A Performance Study of Multiobjective Particle Swarm Optimization Algorithms for Market Timing

Ismail Mohamed
School of Computing
University of Kent, Canterbury, UK
0000-0002-7465-7123

Fernando E. B. Otero
School of Computing
University of Kent, Canterbury, UK
0000-0003-2172-297X

Abstract—**Market timing is the issue of deciding when to buy or sell a given asset on a financial market. As one of the core issues of algorithmic trading systems, designers of such systems have turned to computational intelligence methods to aid them in this task. In our previous work, we introduced a number of Particle Swarm Optimization (PSO) algorithms to compose strategies for market timing using a novel training and testing methodology that reduced the likelihood of overfitting and tackled market timing as a multiobjective optimization problem. In this paper, we provide a detailed analysis of these multiobjective PSO algorithms and address two limitations in the results presented previously. The first limitation is that the PSO algorithms have not been compared to well-known algorithms or market timing techniques. This is addressed by comparing the results obtained against NSGA-II and MACD, a technique commonly used in market timing strategies. The second limitation is that we have no insight regarding diversity of the Pareto sets returned by the algorithms. We address this by using RadViz to visualize the Pareto sets returned by all the algorithms, including NSGA-II and MACD. The results show that the multiobjective PSO algorithms return statistically significantly better results than NSGA-II and MACD. We also observe that the multiobjective PSO^{SP} algorithm consistently displayed the best spread in its returned Pareto sets despite not having any explicit diversity promoting measures.**

Index Terms—**particle swarm optimization, multiobjective optimization, market timing, algorithmic trading**

I. Introduction

Market timing is the issue of deciding when to buy or sell a given asset on a financial market. A market timing strategy can be composed of a set of components that digest current and past market context and return a recommendation on the action to take. In a strategy containing more than one component, the components have a weight associated with them, and the final action taken is based on the aggregate of the individual recommendations multiplied by their respective weights. Previous approaches on using computational intelligence to aid the formation of market timing strategies were vastly dominated by GA based on the volume of publications [1] [2]. These approaches either attempted to optimize the parameters of a preset selection of components or select a subset from a set of components with predefined parameters. None of the surveyed approaches attempted to perform both functions simultaneously. This limits the designer of a market timing strategy to committing computational resources to only one of those functions at a time, curtailing the flexibility in choice of components and parameters for consideration. When it came to training and testing, all of the surveyed approaches followed a protocol known as Step Forward testing. In Step Forward testing, a stream of financial data is split arbitrarily into two sections: the earlier section of the stream to be used for training while the latter was used for testing. One of the main criticisms for the use of Step Forward testing is that algorithms have the liability of overfitting to only the trends observed in the data and hence suffering a significant degradation in performance when encountering unseen trends during live training. The majority of literature observed on the use of computational intelligence for market timing tackled it as a single objective optimization problem. When dealing with market timing strategies deployed in live trading scenarios, users of such strategies would gauge performance using a number of financial metrics that represent various aspects of profits, losses and exposure to risk. Modeling a market timing strategy as a single objective optimization problem limits its utility for live trading, as it constrains optimization to only one of the aforementioned aspects. Although a number of approaches modeled market timing as multiobjective optimization problems, such as [3], [4], they are limited in scope and pale in comparison to those that modeled it as a single objective optimization problem in terms of volume.

In [5], we proposed the use of Particle Swarm Optimization (PSO) to tackle market timing as a single objective optimization problem but in such a fashion that considers both the selection of components and the tuning of their parameters simultaneously. This is further developed in [6] with the introduction of a novel training and testing methodology called Trend Representative Testing. In Trend Representative Testing the strategies are explicitly exposed to a variety of market conditions both during training and testing. This reduces the likelihood of overfitting to particular market conditions, which has been a common criticism of Step Forward testing [7]. We also introduced a GA benchmark in [6] in order to be able to better evaluate the performance of our PSO algorithms. Finally, we evolved our approach to tackling market timing from a single objective optimization problem to considering it as a multiobjective one, and adapting the PSO and GA algorithms introduced in [5] and [6] accordingly. The performance of the multiobjective PSO algorithms was evaluated by measuring the hypervolume [8], [9] obtained against five financial objectives. Although the results showed that a multiobjective PSO variant attained statistical significance under certain market conditions, the evaluative context was limited as the PSO algorithms were only compared to a multiobjective GA benchmark which was introduced within the work. The results also do not provide insight into the diversity of the solutions returned by the algorithms.

In this paper, we provide a deeper insight into the algorithms introduced in [10], and address the limitations in the results there. We will provide a better evaluative context in regards to performance by comparing the results of the algorithms against NSGA-II, a well established multiobjective optimization GA algorithm, and MACD, a component widely used in market timing strategies to generate recommendations for action to take. We will also look into the diversity and spread of the solutions in the Pareto sets returned by the algorithms in [10], and compare them with NSGA-II and MACD by visualizing the results using RadViz [11].

The remainder of this paper is structured as follows: Firstly, We will begin by taking a deeper look at the algorithms introduced in [10], and contrasting their structure with known multiobjective PSO algorithms. Secondly, we will compare the performance of the algorithms against NSGA-II and MACD, as well as their single objective optimization counterparts from [6]. Thirdly, we will look into the diversity of the Pareto sets returned by the algorithms and compare them with NSGA-II and MACD. Finally, we conclude by providing a summary of our findings and suggestions for future research.

II. Algorithms

In this section, we take a detailed look at the PSO and GA algorithms introduced in [10]. In order to avoid confusion during comparisons, the multiobjective variants of the algorithms from [10] will be prefixed with the Greek letter Lambda, resulting in the multiobjective algorithms being labeled as λ-PSO, λ-PSO$^{\mathrm{SP}}$ and λ-GA.[1]

As mentioned in [10], the algorithms underwent a number of modifications to adapt them to multiobjective optimization. The two common adaptations that were applied to all three algorithms are the use of a non-dominated archive and the shift to using of a dominance based fitness approach based on the five optimized financial objectives. The non-dominated archive is used to keep track of the non-dominated solutions discovered by the algorithms during their run. The pseudocode for the non-dominated archive can be seen in Algorithm 1. The admittance of new solutions into the archive are maintained by the central function *ADD* (line 4). In order for a solution to be admitted into the archive it is compared with the current occupants in the archive and its level of dominance is determined (lines 7 to 17). The level of dominance between two solutions is ascertained using the utility function $d(x, y)$, also known as the dominance score (line 1). Given two solutions, x and y, to be evaluated for dominance based on five objectives, a $d(x, y)$ of three would indicate that solution x is better than solution y in three of the five objectives. A potential solution is only allowed into the archive if it gets a dominance score of at least one against any of the current occupants. Also, if while scanning against the current occupants, the incoming solution achieves a dominance score of five (i.e. achieving strong dominance), the occupant is marked for deletion and removed from the archive (lines 18 to 20). The archive is unbounded and is not limited to maintaining only a limited number of non-dominated solutions. The only other function performed by the archive is the $SELECT$ function, which returns a random member from the current occupants of the archive to be used by the algorithms in various operations as we will see shortly (lines 23 to 25).

The λ-GA algorithm maintains a non-dominated archive to keep track of the non-dominated solutions as the algorithm runs, and the contents are returned as the discovered Pareto set at the end of the run. The algorithm begins with a population of N randomly generated solutions, and they all initially admitted to the archive. Then, the algorithm repeats two basic operations across the current generation of solutions: mutation and crossover. For mutation, if the value of a randomly generated value is less than the mutation threshold M, then a solution is picked from the current generation using tournament selection and that solution undergoes mutation and is added to the next generation of solutions. The mutation process replaces a random component with a given solution with a fresh randomly generated copy. A crossover event is also triggered in a fashion similar to mutation by checking if a value of random variable is less than the crossover threshold C. Tournament selection is performed twice to select parents from the current generation and the offspring are added to the next generation. As the components in the solutions are atomic in nature, a crossover point is selected at random such that it falls on the border of components, not within them, in order to generate valid offspring. After performing mutation and crossover, a random candidate is selected from the non-dominated archive and added to the next generation in order to maintain pressure to discover non-dominated areas in the search space in an elitist fashion. The next generation at this point replaces the current one and the fitness of the current generation is calculated; this process is repeated until the stopping criteria for the algorithm are met, and the contents of the non-dominated archive at the end of the run are returned as the discovered Pareto set.

As with λ-GA, multiobjective PSO algorithms—λ-PSO, λ-PSO$^{\mathrm{S}}$ and λ-PSO$^{\mathrm{SP}}$—employ the use of a non-dominated archive to keep track of non-dominated solutions as they are discovered. The λ-PSO algorithm begins by initializing a swarm of N particles and then iterates over all the particles in swarm, calculating each particle's new velocity and use that to update its respective state. Each particle's velocity is calculate by adding three components: bias, cognitive component and social component. Bias multiplies a particle's previous velocity

[1] The reason Lambda was chosen as a prefix is due to its relation to the hypervolume calculation. The hypervolume calculation was used to assess the performance of the multiobjective algorithms in [10] and is based on the Lebesgue measure. The symbol used to represent this measure is the Greek letter Lambda, and we borrow this symbol to demarcate the multiobjective optimization algorithms.

by an exponentially decreasing factor. The use of bias in the velocity update depends on the clamping mechanism, which will be discussed shortly. The cognitive component is the traditionally the difference between a given particle's best discovered solution so far and its current state. Since solutions in this multiobjective optimization context are evaluated based on dominance, each particle maintains a personal non-dominated archive. A candidate is selected at random from the particle's personal non-dominated archive in order to calculate the value for the cognitive component of the velocity update. As for the social component, a temporary non-dominated archive is formed to contain the non-dominated neighbors of a given particle, a candidate is selected at random from that archive to represent a neighbor and calculate the value for the social component of the velocity update. In order to prevent the particles from escaping the search space, the user is given a choice of a velocity clamping mechanism: either Clerc's Constriction [12] or multiplying the entire velocity update by a fixed factor. After arriving at the a particle's velocity update array, it is used to update the current velocity, and subsequently the particle's state. Fitnesses for the particles are then calculated, each particle updates its personal non-dominated archive and the algorithm's non-dominated archive is updated using the current states in the swarm. At the end of algorithm's run, the non-dominated archive is return as the discovered Pareto set. The λ-PSO$^{\mathrm{S}}$ and λ-PSO$^{\mathrm{SP}}$ extend the λ-PSO algorithm with a Stochastic State Update and Pruning mechanisms [6], [13]. When pruning is enabled, the resultant algorithm is labeled λ-PSO$^{\mathrm{SP}}$, otherwise the resultant algorithm is labeled λ-PSO$^{\mathrm{S}}$.

Algorithm 1 Archive used to maintain non-dominated solutions.

```
$d(x,y)$: dominance score – the number of objectives where solution $x$ is better than solution $y$
$N$: number of objectives being optimized.
$archive \leftarrow \{\}$
procedure ADD($x$)
    $remove \leftarrow \{\}$
    $add\_solution \leftarrow False$
    for every $s_i$ in $archive$ do
        if $d(x, s_i) > 0$ then
            $add\_solution \leftarrow add\_solution$ OR $True$
        end if
        if $d(x, s_i) = N$ then
            $remove \leftarrow remove \cup s_i$
        end if
    end for
    if $add\_solution = True$ then
        $archive \leftarrow archive \cup x$
    end if
    for every solution $r_i$ in $remove$ do
        delete $r_i$ from $archive$
    end for
end procedure

function SELECT( )
    return random member from $archive$
end function
```

III. Comparison Against NSGA-II and MACD

The results presented in [10] were limited in two primary ways. The first limitation is that the results of the PSO algorithms are only compared to a GA benchmark that was introduced in the same paper. In order to address this limitation, we compare the performance of these algorithms against a more established, well-researched multiobjective optimization algorithm and a widely used technique for market timing. The second limitation is that we have no insight on the quality of the Pareto sets returned by the algorithms in terms of diversity. Ideally, we would prefer sets that cover a larger area of the Pareto front on the objective space and not sets that are clustered around a few point points on the Pareto front. We address this second limitation by visualizing the returned Pareto sets from all algorithms, including those discussed in this section, using RadViz [11] and inspecting the spread of these sets as will be seen in the next section.

For our choice of a well established multiobjective optimization algorithm, we have selected the Nondominates Sorting Genetic Algorithm (NSGA-II) [14]. The NSGA-II algorithm was first introduced in 2002, and is now amongst the most widely used and cited algorithms within the domain of multiobjective optimization. Early multiobjective optimization algorithms suffered from a number of limitations, including adopting a non-elitist approach, the need for specifying one or more parameters for the algorithm to run and suffering high complexity on the order of $O(MN^3)$, where M is the number of objectives and N is the population size. NSGA-II introduces a number of measures to directly address these limitations. Since its introduction, NSGA-II has been widely adopted in a number of domains, including being the benchmark in a number of market timing applications such as [4]. As for our selection of a technical analysis indicator that is widely used market timing applications, we opted to used the Moving Average Converge Diverge (MACD) indicator [15]. MACD depends on studying the exponential moving averages of the price data in an attempt to identify the momentum behind the underlying trend in the price. The MACD indicator uses two exponential moving averages, a fast one and another slow one, and the period of both being user-defined parameters. By subtracting the fast moving average from its slow counterpart, we arrive at a new series called the MACD series. We then further smooth the MACD series by a user defined parameter to generate what is known as the Signal line. Crossover points between the Signal line and the MACD series indicate an imminent change in momentum and a subsequent change in the direction of the current price trend. Depending on the direction of the crossover, we either buy (when the signal crosses up) or sell (when the signal line crosses down).

In order to perform the comparison, NSGA-II and the MACD indicator utilized the same experimental setup and

TABLE I: IRace discovered configurations for NSGA-II and MACD.

NSGA-II		**MACD**	
Parameter	*Value*	*Parameter*	*Value*
Population Size	47	Short EMA Period	12
Generations	72	Long EMA Period	26
Mutation Probability	0.0462	Signal Period	9
Crossover Probability	0.5559		

dataset as the λ-PSO$^{\mathrm{SP}}$, λ-PSO and λ-GA algorithms in the previous set of experiments. NSGA-II underwent hyperparameter optimization using IRace with the same budget as the other algorithms. The final configuration selected by IRace for NSGA-II can be seen in Table I. From Table I, we can see that although the population size was comparable to our GA benchmark, the number of generations needed was three times that our GA benchmark at 72 compared to the GA's 24. The mutation and crossover probabilities discovered are relatively low compared to typical values used for genetic algorithms, and even our own GA benchmark. There are no values for tournament selection as it is fixed at two as part of the NSGA-II specification. Since MACD also has a set of parameters, we experimented with three versions: one using industry standard default values, one optimized using IRace with the same budget as the other algorithms and one optimized using IRace with three times the budget available to the other algorithms. During experimentation though, the MACD variants using parameter values from both IRace runs did not produce any transactions in any of our training and testing sets. The only configuration that did result in some transactions was the one with industry default values for the parameters, and hence all further references to MACD will be to the one using those values. The values used for the MACD parameters can also be seen in Table I.

Table II shows the hypervolume results for all algorithms including NSGA-II and MACD. As can be seen from Table II, NSGA-II and MACD did not achieve any wins in terms of mean hypervolume when compared to λ-PSO$^{\mathrm{SP}}$, λ-PSO and λ-GA. The λ-PSO$^{\mathrm{SP}}$ algorithm maintains a considerable lead with 21 wins, followed by GA with 6 wins and finally PSO with 3 wins. In some instances, λ-PSO$^{\mathrm{SP}}$ showed a mean hypervolume an order of magnitude higher than NSGA-II as can be seen when triplets 6 and 7 were used in testing. On the other hand, MACD on its own failed to produce transactions under certain testing strands, indicated by achieving zero hypervolume. In cases where it did produce transactions, the mean hypervolume is still significantly lower than the best performing algorithm's mean hypervolume for that strand.

Table III shows the best performance achieved per objective optimized, including solutions from both NSGA-II and MACD. Here we identify the algorithm and strand where the best performing solution was observed in terms of the objective at hand, and compare it with the best performing solutions obtained by the remaining algorithms. Again, in case of more than one solution being non-dominated, we follow a lexicographical approach in our comparison based on the following ordering: AROR, Portfolio Risk, VaR, Transactions Count and Solution Length. As can be seen in Table III, λ-PSO$^{\mathrm{SP}}$ retained its edge over all other algorithms including the two new benchmarks: NSGA-II and MACD. The fact that MACD was able to achieve a value of 2.016 for AROR without any transactions when looking for the best performing solution for VaR is not a mistake but instead a limitation of how the AROR value is calculated. It is in such situations that it is important to consider the number of transactions and, hence, our confidence in the strategy. A transaction count of zero would result in the lowest confidence possible in a candidate market timing strategy, and therefore the candidate presented by MACD for the JBLU1 strand is an unworthy one. We

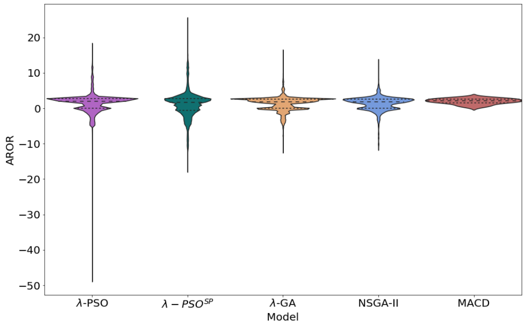


Fig. 1: Violin plot comparing AROR performance for all multiobjective algorithms, along with NSGA-II and MACD, across all testing strands. We can see that the medians and the bulk of their solution distributions for all algorithms are close to each other. Only λ-PSO$^{\mathrm{SP}}$ has shown a tail achieving higher AROR values than the other algorithms in the comparison.

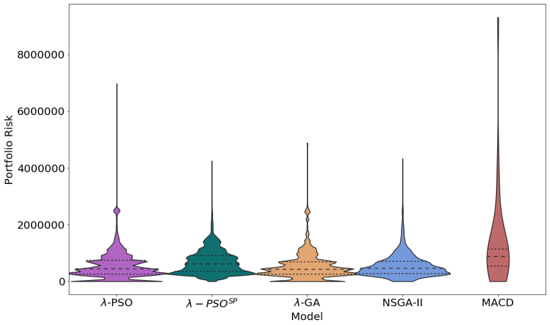


Fig. 2: Violin plot comparing Annualized Portfolio Risk performance for all multiobjective algorithms, along with NSGA-II and MACD, across all testing strands. With the exception of MACD, all multiobjective algorithms have their medians and the bulk of their solution distributions are close to each other. MACD, on the other hand displays a higher median and a long tail stretching into higher values of risk, indicating a worse performance when compared to the other algorithms.

TABLE II: Hypervolume results for each algorithm over the ten datasets including NSGA-II and MACD as benchmarks. The minimum, mean and max values are obtained by running each algorithm ten times on each dataset. Best mean results are highlighted in bold.

				λ-PSO^{SP}			**λ-GA**			**λ-PSO**			**NSGA-II**		**MACD**
#	**Trend**	**Strand**	**Min**	**Mean**	**Max**	**Min**	**Mean**	**Max**	**Min**	**Mean**	**Max**	**Min**	**Mean**	**Max**	
0	↑	IAG1	0.00E+00	2.07E+12	2.07E+13	0.00E+00	3.14E+13	1.57E+14	7.50E+14	**1.08E+15**	1.28E+15	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	↔	MGA4	7.59E+15	**8.68E+15**	9.47E+15	5.27E+15	7.46E+15	8.30E+15	4.15E+15	5.15E+15	5.65E+15	2.18E+14	8.83E+14	2.23E+15	3.33E+14
	↓	IAG2	3.79E+15	**4.28E+15**	4.80E+15	1.03E+15	1.93E+15	2.28E+15	7.45E+14	1.16E+15	1.30E+15	3.10E+13	2.06E+14	5.70E+14	1.44E+15
1	↑	BSX1	3.00E+13	9.91E+14	1.47E+15	1.01E+13	1.81E+15	3.03E+15	1.96E+15	**2.26E+15**	3.14E+15	0.00E+00	2.90E+14	1.29E+15	0.00E+00
	↔	LUV1	0.00E+00	2.49E+13	2.49E+14	0.00E+00	1.86E+14	4.65E+14	0.00E+00	**1.95E+14**	4.46E+14	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	↓	KFY1	8.82E+15	**9.43E+15**	9.59E+15	2.43E+15	5.13E+15	5.71E+15	2.18E+15	2.78E+15	2.99E+15	5.40E+14	1.36E+15	2.47E+15	0.00E+00
2	↑	EXC1	8.61E+15	**9.40E+15**	1.03E+16	3.00E+15	4.47E+15	5.54E+15	1.69E+15	2.37E+15	3.00E+15	3.12E+13	6.36E+14	1.73E+15	5.30E+14
	↔	LUV2	9.90E+15	**1.03E+16**	1.05E+16	5.44E+15	6.74E+15	7.75E+15	3.16E+15	3.71E+15	4.46E+15	1.86E+14	1.05E+15	2.30E+15	1.27E+15
	↓	KFY2	1.09E+16	**1.20E+16**	1.24E+16	4.88E+15	6.84E+15	8.08E+15	4.41E+15	4.94E+15	5.55E+15	2.52E+14	1.01E+15	1.86E+15	0.00E+00
3	↑	AVNW1	2.89E+13	4.05E+14	7.16E+14	2.29E+15	**2.51E+15**	2.63E+15	1.92E+15	2.02E+15	2.13E+15	0.00E+00	3.30E+13	1.46E+14	0.00E+00
	↔	PUK1	0.00E+00	2.82E+14	3.93E+14	7.83E+14	**6.63E+15**	1.11E+16	2.74E+15	4.13E+15	5.26E+15	0.00E+00	2.00E+14	8.40E+14	4.67E+14
	↓	LUV3	7.14E+15	**7.86E+15**	8.22E+15	1.58E+15	2.86E+15	3.70E+15	1.82E+15	2.49E+15	2.66E+15	1.24E+14	6.30E+14	1.19E+15	5.08E+12
4	↑	KFY3	5.96E+15	**7.30E+15**	8.07E+15	1.93E+15	3.57E+15	4.70E+15	2.08E+15	2.66E+15	2.90E+15	0.00E+00	8.35E+14	1.72E+15	0.00E+00
	↔	EXC2	2.48E+16	**2.69E+16**	2.89E+16	1.20E+16	1.52E+16	1.66E+16	1.04E+16	1.17E+16	1.25E+16	0.00E+00	2.05E+15	6.19E+15	0.00E+00
	↓	LUV4	1.06E+16	**1.16E+16**	1.24E+16	5.54E+14	3.70E+15	6.77E+15	1.67E+15	2.38E+15	2.78E+15	0.00E+00	9.08E+14	1.75E+15	2.98E+15
5	↑	EXC3	8.83E+15	**9.78E+15**	1.07E+16	4.63E+15	6.14E+15	6.95E+15	3.55E+15	4.12E+15	4.30E+15	3.70E+14	9.18E+14	1.76E+15	9.38E+13
	↔	PUK2	3.26E+15	4.66E+15	6.31E+15	7.38E+15	**1.33E+16**	1.71E+16	9.46E+15	1.06E+16	1.13E+16	4.66E+14	1.66E+15	3.90E+15	8.41E+14
	↓	MGA1	2.13E+16	**2.57E+16**	2.69E+16	3.58E+15	4.68E+15	5.23E+15	1.89E+15	3.23E+15	3.93E+15	2.62E+14	7.06E+14	1.52E+15	3.86E+14
6	↑	ED1	6.40E+15	**6.86E+15**	7.19E+15	2.31E+15	3.36E+15	3.92E+15	1.46E+15	2.04E+15	2.39E+15	7.19E+13	3.83E+14	1.35E+15	0.00E+00
	↔	EXC4	3.04E+16	**3.27E+16**	3.38E+16	9.92E+15	1.37E+16	1.59E+16	9.40E+15	1.06E+16	1.14E+16	1.16E+14	1.08E+15	3.65E+15	2.57E+14
	↓	PUK3	4.86E+15	**5.73E+15**	6.10E+15	5.92E+14	1.61E+15	2.20E+15	5.31E+14	1.16E+15	1.48E+15	3.29E+13	1.22E+14	3.59E+14	1.05E+15
7	↑	BSX2	1.65E+16	**1.72E+16**	1.75E+16	6.87E+15	9.81E+15	1.17E+16	4.63E+15	5.74E+15	6.20E+15	4.44E+14	1.27E+15	2.16E+15	3.15E+14
	↔	ED2	1.25E+16	**1.28E+16**	1.36E+16	7.41E+15	8.86E+15	9.19E+15	4.23E+15	4.82E+15	5.24E+15	3.78E+14	1.37E+15	2.43E+15	3.31E+15
	↓	JBLU1	7.44E+15	**8.45E+15**	8.89E+15	5.27E+15	7.34E+15	8.37E+15	2.71E+15	3.50E+15	4.19E+15	7.95E+13	7.08E+14	2.02E+15	0.00E+00
8	↑	MGA2	0.00E+00	1.86E+14	2.49E+14	4.68E+14	**2.37E+15**	3.55E+15	9.83E+14	1.55E+15	2.52E+15	0.00E+00	4.20E+13	2.11E+14	0.00E+00
	↔	MGA3	2.00E+15	5.01E+15	7.30E+15	4.66E+15	**7.07E+15**	1.15E+16	4.64E+15	5.61E+15	6.39E+15	0.00E+00	4.02E+14	1.32E+15	0.00E+00
	↓	ATRO1	1.15E+16	**1.37E+16**	1.53E+16	7.63E+14	6.86E+15	8.59E+15	5.85E+15	6.78E+15	7.17E+15	0.00E+00	2.51E+14	6.95E+14	5.56E+14
9	↑	AVNW2	8.00E+15	**8.80E+15**	9.57E+15	3.61E+15	5.24E+15	6.11E+15	4.13E+15	4.58E+15	4.82E+15	1.77E+14	9.44E+14	1.92E+15	0.00E+00
	↔	EXC5	1.00E+16	2.19E+16	2.63E+16	7.02E+15	**2.81E+16**	3.67E+16	2.02E+16	2.29E+16	2.50E+16	0.00E+00	2.87E+15	1.34E+16	7.96+14
	↓	AVNW3	9.10+15	**1.03E+16**	1.16E+16	3.52E+15	6.13E+15	7.61E+15	3.36E+15	3.86E+15	4.20E+15	1.66E+14	1.27E+15	2.35E+15	0.00E+00

can also see that while optimizing for Solution Length, λ-PSO^{SP} was able to achieve a higher AROR and Transactions Count that MACD, albeit with a higher risk profile, while also utilizing a single technical indicator. The indicator that λ-PSO^{SP} ended up utilizing was the Hammer Candlestick pattern with a trending period of 24 and a smoothing period of 10. Another interesting phenomenon is that λ-PSO^{SP} was able to achieve the highest AROR while optimizing for that particular metric by using only two technical indicators. With the exception of MACD, this is an order of magnitude lower in length when compared with λ-PSO, λ-GA and NSGA-II. The strategy proposed by λ-PSO^{SP} in that scenario depends on the Harami Cross and Rickshaw Man Candlestick patterns with the following parameters:

- Harami Cross: Trend Period: 21, Smoothing Period: 5, Weight: 0.97
- Rickshaw Man: Trend Period: 29, Smoothing Period: 12, Weight: 0.03

Table IV shows the comparison of all the multiobjective optimization algorithms mentioned thus far with their single objective counterparts from [6], including the additional benchmarks of MACD and NSGA-II. In Table IV we highlight the highest achiever across all the algorithms per testing strand. The reason behind including this comparison is to see whether following a Pareto dominance-based approach to market timing improved the quality of the solutions generated even if just considering the single objective AROR. The rankings of the algorithms based on the number of wins via attaining maximum AROR per testing strand are as follows: λ-PSO^{SP} (13), λ-PSO (8), MACD (6), λ-GA (3), NSGA-II (1), PSO^{S} (0), GA (0) and PSO (0). We can see that the multiobjective optimization variants of PSO dominate the rest of the algorithms and hold the top two positions. This is followed by MACD, then the multiobjective variants of GA, although NSGA-II was only able to achieve a singular win. The single objective variants of our algorithms fared the worst, scoring no wins when compared with their multiobjective counterparts. This indicates that pursuing a multiobjective optimization approach improves the overall quality of the market timing strategies generated when compared to a single objective optimization approach.

In order to provide an overall picture comparing the performance of the multiobjective algorithms, relative to each other and to the benchmarks, we plot the solutions returned per financial metric and aggregated across all testing strands using violin plots [16]. A violin plot extends Tukey's Box and Whisker plots by displaying a kernel density estimation of the data points along with the summary statistics using a Gaussian

TABLE III: Best Performance Per Objective including both NSGA-II and MACD. For every optimized objective, we find the best performing instance. The test strand where this is observed is in brackets next to the primary objective name. The best discovered solution per algorithm observed within the strand and objective at hand is then listed. The top performing solution is highlighted in bold. In case of a tie, we consider the objectives in a lexicographical approach using the following order: AROR, Portfolio Risk, VaR, Transactions Count and Solution Length.

Primary Objective			**λ-PSO$^{\text{SP}}$**	**λ-GA**	**λ-PSO**	**NSGA-II**	**MACD**
AROR (ATRO1)	►	**AROR**	**2.5162E+01**	1.6296E+01	1.8093E+01	1.3302E+01	1.5705E+00
		Portfolio Risk	**3.3176E+06**	2.0895E+06	2.5298E+06	2.0316E+06	4.1877E+05
		VaR	**3.9490E+05**	2.6933E+05	1.8831E+05	2.0658E+05	1.7218E+04
		Transactions Count	**7.8000E+01**	7.0000E+01	6.8000E+01	6.4000E+01	1.0000E+01
		Solution Length	**2.0000E+00**	5.7000E+01	3.6000E+01	3.2000E+01	1.0000E+00
Portfolio Risk (MGA1)		**AROR**	**2.9173E+00**	2.8743E+00	2.8743E+00	2.8000E+00	2.6536E+00
	►	**Portfolio Risk**	**0.0000E+00**	0.0000E+00	0.0000E+00	0.0000E+00	4.1339E+05
		VaR	**0.0000E+00**	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
		Transactions Count	**2.0000E+00**	2.0000E+00	2.0000E+00	0.0000E+00	4.0000E+00
		Solution Length	**1.0000E+00**	3.8000E+01	3.8000E+01	1.2000E+01	1.0000E+00
VaR (JBLU1)		**AROR**	**1.4984E+01**	1.2095E+01	1.3313E+01	1.1953E+01	2.0160E+00
		Portfolio Risk	**1.0970E+06**	1.0717E+06	1.0115E+06	1.1022E+06	0.0000E+00
	►	**VaR**	**0.0000E+00**	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
		Transactions Count	**1.0200E+02**	6.8000E+01	7.6000E+01	6.6000E+01	0.0000E+00
		Solution Length	**3.8000E+01**	3.5000E+01	5.8000E+01	3.3000E+01	1.0000E+00
Transactions Count (LUV1)		**AROR**	**-5.1663E-02**	-9.9755E-02	-8.5752E-02	-9.9213E-02	6.2546E-01
		Portfolio Risk	**4.6062E+05**	4.7781E+05	4.3228E+05	6.1172E+05	1.1358E+06
		VaR	**5.8006E+04**	6.5248E+04	6.0277E+04	8.3613E+04	3.1387E+04
		Transactions Count	**6.3400E+02**	6.1600E+02	6.1800E+02	6.2800E+02	9.2000E+01
	►	**Solution Length**	**5.1000E+01**	1.0000E+01	4.4000E+01	5.0000E+00	1.0000E+00
Solution Length (ATRO1)		**AROR**	**1.8919E+01**	3.6600E+00	1.2000E+01	9.7551E-01	1.5705E+00
		Portfolio Risk	**2.5523E+06**	1.0528E+06	1.5216E+06	9.9237E+05	4.1877E+05
		VaR	**2.9994E+05**	9.1406E+04	1.4375E+05	1.2718E+05	1.7218E+04
		Transactions Count	**7.8000E+01**	6.6000E+01	6.0000E+01	8.0000E+01	1.0000E+01
	►	**Solution Length**	**1.0000E+00**	2.0000E+00	2.6000E+01	7.0000E+00	1.0000E+00

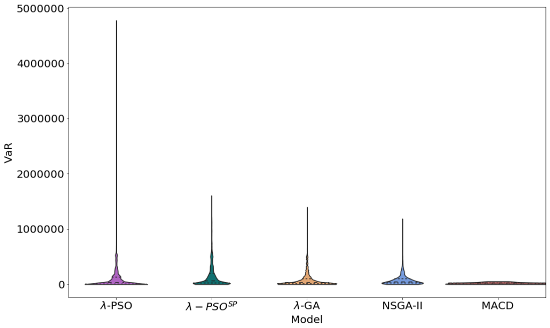


Fig. 3: Violin plot comparing VaR performance for all multi-objective algorithms, along with NSGA-II and MACD, across all testing strands. We can see that the medians and the bulk of their solution distributions for all algorithms are close to each other. λ-PSO displays an exceptionally long tail stretching into higher values of VaR indicating a higher potential for losses when compared to the other algorithms albeit at a lower probability.

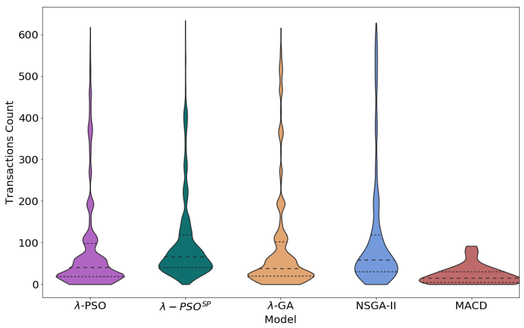


Fig. 4: Violin plot comparing transactions count for all multi-objective algorithms, along with NSGA-II and MACD, across all testing strands. The highest median observed was that of λ-PSO$^{\text{SP}}$, followed by NSGA-II, λ-PSO, λ-GA and finally MACD. With the exception of MACD, all algorithms have tails extending into higher transaction counts. Based on these observations, we can conclude that solutions returned by λ-PSO$^{\text{SP}}$ are relatively the most stable and have the lowest sample error. MACD solutions, on the other hand, are relatively the least stable and have the highest sample error compared to the solutions returned by the other algorithms.

kernel. The kernel density estimation is presented visually as the contours of the shape rendered for every category in the plot, while the summary statistics are represented by three lines rendered within the body of the shape. The dotted middle line represents the median, while the bottom and top dotted lines represent the first and third quartiles of the interquartile range. Figures 1 to 5 show the violin plots of all algorithms for AROR, Portfolio Risk, VaR, Transactions Count and Solution Length. From the AROR violin plot, Figure 1, we can observe

TABLE IV: A comparison of the AROR values between the single objective and multiobjective optimization algorithms including NSGA-II and MACD benchmarks. The single objective algorithm results are obtained from [6]. The maximum values obtained in the experiments are used since AROR is an objective that is maximized. The highest value per Test Strand is highlighted in bold.

		Algorithm							
#		Test Strand	PSO^{S}	λ-PSO^{SP}	GA	λ-GA	PSO	λ-PSO	NSGA-II	MACD
0	↑	IAG1	-3.42	0.04	-1.39	0.06	-3.04	0.97	-0.04	**3.04**
	↔	MGA4	1.70	2.08	1.60	2.14	2.09	**2.30**	1.87	2.07
	↓	IAG2	2.17	2.90	2.16	2.92	2.17	**2.95**	2.42	2.43
1	↑	BSX1	-0.11	0.63	-0.13	0.73	-0.02	1.27	1.13	**1.57**
	↔	LUV1	-0.08	0.07	-0.01	0.03	-0.04	0.07	-0.02	**0.63**
	↓	KFY1	2.17	**3.39**	2.67	2.81	2.66	3.02	3.16	2.55
2	↑	EXC1	2.92	**3.44**	2.90	3.08	2.80	3.05	2.98	2.89
	↔	LUV2	2.46	2.79	2.64	**2.92**	2.62	2.88	2.86	2.53
	↓	KFY2	2.85	6.52	2.05	3.80	3.61	**3.89**	3.42	1.87
3	↑	AVNW1	1.22	0.77	1.29	1.83	1.26	2.08	2.07	**2.77**
	↔	PUK1	0.38	0.06	0.00	**0.76**	0.00	0.64	0.40	0.33
	↓	LUV3	6.12	6.16	4.63	6.00	4.70	**7.13**	4.67	1.83
4	↑	KFY3	2.94	3.08	2.67	2.91	2.80	**3.30**	2.91	2.93
	↔	EXC2	1.62	**5.14**	1.55	2.08	1.60	2.49	2.48	1.39
	↓	LUV4	2.95	**4.89**	2.96	3.75	2.80	3.74	3.68	2.53
5	↑	EXC3	2.39	2.35	2.14	2.31	2.38	**2.55**	2.31	2.11
	↔	PUK2	0.76	2.04	0.51	2.40	1.07	**3.16**	3.09	0.99
	↓	MGA1	3.80	**7.85**	3.15	3.37	2.80	4.35	3.70	2.65
6	↑	ED1	2.32	2.58	1.98	2.75	2.44	2.84	2.59	**2.95**
	↔	EXC4	3.96	**14.27**	3.72	5.88	3.47	6.91	4.87	2.08
	↓	PUK3	3.66	**4.53**	3.66	3.95	2.80	3.64	3.24	3.22
7	↑	BSX2	3.76	**4.33**	4.03	4.15	3.32	**4.33**	3.92	2.54
	↔	ED2	2.67	**2.82**	2.70	2.71	2.63	2.75	2.75	2.24
	↓	JBLU1	13.09	**14.98**	12.06	12.63	11.95	13.43	11.95	2.01
8	↑	MGA2	0.00	0.09	-3.39	0.44	-0.04	0.68	0.20	**1.93**
	↔	MGA3	1.34	**1.79**	0.51	0.79	0.48	1.60	0.81	0.91
	↓	ATRO1	11.51	**25.16**	16.08	16.30	11.31	18.09	13.30	1.57
9	↑	AVNW2	5.67	12.02	5.65	8.60	3.99	10.88	**11.48**	1.60
	↔	EXC5	0.56	3.54	0.96	**3.85**	0.87	3.42	3.17	0.69
	↓	AVNW3	2.20	**4.79**	2.14	4.74	2.10	3.33	3.54	1.75

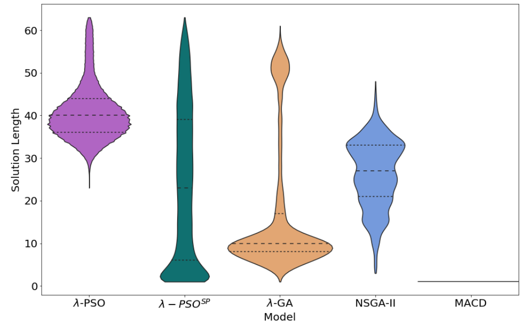


Fig. 5: Violin plot comparing solution length for all multiobjective algorithms, along with NSGA-II and MACD, across all testing strands. When comparing λ-PSO to λ-PSO^{SP}, we can see that the pruning procedure has resulted in comparatively shorter solutions based on the lower median and the main body of the solution distribution manifesting significantly lower than λ-PSO. When considering the other algorithms, we can see that λ-PSO^{SP} has achieved shorter solutions and are directly comparable to the MACD solution which had a fixed length of one component.

that the medians and main bulk of the solution distributions for all algorithms are close to each other. The PSO algorithms display interesting behavior in having long tails extending well beyond those of their counterparts: λ-PSO^{SP} having a tail extending higher than rest showing the potential of achieving higher returns and λ-PSO extending significantly lower than its counterparts showing a higher potential of negative returns albeit at a lower probability. The Portfolio Risk violin plot, Figure 2, shows that all the algorithms in the comparison have the bulk of their distribution and medians around the same level. This is with the exception of MACD that has a higher interquartile range and a long tail stretching into higher values of risk, indicating a worse performance than the other algorithms in the comparison. As for VaR, we can see from the associated plot, Figure 3, that all algorithms have their medians and the main bulk of their distributions close to each other. Only λ-PSO displays a long tail stretching into higher values of VaR indicating a higher potential for losses albeit at a low probability. Figure 4 shows the violin plot for transactions count and from that we can observe that the highest median was achieved by λ-PSO^{SP}, followed by NSGA-II, λ-PSO, λ-GA and MACD. All algorithms, bar MACD, have tails that extend higher into transaction counts. Based on the observations, we can surmise that λ-PSO^{SP} returns the most stable solutions and MACD returns the least stable. Finally, Figure 4 shows the violin plot for solution length. By comparing λ-PSO to λ-PSO^{SP}, we can see that the pruning procedure has resulted in comparatively shorter solutions as evident by the lower interquartile range of λ-PSO^{SP} and the manifestation of its main body of its solution distribution at a significantly lower level than λ-PSO. We can also observe that λ-PSO^{SP} has achieved shorter solutions to its counterparts and is comparable to MACD which had a fixed solution length of one component.

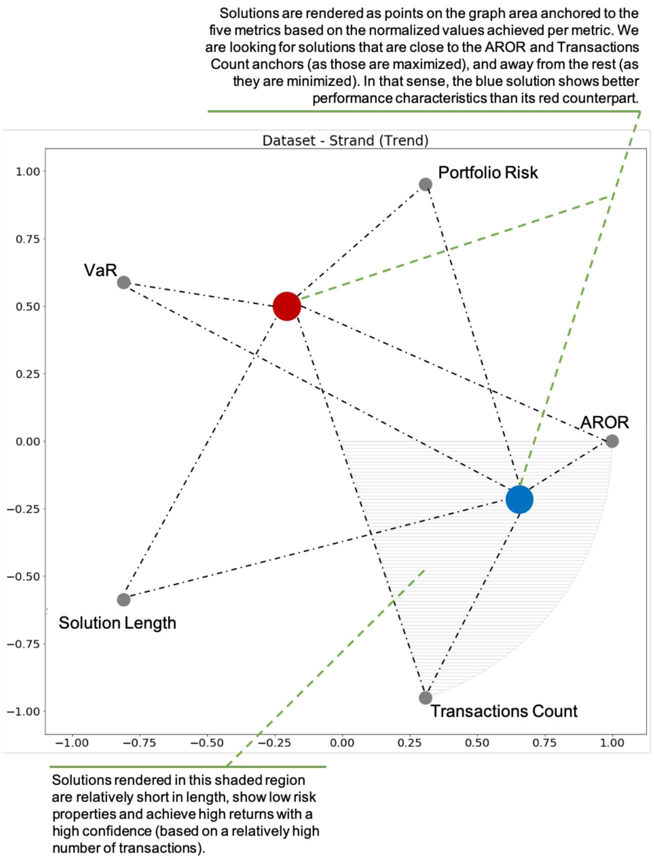
As with the previous set of experiments, we reconducted the Friedman non-parametric test with the Holm correction on the mean hypervolumes to check if there are statistical significance differences. The results can be seen in Table V. From Table V, we can see that NSGA-II and MACD have performed worse than all other multiobjective optimization algorithms across all trend types. In particular, both NSGA-II and MACD performed statistically significantly worse than the control algorithm across all trend types. This suggests that our algorithms are better suited than NSGA-II and MACD when tackling market timing as a multiobjective optimization problem. We can see that λ-PSO^{SP} and our multiobjective GA have consistently ranked in the top two across each trend type, and shows a slight edge in ranking when considering all trends, albeit in a statistically non-significant manner.

