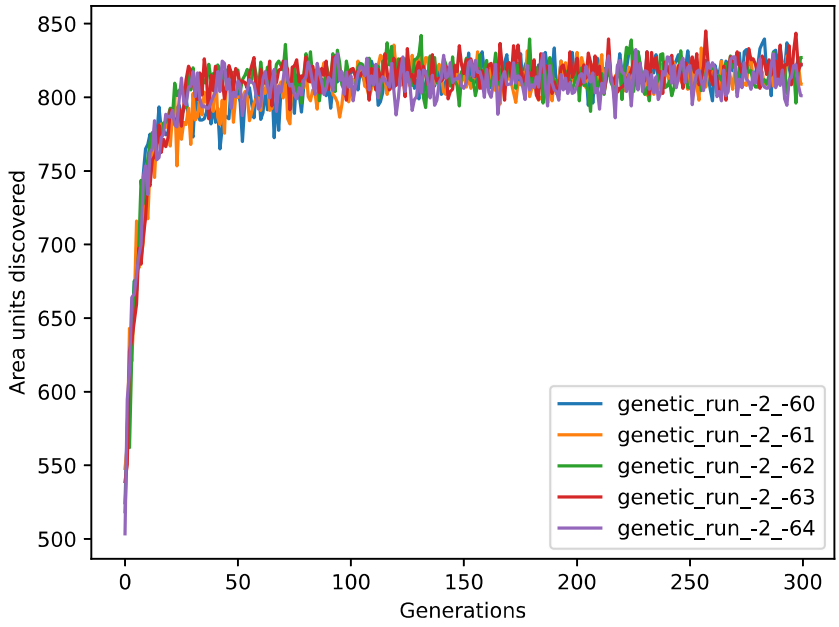


Figure B.9: Plot of the performance of the median agent in the population during genetic training.

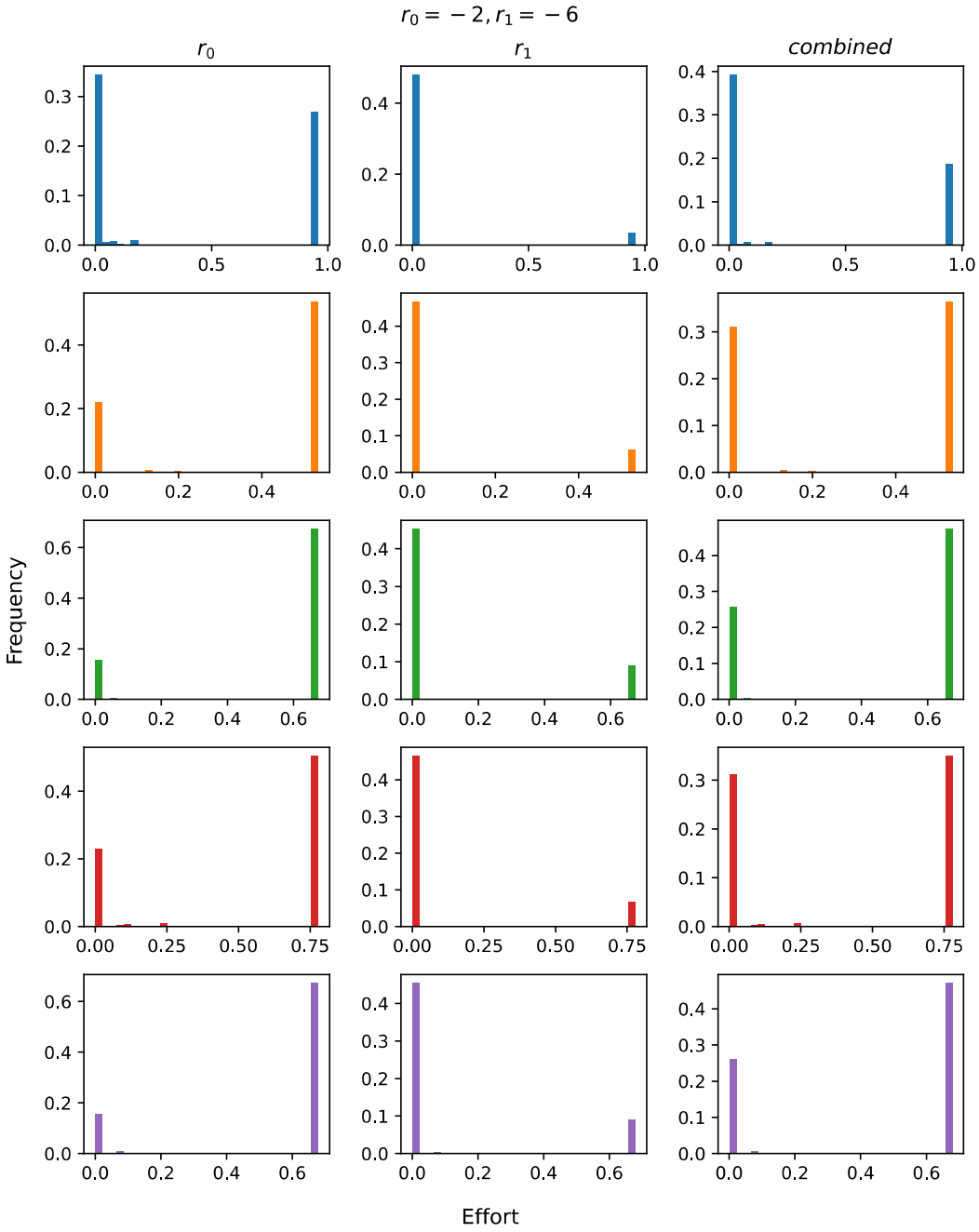


Figure B.10: Distribution of selected effort values for genitically trained agents. Each row contains data from a single agent. THe leftmost column is effort values sampled in resistance $r_0 = 3$, the middle in $r_1 = -4$, and the rightmost is from r_0 and r_1 combined.

B.3 Figures Resistance (0, -6)

Here follows figures from a simulations in an environment with $r_0 = 0, r_1 = -6$.

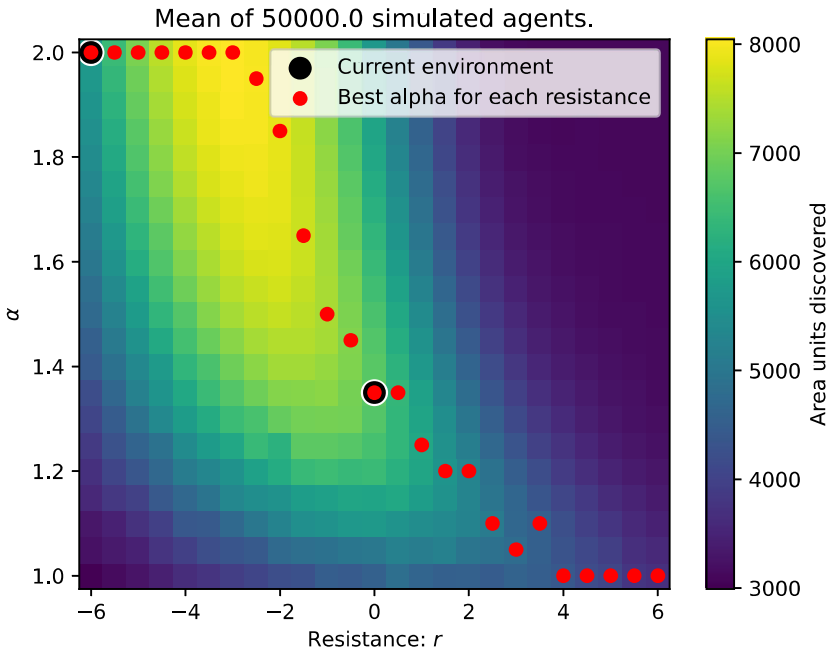


Figure B.11: A plane where color indicates search effectiveness for a given combination of resistance and alpha in the levy distribution. The alpha with the best result for each resistance is marked with red dots. The alphas and resistances that are used in this environment, $r_0 = 0, r_1 = -6$, are marked with a black border around the dot.

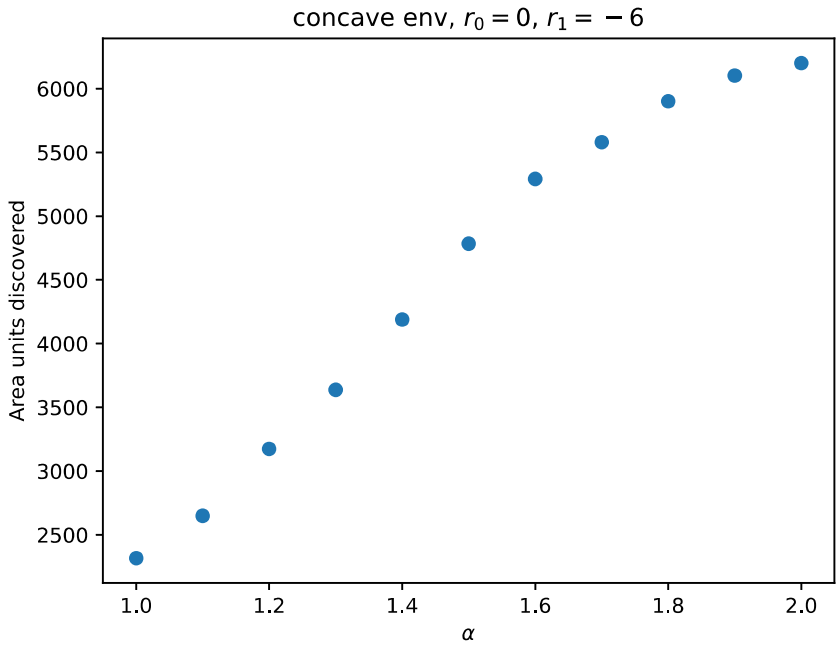


Figure B.12: Performance of each static alpha strategy in environment (0, -6).

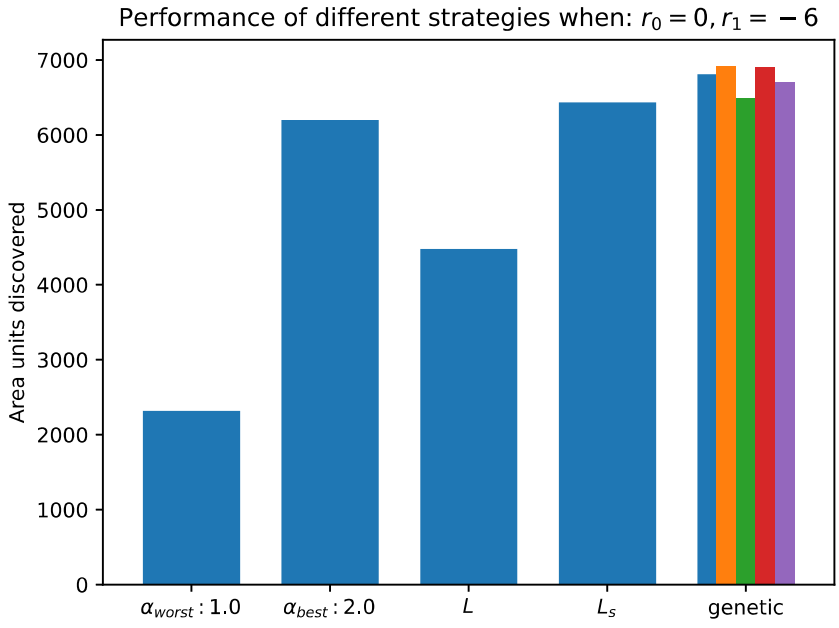


Figure B.13: Effectiveness of different search strategies can be seen in terms of mean area units discovered. α_{worst}: Worst alpha in the simple levy flight strategy. α_{best}: Best alpha in the simple levy flight strategy. local: Using the locally optimal alpha when drawing a new effort. local$_s$: Same as local but resets agent effort when moving into area with new resistance. genetic: Chooses effort through a neural network trained by a genetic algorithm.

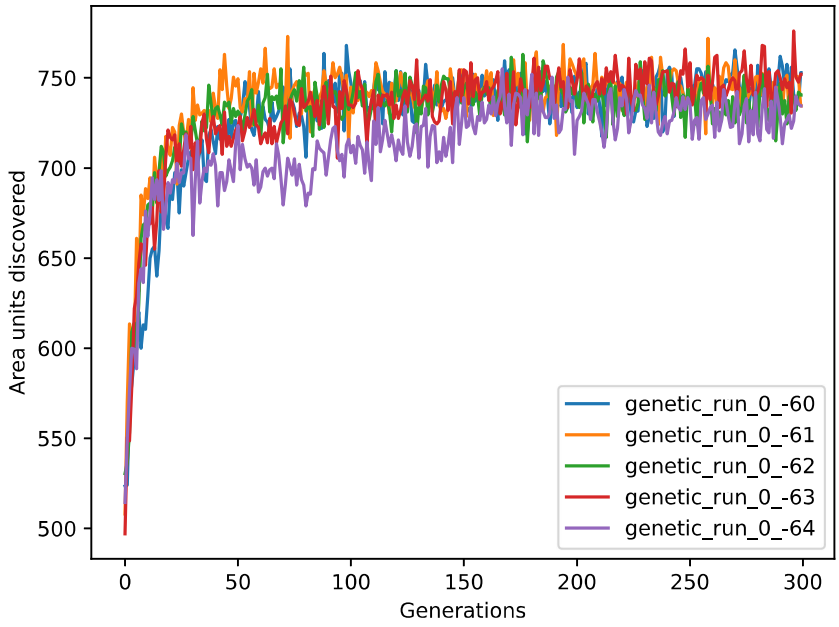


Figure B.14: Plot of the performance of the median agent in the population during genetic training.

