Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12753
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Simplifying Complexity: Scenario Reduction Techniques in Stochastic Programming

Christian Sinaga[1)*], Tulus[2)], Herman Mawengkang[3)]
[1)]Magister of Mathematics, Universitas Sumatera Utara, Indonesia
[2,3)]Department of Mathematics, Universitas Sumatera Utara, Indonesia
[1)]prisaisinaga@gmail.com, [2)]tulus@usu.ac.id, [3)]mawengkang@usu.ac.id

Submitted : Jul 23, 2023 | **Accepted** : Jul 24, 2023 | **Published** : Jul 25, 2023

Abstract: Stochastic programming problems arise as mathematical models for optimizing problems under stochastic uncertainty. Computational approaches for solving these models often involve approximating the underlying probability distribution with a probability measure that has finite support. To mitigate the computational complexity associated with increasing the number of scenarios, it may be necessary to reduce their quantity. The scenario ξ^{u^*} is selected as the first element of supp(Q), and the separable structure is used to determine the second element of supp(Q) while keeping the first element fixed. The process is repeated to establish the remaining indices, and each subsequent scenario is reduced accordingly. This iterative process continues until scenario $N - n$ is reduced.

Keywords: Scenario reduction, Two-stage, Stochastic programming

INTRODUCTION

(Dantzig, 1955) developed the stochastic programming model in 1955. Stochastic programming is a type of mathematical programming that has an objective function or uncertainty constraint that is represented by a probability distribution. When some or all parameters are uncertain, decision makers will choose to minimize risk. Even if uncertainty is precisely defined, it must be prepared in detail in practice with several scenarios as a possible result of the data, in the specification and accuracy of the combined probability distribution.

In general, approximation of the discrete probability distribution limits is applied to tractable problems. Any random event or combination of random events that is included in scenario creation, with each scenario having a different probability of occurring. If a problem's scenario expands into multiple scenarios over time, the boundaries become more complicated, necessitating the use of branched models. Because only one of the best scenarios will be chosen, the others must be eliminated. The stochastic programming problem must be specified in order to take each scenario into account. The specification in this study is limited to a two-step linear stochastic program.

Production planning (Fleten & Kristoffersen, 2008), scheduling (Birge & Dempstert, 1996), routing (Kenyon & Morton, 2003), allocation (Li et al., 2009), capacity expansion (Ahmed et al., 2003), energy investment (Hemmati, 2019), environmental control and management (Dupačová et al., 1991; Niknam et al., 2012), water management (Huang & Loucks, 2000), pertamina (Dempster et al., 2000), portfolio investment (Chen & Yang, 2017), inventory (Doğru et al., 2010), and other applications (Shapiro et al., 2021) are examples of stochastic program applications.

The best decision is determined by the quality of the scenario model, the depiction of the process's influence, the uncertainty of the objective/cost function parameters, and the constraints. Decisions in the formation of scenarios may become more complex as the number of scenarios increases, necessitating the use of techniques to reduce these scenarios. This is accomplished by lowering the number of decision lines. The goal is to find a scenario subset of known cardinals. Probability metrics are used to calculate

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12753 p-ISSN : 2541-044X

the set closest to the initial distribution. A heuristic algorithm is used in this study to select the scenario that will be reduced.

The stochastic program problem is generally stated as follows:

$$\min \left\{ E\big(f_0(x,\xi)\big) = \int_{R^S} f_0(x,\xi) P(d\xi) : x \in X \right\} \quad (1)$$

In the model, X represents a closed subset of R^m that maps the function f_0 from $R^m \times R^s$ to the real number $\bar{R} = R \cup \{-\infty, +\infty\}$ R. E denotes the expectation operator with respect to the probability distribution P, where P is defined on the space R^s. To handle these models, numerical approximations are commonly employed. One approach involves replacing the original probability distribution P, which is defined on a continuous space, with a set of $(\xi_1, \dots, \xi_N)$ scenarios denoted as ξ_i that occur with probability $p_i > 0$, $i = 1, \dots, N$ and $\sum_{i=1}^{n} p_i = 1$. This approximation allows for the simplification and computational tractability of the model.

$$\min \left\{ \sum_{i=1}^{N} p_i f_0(x, \dots, \xi_i) : x \in X \right\} \quad (2)$$

The choice of approximation method for a stochastic program depends on the properties of the integrand function $f_0(x, \dots, \xi_i)$ and the characteristics of the probability distribution P. However, evaluating f_0 on pairs of (x, ξ) can be computationally expensive. In numerical stochastic programming, this poses a challenge as the optimal approximation of P often requires a large number of scenarios N. However, as the search for a solution progresses, the goal is to reduce the number of scenarios.

