Tabu Search for Portfolio Optimization

Topic 5: Extension of Local Search

Yi-Ping Tseng
Uni: yt2690

Introduction

Tabu Search is one of the metaheuristic algorithms that is widely used. It is useful for solving combinatorial optimization problems and can be applied to both discrete and continuous space. In this project, our goal is to apply Tabu Search on constructing a stock portfolio from S&P 500 component stocks. The purpose of building a portfolio is to achieve maximum returns with minimum risks in a given period. The relationships in return and risk are apparent. It is common to see that high return corresponds with high risk. Therefore, the balance for high return and low risk is a complicated optimization problem. We will evaluate our optimized portfolio on two given period: extreme period 2020-01-01~2020-04-17 and standard period 2015-01-01~2015-12-31.

Literature Review

One study of this particular application is the work of Majid M. Aldaihani and Talla M. Aldeehani. They applied two forecasting models, which are optimized by Tabu Search on building stock portfolios in the Kuwait Stock Exchange. The result of their research shows Tabu Search was able to successfully solve models for balancing the tradeoff between risks and returns. They successfully constructed several stock portfolios to outperform the performance of the KSE market index.

Methodology

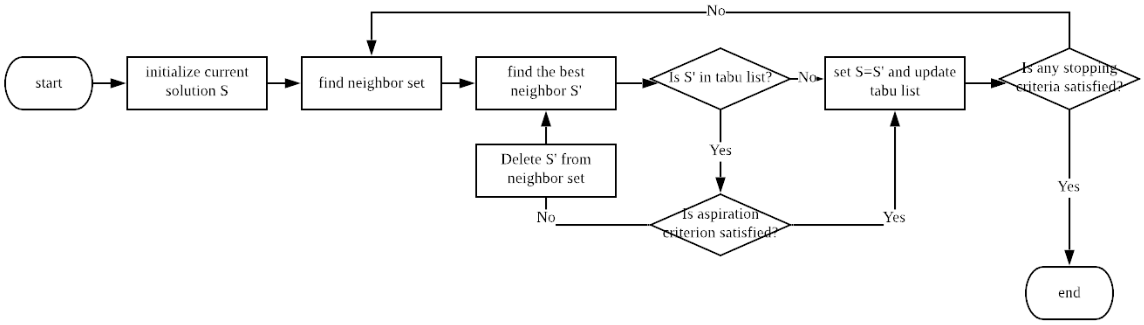
In this part, we are going to review the optimization process of Tabu Search and give a clear general structure of Tabu Search. Since it is one of the metaheuristic algorithms, it possesses some common properties. First of all, metaheuristic algorithms are strategies that guide searching processes. Secondly, it can find a solution that is good enough, but the result is not guaranteed to be the global optimum. Thirdly, metaheuristic algorithms implement some form of stochastic optimization, so the results are highly dependent on the path of optimization. The advantages of this kind of algorithms are straightforward. It is more efficient in exploring the search space, and its usage is not problem-specific. According to Wiki, Tabu Search employs local search methods for optimization, and it enhances the performance by relaxing the basic rule of local search, which is conditionally accepting the worse solution. Several components are essential in Tabu Search.

a. Tabu List: it is a set of solution which are not allowed to be used anymore. The forbidden solutions are called Tabu move.
b. Memory Size: it is the size of Tabu List. Tabu Search will not return to the same answer is because it memorizes the path it moves.
c. Neighbors and Neighborhood Sets: neighbors are all the feasible solutions that can be reached from the current solution, and they form the neighborhood sets. Neighborhood sets provide the next possible solution for each step.
d. Stopping Criteria: because Tabu Search is a metaheuristic algorithm, it is a stochastic searching process. It is controlled by stopping criteria to stop searching for new solutions. Some standard stopping criteria for Tabu Search are maximum iteration times, the size of neighborhood sets (usually set to zero), and early stop for no improvement in the last given steps.

Three strategies control the optimization path of Tabu Search.

a. Forbidding Strategy: it controls what enters the Tabu List. It is usually set to add the path from the current solution to the next one into the Tabu List. This strategy can prevent the algorithm from returning to the previous solution.
b. Freeing Strategy: it controls what exits the Tabu List. It is usually set to remove the earliest path in the Tabu List when it is full. This strategy allows the algorithm to explore more search space.
c. Short-term strategy: it manages the interplay between forbidding and freeing strategy to select a trial solution. For example, 'aspiration criterion' is a criterion that provides an exception to Tabu restrictions for the element inside the Tabu List. It is satisfied when a tabu move has an attractive evaluation.

The flowchart and pseudocode of Tabu Search are provided below. The algorithm starts with randomly select a feasible solution as an initial and current solution. Then, it finds the best-performed neighbor from all the feasible neighbors of the current solution. After checking the tabu restrictions (the next solution should not be in the Tabu List), we update the current solution and Tabu List. At the end of each iteration, the algorithm stops if any stopping criteria are satisfied. The aspiration criterion gives an exception for the forbidden move when it improves the current solution at least a given level. Otherwise, the forbidden move cannot be the next solution and will be deleted from the neighborhood set.




```
sBest ← s0
bestCandidate ← s0
tabuList ← []
tabuList.push(s0)
while (not stoppingCondition())
    sNeighborhood ← getNeighbors(bestCandidate)
    bestCandidate ← sNeighborhood[0]
    for (sCandidate in sNeighborhood)
        if ( (not tabuList.contains(sCandidate)) and (fitness(sCandidate) > fitness(bestCandidate)) )
            bestCandidate ← sCandidate
        end
    end
    if (fitness(bestCandidate) > fitness(sBest))
        sBest ← bestCandidate
    end
    tabuList.push(bestCandidate)
    if (tabuList.size > maxTabuSize)
        tabuList.removeFirst()
    end
end
return sBest
```

Problem Formulation

The objective of building a portfolio is to maximize returns by selecting stocks. And, there are several constraints to restrict the market risks.

1) The number of stocks: we should select at least ten stocks to at most 30 stocks to ensure diversity.
2) The number of stocks in each sector: MSCI classifies all industries into 11 sectors (Energy, Materials, Industrials, Consumer Discretionary, Consumer Staples, Health Care, Financials, Information Technology, Communication Services, Utilities, and

Real Estate). To prevent the portfolio from selecting the same kind of stocks, we restrict it to pick at most 6 stocks from the same industry sector.

3) Correlation: sometimes, stocks may have implicit correlations. To avoid any domination in our portfolio, it is required to limit the correlation. In this portfolio, we restrict the average mutual absolute correlation should not exceed an upper bound. The average mutual absolute correlation is calculated by adding all absolute correlation and then divide by the number of selected stocks.
4) Value at Risk (VaR): it is a standard indicator to evaluate risks. It estimates how much an investor may lose based on the normal theory. The value of VaR means a possible most significant loss at a 95% confidence level. Therefore, we limit our stocks' VaR not to lower than a given lower bound, and the average VaR of the portfolio should not lower than another given lower bound.

Our portfolio optimization problem is an integer programming problem. Hence, it can be formulated in integer programming formulation. Before the details of mathematical formulation, we summarize all notations that are used in our model.

$$x_{s,i} = \begin{cases} 1, \; if\ stock\ i\ in\ sector\ s\ is\ selected \\ \qquad\qquad\qquad 0, \; ow \end{cases}$$

$r_{s,i} = the\ return\ of\ holding\ stock\ i\ in\ sector\ s\ in\ training\ time$

$c_{i,j}^{s,g} = correlation\ between\ stock\ i\ in\ sector\ s\ and\ stock\ j\ in\ sectorg$

$v_{s,i} = the\ VaR\ of\ stock\ i\ in\ sector\ s\ in\ training\ time$

$M = set\ of\ sectors\ in\ stock\ market$

$M_s = set\ of\ stocks\ in\ sector\ s$

$UB_\rho = the\ upper\ bound\ of\ average\ correlation$

$LB_{Tot} = the\ lower\ bound\ of\ the\ number\ of\ total\ selected\ stocks$

$UB_{Tot} = the\ upper\ bound\ of\ the\ number\ of\ total\ selected\ stocks$

$UB_{Sec} = the\ upper\ bound\ of\ the\ number\ of\ selected\ stocks\ in\ each\ sector$

$LB_{VaR,avg} = the\ lower\ bound\ of\ the\ average\ VaR$

$LB_{VaR,ind} = the\ lower\ bound\ of\ the\ individual\ VaR$

$$\max \sum_{s\in M} \sum_{i\in M_s} r_{s,i} x_{s,i} \tag{1}$$

$$\sum_{s\in M} \sum_{i\in M_s} x_{s,i} \geq LB_{Tot} \tag{2}$$

$$\sum_{s\in M} \sum_{i\in M_s} x_{s,i} \leq UB_{Tot} \tag{3}$$

$$\sum_{i\in M_s} x_{s,i} \leq UB_{Sec} \quad \forall s \in M \tag{4}$$

$$\sum_{s\in M} \sum_{i\in M_s} x_{s,i} v_{s,i} \geq LB_{VaR,avg} \sum_{s\in M} \sum_{i\in M_s} x_{s,i} \tag{5}$$

$$x_{s,i} v_{s,i} \geq LB_{VaR} \quad \forall i \in s, \forall s \in M \tag{6}$$

$$\sum_{s=1}^{M} \sum_{g\geq s}^{M} \sum_{i\in M_s} \sum_{j\in M_g} x_{s,i} x_{g,j} \left|c_{i,j}^{s,g}\right| \leq UB_\rho \sum_{s=1}^{M} \sum_{g\geq s}^{M} \sum_{i\in M_s} \sum_{j\in M_g} x_{s,i} x_{g,j} \tag{7}$$

(1) means to maximize our return by selecting a group by stocks. (2) and (3) means the total number of stocks should be at least LB_{Tot} and at most UB_{Tot}. In our experiment, we set LB_{Tot} to 10 and UB_{Tot} to 30. (4) means the maximum number of selected stocks in each sector cannot exceed UB_{Sec}. In our experiment, we set UB_{Sec} to 6. (5) means the average VaR of our portfolio should be at least larger than $LB_{VaR,avg}$. In our experiment, we set $LB_{VaR,avg}$ to -0.03. (6) means the individual VaR of each stock should be at least larger than LB_{VaR}. In our experiment, we set LB_{VaR} to -0.05. (7) means the average absolute correlation of all paired stocks should not exceed UB_ρ. In our experiment, we set UB_ρ to 0.5.

Neighbors of a current solution have three kinds of patterns. For example, if the stock that x = 1 is the selected stocks in the current solution.

Stock	AMGN	ADI	BLL	BK	BBY	CTL	CMA	DOV	ECL
x	1	0	0	0	1	0	0	0	1

Then, it's neighbors can be

a. Removing one of the selected stocks from the portfolio, i.e., removing AMGN makes the neighbor portfolio – (BBY, ECL)
b. Adding one of the unselected stocks to the portfolio, i.e., adding BLL makes the neighbor portfolio – (AMGN, BBY, BLL, ECL)
c. Removing one of the selected stocks from the portfolio and adding one of the unselected stocks to the portfolio at the same time, i.e., removing ECL and adding BK to the portfolio makes the neighbor portfolio – (AMGN, BK, BBY)

Because there are too many possible neighbors for each solution, we modify collecting neighborhood set. We sample the neighbors from the neighborhood set and make a small subset of the original neighborhood set. We then find the best neighbor from it and continue the Tabu Search process.

Dataset

In our experiment, we use the S&P 500 components stocks. Our training period starts on 2016-01-01 and ends on 2019-01-01. We evaluate our result on two testing set: (1) extreme period 2020-01-01~2020-04-17 (the huge drop is on 2020-02-21) and (2) normal period 2015-01-01~2015-12-31.

Because several new component stocks don't cover the training period, they are eliminated out of our stock list. Hence, there are 488 stocks available in our stock list.

The evaluation of a portfolio is to buy all selected stocks in one unit from the beginning of the period and sell all of them at the end of the period. Then, based on the buying price and selling price, we can calculate the total return of the portfolio. To compare with other portfolios, we calculate the return percentage since a percentage is based on the difference between the starting and ending price. It is compatible with others by comparing their return percentages.

The names of the main sectors and the number of companies in each one are shown below.

Sector	Sector Code	# of Company
Energy	10	25
Materials	15	25
Industrials	20	67
Consumer Discretionary	25	62
Consumer Staples	30	32
Health Care	35	60
Financials	40	66
Information Technology	45	71
Communication Services	50	21
Utilities	55	28
Real Estate	60	31
Total number of companies		488

Experiment

Below is one of the Tabu Search searching process and details about it (this part of codes is in tabu_search/tabu-search.ipynb). This is the partial selection of Tabu Search.

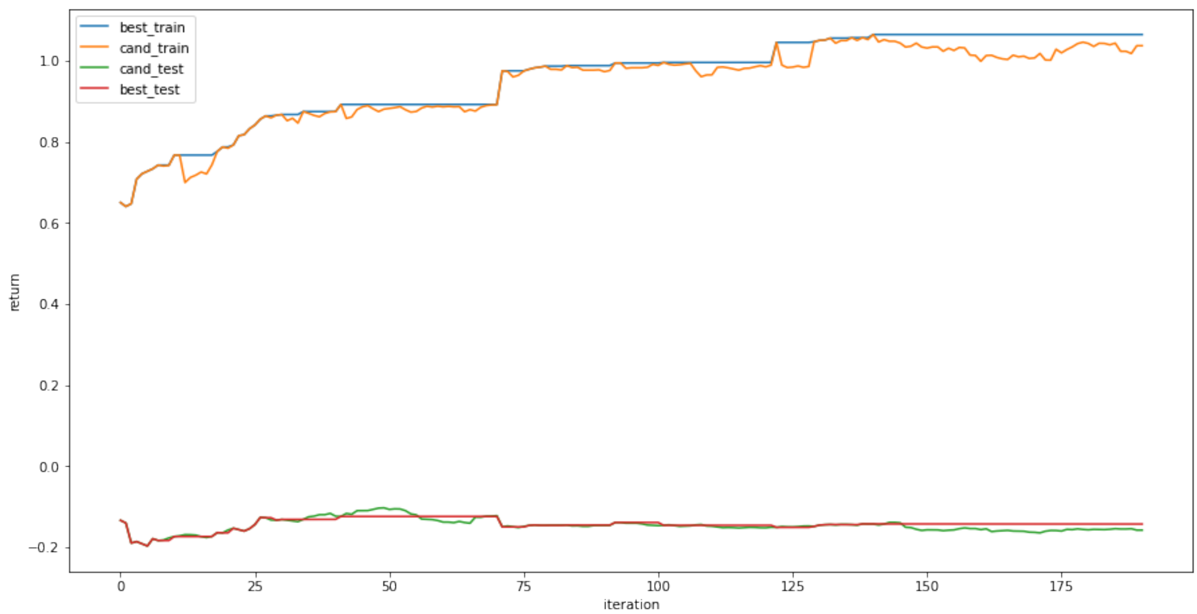
```
pd.set_option('display.max_columns', None)
tryOne['best_series'].to_frame(name='stock').T
```

	PNW	APD	HES	AFL	AMGN	ADI	APA	AMAT	ADSK	AVY	BLL	BK	VZ	BBY	HRB	CAH	CTL	CMA	C	CMS	DHR	DOV	OMC	PKI	ECL
stock	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

And the selected stocks in this searching process are below.

UNH, AZO, HUM, FISV, NVDA, MA, GOOGL, D, IP, SHW, EIX, AON, TFX, NVR, MTD, CHTR, ALGN, TDG, LKQ, MKTX, KMI, LRCX, UHS, ORLY, MLM, VTR, BKNG, EQIX, NFLX, CMG

In our experiment, we buy one unit of selected stocks in each portfolio. For example, we buy one unit of **UNH, AZO, HUM, FISV, NVDA, MA, GOOGL, D, IP, SHW, EIX, AON, TFX, NVR, MTD, CHTR, ALGN, TDG, LKQ, MKTX, KMI, LRCX, UHS, ORLY, MLM, VTR, BKNG, EQIX, NFLX, CMG** in the starting period and sell them at the end of the period. The training result is 106.54%, and the testing result is -14.33%. And the Tabu Search searching process is below. The blue line represents the performance of best training portfolio, the orange line represents the training performance of current solution portfolio during each step in the Tabu Search, the red line represents the performance of best training portfolio on the testing set, and the green line represents the performance on the testing set of current solution portfolio in each step.



The parameters for this Tabu Search are **tabu_list_size** = 20, **iterations_times** = 1000, **early_stop** = 50, **neighbor_size** = 50 and **asp_improve_level** = 1.

1. tabu_list_size: the size of Tabu List represents the memory of Tabu Search.
2. asp_improve_level: aspiration criterion is satisfied when the tabu (forbidden) move improves at least one percent than the current solution.
3. neighbor_size: this is the sampling size of the neighborhood set.
4. iterations_times: it is one of the stopping criteria. The Tabu Search stops when it reaches 1000 iteration times.
5. early_stop: it is one of the stopping criteria. The Tabu Search stops when the last 50 iterations don't improve the best solution.

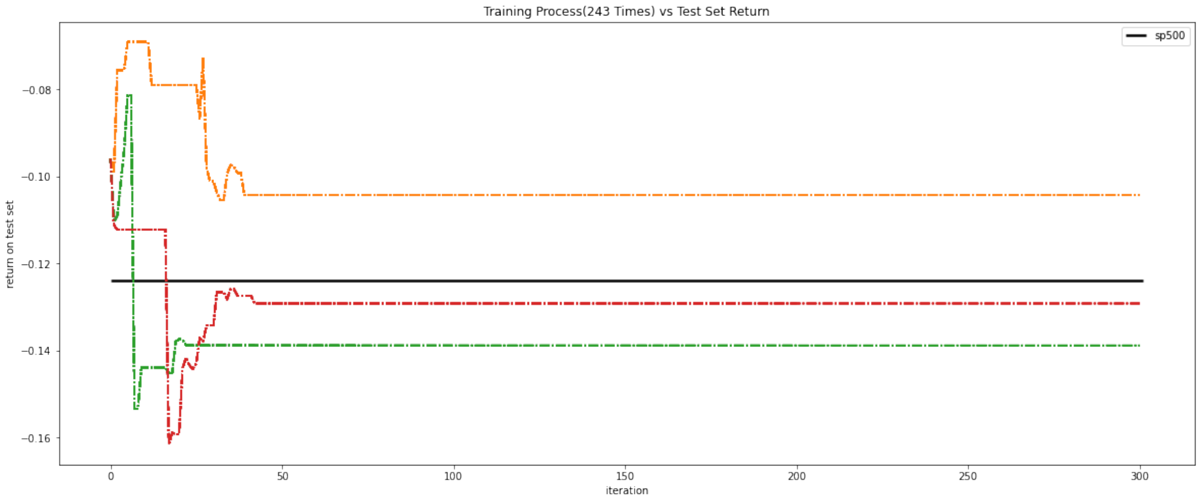
In our experiment, our goal is to test what kind of parameter combination suits the portfolio optimization problem. We use Grid Search on parameter space, and all the combinations are below.

tabu list size	10	30	50
Max iterations	100	200	300
Early stop	20	50	Inf (never stop early)
Aspiration level	-1%	1%	5%
Neighbor size	50	100	500

Firstly, different sizes of Tabu List can test whether the larger memory size improves the Tabu Search. Secondly, different stopping criteria can check whether strict stopping criteria get a solution as good as mild stopping criteria. And, different aspiration level shows how they improve the Tabu Search. Then, different size of sample neighbors shows the importance of neighbors to the final solution.

The result of our experiment shows the parameter combination does not make a significant difference in the return percentage of a portfolio. The research shows the Tabu Search will converge to three portfolios in training no matter what kind of parameters it gets. The plot shows all Tabu Search converge to three portfolios, which have -10.42%, -12.91%, and -13.88% return percentage in 2020, respectively. Although our testing result in 2020 shows that there is no significant difference in return rate between the S&P 500 and the portfolios Tabu Search found, however, the portfolios perform significantly better than S&P 500 in 2015.

We compare the result of Tabu Search with Gurobi. But it seems the portfolio optimization is too complex to find the global optimum for Gurobi, it ran for three days and still could not find the optimum portfolio, so I shut it down. The lp and log can be found in the code repository under the gurobi folder.



Year	mean	std	min	25%	50%	75%	max	S&P 500
2016-2019	0.9290	0.0732	0.8432	0.8432	0.9222	1.0217	1.0217	0.5774
2020	-0.1241	0.0146	-0.1388	-0.1388	-0.1291	-0.1042	-0.1042	-0.1239
2015	0.0595	0.032	0.0156	0.0156	0.0725	0.0905	0.0905	0.0098

Conclusion

Tabu Search can be applied successfully on the discrete problem, and it is more practical on a complex optimization problem. Although there is no significant difference in the result of different Tabu Search parameters, it can provide the same return percentage as the S&P 500 in bad time and give almost ten times better return percentage than the S&P 500 in the standard period. However, there is some room for improvement. We can add some future predictions into the training set to evaluate stocks more precisely.

Reference

Aldaihani, Majid M., and Talla M. Aldeehani. "Portfolio optimization models and a tabu search algorithm for the Kuwait Stock Exchange." Investment management and financial innovations 5, Iss. 2 (2008): 30-39.
Glover, Fred, and Eric Taillard. "A user's guide to tabu search." Annals of operations research 41.1 (1993): 1-28.
Busetti, Franco. "Metaheuristic approaches to realistic portfolio optimization." arXiv preprint cond-mat/0501057 (2005).
IIT Kharagpur (2018). *Tabu Search.* https://www.youtube.com/watch?v=A7cTp1Fhg9o&t=1580s
Wiki. *Metaheuristic*. https://en.wikipedia.org/wiki/Metaheuristic
Wiki. *Tabu Search*. https://en.wikipedia.org/wiki/Tabu_search
Wiki. *Value at Risk*. https://en.wikipedia.org/wiki/Value_at_risk
MSCI. *The Global Industry Classification Standard*. https://www.msci.com/gics
Dataset comes from Wharton Database: https://wrds-www.wharton.upenn.edu/

My Github repository: https://github.com/r50206v/E4008-Computational-Discrete-Optimization/tree/master/Project
(might modify the algorithm in the future: parallel computing and adding the features suggested in the conclusion)