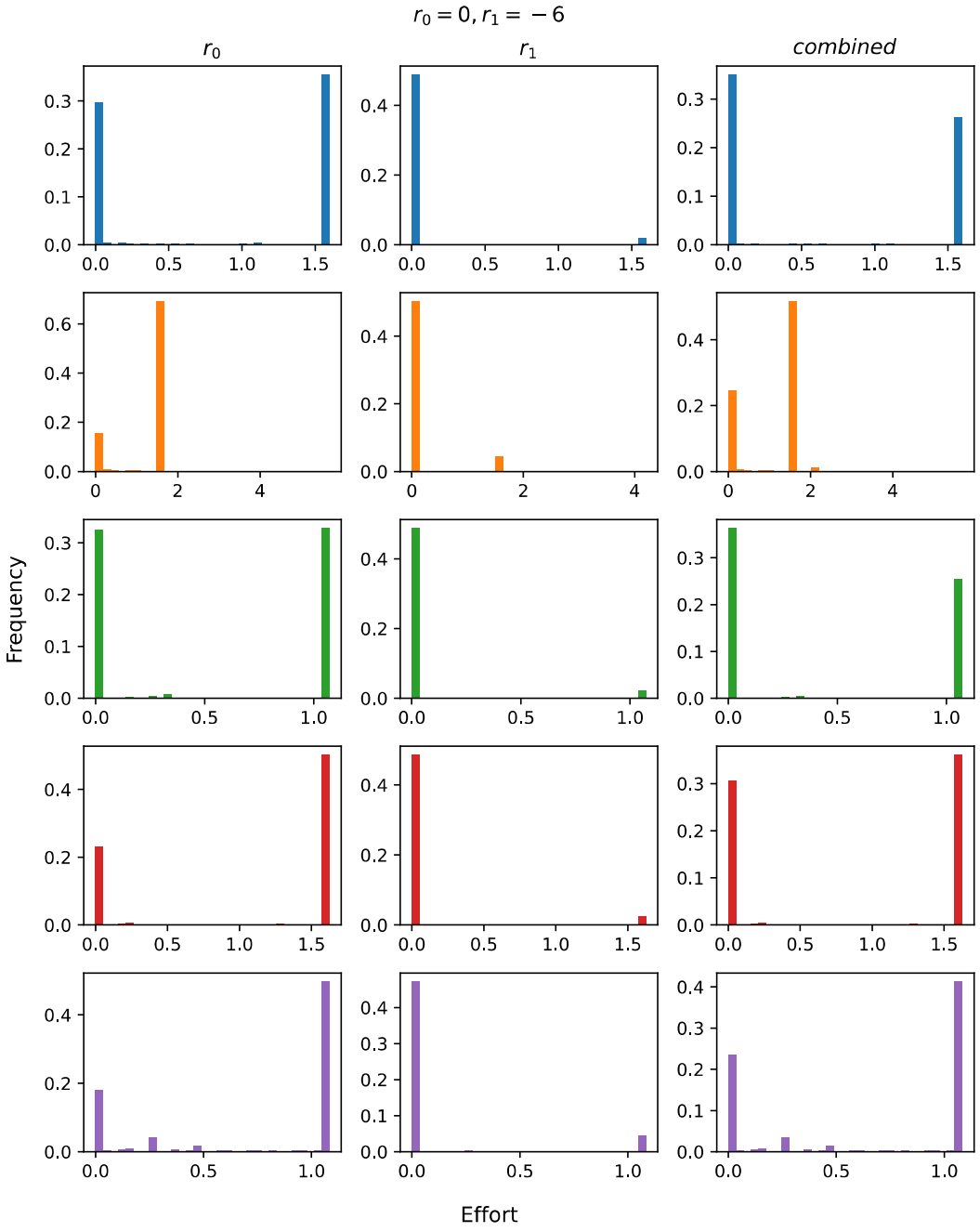


Figure B.15: Distribution of selected effort values for genitically trained agents. Each row contains data from a single agent. THe leftmost column is effort values sampled in resistance $r_0 = 3$, the middle in $r_1 = -4$, and the rightmost is from r_0 and r_1 combined.

B.4 Figures Resistance (2, -6)

Here follows figures from a simulations in an environment with $r_0 = 2, r_1 = -6$.

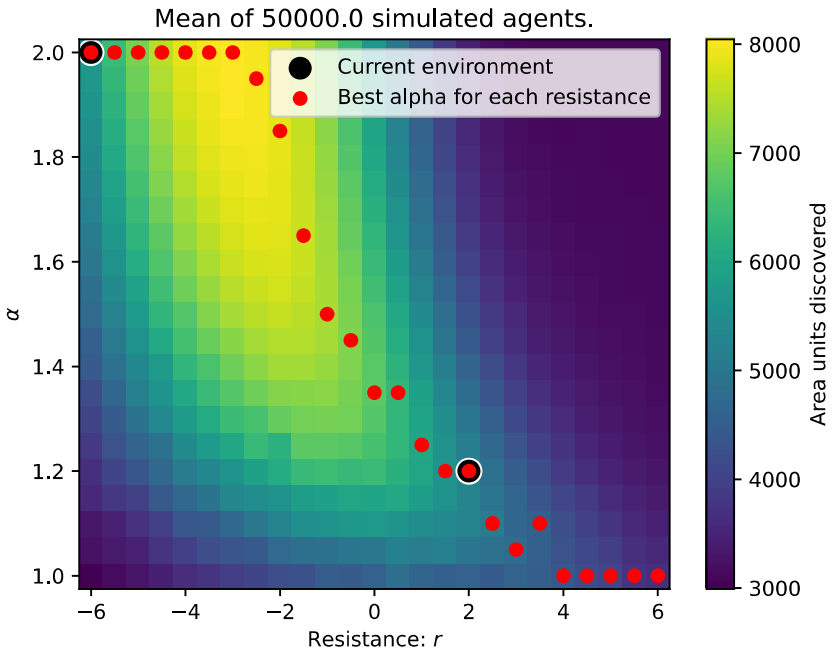


Figure B.16: A plane where color indicates search effectiveness for a given combination of resistance and alpha in the levy distribution. The alpha with the best result for each resistance is marked with red dots. The alphas and resistances that are used in this environment, $r_0 = 2, r_1 = -6$, are marked with a black border around the dot.

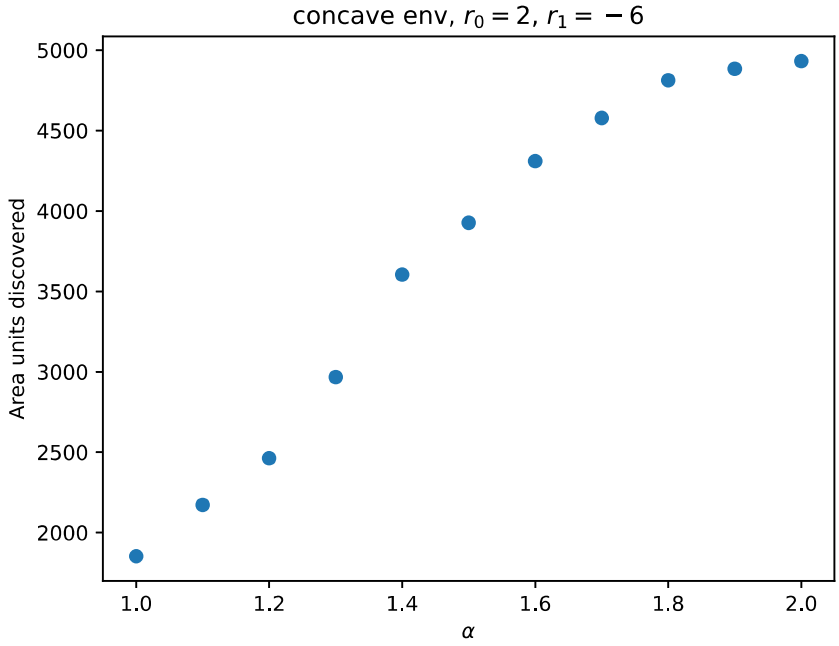


Figure B.17: Performance of each static alpha strategy in environment (2, -6).

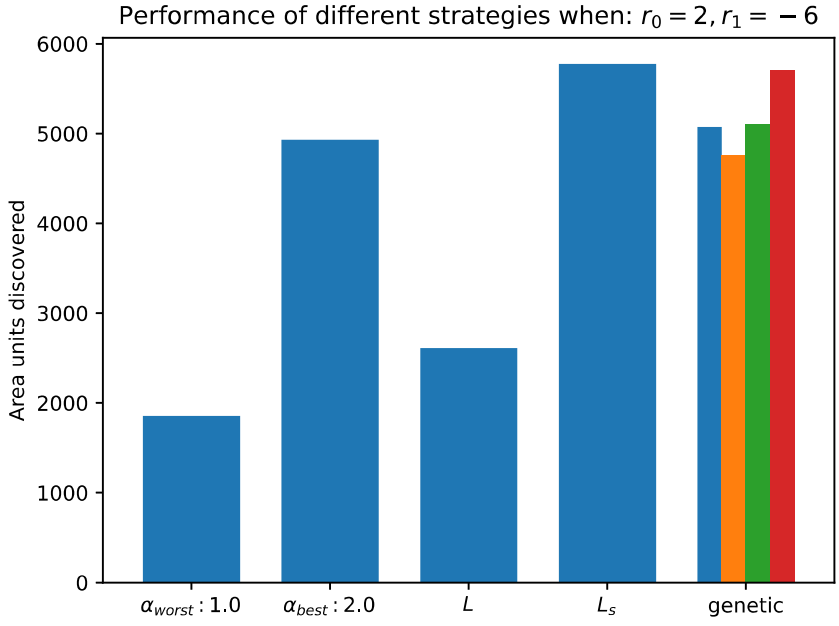


Figure B.18: Effectiveness of different search strategies can be seen in terms of mean area units discovered. α_{worst}: Worst alpha in the simple levy flight strategy. α_{best}: Best alpha in the simple levy flight strategy. local: Using the locally optimal alpha when drawing a new effort. local$_s$: Same as local but resets agent effort when moving into area with new resistance. genetic: Chooses effort through a neural network trained by a genetic algorithm.

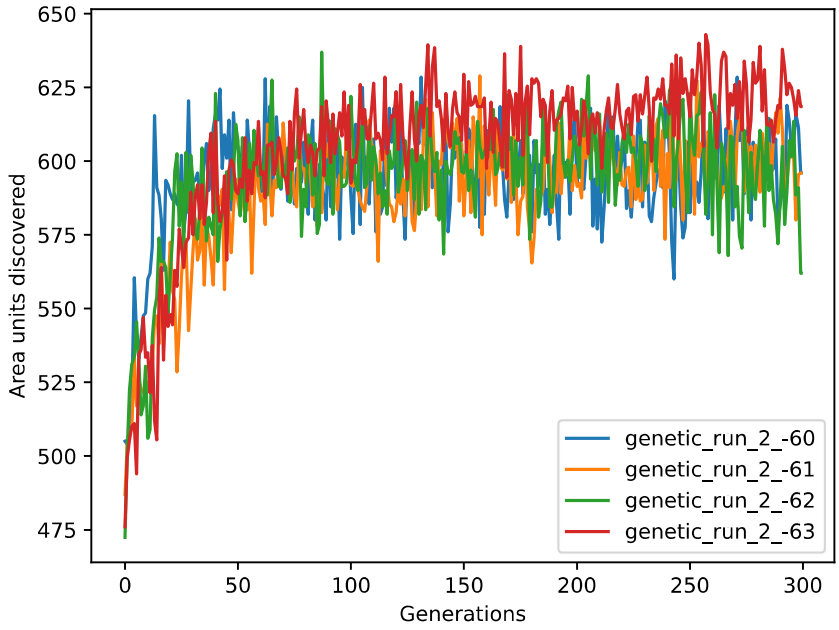


Figure B.19: Plot of the performance of the median agent in the population during genetic training.

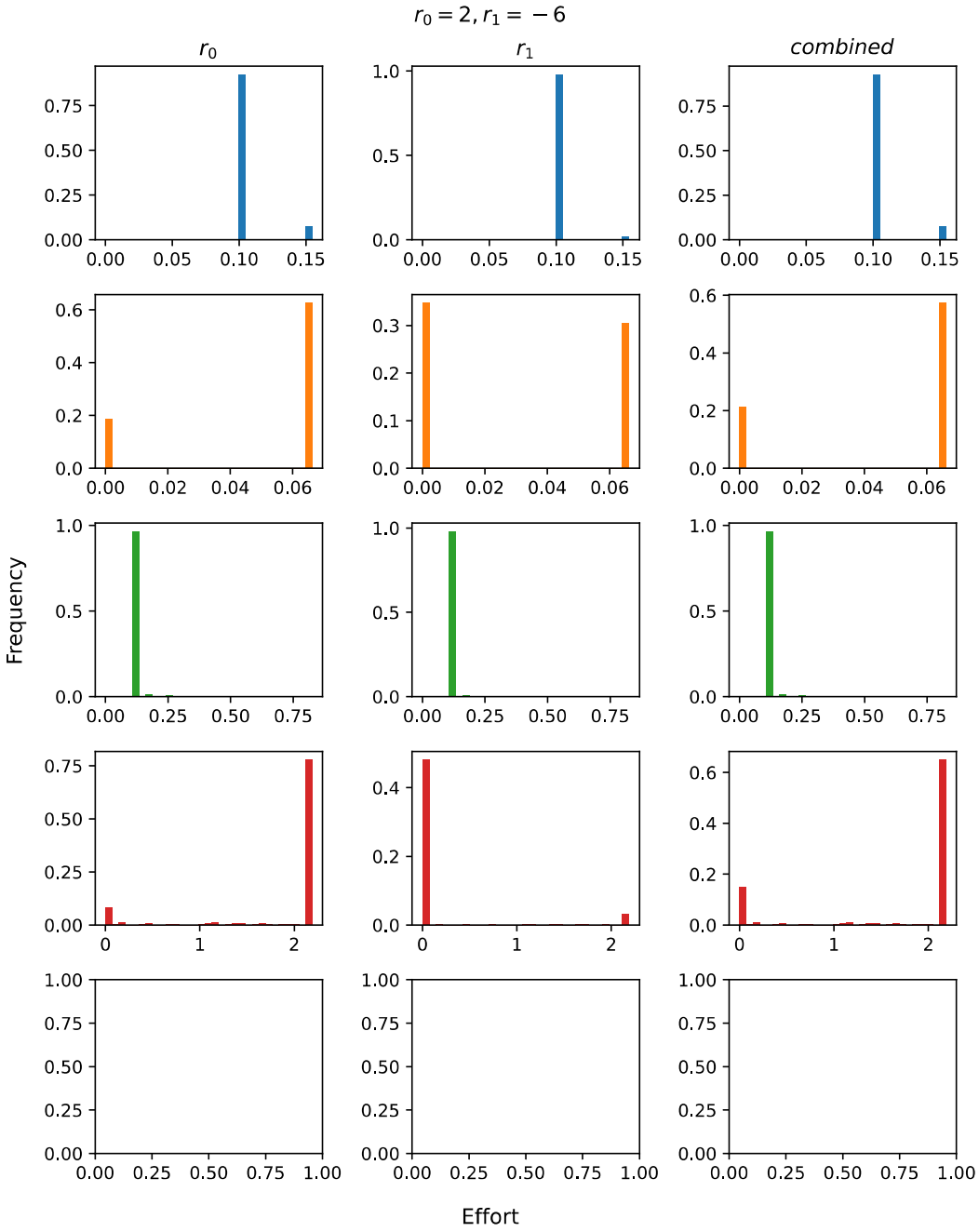


Figure B.20: Distribution of selected effort values for genitically trained agents. Each row contains data from a single agent. THe leftmost column is effort values sampled in resistance $r_0 = 3$, the middle in $r_1 = -4$, and the rightmost is from r_0 and r_1 combined.