The optimization problem with uncertain parameters relies on considering multiple decision scenarios, typically denoted as n out of N scenarios. As the problem evolves over time, these scenarios can lead to various branching possibilities, resulting in complex boundaries. To address this complexity, a technique known as scenario reduction is employed to optimize the objective by reducing the number of scenarios involved. This technique helps streamline the problem and make it more manageable by focusing on the most influential or representative scenarios. By reducing the number of scenarios, the optimization process becomes more efficient and effective.

To address this, decision makers can utilize the information obtained from $(\xi_1, \dots, \xi_N)$ and employ scenario reduction techniques to improve the approximation of P using a smaller number of scenarios n. In this paper, scenario reduction is applied to a two-stage linear stochastic program. The study aims to efficiently reduce the number of scenarios and facilitate the search for the optimal solution in the stochastic program.

LITERATURE REVIEW

In the implementation of a stochastic programming model, a crucial step is to model the random parameters that capture the inherent uncertainty. This uncertainty can be expressed using a multivariate continuous distribution or a discrete distribution, depending on the required number of outcomes. In stochastic programming formulations with discrete distributed parameters, these discrete scenarios are often organized in a scenario tree structure, where each node corresponds to a specific stage.

Scenario reduction techniques have been extensively studied in stochastic programming. In the literature, three common methods for constructing scenario trees are discussed. The first method involves generating scenarios by sampling paths based on given data. These paths can be determined through distributions or obtained from historical observations. Researchers like (Consiglio et al., 2014; Mulvey & Vladimirou, 1991) have developed global scenario systems that can simulate any desired path using calibration models. (Wang, 2010), on the other hand, employed multivariate autoregression models to generate paths for various applications in finance, hydropower planning, and water resource management.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12753 p-ISSN : 2541-044X

Once the paths are generated, the next step is to construct the scenario tree structure based on the model's horizons and stages. This process can be carried out using ad-hoc methods, where paths are intuitively cut and pasted to form the desired structure. Another option is to use cluster analysis techniques, which aim to produce an optimal scenario tree structure through stochastic approximation methods. These approaches help in reducing the number of scenarios while maintaining the essential characteristics of the uncertainty, making the stochastic programming model more computationally tractable and efficient.

(Kuhn, 2006) developed the second method by directly generating scenario trees from curvature sequences. The Morkov data structure is used to generate conditional scenarios while keeping the previous decision tree structure in mind. The most recent technique involves sampling important sequences. The tree is updated in each iteration of this algorithm, and the main path samples that have been selected or previously realized are deleted based on the estimated node importance in the current iteration. The resulting tree is dynamic, and the method's efficiency is determined by the sample used.

(Consiglio et al., 2014)'s third method is more flexible than the previous two methods and is appropriate for more complex distributions with a number of constraints. Specifically, the distance between the generated results and the set value is minimized. Although this method is adaptable, the set of relevant scenarios is determined by the problem and set selection. This will have an effect on the model's accuracy, implying that the scenario tree is oriented to the existing problem.

In practice, however, the size of the scenario tree can quickly grow, causing the stochastic programming problem to become enormous. As a result of memory constraints, performing a search is difficult. To solve this large problem, (Heitsch & Römisch, 2003) reduce one of the scenarios, so that the problem of a large scenario size is broken down into smaller subproblems. When this is implemented, the main problem, which provides a lower bound, is formulated first, followed by a subproblem for each scenario. Upper bounds and deductions from the main problem are generated by combining all of the subproblems. The lower and upper bounds eventually converge on the best solution.

In contrast to the sampling procedure, which may necessitate the implementation of parallel computations, (Dupacová, 1995)'s scenario reduction method performs a one-time reduction of paths or scenario trees. Two heuristic algorithms are used to choose the subset of scenarios with the smallest distance to the original set of scenarios. (Beltratti et al., 1999) divides scenario trees into optimal scenario trees and keeps certain scenario fractions from each cluster to represent stochastic conditions.

