ANT COLONY OPTIMIZATION FOR INDEX FUND PROBLEM

Ashutosh Nigam, Fellow Student, Decision Science Department, IIM Lucknow 226013, India
Email: nigam.ashutosh@iiml.ac.in Phone: +91-9936009555
Yogesh K. Agarwal, Professor, Decision Science Department, IIM Lucknow 226013, India

ABSTRACT

Index Fund Problem (IFP) frequently arises in the financial optimization, particularly in the field of Portfolio management. The index fund strategy tries to replicate the movement of benchmark indices such as S&P500, NIFTY50, and NASDAQ. There have been empirical evidences to suggest that index funds over-perform most of the actively managed stock portfolios. IFP is defined as following: given a set of n stocks in the market index, the objective is to select $m \leq n$ stocks and their corresponding weights in the index fund, such that the weighted portfolio imitates the corresponding market index itself. The problem is combinatorial in nature and is computationally hard to solve optimally, however there are number of polynomial time approximate algorithms. In this paper we propose an effective algorithm based on the Ant Colony Optimization (ACO) and support the efficiency of the algorithm with the help of computational results.

Keywords: Index fund; Metaheuristic; Ant Colony Optimization; Portfolio

INTRODUCTION

Index fund strategy is quite popular among the portfolio managers who try to replicate the behavior of benchmark indices (e.g. S&P500, NASDAQ, BSE-SENSEX, Dow-Jones) in their portfolios. There are evidences to suggest that the performance of index funds, replicating these indices, has been superior to other actively managed funds (Elton et al., 1996). These funds are generally regarded relatively stable in comparison to other actively managed instruments such as the mutual funds (Jensen, 1968).

Index funds are composed of smaller number of stocks than the number of stocks in the corresponding index itself. Ideally the index funds are supposed to include each and every stock included in the corresponding index, however this may lead to an inefficient portfolio management due to associated transactions costs and other issues related with the management of the huge number of stocks included in the index. Thus index fund strategy helps by replicating the movement of indices by selecting a fewer number of stocks than those included in the corresponding index itself. Owing to its combinatorial nature, the Index Fund Problem (IFP) does not have a known *polynomial-time exact algorithm*. However, there are various approximation algorithms in the literature which are discussed in detail in section 2.

In this article, we propose an Ant Colony Optimization (ACO) based algorithm to solve the IFP. ACO is a nature inspired meta-heuristic that mimics the behavior of ants. ACO is extensively used in the literature to solve various combinatorial optimization problems. Through the computational experiments on the empirical stock price data as well as on the some random test instances, we demonstrate that the proposed algorithm outperforms any of the existing approximate algorithms to solve the IFP.

Rest of the paper is structured as follows. In section 2 we formulate the IFP and discuss the work related to the same in the literature. In section 3 we briefly discuss the ACO metaheuristic and the related work. We then describe our ACO based algorithm to solve IFP in section 4. The computational experiments are reported in section 5. We conclude our study in section 6 with some remarks on further direction of research.

PROBLEM DESCRIPTION AND LITERATURE

In this section, first we will formulate the IFP and discuss the existing work that address this problem in the literature.

IFP can be formulated as following integer program (Cornuejols & Tutuncu, 2007). Let N be the set of all n stocks included in the market index. $\rho_{ij} \in [-1,1]$ be the similarity measure between stock $i \in N$ and stock $j \in N$ in the index. Larger the value of ρ_{ij} implies more similairty between the stocks $i \in N$ and $j \in N$. Correrlation betweeen the return of the stocks may be considered as one of such similarity measures. Let x_{ij} be a binary variable indicating whether the stock i is represented by the stock j in the index fund or not. Similarly, binary variable y_j is 1 if the stock j is selected in the index fund and 0 otherwise. The IP model for IFP is then as following:

max $$Z = \sum_{i \in N} \sum_{j \in N} \rho_{ij} x_{ij} \tag{1}$$

s.t. $$\sum_{j \in N} y_j = q \tag{2}$$

$$\sum_{i \in N} x_{ij} = 1 \ \forall i \in N \tag{3}$$

$$x_{ij} \leq y_j \ \forall i \in N; \forall j \in N \tag{4}$$

$$x \in \{0,1\}^{NXN} \tag{5}$$

$$y \in \{0,1\}^N \tag{6}$$

The objective function (1) seeks to maximize the similarity between the stocks chosen in the index fund and the stocks in the market index itself. Constraint (2) ensures that there are only q stocks selected in the index fund. Constraint (3) guarantees that each stock in the market index is represented by exactly one stock selected in the index fund. The coupling constraint (4) ensures that if and only if a stock is selected in the index fund it can represent any other stock in the market index.

Oha et al. (2005) proposed an approximation algorithm based upon Genetic Algorithm for the IFP. In their work, they considered coefficient of determination as the measure of fitness between the total return and the increasing rates. Oritoa et al. (2003) applied a simple method for constructing n company portfolio using coefficeint of determination and conducted numerical experiment using the share price data from the 1st and 2nd sections of Tokyo Stock Exchange. Beasley et al. (2003) proposed an evolutionary heuristic for the index tracking problem. Their model incorporated two different objectives, namely minimizing the tracking error (some function of difference in the return of the index and the return of the fund) and maximizing the excessive return (return over an above the return of the index). Canakgoz & Beasley (2009) formulated a mixed integer linear program for the index fund problem. The formulation includes the transaction cost as a constraint in order to limit the number of stocks in the index fund and another constraint that restricts the transaction cost for the index fund. To the best of our knowledge there does not exist any work related to using ACO in the framework of IFP. In the next section we will describe the characteristics of ACO along with the details of literature review for the same.

ANT COLONY OPTIMIZATION (ACO)

ACO is a part of the larger field of swarm intelligence in which scientists study the behavior pattern of bees, termites, ants and other social insects in order to simulate processes (Bell & McMullen, 2004). ACO was inspired by the foraging behavior of real ants (Dorigo et al. 1999). Ants randomly explore their surrounding using the pheromone trail (a chemical substance that can be sensed by other ants). Ants can follow various paths but in the long run, owing to reinforcement of the pheromone trail on the shortest paths, only the shortest paths remain in the use.

ACO replicates the behaviour of ants, adding some features to make it more attractive for the computer implimentation (Rizzoli et al. 2004) . Ants construct a solution visiting a series of nodes on the graph. They select the move from one node on the graph to another, along an edge, according to two attributes: trails deposit on the edge and the attractiveness of the move.

The attractiveness η_{ij} of the move, from node i to node j, is computed according to an heuristic that express the *a priori* desirability of the move. In a shortest path problem, the desirability can be the inverse of the distance.

The pheromone trail level τ_{ij} depends on the pheromone level, and it represents a dynamic indication *a posteriori* of its goodness. At each iteration t of the computation, the pheromone level on some edge (i, j) is updated according to the following equation:

$$\tau_{ij}^{t} \leftarrow (1-\rho)\tau_{ij}^{t-1} + \left(\sum_{k\in K} \Delta\tau_{ij}^{k}\right) \tag{7}$$

Where ρ is the factor determining the evaporation, which prevents the algorithm from being trapped into some local optima. Trail evaporation reduces pheromone trail on *bad moves* iteration by iteration. K is the number of ants constructing their solution in parallel.

There are several ACO metaheuristic implementations that differ in the way the artificial pheromone is deposited and evaporated. According to Dorigo & Gambardella (1997), when an artificial ant is at node i, the next node j is selected probabilistically according to *random-proportional* rule:

$$p_{ij} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{h\in\Omega_i}\tau_{ih}^{\alpha}\eta_{ih}^{\beta}} & \text{if } j \in \Omega_i \\ 0 & \text{Otherwise} \end{cases} \tag{8}$$

Where, p_{ij} is the probability of moving to node j from i, and Ω_i is the set of neighboring nodes for node i. Parameters α and β are importance weights associated with pheromone trails and the heuristic attractiveness of the move.

ACO was first applied to the Traveling Salesman Problem and the Quadratic Assignment Problem (Dorigo, 1992). ACO has been proved to be very efficient in solving some of the graph routing problems. ACO has been applied to solve various classes of combinatorial optimization problem such as JIT sequencing problems (McMullen, 2001), Hub Location problem (Randall, 2008), Vehicle routing problem (Rizzoli et al. 2004), (Bell & McMullen, 2004).

ACO HEURISTIC FOR IFP

In this section the heuristic based on Ant Colony Optimization for Index Fund is discussed in detail. Before presenting the algorithm first we will discuss the solution representation, swap operation and define the solution neighborhood as the following:

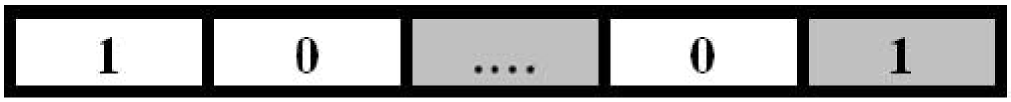

Fig 1. Solution representation for a IFP solution

Solution representation

Solutions are represented in the form of a binary vector, where **1** indicates that the corresponding stock has been selected in the fund and **0** otherwise. A solution vector $\{y_i\}$ can be written as:

$$y_i = \begin{cases} 1 & \text{if stock } i \text{ is included in fund} \\ 0 & \text{Otherwise} \end{cases}$$

Fig.1 gives an example of one such solution representation. Objective function value for a given solution vector may be found using the following function:

$$Z = \sum_{i \in N} \max_{j \in \Theta} \left(\left[\rho_{ij} \right]^{+} \right) \tag{9}$$

Where $\Theta = \{i : y_i = 1\}$, i.e. Θ is the position vector for all the stocks included in the index fund.

Swap Operation

The artificial ants travel from one solution to another using the swap operation. The swap operator (SO) is defined as:

$$A \odot SO(i,j) \equiv B$$

Such that:

$A = \{y_1, y_2, \cdots, y_{i-1}, 1, \cdots, y_{j-1}, 0, \cdots, y_n\}$

and

$$B = \{y_1, y_2, \cdots, y_{i-1}, 0, \cdots, y_{j-1}, 1, \cdots, y_n\}$$

$SO(i,j)$ replaces 1 at i^{th} place in A with a 0 and respectively replaces 0 at the j^{th} place with a 1 in order to provide a new solution vector B. Fig.2 illustrates with the help of an example a swap operation $SO(3,5)$ performed on a binary vector A.

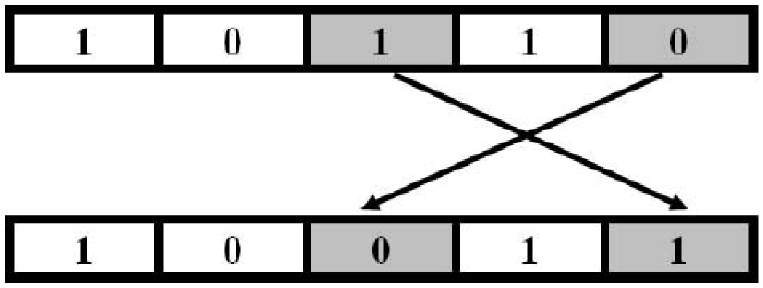

Fig 2: Example of the swap operation

Neighborhood

For any solution the neighborhood is defined as all those solutions, which can be obtained by performing a single swap operation over the current solution vector. If a solution vector Y is defined as:

$$y_i = \begin{cases} 1 & \text{if } i \in \Theta \\ 0 & \text{if } i \in \Psi \end{cases}$$

Then the neighborhood of Y is defined as:

$$\Omega_Y = \bigcup_{i\in\Theta;\, j\in\Psi} \{Y \odot SO(i,j)\} \tag{10}$$

For example, the neighborhood of the solution vector A depicted in Fig.1 can be defined as:

$$\Omega_A = \begin{matrix} \{(0,1,1,1,0),(0,0,1,1,1),(1,1,0,1,0), \\ (1,0,0,1,1),(1,1,1,0,0),(1,0,1,0,1)\} \end{matrix}$$

Heuristic Information

Heuristic information corresponding to each swap operation $SO(i,j), \eta_{ij}$ can be computed as following:

$$\eta_{ij} = \sum_{k\in N} \left([\rho_{jk}]^+ - [\rho_{ik}]^+\right) \tag{11}$$

In other words the heuristic information signifies the difference in objective function corresponding to two index funds, one with only stock j is included in it while the other with only stock i included in it.

Pheromone Trail

The pheromone trail associated with each swap operation $SO(i, j)$ refers to dynamically acquired desirability of replacing stock j in the index fund with the stock i. As per the ACO framework discussed in section 3, the pheromone trail is updated as per the equation (7), at the end of iteration. The incremental pheromone deposited on the move from solution node i to j is defined as:

$$\Delta\tau_{ij} = [Z_{new} - Z_{current}]^{+} \tag{12}$$

Where $Z_{current}$ is the objective function value corresponding to the current solution, while Z_{new} is the objective function value after performing the swap operation $SO(i, j)$ over the current solution.

Algorithm

The ACO based algorithm is mentioned in Fig.3. Few key highlights of the proposed algorithm is as following:

Initialize parameters: The algorithm is initialized with the pre-setting of the following parameter values:

- Number of Ants (K)
- Evaporation factor (ρ)
- Weight of pheromone (α)
- Weight of heuristic information (β)
- Stopping criteria for the local iteration
- Stopping criteria for the global iteration

Pheromone trail initialization: Pheromone trails, for each pair of stocks, is initialized with a value of 1 to ensure that first few iterations are driven by the heuristic attractiveness of the swap information.

Ants' solution initialization: Each ants' initial solution is a random binary vector with $|\Theta| = q$ i.e. the number of 1's in the solution is pre-set as the required number of stocks in the fund.

Local iteration: In the ACO algorithm, each artificial ant finds a new solution from the current solution using stochastic swap operation. Within the local iteration, ants keep on performing the

```
Algorithm: ACO-IFP
1.  Initialize pheromone $\tau_{ij} \leftarrow 1\ \forall i \in N; \forall j \in N$
2.  Initialize random ant solution $P^k\ \forall k \in K$
3.  (Iteration: $t$) While {$global\_stopping\_criteria$ not satisfied} do
4.      While {$local\_stopping\_criteria$ not satisfied} do
5.          For all $k \in K$ do
6.              Construct new solution by performing stochastic swap operation
7.              $\Delta Z \leftarrow$Improvement in solution $(Z_{new} - Z_{current})$
8.              Update $Z_{current} \leftarrow Z_{new}$
9.              If $(Z_{current} > Z_{local\_best})$ update $Y_t^{k*}$
10.         End for
11.     End while
12.     Update Pheromone trail $\tau_{ij}\ \forall i \in N; \forall j \in N$
13.     Update Incumbent $Y^*$
14. End while
```

Fig 3: ACO based algorithm for the index fund problem

swap operation until a local stopping criteria is met. The best solution found by the ant k in any of the iteration is updated as the local best solution (Y_t^{k*}) for an ant k in the current global iteration t. In our algorithm, we have defined the local stopping criteria with the help of following:

- CPU time limit
- Stagnation in the objective function value corresponding to the best local solution.

Global iteration: In each of the global iteration K artificial ants start from a fresh random initial solution and perform the stochastic swap operations in order to find the local best solution Y_t^{k*} the global best solution Y_t^* in a particular iteration is the best solution found by all the ants through their corresponding local iteration. Y_t^* is used to update the best incumbent solution Y^*, explored till now. The stopping criteria for the global iterations are similar to the one for the local iterations.

The above mentioned ACO based algorithm can be used to solve the IFP. Results of the proposed algorithm with various test data and the corresponding comparative analysis with other approaches proposed in the literature is discussed in the following section.

COMPUTATIONAL EXPERIMENTS

In order to analyze the efficiency of the proposed algorithm, the algorithm was tested against Simulated Annealing Heuristic (Kirkpatrick et al., 1983; Černý, 1985; Granville et al., 1994) and the Genetic Algorithm (GA) based heuristic proposed by Oha et al. (2005). The computations were performed on an Intel Core I5 2.4 GHz processor with 2GB RAM running Microsoft Windows XP. MATLAB 2007 was used to code the algorithm discussed in the paper.

Simulated Annealing

Simulated annealing (SA) is inpired by the event of annealing in metallurgy, a technique invovling heating and controlled cooling of material in order to reduce the defect in the putput material. By anology, in SA the current state of the material is replaced by the current random solution. The move from one solution to another is accepted with a probability function that depends on the difference in objective function values of the current and the new solution as well as on some parameter known as the annealing temperature T, that is gradually deacreased during the process. The algorithm accepts any random move when T is higher and as the value of T decreases the emphasis is on selecting better solution only.

The probabity, with which a move from current solution $Y_{current}$ to a new solution Y_{new} is accepeted, may be denoted as:

$$\psi = \min\left(1, \quad \exp\left(\frac{Z_{new} - Z_{current}}{\theta T}\right)\right) \tag{12}$$

Where, Z_{new} and $Z_{current}$ are the objective function values associated with Y_{new} and $Y_{current}$ respectively. T is the initial temperature and θ is the cooling factor; the parameters associated with SA algorithm. For implementation of SA , we have set the values for T and θ as 1000 and 0.1 respectively.

Parameter Setting for ACO

We performed several preliminary experiments in order to find the best fit of the parameters for ACO implementation on IFP. The list of the parameters used in ACO and their corresponding values are mentioned in Table 1.

Test data

In order to compare the efficiency of the propsed ACO based algorithm over GA and SA meta-heuristic we performed our tests on the stock price data of S&P100 index for a 9 year duration

Table 1. Parameters for ACO based algorithm

Number of ants (K)	20
Weight of pheromone trail (α)	1.50
Weight of heuristic information (β)	0.50
Evaporation factor (ρ)	0.15
Local stopping criteria (iterations without improvement)	50
Global stopping criteria (iterations without improvement)	50

(from 14th Feb 2000 to 13th Feb 2009). We used correlation between all stock prices' logarthim returns as the similairty measure for our experiments.

In order to further establish the strength of our algorithm we created several random test instances with the corresponding correlation coefficient defined with following market behaviour (and the corresponding rule):

Scenarios:

A: All the stocks have independent movements ($\rho_{ij} \in [-1.0,1.0]$)

B: All stocks movement have positive correlation $\rho_{ij} \in [0,1.0]$

C: Most of the stocks movement have strong positive correlation while few have weak negative correlation $\rho_{ij} \in [-0.2,0.6]$

D: Most of the stocks movement have strong negative correlation while few have weak positive correlation $\rho_{ij} \in [-0.6,0.2]$

Depending upon the scenario correlation coefficient between a pair of stocks i and j is a random number within the specified range.

Test Results

Table 2 and 3 reports the comparative analysis between ACO, GA and SA heuristics. Table 2 reports the comparative analysis for a CPU time limit on each of the implementation set as 50 seconds, while the Table 3 reports the comparative anlysis for a CPU time limit of 100 seconds. For both, Table 2 and Table 3, 50 simulation runs were used and the reported results are an average of all the simulation runs performed over a particular test instance.

Structure of both the table is similar and is as following. Column 1 specifies the test instances, column 2 corresponds to the number of stocks to be selected in the index funds (q). Column 3

Table 2. Test results (CPU Time limit: 50 seconds & Simulation runs: 50)

Test Instance	q	%Improvement(GA)	%Improvement(SA)
S&P100	50	2.34	10.51
S&P100	30	5.51	12.36
Scenario-A	50	1.56	21.35
Scenario-A	30	3.02	35.13
Scenario-B	50	4.18	12.14
Scenario-B	30	0.01	20.00
Scenario-C	50	9.56	15.03
Scenario-C	30	12.78	-0.14
Scenario-D	50	0.54	3.46
Scenario-D	30	13.21	14.04

demonstrates the strngth of the ACO algorithm over the GA algoirthm, where the %improvement in the solution value is reported that is computated as:

%Improvement(GA)= $\frac{Z_{ACO} - Z_{GA}}{Z_{GA}}$ X 100

Where, Z_{ACO} and Z_{GA} are the objective function values corresponding to the solution obtained with the ACO based algorithm and the GA based algortihm, respectively. Similarly in column 4 the %improvement in solution for ACO vs SA is reported.

Comparative analysis between Table 2 and Table 3 shows that with higher CPU time limit the improvement in the solution obtained with ACO based algorithm is more significant, which can be attributed to faster improvement in solution quality with ACO as opposed to the other two heusristics in solving the IFP. The results reported in both the tables support our argument for the superiority of the ACO algorithm over the other two algorithm in solving IFP.

CONCLUSION

In this paper we proposed an Ant Colony Optimization based algorithm to solve the Index Fund Problem in portfolio management. The algorithm was tested against one of the heusristic proposed in the literature (Genetic Algorithm (GA)) to solve the IFP and another heurstic based upon Simulated Annealing (SA). We demonstrated with the help of computational experiments,

Table 3. Test results (CPU Time limit: 100 seconds & Simulation runs: 50)

Test Instance	*q*	%Improvement(GA)	%Improvement(SA)
S&P100	50	7.56	8.93
S&P100	30	12.32	7.89
Scenario-A	50	2.11	15.51
Scenario-A	30	5.15	13.43
Scenario-B	50	17.91	10.53
Scenario-B	30	13.53	8.75
Scenario-C	50	17.89	12.13
Scenario-C	30	15.56	0.09
Scenario-D	50	12.56	28.23
Scenario-D	30	18.68	20.04

performed on several random test instances and instances based upon historical stock price data, that the ACO based algorithm out-performs the GA and SA based algorithms.

The proposed algorithm can be easily adapted to include other factors in the IFP model such as transcation cost, residual return etc. In order to incorporate these additional features in the model, one need to modify the objective function definition accordinly. One possible direction for the future study may be to relax the constraint of fixed number of stocks in the index fund and model an optimization model with multiple objectives such as: maximizing the similairity between the index fund and the market index; minimizing the transaction cost; maximing the expected residual return. Since our study establishes the strength of the ACO based algorithm over other established algorithms to solve the IFP, the proposed algorithm can be used as a benchmark for development of any other algorithms to solve IFP.

References:

Beasley, J., Meade, N., & Chang, T. (2003). An evolutionary heuristic for the index tracking problem. *European Journal of Operational Research* , 621-643.

Bell, J., & McMullen, P. (2004). Ant Colony Optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics* , 41-48.

Canakgoz, N., & Beasley, J. (2009). Mixed-integer programming approaches for index next term tracking and enhanced indexation. *European Journal of Operational Research* , 384-389.

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization: Theory and Applications* , 41-51.

Cornuejols, G., & Tutuncu, R. *Optimization Methods in Finance.* 2007: Cambridge University Press.

Dorigo, M. (1992). *Optimization learning and natural algorithms.* Dipartimento di Elettronica Italy.

Dorigo, M., & Gambardella, L. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* , 53-66.

Dorigo, M., Gambardella, L., & Di Caro, G. (1999). Ant algorithms for discrete optimization. *Artificial Life* , 137-172.

Elton, E., Gruber, G., & Blake, C. (1996). Survivorship bias and mutual fund performance. *Review of Financial Studies* , 1097-1120.

Granville, V., Krivanek, M., & Rasson, J.-P. (1994). Simulated annealing: A proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 652-656.

Jensen, M. (1968). The performance of mutual funds in the period {1945}-{1964}. *Journal of Finance* , 389-416.

Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by Simulated Annealing. *Science* , 671-680.

McMullen, P. (2001). An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives. *Artificial Intelligence in Engineerin* , 309-317.

Oha, K., Kimb, T., & Mina, S. (2005). Using genetic algorithm to support portfolio optimization for index fund managemen. *Expert Systems with Applications* , 371-379.

Oritoa, Y., Yamamotob, H., & Yamazaki, G. (2003). Index fund selections with genetic algorithms and heuristic classifications. *Computers and Industrial Engineering* , 97-109.

Randall, M. (2008). Solution approaches for the capacitated single allocation hub location problem using ant colony optimization. *Computational Optimization and Applications* , 239-261.

Rizzoli, A., Oliverio, F., Montemanni, R., & Gambardella, L. (2004). *Ant Colony Optimisation for vehicle routing problems: from theory to applications.*