B.5 Figures Resistance (3, -4)

Here follows figures from a simulations in an environment with $r_0 = 3, r_1 = -4$.

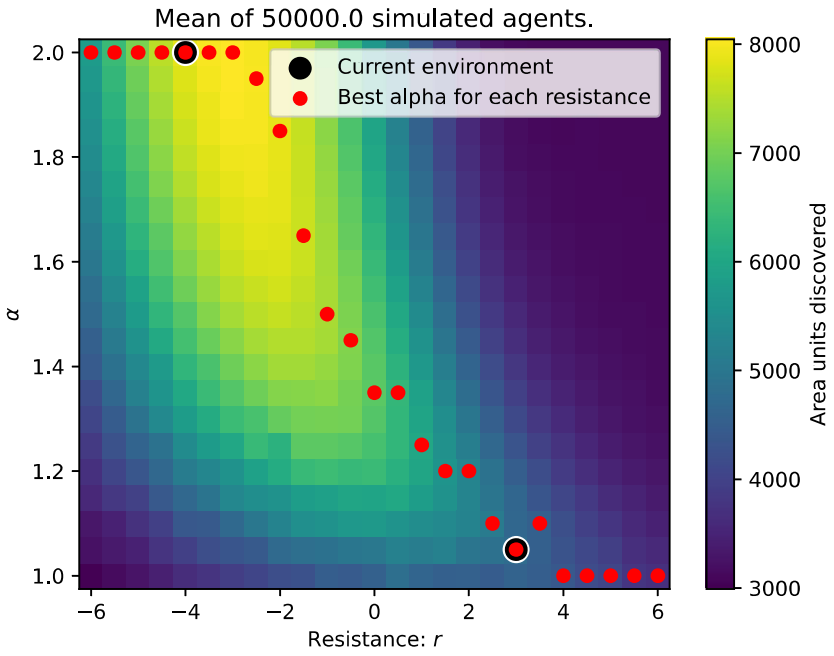


Figure B.21: A plane where color indicates search effectiveness for a given combination of resistance and alpha in the levy distribution. The alpha with the best result for each resistance is marked with red dots. The alphas and resistances that are used in this environment, $r_0 = 3, r_1 = -4$, are marked with a black border around the dot.

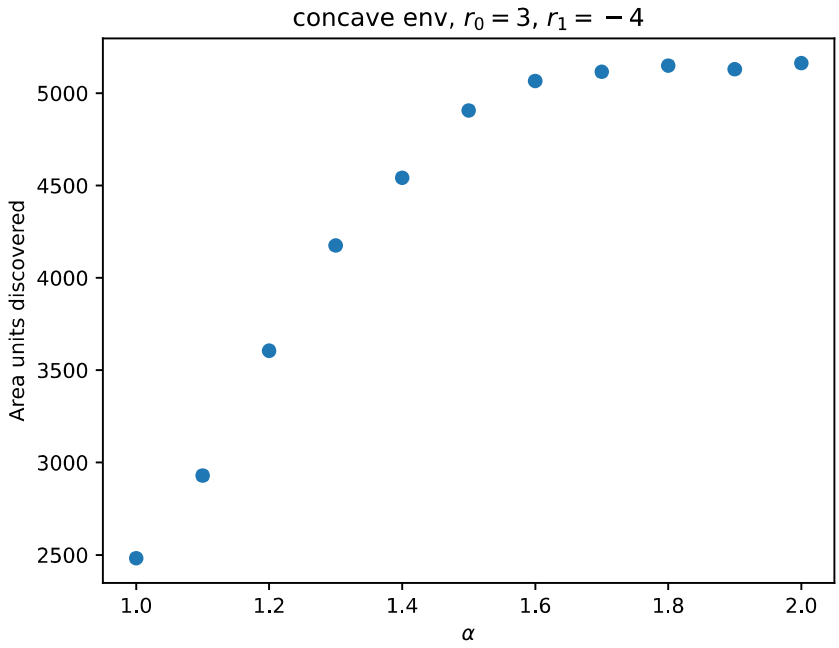


Figure B.22: Performance of each static alpha strategy in environment (3, -4).

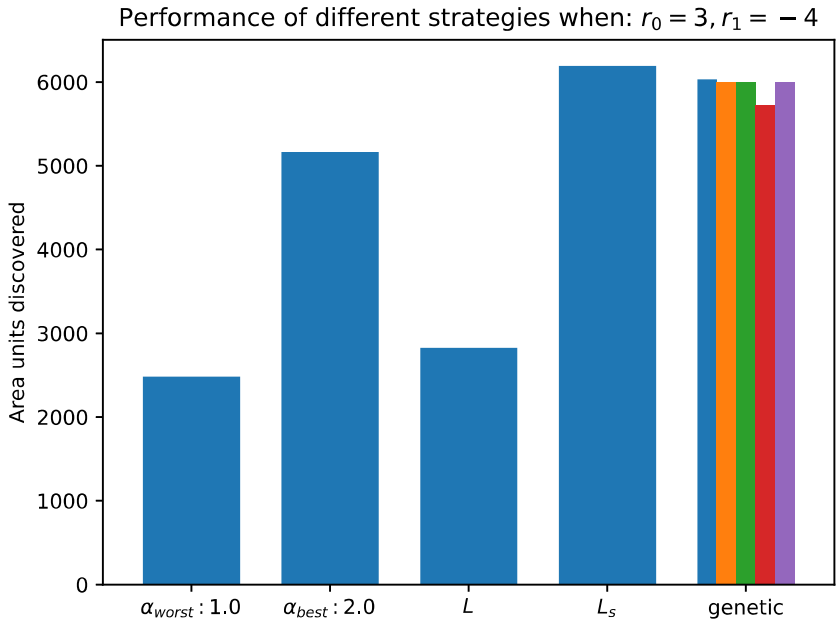


Figure B.23: Effectiveness of different search strategies can be seen in terms of mean area units discovered. α_{worst}: Worst alpha in the simple levy flight strategy. α_{best}: Best alpha in the simple levy flight strategy. local: Using the locally optimal alpha when drawing a new effort. local$_s$: Same as local but resets agent effort when moving into area with new resistance. genetic: Chooses effort through a neural network trained by a genetic algorithm.

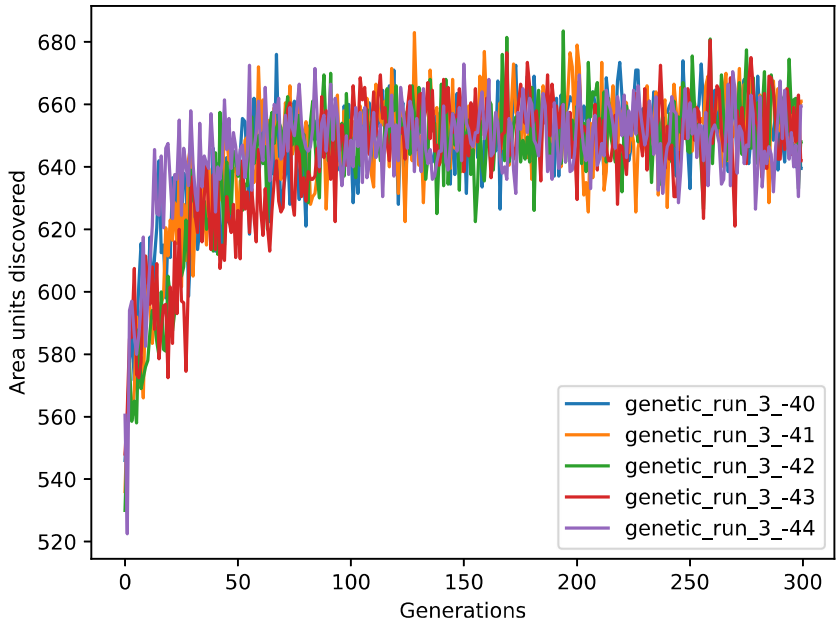


Figure B.24: Plot of the performance of the median agent in the population during genetic training.

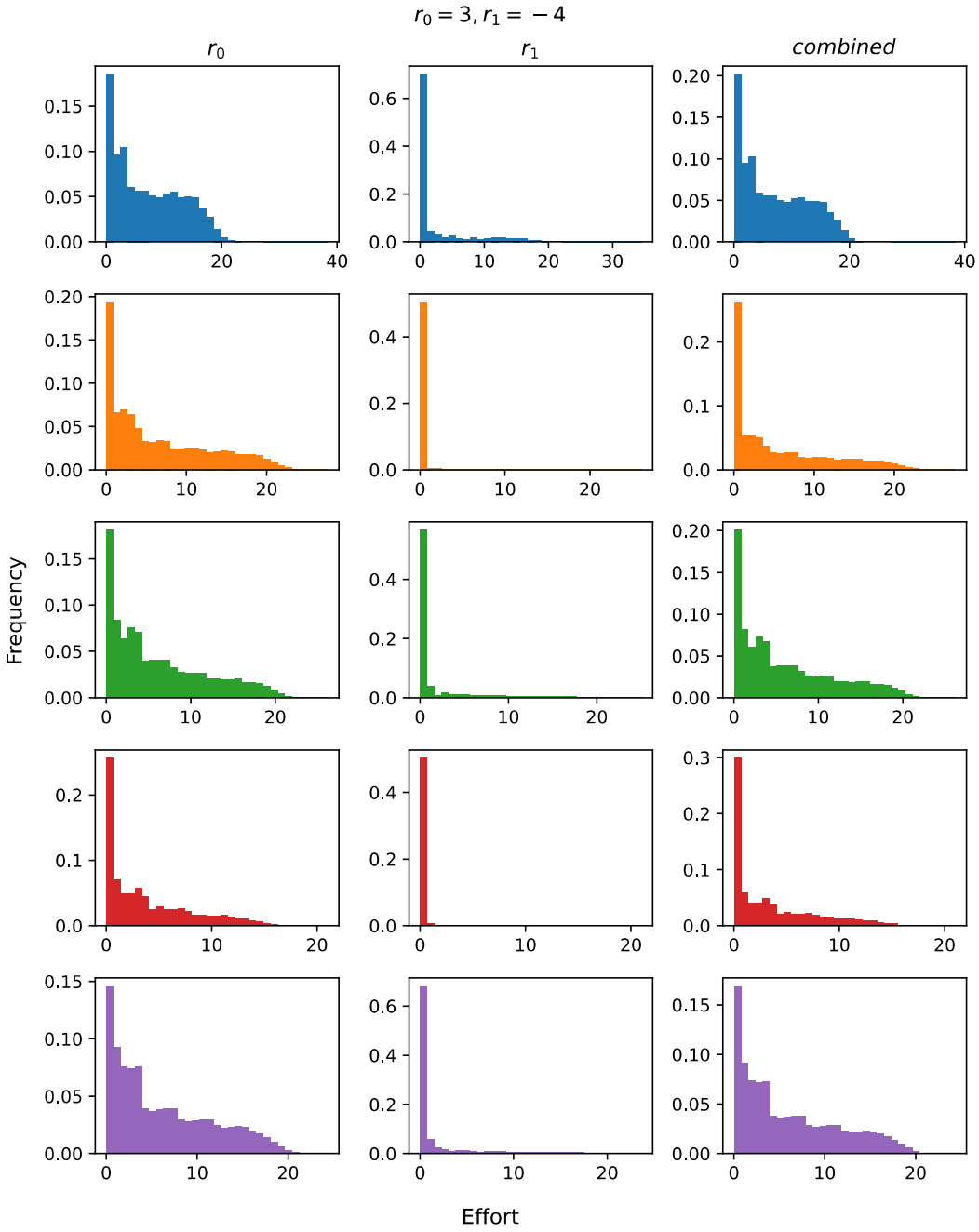


Figure B.25: Distribution of selected effort values for genitically trained agents. Each row contains data from a single agent. THe leftmost column is effort values sampled in resistance $r_0 = 3$, the middle in $r_1 = -4$, and the rightmost is from r_0 and r_1 combined.

B.6 Figures Resistance (4, -3)

Here follows figures from a simulations in an environment with $r_0 = 4, r_1 = -3$.