Various methods have been proposed for constructing scenario trees in stochastic programming models. (Higle et al., 2010; Sen & Higle, 2000) developed a heuristic method that decomposes multivariate problems into univariate ones and utilizes simulation and iterative procedures to construct scenario trees. (Høyland et al., 2003) focused on evaluating the quality of scenario building methods for stochastic programs and established minimum requirements for their use. (Hochreiter & Pflug, 2002) approached scenario tree construction as a multidimensional facility placement problem. Robert proposed scenario reduction techniques based on probability metrics to determine a subset of scenarios that closely approximate the initial distribution.

These methods aim to improve the efficiency and accuracy of stochastic programming models by reducing the number of scenarios or constructing scenario trees that capture the essential characteristics of uncertainty. The effectiveness of these techniques has been demonstrated through numerical examples and comparative analyses.

METHOD

Stochastic Programming

Many decision-making problems can be effectively represented using a mathematical programming approach, with the objective of achieving optimal outcomes. The determination of these optimal decisions is contingent upon various constraints, encompassing factors such as human resources, capital availability, environmental considerations, capacity limitations, and others. The decisions themselves are expressed in terms of variables, which can take on either discrete or non-negative values. Goals and constraints within the mathematical program are formulated as functions of the relevant data, such as unit costs, average production, sales figures, or capacity constraints (Paul & Zhang, 2019).

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12753 p-ISSN : 2541-044X

Let us consider the decision variable $(x_1, \dots, x_n)$, wherein x_n represents the i-th production of n products. The general formulation of the mathematical program can be expressed as follows:

$$\min f(x_1, \dots, x_n) \tag{3}$$

subject to:

$$g_i(x_1, \dots, x_n) \leq 0$$
$$i = (1, \dots, m)$$
$$x_n \in R$$

where X is the set of nonnegative real numbers.

Stochastic programming problems encompass a range of types, including linear, integer, mixed integer, and nonlinear programming problems, each exhibiting specific characteristics associated with the stochastic nature of the data. Hence, it can be asserted that:

1. In deterministic mathematical programs, the data or coefficients involved are specific numbers with known and fixed values. These values are not subject to uncertainty or variability.
2. In contrast, stochastic programming introduces uncertainty into the data or coefficients used in the mathematical program. These uncertain values are represented by probability distributions, reflecting the range of possible outcomes or scenarios. The actual value of the coefficient is not known with certainty but rather follows a probability distribution, capturing the uncertainty associated with the problem.

The stochastic program is a mathematical framework designed to address decision-making situations that involve uncertainty (Hu et al., 2017). This uncertainty is typically characterized by probability distributions assigned to one or more parameters within the constraint function and the objective function. While the underlying uncertainty is well-defined, in practical applications, specific scenarios and composite probability distributions are often utilized to represent the possible outcomes of the uncertain data. The general outcomes are typically described within the scope of a set denoted as $w \in W$.

When certain data elements are stochastic or random, the optimal solution and objective value of the optimization problem also become random. The following are the fundamental models commonly employed in stochastic programming.

Anticipatory model

This model is commonly referred to as a static model, wherein decisions are made without considering any observations or future events. Effective planning necessitates the consideration of all potential future scenarios, as there will be no opportunity to update or modify decisions once they have been implemented.

In the anticipatory model, feasibility is expressed through probabilistic constraints. For instance, the desired level of reliability, denoted as α, where $0 < \alpha \leq 1$, is specified, and the constraints are formulated as follows:

$$P\{w | f_i(x, w) = 0, j = 1, \dots, n \geq \alpha\} \tag{4}$$

In the anticipatory model, the decision variable vector x, with a dimension of m, is subjected to certain constraint characteristics. Specifically, the constraint function $f_j: \Re^m \times \Omega \rightarrow \Re$, $j = 1, \dots, n$, indicating the desired constraints. Additionally, Ω_n represents the set of all random events, encompassing the uncertainty in the problem.

The objective function in the anticipatory model can also take the form of a reliability measure, denoted as $P\{w | f_0(x, w) \leq \gamma\}$, where $f_0: \Re^m \times \Omega \rightarrow \Re$ represents the objective function and γ is a constant. The aim is to select policies that satisfy the given constraint characteristics and optimize the objective function. In summary, the anticipatory model seeks to identify policies that fulfill the prescribed constraint conditions and achieve the objectives outlined in the problem.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12753 p-ISSN : 2541-044X

Adaptive models

In this model, information related to uncertainty is partially available before the decision-making process, enabling optimization within the confines of the available information. Let A represent the collection of relevant information derived from observations, which is a subset of all possible events. The decision variable x is dependent on observable events and is referred to as A-adapted or A-measured. The formulation of an adaptive stochastic program can be expressed as follows:

$$\min E[f_0(x(w), w|A)] \tag{5}$$

subject to:

$$E\left[f_j(x(w), w|A)\right] = 0$$
$$j = 1, \dots, n$$
$$x(w) \in X$$

The mapping of $x: \Omega \to X$ represents a relationship where $x(w)$ is a measured outcome based on the available information A. The adaptive stochastic program is formulated by solving a deterministic program for each possible scenario w. This can be achieved by solving the following deterministic program for each scenario w:

$$\min E[f_0(x,\cdot)|A](w) \tag{6}$$

subject to:

$$E\left[f_j(x,\cdot)|A\right](w) = 0$$
$$j = 1, \dots, n$$
$$x(w) \in X$$

There are indeed two extreme cases: complete information and no information. In the case of complete information, the model takes the form of an anticipatory model. This means that all relevant information is known beforehand, allowing for comprehensive planning and optimization based on the available information.

Conversely, when there is no information available, the model is referred to as a distribution model. This situation occurs when there is a lack of specific knowledge or data to inform the decision-making process. In such cases, the distribution model becomes particularly appealing as it utilizes probability distributions and statistical techniques to make decisions under uncertainty. The distribution model is especially useful when only partial information is available, providing a framework to incorporate uncertainty into the decision-making process.

To summarize, the two extreme cases are complete information leading to an anticipatory model, and no information leading to a distribution model, which becomes more attractive when only partial information is available.

Equivalent deterministic formulation

Based on the provided information, the linear stochastic program model can be represented as follows:

$$\min g_0\left(x, \tilde{\xi}\right) \tag{7}$$

subject to:

$$g_i\left(x, \tilde{\xi}\right) \leq 0$$
$$i = 1, \dots, m$$
$$x \in X \subset \Re^n$$

Here, x is the decision variable vector, $\tilde{\xi}$ is a random vector that varies within the set $\Xi \subset \Re^k$, and F represents the family of events that is a subset of Ξ. The probability distribution P is known for the events in F, and for every event $A \in F$ A ∈ F, its probability $P(A)$ is also known. Additionally, the function $g_i(x,\cdot): \Xi \to \Re \forall i$ represents a random variable with probability distribution P, which is assumed to be independent.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12753 p-ISSN : 2541-044X

Scenario on Stochastic Program

In many practical applications, it is common for the distribution of a random variable to be either unknown or too complex to handle directly in a stochastic programming model. In such cases, an approach is taken to select a representative set of outcomes known as scenarios to capture the randomness of events.

Scenarios are a reduced set of possible outcomes that are chosen to represent the uncertain variables in the problem. These scenarios can be derived from various sources, such as quartiles of known distributions, historical data, predictions, or through simulation methods. Each scenario is assigned a probability value, reflecting the likelihood of its occurrence.

In the context of multi-stage models, scenario information can be organized and represented using a scenario tree structure. This scenario tree provides a visual representation of the decision problem over multiple stages. Each node in the tree represents a specific stage of the problem, and the branches emanating from each node represent different possible scenarios or outcomes at that stage.

Figure 1 illustrates an example of a decision problem that has been presented using a scenario tree for a four-stage problem. This visual representation helps to understand the sequential nature of the decision-making process and the uncertainty associated with each stage.

By incorporating scenarios and their associated probabilities, stochastic programming models can effectively capture and analyze the uncertainty in decision-making, providing insights into optimal strategies under various possible outcomes.

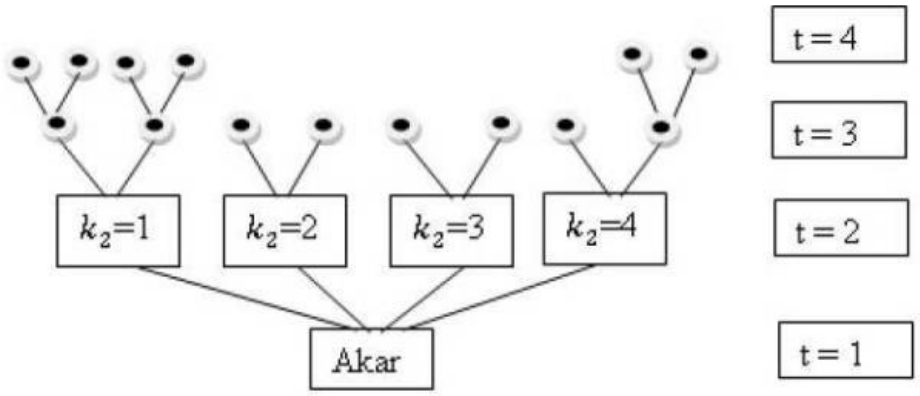

Figure 1. Scenario Tree

In a scenario tree structure, the root node represents the current time or known information, while each subsequent stage denotes a different point in time or part of the data that is uncertain. At each stage, there are multiple possibilities or branches, and each possibility leads to different potential outcomes at the next stage.

A scenario in this context refers to a complete path from the root node to a leaf node, encompassing all the stages and corresponding branches traversed.

Let's consider a scenario tree with T stages. The possible outcomes in each stage can be sequentially labeled as Kt, where $t = 1, \dots, T$. Additionally, within each stage, the branches can be sequentially labeled as kt, where $kt = 1, \dots, Kt$ for all t.

The direct derivative in time t of a particular branch or knot k is denoted as $Dt(k)$. For instance, in Figure 1 of the scenario tree, $D3(1)$ represents the direct derivative of knot 1, which refers to the leftmost two knots at time 3.

At the final stage T, for each knot k, let P_t^k denote the associated probability of that scenario occurring. However, for stages $t = T - 1, \dots, 1$, the probability p_t^k pkt is determined as follows:

$$p_{t+1}^k = \sum_{1 \in D+1 \dots} p_{t+1}^1 \tag{8}$$

With $p_1 = 1$.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12753 p-ISSN : 2541-044X

Decision trees in stochastic programming offer flexibility for modelers to select and prioritize important scenarios. Considering too many scenarios can be impractical, especially for problems with numerous random factors. By focusing on relevant scenarios, decision trees help manage computational complexity and provide valuable insights for decision-making under uncertainty.

Two-stage stochastic programming

Recursive problems in stochastic programming are not limited to two-step formulations. They can involve multiple stages where observations are made, and information is revealed over time. This information is captured in the information pool A_τ. Each stage corresponds to a specific time when decisions can be made based on the available information (denoted by $T(w)$ matrix).

A double-stage stochastic program with recourse incorporates a recourse problem at each stage τ, conditioned on the information provided by A_τ. This includes all information originating from the information set A_τ for subsequent stages $t = \tau + 1, \dots, T$.

Let w be a random vector with supports $\Omega = \Omega_1 \times \dots \Omega_T$, representing the set of products of individual support sets Ω_t for $t = 1, \dots, T$. The first-stage variable vector is denoted as y_0, and for each stage $t = 1, \dots, T$, it is defined as the recourse variable vector $\mathrm{y}_t \in R^{M_n}$. The problem involves random cost functions $q(\mathrm{y}_t, \mathrm{w}_t)$ and random parameters $\{\mathrm{T}_t(\mathrm{w}_t), \Omega_t(\mathrm{w}_t), \mathrm{h}_t(\mathrm{w}_t) | \mathrm{w}_t \in \Omega\}$.

The double-stage stochastic program extends the two-stage model and is formulated as a grouped optimization problem:

$$\min f(y_0) * E\left[\min_{y_1 \in R_+^{M_1}} \xi(y_1, w_1) + \dots + E\left[\min_{y_T \in R_+^{M_T}} \xi(y_T, w_T)\right] + \dots\right] \tag{9}$$

Subject to:

$$T_T(w_T) y_{T-1} + W_T(w_T) y_T = h_k(w_T)$$
$$T = 1, \dots, T$$
$$y_0 \in R_+^{M_0}$$

Discrete probability distributions and finite distributions allow the multi-stage stochastic programming model to be reformulated as a deterministic large-scale non-linear program. This equivalent formulation incorporates uncertainty and enables decision optimization under various scenarios. The resulting deterministic program involves expanding variables and constraints to account for all possible scenarios. Solving this program involves optimizing the objective function while meeting the expanded constraints, using established optimization techniques.

RESULT

In general, the stochastic programming problem is formulated as follows:

$$\min \left\{ E\big(f_0(x, \xi)\big) = \int_{R^S} f_0(x, \xi) P(d\xi) : x \in X \right\} \tag{10}$$

where X represents a closed subset of R^m, f_0 is a mapping function from $R^m \times R^S$ to real numbers $\bar{R} = R \cup \{-\infty, +\infty\}$, and E denotes the expectation operator with respect to the probability distribution P defined on the sample space R^S.

Suppose $\xi = \{\xi_t\}_{t=1}^T$ denotes a d-dimensional discrete-time stochastic process, where each ξ_t is a random vector and $t \in 1, \dots, T$. The objective is to find the decision vector x_t that minimizes the total cost. The optimization model can be formulated as:

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12753
e-ISSN : 2541-2019
p-ISSN : 2541-044X

$$\min \left\{ E\left(\sum_{t=1}^{T} f_t(x_t, \xi_t) \right) : x_t \in X_t, \sum_{\tau=0}^{t-1} A_{t\tau}(\xi_t)\, x_{t-\tau} = h_t(\xi_t), t = 1, \dots, T \right\} \tag{11}$$

The set X_t typically represents a polyhedral set but may include spherical conditions. Additionally, the decision vector $(x_1, \dots, x_T)$ must satisfy dynamic constraints, where the decision x_t depends on previous decisions and achieves equilibrium. The matrix $A_{t\tau}$, $\tau = 1, \dots, t-1$ (containing engineering parameters) and the right-hand side $h_t(\xi_t)$ (representing demands) are random, either partially or entirely. The function f_t represents the cost at time t and is also random, for example, due to uncertain market prices.

At time $t = 1$, which represents the current condition, it is assumed to be deterministic, resulting in the decision x_1 being deterministic as well. The next condition is modeled by the constraint $x_1 = E(x_1)$, which represents the expected value of x_1, given the uncertainty captured by the stochastic process ξ. Thus, the stochastic program in equation (11) can be formulated as an optimization model as follows:

$$\min \left\{ f_1(x_1, \xi_1) + E\left(\phi\left(x_1, \hat{\xi}\right) \right) : x_1 = E(x_1), x_1 \in X_1, A_{10} x_1 = h_1(\xi_t) \right\} \tag{12}$$

For decision x_1 at time $t = 1$, $\hat{\xi}$ represents the uncertainty in the subsequent data process, denoted by $\hat{\xi} := (\xi_2, \dots, \xi_T)$. The function ϕ is defined to capture the dynamic constraints of the problem.

$$\phi\left(x_1, \hat{\xi}\right) := \inf \left\{ E\left(\sum_{t=2}^{T} f_t(x_t, \xi_t) \right) : x_t \in X_t, \sum_{\tau=0}^{t-1} A_{t\tau}\, x_{t-\tau} = h_t(\xi_t), t = 2, \dots, T \right\} \tag{13}$$

The solution to equation (12) provides the minimum cost in the first period, and it is expected that in the subsequent periods, the cost will also be minimized. This is known as the first-stage solution. The stochastic solution $(x_2, \dots, x_T)$ obtained from equation (13) represents the second-stage solution. Thus, model (12) is referred to as a two-stage stochastic program. It is important to note that the first-stage solutions depend on probability distributions from the stochastic process ξ.

In a more realistic scenario, the decision x_t at time t depends only on the available data $(\xi_1, \dots, \xi_t)$, as the data evolves over time. This constraint is modeled using non-anticipativity constraints, defined as follows:

$$x_t = E\left(x_t | (\xi_1, \dots, \xi_t)\right), (t = 1, \dots, T) \tag{14}$$

These nonanticipativity constraints must be incorporated into each set of constraints, including constraints (11) and (13). The expression $E(x_t | \xi_1, \dots, \xi_t)$ in the constraint model (14) represents the expected condition with the random vector $\xi_1, \dots, \xi_t$ assumed to be defined. This constraint model is known as nonanticipativity constraints, and the model (11) with the inclusion of (14) is referred to as a multi-stage stochastic program. When $t = 1$, the constraints in model (14) coincide with the conditions $x_1 = E(x_1)$ in model (11).

If it is specified $x = x_1$ and $X := \{x \in X_1 : x = E(x), A_{10} x = h_1(\xi_1)\}$, so:

$$f_0(x, \xi) := \begin{cases} f_1(x, \xi_1) + \phi\left(x, \hat{\xi}\right), & \text{for } x \in X \text{ and } \phi\left(x, \hat{\xi}\right) \text{ restricted} \\ +\infty, & \text{others} \end{cases}$$

The optimization model (11) takes the form of (10), assuming a feasible ξ and the often-implicit duality condition $\phi\left(x, \hat{\xi}\right) > -\infty$. In this case, a limit of $\phi\left(x, \hat{\xi}\right)$ is guaranteed if a feasible decision $(x_2, \dots, x_T)$ exists for a given x_1.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12753 p-ISSN : 2541-044X

Most approximations to the solution of model (10) are done numerically. This involves replacing the probability distribution P, which is limited to $(\xi_1, \dots, \xi_n)$, with scenarios ξ_i occurring with probabilities $p_i > 0$, $i = 1, \dots, N$ and $\sum_{i=1}^{N} p_i = 1$. The stochastic programming model (10) is thus transformed as follows:

$$\min \left\{ \sum_{i=1}^{N} p_i f_0(x, \xi_i) : x \in X \right\} \tag{15}$$

The integrant property of $f_0(x, \xi_i)$ as a function of ξ together with the characteristics of P determine the type of approach needed in stochastic programming. When seeking solutions (12) and (13), evaluating f_0 on the pair (x, ξ) may be computationally expensive or require substantial time and memory. This poses a challenge in numerical stochastic programming, where an optimal approximation of P may require a large N, but the search for a solution (15) demands a smaller N. Therefore, the application of the stochastic programming model solution often begins with a larger N. However, over time, there is a need to reduce the number of scenarios n.

The decision-maker should leverage the information contained in $(\xi_1, \dots, \xi_n)$ and determine a better approach for P based on the reduced n scenarios compared to the original $(\xi_1, \dots, \xi_n)$. This reduction allows for more efficient and manageable computations while preserving the essential characteristics of the underlying stochastic process.

Optimal Scenario Reduction

The process of optimal scenario reduction aims to approximate the probability distribution P, which consists of N scenarios ξ_i with probabilities p^i, $i \in I := \{1, \dots, N\}$ by determining the probability distribution of Q_n. Q_n is the best approximation of P, considering the distance d to the probability size, which involves subsets of $\xi_1, \dots, \xi_n$ with $n < N$ elements. This is represented as:

$$d(P, Q_n) = \inf \{ d(P, Q) : Q(\mathbb{R}^S) = 1,\ \sup(Q) \subset \sup(P), |\sup(Q)| = n \} \tag{16}$$

The equivalent formulation involves expressing Q_J as the probability measure of $\mathbb{R}^S$ with $\sup(Q_J) = \{\xi^i : i \in \{1, \dots, N\} \backslash J\}$ for some set of indices $J \subset \{1, \dots, N\}$, and q_i, $i \in \{1, \dots, N\} \backslash J$ becomes the probability of the i-th index scenario. The problem can be formulated as follows:

$$\min \left\{ d(P, Q_J) : J \subset I, |J| = N - n, q_i \geq 0, i \in I \backslash J, \sum_{i \in I \backslash J} q_i = 1 \right\} \tag{17}$$

By solving this minimization problem, we can define several sets of indices J^* and corresponding probabilities q_i^* that provide a measure of the scenario probability ξ^i and probability q_i^* for $i \in \{1, \dots, N\}$, which best approximates the original probability distribution P.

The second problem formulation (11) in the optimal scenario reduction problem simplifies the process by allowing for the decomposition into an inner and outer minimization problem:

$$\min_J \left\{ \inf_q \left\{ d(P, Q_J) : q_i \geq 0, i \in I \backslash J, \sum_{i \in I \backslash J} q_i = 1 \right\} : J \subset I, |J| = N - n \right\} \tag{18}$$

The distance d is chosen to ensure that the stochastic program (10) remains stable with respect to d, providing a suitable balance between accuracy and computational efficiency. This enables efficient handling of large-scale stochastic programming problems by reducing the number of scenarios while maintaining the essential characteristics of the underlying probability distribution.

DISCUSSIONS

Many large-scale optimization problems can be effectively addressed using stochastic programming models. The size of the problem has a direct impact on the number of scenarios incorporated in the model to account for uncertainty. Consequently, the computational complexity of solving such a model escalates with an increasing number of scenarios. The fundamental approach to determining the reduction of the optimal scenario commences with the assessment of the probability distribution of Q_n.

The outer minimization problem pertains to a combinatorial optimization problem known as the n-median problem. This problem falls into the category of NP-hard (Nondeterministic Polynomial-Hard) problems. To explore the quest for an optimal solution, two simple heuristic algorithms, namely forward selection and backward reduction, will be employed. Forward selection facilitates the selection of relevant scenarios, while backward reduction enables the elimination of unnecessary scenarios.

REFERENCES

Ahmed, S., King, A. J., & Parija, G. (2003). A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. *Journal of Global Optimization*, *26*, 3–24.

Beltratti, A., Consiglio, A., & Zenios, S. A. (1999). Scenario modeling for the management ofinternational bond portfolios. *Annals of Operations Research*, *85*(0), 227–247.

Birge, J. R., & Dempstert, M. A. H. (1996). Stochastic programming approaches to stochastic scheduling. *Journal of Global Optimization*, *9*(3–4), 417–451.

Chen, H.-H., & Yang, C.-B. (2017). Multiperiod portfolio investment using stochastic programming with conditional value at risk. *Computers & Operations Research*, *81*, 305–321.

Consiglio, A., Carollo, A., & Zenios, S. A. (2014). *Generating multi-factor arbitrage-free scenario trees with global optimization*.

Dantzig, G. B. (1955). Linear programming under uncertainty. *Management Science*, *1*(3–4), 197–206.

Dempster, M. A. H., Hicks Pedron, N., Medova, E. A., Scott, J. E., & Sembos, A. (2000). Planning logistics operations in the oil industry. *Journal of the Operational Research Society*, *51*(11), 1271–1288.

Doğru, M. K., Reiman, M. I., & Wang, Q. (2010). A stochastic programming based inventory policy for assemble-to-order systems with application to the W model. *Operations Research*, *58*(4-part-1), 849–864.

Dupacová, J. (1995). *Scenario-based stochastic programs: Strategies for deleting scenarios*.

Dupačová, J., Gaivoronski, A., Kos, Z., & Szántai, T. (1991). Stochastic programming in water management: A case study and a comparison of solution techniques. *European Journal of Operational Research*, *52*(1), 28–44.

Fleten, S.-E., & Kristoffersen, T. K. (2008). Short-term hydropower production planning by stochastic programming. *Computers & Operations Research*, *35*(8), 2656–2671.

Heitsch, H., & Römisch, W. (2003). Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, *24*(2), 187–206.

Hemmati, R. (2019). Stochastic energy investment in off-grid renewable energy hub for autonomous building. *IET Renewable Power Generation*, *13*(12), 2232–2239.

Higle, J. L., Rayco, B., & Sen, S. (2010). Stochastic scenario decomposition for multistage stochastic programs. *IMA Journal of Management Mathematics*, *21*(1), 39–66.

Hochreiter, R., & Pflug, G. (2002). Scenario tree generation as a multidimensional facility location problem. *AURORA Technical Report, Department of Statistics and Decision Support Systems, University of Vienna*.

Høyland, K., Kaut, M., & Wallace, S. W. (2003). A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, *24*, 169–185.

Hu, S.-L., Han, C.-F., & Meng, L.-P. (2017). Stochastic optimization for joint decision making of inventory and procurement in humanitarian relief. *Computers & Industrial Engineering*, *111*, 39–49.

Huang, G. H., & Loucks, D. P. (2000). An inexact two-stage stochastic programming model for water resources management under uncertainty. *Civil Engineering Systems*, *17*(2), 95–118.

Kenyon, A. S., & Morton, D. P. (2003). Stochastic vehicle routing with random travel times.

Transportation Science, *37*(1), 69–82.

Kuhn, D. (2006). *Generalized bounds for convex multistage stochastic programs* (Vol. 548). Springer Science & Business Media.

Li, Y. P., Huang, G. H., Huang, Y. F., & Zhou, H. D. (2009). A multistage fuzzy-stochastic programming model for supporting sustainable water-resources allocation and management. *Environmental Modelling & Software*, *24*(7), 786–797.

Mulvey, J. M., & Vladimirou, H. (1991). Applying the progressive hedging algorithm to stochastic generalized networks. *Annals of Operations Research*, *31*(1), 399–424.

Niknam, T., Azizipanah-Abarghooee, R., & Narimani, M. R. (2012). An efficient scenario-based stochastic programming framework for multi-objective optimal micro-grid operation. *Applied Energy*, *99*, 455–470.

Paul, J. A., & Zhang, M. (2019). Supply location and transportation planning for hurricanes: A two-stage stochastic programming framework. *European Journal of Operational Research*, *274*(1), 108–125.

Sen, S., & Higle, J. L. (2000). The C3 theorem and a D2 algorithm for large scale stochastic integer programming: Set convexification. *Stochastic E-Print Series*.

Shapiro, A., Dentcheva, D., & Ruszczynski, A. (2021). *Lectures on stochastic programming: modeling and theory*. SIAM.

Wang, Y. (2010). *Scenario reduction heuristics for a rolling stochastic programming simulation of bulk energy flows with uncertain fuel costs*. Iowa State University.