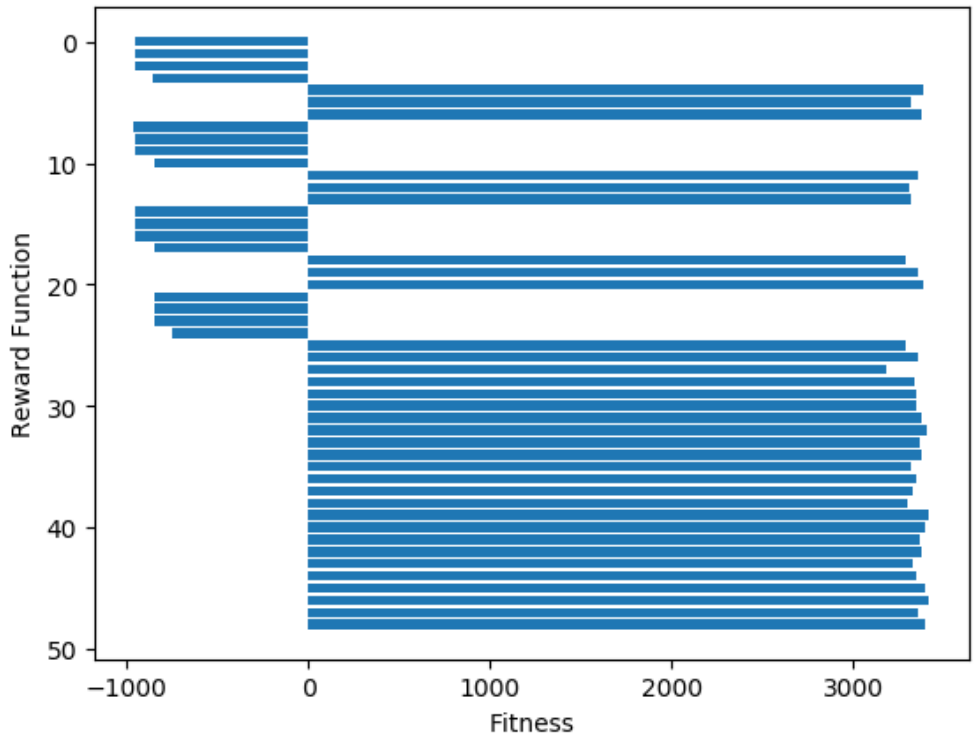
[ ]:

[16]:
```
plt.rcdefaults()
fig, ax = plt.subplots()

print(len(fitness_vals_thousand))
print(fitness_vals_thousand)
# Example data
reward_functions = []
error = np.var(fitness_vals_thousand[1:])


ax.barh(np.arange(0, len(fitness_vals_thousand)), fitness_vals_thousand,␣
 ↪align='center')
ax.invert_yaxis()  # labels read top-to-bottom
ax.set_ylabel('Reward Function Index')
ax.set_xlabel('Fitness')

plt.show()
```

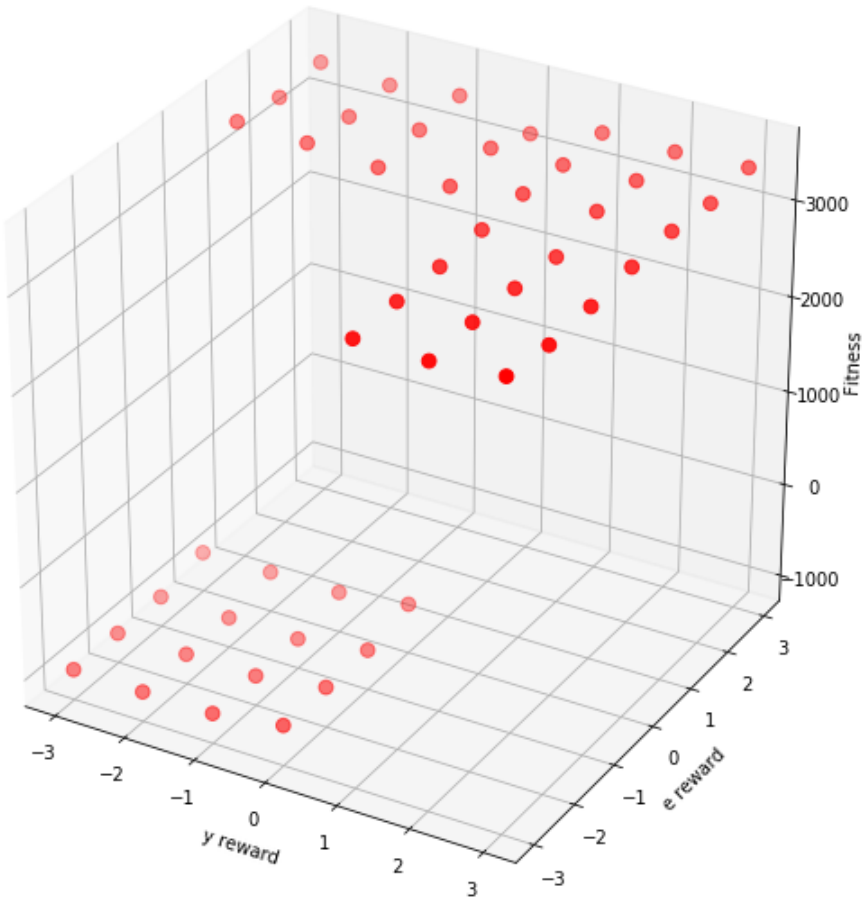
```
49
[-953.2500000001543, -958.9700000001545, -955.1200000001545, -856.2300000001445,
3390.760000000327, 3324.540000000316, 3376.5700000003294, -959.9600000001541,
-956.4400000001548, -957.4300000001547, -848.530000000145, 3357.4300000003254,
3307.2700000003097, 3316.9500000003195, -955.6700000001543, -958.200000000155,
-958.9700000001546, -845.1200000001439, 3295.7200000003186, 3355.3400000003194,
3390.430000000323, -845.2300000001436, -847.1000000001432, -850.0700000001436,
-747.7700000001317, 3292.2000000003122, 3355.010000000326, 3181.6500000003,
3335.320000000325, 3349.8400000003203, 3349.730000000328, 3377.7800000003285,
3406.380000000335, 3371.07000000033, 3375.2500000003297, 3322.450000000321,
3346.760000000331, 3333.2300000003256, 3302.210000000324, 3421.2300000003343,
3400.2200000003286, 3366.3400000003276, 3379.760000000332, 3327.6200000003228,
3353.9100000003227, 3399.0100000003317, 3415.7300000003315, 3362.160000000332,
3394.8300000003283]
```



