Noname manuscript No.
(will be inserted by the editor)

A Hybrid Combinatorial Approach to a Two-Stage Stochastic Portfolio Optimization Model With Uncertain Asset Prices

Tianxiang Cui · Ruibin Bai · Shusheng Ding · Andrew J. Parkes · Rong Qu · Fang He · Jingpeng Li

Received: date / Accepted: date

Abstract Portfolio optimization is one of the most important problems in the finance field. The traditional Markowitz mean-variance model is often unrealistic since it relies on the perfect market information. In this work, we propose a two-stage stochastic portfolio optimization model with a comprehensive set of real world trading constraints to address this issue. Our model incorporates the market uncertainty in terms of future asset price scenarios based on asset return distributions stemming from the real market data. Compared with existing models, our model is more reliable since it encompasses real-world trading constraints and it adopts CVaR as the risk measure. Furthermore, our model is more practical because it could help investors to design their future investment strategies based on their future asset price expectations. In order to solve the proposed stochastic model, we develop a hybrid combinatorial approach, which integrates a hybrid algorithm and a Linear Programming (LP) solver for the problem with a large number of scenarios. The comparison of the computational results obtained with 3 different metaheuristic algorithms and with our hybrid approach shows the effectiveness of the latter. The superiority of our model is mainly embedded in solution quality. The results demonstrate that our model is capable of solving complex portfolio optimization problems

Tianxiang Cui, Ruibin Bai
School of Computer Science, The University of Nottingham Ningbo China
E-mail: zx10551@nottingham.edu.cn

Shusheng Ding
School of Business and Research Academy of Belt & Road, Ningbo University, China

Andrew J. Parkes, Rong Qu
School of Computer Science, The University of Nottingham, UK

Fang He
Department of Computer Science, Faculty of Science and Technology, University of Westminster, UK

Jingpeng Li
Division of Computing Science and Mathematics, University of Stirling, UK

with tremendous scenarios while maintaining high solution quality in a reasonable amount of time and it has outstanding practical investment implications, such as effective portfolio constructions.

Keywords Hybrid Algorithm · Combinatorial Approach · Stochastic Programming · Population-based Incremental Learning · Local Search · Learning Inheritance · Portfolio Optimization Problem

1 Introduction

With the advances in computing and the rise of big data, nowadays investment decisions are made not only by the financial experts, but also based on sophisticated mathematical models and 'number crunching' by mathematicians or computer scientists. The high yield of stock has made it a major investment over the past decades. One typical problem in stock market, portfolio optimization, can be described as allocating the limited capital over a number of potential assets in order to achieve investors risk appetites and the return objectives. The first portfolio optimization model was proposed by Markowitz in the 1950s [78, 79], 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 it can be solved in an exact manner with a reasonable computational time.

The standard Markowitz model can be considered as a simplistic unconstrained risk-return model. One shortcoming of the model is that it assumes there exists a perfect market with no taxes or transaction costs, where short selling is not allowed, and the assets are infinitely divisible (i.e. they can be traded in any non-negative proportion). One limitation of such simplistic approach is that some of the trading restrictions of stock markets in the real-world situation are omitted. The basic mean-variance model can be extended to capture market realism. There exists a wide range of real-world trading constraints in practice. The most common examples include the transaction costs [66] (fixed transaction cost, variable transaction cost), the cardinality constraint [81] (which specifies the total number of the held assets in a portfolio in order to reduce tax and transaction costs), the bounding constraints [81] (which specify the lower and upper bound of the proportion of each held asset in a portfolio in order to avoid unrealistic holdings) and the minimum trading size [28] (which specifies the minimum amount of transaction occurred on each asset). As a result, the complexity of the problem significantly increases with the additional real-world trading constraints involved. In most cases, adding new constraints will lead to a nonconvex search space and quadratic programming technique cannot be used anymore. For example, the inclusion of cardinality constraint into the standard Markowitz portfolio optimization model may be referred as the cardinality constrained mean-variance (CCMV) portfolio selection problem [41]. In fact, CCMV is a Quadratic Mixed-Integer problem (QMIP) which has been proven to be NP-hard [82]. Nevertheless, till now, it might be still incapable of providing an exact algorithm to resolve such optimization problem efficiently. As a result, different heuristics and metaheuristic techniques have been applied to solve extended mean-variance portfolio optimization model [22,33,103,30,76].

Although the real world constraints have later been introduced into the classic mean-variance model, there still remains another important market factor, the uncertainty, that complicates the investors making investment decisions. In the current work of mean-variance portfolio optimization problem [22,33,103,30,76], the mean expected return and the covariance between assets are assumed to be static and perfectly known, which is often unrealistic due to the economic turmoil and the market uncertainties in practice. It has been pointed out in [9,13] that the investment decisions should be made based on the consideration of the market uncertainties. Usually, the probabilistic uncertainty factors are taken into account (i.e. the asset price, the currency exchange rate, the prepayments, the external cashflows, the inflation and the liabilities, 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 [70,107,47,48]. In this work, we will focus on the probabilistic uncertainty of the market, more specifically, we use different asset price scenarios as the representation of market uncertainty based on the real market data. In the real world, most of the financial data is non-Gaussian distributed, therefore, the adoption of non-Gaussian distribution would enhance the model performance. The most recent works have also emphasized the non-Gaussian element in the financial model. For example, Alexandru, Elliott, and Ortega [7] apply the non-Gaussian GARCH models into the European options pricing model. Lanne, Meitz and Saikkonen [68] highlight the non-Gaussian component within the structural vector autoregressive (SVAR) model. Chen, Kou and Wang [23] deal with the non-Gaussian noise structure within the Markov decision processes.

Stochastic programming has been well studied for modeling optimization problems with uncertain factors since the late 1950s [57,18,31,32]. It provides a stochastic view to replace the deterministic one in the sense that the uncertain factors are represented by the assumed probability distributions. It can model uncertainty and impose real world constraints in a flexible way[63]. Over the past decades, stochastic programming has been widely applied to financial optimization problems. Models for the management of fixed income securities [44,102,16] and models for asset/liability management [86,87,85,100,97] have been well studied. A wide range of approaches based on stochastic programming for portfolio management have also been developed [100,96,97,39, 50,14,40,35]. For example, Topaloglou, Vladimirou and Zenios [100] 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. Yu *et al.* [109] proposed a dynamic stochastic programming model for bond portfolio management. They modeled the uncertainty in terms of the interest rates. Stoyan and Kwon [97] 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 [49] proposed a stochastic 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 integrating local search and a default Branch-and-Bound method was proposed to solve the problem. A more comprehensive review can be found in [108]. For this work, we will use stochastic programming to model uncertain future asset prices.

The other drawback of the mean-variance model is how it characterizes risk. In the classical Markowitz portfolio optimization model, risk is measured as the standard deviation of the asset returns. The handicaps of standard deviation might be displayed in a twofold manner. Firstly, the standard deviation may imply a quadratic utility function, which suffers from the peculiarity of satiation and irrationality of raising the level of risk aversion [53]. More importantly, standard deviation relies on the assumption of elliptically return distributions, which indicates that standard deviation delivers equivalent penalties to upside and downside market movements. However, it has been shown that market perceives up and down movements asymmetrically. Given the volatility spillover effects, the stocks tend to co-move in down market movements [64,34]. Practically, people may only want to minimize the possibility of the portfolio losses. Over the last 15 years, the study of extreme events (i.e. the tails of the return distribution) has received increased attention due to the use of sophisticated risk control models in financial institutions and the reaction of the academic community to the attempt of imposing inadequate or improper risk measures in the context of regulations [98]. The new definitions of risk measures have been developed. Two commonly used examples are Value at Risk (VaR) [55,89] and Conditional Value at Risk (CVaR) [92].

It is notable that CVaR also dominates VaR as a risk metric in the portfolio selection model circumstance. One reason mentioned in [106], is that CVaR could be easily optimized compared with VaR. It is because that CVaR retains the feature of convexity and thus the availability of global optimal solutions [92,90]. Accordingly, CVaR fits better than VaR in the context of the portfolio optimization. Another shortcoming of VaR is that it is incapable of tackling the loss beyond the pre-specified level and delivers a lowest bound for losses. Therefore, VaR, as a risk measure, is more optimistic than conservative [93]. Likewise, Alexander and Baptista [3] illustrate that for the same confidence level, the CVaR constraint will be more rigorous than the VaR constraint. They conclude that a CVaR constraint may outperform the VaR constraint in most investment cases. More importantly, as a coherent risk measure, CVaR preserves the sub-additive feature, which complies with the investment diversification effect. In other words, CVaR can be diminished through investment diversification. In addition, minimizing CVaR is more effective than minimizing VaR, since a low CVaR usually produces a portfolio with a small VaR [4]. As a result, minimizing CVaR is better than solely minimizing VaR. In summary, CVaR is a coherent risk measure and it outweighs both standard deviation and VaR as a risk measure. Therefore, adoption of CVaR might endorse better performance of the portfolio selection compared with standard deviation and VaR. It would be a worthwhile endeavor to employ the computational method in order to furnish an up-to-date investigation into portfolio management based on the coherent risk measure of CVaR.

In the portfolio optimization domain, Glasserman and Xu [42] proposed a series of portfolio control rules to handle the model errors within the portfolio optimization problem. They applied a stochastic notion of robustness to reflect model error uncertainty. Gaivoronski, Krylov and Wijst [40] investigated different approaches to portfolio selection based on different risk characterizations. They proposed an algorithm to determine whether to rebalance a given portfolio based on transaction costs and new market condition information. Greco and Matarazzo [45] proposed an approach for portfolio selection in a non-Markowitz way. The uncertainties are modeled in terms of a series of meaningful quantiles of probabilistic distributions. They proposed an Interactive Multiobjective Optimization (IMO) method based on Dominance-based Rough Set Approach (DRSA) to solve the model in two phases. Chen and Wang [24] introduced a hybrid stock trading system based on Genetic Network Programming and mean-CVaR model (GNP-CVaR). The proposed model combines the advantages of statistical models and artificial intelligence in the sense that CVaR measures the market risk and distributes the weights of capital to each asset in the portfolio and GNP decides the trading strategies. More recently, there are also burgeoning studies emerged in portfolio optimization field. Ban et al. [12] use two machine learning methods to optimize portfolio constructions. They uncover that one machine learning method named performance-based regularization overwhelmingly dominates all other solutions including sample average approximation, which is the other machine learning method. Xidonas et al. [105] introduce future returns scenarios into resolving portfolio selection problems. On the basis, they construct portfolios through the conventional minimax regret criterion formulation, which enables their model to solve multiobjective portfolio optimization problem. Lwin et al. [77] propose an efficient hybrid multi-objective evolutionary algorithm (MODE-GL) to accommodate mean-VaR portfolio optimization problems with real-world constraints. They reveal that the MODE-GL performs more favorably compared with two other existing techniques. Ahmadi-Javid and Fallah-Tafti [2] introduce a risk measure with monotone property named entropic value-at-risk (EVaR) into portfolio optimization. They unveil that EVaR-based portfolios could have better return rates and Sharpe ratios compared with CVaR-based portfolios.

Our approach of investigating risky asset allocation problem is based on an integrated simulation and optimization framework with the adoption of CVaR as the risk measure as well as the real world trading constraints. Our scenario-based optimization model incorporates the future asset price uncertainty within the joint distributions of asset returns. The benefits of this methodology exhibit in a twofold fashion. Firstly, our model legitimizes investors to take the future asset price uncertainty into considerations in the meantime of minimizing CVaR. This model simulation result could help investors to entail an approximation of the cardinality constrained efficient frontier with the risk measure CVaR instead of standard deviation. Investors can then select their investments according to their risk preferences along the approximated cardinality constrained efficient frontier. As a result, our model becomes highly relevant to investors with different risk preferences, including risk averse, risk neutral and risk seeking investors. Secondly, our cardinality constrained approximated efficient frontier could serve as a basis to engender the approximated Capital Market Line (CML) with the risk measure CVaR. Therefore, investors could be not only capable of choosing the portfolio investment along the approximated efficient frontier, but also capable of allocating investment between risk free asset and portfolio investment along the approximated CML based on the risk measure CVaR.

One common method used in the literature to deal with stochastic portfolio optimization model is decomposition. Benders decomposition [35], scenario decomposition [94], time decomposition [14] and other novel decomposition methods [97] have been proposed. The problem is simplified when it is decomposed into different parts.

In our previous work [29], we improved the stochastic portfolio optimization models in the literature [100, 49] and proposed a hybrid algorithm for the two-stage stochastic portfolio optimization problem with a comprehensive set of real world trading constraints. A Genetic Algorithm (GA) together with a commercial LP solver was used where GA searches for the assets selection heuristically and the LP solver solves the corresponding sub-problems optimally. The proposed hybrid GA can solve the problem to a good degree of accuracy, however, it has a slow convergent speed. The standard GA relies heavily on designing effective genetic operators which are highly specific to different problems. In order to solve the two-stage stochastic model more efficiently, in this work, we propose a light weight approach based on Population Based Incremental Learning (PBIL). The whole idea of PBIL is on learning statistically, leading to efficiency and effectiveness compared to GA. Instead of relying on heavy computations, PBIL employs light weight learning adaptively on the fly. It intends to solve the model via adaptive learning upon a larger number of scenarios. Local search, hash search, elitist selection and partially guided mutation are also adopted in order to enhance the evolution.

The outline of the rest part is as follows: Section 2 introduces the background information. Section 3 gives the statement of the problem as well as the corresponding notations. The detailed description of our hybrid combinatorial approach is given in Section 4. The datasets and scenario generation methods are described in Section 5. Experimental results are presented in Section 6. The final conclusion is given in Section 7.

2 Preliminaries

2.1 Stochastic programming

This section presents preliminaries for the two-stage stochastic programming, including scenario tree definition and the calculation of percentile risk function, such as VaR and CVaR.

In real-world situations, many parameters of a problem are not precisely known but are subject to uncertainties - due to future events and human variabilities, etc. Generally, there are two approaches to deal with the uncertainties:

- *Robust Optimization*: when the uncertain variables are given within some certain boundaries, robust optimization is applied for such problems. The idea is to find a solution which is feasible for all the data and optimal for the worst case scenario.
- *Stochastic Programming*: when the probability distribution of the uncertain variables are known or can be estimated, stochastic programming is applied for such problems. The idea is to find a policy which is feasible for all (or at least almost all) possible data instances and maximizes/minimizes the expectation of the objective function with the decision and random variables involved. The decision-maker can gather some useful information by solving such models either analytically or numerically.

For this work, we use stochastic programming to deal with the uncertain future asset prices. The comprehensive concepts of stochastic programming can be found in [57,18].

2.2 Two-stage stochastic programming problem with recourse

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 the temporarily available information.

The simplest case of the recourse problem have two stages:

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

Formally, the two-stage stochastic programming problem with recourse can be described as the follows [108]:

$$\begin{aligned} \min \quad & f(x) + E[\mathcal{Q}(x, \xi)] \\ s.t. \quad & \\ & Ax = b \\ & x \in \mathbb{R}^n \end{aligned} \tag{1}$$

where ξ represents the uncertain data, $f(x)$ is the objective function where x is the first-stage decision variable vector which should be decided before the uncertain variables are revealed, $E[\mathcal{Q}(x, \xi)]$ is the expectation value of the function $\mathcal{Q}(x, \xi)$ and $\mathcal{Q}(x, \xi)$ is the optimal value for the following nonlinear program:

$$\begin{aligned} \min \quad & q(u, \xi) \\ s.t. \quad & \\ & W(\xi)u = h(\xi) - T(\xi)x \\ & u \in \mathbb{R}^m \end{aligned} \tag{2}$$

where u is the vector of the second-stage decision variables which depends on the realization of the first-stage uncertain variables. $q(u, \xi)$ represents the second-stage cost function. $W(\xi)$, $h(\xi)$ and $T(\xi)$ are model parameters with reasonable dimensions. These parameters are the functions of the uncertain data ξ, therefore they are also random. W is the recourse matrix, and h is the second-stage resource vector. T is the technology matrix which contains the technology coefficients, therefore it can convert the first-stage decision variable vector x into resources for the second-stage problem.

Hence, the general two-stage stochastic programming problem with recourse can be rewritten as follows:

$$\begin{aligned} \min \quad & f(x) + E[\min\{q(u, \xi) | W(\xi)u + T(\xi)x = h(\xi)\}] \\ s.t. \quad & \\ & Ax = b \\ & x \in \mathbb{R}^n \\ & u \in \mathbb{R}^m \end{aligned} \tag{3}$$

In this formulation, a "here and now" decision x is made before the uncertain data ξ is realized. At the second stage, after the value of the uncertain data ξ is revealed, we can modify our behavior by solving the corresponding optimization problem.

The recourse problem is not restricted to the two-stage formulation and it is possible to extend the problem into a multistage model.

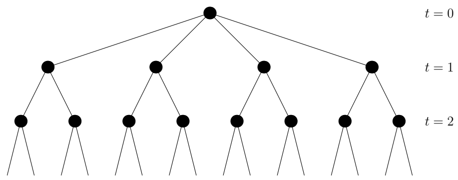


Fig. 1 An example of a scenario tree. At stage $t = 0$ there is 1 scenario, at stage $t = 1$ there are 4 scenarios and at stage $t = 2$ there are 8 scenarios.

2.3 Scenario tree

There are two common methods which can be used to deal with the multistage stochastic programming problems, namely decision rule approximation and scenario tree approximation. For this work, we will focus on 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 the node n is represented by $a_{t-1}(n)$. The branching probability of the node n is denoted by p_n which is a conditional probability on its parent node $a_{t-1}(n)$. The path to the 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 [59].

2.4 Percentile Risk Function

2.4.1 Value at Risk (VaR)

In the real-world situation, portfolio managers may only need to reduce the possibility of the high loss. Value at Risk (VaR) [55,89] 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 $x \in \mathbb{R}^n$ is the decision vector and $\xi \in \mathbb{R}^m$ is the uncertain (random) vector. The uncertain variable is a measurable function from an uncertainty space to the set of real number-

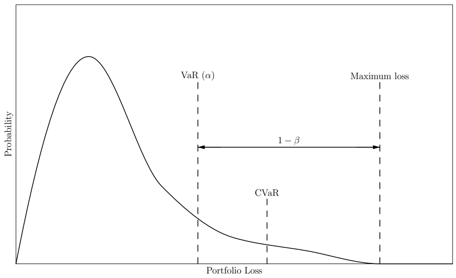


Fig. 2 Value at Risk (VaR) and Conditional Value at Risk (CVaR)

s[72]. 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:

$$\Psi(x, \alpha) = \int_{f(x,\xi)\leq\alpha} p(\xi)d\xi \tag{4}$$

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:

$$\alpha_\beta(x) = \min\{\alpha \in \mathbb{R} : \Psi(x, \alpha) \geq \beta\} \tag{5}$$

However, VaR is inadequate for market risk evaluation. As it has been pointed out in [6], VaR does not satisfy the sub-additivity and the convexity and generally it is not a coherent risk measure (VaR is only coherent for standard deviation of normal distributions). Also, VaR is difficult to optimize using scenarios[80]. Furthermore, VaR does not take the distribution of the loss exceeding the threshold into account and it would become unstable if there is a sharp and heavy tail in the loss distribution.

2.4.2 Conditional Value at Risk (CVaR)

Conditional Value at Risk (CVaR) (also called Mean Excess Loss, Mean Expected Shortfall, or Tail VaR) is proposed in [92] in order to eliminate the drawbacks of VaR. CVaR is a more consistent risk measure because of its sub-additivity and the convexity [6], and it is proven to be a coherent risk measure [88].

CVaR calculates the average value of the loss which exceeds the VaR value (see Figure 2). Formally, CVaR is defined as the follows [92]:

$$\phi_\beta(x) = (1 - \beta)^{-1} \int_{f(x,\xi)\geq\alpha_\beta(x)} f(x, \xi)p(\xi)d\xi \tag{6}$$

The function above is a little bit difficult to handle because the VaR value $\alpha_\beta(x)$ is involved in it. Alternatively, we can have the analytical representation to replace VaR. A simpler function can be used instead of CVaR:

$$F_\beta(x,\alpha) = \alpha + (1-\beta)^{-1} \int_{f(x,\xi)\le\alpha} (f(x,\xi)-\alpha)p(\xi)d\xi \tag{7}$$

It has been proved in [92] 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.

2.4.3 Minimizing CVaR

From the definitions of VaR and CVaR we can see that given a specified probability level β, β-CVaR should always be greater or equal to β-VaR. In fact, we can optimize CVaR and obtain VaR simultaneously by minimizing the function $F_\beta(x,\alpha)$ [101]. Suppose we have the solution of the minimization of $F_\beta(x,\alpha)$, (x^*,α^*), then the optimal CVaR value equals $F_\beta(x^*,\alpha^*)$ and the corresponding VaR value equals α^*.

We can minimize the function $F_\beta(x,\alpha)$ by introducing an auxiliary function $Z(\xi)$ such that $Z(\xi) \ge f(x,\xi) - \alpha$ and $Z(\xi) \ge 0$. Formally we have the following:

$$\begin{aligned}
\min \quad & \alpha + (1-\beta)^{-1} E(Z(\xi)) \\
s.t. \quad & \\
& Z(\xi) \ge f(x,\xi) - \alpha \\
& Z(\xi) \ge 0 \\
& \alpha \in \mathbb{R}
\end{aligned} \tag{8}$$

Now let us consider the portfolio optimization problem. Here the uncertain data ξ refers to the future asset prices. Normally the analytical representation of density function $p(\xi)$ is not available but instead the scenarios can be generated from the historical observations of each asset price. The scenario generation can use the property matching method [52,51] or even simply Monte Carlo simulations. Suppose we have generated N scenarios from the density $p(\xi)$, y_n where $n = 1,\ldots,N$. Function $F_\beta(x,\alpha)$ can be therefore calculated as the follows:

$$F_\beta(x,\alpha) = \alpha + (1-\beta)^{-1} \sum_{i=1}^{N} p_i (f(x,y_i)-\alpha)^+ \tag{9}$$

where $f(x,y_i)$ is the portfolio loss function in scenario i and it is defined as the negative of the total portfolio return. p_i is the probability of scenario i and $(f(x,y_i)-\alpha)^+ = \max(0,(f(x,y_i)-\alpha))$. By introducing the auxiliary function $Z(\xi)$ and we can have the auxiliary variable z_i where $z_i \ge f(x,y_i) - \alpha, z_i \ge 0, i = 1,\ldots,N$. Therefore the minimization of the function $F_\beta(x,\alpha)$ can be reduced to the simplified form:

$$\min \quad \alpha + (1-\beta)^{-1} \sum_{i=1}^{N} p_i z_i \tag{10}$$

$$\begin{aligned} s.t.& \\ & z_i \geq f(x, y_i) - \alpha \quad i = 1, \ldots, N \\ & z_i \geq 0 \quad i = 1, \ldots, N \\ & \alpha \in \mathbb{R} \\ & x \in \mathbb{R}^n \end{aligned}$$

It has been showed in [5,67,92] that such formulation can provide the numerically stable technique to the problem with a large number of scenarios.

3 Model Statement

This section exhibits the two-stage stochastic model formulation, including model notations and model constraints.

3.1 Notations

We first introduce the notions that will be applied in the two-stage stochastic model as follows:

- Set:
  - A: The set of assets $A = \{a_1, \ldots, a_n\}$. Index i, $i \in A$.
  - N_r: The set of recourse nodes. One node corresponds to one recourse portfolio. Index j, $j \in N_r$.
  - N_e^j: The set of evaluate nodes on recourse node j where $j \in N_r$. Index e, $e \in N_e^j$.
- User-specific parameter:
  - μ: The target return specified by the investor.
  - β: The Quantile (percentile) for VaR and CVaR.
  - M: A very large constant.
- Deterministic input data:
  - h: The initial amount of cash to invest.
  - w_i^0: The initial position of asset a_i (in number of units).
  - η_b: The fixed buying cost.
  - η_s: The fixed selling cost.
  - ρ_b: The variable buying cost.
  - ρ_s: The variable selling cost.
  - K: The number of asset held in the portfolio (cardinality).
  - w_{min}: The minimum holding position.
  - t_{min}: The minimum trading size.
- Scenario dependent data:

  - p_j: The probability of recourse node j in the second stage.
  - $p_{(j,e)}$: The probability of evaluate node e of recourse node j in the second stage.
  - P_i^0: The price of asset a_i in the first stage (per unit).
  - P_i^j: The price of asset a_i on recourse node j in the second stage (per unit).
  - $P_i^{(j,e)}$: The price of asset a_i on evaluate node e of recourse node j in the second stage (per unit).
  - V^0: The initial portfolio wealth.
  - V^j: The final portfolio wealth on recourse node j.
  - R^j: The final portfolio return on recourse node j.
- Auxiliary variable:
  - z_j: Portfolio shortfall in excess of VaR at recourse node j.
  - α: The optimal VaR value.
- Decision variable:
  - b_i: The number of units of asset a_i purchased in the first stage.
  - s_i: The number of units of asset a_i sold in the first stage.
  - w_i: The final position of asset a_i in the first stage.
  - b_i^j: The number of units of asset a_i purchased on recourse node j in the second stage.
  - s_i^j: The number of units of asset a_i sold on recourse node j in the second stage.
  - w_i^j: The final position of asset a_i on recourse node j in the second stage.
  - c_i: The binary holding decision variable in the first stage. $c_i = 1$ if a non-zero value of asset i is held after the first stage decision.
  - f_i: The binary buying decision variable in the first stage. $f_i = 1$ if asset i is chosen to be bought in the first stage.
  - g_i: The binary selling decision variable in the first stage. $g_i = 1$ if asset i is chosen to be sold in the first stage.
  - c_i^j: The binary holding decision variable on recourse node j in the second stage. $c_i^j = 1$ if a non-zero value of asset i is held after the recourse decision in scenario j.
  - f_i^j: The binary buying decision variable on recourse node j in the second stage. $f_i^j = 1$ if asset i is chosen to be bought in scenario j.
  - g_i^j: The binary selling decision variable on recourse node j in the second stage. $g_i^j = 1$ if asset i is chosen to be sold in scenario j.

3.2 Two-stage stochastic portfolio optimization model with recourse

The model we used for this work is the same as our previous work [29]. Inspired by [100], the original form of the model was proposed in [49]. Although [49] is formulated as a two stage model, it does not include a possibility that the costs and values change after the recourse decision is enacted. Hence the recourse in that model could have no monetary effect, and so would obtain the same decisions as a simpler single stage formulation. A contribution of this present work is to extend the model so that values can change after the recourse, and nontrivial recourse decisions can improve the portfolio performance. The proposed model is divided into two stages.

$$\min\left(\alpha + (1-\beta)^{-1}\sum_{j\in N_r} p_j z_j\right) \tag{11}$$

s.t.

First Stage - Portfolio Selection:

$$w_i = w_i^0 + b_i - s_i, \quad \forall i \in A \tag{12}$$

$$h + \sum_{i\in A}(s_i P_i^0) - \sum_{i\in A}(\eta_s g_i + s_i \rho_s P_i^0) = \sum_{i\in A}(b_i P_i^0) + \sum_{i\in A}(\eta_b f_i + b_i \rho_b P_i^0) \tag{13}$$

$$\sum_{i\in A} c_i = K \tag{14}$$

$$w_{\min} c_i \leq w_i \quad \forall i \in A \tag{15}$$

$$t_{\min} f_i \leq b_i \quad \forall i \in A \tag{16}$$

$$t_{\min} g_i \leq s_i \quad \forall i \in A \tag{17}$$

$$f_i + g_i \leq 1 \quad \forall i \in A \tag{18}$$

$$f_i M \geq b_i \quad \forall i \in A \tag{19}$$

$$g_i M \geq s_i \quad \forall i \in A \tag{20}$$

$$b_i M \geq f_i \quad \forall i \in A \tag{21}$$

$$s_i M \geq g_i \quad \forall i \in A \tag{22}$$

$$w_i,\ b_i,\ s_i \in \mathbb{R} \tag{23}$$

$$c_i,\ f_i,\ g_i \in \mathbb{B} \tag{24}$$

Second Stage - Recourse:

$$w_i^j = w_i + b_i^j - s_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{25}$$

$$\sum_{i\in A}(s_i^j P_i^j) - \sum_{i\in A}(\eta_s g_i^j + s_i^j \rho_s P_i^j) = \sum_{i\in A}(b_i^j P_i^j) + \sum_{i\in A}(\eta_b f_i^j + b_i \rho_b P_i^j) \quad \forall j \in N_r \tag{26}$$

$$\sum_{i\in A} c_i^j = K \quad \forall j \in N_r \tag{27}$$

$$w_{\min} c_i^j \leq w_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{28}$$

$$t_{\min} f_i^j \leq b_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{29}$$

$$t_{\min} g_i^j \leq s_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{30}$$

$$f_i^j + g_i^j \leq 1 \quad \forall i \in A,\ \forall j \in N_r \tag{31}$$

$$f_i^j M \geq b_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{32}$$

$$g_i^j M \geq s_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{33}$$

$$b_i^j M \geq f_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{34}$$

$$s_i^j M \geq g_i^j \quad \forall i \in A,\ \forall j \in N_r \tag{35}$$

$$V^j = \sum_{\substack{e \in N_e^j \\ i \in A}} p_{(j,e)} P_i^{(j,e)} w_i^j \quad \forall j \in N_r \tag{36}$$

$$R^j = V^j - V^0 \quad \forall j \in N_r \tag{37}$$

$$z_j \geq -R^j - \alpha \quad \forall j \in N_r \tag{38}$$

$$z_j \geq 0 \quad \forall j \in N_r \tag{39}$$

$$\sum_{j \in N_r} p_j R^j \geq \mu \tag{40}$$

$$w_i^j,\ b_i^j,\ s_i^j \in \mathbb{R} \tag{41}$$

$$c_i^j,\ f_i^j,\ g_i^j \in \mathbb{B} \tag{42}$$

$$\alpha,\ z_j \in \mathbb{R} \tag{43}$$

The objective function (11) calculates the β-percentile CVaR of the portfolio loss at the end of the second stage, where α is the corresponding optimal VaR value.

Equation (12) is the first stage asset balance condition and Equation (25) is the second stage asset balance condition. The idea is straightforward, the current position of an asset should be equal to its initial holding (in the first/second stage) plus/minus the corresponding buying/selling amount (in the first/second stage).

Equations (13) and (26) are the cash balance conditions for the first and second stage, respectively. The idea is that the cash inflows should equal to the cash outflows in both stages (i.e. no cash left). In financial practice, transaction costs have the significant effects on portfolio selection, and ignoring them could lead to inefficient portfolios [91]. The effects of transaction cost on the portfolio constructions have been unveiled [62]. Large transaction costs might narrow the potential securities involved in the portfolio constructions and thus result in the portfolio structure adjustment. In the real world situation, the transaction costs are often formulated via non-convex functions [66], which make the optimization problem more challenging. In traditional financial literature, transaction cost is often modeled as a linear function that is proportional to the trading size [69]. For our work, we adopt a more general form of transaction cost, which encompasses both a fixed transaction cost and a linear variable transaction cost to both buying and selling an asset [73]. The transaction costs can be explicitly written as the follows:

- No holdings of an asset: transaction cost = 0;
- Selling an asset: transaction cost = $\eta_s + \rho_s * SellingValue$;
- Buying an asset: transaction cost = $\eta_b + \rho_b * BuyingValue$;

Therefore it results in a non-convex function, as the transaction cost decreases relatively when the trading amount increases [66, 74].

Equations (14) and (27) 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 (15) and (28) 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 (16) and (17) are the minimum trading conditions for the first stage and equations (29) and (30) are the minimum trading conditions for the second stage. The idea is to prevent it from trading a very small proportion of an asset.

Buying and selling the same asset at the same time is not allowed, which is given in Equation (18) for the first stage and in Equation (31) 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 (19), (20), (21) and (22) for the first stage and constraints (32), (33), (34) and (35) 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 to 1; if the decision variables for buying/selling an asset is 0, then the corresponding binary decision variables should be 0 and vice versa.

Equations (36), (37) 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. An early version of this model [49] uses the same set of scenarios to calculate the portfolio return (i.e. $P_i^j = P_i^{(j,e)}$), therefore it does not include a possibility that the costs and values change after the recourse decision is enacted. Hence the recourse in that model could have no monetary effect, and so would obtain the same decisions as a simpler single stage formulation. The current model allows values change after the recourse, and nontrivial recourse decisions can improve the portfolio performance.

Equations (38), (39) define the excess shortfall z_j of the recourse portfolio where $z_j = \max[0, -R^j - \alpha]$ for each recourse node.

Equation (40) gives the minimum portfolio target return μ.

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. 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 [104] to relax these decision variables to having continuous values.

4 The Proposed PBIL-based Hybrid Approach

This section gives the procedure of PBIL-based hybrid approach, including hash search, local search process and relevant algorithm details.

Exact methods and metaheuristic approaches are two successful streams for solving combinatorial optimization problems. Over the last few years, many works have been developed on building hybrids of exact methods and metaheuristic approaches. In fact, many real-world problems can be practically solved much better using hybrid strategies since the advantages of both types of methods are simultaneously exploited. For this work, we integrate metaheuristic approaches with exact methods. The idea is, we divide the two-stage stochastic problem into two parts. The first part is to determine the asset combination by the PBIL-based hybrid algorithm while the second part is to calculate the optimal weights of the selected assets by a LP solver correspondingly. Local search, hash search, elitist selection and partially guided mutation are also adopted in order to enhance the evolution.

4.1 Overview of PBIL

Population Based Incremental Learning (PBIL) was originally introduced by Baluja [11,10]. It is one of the simplest form of Estimation of Distribution Algorithms (EDAs). It combines genetic algorithms and competitive learning for optimization problems by evolving the entire population rather than each single individual members. The idea is, a probability vector which represents the distribution of all individual alleles (variables) is updated by learning from the best and the worst solutions during the evolution. Mutation is also performed on the probability vector in order to help preserve diversity. Then the new generation of population is created based on the updated probability vector. PBIL is closely related to GA, but it is simpler and more efficient since it does not require all the mechanisms of a standard GA.

4.2 Problem representation

In this work, a PBIL-based hybrid algorithm is utilized to evolve the best values for discrete variables c_i in the stochastic model. The search space is different for different benchmark datasets (characterized by Q, see Section 5). 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:

- One probability vector $\mathbf{v} = (v_1, v_2, ..., v_Q)$ of size Q represents the possibility of each asset to be chosen in the portfolio.
- One binary vector $\mathbf{c} = (c_1, c_2, ..., c_Q)$ of size Q is used to denote if asset is chosen in the portfolio.
- One vector $\mathbf{k} = (k_1, k_2, ..., k_K)$ of size K is used to represent the selected K assets of the portfolio where $k_i \in \{1, 2, \ldots, Q\}$ $(i = 1, \ldots, K)$ and $k_i \neq k_j \forall i, j$.
- The evaluation of vector $\mathbf{k}$ is done by calculating a fitness function F which is implemented using a standard LP solver. It maps from a list of K integers (i.e. the selected portfolio with K assets) and a target return μ to a real number: $F(\mathcal{Z}^K, \mu) \rightarrow \mathcal{R}$. For a given target return level μ, F calculates the CVaR value $\mathcal{R}$ of a selected portfolio.
- The probability vector is updated by learning from the best and the worst solutions obtained from the population at the end of each generation.

– Elitist selection is used in our PBIL-based hybrid algorithm, i.e., we keep the best solution 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 the PBIL-based hybrid algorithm used in this work is given in Algorithm 1, and the parameters are given in Section 6.1.

Algorithm 1: PBIL-based hybrid algorithm for searching the set of assets

for *i = 1 to Q* **do**
    $v_i = 0.5$;
while *stopping criteria are not met;*
do
    Generating individuals: Create a population of individuals (see section 4.3);
    Hash search: Search the infeasible solution hash table $HashInfeasibleSolution$ and bad solution hash table $HashBadSolution$ in the archive. If it is very similar to the entries of the hash table, re-generate (see section 4.4);
    Evaluation: Evaluate each individual's fitness by using CPLEX LP solver (see section 4.5);
    Local search: Perform the local search for the top 20% individuals (see section 4.6);
    Archive: Record the current best solution. Add the current worst solution to $HashBadSolution$ and add all the infeasible solutions to $HashInfeasibleSolution$ (see section 4.7);
    Update: Update the probability vector $\mathbf{v}$ by learning from the current best and the current worst solution (see section 4.8);
    Mutation: Mutate the probability vector $\mathbf{v}$ by using partially guided mutation (see section 4.9);
    Elitism: Select the best individual from the current generation and insert it into the next new generation;

4.3 Generating individuals

The probability vector $\mathbf{v}$ is used to determine whether asset i is chosen in the portfolio. Initially v_i is set to 0.5 where $i = 1, \ldots, Q$ so that every asset can have an equal chance to be chosen. The binary vector $\mathbf{c}$ is created according to $\mathbf{v}$, if asset i is selected, $t_i = 1$; otherwise $t_i = 0$. Vector $\mathbf{k}$ is used to represent the chosen assets in the portfolio, and it is generated based on the vector $\mathbf{c}$. The idea is to choose exact K number of assets to form the portfolio in order to satisfy the cardinality constraint. Suppose there are K' assets selected in $\mathbf{c}$. If $K' \geq K$, we randomly choose K among K' assets and insert them into $\mathbf{k}$. If $K' < K$, we first insert K' assets into $\mathbf{k}$ and then randomly choose another $K - K'$ different assets and insert them into $\mathbf{k}$ as well.

It might seem that using both vector $\mathbf{c}$ and vector $\mathbf{k}$ is redundant. The idea is that vector $\mathbf{c}$ is used for learning purpose, as we need to trace the whole efficient frontier, and vector **k** is only used for one point (portfolio) each time. The details are given in Section 6.3.

4.4 Hash search

By using our two-stage stochastic model, for any given asset combination, it does not necessarily always lead to a feasible solution. We maintain a hash table to keep all the infeasible solutions explored. We also have a hash table to keep all the worst feasible solutions of each generation (see section 4.7). Each time when a new individual is generated, we check its similarity with the existing entries in the hash tables (i.e. how many assets are identical in the new individual compared with the existing entries in the hash tables). If it is very similar (just one selection different from an infeasible solution, or just one or two from a bad solution) to the existing entries, we discard it and re-generate the individual. As pointed out in [43], good solutions tend to have similar structures. Although there may be many bad solutions and we may not be able to record all of them, we can still discard the solutions that have the similar structures with the known bad ones since they tend to have the poor fitness values as well. There are two advantages of performing the hash search. Firstly the computational cost of the hash table lookup is amortized $O(1)$ ($O(1)$ on average, $O(n)$ for the worst case), which is cheaper than calling the LP solver, therefore it improves the efficiency; secondly it can explore different areas of the solution space by avoiding unnecessary search, and it may explain why, in Figure 3, PBIL with the hash search tends to obtain better global solutions. Details of the hash search are given in Algorithm 2.

Algorithm 2: Hash search

```
for each new individual h generated do
    for each entry in HashInfeasibleSolution do
        if h has at least K-1 identical assets compared with the entries in the hash
        table then
            re-generate individual h;
            break;
    for each entry in HashBadSolution do
        if h has at least K-2 identical assets compared with the entries in the hash
        table then
            re-generate individual h;
            break;
```

4.5 Evaluation

The fitness of the individual generated is evaluated by solving the corresponding sub-problem using an LP solver in order to get the weight allocation of

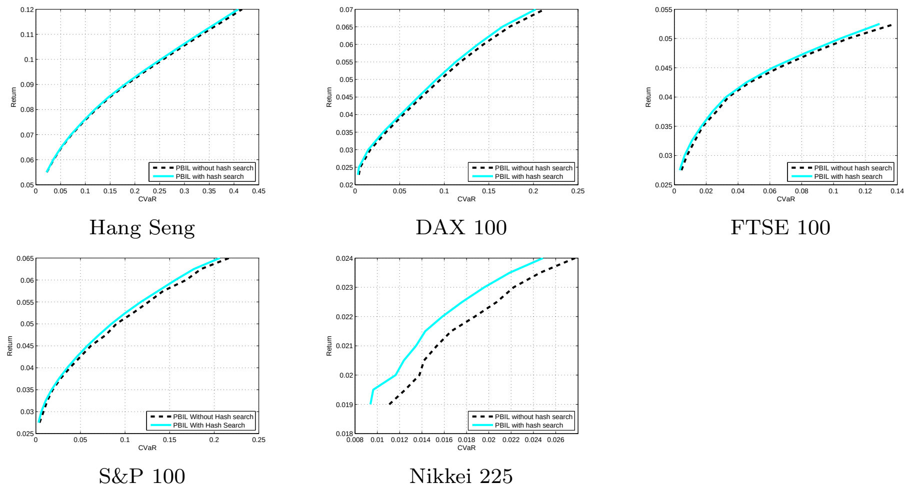


Fig. 3 Comparative results of PBIL with and without hash search for 5 general market instances. PBIL with hash search dominates PBIL without hash search, as for each return level, PBIL with hash search can obtain better (smaller) CVaR value.

the selected assets. We can control the numerical properties of the solutions to the sub-problems by setting up different Markowitz threshold [1] (which is used to control the kinds of pivots permitted) and the time allowed for each fitness calculation. That means we do not need to compute the optimal values for every individual since the chosen assets may not be global optimal, thus searching for optimal weights is not necessary. Rather, we only need to calculate the optimal value once for the global best solution after the search of the hybrid algorithm is finished. This will help to improve the efficiency.

4.6 Local search

After the evaluation is done, the top 20% individuals with the best fitness value of the population are selected and the local search are applied to them in order to seek for better solutions and evolve better individuals within a neighbourhood. Each time we replace one asset with a neighbourhood asset and then re-evaluate the new portfolio. The neighborhood relation of an asset is defined as the asset with the closest probability according to the probability vector **v**. If a better solution is obtained, the current best solution is updated. For each asset, the number of neighbours we search is controlled by the parameter *na* (i.e. *na* closest probability successors). The local search applied here is an incomplete neighbourhood search. It aims to seek for possible improvements of the current solution. Figure 4 shows that the local search can indeed help the algorithm to find better global solutions.

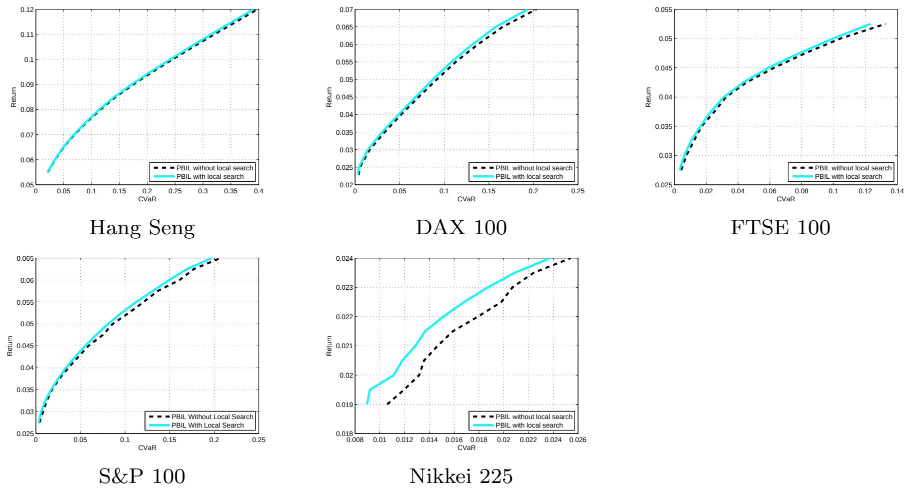


Fig. 4 Comparative results of PBIL with and without local search for 5 general market instances. PBIL with local search dominates PBIL without local search, as for each return level, PBIL with local search can obtain better (smaller) CVaR value.

4.7 Archive

As mentioned in Section 4.4, during the evolution, it is possible to obtain infeasible solutions. It is important to keep an archive of them. We use a hash table to record all the infeasible solutions obtained. Similarly, we use a hash table to keep all bad solutions of every iteration. The purpose of maintaining the two hash tables is to avoid unnecessary search so that better solutions can be explored. The main computational cost is on the evaluations of the solutions during the search, thus pre-selection or filtering based on learning avoid wasting time on less promising solutions. We also keep a record of the best solutions obtained at each iteration to ensure that the good solutions found by the algorithm are not lost (i.e. elitist selection).

4.8 Update

In PBIL, the probability vector $\mathbf{v}$ can be considered as a prototype vector which is used to store the knowledge collected during the evaluation of current generation in order to guide the following population generations. $\mathbf{v}$ is updated by learning from the current best solution s_i^{cbest} and the current worst solution s_i^{cworst} using a positive learning rate lr and a negative learning rate $nelr$ correspondingly. Thus, the learning rates are used to control the speed of the prototype vector shifting to the better solution vector and the portions of exploration of the search space [11,95]. Details of the probability updated are given in Algorithm 3.

Algorithm 3: Probability update

```
for i = 1 to Q do
    v_i = v_i × (1 − lr) + s_i^{cbest} × lr;
    if s_i^{cbest} ≠ s_i^{cworst} then
        v_i = v_i × (1 − nelr) + s_i^{cbest} × nelr;
```

4.9 Mutation

At the end of each generation, the probability vector **v** is mutated according to a certain mutation probability mp. In this work, we use a mutation strategy, namely partially guided mutation[76]. It gives an equal chance to mutate **v** either randomly or based on the global best solution using a mutation rate mr. The advantage of doing this is that it can exploit the good structures in the current best solutions as well as giving chance to exploring other regions of the search space at the same time. Details of the partially guided mutation are given in Algorithm 4.

Algorithm 4: Mutation (partially guided mutation)

```
for i = 1 to Q do
    if rand(0, 1] < mp then
        if rand(0, 1] < 0.5 then
            r = Rand[0, 1];
            v_i = v_i × (1 − mr) + r × mr ;
        else
            v_i = s_i^{cbest} ;
```

5 Datasets And Scenarios

This section generates scenarios based on stock market datasets from OR-Library, including Hang Seng, DAX 100, FTSE 100, S&P 100, and Nikkei 225. Detailed stability test have also been attempted.

5.1 Benchmark Datasets

In this work, we use the five benchmark instances which are extended from the OR-Library [15]. 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, $Q = 225$.

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

5.2 Distribution

It is well known that vast of financial literatures often postulate the asset return follows a normal distribution. However, the fat-tail feature of financial data distribution has been documented [36]. The asset return distribution will be noted as the fat-tail distribution when the distribution is leptokurtic. The leptokurtic distribution will be defined based on the kurtosis of the distribution. For normal distribution, kurtosis is equal to 3 and for those fat-tail distributions, kurtosis is usually greater than 3 [8]. Tables 1 calculates the mean value, the standard deviation, the skewness and the kurtosis of the asset return for the benchmark market index. It could be observed that asset returns in all five markets exhibit leptokurtic characteristics. In other words, the kurtosis of asset return distributions in all five markets are greater than 3, especially for the Nikkei 225 index and the FTSE 100 index. Therefore, it can be concluded that the empirical asset return distributions are fat-tail distributions than normal distributions given the market data we have selected.

Table 1 Four moments of the asset return for benchmark market index

Instance Index	N	Mean	stdev	Skewness	Kurtosis
Hang Seng	31	0	0.03	-0.19	3.93
DAX 100	85	0	0.02	-0.24	3.61
FTSE 100	89	0	0.02	0.37	5.15
S&P 100	98	0	0.02	0.11	3.71
Nikkei 225	225	0	0.03	0.28	4.68
Average		0	0.02	0.07	4.22

5.3 Scenario Generation

Since the distribution of the asset return might not be known, a plethora of studies have applied Gaussian distribution to estimate the real asset return distribution. This approach might be precarious and is often prone to errors. Copious of empirical studies in the literature [17, 20, 25] discover that parameters perturbation may result in huge estimation errors and thereby parameters estimations request a high level of accuracy. However, the parameters estimated from real data are often susceptible to estimation errors. As argued in [90], the portfolio solutions generated by the optimization process massively rely on the parameters estimation. The approach to approximate the data parameters may put theoretical and numerical results into a suspicious status. Thus, non-robust parameter inputs might lead to unreliable asset allocation, which, in turn, yields undesired out-of-sample performance [19, 71]. As mentioned in [99], the uncertain returns of the investment alternatives play important roles in portfolio selection model. Thus, the representation and incorporation of the uncertain returns will be crucial during the model establishment. Our approach, on the other hand, has unbounded the specified form of distribution and inferred the asset return distribution stemming from real market data. We have employed the copula-based method [59] that can turn distribution estimation into scenarios.

The scenarios are generated in order to represent the uncertain asset prices. The model has two stages. We use the price information on the recourse nodes to form the initial portfolio and use the price information on the successor evaluate nodes to perform the portfolio rebalancing actions. In this case, the full data of 261 weekly historical prices from the OR-Library cannot be used directly as 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. Starting from week 2, we compute the ratio 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 apply the copula scenario generation method [59] using the 260 new price data as the inputs to generate 400 recourse node scenarios. The evaluate nodes should be only dependent on their predecessor recourse node. Therefore, for each recourse node, we use the price scenario on that node and multiply a random coefficient within $(0.9, 1.1)$ to produce 40 corresponding scenarios for each of the evaluate nodes. The random coefficients are used to simulate the fluctuation of asset price in the second stage. It has been showed that market turmoil is linked with strong correlations between stocks [56]. Based on the empirical fact, our model is multivariate structured and takes into account the correlation matrix among stocks by utilizing the copula function. This correlation matrix consideration allows us to alleviate the market estimation errors during the highly fluctuated periods. We do not claim they are the optimal choices. There will be $400 \times 40 = 16000$ possibilities of scenarios in total and the evaluate scenarios are different for each different recourse node. By performing some experiments, we found the computational results were sensitive to the scenarios generated, especially for the evaluate node scenarios. Again, we do not claim the scenario generation methods we used are the best choices. Our primary aim is rather to develop an efficient method that can solve the two-stage stochastic portfolio optimization problem with a larger number of scenarios and to test the effectiveness of our hybrid combinatorial approach.

Since the scenarios are generated from the real market data, the past winners and losers in the market are automatically identified in the scenarios, which helps our model to reserve the potential of developing momentum-based investment strategies. The momentum phenomena in the stock market has been widely reported in the financial literature [54,27]. Fama and French [38] also enhance their three-factor model [37] with a momentum factor based on the previous work of Carhart [21]. The remarkable momentum returns across 23 countries have been demonstrated in their work. More importantly, it has been showed that the prevalence of momentum phenomena not only exist in the stock market, but also in other financial markets such as currency markets, commodity markets, and bond futures markets [84]. Based on the momentum fact, momentum strategies for the portfolio construction have been developed [26]. The profitability of momentum strategies has also been verified in the finance research spectrum [83,46]. Therefore, our model that takes asset future price into account can effectively identify different potential future winners under different scenarios. Given the identified potential future winners, possible momentum strategies can be then formulated with the consideration of CVaR. The distinguish feature of our model in terms of developing momentum strategies is that our model does not entirely rest on the past asset performance information, but also envisages the future asset performance. Therefore, our model could help investors to develop the momentum strategies with both future asset price uncertainty and the risk metric of CVaR.

5.4 Stability

One potential drawback of CVaR is that the CVaR will be considerably unstable when the asset return distribution is fat-tail [106]. The fat-tail distribution was evident in the financial literatures. For example, the normality of daily asset returns using the Dow Jones Industrial Stocks was tested in [36]. The evidence can be found that the sample was generated by a distribution which is leptokurtic or "fat-tailed" relative to the normal distribution. In real-world situations, most of the financial return distributions are fat-tail [75,61,65], and in fact, we have demonstrated that the return distributions for the benchmark market index from the OR-library are also fat-tail. Our model's stability might be challenged because of the CVaR adoption in the model. Consequently, the stability of our model's results has been scrutinized by the virtue of out-of-sample simulations.

In stochastic programming, scenario generation methods are used to create a limited discrete distribution from the input data. The statistical properties of the scenario sets created should match the corresponding values estimated from the input data, and the scenario generation method should not lay bias on the results by causing instability of the solutions. Usually, the stability tests are performed [60,58] and there are two types of stability.

- *In-sample stability:* The scenario generation method is assessed in terms of its ability to match the benchmark distribution. We generate several scenario sets of a given size using the same input data. The idea is, no matter which scenario set we choose, the optimal objective value of the model should be approximately the same. The objective values should not vary across scenario sets. For this work, we use copula-based scenario generation method to generate 25 different scenario trees with the size 400 using the same input data. Then we use CPLEX to compute the optimal objective value of a same target return level for each scenario tree and compare the results. Ideally these results should be equal.

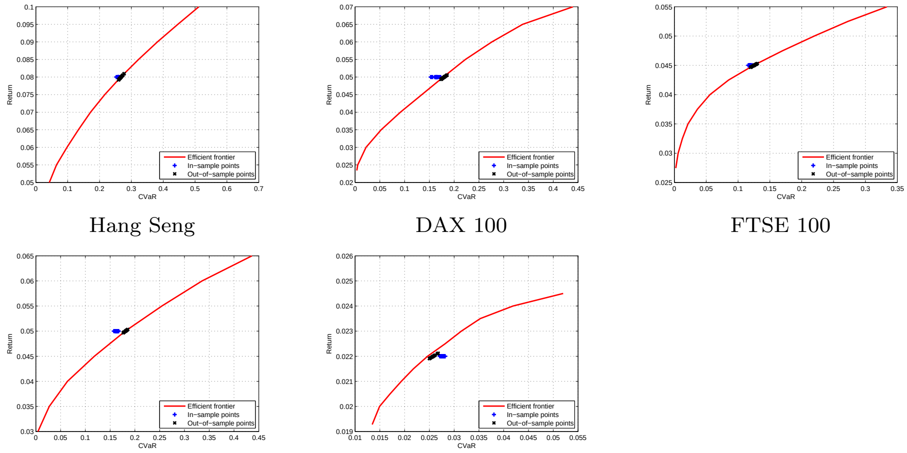
– *Out-of-sample stability:* The scenario generation method is assessed in terms of its ability to provide the stable results with respect to the benchmark distribution. We generate several scenario sets of a given size using the same input data and solve the model with each scenario set. The idea is, if we simulate the solutions obtained for each scenario set on the benchmark distribution, the value of the true objective function should be approximately the same. Again, we use copula-based scenario generation method to generate 25 different scenario trees with the size 400 using the same input data and then use CPLEX to solve the model with a same target return level for each scenario tree. After that we simulate the solutions obtained for each scenario tree on the benchmark distribution to compute the true objective function. It is important that the benchmark distribution is not generated by the same method we are using and in our case, we use the input data directly as our benchmark distribution. Finally we compare the results. Ideally these results should be equal, and they should be also equal to the in-sample values (approximately).

For this work, the copula-based scenario generation method is only used to create the scenario sets for the recourse nodes. The scenarios for the evaluate nodes are dependent on their predecessor nodes and the random coefficients are also involved. Therefore, we only examine the stability tests on the single-stage model (i.e. without rebalancing actions). The purpose of performing the stability tests here is to show the copula-based scenario generation method will not influence the results and it is a suitable scenario generation method for this work.

The results are shown in Figure 5. Tables 2 and 3 calculate the mean value, the median value and the standard deviation of the in-sample and out-of sample results, respectively. The simulation results have demonstrated that our model's performance is fairly stable since the results from in-sample and out-of-sample simulations have little fluctuation. Thus, we can conclude that the scenario generation method we use is effective, in the sense that it will not cause instability in the solutions of the model.

Table 2 In-sample stability test results for 5 general market instances using 400 scenarios

Instance			Mean(%)	Median(%)	stdev(%)
Index	Q	N_r			
Hang Seng	31	400	25.8579	25.8947	0.2408
DAX 100	85	400	16.3507	16.4025	0.4873
FTSE 100	89	400	11.9221	11.9367	0.1805
S&P 100	98	400	16.2933	16.3385	0.2646
Nikkei 225	225	400	2.7621	2.7564	0.0321
Average					0.2411



Hang Seng DAX 100 FTSE 100

S&P 100 Nikkei 225

Fig. 5 Stability test results for 5 general market instances of the one-stage model using 400 scenarios. The in-sample points are very close to the out-of-sample points, indicating that both in-sample and out-of-sample simulations have little fluctuation.

Table 3 Out-of-sample stability test results for 5 general market instances using 400 scenarios

Instance Index	Q	N_r	Mean(%)	Median(%)	stdev(%)
Hang Seng	31	400	26.7367	26.6460	0.5159
DAX 100	85	400	17.8633	17.8262	0.3993
FTSE 100	89	400	12.5446	12.5495	0.3330
S&P 100	98	400	18.0863	18.1676	0.2857
Nikkei 225	225	400	2.5850	2.5751	0.0610
Average					0.3190

6 Experimental Results

This section exhibits experiment results of the portfolio compositions based on our model. Both model parameters and algorithmic parameters have been taken into account. Our model results has been compared with other 3 algorithms for the performance evaluation.

6.1 Parameter settings

The parameter settings used in this work are shown as follows:

6.1.1 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 $\rho_b = 0.5\%$, fixed selling cost $\eta_s = 0.5$, variable selling cost $\rho_s = 0.5\%$, cardinality $K = 10$, minimum holding position $w_{min} = 1\%$ and minimum trading size $t_{min} = 0.1\%$. The initial portfolio only involves cash, and we set the initial cash $h = 100000$. For the model demonstration purpose, we assume the probability of each scenario is equal, and therefore $p_j = 1/400 = 0.0025$, and $p_{(j,e)} = 1/40 = 0.025$. However, the probability of each scenario can be set differently based on investors' future expectations of future asset prices. Thus, our model is capable of designing investment strategies customized by investors' future expectations.

6.1.2 Algorithmic parameters

We set population size $Po = 200$, number of generations $Ge = 50$, mutation rate $mr = 0.05$, mutation probability $mp = 0.05$ and number of neighbourhood assets $na = 15$.

The learning rate has a big effect on our hybrid algorithm. The algorithm will focus on searching using the information gained about the search space by using a larger learning rate, which is called exploitation. On the other hand, the algorithm will jump to other areas in the search space by using a lower learning rate, which is called exploration. In order to choose suitable learning rates, we test four different sets of learning rates and run a simple ranking test (if one set of learning rates obtains the best (minimum) CVaR value, we rank it as 1; if one set of learning rates obtains the second-best CVaR value, we rank it as 2 and so on). The results are shown in Table 4.

Table 4 Average ranks (aveRk) of the hybrid combinatorial algorithm with different sets of learning rates for 5 general market instances using 16000 possibilities of scenarios

Instance					$lr=$	0.001	0.01	0.1	1
Index	Q	$\lvert N_r\rvert$	$\lvert N_e^j\rvert$		$nelr=$	0.00075	0.0075	0.075	0.75
Hang Seng	31	400	40	aveRk		1.1429	1.1429	**1.0000**	**1.0000**
DAX 100	85	400	40	aveRk		1.4545	1.3636	**1.1818**	1.3636
FTSE 100	89	400	40	aveRk		1.6000	1.5000	**1.4000**	2.1000
S&P 100	98	400	40	aveRk		1.9286	**1.2143**	1.9286	2.1429
Nikkei 225	225	400	40	aveRk		**2.0000**	2.2500	2.7500	3.0000
Average				aveRk		1.6252	**1.4942**	1.6521	1.9213

There is always a trade-off between exploration and exploitation. In our context, exploration refers to the ability of our hybrid algorithm to fully search the entire market instance, while exploitation refers to the ability of our hybrid algorithm to use the knowledge learned about the assets to narrow down the future search. The lower the learning rates are set, the wider the areas of the instance will be searched. Likewise, the higher the learning rates are set, the more the refinement of the current solutions will be obtained. For this work, the search space of each market instance is different from each other. We can see from Table 4 that, the larger learning rates lead to better performance for the smaller instances. $lr = 1$, $nelr = 0.75$ and $lr = 0.1$, $nelr = 0.075$ have the best average rank for Hang Seng index (with $Q = 31$), and $lr = 0.1$, $nelr = 0.075$ has the best average rank for DAX 100 index (with $Q = 85$) and FTSE 100 index (with $Q = 89$). As the size of the instance increases, the smaller learning rates tend to perform better. $lr = 0.01$, $nelr = 0.0075$ has the best average rank for S&P 100 index (with $Q = 89$), and $lr = 0.001$, $nelr = 0.00075$ has the best average rank for Nikkei 225 index (with $Q = 225$).

One strategy of our hybrid algorithm is, the information gained from the previous return levels can be carried to the next following return levels. Therefore, we can set the lower learning rates in the first half of the return levels to have a better exploration and set the lower learning rates in the second half of the return levels in order to have a better exploitation. For Hang Seng, DAX 100 and FTSE 100 instances, we set the positive learning rate $lr = 0.1$, the negative learning rate $nelr = 0.075$ for the first 10 return levels and then change to $lr = 1$, $nelr = 0.75$ for the last 10 return levels. For S&P 100 and Nikkei 225 instances, we set the positive learning rate $lr = 0.001$, the negative learning rate $nelr = 0.00075$ for the first 10 return levels and then change to $lr = 0.01$, $nelr = 0.0075$ for the last 10 return levels.

Please note that as our main purpose is to test the effectiveness of our hybrid combinatorial approach, we do not claim these parameter settings are the optimal choices. Our primary aim is rather to develop an efficient method that can solve the two-stage stochastic portfolio optimization problem with a larger number of scenarios.

6.2 Comparison of computational results for the 5 general benchmark instances

The main idea of our combinatorial approach is the decomposition of the two-stage stochastic model into two parts. The first part is to search for the selection of assets, and the second part is to determine the corresponding weights of the selected assets. The first part is solved by using PBIL-based hybrid algorithm, and the second part can be solved by a standard LP solver.

Considering the time limitation, we choose 20 equally spaced return levels and for each return level, we run our hybrid algorithm to obtain a portfolio. The set of the portfolios obtained can form a frontier which represents the trade-offs between the expected return and the CVaR value which is a risk indicator.

In order to test the effectiveness of our proposed hybrid algorithm, we compare our results with three 3 different approaches. These 3 approaches use GA mutation only, PSO and random search for the first part respectively, while the second part is solved by the same LP solver. All these 3 approaches also maintain an archive (see section 4.7) and perform the hash search at each iteration (see section 4.4) in order to avoid unnecessary search, and to explore wider solution space. The comparative results can be found in Figure 6.

We run each of different algorithms 10 times and take the simple ranking test. The final average ranks of the 4 different algorithms for 20 return levels are shown in Table 5.

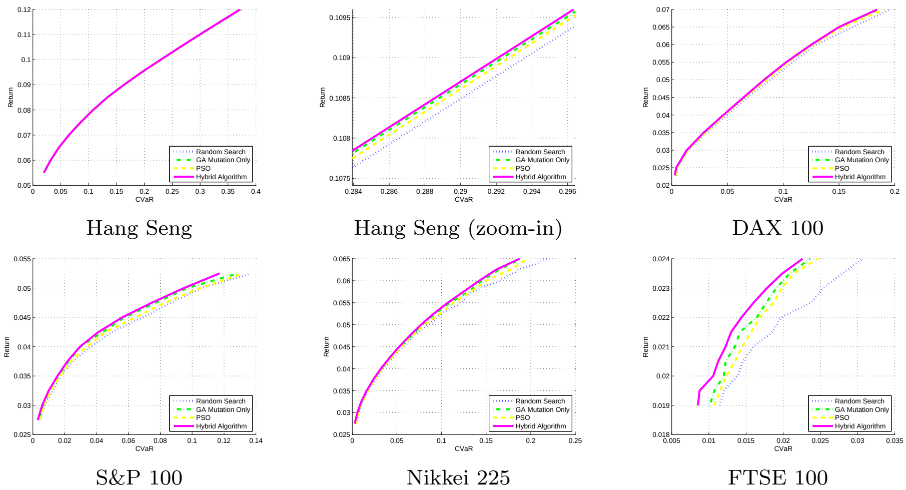


Fig. 6 Comparative results of the hybrid algorithm with 3 other different approaches for 5 general market instances using 16000 possibilities of scenarios. The proposed hybrid algorithm dominates other 3 approaches, as for each return level, our hybrid algorithm can obtain better (smaller) CVaR value.

Table 5 Average ranks (aveRk) of the 4 different algorithms for 5 general market instances using 16000 possibilities of scenarios

Index	Instance Q	$\|N_r\|$	$\|N_e^j\|$		Hybrid -Algorithm	GA -Mutation	PSO	Random -Search
Hang Seng	31	400	40	aveRk	**1.0000**	1.8571	3.1429	3.7857
DAX 100	85	400	40	aveRk	**1.0000**	1.9091	3.0000	4.0000
FTSE 100	89	400	40	aveRk	**1.0000**	2.0000	3.0000	4.0000
S&P 100	98	400	40	aveRk	**1.0000**	2.0000	3.0000	4.0000
Nikkei 225	225	400	40	aveRk	**1.0000**	2.0000	3.0000	4.0000
Average				aveRk	**1.0000**	1.9532	3.0286	3.9571

From Figure 6 and Table 5 we can see that our hybrid combinatorial algorithm outperforms all the other 3 algorithms on all 5 instances.

6.3 Portfolio composition

As we mentioned previously, good solutions tend to have similar structures. In fact, for this two-stage stochastic model, good solutions for two consecutive return levels also share some similarities. For each market instance, we run our hybrid algorithm for 10 equally spaced return levels to obtain the assets selections. The results are shown in Figure 7. We calculate the average similarities for two consecutive return levels and the results are shown in Table 6. We can see that for two consecutive return levels, there are approximately 6.53 out of 10 identical asset choices on average.

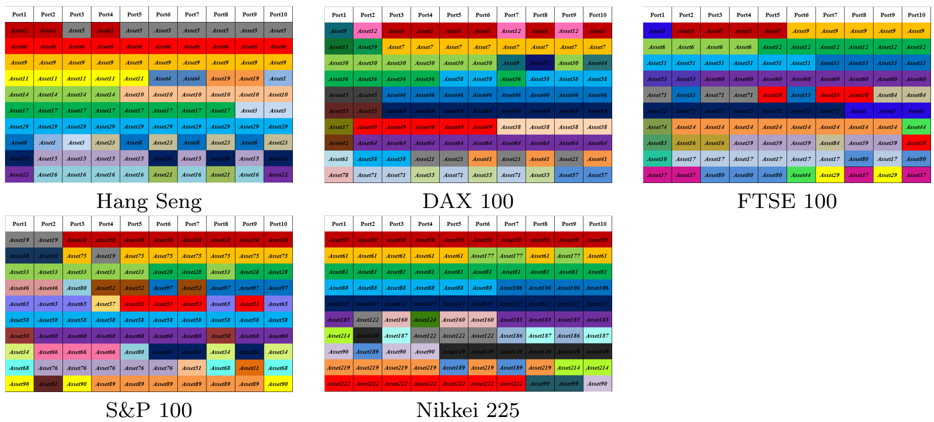


Fig. 7 Portfolio composition results of our hybrid algorithm for 5 general market instances using 16000 possibilities of scenarios. Each column represents the assets selection of the best portfolio obtained for one specified return level. One portfolio is composed of 10 different assets represented by 10 different color sectors. The same asset is represented by the same color sector in each market instance.

Table 6 Average similarities for two consecutive return levels of 5 general market instances using 16000 possibilities of scenarios

	Instance			
Index	Q	$\|N_r\|$	$\|N_e^j\|$	Average similarities
Hang Seng	31	400	40	6.78
DAX 100	85	400	40	6.11
FTSE 100	89	400	40	6.78
S&P 100	98	400	40	6.11
Nikkei 225	225	400	40	6.89
Average				6.53

This observation can be used to guide our search. The idea is, we keep the best solution of one return level and use it as the starting search point of the next return level. This mechanism can be adopted in all 4 algorithms mentioned in Section 6.2. Another important component in our hybrid algorithm, the probability vector, also contains useful information. Derived from the ideas used in competitive learning, the whole population is defined as the probability vector representation. Therefore the good asset tends to have a high probability to be selected. The knowledge learned in one return level can be transferred to the next return level. The probability vector is adjusted accordingly in each generation and in each return level and is gradually shifted towards representing better solutions.

6.4 Performance

All the algorithms for the two-stage stochastic portfolio optimization model mentioned in section 6.2 were 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 market instance, the computational time is given in table 7.

Table 7 Computational time of the 4 different algorithms for 5 general market instances using 16000 possibilities of scenarios

Instance								
Index	Q	$\|N_r\|$	$\|N_e^j\|$		Hybrid -Algorithm	GA -Mutation	PSO	Random -Search
Hang Seng	31	400	40	min	17	15	15	11
DAX 100	85	400	40	min	31	30	30	23
FTSE 100	89	400	40	min	32	30	31	23
S&P 100	98	400	40	min	35	34	34	24
Nikkei 225	225	400	40	min	56	54	54	45
Average				min	34.2	32.6	32.8	25.2

In order to conduct fair comparisons between the algorithms, all the tests were run under the same condition (i.e. the same number of generations). The performance of 4 algorithms are shown in Figure 8. As we can see that our hybrid algorithm converges within less than 50 generations for all 5 market instances while the other 3 algorithms fail to converge within 100 generations. In fact, our hybrid algorithm can achieve better results with less computational time compared to the other 3 algorithms. The hybrid algorithm is based on PBIL which has an important component, the probability vector. It enables learning during the whole execution in the sense that the knowledge from the previous return levels can be inherited to problems with the similar return levels.

A concern is that, the local search adopted in our hybrid algorithm can be also adopted in GA with mutation only and PSO. The idea is that, the probability vector in our hybrid algorithm can provide a more meaningful neighbourhood structure, therefore the local search are much more effective compared to using a random neighbourhood structure (i.e. replacing an asset with a random one).

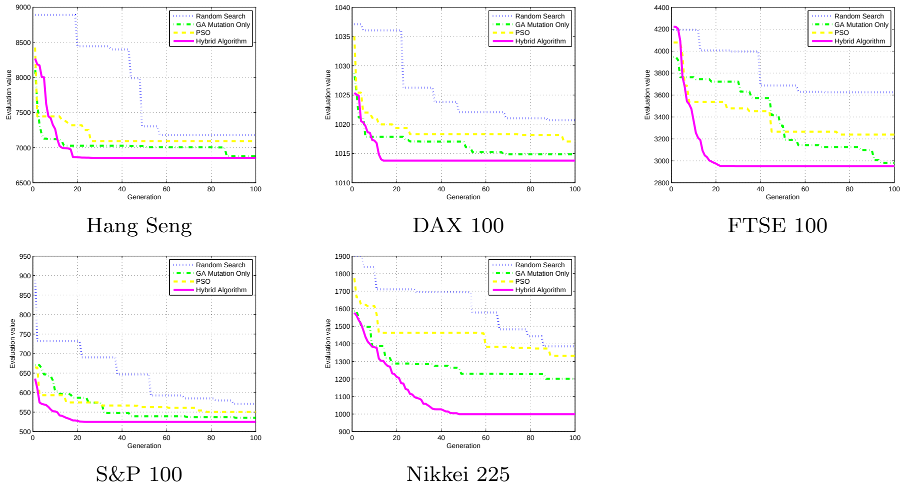


Fig. 8 The performance of 4 different algorithms for 5 general market instances using 16000 possibilities of scenarios. The proposed hybrid algorithm has the fastest convergence speed compared with the other 3 approaches.

7 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. Due to its difficulty, the problem has not been much explored in the literature. The two-stage stochastic model can capture the market uncertainty in terms of future asset prices therefore it enables the investors rebalancing the assets. Practically, our model can help investors for portfolio investment in a twofold fashion. Firstly, our optimization model takes a set of real world constraints into account, adopting a more reliable risk measure, CVaR, which makes our model more reliable. Secondly, our model incorporates future asset price uncertainty based on different scenarios. Consequently, our scenario-based model is more applicable to investors since they can design their future investment strategies in accordance with their future asset price expectations, which can be embodied in the model scenarios with customized probabilities. More importantly, our future asset price scenarios are produced from the real market data distributions. In fact, the copula-based scenario generation method can deal with any possible distributions, which strongly reinforces the applicability of our model since it allows us to capture the unique features of data distributions. Therefore, our model can be easily extended to develop momentum-based investment strategies in the future work.

A key contribution of this paper is that it develops an effective hybrid combinatorial approach for the two-stage stochastic portfolio optimization model. The proposed solution approach integrates a hybrid algorithm and an LP solver in the sense that hybrid algorithm can search for the assets selection heuristically while the LP solver can solve the corresponding reduced subproblems optimally. The hybrid algorithm for assets searching is based on PBIL while local search and hash search are adopted in order to solve the two-stage stochastic model with a larger number of scenarios effectively and efficiently. Elitist selection and partially guided mutation are also adopted in order to enhance the evolution.

Comparison results against 3 other algorithms are given for 5 general market instances and our hybrid combinatorial approach outperforms the 3 algorithms on all instances. We also investigate the structure of the solutions obtained and we demonstrate that the knowledge learned in one return level can be inherited to the next following return levels. This can enhance the search process and makes the whole execution more efficient. The effects of different learning rates are also examined in order to choose for the better settings for the hybrid algorithm. The proposed solution approach in this paper may have potential to be applied to other hard problems in a wide spectrum of OR applications where cardinality is a major concern.

Ethical Statements

The authors declare that they have no conflict of interest.

References

1. *IBM ILOG CPLEX V12.1 User's Manual for CPLEX*, 2009.
2. Amir Ahmadi-Javid and Malihe Fallah-Tafti. Portfolio optimization with entropic value-at-risk. *European Journal of Operational Research*, 279(1):225–241, 2019.
3. Gordon J. Alexander and Alexandre M. Baptista. A comparison of var and cvar constraints on portfolio selection with the mean-variance model. *Management Science*, 50(9):1261–1273, 2004.
4. Siddharth Alexander, Thomas F Coleman, and Yuying Li. Minimizing cvar and var for a portfolio of derivatives. *Journal of Banking & Finance*, 30(2):583–605, 2006.
5. Fredrik Andersson, Stanislav Uryasev, and Stanislav Uryasev. Credit risk optimization with conditional value-at-risk criterion. *Mathematical Programming*, 89(2):273–291, January 2001.
6. Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.
7. Alexandru Badescu, Robert J. Elliott, and Juan-Pablo Ortega. Non-gaussian {GARCH} option pricing models and their diffusion limits. *European Journal of Operational Research*, 247(3):820 – 830, 2015.
8. Kevin P. Balanda and H. L. Macgillivray. Kurtosis: A critical review. *The American Statistician*, 42(2):111–119, 1988.
9. R. Baldacci, M.A. Boschetti, N. Christofides, and S. Christofides. Exact methods for large-scale multi-period financial planning problems. *Computational Management Science*, 6(3):281–306, 2009.
10. Shumeet Baluja and Rich Caruana. Removing the genetics from the standard genetic algorithm. Technical report, Pittsburgh, PA, USA, 1995.
11. Shummet Baluja. Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. Technical report, Pittsburgh, PA, USA, 1994.
12. Gah-Yi Ban, Noureddine El Karoui, and Andrew EB Lim. Machine learning and portfolio optimization. *Management Science*, 64(3):1136–1154, 2016.

13. Diana Barro and Elio Canestrelli. Dynamic portfolio optimization: time decomposition using the maximum principle with a scenario approach. *European Journal of Operational Research*, 163(1):217–229, 2005. Financial Modelling and Risk Management.
14. Diana Barro and Elio Canestrelli. Dynamic portfolio optimization: Time decomposition using the maximum principle with a scenario approach. *European Journal of Operational Research*, 163(1):217 – 229, 2005. Financial Modelling and Risk Management.
15. J. E. Beasley. OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.
16. Andrea Beltratti, Andrea Consiglio, and StavrosA. Zenios. Scenario modeling for the management of international bond portfolios. *Annals of Operations Research*, 85(0):227–247, 1999.
17. Michael J. Best and Robert R. Grauer. On the sensitivity of mean-variance-efficient portfolios to changes in asset means: Some analytical and computational results. *The Review of Financial Studies*, 4(2):315, 1991.
18. J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. U.S. Government Printing Office, 1997.
19. Fischer Black and Robert Litterman. Global portfolio optimization. *Financial Analysts Journal*, 48(5):28–43, 1992.
20. Mark Broadie. Computing efficient frontiers using estimated parameters. *Annals of Operations Research*, 45(1):21–58, 1993.
21. Mark M Carhart. On persistence in mutual fund performance. *The Journal of finance*, 52(1):57–82, 1997.
22. T. J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13):1271–1302, November 2000.
23. Ningyuan Chen, Steven Kou, and Chun Wang. A partitioning algorithm for markov decision processes with applications to market microstructure. *Management Science*, 2017.
24. Yan Chen and Xuancheng Wang. A hybrid stock trading system using genetic network programming and mean conditional value-at-risk. *European Journal of Operational Research*, 240(3):861 – 871, 2015.
25. Vijay K Chopra and William T Ziemba. The effect of errors in means, variances, and covariances on optimal portfolio choice. *Journal of Portfolio Management*, 19(2):6–11, 1993.
26. Jennifer Conrad and Gautam Kaul. An anatomy of trading strategies. *Review of Financial studies*, 11(3):489–519, 1998.
27. Michael J Cooper, Roberto C Gutierrez, and Allaudeen Hameed. Market states and momentum. *The Journal of Finance*, 59(3):1345–1365, 2004.
28. Y. Crama and M. Schyns. Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research*, 150(3):546 – 571, 2003.
29. Tianxiang Cui, Ruibin Bai, Andrew J. Parkes, Fang He, Rong Qu, and Jingpeng Li. A hybrid genetic algorithm for a two-stage stochastic portfolio optimization with uncertain asset prices. In *proceedings of the Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 2519–2525, May 2015.
30. Tianxiang Cui, Shi Cheng, and Ruibin Bai. A combinatorial algorithm for the cardinality constrained portfolio optimization problem. In *proceedings of the Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 491–498, July 2014.
31. George B. Dantzig. *Linear Programming and Extensions*. Rand Corporation Research Study. Princeton Univ. Press, Princeton, NJ, 1963.
32. George B. Dantzig. Linear programming under uncertainty. *Management Science*, 50(12 Supplement):1764–1769, December 2004.
33. Luca Di Gaspero, Giacomo Di Tollo, Andrea Roli, and Andrea Schaerf. Hybrid metaheuristics for constrained portfolio selection problem. *Quantitative Finance*, 11(10):1473–1488, October 2011.
34. Paul Embrechts, Sidney I Resnick, and Gennady Samorodnitsky. Extreme value theory as a risk management tool. *North American Actuarial Journal*, 3(2):30–41, 1999.

35. L.F. Escudero, A. Garín, M. Merino, and G. Pérez. A two-stage stochastic integer programming approach as a mixture of branch-and-fix coordination and benders decomposition schemes. *Annals of Operations Research*, 152(1):395–420, 2007.
36. Eugene F. Fama. The behavior of stock-market prices. *The Journal of Business*, 38(1):34–105, 1965.
37. Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3 – 56, 1993.
38. Eugene F. Fama and Kenneth R. French. Size, value, and momentum in international stock returns. *Journal of Financial Economics*, 105(3):457 – 472, 2012.
39. Stein-Erik Fleten, Kjetil Høyland, and Stein W Wallace. The performance of stochastic dynamic and fixed mix portfolio models. *European Journal of Operational Research*, 140(1):37 – 49, 2002.
40. Alexei A. Gaivoronski, Sergiy Krylov, and Nico van der Wijst. Optimal portfolio selection and dynamic benchmark tracking. *European Journal of Operational Research*, 163(1):115 – 131, 2005. Financial Modelling and Risk Management.
41. Jianjun Gao and Duan Li. Optimal cardinality constrained portfolio selection. *Operations Research*, 61(3):745–761, 2013.
42. Paul Glasserman and Xingbo Xu. Robust portfolio control with stochastic factor dynamics. *Operations Research*, 61(4):874–893, 2013.
43. Fred Glover and Manuel Laguna. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.
44. Bennett Golub, Martin Holmer, Raymond McKendall, Lawrence Pohlman, and S-tavros A. Zenios. A stochastic programming model for money management. *European Journal of Operational Research*, 85(2):282 – 296, 1995.
45. Salvatore Greco, Benedetto Matarazzo, and Roman Słowiński. Beyond markowitz with multiple criteria decision aiding. *Journal of Business Economics*, 83(1):29–60, 2013.
46. Bruce D Grundy and J Spencer Martin Martin. Understanding the nature of the risks and the source of the rewards to momentum investing. *Review of Financial studies*, 14(1):29–78, 2001.
47. Pankaj Gupta, Masahiro Inuiguchi, Mukesh Kumar Mehlawat, and Garima Mittal. Multiobjective credibilistic portfolio selection model with fuzzy chance-constraints. *Information Sciences*, 229:1 – 17, 2013.
48. Pankaj Gupta, Mukesh Kumar Mehlawat, and Anand Saxena. A hybrid approach to asset allocation with simultaneous consideration of suitability and optimality. *Information Sciences*, 180(11):2264 – 2285, 2010.
49. Fang He and Rong Qu. A two-stage stochastic mixed-integer program modelling and hybrid solution approach to portfolio selection problems. *Information Sciences*, 289:190 – 205, 2014.
50. Julia L. Higle and Stein W. Wallace. Sensitivity analysis and uncertainty in linear programming. *Interfaces*, 33(4):53–60, 2003.
51. Kjetil Høyland, Michal Kaut, and Stein W. Wallace. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24(2–3):169–185, 2003.
52. Kjetil Høyland and Stein W. Wallace. Generating scenario trees for multistage decision problems. *Manage. Sci.*, 47(2):295–307, February 2001.
53. Chi-fu. Huang and Robert H. Litzenberger. *Foundations for financial economics / Chi-fu Huang, Robert H. Litzenberger*. North-Holland New York, 1988.
54. Narasimhan Jegadeesh and Sheridan Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of finance*, 48(1):65–91, 1993.
55. Philippe Jorion. *Value at Risk: The New Benchmark for Managing Financial Risk*. McGraw-Hill Education, 3rd edition edition, 2006.
56. Leonidas Sandoval Junior and Italo De Paula Franca. Correlation of financial markets in times of crisis. *Physica A: Statistical Mechanics and its Applications*, 391(1–2):187 – 208, 2012.
57. P. Kall and S.W. Wallace. *Stochastic Programming*. Wiley-Interscience series in Systems and Optimization. Wiley, 1994.
58. Michal Kaut and Stein W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257–271, 2007.

59. Michal Kaut and Stein W. Wallace. Shape-based scenario generation using copulas. *Computational Management Science*, 8(1–2):181–199, 2011.
60. Michal Kaut, Stein W. Wallace, Hercules Vladimirou, and Stavros Zenios. Stability analysis of portfolio management with conditional value-at-risk. *Quantitative Finance*, 7(4):397–409, 2007.
61. Phillip Kearns and Adrian Pagan. Estimating the density tail index for financial time series. *Review of Economics and Statistics*, 79(2):171–175, 1997.
62. Hans Kellerer, Renata Mansini, and M. Grazia Speranza. Selecting portfolios with fixed costs and minimum transaction lots. *Annals of Operations Research*, 99(1):287–304, 2000.
63. Alan J. King and Stein W. Wallace. *Modeling with Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2012.
64. Mervyn A King and Sushil Wadhwani. Transmission of volatility between stock markets. *Review of Financial studies*, 3(1):5–33, 1990.
65. Kees G Koedijk and Clemens JM Kool. Tail estimates of east european exchange rates. *Journal of Business & Economic Statistics*, 10(1):83–96, 1992.
66. Hiroshi Konno and Annista Wijayanayake. Portfolio optimization problem under concave transaction costs and minimal transaction unit constraints. *Mathematical Programming*, 89(2):233–250, 2001.
67. Pavlo Krokhmal, Jonas Palmquist, and Stanislav Uryasev. Portfolio optimization with conditional value-at-risk objective and constraints. *Journal of Risk*, 4:11–27, 2002.
68. Markku Lanne, Mika Meitz, and Pentti Saikkonen. Identification and estimation of non-gaussian structural vector autoregressions. *Journal of Econometrics*, 196(2):288 – 304, 2017.
69. Hayne Leland. Optimal portfolio implementation with transactions costs and capital gains taxes. *Haas School of Business Technical Report*, 2000.
70. Jun Li and Jiuping Xu. Multi-objective portfolio selection model with fuzzy random returns and a compromise approach-based genetic algorithm. *Information Sciences*, 220(0):507 – 521, 2013. Online Fuzzy Machine Learning and Data Mining.
71. Bob Litterman et al. *Modern investment management: an equilibrium approach*, volume 246. John Wiley & Sons, 2003.
72. Baoding Liu. Some research problems in uncertainty theory. *Journal of Uncertain Systems*, 3(1):3–10, 2009.
73. Hong Liu. Optimal consumption and investment with transaction costs and multiple risky assets. *The Journal of Finance*, 59(1):289–338, 2004.
74. MiguelSousa Lobo, Maryam Fazel, and Stephen Boyd. Portfolio optimization with linear and fixed transaction costs. *Annals of Operations Research*, 152(1):341–365, 2007.
75. François M Longin. From value at risk to stress testing: The extreme value approach. *Journal of Banking & Finance*, 24(7):1097 – 1130, 2000.
76. Khin Lwin and Rong Qu. A hybrid algorithm for constrained portfolio selection problems. *Applied Intelligence*, 39(2):251–266, September 2013.
77. Khin T Lwin, Rong Qu, and Bart L MacCarthy. Mean-var portfolio optimization: A nonparametric approach. *European Journal of Operational Research*, 260(2):751–766, 2017.
78. Harry M. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, March 1952.
79. Harry M. Markowitz. *Portfolio Selection: Efficient Diversification of Investments*. Wiley, 2nd edition, March 1991.
80. Helmut Mausser and Dan Rosen. Beyond var: from measuring risk to managing risk. In *Proceedings of the IEEE/IAFE 1999 Conference on Computational Intelligence for Financial Engineering (CIFEr 1999)*, pages 163–178, 1999.
81. K. Metaxiotis and K. Liagkouras. Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review. *Expert Systems with Applications*, 39(14):11685 – 11698, 2012.
82. Rafael Moral-Escudero, Rubén Ruiz-Torrubiano, and Alberto Suárez. Selection of optimal investment portfolios with cardinality constraints. In *Proceedings of the 2006 Congress on Evolutionary Computation (CEC2006)*, pages 2382–2388, 2006.

83. Tobias J Moskowitz and Mark Grinblatt. Do industries explain momentum? *The Journal of Finance*, 54(4):1249–1290, 1999.
84. Tobias J. Moskowitz, Yao Hua Ooi, and Lasse Heje Pedersen. Time series momentum. *Journal of Financial Economics*, 104(2):228 – 250, 2012. Special Issue on Investor Sentiment.
85. John M. Mulvey, Daniel P. Rosenbaum, and Bala Shetty. Parameter estimation in stochastic scenario generation systems. *European Journal of Operational Research*, 118(3):563 – 577, 1999.
86. John M. Mulvey and Hercules Vladimirou. Stochastic network programming for financial planning problems. *Manage. Sci.*, 38(11):1642–1664, November 1992.
87. John M. Mulvey and William T. Ziemba. *Handbooks in Operations Research and Management Science*, volume Volume 9, chapter Chapter 15 Asset and liability allocation in a global environment, pages 435–463. Elsevier, 1995.
88. Georg Ch. Pflug. Some remarks on the value-at-risk and the conditional value-at-risk. In *Probabilistic Constrained Optimization*, volume 49, pages 272–281. Springer US, 2000.
89. Matthew Pritsker. Evaluating value at risk methodologies: accuracy versus computational time. *Journal of Financial Services Research*, 12(2):201–242, October 1997.
90. Anna Grazia Quaranta and Alberto Zaffaroni. Robust optimization of conditional value at risk and portfolio selection. *Journal of Banking & Finance*, 32(10):2046 – 2056, 2008.
91. Wayne H. Wagner Robert D. Arnott. The measurement and control of trading costs. *Financial Analysts Journal*, 46(6):73–80, 1990.
92. R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.
93. R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.
94. Maciej Rysz, Alexander Vinel, Pavlo Krokhmal, and Pasiliao Eduardo. A scenario decomposition algorithm for stochastic programming problems with a class of downside risk measures. *INFORMS Journal on Computing*, 27(2):416 – 430, 2015.
95. Jonathan L. Shapiro. The sensitivity of PBIL to its learning rate, and how detailed balance can remove it. In *Proceedings of the Seventh Workshop on Foundations of Genetic Algorithms, Torremolinos, Spain, September 2-4, 2002*, pages 115–132, 2002.
96. Stephen J. Stoyan and Roy H. Kwon. A two-stage stochastic mixed-integer programming approach to the index tracking problem. *Optimization and Engineering*, 11(2):247–275, 2010.
97. Stephen J. Stoyan and Roy H. Kwon. A stochastic-goal mixed-integer programming approach for integrated stock and bond portfolio optimization. *Computers & Industrial Engineering*, 61(4):1285 – 1295, 2011.
98. Giorgio Szego. Measures of risk. *Journal of Banking & Finance*, 26(7):1253–1272, 2002.
99. Nikolas Topaloglou, Hercules Vladimirou, and Stavros A. Zenios. {CVaR} models with selective hedging for international asset allocation. *Journal of Banking & Finance*, 26(7):1535 – 1561, 2002.
100. Nikolas Topaloglou, Hercules Vladimirou, and Stavros A. Zenios. A dynamic stochastic programming model for international portfolio management. *European Journal of Operational Research*, 185(3):1501–1524, 2008.
101. S. Uryasev. Introduction to the theory of probabilistic functions and percentiles (value-at-risk). In StanislavP. Uryasev, editor, *Probabilistic Constrained Optimization*, volume 49 of *Nonconvex Optimization and Its Applications*, pages 1–25. Springer US, 2000.
102. Christiana Vassiadou-Zeniou and Stavros A. Zenios. Robust optimization models for managing callable bond portfolios. *European Journal of Operational Research*, 91(2):264 – 273, 1996.
103. M. Woodside-Oriakhi, C. Lucas, and J. E. Beasley. Heuristic algorithms for the cardinality constrained efficient frontier. *European Journal of Operational Research*, 213(3):538–550, September 2011.

104. M. Woodside-Oriakhi, C. Lucas, and J.E. Beasley. Portfolio rebalancing with an investment horizon and transaction costs. *Omega*, 41(2):406 – 420, 2013. Management science and environmental issues.
105. Panos Xidonas, George Mavrotas, Christis Hassapis, and Constantin Zopounidis. Robust multiobjective portfolio optimization: A minimax regret approach. *European Journal of Operational Research*, 262(1):299–305, 2017.
106. Yasuhiro Yamai, Toshinao Yoshiba, et al. Comparative analyses of expected shortfall and value-at-risk: their estimation error, decomposition, and optimization. *Monetary and economic studies*, 20(1):87–121, 2002.
107. Hitoshi Yano. Fuzzy decision making for fuzzy random multiobjective linear programming problems with variance covariance matrices. *Information Sciences*, 272(0):111 – 125, 2014.
108. Li-Yong Yu, Xiao-Dong Ji, and Shou-Yang Wang. Stochastic programming models in financial optimization: A survey. *Advanced Modeling and Optimization*, 5(1), 2003.
109. Liyong Yu, Shouyang Wang, Yue Wu, and K.K. Lai. A dynamic stochastic programming model for bond portfolio management. In Marian Bubak, GeertDick van Albada, Peter M.A. Sloot, and Jack Dongarra, editors, *Computational Science - ICCS 2004*, volume 3039 of *Lecture Notes in Computer Science*, pages 876–883. Springer Berlin Heidelberg, 2004.