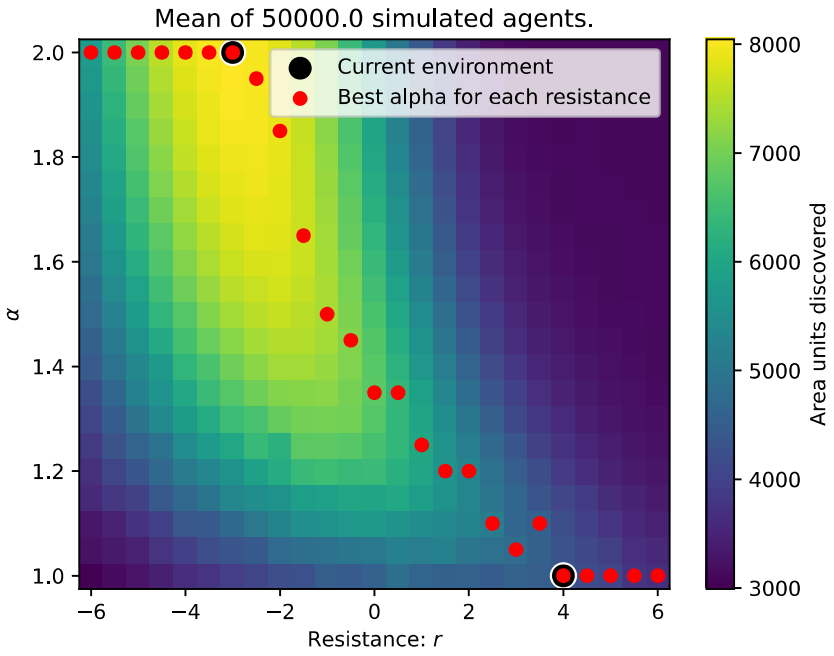


Figure B.26: A plane where color indicates search effectiveness for a given combination of resistance and alpha in the levy distribution. The alpha with the best result for each resistance is marked with red dots. The alphas and resistances that are used in this environment, $r_0 = 4, r_1 = -3$, are marked with a black border around the dot.

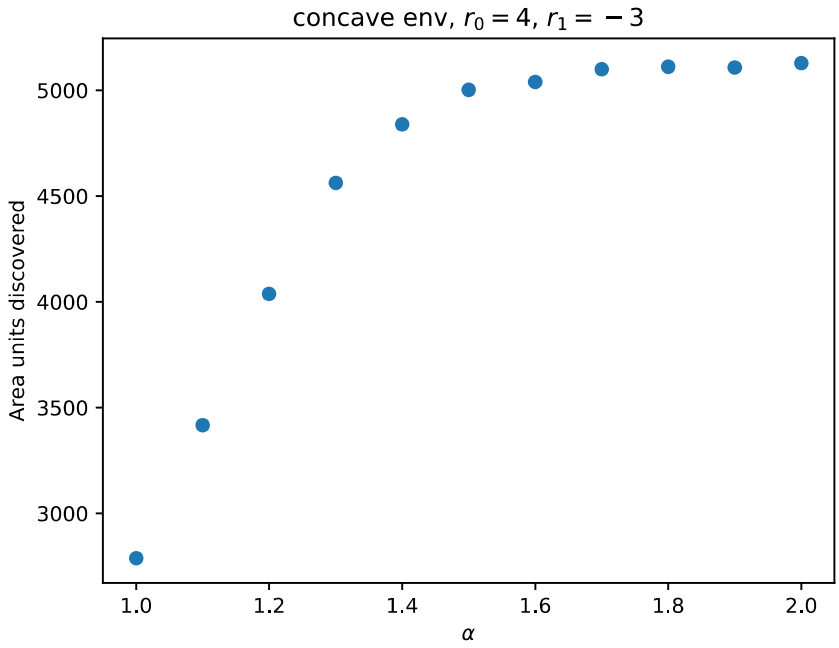


Figure B.27: Performance of each static alpha strategy in environment (4, -3).

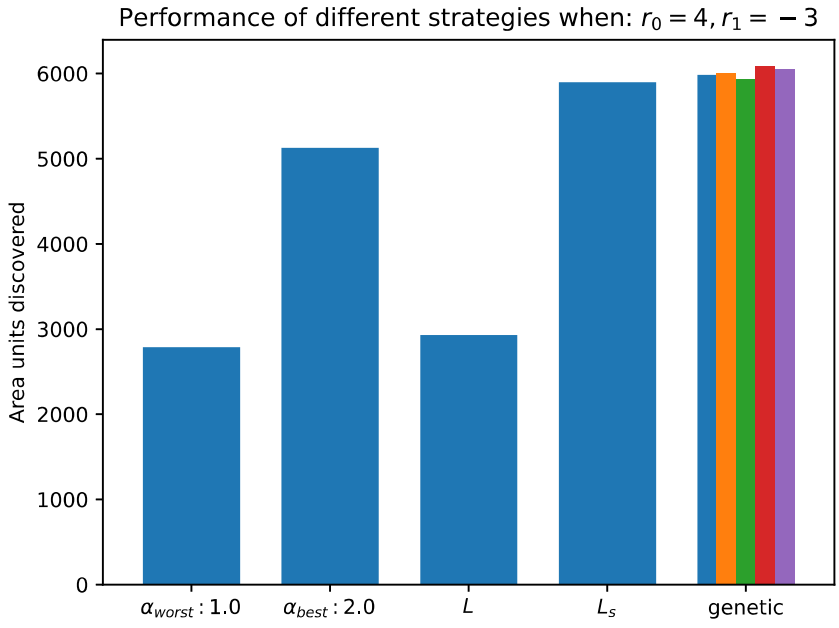


Figure B.28: Effectiveness of different search strategies can be seen in terms of mean area units discovered. α_{worst}: Worst alpha in the simple levy flight strategy. α_{best}: Best alpha in the simple levy flight strategy. local: Using the locally optimal alpha when drawing a new effort. local$_s$: Same as local but resets agent effort when moving into area with new resistance. genetic: Chooses effort through a neural network trained by a genetic algorithm.

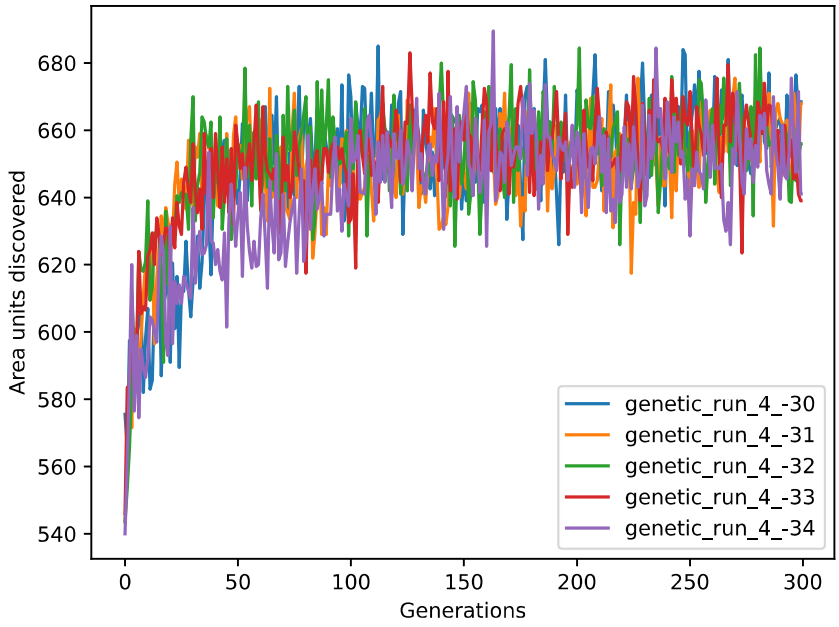


Figure B.29: Plot of the performance of the median agent in the population during genetic training.

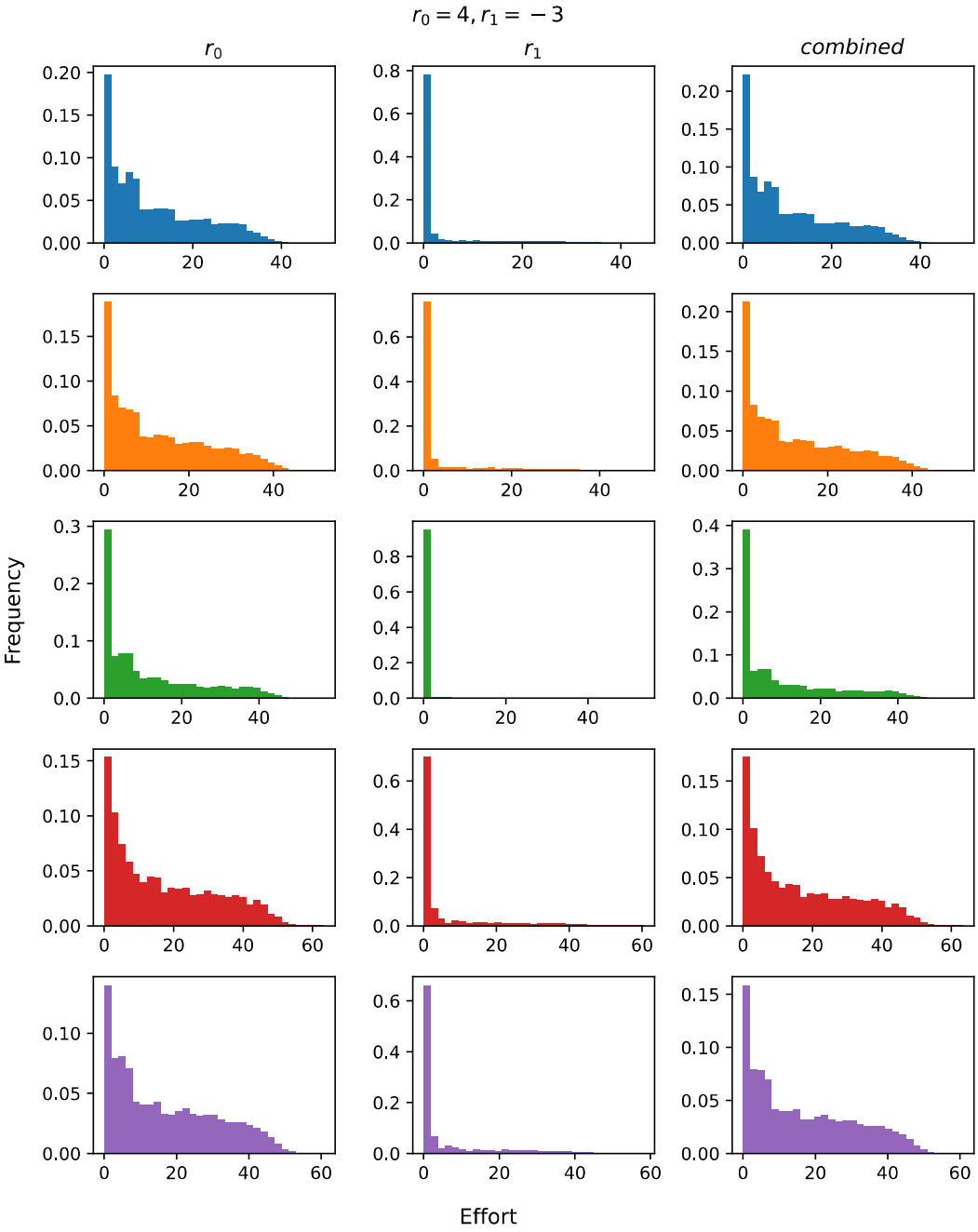


Figure B.30: Distribution of selected effort values for genitically trained agents. Each row contains data from a single agent. THe leftmost column is effort values sampled in resistance $r_0 = 3$, the middle in $r_1 = -4$, and the rightmost is from r_0 and r_1 combined.

B.7 Figures Resistance (6, -6)

Here follows figures from a simulations in an environment with $r_0 = 6, r_1 = -6$.

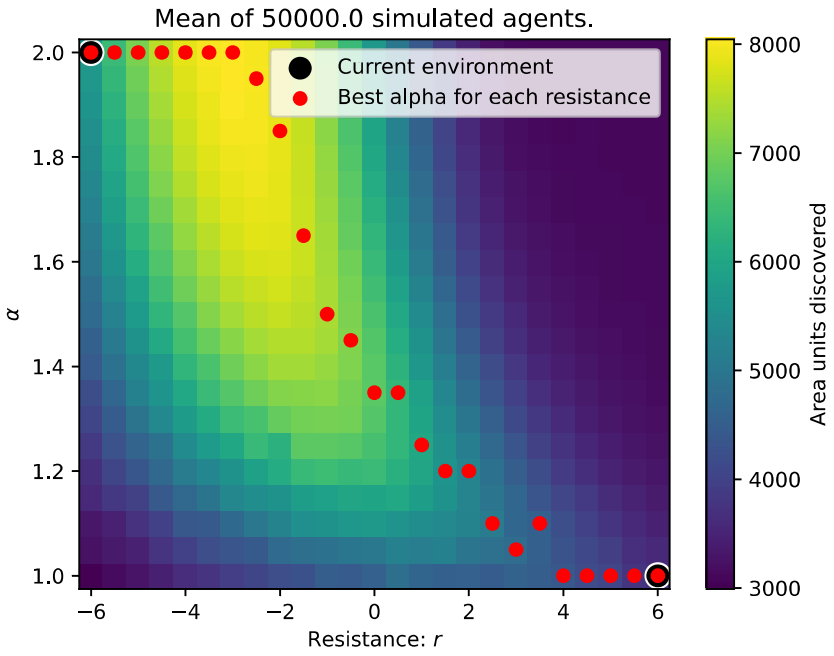


Figure B.31: A plane where color indicates search effectiveness for a given combination of resistance and alpha in the levy distribution. The alpha with the best result for each resistance is marked with red dots. The alphas and resistances that are used in this environment, $r_0 = 6, r_1 = -6$, are marked with a black border around the dot.