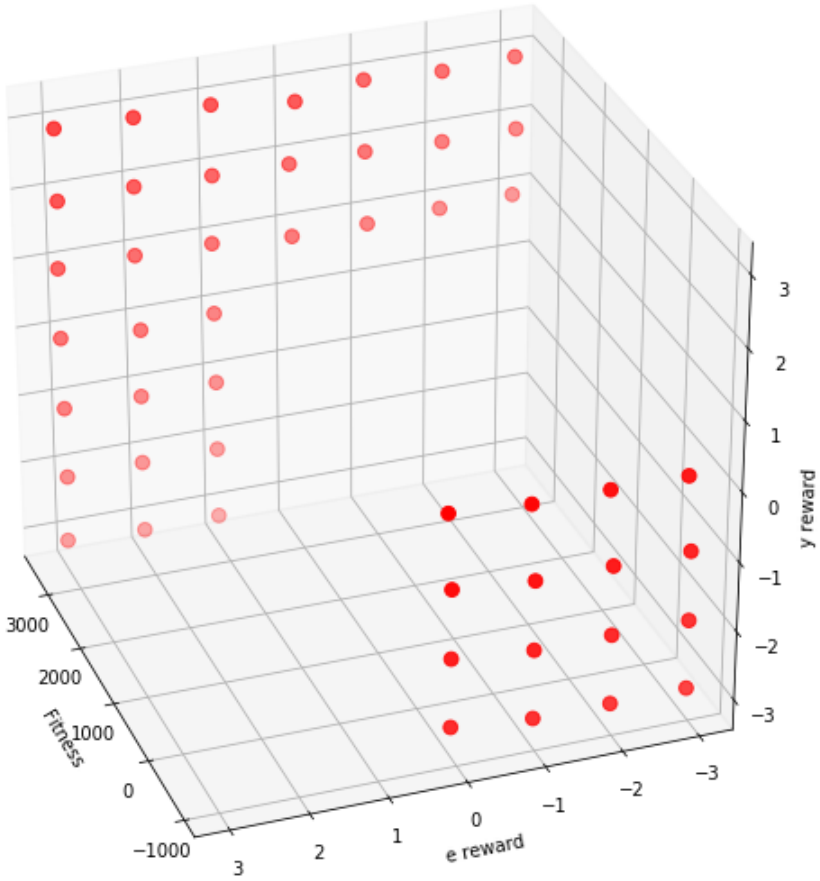
[32]:
```
max_fitness = np.max(fitness_vals_thousand)
max_index = np.where(fitness_vals_thousand == max_fitness)

max_y = y_vals_thousand[max_index[0][0]]
max_e = e_vals_thousand[max_index[0][0]]
#for i in range(len(max_index)):
    #max_y_e_tuples.append((y_vals_thousand[max_index[i]],␣
↪e_vals_thousand[max_index[i]]))


print(max_fitness)
print(type(max_index[0][0]))
print("Max y = ", max_y, "Max e = ", max_e)
#print(max_y_e_tuples)
```

```
<class 'tuple'>
3421.2300000003343
<class 'numpy.int64'>
Max y =  2 Max e =  1
```

[80]:
```
fig = plt.figure(figsize=(10,10))
ax = plt.axes(projection="3d")

ax.scatter3D(y_vals_thousand, e_vals_thousand, fitness_vals_thousand, c='r',␣
 ↪marker='o', s=60)

ax.set_xlabel('y reward')
ax.set_ylabel('e reward')
ax.set_zlabel('Fitness')

plt.show()
```

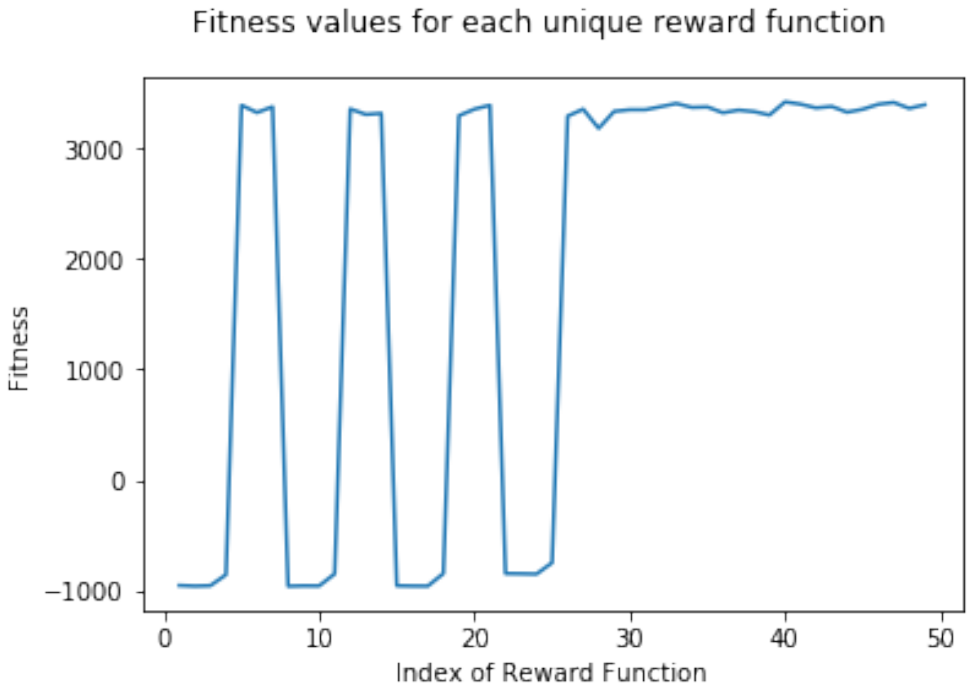


[79]:
```
fig = plt.figure(figsize=(10,10))
ax = plt.axes(projection="3d")

ax.scatter3D(fitness_vals_thousand, y_vals_thousand, e_vals_thousand, c='r',␣
 ↪marker='o', s=60)

ax.set_zlabel('y reward')
ax.set_ylabel('e reward')
ax.set_xlabel('Fitness')

ax.view_init(30, 160)
plt.draw()
```



[82]:
```
fig, axis = plt.subplots()

axis.plot(np.arange(1,len(fitness_vals_thousand)+1),fitness_vals_thousand)

fig.suptitle("Fitness values for each unique reward function", fontsize=12)
axis.set_xlabel('Index of Reward Function', fontsize=10)
axis.set_ylabel('Fitness', fontsize=10)

mean_center_init = np.mean(fitness_vals_thousand)
variance_center_init = np.var(fitness_vals_thousand)
cov_center_init = (variance_center_init/mean_center_init)*100

print("Mean Fitness over all reward functions = ", mean_center_init)
print("Variance of Fitness over all reward functions = ", variance_center_init)
print("Coefficient of Variation over all reward functions = ", cov_center_init)

mean_center_init = np.mean(fitness_vals_thousand[1:])
variance_center_init = np.var(fitness_vals_thousand[1:])
cov_center_init = (variance_center_init/mean_center_init)*100

print("Mean Fitness over all reward functions except case 1 with all rewards set␣
 ↪0 = ", mean_center_init)
print("Variance of Fitness over all reward functions except case 1 with all␣
 ↪rewards set 0 = ", variance_center_init)
print("Coefficient of variation over all reward functions except case 1 with all␣
 ↪rewards set 0 = ", cov_center_init)
```

```
Mean Fitness over all reward functions =  1962.8702040818025
Variance of Fitness over all reward functions =  3986013.3501988053
Coefficient of Variation over all reward functions =  203070.6534701002
Mean Fitness over all reward functions except case 1 with all rewards set 0 =
2023.62270833351
Variance of Fitness over all reward functions except case 1 with all rewards set
0 =  3888202.823124781
Coefficient of variation over all reward functions except case 1 with all
rewards set 0 =  192140.69930687753
```


Fitness values for each unique reward function
Fitness
3000
2000
1000
0
−1000
0
10
20
30
40
50
Index of Reward Function

[ ]: