A Hybrid Genetic Algorithm for a Two-Stage Stochastic Portfolio Optimization With Uncertain Asset Prices

Tianxiang Cui*, Ruibin Bai*, Andrew J. Parkes†, Fang He†, Rong Qu†, Jingpeng Li‡
*Division of Computer Science, The University of Nottingham Ningbo, China
†School of Computer Science, The University of Nottingham,UK
‡Computing Science and Mathematics, University of Stirling, UK
Tianxiang.cui@nottingham.edu.cn

***Abstract*—Portfolio optimization is one of the most important problems in the finance field. The traditional mean-variance model has its drawbacks since it fails to take the market uncertainty into account. In this work, we investigate a two-stage stochastic portfolio optimization model with a comprehensive set of real world trading constraints in order to capture the market uncertainties in terms of future asset prices. A hybrid approach, which integrates genetic algorithm (GA) and a linear programming (LP) solver is proposed in order to solve the model, where GA is used to search for the assets selection heuristically and the LP solver solves the corresponding sub-problems of weight allocation optimally. Scenarios are generated to capture uncertain prices of assets for five benchmark market instances. The computational results indicate that the proposed hybrid algorithm can obtain very promising solutions. Possible future research directions are also discussed.**

***Index Terms*—Hybrid Algorithm; Portfolio Optimization; Stochastic Programming; Genetic Algorithm.**

I. Introduction

With the advances in computer processors in recent decades, quantitative trading is increasingly taking the place of professional investors in financial centers across the world. The investment decisions are made not only by the financial experts, but also based on mathematical computations and number crunching by mathematicians or computer scientists. Portfolio optimization is one of the important areas in quantitative trading. The idea is to allocate a certain amount of capital over different assets to form a portfolio. The goal is to minimize the portfolio risk for a specific level of return set by investors. This is often referred as the portfolio optimization problem. The first modern portfolio optimization model was proposed by Markowitz in the 1950s [1], [2], where, the risk of the portfolio is measured as the variance of the asset return and therefore the problem can be viewed as a mean-variance optimization problem. The original problem is a quadratic programming problem therefore can be solved in an exact manner with a reasonable computational time. By imposing more real world constraints, for example cardinality and bounding, it changes the model into an NP-hard problem. In our previous work [3], we proposed a combinatorial algorithm for the cardinality constrained portfolio optimization problem using the classic mean-variance model.

Despite the real world constraints being added in the classic mean-variance model, the investors still need to consider one more important market factor in order to make better investment decisions - the uncertainty. In the current work of mean-variance portfolio optimization problem [3]–[6], the mean expected return and the covariance between assets are assumed to be static, which is often not true due to the economic turmoil and the market uncertainties in practice. It has been pointed out in [7], [8] that the investment decisions should be made based on consideration of the market uncertainties. Usually, the random uncertainty factors are taken into account (i.e. the asset price and the currency exchange rate, etc.). There are also some other non-probabilistic uncertainty factors (i.e. the vagueness and the ambiguity, etc.) which are mainly modeled using fuzzy techniques [9], [10]. In this work, we will mainly focus on the random uncertainty of the market, more specifically, we consider the future asset prices to be uncertain.

Stochastic programming [11], [12] is a useful technique for modeling optimization problems with uncertain factors. It can model uncertainty and impose real world constraints in a flexible way [13]. As it has been showed in [14], stochastic programming has been applied to many different areas successfully (finance, sports, scheduling [15], telecommunications, energy, production control and capacity planning, etc.). For this work, we propose to use stochastic programming to model uncertain future asset price.

Another drawback of the mean-variance model is how it characterizes the risk. In the classical Markowitz portfolio optimization model, the risk is measured as the variance of the asset returns. Because such characterization of the risk is a measure of the dispersion of the values of the variable around its expected value, therefore it penalizes the portfolio profits and the portfolio losses at the same time. Practically, people may only want to minimize the possibility of the portfolio losses, therefore Rockafella and Uryasev [16] proposed a different risk measure namely Conditional Value at Risk (CVaR) which calculates the expected loss for the worst case scenarios. As it has been showed in [17], CVaR is a sub-additive and convex risk measure, therefore it can be optimized using stochastic programming.

There exists some research using stochastic programming models in the portfolio optimization literature. For example Topaloglou *et al.* [18] proposed a multi-stage stochastic programming model for international portfolio management in a dynamic setting. The uncertainties are modeled in terms of the asset prices and exchange rates. Stoyan and Kwon [19] considered a stochastic-goal mixed-integer programming model for the integrated stock and bond portfolio problem. The uncertainties are modeled in terms of the asset prices and the real world trading constraints are imposed. The model was solved by a decomposition based algorithm. He and Qu [20] proposed a two stage portfolio selection problem with a comprehensive set of real world trading constraints. The uncertainties are modeled in terms of the asset prices. A hybrid algorithm integrated local search and a default Branch-and-Bound method is proposed to solve the problem.

In this work, we adapt the stochastic portfolio optimization model in the literature [18], [20] and propose a hybrid algorithm for the two-stage stochastic portfolio optimization problem with a comprehensive set of real world trading constraints. We integrate genetic algorithm (GA) together with a commercial LP solver so that GA is used to search for the assets selection heuristically while the LP solver can solve the corresponding sub-problems optimally. The main advantage of such an approach is that we can guarantee the optimal allocation for a given assets combination provided by GA and the computational cost for solving a sub-problem is inexpensive since all the stochastic zero-value variables are not considered.

The outline of the rest part is as follows: Section II reviews the general concepts of stochastic programming, scenario tree and percentile risk function. Section III gives the statement of the problem as well as the corresponding notations used. In section IV we provide a detailed description of our hybrid algorithm. The datasets, scenario generation method and parameter settings are stated in section V. Computational results are presented in section VI and final conclusions are given in section VII.

II. Preliminaries

A. Two-Stage Stochastic Programming Problem With Recourse

Stochastic programming is a common approach to deal with uncertainty. The general concepts of stochastic programming have been discussed in [11], [12]. For this work, we consider a widely applied class of stochastic programming problem, namely the recourse problem. It seeks a policy that can take the actions after some realisation of the uncertain variables as well as make the recourse decisions based on temporarily available information.

The simplest case of the recourse problem have two stages:

- *first stage*: A current decision needs to be made.
- *second stage*: The values of the uncertain variables are revealed and further decisions are made in order to avoid constraints violations. Usually a decision in the second stage will depend on a particular realisation of the uncertain variables.

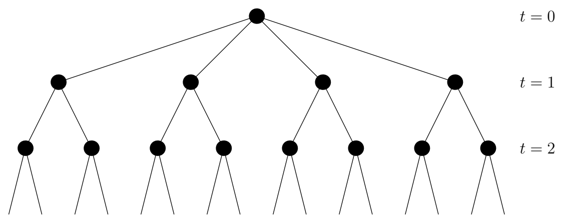

Fig. 1. An example of a scenario tree. Each path from the root node to a leaf node represents one scenario.

B. Scenario Tree

There are two common methods which can be used to deal with multistage stochastic programming problems, namely decision rule approximation and scenario tree approximation. For this work, we will use the scenario tree approximation tree method.

A scenario is defined as the possible realisation of the uncertain data ξ in each stage $t \in T$. An example of a scenario tree is showed in Figure 1. The nodes in the scenario tree represent a possible realisation of the uncertain data ξ^T. Each node is denoted by $n = (s, t)$ where s is a scenario and t is the level of the node in the tree and the decisions will be made at each node. The parent of node n is represented by $a_{t-1}(n)$. The branching probability of node n is denoted by p_n which is a conditional probability on its parent node $a_{t-1}(n)$. The path to node n is a partial scenario with the probability $Pr_n = \prod p_n$ along the path and the sum of Pr_n is up to 1 across each level of the scenario tree.

In order to apply the scenario tree approximation method for the stochastic programming problem with recourse, the uncertain data ξ needs to be discretized and all possible realisations of ξ can be represented by a discrete set of scenarios. Thus, scenario generation methods are required. There are several scenario generation methods in the literature. For this work, we applied a shape based method [21].

C. Percentile Risk Function

In the real-world situation, portfolio managers may only need to reduce the possibility of high loss. Value at Risk (VaR) [22], [23] gives the maximum possible loss α with a specified confidence level β. That is, by the end of the investing period, the probability of the loss exceeding the threshold α is $1 - \beta$ (see Figure 2).

Formally, let $f(x, \xi)$ be the loss function where $x \in R$ is the decision vector and $\xi \in R$ is the uncertain (random) vector. The density of the probability distribution of ξ is denoted by $p(\xi)$. The probability of the loss function $f(x, \xi)$ not exceeding a threshold α is given by:

$$\Psi(x, \alpha) = \int_{f(x,\xi) \leq \alpha} p(\xi) d\xi$$

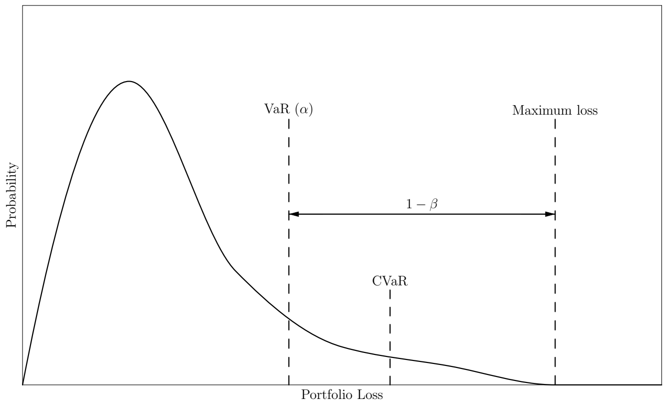

Fig. 2. Value at Risk (VaR) and Conditional Value at Risk (CVaR).

The β-VaR for the loss random variable associated with x and the specified probability β in $(0,1)$ is denoted by $\alpha_\beta(x)$ and formally we have the following:

$$\alpha_\beta(x) = \min\{\alpha \in R : \Psi(x,\alpha) \geq \beta\}$$

As it has been pointed out in [17], VaR does not satisfy the sub-additivity and the convexity and generally it is not a coherent risk measure. In order to eliminate the drawbacks of VaR, a more consistent risk measure, Conditional Value at Risk (CVaR) is proposed. It satisfies sub-additivity and the convexity [17] and is also proven to be a coherent risk measure [24].

CVaR calculates the expected value of the loss which exceeds the VaR value (see Figure 2). Formally, CVaR is defined as the follows [16]:

$$\phi_\beta(x) = (1-\beta)^{-1} \int_{f(x,\xi)\geq\alpha_\beta(x)} f(x,\xi)p(\xi)d\xi$$

By having the analytical representation to replace VaR $\alpha_\beta(x)$, we can have:

$$F_\beta(x,\alpha) = \alpha + (1-\beta)^{-1} \int_{f(x,\xi)\leq\alpha} (f(x,\xi)-\alpha)p(\xi)d\xi$$

It has been proved in [16] that $F_\beta(x,\alpha)$ is a convex function with respect to α and the minimum point of $F_\beta(x,\alpha)$ is VaR with respect to α. The CVaR value can be obtained by minimizing $F_\beta(x,\alpha)$ with respect to α.

III. Problem Statement

A. Notations

The notations we used in this work are given in Table I.

B. Two-stage stochastic portfolio optimization model with recourse

Now we propose our two-stage stochastic portfolio optimization model with recourse. Inspired by [18], the model was originally proposed in [20] and we adapt the model by changing and adding some conditions. The proposed model is divided into two stages.

$$\min \left(\alpha + (1-\beta)^{-1} \sum_{j\in N_r} p_j z_j \right) \tag{1}$$

$s.t.$

First Stage - Portfolio Selection:

$$w_i = w_i^0 + b_i - s_i, \quad \forall i \in A \tag{2}$$

$$h + \sum_{i\in A}(s_i P_i^0) - \sum_{i\in A}(\eta_s g_i + s_i \rho_s P_i^0) = \sum_{i\in A}(b_i P_i^0) + \sum_{i\in A}(\eta_b f_i + b_i \rho_b P_i^0) \tag{3}$$

$$\sum_{i\in A} c_i = K \tag{4}$$

$$w_{\min} c_i \leq w_i \quad \forall i \in A \tag{5}$$

$$t_{\min} f_i \leq b_i \quad \forall i \in A \tag{6}$$

$$t_{\min} g_i \leq s_i \quad \forall i \in A \tag{7}$$

$$f_i + g_i \leq 1 \quad \forall i \in A \tag{8}$$

$$f_i M \geq b_i \quad \forall i \in A \tag{9}$$

$$g_i M \geq s_i \quad \forall i \in A \tag{10}$$

$$b_i M \geq f_i \quad \forall i \in A \tag{11}$$

$$s_i M \geq g_i \quad \forall i \in A \tag{12}$$

$$w_i,\ b_i,\ s_i \in \mathbb{R} \tag{13}$$

$$c_i,\ f_i,\ g_i \in \mathbb{B} \tag{14}$$

Second Stage - Recourse:

$$w_i^j = w_i + b_i^j - s_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{15}$$

$$\sum_{i\in A}(s_i^j P_i^j) - \sum_{i\in A}(\eta_s g_i^j + s_i^j \rho_s P_i^j) = \sum_{i\in A}(b_i^j P_i^j) + \sum_{i\in A}(\eta_b f_i^j + b_i \rho_b P_i^j) \quad \forall j \in N_r \tag{16}$$

$$\sum_{i\in A} c_i^j = K \quad \forall j \in N_r \tag{17}$$

$$w_{\min} c_i^j \leq w_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{18}$$

$$t_{\min} f_i^j \leq b_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{19}$$

$$t_{\min} g_i^j \leq s_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{20}$$

$$f_i^j + g_i^j \leq 1 \quad \forall i \in A,\ \forall j \in N_r \tag{21}$$

$$f_i^j M \geq b_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{22}$$

$$g_i^j M \geq s_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{23}$$

$$b_i^j M \geq f_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{24}$$

$$s_i^j M \geq g_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{25}$$

$$V^j = \sum_{\substack{i\in A \\ e\in N_e^j}} p_{(j,e)} P_i^{(j,e)} w_i^j \quad \forall j \in N_r \tag{26}$$

$$R^j = V^j - V^0 \quad \forall j \in N_r \tag{27}$$

$$z_j \geq -R^j - \alpha \quad \forall j \in N_r \tag{28}$$

$$z_j \geq 0 \quad \forall j \in N_r \tag{29}$$

$$\sum_{j\in N_r} p_j R^j \geq \mu \tag{30}$$

$$w_i^j,\ b_i^j,\ s_i^j \in \mathbb{R} \tag{31}$$

$$c_i^j,\ f_i^j,\ g_i^j \in \mathbb{B} \tag{32}$$

$$\alpha,\ z_j \in \mathbb{R} \tag{33}$$

The objective function (1) calculates the β-percentile CVaR of the portfolio loss at the end of the second stage where α is

TABLE I
NOTATIONS IN THE MODEL

Type of Data	**Notation**	**Meaning**
Set	A	The set of assets
Set	N_r	The set of recourse nodes. One node corresponds to one recourse portfolio
Set	N_e^j	The set of evaluate nodes on recourse node j where $j \in N_r$
User-specific parameter	μ	The target return
User-specific parameter	β	Quantile (percentile) for VaR and CVaR
User-specific parameter	M	The big constant
Deterministic input data	h	The initial cash to invest
Deterministic input data	w_i^0	The initial position of asset i (in number of units)
Deterministic input data	η_b	The fixed buying cost
Deterministic input data	η_s	The fixed selling cost
Deterministic input data	ρ_b	The variable buying cost
Deterministic input data	ρ_s	The variable selling cost
Deterministic input data	K	The number of asset held in the portfolio (cardinality)
Deterministic input data	w_{min}	The minimum holding position
Deterministic input data	t_{min}	The minimum trading size
Scenario dependent data	p_j	The probability of recourse node j in the second stage
Scenario dependent data	$p_{(j,e)}$	The probability of evaluate node e of recourse node j in the second stage
Scenario dependent data	P_i^0	The price of asset i in the first stage (per unit)
Scenario dependent data	P_i^j	The price of asset i on recourse node j in the second stage (per unit)
Scenario dependent data	$P_i^{(j,e)}$	The price of asset i on evaluate node e of recourse node j in the second stage (per unit)
Scenario dependent data	V^0	The initial portfolio wealth
Scenario dependent data	V^j	The final portfolio wealth on recourse node j
Auxiliary variable	z_j	Portfolio shortfall in excess of VaR at recourse node j
Auxiliary variable	α	The optimal VaR value
Decision variable	b_i	The number of units of asset i purchased in the first stage
Decision variable	s_i	The number of units of asset i sold in the first stage
Decision variable	w_i	The final position of asset i in the first stage
Decision variable	b_i^j	The number of units of asset i purchased on recourse node j in the second stage
Decision variable	s_i^j	The number of units of asset i sold on recourse node j in the second stage
Decision variable	w_i^j	The final position of asset i on recourse node j in the second stage
Decision variable	c_i	The binary holding decision variable in the first stage
Decision variable	f_i	The binary buying decision variable in the first stage
Decision variable	g_i	The binary selling decision variable in the first stage
Decision variable	c_i^j	The binary holding decision variable on recourse node j in the second stage
Decision variable	f_i^j	The binary buying decision variable on recourse node j in the second stage
Decision variable	g_i^j	The binary selling decision variable on recourse node j in the second stage

the corresponding optimal VaR value. Eq. (2) is the first stage asset balance condition and Eq. (15) is the second stage asset balance condition. Equations (3), (16) are the cash balance conditions for the first and second stage respectively. We apply a fixed transaction cost and a linear variable transaction cost to both buying and selling an asset. The idea is that the cash inflows should equal to the cash outflows in both stages (i.e. no cash left). Equations (4), (17) are the cardinality constraints for the first and second stage respectively where K is the desired number of the assets held within a portfolio. Equations (5) and (18) put the restrictions on the minimum holding size of an asset in order to prevent very small asset positions for the first and second stages. Equations (6),(7), are the minimum trading conditions for the first stage and equations 19,20, are the minimum trading conditions for the second stage. The idea is to prevent trading a very small proportion of an asset. Buying and selling the same asset at the same time is not allowed, this is given in equation (8) for the first stage and equation (21) for the second stage. The big-M formulations are used in the model in order to bound the decision variables and the binary decision variables (constraints (9), (10), (11),(12) for the first stage and constraints (22), (23), (24), (25) for the second stage). The idea is, if the decision variables for buying/selling an asset is greater than 0, then the corresponding binary decision variables should equal; if the decision variables for buying/selling an asset is 0, then the corresponding binary decision variables should be 0 and vice versa. Equations (26), (27) calculate the portfolio return on each recourse node by using a different set of evaluate scenarios in order to have a better reflection of changing price scenarios in the reality. Equations (28), (29) define the excess shortfall z_j of the recourse portfolio where $z_j = \max[0, -R^j - \alpha]$ for each recourse node. The minimum portfolio target return μ is given in equation (30). The decision variables $w_i, b_i, s_i, w_i^j, b_i^j, s_i^j$ specify the exact amount of the units for an asset to buy or sell and in a real-world situation, these decision variables should be integers. As they increase the computational difficulty significantly, we took the same method suggested in [20], [25] to relax these decision variables as continuous variables.

C. *Computational complexity*

The optimization problems using a stochastic model are much more difficult compared to using a deterministic model. It has been proved in [26] that, linear two-stage stochastic programming problems are #P-hard, which is at least as hard as the corresponding NP-hard problem and for multistage stochastic programming problems, it is generally computationally intractable even for the medium-accuracy solutions [27].

IV. The Proposed Hybrid Algorithm

It has been pointed out that in the classic constrained mean-variance portfolio optimization model, the most challenging part comes from the cardinality constraint [19], [28]. It is mainly because the cardinality constraint is discrete and therefore the solution space is discontinuous. When comes to the two-stage stochastic portfolio optimization model, the problem becomes even more challenging since it dramatically increases the search space due to the additional variables involved in the second stage.

To deal with the cardinality constraint, we took the idea of variable fixing [29]. Instead of having the binary decision variables c_i, c_i^j, we have exact K core variables. Each of the core variables has a numerical value. The main advantage of doing this is it drops off the cardinality binary decision variables thus reduces the computational complexity.

In order to reduce the search space, we generate the reduced sub-problems by removing all the non-selected assets (the details are discussed later in section IV-A). The purpose of solving the reduced sub-problems is to assign weights to the selected assets. The main advantage of generating the reduced sub-problems is it can significantly narrow the search space. Then the reduced sub-problem can be solved by the LP solver efficiently.

For this work, we used genetic algorithm (GA) to search for the asset combination and a CPLEX LP solver to solve the sub-problems.

A. The reduced sub-problem

In this work, GA is used to search for the assets combination. Each chromosome in GA represents a potential sub-problem to the original problem defined on the two-stage stochastic model. The sub-problems are generated by dropping off all the non-selected assets, in other words, each chromosome is fixed in both the first and second stage (i.e. $c_i = c_i^j = 1$ if GA picks asset i). The recourse are limited to assets rebalancing, but not swapping the assets and therefore we can call such sub-problem the reduced sub-problem. The logic behind this idea is, if some assets need to be dropped in the second stage for most of the scenarios, they will be probably not selected in the first stage. As the transaction costs and the minimum holding constraint are considered in the model, the cost of buying an entirely new asset is probably significantly higher than just adjusting the holding of an existing one.

Then the fitness value is obtained by solving the corresponding sub-problem using an LP solver in order to get the weight allocation of the selected assets. There are three main advantages of such an approach. Firstly, the binary decision variables for the cardinality constraints are dropped off, therefore it reduces the complexity; Secondly, all the zero-value variables are removed, therefore it can significantly narrows the search space. It is especially useful when there is a big amount of stochastic variables in the second stage. Thirdly, we can control the numerical properties of the solutions to the sub-problems by setting up different Markowitz threshold (which is used to control the kinds of pivots permitted), and in most cases, we can obtain the optimal allocations for a given asset combination.

B. Genetic algorithm

Genetic algorithm (GA) is a metaheuristics searching technique inspired by natural evolution process. It was first proposed by John Holland in the 1970s [30]. It simulates the survival of the fittest principal among a population of individuals over iterative generations in order to solve an optimization problem. Each individual is encoded into a string (called chromosome or genotype) which represents a possible solution to the given problem.

Starting with an initial generation of the population, each individual is evaluated by a fitness function. Based on the fitness values, some individuals are probabilistically selected to remain unchanged to the next generation, some individuals are probabilistically selected to participate in the genetic operations to produce offsprings for the next generation. Normally there are two genetic operations, crossover and mutation. After a new generation of individuals is created, the fitness of each individual is evaluated again. This whole process is repeated until an acceptable individual is found or other termination criterion is satisfied (usually up to some certain number of generations). The best individual will be returned as the solution.

C. Problem representation

In this paper, genetic algorithm is utilized to evolve best values for discrete variables in the stochastic model. The search space is different for different benchmark datasets (characterized by Q, see Section V-A). The objective is to find the best K items from Q possible assets for a given target return μ specified by the investor. The details of problem representation are as follows:

- The fitness function F maps from a list of K integers and a target return μ to a real number: $F(\mathcal{Z}^K, \mu) \rightarrow \mathcal{R}$.
- Each chromosome represents a potential sub-problem to the original problem defined on the two-stage stochastic model. It is encoded as a fixed length vector of $\mathbf{k} = (k_0, k_1, ..., k_K)$ which represents the selected K assets of the portfolio and $k_i \in \{1, 2, \ldots, Q\}$. The selection of the ith asset is dependent on other assets, i.e., the search problem is a dimensional dependent problem and the dimension of GA's search space is increased with the increase of dataset's dimension. The sub-problem can be solved by a standard LP solver.
- Elitist selection is used in our GA approach, i.e., we keep the best chromosome in each generation.

- The global best solution $\mathbf{x}_{\text{gb}}$ is recorded such that $F(\mathbf{x}_{\text{gb}}, \mu) \leq F(\mathbf{x}_i, \mu)$ for all $\mathbf{x}_i$ at the given return level μ.

The procedure of genetic algorithm used in this work is given as Algorithm 1 and the parameters are given in section V-C1.

Algorithm 1: Genetic algorithm for searching the set of assets

```
1  Initialize generation 0 by randomly creating a population
   of individuals;
2  for each individual in the initial generation do
3      Solve the corresponding sub-problems using CPLEX
       LP solver;
4      Evaluate the fitness;
5  while stopping criteria is not met;
6  do
7      Elitism: Select the best individual from the current
       generation and insert it into the next new generation;
8      Copy: Select $(\vartheta \times Po - 1)$ individuals using a
       roulette wheel selection from the current generation,
       copy them and insert into the next new generation;
9      Crossover: Select the individuals from the current
       generation using a roulette wheel selection, pair them
       up to produce $(\varrho \times Po)$ offsprings and insert the
       offsprings into the next new generation;
10     Mutate: Select $(\varsigma \times Po)$ individuals from the current
       generation randomly, alter one asset number and
       insert it into the next new generation;
11     for each individual in the new generation do
12         Evaluate individual's fitness by solving the
           corresponding sub-problem using CPLEX LP
           solver;
```

V. Data Set, Scenarios And Parameter Settings

A. The data set

In this paper, we extend the five benchmark instances which are available from the OR-library [31]. It contains 261 weekly historical price data for each asset of the following five different capital market indices:

- Hang Seng in Hong Kong, $Q = 31$.
- DAX 100 in Germany, $Q = 85$.
- FTSE 100 in UK, $Q = 89$.
- S&P 100 in US, $Q = 98$.
- Nikkei 225 in Japan, $Q = 225$.

Q is the number of assets available for each market index. The weekly historical price data are used for generating the scenarios for the two-stage stochastic portfolio optimization model.

B. Scenario generation

Since we have a two-stage model, we cannot use the 261 weekly historical price data directly because it would lead to a prohibitively huge multi-stage problem. Instead, we have the following: we take the week 1 data $q_1^1, \ldots, q_1^Q$ as the initial price for the assets. Start from week 2, we calculate the difference between the price of the assets of two consecutive weeks $\Delta_t = q_{t+1}^i - q_t^i$ where $i = 1 \ldots Q$, $t = 1 \ldots 260$. Then we can obtain 260 new price data by computing $q_1^i + \Delta_t \forall i \in Q, t = 1 \ldots 260$.

We take the copula scenario generation method presented in [21] to generate 100 scenarios for the recourse nodes and 20 scenarios for the evaluate nodes by using 260 newly-generated data described above. Therefore there will be 2000 possibilities of scenarios in total. The reason of doing that is because we want to test the effectiveness of our proposed hybrid algorithm. We do not claim this is the optimal number for generating scenarios. Our primary aim is rather to develop an efficient method that can solve two-stage stochastic portfolio optimization problem.

C. Parameter settings

1) GA parameters: We set population size $Po = 500$, the number of generations $Ge = 500$, copy rate $\vartheta = 10\%$, crossover rate $\varrho = 80\%$ and mutation rate $\varsigma = 10\%$.

2) Model parameters: For each given target expected return μ, we set the critical percentile level of CVaR $\beta = 95\%$, fixed buying cost $\eta_b = 0.5$, variable buying cost $\rho_b = 0.1\%$, fixed selling cost $\eta_s = 0.5$, variable selling cost $\rho_s = 0.1\%$, cardinality $K = 10$, minimum holding position $w_{min} = 1\%$, minimum trading size $t_{min} = 0.1\%$. The initial portfolio only involves cash and we set the initial cash $h = 100000$. We assume the probability of each scenario is equal and therefore $p_j = 1/N_r$, $p_{(j,e)} = 1/N_e^j$.

VI. Experimental Results

A. Computational results for a small number of scenarios

In order to test the effectiveness of our algorithm, we use CPLEX (version 12.4) to obtain the optimal solutions for Hang Seng instance ($Q = 31$) with a small number ($N_r = 20, N_e^j = 5$, 100 possibilities in total) of scenarios. Consider the time limitation, we choose 20 equally spaced return levels and for each return level, we use CPLEX to solve the two-stage stochastic model directly. For each return level, CPLEX can solve the whole model to the optimality within a few minutes. We also use CPLEX to solve the model without cardinality constraint using the same data and parameter settings to obtain an efficient frontier (see the red line in Figure 3). This efficient frontier can be considered as an upper bound of the frontier obtained with the cardinality constraint [4].

For each of the same return level, we run our hybrid algorithm to obtain a portfolio. The results are shown in Figure 3. We can see that all the green points lie exactly on the blue dashed line. In fact, for each of the 20 different return levels, our algorithm obtains exact the same CVaR value with CPLEX (i.e. optimal).

We also apply a percentage deviation method which is widely used in the literature [3], [4], [6] to determine the

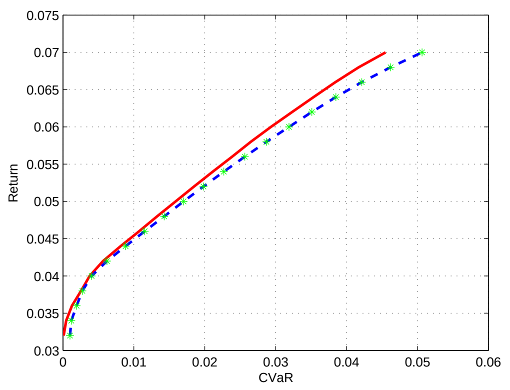

Fig. 3. Computational results for the Hang Seng instance with 100 possibilities of scenarios. The red line is the optimal upper bound efficient frontier (i.e. without cardinality constraint), the blue dashed line is the optimal efficient frontier for the whole problem computed by CPLEX and the green points are the portfolios obtained by our algorithm.

quality of the portfolios obtained. The percentage deviation error is measured by calculating the distance between the obtained portfolio and the upper bound efficient frontier, both horizontally and vertically. Formally, let $(x_{\text{uef}}, y_{\text{uef}})$ be a discrete point on the upper bound efficient frontier. The horizontal distance is calculated by taking the portfolio expected return as fixed ($y = y_{\text{uef}}$), linearly interpolating the point on the upper bound efficient frontier to get the x value $x_{\text{interpolation}}$ and take the absolute value of the difference between x_{uef} and $x_{\text{interpolation}}$. Then the percentage deviation error in the x-direction is computed as $|x_{\text{uef}} - x_{\text{interpolation}}|/x_{\text{uef}} \times 100\%$. The percentage deviation error in the y-direction can be calculated in a similar way. The final percentage deviation error is the minimum of the percentage deviation error of the both directions. The results are given in Table II. Here BPE denotes the best percentage deviation error, MedPE denotes the median percentage deviation error and MPE denotes the mean percentage deviation error.

TABLE II
PERCENTAGE DEVIATION ERROR OF HANG SENG INSTANCE WITH 100 POSSIBILITIES OF SCENARIOS

Instance				BPE(%)	MedPE(%)	MPE(%)
Index	Q	N_r	N_e^j			
Hang Seng	31	20	5	0.7421	2.4866	2.0687

B. Computational results for the five general benchmark instances

For each market instance, we generate 100 scenarios for the recourse nodes and 20 scenarios for the evaluate nodes ($N_r = 100, N_e^j = 20$, 2000 possibilities in total). Then we choose 20 equally spaced return levels. For the full problem, CPLEX fails to give any feasible solution within a time limit of 3 hours. But we can use CPLEX to compute the corresponding optimal upper bound efficient frontier (i.e. without cardinality). Figure 4 shows the comparison results of the frontier obtained by our hybrid algorithm with the optimal upper bound efficient frontier. Computational results for percentage deviation method are given in Table III.

TABLE III
PERCENTAGE DEVIATION ERROR OF 5 BENCHMARK INSTANCES WITH 2000 POSSIBILITIES OF SCENARIOS

Instance				BPE(%)	MedPE(%)	MPE(%)
Index	Q	N_r	N_e^j			
Hang Seng	31	100	20	2.0457	2.2093	2.2090
DAX 100	85	100	20	0.2189	1.5361	1.4772
FTSE 100	89	100	20	0.3341	1.1789	1.0475
S&P 100	98	100	20	0.2986	1.2584	1.3120
Nikkei 225	225	100	20	0.3550	1.2731	1.3189
Average				0.6505	1.4912	1.4729

C. The Computational time

The hybrid algorithm for the two-stage stochastic portfolio optimization model was implemented in C# with concert technology in CPLEX on top of CPLEX 12.4 solver. All the tests were run on the same Intel(R) Core(TM) i7-4600M 2.90GHz processor with 16.00 GB RAM PC and Windows 7 operating system. For a given return level of each different instance, the computational time is given in table IV.

TABLE IV
COMPUTATIONAL TIME OF PROPOSED HYBRID ALGORITHM FOR A GIVEN RETURN LEVEL OF 5 BENCHMARK INSTANCES USING 2000 POSSIBILITIES OF SCENARIOS

Instance				min
Index	Q	N_r	N_e^j	
Hang Seng	31	100	20	32.1
DAX 100	85	100	20	57.3
FTSE 100	89	100	20	54.9
S&P 100	98	100	20	63.7
Nikkei 225	225	100	20	75.6
Average				56.7

D. Discussions

The aims of our experiments are to test the effectiveness of our hybrid algorithm and to evaluate the performance of the two-stage stochastic portfolio optimization model. Although we cannot guarantee the optimality of the solutions obtained due to the heuristic nature of our proposed hybrid algorithm, as it has been shown in Section VI-A, for the problem using a minor instance with a small number of scenarios, our algorithm can get the optimal results. For the problem using a larger instance with a bigger number of scenarios, we use the percentage deviation method which is widely applied in the literature for the classic cardinality constrained mean-variance portfolio optimization problem to determine the quality of the solutions obtained. We can see from Section VI-A that the MPE of the optimal solution for Hang Seng instance with 100 possibilities of scenarios is 2.0687%. The average MPE of our results for Hang Seng instance with 2000 possibilities

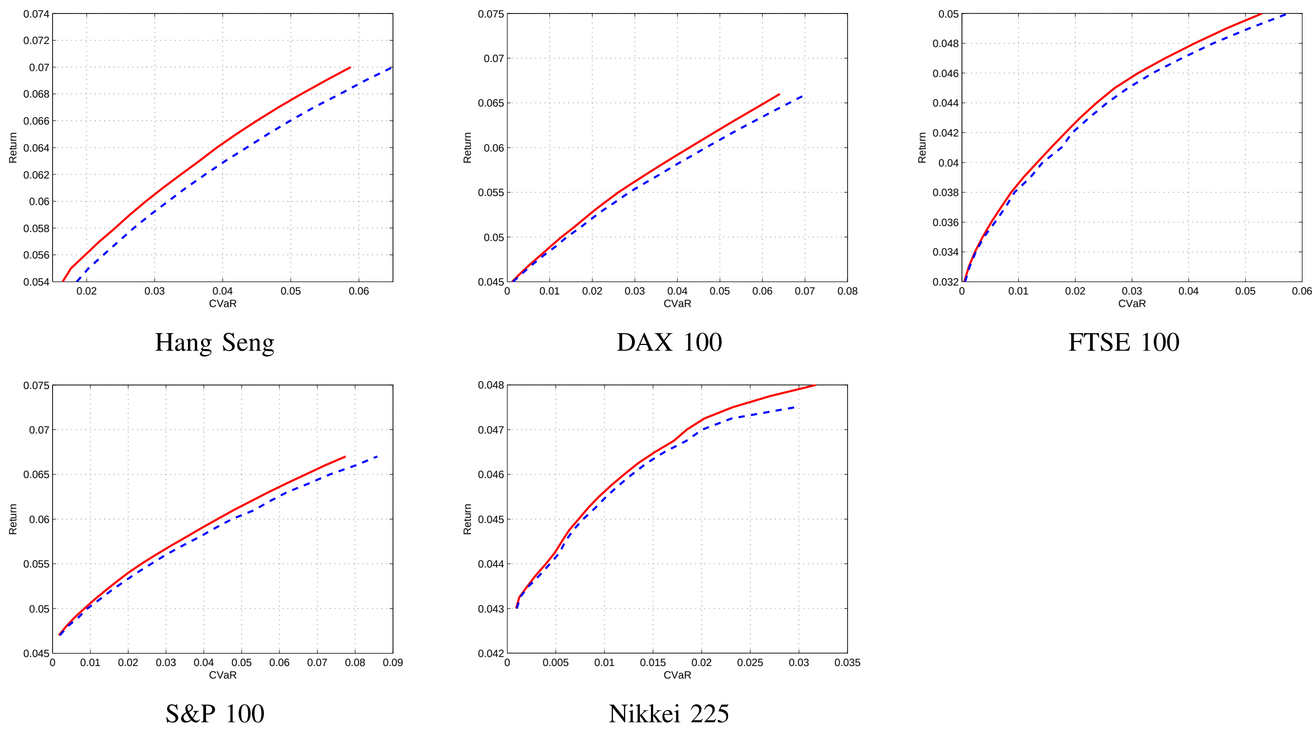

Fig. 4. Comparison of the frontier obtained by our algorithm with the optimal upper bound efficient frontier using 2000 possibilities of scenarios. The red line is the optimal upper bound efficient frontier and the blue dashed line is the final frontier found by our algorithm.

of scenarios is 2.2090%, indicating that our solutions to the overall problem are very promising, at least not far from the optimal ones.

In the literature, it is difficult to conduct fair comparisons of the related work using the two-stage stochastic model. One main reason is that involves a lot of uncertainties. Different scenario generation methods will lead to different scenarios used in the second stage. One interesting observation is, by performing some experiments, we find our two-stage stochastic portfolio optimization model is sensitive to the generated scenarios. For example we can see that the MPE of Hang Seng instance is around 2.2% while the MPE of other instances is less than 1.5%. This is mainly because of the randomness involved in the generated scenarios. To compare and analyze the different scenario generation methods, as well as to reduce the number of scenarios generated in order to reduce the computational complexity is not the main focus of this work. This leads to some possible interesting future research directions.

VII. Conclusion And Further Work

In this work, we investigate a two-stage stochastic portfolio optimization model which minimizes the Conditional Value at Risk (CVaR) of the portfolio loss with a comprehensive set of real world trading constraints. The two-stage stochastic model can capture the market uncertainty in terms of future asset prices therefore it enables the investors rebalancing the assets.

A hybrid approach which integrates genetic algorithm (GA) and an LP solver is proposed for the two-stage stochastic model. The idea is that GA can search for the assets selection heuristically while the LP solver can solve the corresponding reduced sub-problems optimally. The main advantage of such an approach is, by solving a sub-problem, some original constraints can be eliminated and all zero-value variables are dropped off. Therefore it reduces the complexity and narrows the search space. The experimental results indicate that it is very useful strategy for the problems using a stochastic programming model.

We used a copula-based method to generate scenarios for this work. Comparing and analyzing different scenario generation methods, and reducing the number of the scenarios generated in order to reduce the computational complexity further, can be the possible future research directions.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC 71471092, NSFC-RS 71311130142), Ningbo Sci&Tech Bureau (2012B10055).

References

[1] H. Markowitz, "Portfolio Selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, March 1952.
[2] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments*, 2nd ed. Wiley, March 1991.
[3] T. Cui, S. Cheng, and R. Bai, "A combinatorial algorithm for the cardinality constrained portfolio optimization problem," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, July 2014, pp. 491–498.
[4] T. J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha, "Heuristics for cardinality constrained portfolio optimisation," *Computers & Operations Research*, vol. 27, no. 13, pp. 1271–1302, November 2000.

[5] L. Di Gaspero, G. Di Tollo, A. Roli, and A. Schaerf, “Hybrid metaheuristics for constrained portfolio selection problem,” *Quantitative Finance*, vol. 11, no. 10, pp. 1473–1488, October 2011.

[6] M. Woodside-Oriakhi, C. Lucas, and J. E. Beasley, “Heuristic algorithms for the cardinality constrained efficient frontier,” *European Journal of Operational Research*, vol. 213, no. 3, pp. 538–550, September 2011.

[7] R. Baldacci, M. Boschetti, N. Christofides, and S. Christofides, “Exact methods for large-scale multi-period financial planning problems,” *Computational Management Science*, vol. 6, no. 3, pp. 281–306, 2009.

[8] D. Barro and E. Canestrelli, “Dynamic portfolio optimization: Time decomposition using the maximum principle with a scenario approach,” *European Journal of Operational Research*, vol. 163, no. 1, pp. 217–229, 2005, financial Modelling and Risk Management.

[9] J. Li and J. Xu, “Multi-objective portfolio selection model with fuzzy random returns and a compromise approach-based genetic algorithm,” *Information Sciences*, vol. 220, no. 0, pp. 507 – 521, 2013, online Fuzzy Machine Learning and Data Mining.

[10] H. Yano, “Fuzzy decision making for fuzzy random multiobjective linear programming problems with variance covariance matrices,” *Information Sciences*, vol. 272, no. 0, pp. 111 – 125, 2014.

[11] P. Kall and S. Wallace, *Stochastic programming*, ser. Wiley-Interscience series in systems and optimization. Wiley, 1994.

[12] J. Birge and F. Louveaux, *Introduction to Stochastic Programming*, ser. Springer Series in Operations Research and Financial Engineering. U.S. Government Printing Office, 1997.

[13] A. J. King and S. W. Wallace, *Modeling with Stochastic Programming*, ser. Springer Series in Operations Research and Financial Engineering. Springer New York, 2012.

[14] S. W. Wallace and W. T. Ziemba, *Applications of Stochastic Programming*, 1st ed. Society for Industrial and Applied Mathematics, June 2005.

[15] R. Bai, S. W. Wallace, J. Li, and A. Y.-L. Chong, “Stochastic service network design with rerouting,” *Transportation Research Part B: Methodological*, vol. 60, no. 0, pp. 50 – 65, 2014.

[16] R. T. Rockafellar and S. Uryasev, “Optimization of conditional value-at-risk,” *Journal of Risk*, vol. 2, pp. 21–41, 2000.

[17] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, “Coherent measures of risk,” *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.

[18] N. Topaloglou, H. Vladimirou, and S. A. Zenios, “A dynamic stochastic programming model for international portfolio management,” *European Journal of Operational Research*, vol. 185, no. 3, pp. 1501–1524, 2008.

[19] S. J. Stoyan and R. H. Kwon, “A stochastic-goal mixed-integer programming approach for integrated stock and bond portfolio optimization,” *Computers & Industrial Engineering*, vol. 61, no. 4, pp. 1285 – 1295, 2011.

[20] F. He and R. Qu, “A two-stage stochastic mixed-integer program modelling and hybrid solution approach to portfolio selection problems,” *Information Sciences*, vol. 289, no. 0, pp. 190 – 205, 2014.

[21] M. Kaut and S. W. Wallace, “Shape-based scenario generation using copulas,” *Computational Management Science*, vol. 8, no. 1–2, pp. 181–199, 2011.

[22] P. Jorion, *Value at Risk: The New Benchmark for Managing Financial Risk*, 3rd ed. McGraw-Hill Education, 2006.

[23] M. Pritsker, “Evaluating Value at Risk Methodologies: Accuracy versus Computational Time,” *Journal of Financial Services Research*, vol. 12, no. 2, pp. 201–242, October 1997.

[24] G. C. Pflug, “Some remarks on the value-at-risk and the conditional value-at-risk,” in *Probabilistic Constrained Optimization*. Springer US, 2000, vol. 49, pp. 272–281.

[25] M. Woodside-Oriakhi, C. Lucas, and J. Beasley, “Portfolio rebalancing with an investment horizon and transaction costs,” *Omega*, vol. 41, no. 2, pp. 406 – 420, 2013, management science and environmental issues.

[26] M. Dyer and L. Stougie, “Computational complexity of stochastic programming problems,” *Mathematical Programming*, vol. 106, no. 3, pp. 423–432, May 2006.

[27] A. Shapiro and A. Nemirovski, “On complexity of stochastic programming problems,” *Continous Optimization*, pp. 111–146, 2004.

[28] S. J. Stoyan and R. H. Kwon, “A two-stage stochastic mixed-integer programming approach to the index tracking problem,” *Optimization and Engineering*, vol. 11, no. 2, pp. 247–275, 2010.

[29] E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling, “Mip: Theory and practice—closing the gap,” in *System Modelling and Optimization*, ser. IFIP—The International Federation for Information Processing, M. Powell and S. Scholtes, Eds. Springer US, 2000, vol. 46, pp. 19–49.

[30] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Ann Arbor, MI, USA: The University of Michigan Press, December 1975.

[31] J. E. Beasley, “OR-Library: distributing test problems by electronic mail,” *Journal of the Operational Research Society*, vol. 41, no. 11, pp. 1069–1072, 1990.