Optimizing benchmark-based utility functions

DAVID P. MORTON
Graduate Program in Operations Research and Industrial Engineering, The University of Texas at Austin, Austin, TX 78712 USA.
e-mail: morton@mail.utexas.edu

ELMIRA POPOVA
Graduate Program in Operations Research and Industrial Engineering, The University of Texas at Austin, Austin, TX 78712 USA.
e-mail: elmira@mail.utexas.edu

IVILINA POPOVA
Ceiba Capital, LP, 106 E 6th Street, Suite 650, Austin, TX 78701 USA
e-mail: ipopova@ceibacapital.com

MING ZHONG
Robert H. Smith School of Business, University of Maryland, College Park, MD 20742 USA
e-mail: mzhong@wam.umd.edu

Abstract. We consider four utility functions, each of which incorporates a benchmark to better capture the motivations of today’s portfolio managers. Assuming instrument returns are normally distributed, we establish conditions under which optimal portfolios for these utilities are mean-variance efficient and we briefly discuss computing solutions of the models via standard nonlinear programming tools. When returns are not normally distributed, we cannot, in general, solve the optimal allocation problems exactly. Instead we use an approximation procedure rooted in Monte Carlo simulation. Our approach requires mixed-integer programming, and we describe computational enhancements that significantly improve our ability to solve these models.

Keywords: Portfolio allocation, mean-variance efficiency, stochastic programming, Monte Carlo simulation.

JEL classification: C15, C44
AMS classification: 90C11, 90C15

*This research was sponsored by Deutsche Asset Management, Deutsche Bank, New York. Much of the work was done while Ivilina Popova was a Director, and David Morton and Elmira Popova were consultants, at the Global Research Center of Deutsche Asset Management, New York. David Morton’s work was also partially supported by the National Science Foundation under Grants DMI-9702217 and DMI-0217927.
1. Introduction

In the early 1950s, Markowitz [16] introduced the mean-variance approach as a rational tool to help guide portfolio allocation decisions. Since then his method has become the standard among portfolio managers. However, many researchers have argued that variance is not an appropriate risk measure because it weighs equally deviations on the positive and the negative side of the mean. (See, for example, the discussion in [12].) Today’s portfolio managers are usually evaluated by comparing their performance to that of their peers or to a benchmark published in guidelines made available to investors. This type of evaluation process motivates a manager to be in the top among his peers or to outperform the benchmark. In this paper, we focus on the latter case. We consider four specific utility functions, each of which incorporates a benchmark in order to better capture the manager’s motivation. Assuming returns are normally distributed, we establish conditions under which optimal portfolios for these utility functions are mean-variance efficient. We obtain analytical expressions that, under appropriate conditions, allow maximization of expected utility to be carried out via standard nonlinear programming tools. When returns are not normally distributed, we cannot, in general, solve the optimal allocation problems exactly. Instead we use an approximation procedure rooted in Monte Carlo simulation, and we quantify the quality of the associated allocation decisions. Our approach requires solution of a difficult stochastic mixed-integer program, and we describe computational enhancements that significantly improve our ability to solve these models. We note that there has been significant work on using stochastic programming in modeling and solving related optimization problems arising in finance. See, for example, Dupaˇcová, Hurt and Štˇep´an [8] and Ziemba and Mulvey [28] and references therein.

2. Problem Statement

Assume that a portfolio manager can invest in \( m \) different instruments, e.g., \( m \) asset classes or \( m \) funds, with random returns, \( \tilde{\xi}_1, \ldots, \tilde{\xi}_m \). The manager must decide the proportion of the portfolio to allocate to each instrument, and we denote this decision \( x = (x_1, \ldots, x_m) \). These weights are constrained via \( x \in X = \{ x : \sum_{i=1}^{m} x_i = 1, x \geq 0 \} \). The addition of constraints, such as simple bounds of the form \( l \leq x \leq u \), is easily handled within our approach. Let \( \tilde{R}_x = \sum_{i=1}^{m} \tilde{\xi}_i x_i \) be the random return of the manager’s portfolio. The manager’s goal is to find an allocation \( x^* \), which solves

\[
z^* = \max_{x \in X} Eu(\tilde{R}_x),
\]

where \( u \) is a real-valued utility function that represents the manager’s objectives. We use the term “utility” in a liberal sense, e.g., we do not require that \( u \) satisfy the axioms of Von Neumann and Morgenstern [27].

We assume \( Eu(\tilde{R}_x) \) is upper semi-continuous on \( X \), and since \( X \) is compact this ensures that \( (P) \) has a finite optimal solution, \( x^* \), which is achieved on \( X \). Our Monte Carlo approximations will require use of the standard central limit theorem and so we assume \( Eu^2(\tilde{R}_x) < \infty \, \forall x \in X \). We use \( \xi \) and \( R_x \) to denote random elements and \( \xi \) and \( R_x \) to denote their realizations.
Clearly, the choice of the utility function \( u \) is paramount. In this paper, we consider the following four utility functions whose definitions depend on a benchmark return \( r_o \), which may be random:

\[
\begin{align*}
    u_1(R_x) &= I(R_x \geq r_o), \quad (1) \\
    u_2(R_x) &= -(r_o - R_x)^+, \quad (2) \\
    u_3(R_x) &= -(r_o - R_x)^+ + \lambda I(R_x \geq r_o), \quad \lambda \geq 0, \quad \text{and} \quad (3) \\
    Eu_4(\tilde{R}_x) &= -E\left[ r_o - \tilde{R}_x \mid \tilde{R}_x < r_o \right]. \quad (4)
\end{align*}
\]

Here, \( I(\cdot) \) takes value one if its argument is true and zero otherwise and \((\cdot)^+ = \max\{\cdot, 0\}\).

Maximizing expected utility under definition (1) corresponds to maximizing the probability that the benchmark is achieved. This objective is sometimes dismissed because it ignores the magnitude by which the target is missed or exceeded. However, when a portfolio manager’s performance is reviewed solely on whether the benchmark is achieved, it arguably provides a reasonable model of the manager’s objective. Utility (2) leads to minimizing the expected shortfall, relative to benchmark \( r_o \). In this case, magnitude matters when there is a shortfall but not when \( R_x \geq r_o \). Utility (3) is a composite of (1) and (2) and is parameterized by \( \lambda \geq 0 \). Utility (4) leads to minimizing the expected shortfall, conditional on there being a shortfall and can be rewritten

\[
Eu_4(\tilde{R}_x) = -\frac{E(r_o - \tilde{R}_x)^+}{1 - P(\tilde{R}_x \geq r_o)}.
\]

As a result, all four of our expected utility functions can be expressed in terms of \( E(r_o - \tilde{R}_x)^+ \) and \( P(\tilde{R}_x \geq r_o) \).

Utilities \( u_1 \) and \( u_4 \) are related to risk measures known as value-at-risk (VaR) and conditional value-at-risk (CVaR), respectively. VaR and CVaR are defined with respect to loss \(( -\tilde{R}_x )\) and require exogenous specification of a probability level, \( \beta \). Minimizing VaR corresponds to selecting an \( x \) that minimizes the \( \beta \)-level quantile associated with loss, \(-\tilde{R}_x\), and minimizing CVaR minimizes the associated conditional expected value of loss. In our setting, where portfolio managers are judged by comparing their performance with that of a benchmark, it is more natural to specify the benchmark than the probability level \( \beta \), but the two approaches are clearly related. See [21] for further discussion and references.

3. Mean-Variance Efficiency

The mean-variance efficient frontier [16] can be defined as

\[
\mathcal{F} = \left\{ (\mu, \sigma^2) : \exists x \in X \text{ with } \mu_x = \mu, \sigma_x^2 = \sigma^2 \text{ and } \exists \hat{x} \in X \text{ with } \mu_{\hat{x}} \geq \mu, \sigma_{\hat{x}}^2 < \sigma^2 \right\},
\]
where \( \mu_x = \sum_{i=1}^{m} E\tilde{\xi}_i x_i \) and \( \sigma_x^2 = \sum_{i=1}^{m} \sum_{j=1}^{m} \text{cov}(\tilde{\xi}_i, \tilde{\xi}_j) x_i x_j \). A feasible allocation decision, \( x \in X \), is said to be on the efficient frontier if \( (\mu_x, \sigma_x^2) \in \mathcal{F} \). The following lemma (see also [14]) provides our main tool for establishing mean-variance efficiency of optimal solutions of \((P)\) with utilities (1)-(4) under conditions explored below.

**Lemma 1.** Assume \( \text{Eu}_1(\tilde{R}_x) \) can be expressed as \( \text{Eu}(\tilde{R}_x) = F(\mu_x, \sigma_x^2) \) for some function \( F \), and let \( x^* \) solve \((P)\). If \( F(\mu_x, \sigma_x^2) \) is increasing in \( \mu_x \) and decreasing in \( \sigma_x^2 \) then \( x^* \) is on the efficient frontier.

**Proof:** By optimality of \( x^* \),

\[
F(\mu_{x^*}, \sigma_{x^*}^2) \geq F(\mu_x, \sigma_x^2) \forall x \in X. \tag{5}
\]

Suppose \( (\mu_{x^*}, \sigma_{x^*}^2) \notin \mathcal{F} \). Then there exists \( \hat{x} \in X \) with \( \mu_{\hat{x}} \geq \mu_{x^*} \) and \( \sigma_{\hat{x}}^2 < \sigma_{x^*}^2 \). This, coupled with the strict monotonicity of \( F \) contradicts \( (5) \). \( \square \)

The definition of mean-variance efficiency and Lemma 1 make no specific distributional assumptions on \( \tilde{\xi} \) and \( \tilde{R}_x \) beyond existence of the first two moments. That said, because of its properties, the normal distribution holds a special place in the theory of mean-variance efficiency. In the remainder of this section we assume \( \tilde{\xi} \) has a multivariate normal distribution. We denote the distribution function of a standard normal by \( \Phi \), and define \( m_x = \gamma_o - \mu_x \). Without loss of generality, we will proceed with \( \gamma_o \) being deterministic. If instead the benchmark is random then we simply redefine the return vector via \( \tilde{\xi} \leftarrow \tilde{\xi} - \gamma_o \) and use a deterministic benchmark of 0. In this case, the normal hypothesis and the definitions of \( \mu_x \) and \( \sigma_x^2 \) would all correspond to this translated excess return vector.

As indicate above, all four expected utility functions depend on \( P(\tilde{R}_x \geq \gamma_o) \) and \( E(\gamma_o - \tilde{R}_x)^+ \). Under our multivariate normal hypothesis

\[
P(\tilde{R}_x \geq \gamma_o) = 1 - \Phi \left( \frac{m_x}{\sigma_x} \right) \quad \text{and} \tag{6}
\]

\[
E(\gamma_o - \tilde{R}_x)^+ = m_x \Phi \left( \frac{m_x}{\sigma_x} \right) + \frac{\sigma_x}{\sqrt{2\pi}} \exp \left( \frac{-m_x^2}{2\sigma_x^2} \right). \tag{7}
\]

These expressions establish the existence of an \( F(\mu_x, \sigma_x^2) \) function for each expected utility. So, forming \( F(\mu_x, \sigma_x^2) \) for each utility allows us to explore conditions under which the monotonicity hypotheses for \( F \) in Lemma 1 are satisfied. We summarize our results in the following theorem and skip the details of the proof as it is straightforward to carry out given the above development.

**Theorem 2.** Let \( \tilde{\xi} \) be multivariate normal.

(i) If there exists \( \hat{x} \in X \) with \( \mu_{\hat{x}} > \gamma_o \) then \( x^* \in \text{argmax}_{x \in X} \text{Eu}_1(\tilde{R}_x) \) is on the efficient frontier.

(ii) \( x^* \in \text{argmax}_{x \in X} \text{Eu}_2(\tilde{R}_x) \) is on the efficient frontier.

(iii) If \( \lambda \geq 0 \) and there exists \( \hat{x} \in X \) with \( \mu_{\hat{x}} > \gamma_o \) then \( x^* \in \text{argmax}_{x \in X} \text{Eu}_3(\tilde{R}_x) \) is on the efficient frontier.

(iv) \( x^* \in \text{argmax}_{x \in X} \text{Eu}_4(\tilde{R}_x) \) is on the efficient frontier.
Theorem 2 states that the hypothesis of normal returns yields a mean-variance efficient solution when minimizing expected shortfall (i.e., when solving $(P)$ with utility $u_2$) and when minimizing conditional expected shortfall ($u_4$). When maximizing the probability of achieving the benchmark ($u_1$) or when using the composite probability-shortfall utility ($u_3$) mean-variance efficiency is obtained if we also assume there exists $\hat{x} \in X$ with $\mu_{\hat{x}} > r_o$. This additional hypothesis is equivalent to saying we have at least a 50% chance of achieving the benchmark and is closely related to ideas going back, e.g., to Kataoka [11] and to ideas that repeatedly surface in the chance-constraints literature [20]. The additional hypothesis is easy to check by simply testing whether $\max_{x \in X} \mu_x > r_o$, which amounts to testing whether $\mu_i > r_o$ for at least one $i = 1, \ldots, m$, when $X$ is a simplex. The intuition for needing this additional hypothesis is clear: If we can invest in two instruments (i.e., $m = 2$) that have identical mean returns of $\mu_1 = \mu_2 < r_o$ then to maximize the probability of achieving the benchmark we prefer the instrument with larger variability. This violates the spirit of mean-variance efficiency.

The following theorem is due to Levy and Markowitz [14] and establishes mean-variance efficiency for a wide class of utility functions.

**Theorem 3.** If $\tilde{\xi}$ is multivariate normal and $u$ is increasing, concave and twice differentiable then $x^* \in \arg\max_{x \in X} E u(\tilde{R}_x)$ is on the efficient frontier.

Our benchmark-based utility functions (1)-(4) violate the hypothesis of Theorem 3. However, we still achieve the mean-variance efficiency result in Theorem 2, under some conditions.

We conclude this section by examining the conditional expected shortfall utility in more detail. Consider the fractional programming problem

$$\min_{x \in X} \frac{f(x)}{g(x)},$$

(8)

where $g(x) > 0$ for all $x \in X$, and let

$$Q(\lambda) = \min_{x \in X} f(x) + \lambda g(x).$$

(9)

Let $\lambda^*$ satisfy $Q(\lambda^*) = 0$. Then, the solution $x^*(\lambda^*)$ of (9) with $\lambda = \lambda^*$ solves (8) (see, e.g., [2, p.209]).

So, $\max_{x \in X} E u_4(\tilde{R}_x)$ is solved by $x^*(\lambda^*)$, where $\lambda^*$ satisfies

$$0 = Q(\lambda^*) = \min_{x \in X} E(r_o - \tilde{R}_x)^+ + \lambda^* P(\tilde{R}_x < r_o).$$

(10)

As a result, model $(P)$ with objective function $E u_4(\tilde{R}_x)$ may be viewed as a special case of optimizing the composite objective $E u_3(\tilde{R}_x)$ for the appropriate choice of $\lambda$. However, $\lambda = \lambda^*$ satisfying (10) has $\lambda^* < 0$, and this choice seems unnatural from the perspective of the composite utility. In particular, the resulting utility appears to be a weighted sum of penalizing expected shortfall and encouraging the probability of shortfall. On the other hand, as Theorem 2 indicates, the conditional expected shortfall utility yields mean-variance efficient solutions. These results suggest that the “$\lambda \geq 0$” hypothesis in part (iii) of Theorem 2 could be weakened so that efficient-frontier solutions are obtained for negative values of $\lambda$ exceeding a threshold. (This is borne out in the analysis in assessing conditions under which $E u_3(\tilde{R}_x)$, expressed as $F(\mu_x, \sigma_x^2)$, satisfies the monotonicity conditions of Lemma 1.)
4. Solution Procedures and Solution Quality

This section concerns computationally solving model \((P)\). We begin in Section 4.1., with a brief discussion of the case of normally-distributed returns. Then, we turn to our main focus in Section 4.2., where returns are non-normal. In Section 4.2., we limit our discussion to the problem of maximizing the probability of achieving the benchmark. The primary reason for this is that all four expected utility functions involve \(E(r_o - \tilde{R}_x)^+\) and \(P(\tilde{R}_x \geq r_o)\) and the primary computational challenge lies in handling the latter term.

4.1. Normal Returns

After assuming a specific form for \(u\) in model \((P)\) and assuming that \(\tilde{\xi}\) is multivariate normal, one can: appeal to Theorem 3 if \(u\) is twice differentiable, concave and increasing or appeal to Theorem 2 if \(u\) is of the form of (1)-(4) or develop an analogous result. Let us assume that mean-variance efficiency of a solution to \((P)\) has been established. Then, at least in the case of utilities (1)-(4), we have analytical formulae indicating that the objective functions are amply smooth, and so we could attempt to solve \((P)\) via commercially-available nonlinear programming software (e.g., CONOPT [6], GRG2 [13], and MINOS [19]). For the purposes of employing such software, evaluating \(\Phi\), and its derivatives, may be regarded as analytical function evaluations. Such implementations of nonlinear programming algorithms are designed to find solutions that are locally optimal. If \(Eu(\tilde{R}_x)\) is concave then a local optimal solution is a global optimal solution. In such situations, a viable route for obtaining an optimal allocation decision appears available. Of course, further analysis may lead to more efficient approaches. For example, \(\max_{x \in X} Eu_1(\tilde{R}_x)\) simplifies to \(\min_{x \in X} m_x/\sigma_x\) and, again, ideas from fractional programming can be used, but it is not our purpose to explore these possibilities here.

Instead, we consider the case in which \(Eu(\tilde{R}_x)\) is not concave, or concavity has not been established. In general, finding a global maximizer of a nonconcave function is difficult. However, assume mean-variance efficiency has been established, or that we wish to constrain the solution to be mean-variance efficient, so that we know the solution is in the set

\[
\left\{ x(\lambda') : x(\lambda') \in \arg\max_{x \in X} \mu_x - \lambda' \sigma_x^2, \lambda' \geq 0 \right\}.
\]

Procedures for efficiently generating (11) via parametric quadratic programming begin with Markowitz [17]. This reduces the \(m\)-dimensional optimal allocation problem to a line search, \(\max_{\lambda' \geq 0} Eu(\tilde{R}_x(\lambda'))\). From a practical point of view, the task of finding a global optimal solution is now greatly simplified.

4.2. Non-Normal Returns

This section considers the case in which \(\tilde{\xi}\) does not have a normal distribution. For notational simplicity we redefine the return vector via \(\tilde{\xi}_i \leftarrow \tilde{\xi}_i - r_o, i = 1, \ldots, m\), as described earlier, and then, because \(X\) includes \(\sum_{i=1}^m x_i = 1\), \(\max_{x \in X} Eu_1(\tilde{R}_x)\) may
be written
\[ z^* = \max_{x \in X} \left[ P \left( \tilde{\xi} x \geq 0 \right) = E I \left( \tilde{\xi} x \geq 0 \right) \right], \] (12)
where \( \tilde{\xi} x \) denotes the inner product \( \sum_{i=1}^{n} \tilde{\xi}_i x_i \). Let \( x^* \) denote an optimal solution to (12).

We cannot, in general, compute \( x^* \) under non-normal returns and so we resort to an approximation based on Monte Carlo simulation. In particular, let \( \tilde{\xi}^1, \ldots, \tilde{\xi}^n \) be independent and identically distributed (i.i.d.) as \( \tilde{\xi} \). Then, we can form
\[ z^*_n = \max_{x \in X} \frac{1}{n} \sum_{i=1}^{n} I \left( \tilde{\xi}^i x \geq 0 \right). \] (13)
Denote an optimal solution to (13) by \( x^*_n \). We can view \( x^*_n \) and \( z^*_n \) as estimators arising from the sample-mean problem (13). Of course, we would like to know whether these estimators have desirable large-sample size properties relative to their population counterparts \( x^* \) and \( z^* \) from (12). This is available via the theory of epi-convergence (even though \( u_1(R_x) \) is not continuous). The following is a special case of a theorem of Attouch and Wets [1][Theorem 4.3] useful in our setting.

**Theorem 4.** Let \( \tilde{\xi}^1, \ldots, \tilde{\xi}^n \) be i.i.d. as \( \tilde{\xi} \) and let \( -u_o(x, \xi) \) be lower semicontinuous in \( x \) for all \( \xi \). If there exists \( \beta \in \mathbb{R} \) with \( u_o(x, \tilde{\xi}) \leq \beta \), with probability one (wp1) then \( -n^{-1} \sum_{i=1}^{n} -u_o(x, \tilde{\xi}^i) \) epi-converges to \( -E u_o(x, \tilde{\xi}) \), wp1.

(The theorem can also be cast in terms of \( u_o(x, \xi) \) instead of \( -u_o(x, \xi) \) by using hypo-convergence instead of epi-convergence.) Our utility \( u_o(x, \xi) = u_1(R_x) \) satisfies the conditions of Theorem 4 with \( \beta = 1 \). The epi-convergence property, together with the fact that the constraint set \( X \) is compact, implies consistency of \( z^*_n \) and \( x^*_n \), i.e., \( z^*_n \) converges to \( z^* \), wp1, and all limit points of \( \{x^*_n\}_{n=1}^{\infty} \) solve (12), wp1. Existence of at least one limit point of \( \{x^*_n\}_{n=1}^{\infty} \) in \( X \) is ensured as \( X \) is compact.

While we know that as \( n \) grows large the allocation decision \( x^*_n \) solves (12), wp1, we can only solve (13) for finite \( n \) in practice. So, we seek to quantify the quality of a candidate solution \( \hat{x} \in X \), e.g., \( \hat{x} = x^*_n \). To do so, we follow the procedure of [15] to construct a confidence interval on the optimality gap
\[ P \left\{ z^* - P \left( \tilde{\xi} \hat{x} \geq 0 \right) \leq \bar{\epsilon} \right\} \approx 1 - \alpha, \] (14)
where \( \bar{\epsilon} \) is a random confidence interval width. If \( \alpha = 0.05 \) (say) and the realization of the confidence interval width is sufficiently small, we will regard \( \hat{x} \) as a high-quality solution.

We now briefly review the procedure for constructing (14) from [15, 18], adapted to our setting. Given \( \hat{x} \in X \), generate \( \tilde{\xi}^1, \ldots, \tilde{\xi}^{n_\ell} \) i.i.d. as \( \tilde{\xi} \). Define the point estimate \( \bar{L}(n_\ell) = n_\ell^{-1} \sum_{i=1}^{n_\ell} I \left( \tilde{\xi}^i \hat{x} \geq 0 \right) \). Then, \( E \bar{L}(n_\ell) = P(\tilde{\xi} \hat{x} \geq 0) \) and we can infer
\[ P \left\{ \bar{L}(n_\ell) - \bar{\epsilon}_\ell \leq P \left( \tilde{\xi} \hat{x} \geq 0 \right) \right\} \approx 1 - \frac{\alpha}{2}, \] (15)
where \( \bar{\epsilon}_\ell = t_{n_\ell-1,\alpha/2}s_\ell(n_\ell)/\sqrt{n_\ell} \), \( t_{n_\ell-1,\alpha/2} \) is the \( 1 - \alpha/2 \) quantile from a \( t \)-distribution with \( n_\ell - 1 \) degrees of freedom, and \( s_\ell^2(n_\ell) \) is the standard sample-variance estimator
of $I(\hat{\xi} x \geq 0)$. In order to assess how close $P(\hat{\xi} x \geq 0)$ is to $z^*$, we need a point estimate of $z^*$ or of an upper bound on $z^*$. We obtain the latter as follows

$$z^* = \max_{x \in X} E \frac{1}{n} \sum_{i=1}^{n} I(\hat{\xi}^i x \geq 0) \leq E \max_{x \in X} \frac{1}{n} \sum_{i=1}^{n} I(\hat{\xi}^i x \geq 0) = E z^*_n.$$  

In general, when the original problem (12) has multiple optimal solutions, $z^*_n$ is not asymptotically normal (e.g., [7]). So, we generate $n_u$ i.i.d. replications of $z^*_n$, $z^*_n;i$, $i = 1, \ldots, n_u$, define $\hat{U}(n_u) = n_u^{-1} \sum_{i=1}^{n_u} z^*_n;i$ and infer

$$P \left\{ z^* \leq \hat{U}(n_u) + \hat{\epsilon}_u \right\} \approx 1 - \frac{\alpha}{2},$$

(16)

where $\hat{\epsilon}_u = t_{n_u-1,\alpha/2} s_u(n_u)/\sqrt{n_u}$ and $s^2_u(n_u)$ is the standard sample-variance estimator of $z^*_n$.

Defining $\bar{\epsilon}$ from (14) as $\bar{\epsilon} = (\hat{U}(n_u) - \hat{L}(n_{\ell})) + \hat{\epsilon}_u + \bar{\epsilon}_{\ell}$ and combining (15) and (16) we have

$$P \left\{ z^* - P(\hat{\xi} x \geq 0) \leq \bar{\epsilon} \right\} \geq P \left\{ \hat{L}(n_{\ell}) - \bar{\epsilon}_{\ell} \leq P(\hat{\xi} x \geq 0), z^* \leq \hat{U}(n_u) + \hat{\epsilon}_u \right\} \geq 1 - P \left\{ \hat{L}(n_{\ell}) - \bar{\epsilon}_{\ell} > P(\hat{\xi} x \geq 0) \right\} - P \left\{ z^* > \hat{U}(n_u) + \hat{\epsilon}_u \right\} \approx 1 - \alpha,$$

where the second inequality follows from the Boole-Bonferroni inequality.

Computing $\hat{L}(n_{\ell})$ is very inexpensive because we need only generate $n_{\ell}$ observations of $\hat{\xi}$, compute $n_{\ell}$ inner products, and perform $n_{\ell}$ comparisons with 0. As a result, we can use a large sample size $n_{\ell}$ (in our computational results in Section 6., we use $n_{\ell} = 20,000$). The vast majority of the computational effort in employing the above procedure lies in forming $\hat{U}(n_u)$. To do so requires solving $n_u$ i.i.d. replications of (13). We compute $z^*_n$ from (13) by forming and solving the following mixed-integer program (MIP)

$$z^*_n = \max_{x,y} \frac{1}{n} \sum_{j=1}^{n} y_j$$

s.t. $\sum_{i=1}^{m} x_i = 1,$

$$\hat{\xi}^j x \geq -M_j(1 - y_j), j = 1, \ldots, n,$$

$y \in \{0,1\}^n,$

$x \geq 0.$

The binary decision variable $y_j$ takes value one if the benchmark is achieved under return scenario $\hat{\xi}^j$. The value of data parameter $M_j$ is chosen sufficiently large so that the associated constraint is vacuous if $y_j = 0$.

To summarize, the procedure works as follows. First we solve $n_u$ instances of (17), each with independent sets of $n$ i.i.d. return scenarios. In so doing, we obtain feasible allocation decisions, $x^*_n, \ldots, x^*_{n_u}$. We form the candidate solution
\[ \hat{x} = n_u^{-1} \sum_{i=1}^{n_u} x_n^i, \]
which is feasible by convexity of \( X \). (Other possibilities including using a screening method to select one of the decisions, e.g., [24], but the averaging method proved robust in our computational work in this setting.) Then, we generate \( n \) i.i.d. return scenarios, which are also independent of those used in forming \( \bar{U}(n_u) \) and \( \hat{x} \). Construction of the approximate \((1 - \alpha)\)-level confidence interval
\[
[0, (\bar{U}(n_u) - \bar{L}(n_u))^{+} + \bar{\epsilon}_u + \bar{\epsilon}_l]
\]
on the optimality gap, \( z^* - P(\xi \hat{x} \geq 0) \), is then straightforward.

5. Computational Enhancements

The solution procedure described in Section 4.2., for non-normal returns, requires solving multiple instances of a difficult MIP (17). In this section we outline five ideas to reduce computational effort.

**MIP stopping criterion:** The branch-and-bound (b&b) solution procedure for solving an MIP iteratively tightens linear-programming relaxations until the corresponding relaxation bound (an upper bound when maximizing) is within a user-prescribed tolerance of the objective function value of the incumbent solution. Due to the nature of its objective function, an optimal solution to (17) is obtained if the difference between the b&b procedure’s upper and lower bounds is strictly less than \( 1/n \). Furthermore, (17) should be solved with a stopping tolerance that is commensurate with the confidence interval width \( \bar{\epsilon} \) we anticipate obtaining. Note that lower bounds on (the mean of) \( \bar{\epsilon} \) are easily obtained given that we are dealing with expectations of a Bernoulli random variable. When (17) is not solved exactly, the linear-programming relaxation bounds must be used in place of \( z_n^i, i = 1, \ldots, n_u \).

**Small values of “big \( M \):** As just indicated, linear-programming relaxations play a key role in the MIP b&b procedure. We obtain tighter linear-programming relaxations by ensuring that \( M_j \) is no larger than need be. To accomplish this we allow \( M_j \) to depend on \( j \) and set \( M_j = -\min_{i=1,\ldots,m} \hat{\xi}_j^i, j = 1, \ldots, n \). If the convexity restrictions in \( X \) are appended, e.g., via tighter simpler bounds, \( l \leq x \leq u \), then we can tighten the values of \( M_j, j = 1, \ldots, n \). Dynamic updating of \( M_j, j = 1, \ldots, n \), within the b&b procedure would be possible if such simple bounds on \( x^* \) could be automatically inferred within the algorithm. (We do not pursue this issue here.)

**Variable fixing:** If \( \min_{i=1,\ldots,m} \hat{\xi}_i^j \geq 0 \) then even if we allocate all of the portfolio to the worst-performing instrument under scenario \( j \) we will achieve the benchmark under \( j \), i.e., we know a priori that \( y^j = 1 \). Similarly, \( \max_{i=1,\ldots,m} \hat{\xi}_i^j < 0 \) implies \( y^j = 0 \). Preprocessing to reduce the number of binary decision variables in this way can decrease the computational effort required to solve (17). In Section 6., we will describe an 10-instrument model with a benchmark return of 12%. In this case, when we use 500 scenarios to form (17), the typical number of binary variables fixed is in the range 40-65. Under tighter simple bounds of the form \( l \leq x \leq u \), the above rules for fixing the \( y_j \)s could be tightened.

**Dominating scenarios:** We say that scenario \( i \) dominates \( i' \) if \( \hat{\xi}_j^i \geq \hat{\xi}_j^{i'}, j = 1, \ldots, m \). In general, we will not know a priori whether the benchmark will be achieved under scenarios \( i \) and \( i' \), but we do know that if it is achieved under \( i' \) then it will be achieved under \( i \). And, if it is not achieved under \( i \), then it will not be achieved under \( i' \). So, if scenario \( i \) dominates \( i' \) we can append a logical constraint
of the form \( y_i \geq y_{i'} \) to (17). Even though the resulting MIP is larger in size, including such constraints can speed the b\&b solution process. For the 10-instrument 500-scenario model, the typical number of pairs of dominated scenarios ranges from 2300-3900. Under tighter simple bounds of the form \( l \leq x \leq u \), the above notion of dominating scenarios can be refined and more logical constraints could be generated.

**Use of target solutions:** Modifying the objective function of an optimization model by subtracting (for a maximization model) a term of the form \( \rho \| x - x^t \| \), \( \rho > 0 \), for some target solution \( x^t \), has a rich history in mathematical programming. From a modeling perspective of desiring to pose optimization models with a unique optimal solution, this idea goes back to Tikhonov [25, 26]. From an algorithmic perspective such proximal, or regularizing, terms are commonly employed in nonlinear programming [2] and in stochastic programming [10, 22, 23]. In these cases, \( x^t \) is updated as the algorithm proceeds. Our interest in using such target terms in the objective function is from both the modeling and computational perspectives.

Brown et al. [3] discuss a practical modeling motivation for incorporating such a term: Suppose that optimal decision \( x^* \) has been announced or published. Now, assume the underlying problem is perturbed and the model is reoptimized to obtain solution \( x^{**} \). The new solution, \( x^{**} \), can differ dramatically from \( x^* \). These differences can be disconcerting to the decision-maker and are often unnecessary. To circumvent this, two criteria are used. The primary criterion is the original objective function and the secondary criterion is that we prefer not to deviate from the published solution, \( x^t = x^* \). Interestingly, Brown et al. report decreased run times by a factor of (roughly) three for several real-world MIPs.

In our setting an investor holds a current portfolio, \( x^t \). Unless given compelling reasons to change the investor prefers to stay close to his current portfolio. Among other reasons, transaction costs, which can be difficult to model when nonlinear, would be incurred. This leads to the modified model

\[
z^* = \max_{x \in X} P(\xi x \geq 0) - \rho \| x - x^t \|, \tag{18}
\]

where \( \rho \geq 0 \) is a small penalty weight. Of course, we cannot solve (18) exactly for the same reasons we cannot solve it with \( \rho = 0 \), as discussed above, but we can modify the approximating model (17) to incorporate the additional objective function term. By using a one- or infinity-norm we can preserve the linear structure of (17). We use the one-norm in our computations.

Even if we don’t have an \( x^t \) from the investor, we may create \( x^t \) by first solving a simpler problem. For example, we could obtain \( x^t \) by: (i) solving \( \max_{x \in X} P(\xi x \geq 0) \) under the normal distribution assumption, (ii) solving a variant of (17) which minimizes expected shortfall from the benchmark, or (iii) solving (17) with a smaller sample size. Computing \( x^t \) in one of these ways is obviously not for modeling reasons but rather for the potential computational benefit, which we examine in the next section.

6. Computational Results

We consider a real-world instance of \((P)\) under utility \( u_1 \) with the following characteristics. The investment manager is to construct a portfolio in which the underly-
ing instruments consist of 10 funds. The benchmark is a deterministic 12% annual return. The distribution of $\tilde{\xi}$ is modeled, and simulated from, using the so-called NORTA (normal-to-anything) approach in which non-normal marginal distributions are used in combination with a specified covariance matrix [4, 9].

In terms of the solution and quality-testing procedure of Section 4.2., we solve $n_u = 30$ i.i.d. instances of (17). We perform tests in which these 30 instances have both $n = 300$ and $n = 500$ scenarios. As described above, we then compute the candidate solution via $\hat{x} = n_u^{-1} \sum_{i=1}^{n_u} x_i^t$, using the solutions to each of the instances of (17). Then, we use $n_\ell = 20,000$ to form the point estimate of $P(\tilde{\xi} \hat{x} \geq 0)$. The solution procedure is implemented in C code calling CPLEX 7.5 [5]. All computational tests reported here were run on a 1.6 GHz Dell Pentium IV computer with 512 Mb of memory.

We first solved 30 instances of (17) with $n = 300$ scenarios without any of the enhancements described in Section 5. Then, we re-ran these instances using scenario-dependent big $M$ values (rather than a uniformly conservative choice), variable fixing, and constraints induced by dominating scenarios. These three enhancements reduced running times by roughly 30%. We won’t explore the impact of these enhancements in greater detail as we view the results from using target-based solutions as much more interesting.

To investigate the value of using a target solution, $x^t$, we consider the following ways of defining the target:

- $x^t$ is the average solution obtained by solving three empirical distribution approximations of $\max_{x \in X} E u_\delta(\tilde{R}_x)$ (i.e., expected shortfall) with the 12% benchmark and with $n = 200$ scenarios.

- $x^t$ is initially the above expected shortfall solution when we solve the first instance of (17), but $x^t$ is then updated using a weighted sum of the solutions of (17) to-date.

- $x^t$ is the average of three solutions of (17) (i.e., probability of achieving benchmark) with $n = 200$.

- $x^t$ is initially the above solution from maximizing the probability of achieving the benchmark when we solve the first instance of (17), but $x^t$ is then updated using a weighted sum of the solutions of (17) to-date.

When updating the target solution via a weighted sum, we consider two methods. The first simply averages all solutions to-date and the second forms the current target solution through exponential smoothing with a weighting factor of 0.5.

Tables 1 and 2 report computational results for each of these methods for obtaining and updating $x^t$. In both tables, the columns of data correspond to solving (17), or rather its variant with a one-norm target term, with increasing values of $\rho$. “CU upper” is $\bar{U}(n_u) + \tilde{\epsilon}_u$ with $n = 300$ and $n_u = 30$. “CI lower” is $\bar{L}(n_\ell) - \tilde{\epsilon}_\ell$ with $n_\ell = 20,000$. The “CI upper” values are reported with respect to $P\left(\tilde{R}_x \geq r_o\right) - \rho\|x-x^t\|$ while the “CI lower” values are with respect to $P\left(\tilde{R}_x \geq r_o\right)$. The “Adjusted” rows are “CI upper” values, modified so that they are also with respect to $P\left(\tilde{R}_x \geq r_o\right)$. (We return to this issue below.) The reported run times
are elapsed computation time in seconds and include both the time to compute the target solution and carry out the solution and quality-testing procedure. The $\rho = 0$ column corresponds to the original model, i.e., without a target term. As $\rho$ grows we more strongly encourage proximity to the target solution, $x'$. As can be seen from Table 1, when $x'$ is the expected shortfall solution with no updating there is significant degradation of solution quality, even for small values of $\rho$. This trend in solution quality can be seen by examining “CI lower” values. Computation times decrease substantially only after solution quality has dropped significantly and solution times even increase for small values of $\rho$. When the target solution is updated by averaging, the trend with respect to solution quality is similar while computation times for small value of $\rho$ are slightly better. When the target is updated via exponential smoothing, so that the expected shortfall target solution drops away more quickly, solution quality degrades more slowly and computation times drop more quickly, but substantial decrease in computational effort is only obtained at the unacceptable price of decreased solution quality.

Table 2 was constructed in the same manner as Table 1, except that the initial target solution comes from averaging solutions to three instances of (17) with $n = 200$. When the target solution is not updated, solution quality degrades only slightly and when the solution is updated via averaging or exponential smoothing there is no discernible decrease in solution quality as $\rho$ grows. And, computation times decrease markedly as $\rho$ grows. Tables 1 and 2 are each reported using $n = 300$ scenarios. Based on this smaller sample size we conclude that the method of using an expected shortfall solution as $x'$ is not promising.

Based on the results from Tables 1 and 2 we report in Table 3 results when $n = 500$ (instead of $n = 300$). The initial target solution again comes from averaging solutions to the same three instances of (17) with $n = 200$ and we either perform no updating or update using exponential smoothing. The corresponding results are similar to those of Table 2, with minor degradation in solution quality without updating and little, if any, drop in solution quality when updating. Solution times drop from almost six hours with $\rho = 0$ to about three minutes for $\rho = 0.2$.

Let $x_{ES}(200)$ denote the expected shortfall solution and $x_P(200)$ denote the benchmark-probability solution used as targets above. Also, let $x_P(500)$ denote the averaged optimal solution to 30 instances of (17) with $n = 500$ scenarios (and $\rho = 0$). Using a common set of 20,000 scenarios, we obtain the following point estimates and 0.95-level confidence intervals: $P(\xi x_{ES}(200) \geq 0)$ as $0.511 \pm 0.003$; $P(\xi x_P(200) \geq 0)$ as $0.561 \pm 0.003$; and, $P(\xi x_P(500) \geq 0)$ as $0.564 \pm 0.003$. Paired-$t$ tests reject the null hypothesis that “$P(\xi x_{ES}(200) \geq 0) = P(\xi x_P(200) \geq 0)$” and also reject the null hypothesis that “$P(\xi x_P(200) \geq 0) = P(\xi x_P(500) \geq 0)$.”

We summarize and explain these computational results as follows. When the target solution is far from optimal, as is the case for the expected shortfall solution, there is little value in using the target-based approach in our setting. However, when $x'$ is a near-optimal solution, computation times decrease significantly with a minor drop in solution quality. In our tests, averaging the solutions to three 200-scenario approximating problems yields a target solution that is sufficiently near-optimal for the approach to work well. When the set of $\epsilon$-optimal allocation decisions to (17) contains $x$’s that achieve the benchmark with different sets of scenarios (of nearly the
same cardinality), the b&b solution algorithm must work harder due to symmetries in the b&b tree. The target-based solution method creates asymmetries that aid the b&b solution method. We note that the exponential smoothing weighting factor of 0.5 puts significant weight on the most recent observations. We chose the weight in this way to be aggressive with respect to dropping old solutions that may be of low quality and so that the averaging and exponential smoothing methods would differ significantly in this regard.

A potential criticism is that the target-based computational enhancement only works (or, only works well) when we already have a high-quality target solution. Such criticism would be misplaced. It is often the case that one has what is believed to be a high-quality solution but quantitative proof, or statistical evidence, to support this belief is unavailable. Our emphasis here is on establishing solution quality, and if solution quality is high then the target-based enhancement may help substantially.

We conclude this section with the following remark. One view of the target-based approach is that we have a modified optimization model, i.e., (12) has been modified to (18). So, from this perspective upper-bound estimators and lower-bound estimators and associated confidence intervals should be reported with respect to the objective function $P\left(\tilde{R}_x \geq r_o\right) - \rho \|x - x^t\|$. Another view is that the decision-maker is interested in the probability of achieving the benchmark. The target term in the objective is just a computational tool that enables us to carry out the quality-testing procedure more quickly, and arguably, we should not burden the decision-maker with such details. Also, note that there are difficulties with the former view when $x^t$ is not fixed, i.e., when it is updated. In Tables 1-3 we have reported the “CI upper” values with respect to $P\left(\tilde{R}_x \geq r_o\right) - \rho \|x - x^t\|$ but then adjusted them, by adding back the “$\rho \|x - x^t\|$” value, to $P\left(\tilde{R}_x \geq r_o\right)$. We have also reported the “CI lower” values with respect to $P\left(\tilde{R}_x \geq r_o\right)$, as we lean toward the decision-makers perspective. Strictly speaking, the “CI upper” and “CI lower” values are not comparable, but as the tables indicate the error is small for small $\rho$, at least when $x^t$ is of high quality.

7. Summary

In this paper we consider optimal portfolio allocation under some non-standard utility functions. These utility functions are defined based on a benchmark return and arise naturally due to the manner in which today’s portfolio managers are judged. In the case of normally-distributed returns, we establish connections to Markowitz’ classical portfolio theory by giving conditions under which these utilities lead to mean-variance efficient solutions. We also discuss computational issues associated with optimizing these and other utility functions, which yield mean-variance efficient solutions. When returns are non-normal, we propose an approximating technique that allows for constructing high-quality solutions with modest computational effort. We investigate the implications, in terms of both speed and quality, of penalizing deviations from target allocation decisions in the solution procedure. We illustrate our approach by solving a 10-instrument portfolio allocation problem in which the manager’s objective is to outperform a benchmark of 12% return per year.
References


Table 1: Initial target solution $x^t$ is average of three solutions minimizing expected shortfall using 200 scenarios. $x^t$ is then either not updated (top results) or updated as we solve the upper-bounding problems (17) for maximizing the probability of achieving the benchmark, each of which has $n = 300$ scenarios. The second set of results correspond to updating via averaging and the last set updating via exponential smoothing.

\begin{tabular}{lcccccccccc}
\hline
$n = 300$; $n_u = 30$; $n_\ell = 20,000$ & & & & & & & & & & \\
$\rho$ & 0.00 & 0.02 & 0.04 & 0.06 & 0.08 & 0.10 & 0.12 & 0.14 & 0.16 & 0.18 & 0.20 \\
\hline
\multicolumn{12}{c}{$x_t$: expected shortfall solution (200 scen.) with no updating} \\
CI lower & 0.564 & 0.559 & 0.544 & 0.529 & 0.519 & 0.516 & 0.513 & 0.512 & 0.512 & 0.512 & 0.512 \\
CI upper & 0.594 & 0.571 & 0.552 & 0.542 & 0.535 & 0.531 & 0.529 & 0.527 & 0.527 & 0.526 & 0.525 \\
Adjusted & 0.594 & 0.591 & 0.577 & 0.563 & 0.547 & 0.540 & 0.536 & 0.533 & 0.533 & 0.531 & 0.529 \\
run time & 1825 & 4365 & 3895 & 2583 & 2068 & 1104 & 717 & 441 & 307 & 220 & 165 \\
\hline
\multicolumn{12}{c}{$x_t$: expected shortfall solution (200 scen.) updated via averaging} \\
CI lower & 0.564 & 0.558 & 0.550 & 0.534 & 0.525 & 0.516 & 0.515 & 0.515 & 0.511 & 0.512 & 0.512 \\
CI upper & 0.594 & 0.583 & 0.568 & 0.552 & 0.542 & 0.533 & 0.531 & 0.530 & 0.526 & 0.527 & 0.526 \\
Adjusted & 0.594 & 0.588 & 0.579 & 0.564 & 0.554 & 0.540 & 0.538 & 0.537 & 0.531 & 0.532 & 0.531 \\
run time & 1825 & 1425 & 1584 & 1816 & 1585 & 1181 & 631 & 406 & 303 & 264 & 158 \\
\hline
\multicolumn{12}{c}{$x_t$: expected shortfall solution (200 scen.) updated via exponential smoothing} \\
CI lower & 0.564 & 0.562 & 0.557 & 0.549 & 0.548 & 0.525 & 0.522 & 0.519 & 0.514 & 0.515 & 0.515 \\
CI upper & 0.594 & 0.585 & 0.577 & 0.566 & 0.562 & 0.542 & 0.537 & 0.534 & 0.528 & 0.528 & 0.529 \\
Adjusted & 0.594 & 0.589 & 0.582 & 0.573 & 0.570 & 0.550 & 0.543 & 0.540 & 0.535 & 0.534 & 0.535 \\
run time & 1825 & 1317 & 1023 & 946 & 683 & 857 & 569 & 370 & 275 & 208 & 141 \\
\hline
\end{tabular}
Table 2: The initial target solution $x^t$ is obtained by averaging three solutions from maximizing the probability of achieving the benchmark using 200 scenarios. This target solution is then either not updated or updated via averaging or exponential smoothing as we solve the upper-bounding problems (17), each of which has $n = 300$ scenarios.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>0.00</th>
<th>0.02</th>
<th>0.04</th>
<th>0.06</th>
<th>0.08</th>
<th>0.10</th>
<th>0.12</th>
<th>0.14</th>
<th>0.16</th>
<th>0.18</th>
<th>0.20</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI lower</td>
<td>0.564</td>
<td>0.563</td>
<td>0.562</td>
<td>0.562</td>
<td>0.562</td>
<td>0.561</td>
<td>0.561</td>
<td>0.561</td>
<td>0.561</td>
<td>0.561</td>
<td>0.561</td>
</tr>
<tr>
<td>CI upper</td>
<td>0.594</td>
<td>0.587</td>
<td>0.583</td>
<td>0.581</td>
<td>0.580</td>
<td>0.578</td>
<td>0.577</td>
<td>0.575</td>
<td>0.574</td>
<td>0.574</td>
<td>0.574</td>
</tr>
<tr>
<td>Adjusted</td>
<td>0.594</td>
<td>0.590</td>
<td>0.586</td>
<td>0.585</td>
<td>0.584</td>
<td>0.582</td>
<td>0.582</td>
<td>0.580</td>
<td>0.578</td>
<td>0.578</td>
<td>0.578</td>
</tr>
<tr>
<td>run time</td>
<td>1848</td>
<td>1102</td>
<td>561</td>
<td>256</td>
<td>156</td>
<td>121</td>
<td>100</td>
<td>88</td>
<td>80</td>
<td>71</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: The initial target solution $x^t$ is obtained by averaging three solutions from maximizing the probability of achieving the benchmark using 200 scenarios. This target solution is then either not updated or updated via exponential smoothing as we solve the upper-bounding problems (17), each of which has $n = 500$ scenarios.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>0.00</th>
<th>0.02</th>
<th>0.04</th>
<th>0.06</th>
<th>0.08</th>
<th>0.10</th>
<th>0.12</th>
<th>0.14</th>
<th>0.16</th>
<th>0.18</th>
<th>0.20</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI lower</td>
<td>0.564</td>
<td>0.563</td>
<td>0.562</td>
<td>0.562</td>
<td>0.562</td>
<td>0.561</td>
<td>0.561</td>
<td>0.561</td>
<td>0.561</td>
<td>0.561</td>
<td>0.561</td>
</tr>
<tr>
<td>CI upper</td>
<td>0.594</td>
<td>0.590</td>
<td>0.586</td>
<td>0.584</td>
<td>0.583</td>
<td>0.583</td>
<td>0.583</td>
<td>0.583</td>
<td>0.582</td>
<td>0.582</td>
<td>0.582</td>
</tr>
<tr>
<td>Adjusted</td>
<td>0.594</td>
<td>0.590</td>
<td>0.587</td>
<td>0.585</td>
<td>0.585</td>
<td>0.584</td>
<td>0.583</td>
<td>0.583</td>
<td>0.583</td>
<td>0.582</td>
<td>0.581</td>
</tr>
<tr>
<td>run time</td>
<td>20652</td>
<td>4850</td>
<td>773</td>
<td>1256</td>
<td>1110</td>
<td>657</td>
<td>50</td>
<td>207</td>
<td>94</td>
<td>85</td>
<td>76</td>
</tr>
</tbody>
</table>

Table 4: The initial target solution $x^t$ is obtained by averaging three solutions from maximizing the probability of achieving the benchmark using 200 scenarios. This target solution is then either not updated or updated via exponential smoothing as we solve the upper-bounding problems (17), each of which has $n = 1000$ scenarios.