TÉCNICO LISBOA

Instituto Superior Técnico
Computação Inteligente / Applied Computational Intelligence, MEEC, 1st Semester 2023/2024

Project 2 - EAs for Single and Multi-Objective Optimization

Students (Group 9):
Afonso Brito Caiado Correia Alemão | 96135 | afonso.alemao@tecnico.ulisboa.pt
Rui Pedro Canário Daniel | 96317 | ruipcdaniel@tecnico.ulisboa.pt
Teachers:
Nuno Horta, Joao Carvalho, Rui Neves, Vasco Pearson

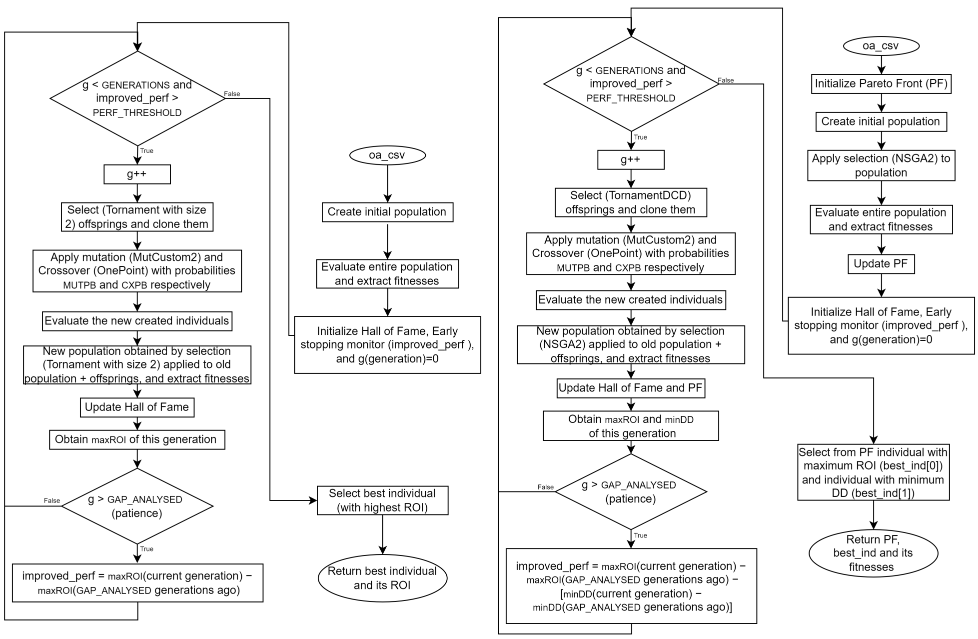
October 29, 2023

0 Introduction

This project aims at applying Evolutionary Algorithms (EAs) to solve a Computational Finance Challenge. The code for Single Objective Optimization (SOO) can be found in `ROImaximization.py` and for Multi-Objective Optimization (MOO) in `ROImaximization_DDminimization.py`. It also includes `computeRSI.py` and a `README` file describing the use and the sequence of results.

1 The Problem of Investing in the Stock Market

In the following sections, we describe the problem representation for each problem, and we used `DEAP` library [2] to implement the chosen EAs represented in Fig. 1.



(a): Single objective. (b): Multi-objective.

Figure 1: Fluxochart of used algorithms.

2 SOO Applied to ROI using Technical Indicators

We define a SOO problem to maximize Return On Investment (ROI) using the Relative Strength Index (RSI) as technical indicator, where the parameters to be optimized are: Lower band to open a long position (`LB_LP`); Upper band to close a long position (`UB_LP`); RSI period to apply for long positions (`RSI_long`); Lower band to close a short position (`LB_SP`); Upper band to open a short position (`UB_SP`); RSI period to apply for short positions (`RSI_short`). So we considered an individual = [`RSI_long`, `RSI_short`, `LB_LP`, `UB_LP`, `LB_SP`, `UB_SP`]. The first 2 features are 7, 14, or 21 days, and the remaining have values from 0 to 100, multiples of 5.

In this problem, an EA is more efficient than an exhaustive search method which would operate in $20^4 \times 3^2 = 1440000$ candidates, because there are 20 possibilities for the four bound features and 3 possibilities for `RSI_long` and `RSI_short`. We tuned the following hyper-parameters: mutation type and probability, crossover type and probability, and selection type. The applied strategy was fixing all variables and tuning one at a time by maximizing the average ROI from all stocks. The results in Table 1 were obtained with 3 runs, 20 generations with Early stopping patience of 8, and a population of 64. `MutCustom` mutates each feature with a probability of `indpbMut` by assigning a new random value within the entire range of the feature. On the other hand, `MutCustom2` mutates `LB_LP`, `UB_LP`, `LB_SP`, `UB_SP` features with a probability of `indpbMut`, by introducing a small variation of ± 5, if possible.

Table 1: SO Approach and Parameters.

Mutation	ProbMut	indpbMut	Crossover	ProbCross	Selection	Max ROI
MutCustom	0.1	0.5	OnePoint	0.7	Tournament(2)	90.364
	0.2					90.833
	0.3					92.295
	0.4					93.527
	0.5					92.121
	0.6					89.352
	0.7					93.597
	0.8					89.176
	0.9					89.408
MutCustom	0.7	0.1	OnePoint	0.7	Tournament(2)	92.465
		0.2				90.424
		0.3				90.167
		0.4				87.240
		0.5				93.597
		0.6				88.673
		0.7				86.271
		0.8				90.950
		0.9				91.248
MutCustom	0.7	0.5	OnePoint	0.7	Tournament(2)	93.597
			TwoPoint			85.550
			Uniform			93.565
			blending			76.793

Mutation	ProbMut	indpbMut	Crossover	ProbCross	Selection	Max ROI
MutCustom	0.7	0.5	OnePoint	0.1	Tournament(2)	80.098
				0.2		77.708
				0.3		81.335
				0.4		85.295
				0.5		90.358
				0.6		87.554
				0.7		93.597
				0.8		90.060
				0.9		94.244
MutCustom	0.7	0.5	OnePoint	0.9	Tournament(2)	94.244
					Tournament(3)	92.156
					Tournament(4)	86.363
					Tournament(8)	89.846
					Roulette	82.037
					Random	62.215
					Best	91.058
					StochasticUniversalSampling	86.355
					Lexicase	76.793
MutCustom	0.7	0.5	OnePoint	0.9	Tournament(2)	94.244
MutCustom2						95.065

We obtained the tuned configuration: {Mutation: MutCustom2; ProbMut: 0.7; indpbMut: 0.5; Crossover: OnePoint; ProbCross: 0.9; Selection: Tournament with size 2}. Then, using this configuration, we execute 30 runs (with different random seeds), 100 generations with an Early stopping patience of 10, and a population of 100 individuals, obtaining the results of Table 2 for the complete data series from Jan 2020 until Dec 2022.

Table 2: ROI (Max, Min, Mean and STD) over 30 runs.

Stocks (.csv)	Max	Min	Mean	STD
AAL	132.712	87.777	108.245	11.083
AAPL	91.191	52.615	73.644	10.673
AMZN	95.756	51.878	76.228	11.588
BAC	52.681	32.527	43.304	4.421
F	102.308	40.305	75.191	14.702

Stocks (.csv)	Max	Min	Mean	STD
GOOG	60.241	38.085	51.198	5.197
IBM	59.722	40.847	51.973	5.294
INTC	64.633	50.706	56.045	3.270
NVDA	282.386	209.333	242.684	20.489
XOM	90.099	55.255	74.208	9.113

We generate a boxplot graph in Fig. 2 with the results obtained from each of the runs, using min-max normalization so that the boxplot for different stocks can be compared.

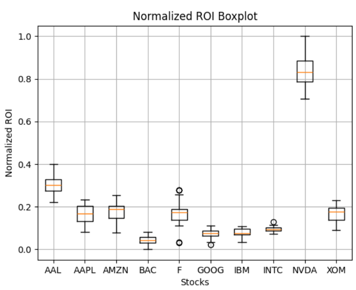
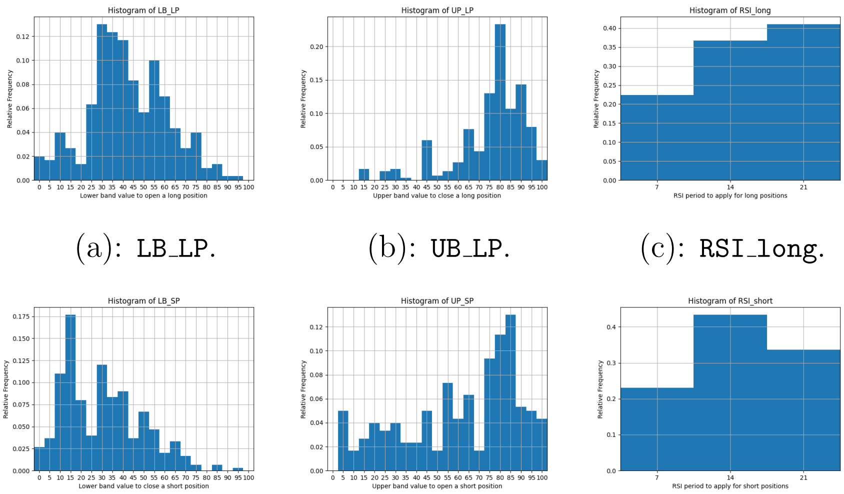


Figure 2: Normalized ROI Boxplots

The boxplot provided depicts the normalized ROI for various stocks, providing a visual comparison of the distribution and variability of ROI across different stocks. The stock with the highest median ROI is NVDA since it has, on average, provided far higher returns among the stocks listed. AAL, AAPL, and XOM have a relatively wide spread in their ROI as seen from the size of their boxplot, indicating higher variability in their returns. On the other hand, stocks like BAC, GOOG, IBM, and INTC have narrower boxes, signifying a smaller interquartile range and more consistency. BAC has one of the lowest median ROIs, indicating that it has a lower average return relative to the other stocks on the chart.

Finally, we create 6 histograms, one for each of the optimization variables with all the achieved results in the above experiments, using relative frequencies (obtained by normalization). Each histogram provides insights into the distribution of specific values for different trading strategies. Analyzing them we verify that the most frequent value for each bound is `LB_LP` = 30, `UB_LP` = 80, `LB_SP` = 15, `UB_SP` = 85. As expected `LB_LP` < `UB_LP` and `LB_SP` < `UB_SP`, meaning the trading strategy is considering both long position and short position actions, allowing for more flexibility in the trading approach. For long positions, there is a relatively longer Mode for the RSI period (21 days), which may capture more significant trends. On the other hand, for short positions, the most frequent RSI period (14 days) is shorter, capturing quicker price movements.



(a): `LB_LP`. (b): `UB_LP`. (c): `RSI_long`.

(d): `LB_SP`. (e): `UB_SP`. (f): `RSI_short`.

Figure 3: Histogram for each of the optimization variables.

3 SOO Train and Test Scheme

The ROI results obtained in the previous section can be optimal for the presented data series but are not realistic once it considers future values to optimize the RSI parameters. A more realistic approach would be to define a training period to obtain the optimal parameters and then apply them in a test period. This way, we considered a scheme where 9 years are used for training (Jan 2011 until Dec 2019) and 3 years for testing (Jan 2020 until Dec 2022). Considering the same conditions as section 2, the obtained results are reported in Table 3 for the training period, with 30 runs (using different seeds), 100 generations with an Early stopping patience of 10, and a population of 100 individuals. Then, for each run and for each stock, we apply the obtained best individual parameters and use them during the test period, obtaining the results in Table 4.

Table 3: ROI in the Train Period (Max, Min, Mean and STD) over 30 runs.

Stocks (.csv)	Max	Min	Mean	STD
AAL	203.673	127.327	182.930	20.972
AAPL	139.975	96.218	119.930	10.903
AMZN	404.874	158.024	365.451	60.908
BAC	92.870	71.315	86.630	5.618
F	75.514	50.431	64.975	7.360

Stocks (.csv)	Max	Min	Mean	STD
GOOG	137.404	99.128	120.912	8.463
IBM	83.660	41.277	65.477	10.237
INTC	82.910	63.909	71.657	4.377
NVDA	252.221	217.950	236.470	9.291
XOM	59.548	39.856	50.301	4.302

Table 4: ROI in the Test Period (Max, Min, Mean and STD) over 30 runs.

Stocks (.csv)	Max	Min	Mean	STD
AAL	36.841	-24.809	1.637	11.590
AAPL	50.627	-30.566	-0.736	13.135
AMZN	11.256	-9.878	1.950	5.504
BAC	19.234	-13.653	-1.819	9.842
F	69.771	-79.009	-11.976	36.139

Stocks (.csv)	Max	Min	Mean	STD
GOOG	44.283	-8.492	3.828	8.718
IBM	44.128	9.174	29.525	9.982
INTC	14.256	-21.257	-2.845	10.499
NVDA	210.477	101.063	179.174	38.037
XOM	6.422	-56.460	-28.853	17.325

The ROI figures in Table 2, where the testing period equals the training period, are typically higher than the ones in Table 4, where a more realistic split between the train and test period was applied. The ROI figures in Table 2 present an optimistic scenario which is often the case when models are tested on data they were trained on. This leads to overfitting where the model becomes too specialized to the training data, reducing its ability to generalize to new data. This event is evident when comparing the results to Table 4, which depicts ROI figures from a more realistic test period. The decrease, and in some cases negative ROI in Table 4, highlights the importance of robust out-of-sample testing to evaluate the true performance of a strategy, ensuring it can adapt and perform well for all kinds of situations.

4 MOO Applied to ROI and to DD using Technical Indicators

We define a MOO problem to maximize ROI and minimize the Drawdown (DD) using RSI as a technical indicator. From each run of a stock dataset, we obtain the individuals in the Pareto Front, highlighting the one with the highest ROI (`maxROI`) and the one with the lowest DD (`minDD`). For each stock, considering all solutions from the 30 runs, the one with the highest ROI corresponds to a fitness (ROI, DD) of (`MAXROI_ROI`, `MAXROI_DD`), with priority to the lowest DD in case of ties. On the other hand, the solution with the lowest DD corresponds to a fitness of (`MINDD_ROI`, `MINDD_DD`), prioritizing the highest ROI in case of ties.

We tune the following EA hyper-parameters: mutation type and probability, crossover type and probability, and selection type. Then, we fixed all variables and tuned one at a time by maximizing a defined performance Metric $= \texttt{MAXROI\_ROI} - \texttt{MAXROI\_DD} + \texttt{MINDD\_ROI} - \texttt{MINDD\_DD}$. The results in Table 5 were obtained with 3 runs, 20 generations with Early stopping patience of 8, and a population of 64 individuals.

Table 5: MO Approach and Parameters.

Mutation	ProbMut	indpbMut	Crossover	ProbCross	Selection	Metric
MutCustom2	0.1	0.5	OnePoint	0.7	NSGA2 + TournamentDCD	118.407
	0.2					100.665
	0.3					110.492
	0.4					101.768
	0.5					112.389
	0.6					100.358
	0.7					100.227
	0.8					99.190
	0.9					100.131
MutCustom2	0.1	0.1	OnePoint	0.7	NSGA2 + TournamentDCD	111.129
		0.2				99.747
		0.3				103.810
		0.4				110.326
		0.5				118.407
		0.6				101.179
		0.7				103.228
		0.8				110.040
		0.9				108.402
MutCustom2	0.1	0.5	OnePoint	0.7	NSGA2 + TournamentDCD	118.407
			TwoPoint			102.170
			Uniform			101.683
			blending			78.811

Mutation	ProbMut	indpbMut	Crossover	ProbCross	Selection	Metric
MutCustom2	0.1	0.5	OnePoint	0.1	NSGA2 + TournamentDCD	87.460
				0.2		92.067
				0.3		93.729
				0.4		97.747
				0.5		99.318
				0.6		99.826
				0.7		118.407
				0.8		99.322
				0.9		111.097
MutCustom2	0.1	0.5	OnePoint	0.7	NSGA2 +TournamentDCD	118.407
					NSGA2 +Tournament(2)	105.982
					NSGA2 +Tournament(3)	101.556
					NSGA2 +Tournament(4)	96.067
					NSGA2 +Tournament(8)	93.046
					NSGA2 +Roulette	90.351
					NSGA2 +Random	100.709
					NSGA2 +Best	89.206
					NSGA2 +StochasticUniversalSampling	86.201
					NSGA2 +Lexicase	86.616

We obtained the tuned configuration: {Mutation: MutCustom2; ProbMut: 0.1; indpbMut: 0.5; Crossover: OnePoint; ProbCross: 0.7; Selection: NSGA2 + TournamentDCD}. Then, by executing 30 runs (with different random seeds), 160 generations with an Early stopping patience of 10, and a population of 64 individuals, the results of Table 6 were obtained for the complete data series values from Jan 2020 until Dec 2022.

Table 6: ROI and DD for the MO approach over 30 runs.

Stocks (.csv)	`MAXROI_ROI`	`MAXROI_DD`	`MINDD_ROI`	`MINDD_DD`
AAL	132.712	4.920	0.000	0.000
AAPL	83.512	0.595	4.113	0.000
AMZN	91.405	1.132	65.473	0.000
BAC	51.630	0.620	23.064	0.000
F	106.449	0.157	102.308	0.000

Stocks (.csv)	`MAXROI_ROI`	`MAXROI_DD`	`MINDD_ROI`	`MINDD_DD`
GOOG	56.217	1.207	30.039	0.000
IBM	56.599	1.758	23.660	0.000
INTC	63.541	1.005	41.059	0.000
NVDA	266.714	0.076	255.749	0.000
XOM	88.300	1.907	7.172	0.000

Then, we generate the Pareto Front graphs for each of the above stocks, by superimposing the results from the 30 runs in a single plot, as represented in Fig. 4.

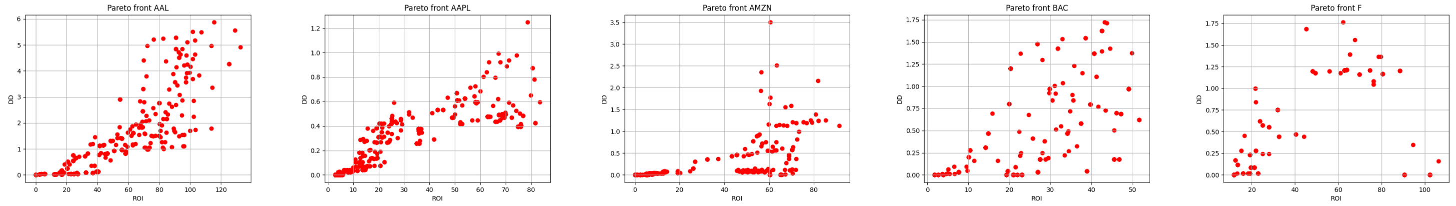
4.1 MO Train and Test Scheme

Next, we suggest and justify a train and test scheme when using a MO approach. The previously obtained results can be optimal for the presented data series but are not realistic once it considers future values to optimize the objective functions. Using a more realistic approach we define a training period of 9 years (Jan 2011 until Dec 2019) to obtain the optimal parameters and then apply them in a test period of 3 years (Jan 2020 until Dec 2022). Maintaining the MO evolutionary algorithm conditions, the obtained results are reported in Table 7 for the training period, with 30 runs (using different seeds), 160 generations with an Early stopping patience of 10, and a population of 64 individuals.

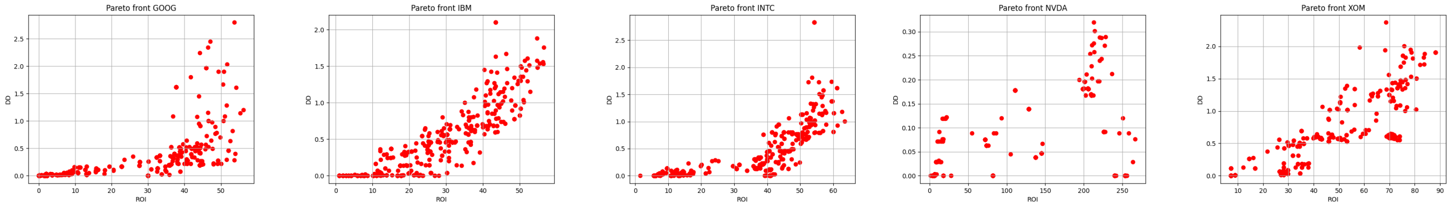
Table 7: Results of MO in the Train Period over 30 runs.

Stocks (.csv)	`MAXROI_ROI`	`MAXROI_DD`	`MINDD_ROI`	`MINDD_DD`
AAL	204.008	0.620	22.036	0.000
AAPL	129.302	0.525	71.502	0.000
AMZN	395.444	0.682	3.460	0.000
BAC	92.870	1.515	4.552	0.000
F	80.145	1.095	5.696	0.000

Stocks (.csv)	`MAXROI_ROI`	`MAXROI_DD`	`MINDD_ROI`	`MINDD_DD`
GOOG	133.351	1.255	8.701	0.000
IBM	83.660	2.069	6.968	0.000
INTC	76.871	3.969	72.709	0.000
NVDA	251.591	0.559	12.106	0.000
XOM	55.854	3.404	22.733	0.000



(a): `AAL.csv`. (b): `AAPL.csv`. (c): `AMZN.csv`. (d): `BAC.csv`. (e): `F.csv`.



(f): `GOOG.csv`. (g): `IBM.csv`. (h): `INTC.csv`. (i): `NVDA.csv`. (j): `XOM.csv`.

Figure 4: Pareto Front graphs.

Then, for each of the 30 runs, for each stock, it was obtained a Pareto Front (PF) of feasible parameter configurations, and we used all these individuals during the test period, obtaining the results in Table 8. The PF represents a range of optimal trade-offs between conflicting objectives. Choosing the entire PF for testing accounts for robustness, adaptability, risk mitigation, and decision support, which are crucial factors when optimizing and deploying strategies in dynamic and uncertain financial markets. Furthermore, it captures the entire spectrum of trade-offs between the objectives, ensuring that no potential solution is overlooked.

Table 8: Results of MO in the Test Period over 30 runs.

Stocks (.csv)	`MAXROI_ROI`	`MAXROI_DD`	`MINDD_ROI`	`MINDD_DD`	Stocks (.csv)	`MAXROI_ROI`	`MAXROI_DD`	`MINDD_ROI`	`MINDD_DD`
AAL	7.949	0.436	0.000	0.000	GOOG	37.289	0.420	1.493	0.000
AAPL	67.762	0.446	2.810	0.000	IBM	43.583	0.786	5.842	0.000
AMZN	46.413	0.083	0.000	0.000	INTC	15.859	0.243	1.409	0.000
BAC	27.497	0.176	19.234	0.000	NVDA	253.473	0.000	253.473	0.000
F	58.147	1.160	11.624	0.000	XOM	30.206	0.605	0.000	0.000

The results in Table 6 are typically more optimal than the ones in Table 8. In Table 6 the test period equals the training period, presenting an optimistic scenario where the model is tested on data it was trained on, which leads to overfitting where it becomes too specialized to the training data, reducing its ability to generalize to new data. On the other hand, Table 8 depicts a more realistic split between the train and test period. Performance decreased in this, demonstrating the importance of robust out-of-sample testing to evaluate the true performance of a strategy, ensuring it can adapt and perform well for all kinds of situations.

5 Concluding Remarks

By applying EAs we solve SOO and MOO Computational Finance Challenges. EAs are often more efficient than exhaustive search methods, as they can explore a wide range of the search space while also exploiting promising regions. In contrast, exhaustive search typically explores every possible solution, which becomes impractical when the search space is large (high computational complexity). This way EAs are highly scalable and their adaptability helps them focus on promising regions, even in vast and complex spaces.

The obtained results demonstrate the importance of assessment with a realistic split between the train and test period. The fine-tuning applied to the training set of the model may overfit that dataset, which is an important factor to avoid. Several techniques can be used to mitigate overfitting as maintaining diversity in the population by using weaker selection methods and/or broad search mutation, and by using a validation set in the algorithm training process to tune the hyper-parameters.

References

[1] CInte 2022/2023, Project 2 – EAs for Single and Multi-Objective Optimization;

[2] DEAP 1.4.1 documentation. Available: `https://deap.readthedocs.io/en/master/api/tools.html`.