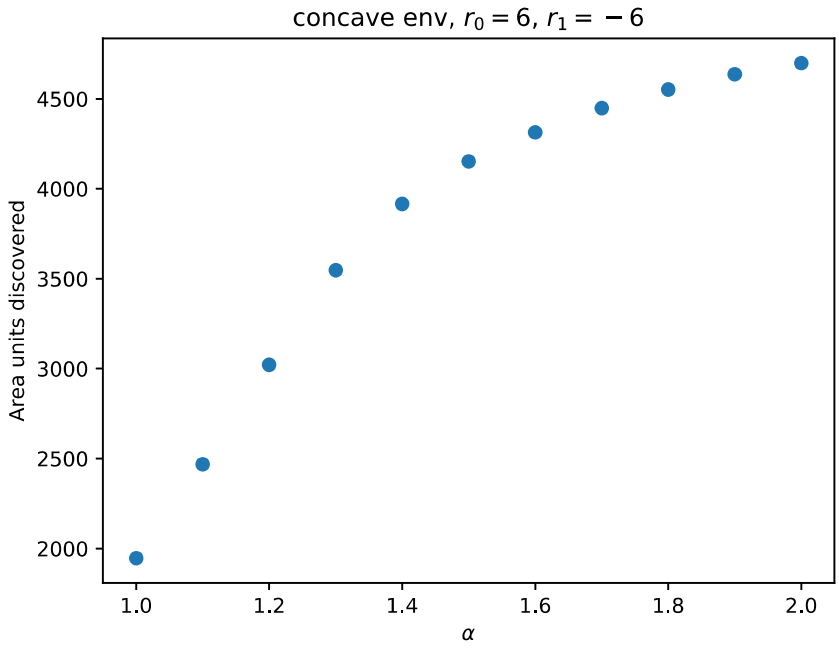


Figure B.32: Performance of each static alpha strategy in environment (6, -6).

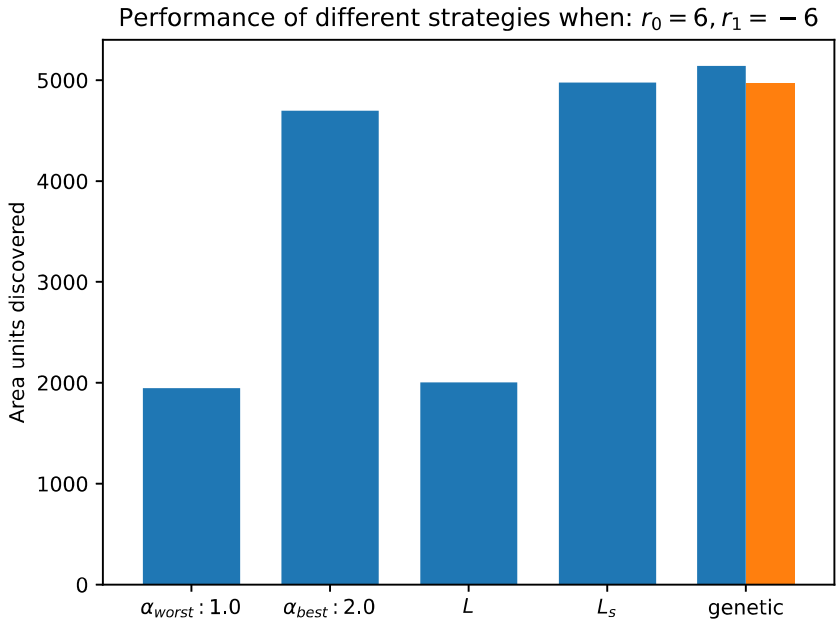


Figure B.33: Effectiveness of different search strategies can be seen in terms of mean area units discovered. α_{worst}: Worst alpha in the simple levy flight strategy. α_{best}: Best alpha in the simple levy flight strategy. local: Using the locally optimal alpha when drawing a new effort. local$_s$: Same as local but resets agent effort when moving into area with new resistance. genetic: Chooses effort through a neural network trained by a genetic algorithm.

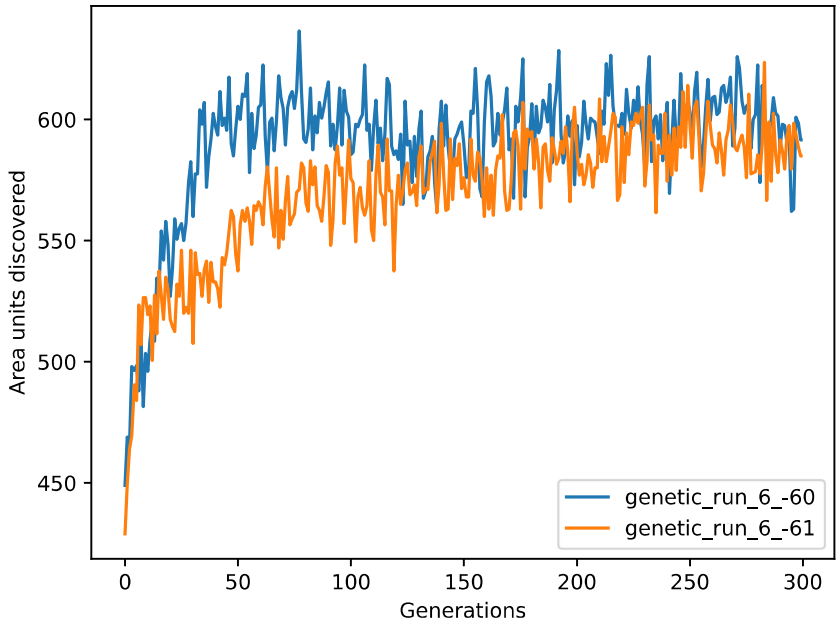


Figure B.34: Plot of the performance of the median agent in the population during genetic training.

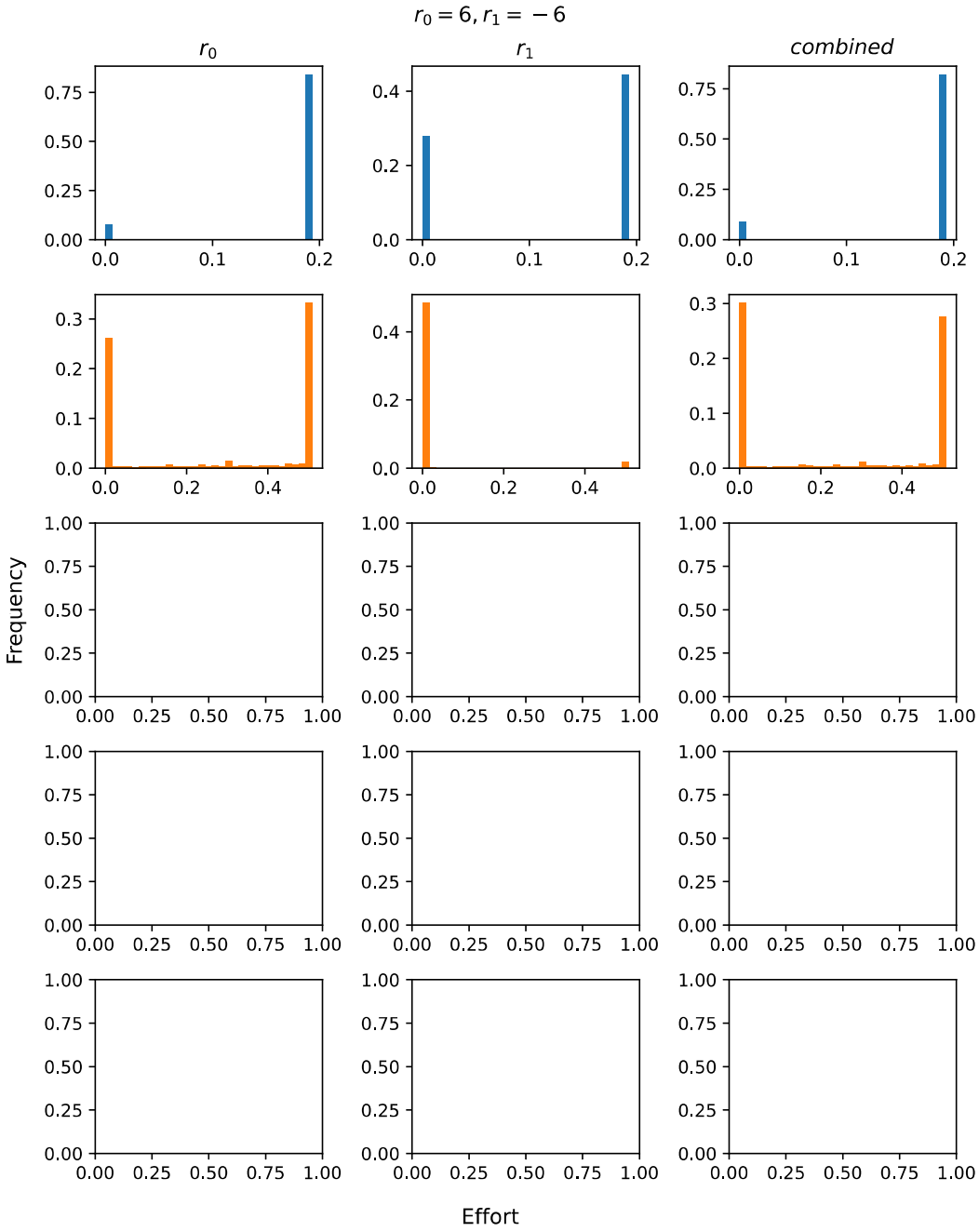


Figure B.35: Distribution of selected effort values for genitically trained agents. Each row contains data from a single agent. THe leftmost column is effort values sampled in resistance $r_0 = 3$, the middle in $r_1 = -4$, and the rightmost is from r_0 and r_1 combined.

B.8 Figures Resistance (6, -2)

Here follows figures from a simulations in an environment with $r_0 = 6, r_1 = -2$.

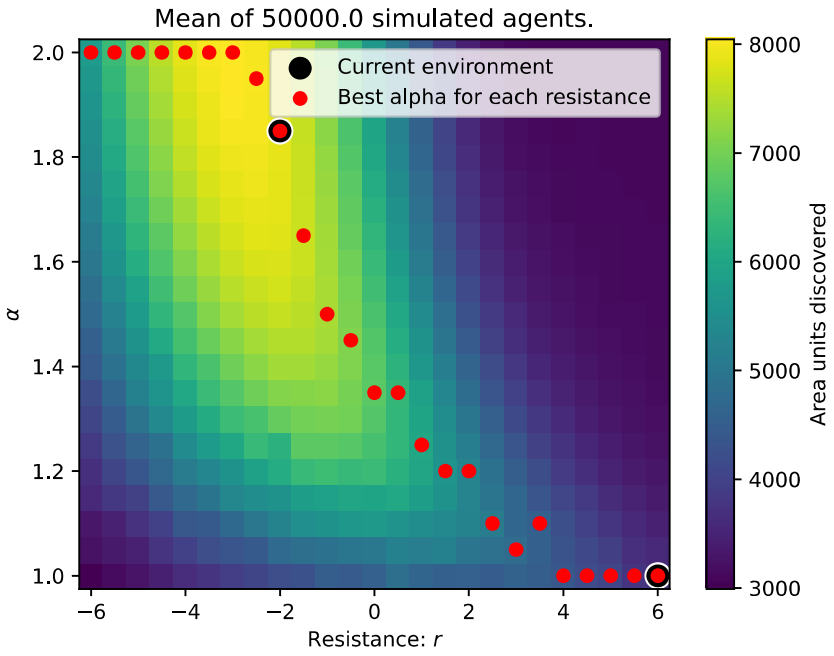


Figure B.36: A plane where color indicates search effectiveness for a given combination of resistance and alpha in the levy distribution. The alpha with the best result for each resistance is marked with red dots. The alphas and resistances that are used in this environment, $r_0 = 6, r_1 = -2$, are marked with a black border around the dot.

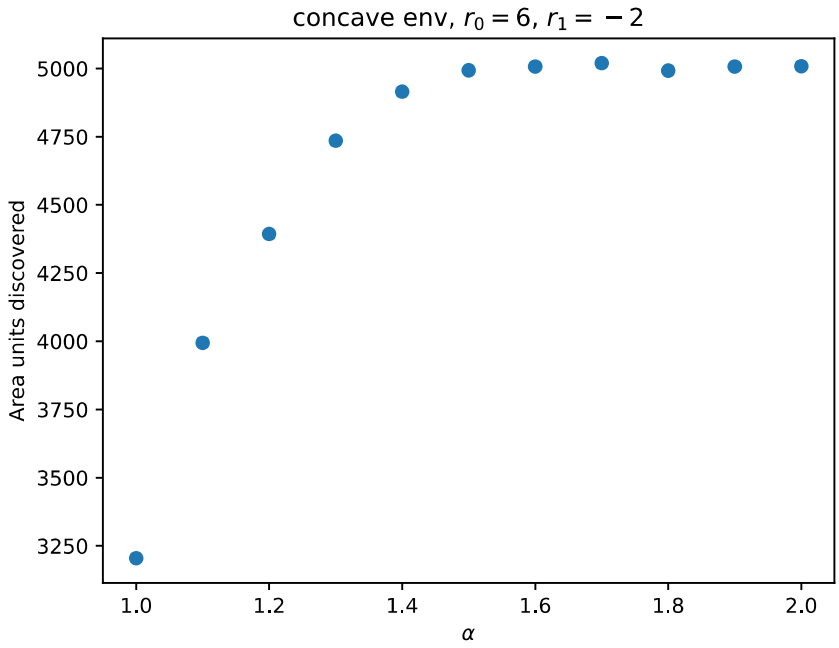


Figure B.37: Performance of each static alpha strategy in environment (6, -2).

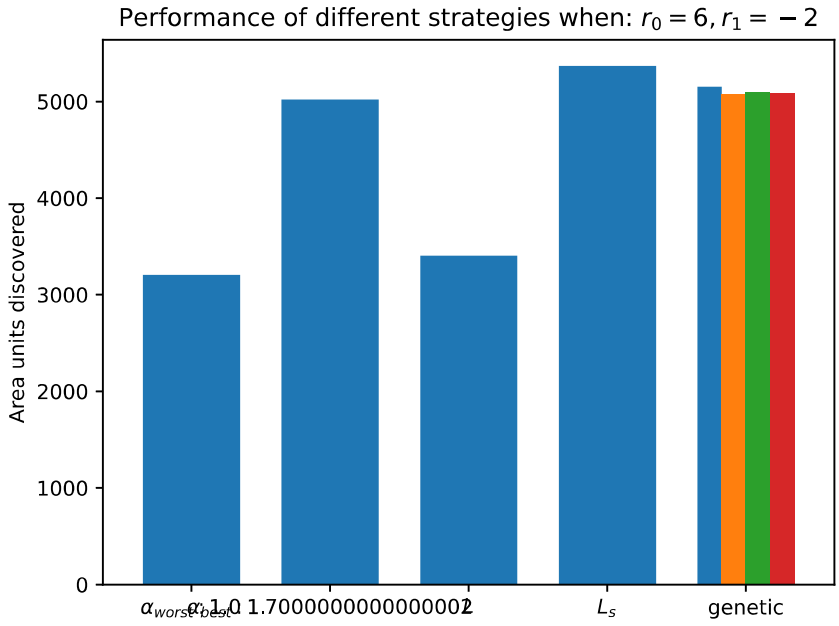


Figure B.38: Effectiveness of different search strategies can be seen in terms of mean area units discovered. α_{worst}: Worst alpha in the simple levy flight strategy. α_{best}: Best alpha in the simple levy flight strategy. local: Using the locally optimal alpha when drawing a new effort. local$_s$: Same as local but resets agent effort when moving into area with new resistance. genetic: Chooses effort through a neural network trained by a genetic algorithm.

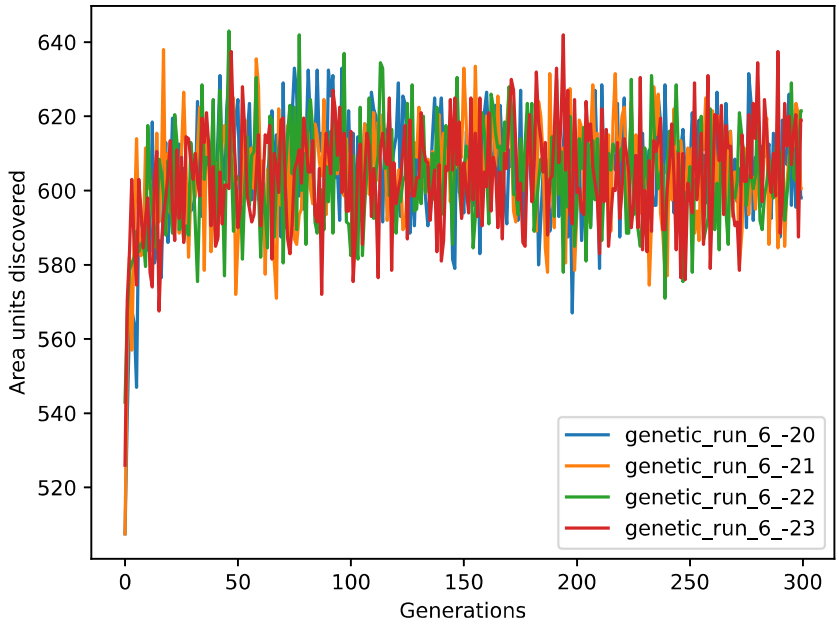


Figure B.39: Plot of the performance of the median agent in the population during genetic training.

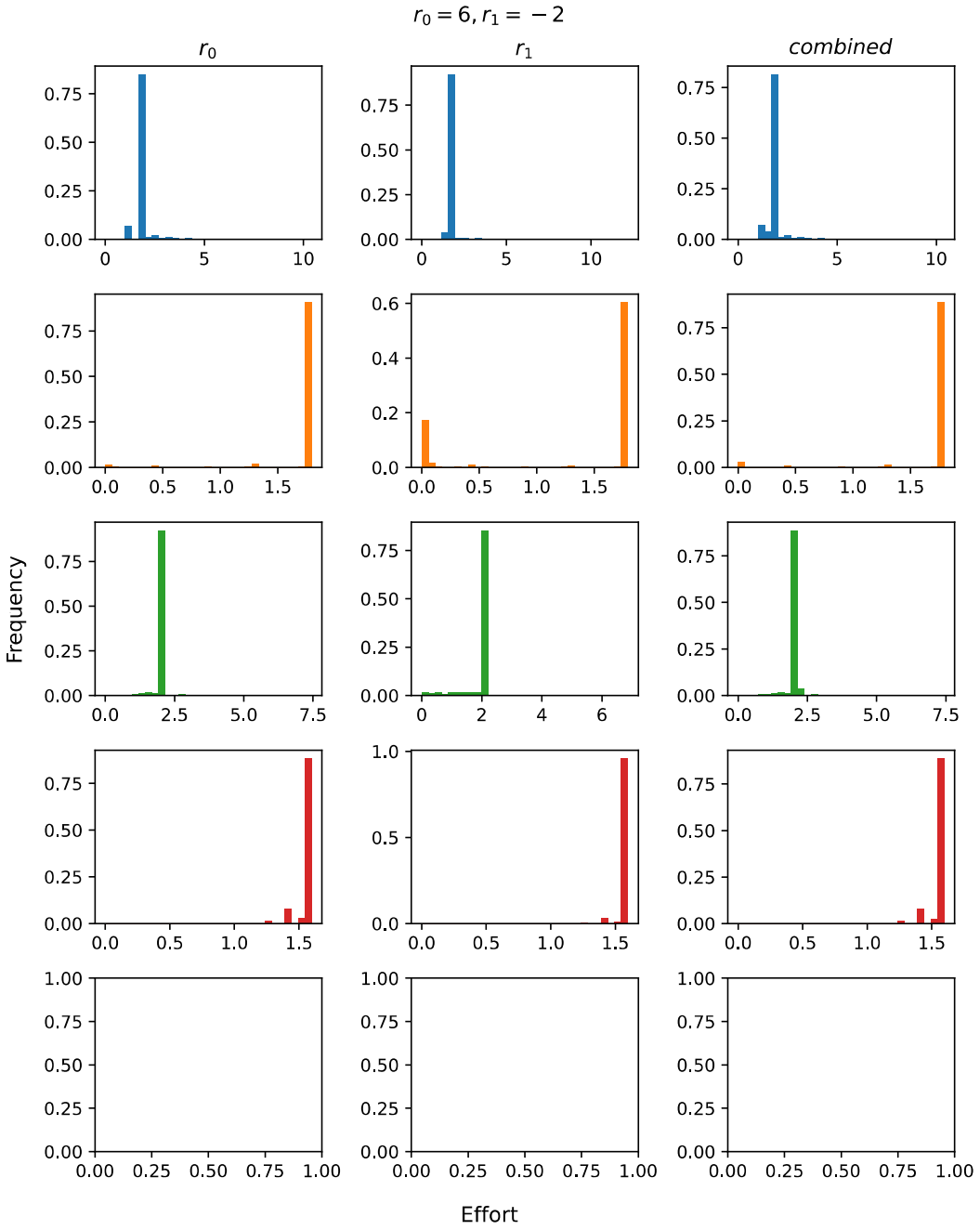


Figure B.40: Distribution of selected effort values for genitically trained agents. Each row contains data from a single agent. THe leftmost column is effort values sampled in resistance $r_0 = 3$, the middle in $r_1 = -4$, and the rightmost is from r_0 and r_1 combined.

B.9 Figures Resistance (6, 0)

Here follows figures from a simulations in an environment with $r_0 = 6, r_1 = 0$.

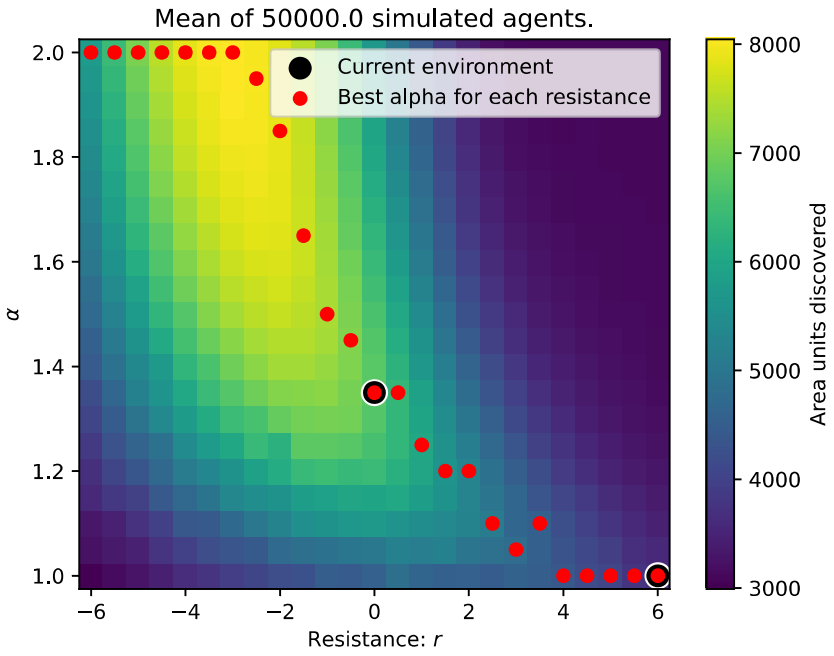


Figure B.41: A plane where color indicates search effectiveness for a given combination of resistance and alpha in the levy distribution. The alpha with the best result for each resistance is marked with red dots. The alphas and resistances that are used in this environment, $r_0 = 6, r_1 = 0$, are marked with a black border around the dot.

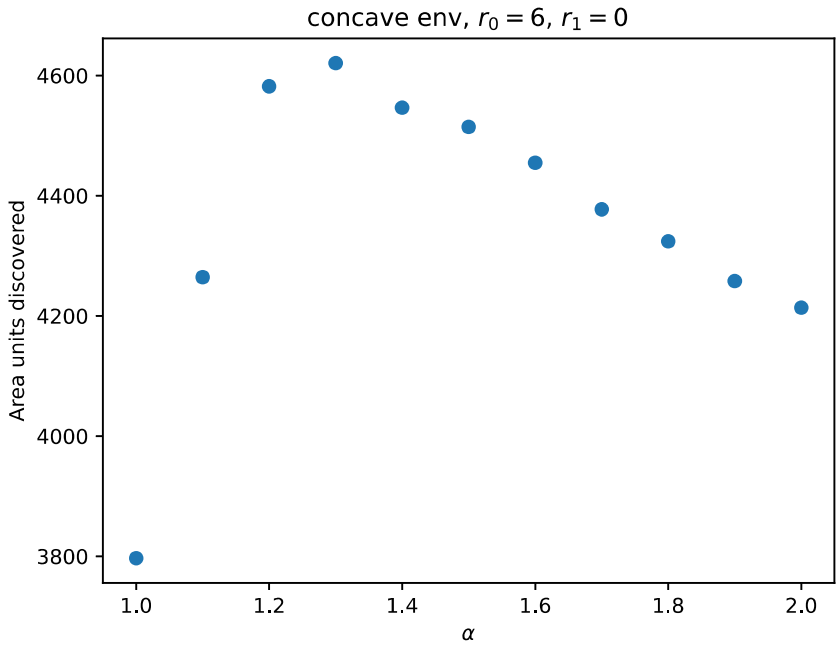


Figure B.42: Performance of each static alpha strategy in environment (6, 0).

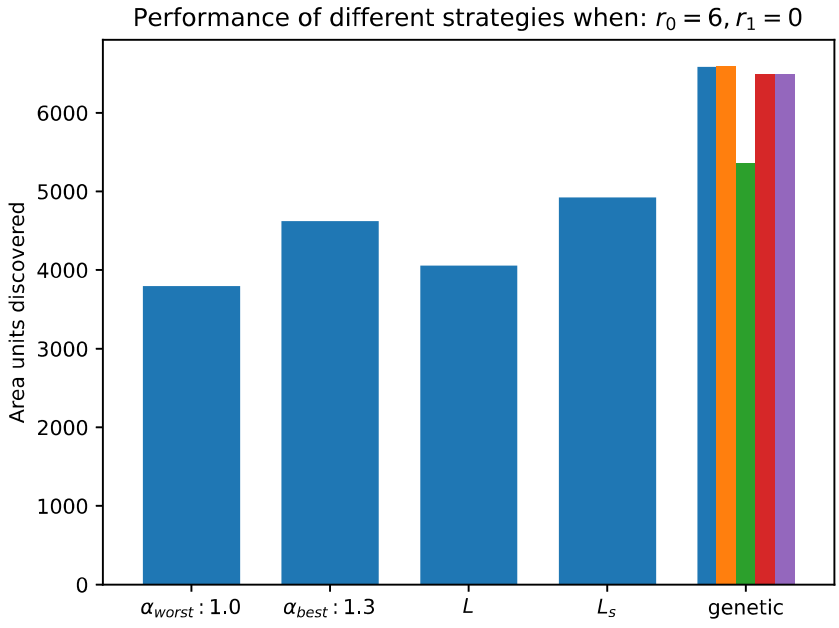


Figure B.43: Effectiveness of different search strategies can be seen in terms of mean area units discovered. α_{worst}: Worst alpha in the simple levy flight strategy. α_{best}: Best alpha in the simple levy flight strategy. local: Using the locally optimal alpha when drawing a new effort. local$_s$: Same as local but resets agent effort when moving into area with new resistance. genetic: Chooses effort through a neural network trained by a genetic algorithm.

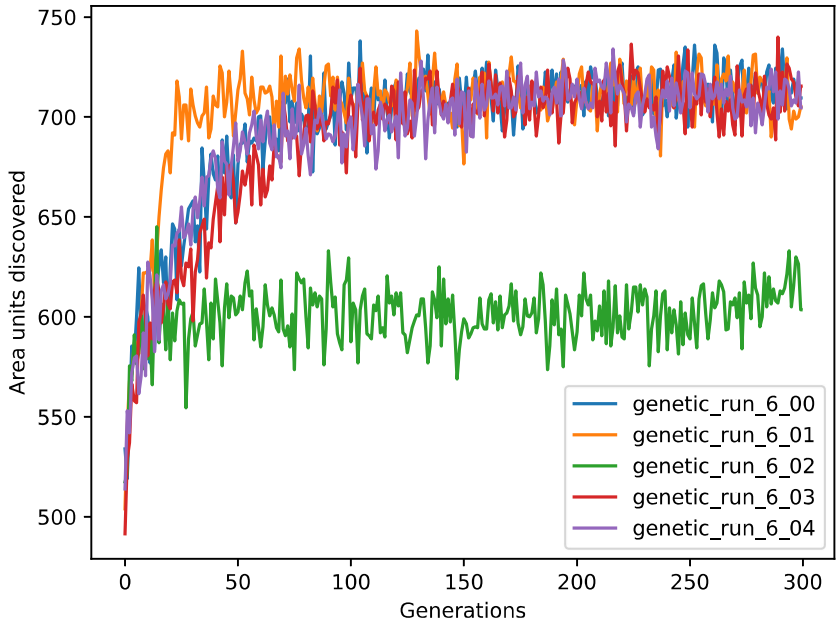


Figure B.44: Plot of the performance of the median agent in the population during genetic training.

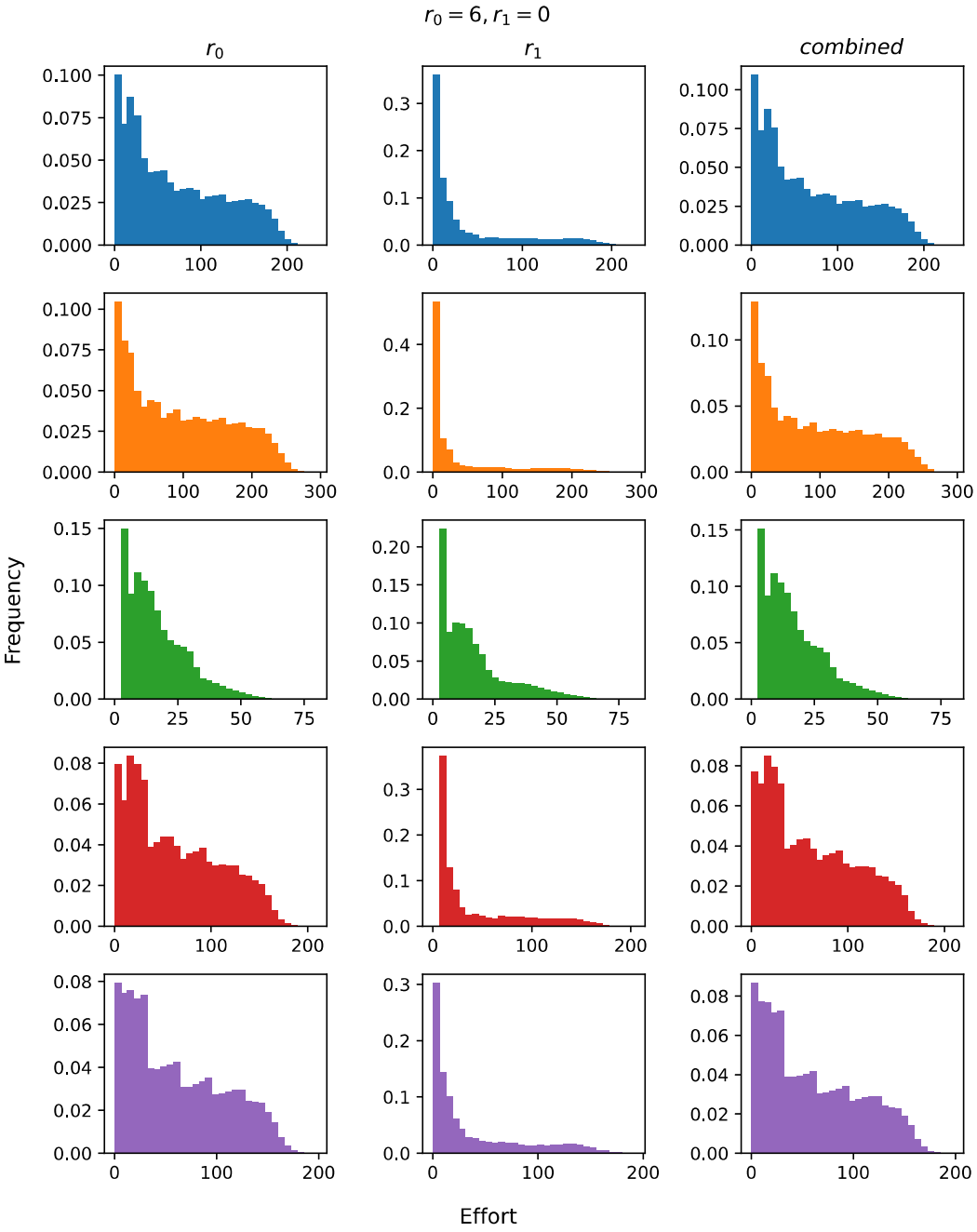


Figure B.45: Distribution of selected effort values for genitically trained agents. Each row contains data from a single agent. THe leftmost column is effort values sampled in resistance $r_0 = 3$, the middle in $r_1 = -4$, and the rightmost is from r_0 and r_1 combined.

B.10 Figures Resistance (6, 2)

Here follows figures from a simulations in an environment with $r_0 = 6, r_1 = 2$.

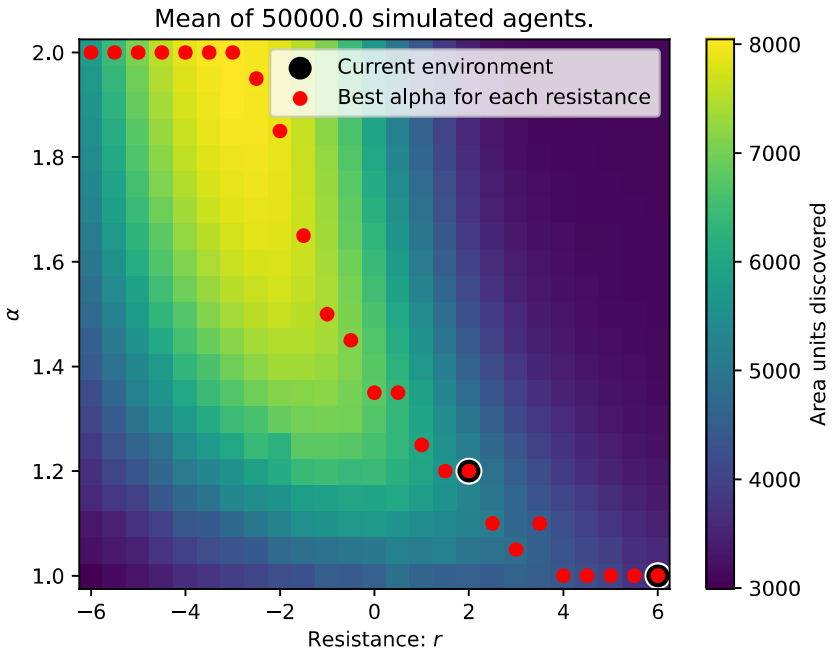


Figure B.46: A plane where color indicates search effectiveness for a given combination of resistance and alpha in the levy distribution. The alpha with the best result for each resistance is marked with red dots. The alphas and resistances that are used in this environment, $r_0 = 6, r_1 = 2$, are marked with a black border around the dot.

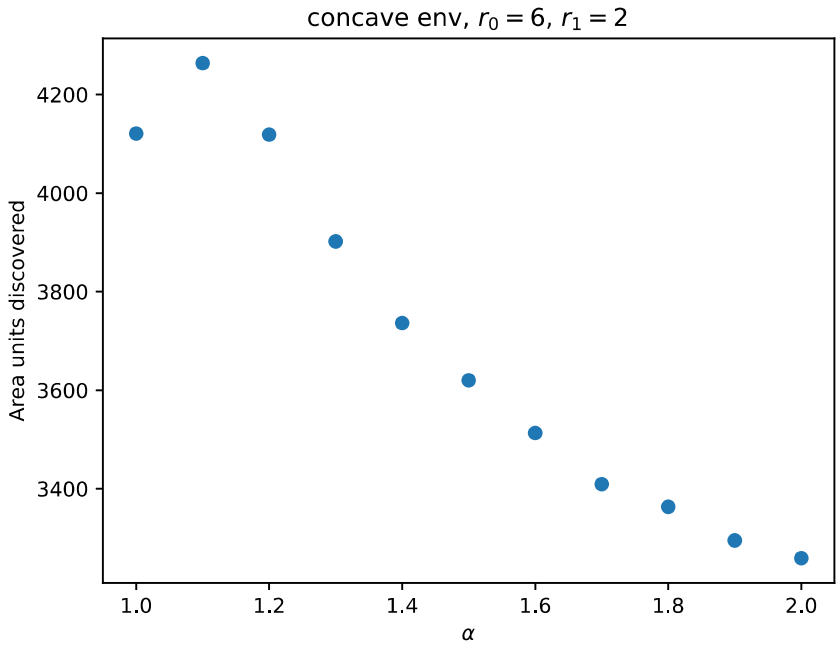


Figure B.47: Performance of each static alpha strategy in environment (6, 2).

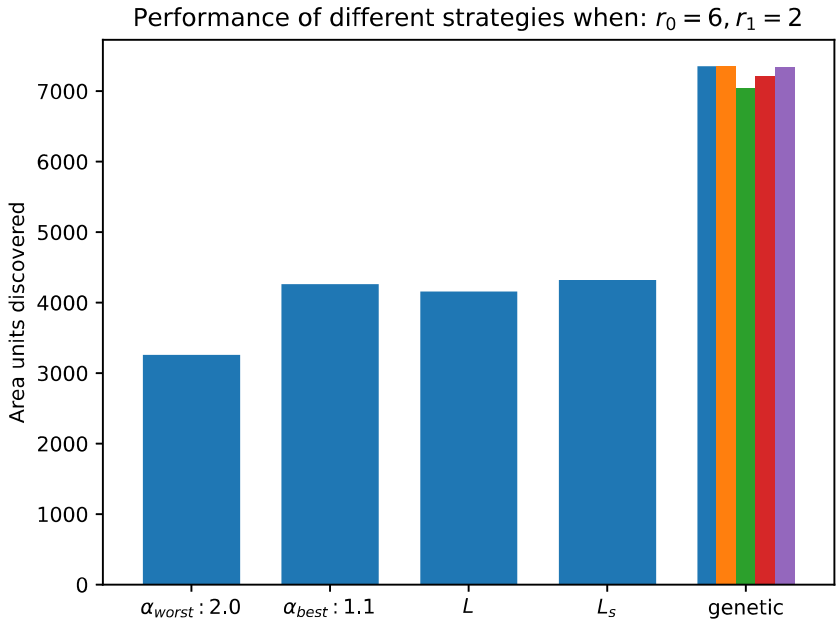


Figure B.48: Effectiveness of different search strategies can be seen in terms of mean area units discovered. α_{worst}: Worst alpha in the simple levy flight strategy. α_{best}: Best alpha in the simple levy flight strategy. local: Using the locally optimal alpha when drawing a new effort. local$_s$: Same as local but resets agent effort when moving into area with new resistance. genetic: Chooses effort through a neural network trained by a genetic algorithm.

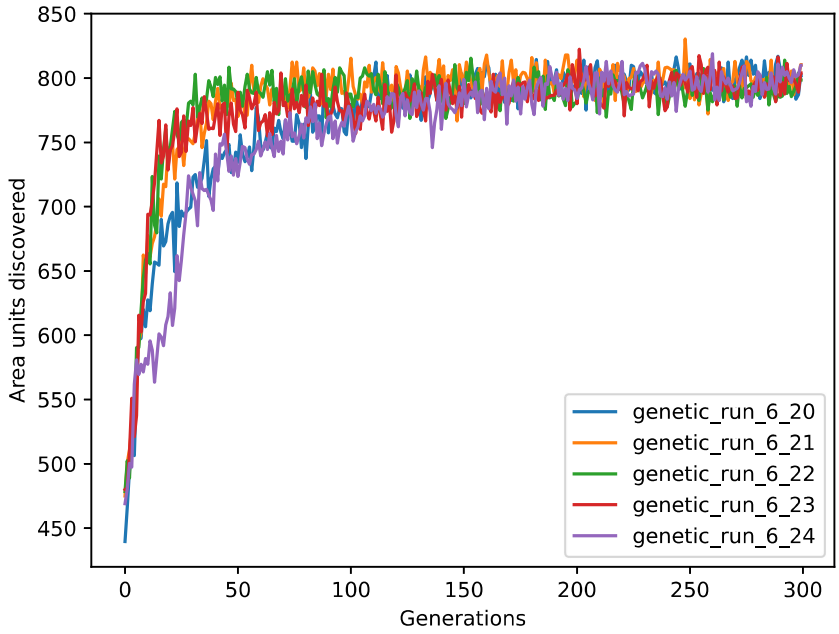


Figure B.49: Plot of the performance of the median agent in the population during genetic training.

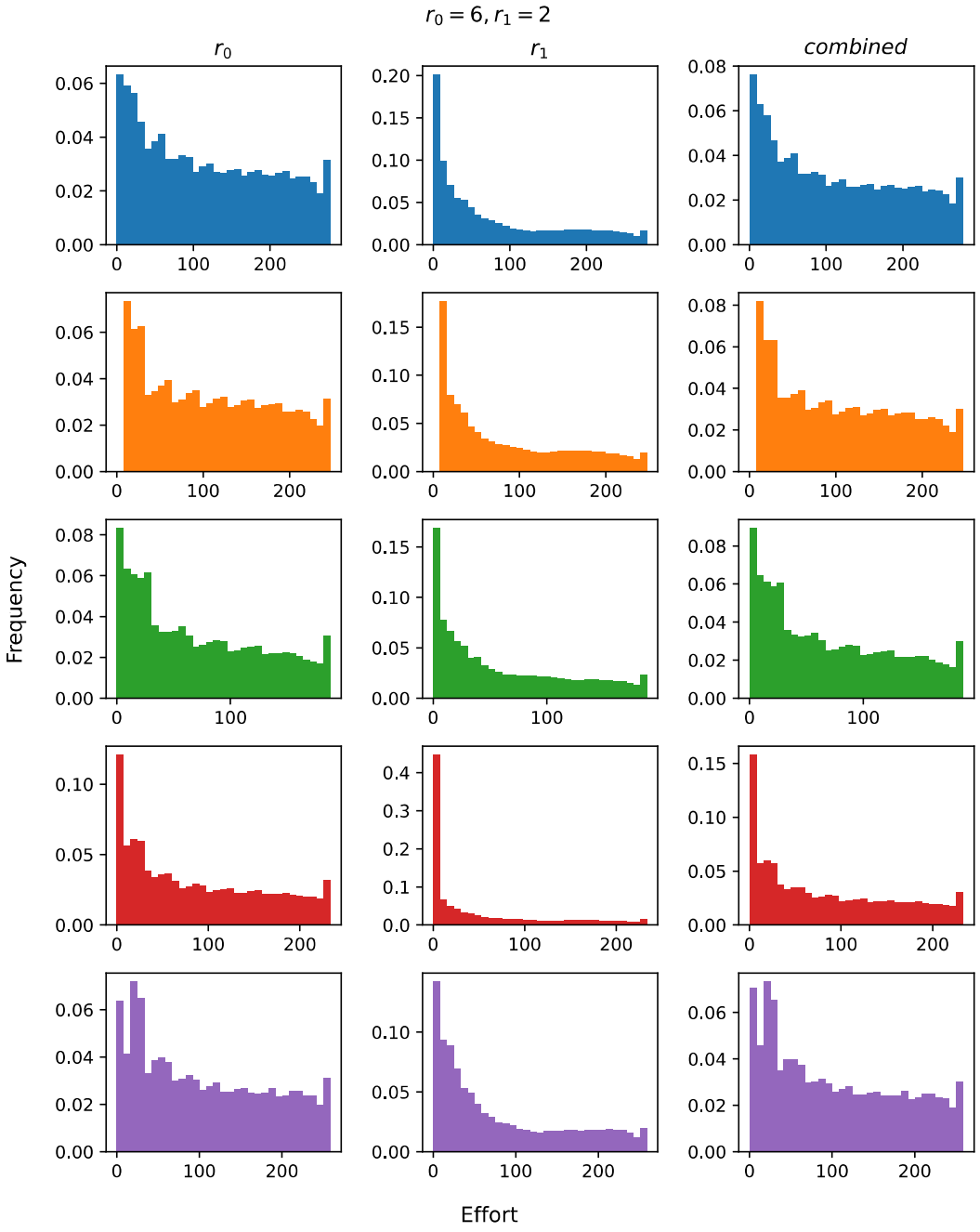


Figure B.50: Distribution of selected effort values for genitically trained agents. Each row contains data from a single agent. THe leftmost column is effort values sampled in resistance $r_0 = 3$, the middle in $r_1 = -4$, and the rightmost is from r_0 and r_1 combined.

B.11 Figures Resistance (6, 4)

Here follows figures from a simulations in an environment with $r_0 = 6, r_1 = 4$.

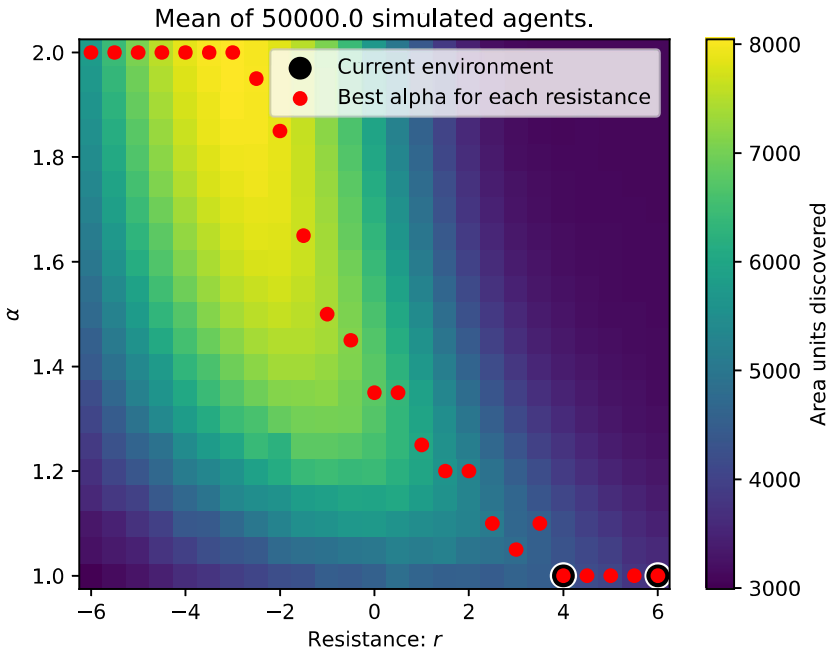


Figure B.51: A plane where color indicates search effectiveness for a given combination of resistance and alpha in the levy distribution. The alpha with the best result for each resistance is marked with red dots. The alphas and resistances that are used in this environment, $r_0 = 6, r_1 = 4$, are marked with a black border around the dot.

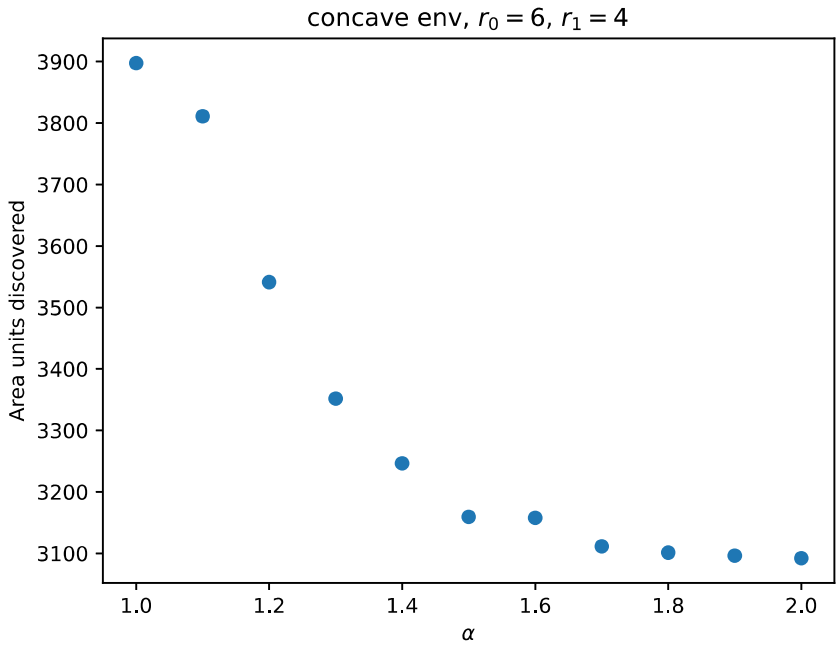


Figure B.52: Performance of each static alpha strategy in environment (6, 4).

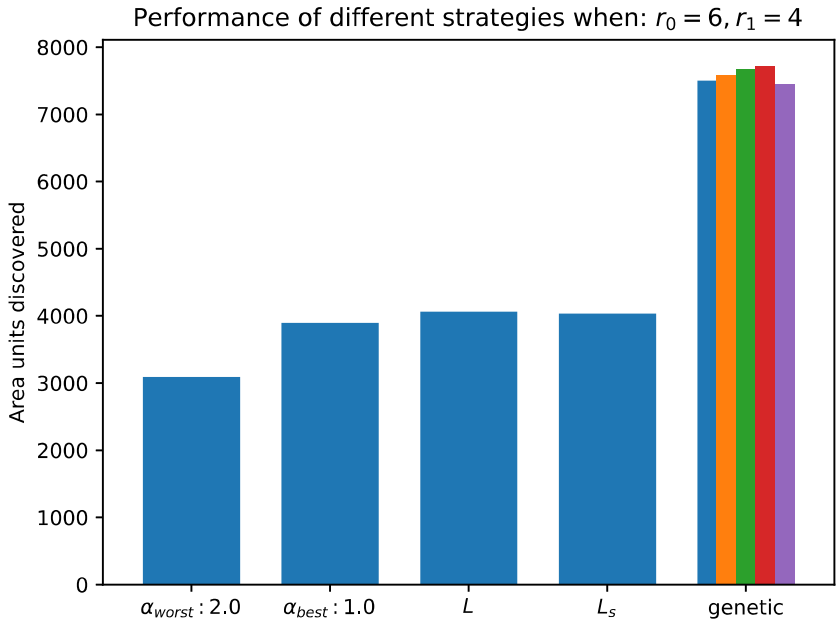


Figure B.53: Effectiveness of different search strategies can be seen in terms of mean area units discovered. α_{worst}: Worst alpha in the simple levy flight strategy. α_{best}: Best alpha in the simple levy flight strategy. local: Using the locally optimal alpha when drawing a new effort. local$_s$: Same as local but resets agent effort when moving into area with new resistance. genetic: Chooses effort through a neural network trained by a genetic algorithm.

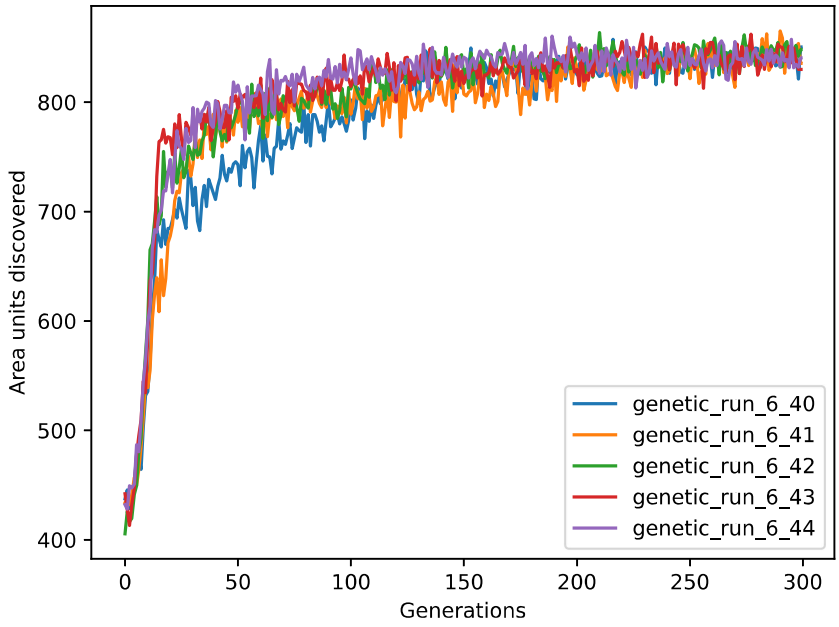


Figure B.54: Plot of the performance of the median agent in the population during genetic training.

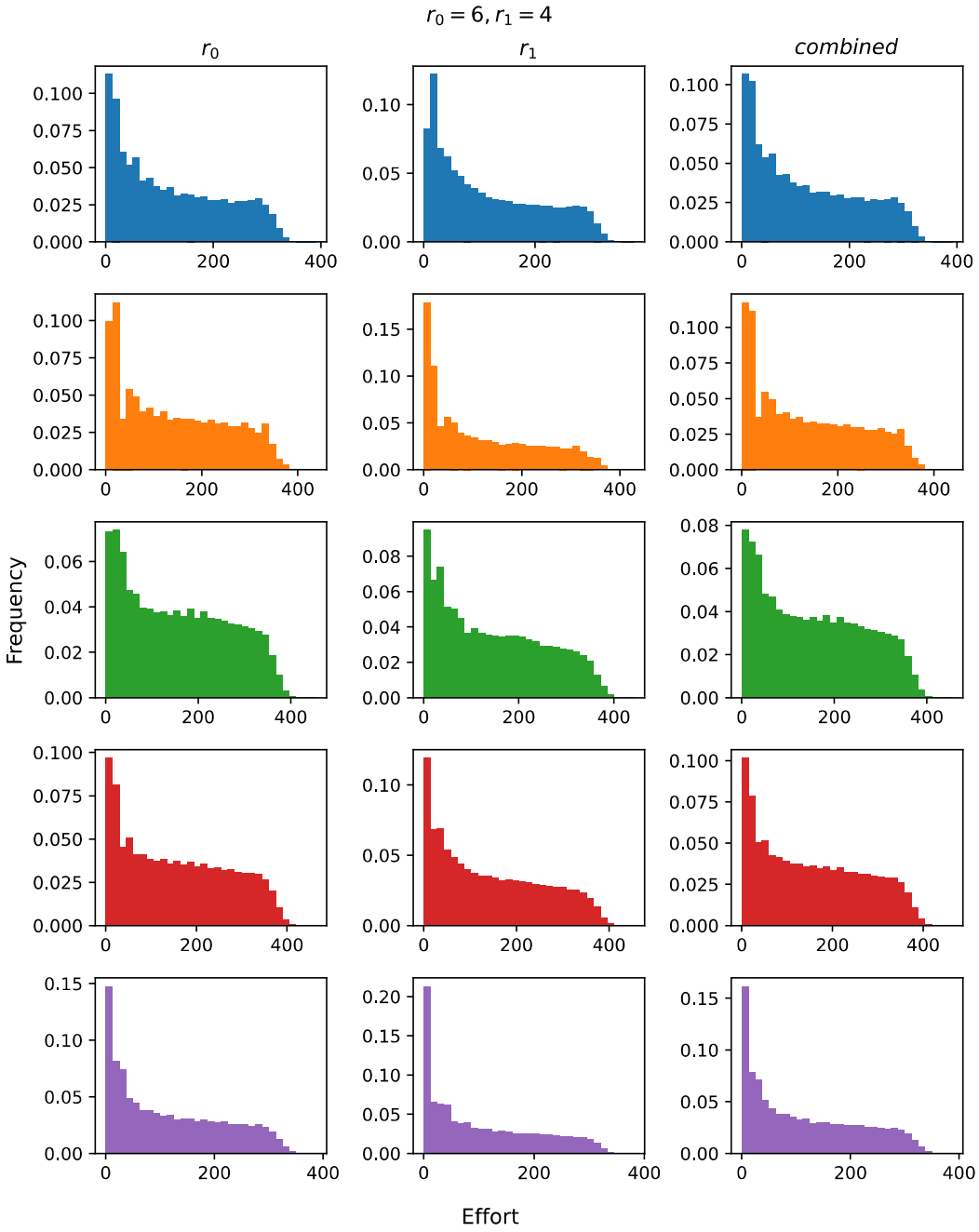


Figure B.55: Distribution of selected effort values for genitically trained agents. Each row contains data from a single agent. THe leftmost column is effort values sampled in resistance $r_0 = 3$, the middle in $r_1 = -4$, and the rightmost is from r_0 and r_1 combined.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY