



Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

Stochastic Resource Optimization

Von der Fakultät für Informatik, Wirtschafts- und Rechtswissenschaften
der Carl von Ossietzky Universität Oldenburg
zur Erlangung des Grades und Titels

Doktor der Naturwissenschaften (Dr. rer. nat.)

angenommene Dissertation

von Herrn

Jens Sager

geboren am 16.03.1992 in Schwerin

- 1. Gutachterin:* Prof. Dr.-Ing. Astrid Nieße
Department für Informatik
Carl von Ossietzky Universität Oldenburg
 - 2. Gutachter:* Prof. Dr. Martin Fränze
Department für Informatik
Carl von Ossietzky Universität Oldenburg
 - 3. Gutachter:* Prof. Dr. Lars Nordström
Electric Power & Energy Systems
KTH Royal Institute of Technology Stockholm
- Tag der Disputation: 17.08.2026

Zusammenfassung

Die fortschreitende Einbindung erneuerbarer Energiequellen in das Energienetz stellt neue Herausforderungen an die Koordination von Netzteilnehmern und die zugrundeliegende Infrastruktur. Diese Entwicklung ist ein zentraler Teil der Dekarbonisierung der europäischen Energieerzeugung. Folglich ist die Untersuchung dieser Herausforderungen von kritischer Wichtigkeit.

Diese Arbeit präsentiert Stochastische Ressourcen (SR) – ein Modell unsicherer und volatiler verteilter Energieressourcen. Als Teil des Stochastic Resource Optimization (SRO) Problems untersuchen wir ihre Nutzung zur Einsatzplanung verteilter Energieanlagen. Stochastische Ressourcen sind ein einfaches Modell, das univariate Vorhersagen mit einer zugehörigen Kostenfunktion kombiniert. Diese Kostenfunktion kann z.B. den Verkaufswert der erzeugten Energie darstellen. SRO beschreibt die Aufgabe Energieressourcen aus einer vorgegebenen Menge für ein bestimmtes Ziel kosteneffizient auszuwählen. Dieser Prozess bezieht die Korrelationsinformationen und die resultierende multivariate Verteilung der Ressourcen mit ein.

Dieser Arbeit definiert das formale Modell Stochastischer Ressourcen und führt eine Analyse zu ihrer Aufsummierung durch. Diese Berechnung ist ein kritischer Teil des untersuchten Optimierungsproblems. Für SRO führen wir eine formale Kategorisierung und Komplexitätsanalyse durch und ordnen SRO in den Kontext kombinatorischer und stochastischer Optimierungsprobleme ein. Wir zeigen, dass stark SRO NP-hart ist.

Unsere Komplexitätsanalyse zeigt, dass SRO in einigen Fällen effizient von metaheuristischen Algorithmen gelöst werden kann. Die Lösungsqualität hängt stark von einer vollständigen und genauen Modellierung der unterliegenden Korrelationen der Ressourcen ab. Selbst kleine Ungenauigkeiten und fehlende Informationen können zu einer drastischen Verschlechterung der Ergebnisse führen.

Die entwickelten Optimierungsmethoden werden im Rahmen einer Sensitivitätsanalyse auf ihre Stabilität unter kleinen Änderungen der Eingangsparameter untersucht. Diese Analyse zeigt, dass die untersuchten Methoden generell stabil auf kleine Veränderungen reagieren.

Abschließend präsentieren wir zwei Anwendungsfallstudien für SRO. Die erste untersucht ein idealisiertes Engpassmanagementszenario für ein kleines Nachbarschaftsnetz. Die zweite beschreibt ein großflächiges Planungsproblem für Nordsee Windparks. Diese Studien zeigen, dass das SR Modell und SRO für realweltliche Planungsprobleme nutzbar sind.

Abstract

Continued integration of renewable energy resources into the power grid provides a challenge for the coordination of grid participants and considerations for the underlying infrastructure. As a core part of the decarbonization of European power generation, it is vital that these issues are addressed.

This work presents a model of uncertain distributed energy resources, called Stochastic Resources (SR) and investigates their economic commitment planning as part of the Stochastic Resource Optimization (SRO) problem. Stochastic Resources are a simple model of univariate forecasts for energy resources combined with a cost function for selling their generated energy or power. SRO is the task of selecting the most cost-efficient subset from a set of resources for a given task (determined by a target value). This process considers the correlations of resources and the joint probability of meeting the target.

This thesis contributes the formal model of Stochastic Resources, as well as a formal analysis of their summation. Based on this, we analyze the various properties of the optimization problem. We provide a categorization and complexity analysis of SRO in the context of stochastic and combinatorial optimization problems. SRO is proven to be strongly NP-hard in general.

The complexity analysis shows that SRO can be efficiently solved by meta-heuristic algorithms in some cases. Solution quality depends heavily on the availability of complete and accurate correlation models. Even small inconsistencies in the correlation information can result in large drops in solution quality.

We investigate the implemented solvers for their sensitivity to minor changes in the input values and their performance under imperfect correlation information. The analysis shows that SRO solutions show only minor variation under small changes in the inputs.

Lastly, SRO is evaluated on two application studies – an idealized congestion management problem for a small neighborhood grid as well as a large scale planning problem for North Sea wind parks. The studies show that the SR model and the SRO problem can be used for real world unit commitment problems.

Acknowledgements / Danksagung

Die vorliegende Dissertation ist im Rahmen meiner Tätigkeit beim OFFIS – Institut für Informatik, und der Carl von Ossietzky Universität Oldenburg entstanden. Der Weg hierher umfasst viele unerwartete Wendungen und das finale Ergebnis hat letztlich nicht mehr viel mit der initialen Idee (Multiagentensysteme mit Kontrakten verifizieren) zu tun. Eine Vielzahl von Personen hat mich hierbei über die Jahre mit Rat und Tat unterstützt, und ohne ihre Hilfe wäre diese Arbeit vermutlich niemals fertiggestellt worden.

Allen voran ist hier meine Doktormutter Astrid Nieße, die bei jeder wunderlichen Wendung immer überzeugt war, dass am Ende etwas Sinnvolles entstehen würde. Ebenso danke ich Prof. Martin Fränzle, der ebenfalls von Anfang an in diese Arbeit eingebunden war und dessen Ratschläge und Diskussionen zu den Zwischenständen der Arbeit immer sehr hilfreich und richtungsgebend waren.

Hinweise und Ideen gab es auch regelmäßig von den verschiedenen Mitgliedern meiner Forschungsgruppen am OFFIS (DAI) und der Universität (DES), vor allem durch meine langjährigen Bürokollegen Rico Schrage, Thomas Wolgast und Paul Hendrik Tiemann. Besonders möchte ich an dieser Stelle meinem ehemaligen DAI-Gruppenleiter Martin Tröschel danken, der auch in schwierigen Zeiten immer Zeit und Rat für mich hatte.

Auch außerhalb der Arbeitsgruppen gab es natürlich Unterstützung. Mein ehemaliger Kommilitone Sallar Ahmadi-Pour hatte vor allem in der Orientierungsphase der Arbeit viele spannende Anreize und kritische Fragen für meine eigenen Ideen. Zudem danke ich Prof. Alexey Chernov von der Fakultät für Mathematik für eine interessante Diskussion mit seiner Arbeitsgruppe und Hilfe bei der Formalisierung des SRO-Problems.

Zuletzt möchte ich meiner Familie danken, die mich in allen Lebenslagen unterstützt und auf deren Rückhalt ich mich über all die Jahre immer verlassen konnte.

Wie bei wissenschaftlichen Arbeiten üblich, schließe ich dieses Kapitel mit mehr offenen Fragen, als ich es eröffnet habe. In diesem Sinne hoffe ich, in Zukunft noch viele neue Fragen zu finden und die ein oder andere zu beantworten.

Contents

1	Introduction	5
1.1	Forecast Methods	6
1.2	Planning under Uncertainty	7
1.3	Research Gap and Research Questions	9
2	Preliminaries	13
2.1	Random Variables	13
2.2	Univariate Distributions	16
2.3	Multivariate Distributions	18
2.4	Copulas	21
2.5	Measuring Differences in Distributions	23
2.6	Monte Carlo Simulation	25
2.7	Optimization	29
3	Stochastic Resources	33
3.1	Definition	33
3.2	Sums of Stochastic Resources	34
3.3	Summary	45
4	Stochastic Resource Optimization	47
4.1	Problem Formulation	47
4.2	Categorization	50
4.3	Independent SRO	51
4.4	Implementation in Software	53
4.5	Summary	56
5	SRO Solution Quality	57
5.1	SRO with Full Correlation Information	57
5.2	SRO with Partial Correlation Information	64
5.3	Sensitivity Analysis	69
5.4	Summary	71

<i>CONTENTS</i>	1
6 Case Studies	75
6.1 Idealized Congestion Management	75
6.2 Correlated North Sea Scenario	80
6.3 Summary	85
7 Future Work and Conclusion	87
7.1 Summary	87
7.2 Future Work	88
7.3 Conclusion	89
List of Figures	96
List of Tables	96
A Publications	101
B Full Information Results	103
C Partial Information and Sensitivity Results	111

Acronyms

BPSO Binary Particle Swarm Optimization. 57, 59–61, 79, 80

CCKP Chance-Constrained Knapsack Problem. 8

CDF Cumulative Distribution Function. 13, 15, 18, 21, 22, 25, 30

ED Economic Dispatch. 5, 7

EVO Evolutionary Algorithm. 59

FFT Fast Fourier Transform. 37

FPTAS Fully Polynomial Time Approximation Scheme. 31

OPF Optimal Power Flow. 7

PDF Probability Density Function. 13–18, 24–26, 36, 40

PMF Probability Mass Function. 14–16, 18, 19, 24

PSO Particle Swarm Optimization. 60

RO Robust Optimization. 7, 30, 31

SO Stochastic Optimization. 7, 29, 31

SR Stochastic Resource. 11, 33, 34, 96

SRO Stochastic Resource Optimization. 8–11, 47–58, 61, 62, 64, 66, 67, 69, 71, 75, 78–85, 87–90, 96, 105

TVD Total Variation Distance. 24, 70, 71, 73, 91

UC Unit Commitment. 5, 7

Chapter 1

Introduction

Over the past years, the European energy system has seen a continually increasing share of distributed renewable energy resources. These resources, mostly photovoltaic and wind generators, play a vital role in the decarbonization of our energy production. At the same time they pose big challenges for an energy grid originally designed for centralized energy production. Moving generation to the outer branches of the power grid creates an increased demand for proper coordination to not damage power lines while the increasing number of grid participants complicates this same coordination process. This is especially relevant, as weather-dependent generators have uncertain outputs.

The aim of this thesis is to propose a model for the commitment planning of uncertain energy resources. Operation planning for energy systems is separated into deciding which units to use for a given task (Unit Commitment (UC)) and the efficient operation of the committed units (Economic Dispatch (ED)). This planning involves detailed physical models of the devices and their interconnection. The optimization problem developed in this thesis operates at a higher level, attempting to answer questions like: "Can a given set of renewable energy sources, distributed all over the country, match an expected demand peak on the following day?" It takes their stochastic nature and their various correlations into account but sacrifices physical model detail to incorporate more units and cover larger (physical) areas. This makes it suitable for higher level planning tasks like which types of generators to build and where to put them and thus bridges the gap between the two areas of planning and operation – two areas which have been analyzed separately for conventional systems so far.

The remainder of this chapter gives a short overview of the state of the art in energy system forecasting and methods of dealing with uncertainty in energy systems. Based on this, we derive a set of requirements for a planning model and the core research questions of this thesis.

1.1 Forecast Methods

This section gives an overview of forecast methods for load [27], wind generation [2] and photovoltaic generation [20]. Forecasting in these areas is typically grouped by the time horizon into:

- Very Short Term - seconds or minutes to several hours
- Short Term - one hour to one week
- Medium Term - one week to multiple months
- Long Term - multiple months to years

This summary is focused on Short Term forecasts. These time frames offer the most beneficial trade-off between prediction accuracy and time to act on predictions. Longer term forecasts are similarly suitable for this work in principle but do not cover the intended gap between commitment planning and operation planning for this thesis. Very Short Term forecasts are better suited to the execution of pre-planned strategies.

State-of-the-art forecast methods can be broadly categorized into:

- statistical and probabilistic methods (e.g. based on time series)
- AI methods (e.g. neural network and meta-heuristic algorithms)
- hybrid methods (some combination of the above)

Over the past years, there has been a clear trend in all three forecast areas from statistical methods towards AI and hybrid methods. For example, Kondiah et al. [27] list an Ant Colony Optimization approach using fuzzy logic and deep learning by Soares et al. [45] as achieving particularly high accuracy for load forecasts. A similar result for photovoltaic forecasting is achieved by a hybrid approach of fuzzy logic, neural networks and Particle Swarm Optimization in Dawan et al. [5].

The outputs of different forecast methods vary between giving a single prediction point or a likelihood range of outcomes. Methods that provide a prediction interval are called probabilistic and – if they assume a specific distribution of outcomes – parametric. E.g. if they assume outputs always follow a Gaussian or Beta distribution. Many AI methods produce non-deterministic outcomes so while they may make point predictions, repeated runs of the inference may result in different points, yielding a non-parametric distribution of predictions. Naturally, the same applies to hybrid methods incorporating these AI tools.

Forecasts methods are constantly evolving. As a consequence, the tools developed in this thesis should be general enough to incorporate outputs of most (ideally all) existing and future forecast methods. The issue of different forecast methods can be partially addressed by using the forecast error as the source of uncertainty. The deviation from any given forecast is modeled per time step. The error is commonly assumed to have a Gaussian distribution, e.g.

in Wu et al. [51] although in some cases it may be more accurate to assume other distributions like a hyperbolic distribution as shown in a report by Hodge et al. [16].

1.2 Planning under Uncertainty

This section covers existing methods for the planning of energy systems under uncertainty. It lays out that existing methods with highly detailed physical models are not well suited for the high level commitment planning of energy resources. Secondly, it shows that general stochastic knapsack formulations do not cover all requirements for this task either.

1.2.1 Energy System Planning

In the past years, various methods of addressing uncertainty in energy systems have been developed. The majority of these works can be broadly divided into Stochastic Optimization (SO) and Robust Optimization (RO) approaches. Stochastic optimization assumes a model of the underlying probability distribution of uncertain resources. Robust optimization makes no such assumptions to avoid errors introduced by false distribution models and instead tends to rely on a fixed set of scenarios and pre-defined acceptability bounds.

Works can be divided by the type of problem they attempt to solve. A first level problem like UC usually requires a less detailed model of the devices than an operation scheduling problem like ED or a physical calculation like the Optimal Power Flow (OPF), which additionally considers physical constraints of power lines.

Generally, energy resource forecasts are not independent. E.g. if a PV unit is predicted to produce a certain amount of power, then it would be unreasonable to assume a different PV unit 500 meters down the road would have a completely different output. Some existing works consider correlations [18, 48, 40, 8, 10, 47, 51]. The majority of existing works ignore correlations [4, 19, 32, 35, 37, 50, 55, 30, 39, 46, 14, 56, 53].

For example, Huo et al. [18] solve the planning and operation problem for an energy hub considering uncertainties in wind power generation. Their model uses hourly time steps and Gaussian copulas to model correlations. The non-linear formulation of the power and gas flows is then relaxed into a convex second-order cone problem. The model is evaluated on a network of six energy hubs.

Taking a limited number of possible components, formulating their physical constraints and relaxing the resulting problem to be tractable is archetypal for the majority of works in this field. The amount of detail captured in these models naturally limits their scope. If an energy hub network model is to be expanded by a new energy hub, it requires details on the gas and electricity grid between the new units as well as any energy resources present in the energy

hub. This quickly puts a physical limit on the tractable system size (of course, this is subject to change if computing power increases substantially).

As a result, these methods are a poor fit for higher level planning problems with many energy resources spread over large physical distances. Their physical connections would impose a huge number of optimization constraints, if physical models for the relevant infrastructure exist at all.

1.2.2 Stochastic Knapsacks

If detailed physical models are ill-suited for high level energy resource planning, then perhaps more general selection methods should be applied. The classical knapsack problem describes the task of putting the highest possible value of available items into a knapsack, considering the weight limitation of the knapsack and the value of each item. The commitment planning of energy resources can similarly be formulated as a knapsack-like problem where a limited number of possible energy resources are considered for a task and should be selected to minimize the expected operational costs.

Stochastic variants of the knapsack problem like this have a long history. Different variants consider uncertainty in any part of the knapsack problem, e.g.:

- Available items may be uncertain (chosen from a given pool).
- Costs of items may be uncertain.
- Weights of items may be uncertain.

When the goal is to select an optimal set of items with a minimum amount of certainty (e.g. 95%), then this is called a chance constraint and the optimization task is called a Chance-Constrained Knapsack Problem (CCKP) [52].

One of the most common assumptions is that knowledge of the properties of a selected item is revealed when it is put into the knapsack [7, 6, 26, 54]. That means while the weight of an item may be uncertain in a range from 5 kg to 10 kg, the moment it is actually selected, it becomes known that it weighs 6 kg. This assumption is clearly infeasible for energy system planning. Decision on resources like PV or wind generators has to be made at the time of the forecast and the true value only becomes known after all selection decisions have been made.

Xie et al. [52] show evolutionary algorithms and surrogate functions for the chance constraint of the CCKP. Their items have fixed costs and independent weights. To our knowledge, this is the closest existing knapsack formulation to the Stochastic Resource Optimization (SRO) problem described in this thesis. In fact, this CCKP is a strict subset of SRO. For energy system predictions, independence of generation is not a viable assumption. Energy resources tend to be correlated to varying degrees and these correlations can not be simply ignored. Additionally, costs between different energy resources can vary by time and amount of generation, making the assumption of fixed costs unnecessarily restrictive.

Energy systems require a more generalized formulation than has been studied for knapsack problems thus far. One main reason for this is that existing works prefer strong assumptions on knapsack items. This is mostly used to derive provable properties and closed-form solutions that are of interest from a theoretical viewpoint. Typically, this type of research starts with the simplest possible formulations and reduces assumptions until the problem stops exhibiting a desired property.

On the flip-side, the aims of this thesis are motivated by the application context of energy systems. The ability to incorporate energy system methods as they exist takes precedence. This comes at the cost of strong provable theoretical results for the general formulation of SRO but allows the model to encapsulate practically relevant system models.

1.3 Research Gap and Research Questions

As shown in the previous section, there exists a research gap for a chance-constrained high-level planning approach for distributed energy resources. Existing methods for the planning of energy systems under uncertainty are unsuited for this task due to their reliance on highly detailed physical models and constraints. General stochastic problem formulations are similarly unsuited due to their use of strong assumptions that are not fulfilled by the energy systems to be considered here.

As a result, the aim of this thesis should be to develop a middle-ground formalism that enables the high-level commitment planning of energy resources. This new approach should cover the following functional requirements (FR):

FR 1: Incorporate energy resources with large geographic distances.

The stated aim of this thesis is to enable high-level commitment planning of energy resources. This is only possible if sets of widely distributed resources can be considered within the formalism.

FR 2: Be applicable to most state-of-the-art forecast methods.

As we have shown in the previous sections, forecast methods for energy systems are highly diverse in their inputs, outputs, and accuracy. The methods developed here need to be able to encapsulate these different methods to be of practical use.

FR 3: Incorporate available correlation information and generation costs.

Correlations between energy resources are critically important to commitment planning problems. Even for larger geographic distances, resources can generally not be assumed to be independent. Consequently, the ability to incorporate all available correlation information into the decision-making process is required. This includes both situations when correlation information is complete (i.e. all correlations between all resources are modeled) and when it is incomplete or partial (i.e. only some correlations between resources are known).

Additionally, we define two non-functional requirements (NFR):

NFR 1: Be suitable for Short Term forecasts. While the developed methods and implementations should in principle be applicable to any time horizon, suitability for Short Term and Medium Term forecasts is explicitly required. As outlined in the Forecast Methods section, Short Term forecasts offer the best trade-off between prediction accuracy and the available time to compute cost-effective and reliable commitment plans.

NFR 2: Be efficiently solvable. Here, efficiently means that optimization solutions should be computed or approximated sufficiently quickly for the given prediction horizon. For example, for a week-ahead planning problem, solution times of up to 24 hours may be acceptable. For a time horizon of an hour, solutions (or approximations) need to be available within minutes. In essence, this requirement encapsulates the need for multiple solution strategies applicable to vastly different time horizons to enable any-time decision-making.

From these requirements, we derive the following research question (RQ) and sub research questions (S-RQ) listed in Table 1.1.

ID	Question	Req.
RQ	How can Short Term chance-constrained economic decisions be made for widely distributed energy resources based on arbitrary forecast methods, considering their correlations and generation costs?	FR 1,2,3
S-RQ1	How can widely distributed energy resources be incorporated into the decision-making process?	FR 1
S-RQ2	How can a generally applicable stochastic model with full correlation information for distributed energy resources be constructed?	FR 2, 3
S-RQ3	How can partial correlation information be incorporated in the model?	FR 3

Table 1.1: Research Questions

Artifacts To answer these questions, this work presents three main artifacts. First, the theoretical formulation of the SRO model. Second, the corresponding software implementation and solvers. Third, evaluation of the effects of correlation information on SRO solutions and its applicability to a real world scenario. Table 1.2 shows where in this document each artifact is described.

Publications Parts of this thesis were previously published [41] or are currently being prepared for submission [21]. Table 1.3 lists the publications and where their contents are used in this work.

Artifact	Chapters
Formulation	Stochastic Resources 3, Stochastic Resource Optimization 4
Software	Stochastic Resource Optimization 4
Evaluation	SRO Solution Quality 5, Case Studies 6

Table 1.2: Artifacts and Chapters

Publication	Contents	Chapters
[41]	Stochastic Resource (SR) and SRO definitions, solution quality under full correlation information, congestion management case study	3, 4, 5, 6
[21]	SRO under partial correlation information, sensitivity analysis, North Sea case study	5, 6

Table 1.3: Previous publications and their use in this work.

The remainder of this document is structured as follows. Chapter 2 gives an overview of the mathematical preliminaries of the developed model. Chapters 3 and 4 introduce the Stochastic Resource model and the Stochastic Resource Optimization problem, respectively. Chapter 5 provides analysis of the solution quality of SRO solvers under different conditions and describes a sensitivity analysis of SRO solutions under small changes. Chapter 6 shows two application studies for the SRO model. One for an idealized congestion management scenario in a residential power grid and a second for the planning of correlated wind parks around the North Sea. Lastly, Chapter 7 gives an overview of open questions and future work as well as a summary and conclusion to this thesis.

Chapter 2

Preliminaries

This section introduces some preliminaries from probability theory that are used in this document and their chosen notations.

2.1 Random Variables

Sample spaces and random variables are the fundamental mathematical building blocks to talk about randomness. Their definitions, as used in this work are taken from [3]:

Definition 1 (Sample space and event). The *sample space* S of an experiment is the set of all possible outcomes of the experiment. An *event* A is a subset of the sample space S , and we say that A *occurred* if the actual outcome is in A .

Definition 2 (Random variable). Given an experiment with sample space S , a *random variable* is a function from the sample space S to the n -dimensional real numbers $\mathbb{R}$. It is common to denote random variables by capital letters.

Throughout this dissertation, the focus is on the forecast of load and renewable generation through random variables. For the most part, these random variables are *continuous*. This means that the image of a random variable X is uncountably infinite as opposed to a *discrete* random variable, where it is countable.

Probability Density and Cumulative Distribution Function To reason about the probability that a certain range of values occurs (called the *distribution* of X), we use the Probability Density Function (PDF) and Cumulative Distribution Function (CDF):

Definition 3 (CDF and PDF). The *cumulative distribution function* (CDF) of a random variable. X is the function F_X given by $F_X(x) = \mathbb{P}(X \leq x)$.

The *probability density function* (PDF) of a continuous random variable is the derivative f of the CDF, given by $f(x) = F'_X(x)$. The *support* of X , and of its distributions, is the set of all x where $f(x) > 0$.

A PDF is *valid* if the following properties hold:

- Nonnegative: $f(x) \geq 0$
- Integrates to 1: $\int_{-\infty}^{\infty} f(x)dx = 1$

The equivalent of the PDF for discrete random variables is called a Probability Mass Function (PMF).

Expectation, Variance, and Standard Deviation The expected value, expectation, or mean of a random variable is the measure of its average outcome.

Definition 4. The *expected value* of a discrete random variable X is defined as:

$$\mathbb{E}(X) = \sum_{j=1}^{\infty} x_j(\mathbb{P}(X = x_j)) \quad (2.1)$$

In the continuous case:

$$\mathbb{E}(X) = \int_{-\infty}^{\infty} xf(x)dx \quad (2.2)$$

We use $\mathbb{E}X$ as a shorthand for $\mathbb{E}(X)$

The expectation of the sum of any random variables is always equal to the sum of their expectations. This property will later be important for the calculation of expected costs in stochastic resources.

Theorem 1 (Linearity of Expectation). For any random variables X and Y and any constant c :

$$\mathbb{E}(X + Y) = \mathbb{E}(X) + \mathbb{E}(Y) \quad (2.3)$$

$$\mathbb{E}(cX) = c\mathbb{E}(X) \quad (2.4)$$

The variance and standard deviation of a random variable area measures of how spread out the distribution is from its mean.

Definition 5 (Variance and Standard Deviation). The *variance* of a random variable X is:

$$\text{Var}(X) = \mathbb{E}(X - \mathbb{E}X)^2. \quad (2.5)$$

Its *standard deviation* (SD) is the square root of its variance:

$$SD(X) = \sqrt{\text{Var}(X)} \quad (2.6)$$

Independence Random variables are called independent if knowing the value of one provides no information on the value of another. In this case, probabilities of joint events can be calculated by multiplying each of the constituent probabilities.

Definition 6 (Independence). Random variables $X_1, \dots, X_n$ are independent if for all $x_1, \dots, x_n \in \mathbb{R}$:

$$\mathbb{P}(X_1 \leq x_1, \dots, X_n \leq x_n) = \prod_{i=1}^n \mathbb{P}(X_i \leq x_i) \quad (2.7)$$

Sums of Random Variables One of the core operation of Stochastic Resource Optimization is summing up sets of random variables to find out how much value can be expected to be generated from them and at what cost. The sum of independent random variables is called a convolution.

Theorem 2 (Sums of Random Variables). If X and Y are independent discrete random variables, then the PMF of their sum $T = X + Y$ is:

$$\mathbb{P}(T = t) = \sum_x \mathbb{P}(Y = t - x)\mathbb{P}(X = x) = \sum_y \mathbb{P}(X = t - y)\mathbb{P}(Y = y) \quad (2.8)$$

If X and Y are independent continuous random variables, then the PDF of their sum $T = X + Y$ is:

$$f_T(t) = \int_{-\infty}^{\infty} f_Y(t - x)f_X(x)dx = \int_{-\infty}^{\infty} f_X(t - y)f_Y(y)dy \quad (2.9)$$

Similarly for the CDF (note the use of both CDF and PDF on the right-hand side):

$$F_T(t) = \int_{-\infty}^{\infty} F_Y(t - x)f_X(x)dx = \int_{-\infty}^{\infty} F_X(t - y)f_Y(y)dy \quad (2.10)$$

For dependent random variables, summation requires knowledge of the joint distribution function $f_{X,Y}$:

$$f_T(t) = \int_{-\infty}^{\infty} f_{X,Y}(x, t - x)dx = \int_{-\infty}^{\infty} f_{X,Y}(t - y, y)dy \quad (2.11)$$

Example 1. Consider tossing two independent fair coins where we encode heads as 1 and tails as 0. Then the sum of the results of the two coin tosses $T = C_1 + C_2$ can take on values in $\{0, 1, 2\}$ with:

$$\begin{aligned} \mathbb{P}(T = 0) &= \mathbb{P}(C_1 = 0)\mathbb{P}(C_2 = 0) &&= 0.25 \\ \mathbb{P}(T = 1) &= \mathbb{P}(C_1 = 1)\mathbb{P}(C_2 = 0) + \mathbb{P}(C_1 = 0)\mathbb{P}(C_2 = 1) &&= 0.5 \\ \mathbb{P}(T = 2) &= \mathbb{P}(C_1 = 1)\mathbb{P}(C_2 = 1) &&= 0.25 \end{aligned}$$

Example 2. In an energy system context, we may be interested in the sum of multiple generation forecasts to estimate how much power output we can expect at a given time. Let's say we have two independent resources X and Y with expected power outputs of $\{3, 5\}MW$ and a normally distributed forecast errors with variance $\{2, 3\}$. From the linearity of expectation we can immediately tell that the expected sum output will be:

$$\mathbb{E}(X + Y) = \mathbb{E}(X) + \mathbb{E}(Y) = (3 + 5)MW = 8MW$$

Additionally, the sum of two Gaussian distributions is again a Gaussian distribution. In this case, the sum distribution will have the sum variance of its parts:

$$\sigma_{X+Y}^2 = \sigma_X^2 + \sigma_Y^2 = 2 + 3 = 5$$

Functions of Random Variables Given any function $g : \mathbb{R} \rightarrow \mathbb{R}$ and random variable X , $g(X)$ is also a random variable. If X is discrete, then the PMF of $g(X)$ can be computed as a sum:

Theorem 3 (PMF of $g(X)$). Let X be a discrete random variable and $g : \mathbb{R} \rightarrow \mathbb{R}$. Then the support of $g(X)$ is the set of all y such that $g(x) = y$ for at least one x in the support of X , and the PMF of $g(X)$ for all y in the support of $g(X)$ is:

$$\mathbb{P}(g(X) = y) = \sum_{x:g(x)=y} \mathbb{P}(X = x) \quad (2.12)$$

For continuous random variables this becomes more complicated and requires differentiation (the multivariate case requires a Jacobian matrix which is omitted here).

Theorem 4 (PDF of $g(X)$). Let X be a continuous random variable with PDF f_X and let $Y = g(X)$, where g is differentiable and strictly monotonous. Then the PDF of Y is given by

$$f_Y(y) = f_X(x) \left| \frac{dx}{dy} \right| \quad (2.13)$$

where $x = g^{-1}(y)$. The support of Y is all $g(X)$ with x in the support of X .

The special case where $g(X)$ is linear with positive incline has a name:

Definition 7 (Location-scale transformation). Let X be a random variable and $Y = aX + b$ where a and b are constants with $a > 0$. Then we say that Y has been obtained as a *location-scale transformation* of X where b controls how the location is changed and a controls how the scale is changed.

The location-scale transformation is particularly useful because it allows us to easily fit a given base distribution to a desired range of values. For example, we may have a forecast of solar irradiance for the following day given by a Beta distribution. We can use a location-scale transformation to map this distribution to different PV generators to get an estimate of the distribution of their power outputs by scaling the distribution to their respective nominal power (this is of course a bit more complicated in practice and multiple factors can introduce nonlinear behavior in the relationship between solar irradiance and PV output).

2.2 Univariate Distributions

This section introduces a number of univariate random distributions that are repeatedly used in this work as defined in [3].

Definition 8 (Uniform Distribution). A continuous random variable U is said to have the **Uniform distribution** on the interval (a, b) if its PDF is:

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if } a < x < b, \\ 0 & \text{otherwise} \end{cases} \quad (2.14)$$

This is denoted by $U \sim \text{Unif}(a, b)$.

The uniform distribution has equal probability for any value along its support (the PDF is a horizontal line). The uniform distribution $\text{Unif}(0, 1)$ is called the standard uniform distribution.

Definition 9 (Gaussian Distribution). A continuous random variable Z is said to have the Gaussian or Normal distribution if its PDF f is given by

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}. \quad (2.15)$$

Here, σ is the standard deviation (and consequently, σ^2 the variance) of the distribution with mean μ . Gaussian distributions are commonly denoted by these two values as $N(\mu, \sigma^2)$.

Gaussian distributions are ubiquitous in a wide variety of fields and applications. In this work they are most notable for the distribution of forecast errors for different energy resources. Other distributions, like Beta and Weibull distributions have also been used for the same purpose in wind power forecasting [17]. Although, as Hodge et al. point out, hyperbolic distributions may be better suited in some cases [16].

Within this work, we limit our models to Gaussian, Beta, and Weibull distributions. Here, Beta and Weibull distributions serve as stand-ins for PV and Wind forecasting, respectively. This is not an exhaustive list of distributions commonly used in energy system forecasting. As outlined in the introduction, modern forecast methods may not even provide analytical formulations of their output distributions. The aim of this selection is to represent some variety in the forecast distributions. The methods developed in this thesis can operate on any probability distribution.

Definition 10 (Beta Distribution). A continuous random variable X is said to have the *Beta distribution* with parameters a, b with $a > 0$ and $b > 0$, if its PDF is:

$$f(x) = \frac{1}{\beta(a, b)} x^{a-1} (1-x)^{b-1}, \quad 0 < x < 1 \quad (2.16)$$

Here, $\beta(a, b)$ is a constant chosen to make the PDF integrate to 1. The distribution is commonly denoted as $\text{Beta}(a, b)$.

Definition 11 (Weibull Distribution). A continuous random variable X is said to have the *Weibull distribution* with parameters k, λ with $k > 0$ and $\lambda > 0$, if its PDF is:

$$f(x; \lambda, k) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.17)$$

Here, k is called the *shape parameter* and λ the *scale parameter*.

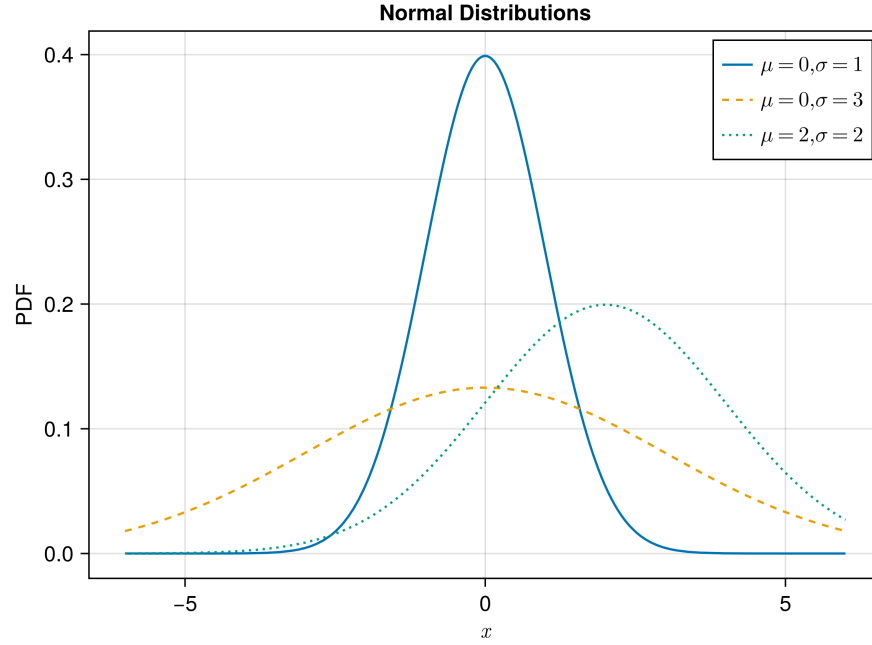


Figure 2.1: Gaussian distributions with different means and standard deviations.

2.3 Multivariate Distributions

Multivariate or joint distributions capture the relationship between multiple (univariate) random variables. The concepts of this section are again taken from Blitzenstein and Hwang [3]. The definitions of the CDF, PMF and PDF are analogous to the univariate case.

Definition 12 (Joint CDF, PMF, PDF). The *joint* CDF of random variables X and Y is given by:

$$F_{X,Y}(x, y) = \mathbb{P}(X \leq x, Y \leq y) \quad (2.18)$$

The *joint* PMF of discrete random variables X and Y is given by:

$$p_{X,Y}(x, y) = \mathbb{P}(X = x, Y = y) \quad (2.19)$$

For continuous joint distributions, we additionally require the CDF to be differentiable for with respect to x and y . We then can derive the *joint* PDF as:

$$f_{X,Y}(x, y) = \frac{\partial^2}{\partial x \partial y} F_{X,Y}(x, y) \quad (2.20)$$

Just as in the univariate case, joint PDFs are required to be nonnegative and integrate to 1. The joint CDF, PMF and PDF of n random variables are defined analogously.

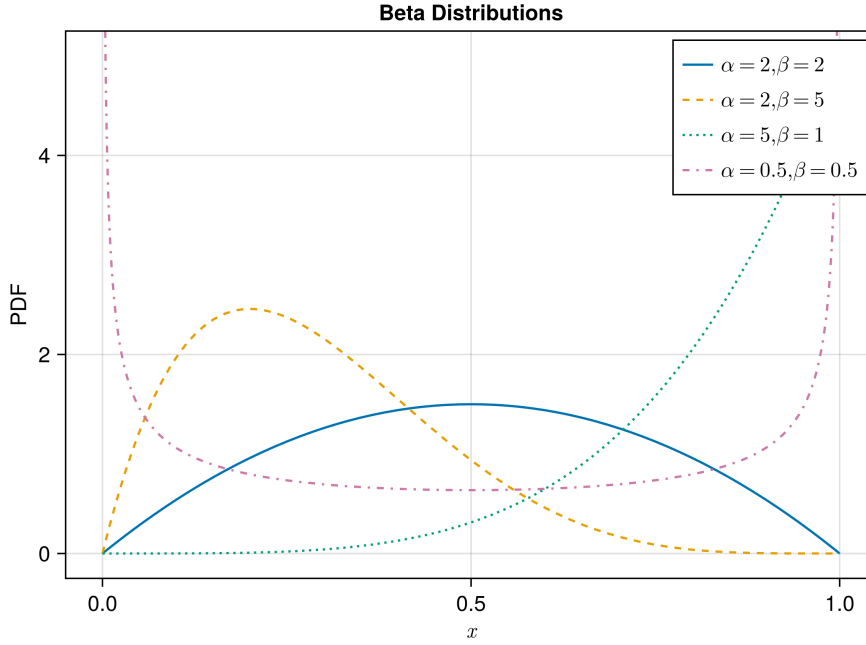


Figure 2.2: Beta distributions with different parameters.