TABLE V: Average rankings of each algorithm according to the Friedman non-parametric test with the Holm post-hoc test over the mean hypervolume. Although λ-PSO^{SP} lost its edge over λ-GA when compared in a wider context containing both NSGA-II and MACD, it retains a statistically significant edge over PSO and the two additional benchmarks. NSGA-II and MACD have performed worse than all other algorithms under various trend conditions in a statistically significant manner.

Trend	Algorithm	Ranking	p-value	Holm
Uptrend	λ-GA (control)	1.8	–	–
	λ-PSO^{SP}	1.8	1.0	0.05
	λ-PSO	2.4	0.39614	0.025
	NSGA-II	**4.05**	**0.00156**	**0.0167**
	MACD	**4.95**	**8.3982E-6**	**0.0125**
Sideways	λ-GA (control)	1.5	–	–
	λ-PSO^{SP}	2.1	0.3961	0.05
	λ-PSO	2.5	0.1573	0.025
	NSGA-II	**4.35**	**5.5656E-5**	**0.0167**
	MACD	**4.55**	**1.608E-5**	**0.0125**
Downtrend	λ-PSO^{SP} (control)	1.0	–	–
	λ-GA	2.0	0.1573	0.05
	λ-PSO	**3.2**	**0.0019**	**0.025**
	NSGA-II	**4.4**	**1.522E-6**	**0.0167**
	MACD	**4.4**	**1.522E-6**	**0.0125**
All	λ-PSO^{SP} (control)	1.3	–	–
	λ-GA	1.8	0.4795	0.05
	λ-PSO	**2.9**	**0.0237**	**0.025**
	NSGA-II	**4.2**	**4.1098E-5**	**0.0167**
	MACD	**4.8**	**7.431E-7**	**0.0125**

IV. Diversity of the Pareto Front

In the previous section, we addressed the first limitation in the results obtained by our multiobjective optimization algorithms by comparing them with NSGA-II and MACD as benchmarks. In this section, we address the second limitation which was lack of insight into the diversity of the solutions in the returned Pareto sets and how they are spread across their respective Pareto fronts. The more diversity in the solutions contained within a given Pareto set, the more spread they are across their respective Pareto front giving the end user more choice to select solutions that better suit their needs. In order to get insight into the diversity of the Pareto sets returned by the multiobjective optimization algorithms, we plot their results against each test strand using RadViz [11]. RadViz is a method of adapting a scatter plot to display multivariate data containing more than two dimensions. The various dimensions being plotted in a RadViz diagram are rendered as anchors distributed equally across the circumference of a circle containing points representing the dataset being visualized. Each point in the dataset is rendered as a point in this circle tethered to each of the dimensional anchors on the circumference on the plot area. The amount of tension in each tether is commensurate to the normalized value each point has to a given dimensional anchor and the points are rendered within the plot area, where they reach an equilibrium across all the tensions contained within the tethers. In our case, we have five dimensional anchors: AROR, Portfolio Risk, VaR, Transactions Count and Solution Length. As Portfolio Risk, VaR and Solutions Length are metrics that are minimized, we are interested in points that are situated far from the anchors associated with those metrics. The opposite is true for AROR and Transactions Count, where we are interested in points that are rendered close to those dimensional anchors. An example of a RadViz diagram and how we can interpret it can be seen in Figure 6. By observing the RadViz plots, we can visually identify which algorithms attained better diversity by looking at the spread of their corresponding Pareto sets. We use RadViz to plot the Pareto sets returned by λ-PSO^{SP}, λ-PSO, λ-GA, NSGA-II and MACD for each testing strand, and a selection of plots can be seen in Figure 7.


Solutions are rendered as points on the graph area anchored to the five metrics based on the normalized values achieved per metric. We are looking for solutions that are close to the AROR and Transactions Count anchors (as those are maximized), and away from the rest (as they are minimized). In that sense, the blue solution shows better performance characteristics than its red counterpart.

Solutions rendered in this shaded region are relatively short in length, show low risk properties and achieve high returns with a high confidence (based on a relatively high number of transactions).

Fig. 6: RadViz Interpretation

From the RadViz plots, we can see that the λ-PSO^{SP} algorithm consistently achieves the largest spread across the plot area, covering all areas covered by the other multiobjective optimization algorithms. This is despite the fact that no explicit measures where taken to promote diversity, such as the crowding distance measure used in NSGA-II. When present, MACD has consistently existed outside the main body

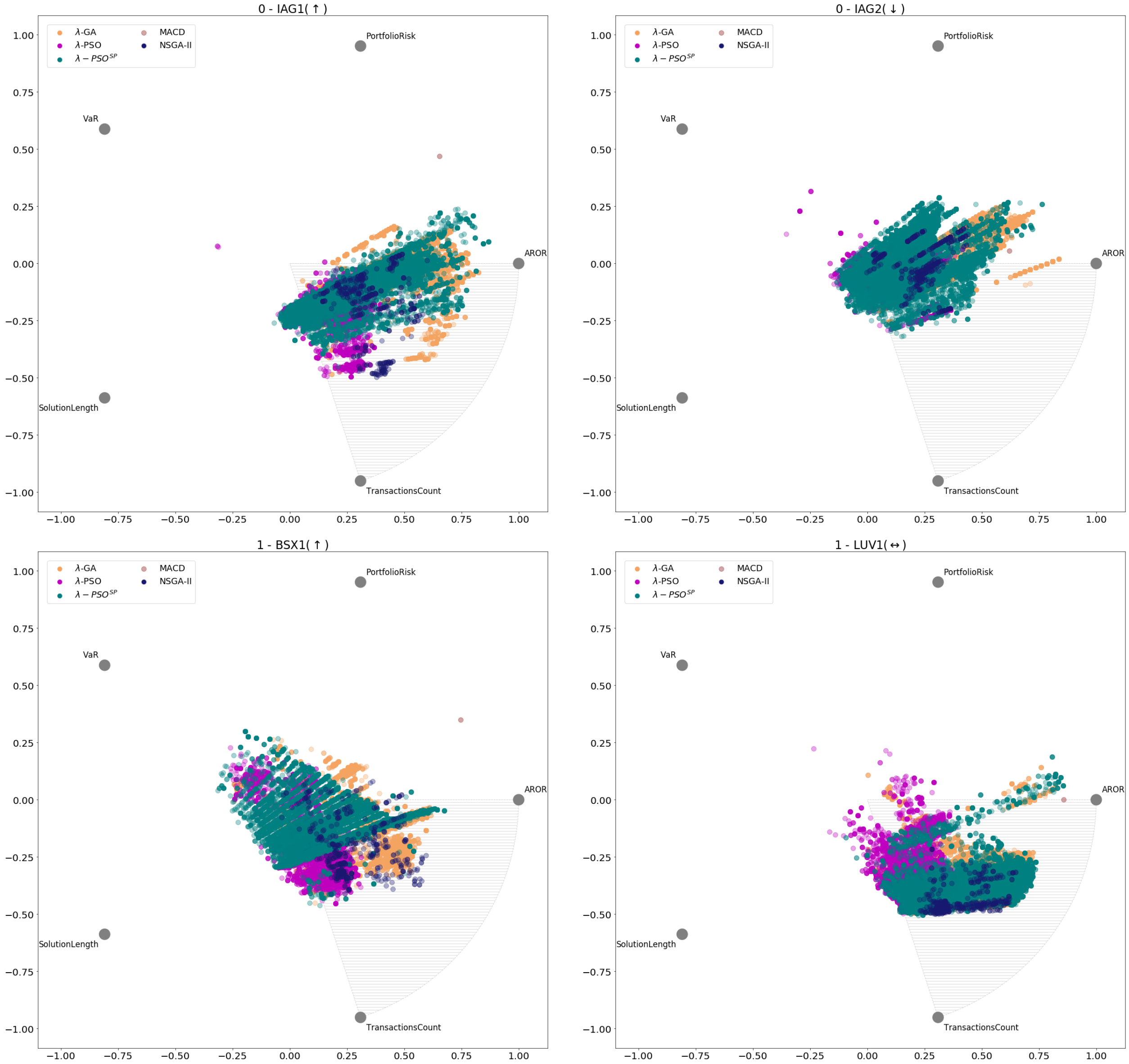


Fig. 7: A sample of RadViz plots.

occupied by the Pareto sets of the other algorithms, in an area to the upper right of the plotting area. This implies that the solution presented by MACD can achieve good values of AROR but at the expense of high risk (attraction to the Portfolio Risk and VaR anchors) and low confidence (repulsion from the Transactions Count anchor). The NSGA-II Pareto sets are smaller in size when compared to the ones generated by the other algorithms, with the exception of BSX1 (↑), where the NSGA-II Pareto set always lies to the right of the PSO Pareto sets and to the left of both λ-GA and λ-PSO$^{\text{SP}}$. This implies that the λ-GA and λ-PSO$^{\text{SP}}$ have a higher probability of landing in the desired quadrant of solutions with high profitability, low risk and high confidence – the quadrant indicated by gray shading in the plots. The algorithm that shows the least diversity is the PSO algorithm, resulting in Pareto sets that are relatively tightly clustered near the center of the plotting area. Based on the RadViz visualizations of the algorithms and the observations seen in the results earlier, we can observe that the multiobjective λ-PSO$^{\text{SP}}$ algorithm returns Pareto sets with a high level of diversity and those Pareto sets contain competent solutions across all the metrics being considered. This is followed closely by λ-GA, then by λ-PSO, NSGA-II and MACD, respectively.

V. Conclusions and Future Research

In this paper, we have expanded on the work introduced in [10] by providing a deeper insight into the PSO algorithms there and addressing two limitations of the results: a limited comparative context for the PSO algorithms and lack of insight into the diversity of the Pareto sets returned by the algorithms.

In order to provide a better comparative context for the algorithms, we compared their performance against NSGA-II (a well established multiobjective optimization algorithm) and MACD (a widely used technical indicator in market timing applications). Results from the comparison showed that NSGA-II and MACD performed statistically significantly worse than λ-PSO, λ-GA and λ-PSO$^{\text{SP}}$ across all of the three trend types based on mean hypervolume. The Pareto sets returned by the algorithms for each testing strand was visualized using RadViz in order to gain insight about the diversity of those Pareto sets. The RadViz plots show that λ-PSO$^{\text{SP}}$ displayed the most diversity and its solutions were present in all areas of the plot that the Pareto sets of the other algorithms occupied. The least diverse algorithm was λ-PSO, consistently producing Pareto sets that occupied the center of the plotting area in one contiguous body.

The research presented here can be extended in two ways: approaches where we extend the capabilities of the algorithms or approaches where we expand the scope of how market timing is tackled. Examples of approaches extending the capabilities of the algorithms discussed in this paper include expanding the set of financial metrics being optimized, considering the diversity of the returned Pareto sets as a first class objective to be optimized along with the rest of the objectives under consideration, considering optimization for each trend type (akin to a niching approach) and using fundamental analysis components along with the technical analysis ones used in this research. As for exppanding the scope of how market timing is algorithmically tackled, examples of approaches would include the application of other algorithms besides PSO and then use Meta-learning to arrive at an ensemble approach, including other aspects of algorithmic trading to optimized by the algorithms (such as portfolio optimization and execution optimization) and to consider dynamic optimization whereby the algorithms are capable of detecting degradation in performance and adapting accordingly.

References

[1] Y. Hu, K. Liu, X. Zhang, L. Su, E. W. T. Ngai, and M. Liu, "Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review," *Applied Soft Computing*, vol. 36, pp. 534–551, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2015.07.008

[2] A. Soler-Dominguez, A. A. Juan, and R. Kizys, "A Survey on Financial Applications of Metaheuristics," *ACM Computing Surveys*, vol. 50, no. 1, pp. 1–23, 2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=3054133{%}5Cnhttp://dl.acm.org/citation.cfm?doid=3058791.3054133

[3] H. Subramanian, S. Ramamoorthy, P. Stone, and B. Kuipers, "Designing Safe, Profitable Automated Stock Trading Agents Using Evolutionary Algorithms," in *Genetic and Evolutionary Computation Conference*, vol. 2, 2006, p. 1777. [Online]. Available: http://eprints.pascal-network.org/archive/00004834/

[4] A. C. Briza and P. C. Naval Jr., "Stock trading system based on the multi-objective particle swarm optimization of technical indicators on end-of-day market data," *Applied Soft Computing*, vol. 11, no. 1, pp. 1191–1201, 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1568494610000621

[5] I. Mohamed and F. E. B. Otero, "Using Particle Swarms to Build Strategies for Market Timing: A Comparative Study," in *Swarm Intelligence: 11th International Conference, ANTS 2018, Rome, Italy, October 29–31, 2018, Proceedings*. Springer International Publishing, 2018, pp. 435–436.

[6] I. Mohamed. and F. E. B. Otero., "Using population-based metaheuristics and trend representative testing to compose strategies for market timing," in *Proceedings of the 11th International Joint Conference on Computational Intelligence - Volume 1: ECTA, (IJCCI 2019)*, INSTICC. SciTePress, 2019, pp. 59–69.

[7] P. J. Kaufman, *Trading Systems and Methods*, 5th ed. John Wiley & Sons, Inc, 2013.

[8] C. M. Fonseca, L. Paquete, and M. López-Ibáñez, "An improved dimension-sweep algorithm for the hypervolume indicator," in *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*. Piscataway, NJ: IEEE Press, Jul. 2006, pp. 1157–1163.

[9] N. Beume, C. M. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold, "On the complexity of computing the hypervolume indicator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1075–1082, 2009.

[10] I. Mohamed and F. E. B. Otero, "A multiobjective optimization approach for market timing," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, ser. GECCO '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 22–30. [Online]. Available: https://doi.org/10.1145/3377930.3390156

[11] P. Hoffman, G. Grinstein, and D. Pinkney, "Dimensional anchors: A graphic primitive for multidimensional multivariate information visualizations," in *Proceedings of the 1999 Workshop on New Paradigms in Information Visualization and Manipulation in Conjunction with the Eighth ACM Internation Conference on Information and Knowledge Management*, ser. NPIVM '99. New York, NY, USA: Association for Computing Machinery, 1999, p. 9–16.

[12] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 3, 1999, pp. 1951–1957.

[13] I. Mohamed and F. E. B. Otero, "Building market timing strategies using trend representative testing and computational intelligence metaheuristics," in *Computational Intelligence, 11th International Joint Conference, IJCCI 2019 Vienna, Austria, September 17-19, 2019, Revised Selected Papers*. Springer, 2021.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002.

[15] M. Pring, *Technical Analysis Explained*. McGraw-Hill, 2002.

[16] J. L. Hintze and R. D. Nelson, "Violin plots: A box plot-density trace synergism," *The American Statistician*, vol. 52, no. 2, pp. 181–184, 1998.