A joint distribution naturally contains full information on its constituent univariate distributions. These are called the marginal distributions.

Definition 13 (Marginal PMF and PDF). For discrete random variables X and Y , the *marginal* PMF of X is:

$$\mathbb{P}(X = x) = \sum_y \mathbb{P}(X = x, Y = y) \quad (2.21)$$

Similarly, for the continuous case:

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y) dy \quad (2.22)$$

The covariance of two random variables is a measure of the relation between two random variables. If two random variables X and Y have positive covariance, then if the outcome of X is high, the outcome of Y is likely to be high too. If they have negative covariance, then if the outcome of X is low, the outcome of Y tends to be high.

Definition 14 (Covariance). The *covariance* between random variables X and Y is defined as:

$$\text{Cov}(X, Y) = \mathbb{E}((X - \mathbb{E}X)(Y - \mathbb{E}Y)) \quad (2.23)$$

Random variables with $\text{Cov}(X, Y) = 0$ are called *uncorrelated*.

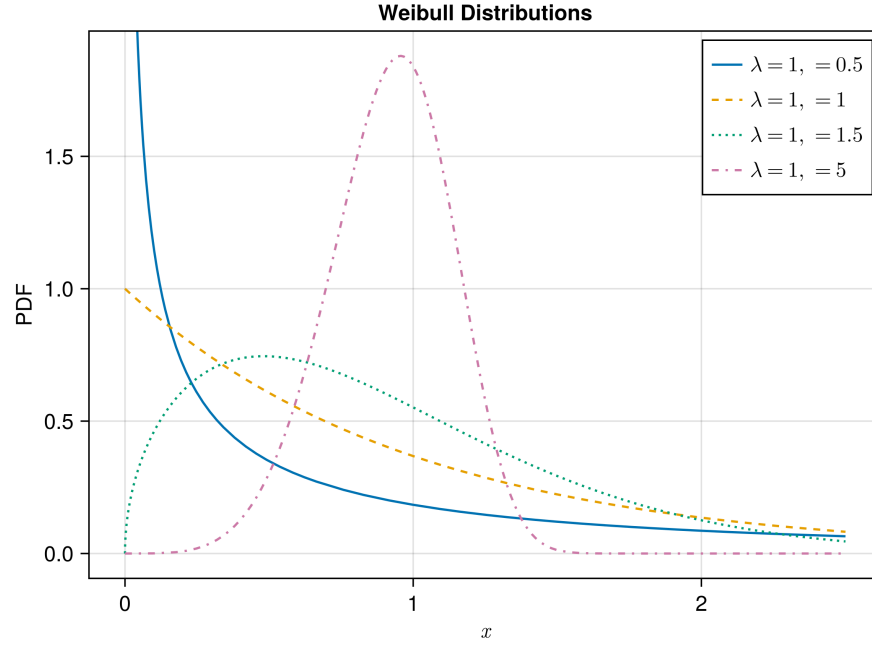


Figure 2.3: Weibull distributions with different parameters.

Theorem 5. If two random variables X and Y are independent, then they are uncorrelated.

We use variance and covariance to define the correlation between random variables.

Definition 15 (Correlation). The *correlation* between random variables X and Y is:

$$\text{Corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}} \quad (2.24)$$

This is undefined if $\text{Var}(X) = 0$ or $\text{Var}(Y) = 0$. In general, it holds that:

$$-1 \leq \text{Corr}(X, Y) \leq 1 \quad (2.25)$$

A common way to display correlation and covariance between multiple random variables are correlation and covariance matrices.

Definition 16 (Covariance and Correlation Matrix). For n random variables, their covariance matrix is the $n \times n$ matrix $\mathbf{M}_{\text{Cov}}$ with:

$$\mathbf{M}_{\text{Cov}}[i, j] = \text{Cov}(X_i, X_j) \quad (2.26)$$

Their *correlation matrix* is the $n \times n$ matrix $\mathbf{M}_{\text{Corr}}$:

$$\mathbf{M}_{\text{Corr}} = \text{Corr}(X_i, X_j) \quad (2.27)$$

2.4 Copulas

This section gives a brief overview of copulas as a method of decoupling the correlation part of a multivariate distribution from its univariate marginal distributions. The definitions used here are based on "Introduction to Copulas" by Nelsen [33]. In particular, we are interested in the construction of multivariate distributions with specific correlation properties (as given by a correlation matrix) using copulas.

Nelsen gives the following intuition for what a copula is: Consider two random variables X and Y with CDF $F(x) = \mathbb{P}(X \leq x)$ and $G(y) = \mathbb{P}(Y \leq y)$ and a joint distribution function $H(x, y) = \mathbb{P}(X \leq x, Y \leq y)$. We can write the ordered pair $(F(x), G(y))$ that maps the real numbers x, y to two values in the unit square $[0, 1] \times [0, 1]$. At the same time $H(x, y)$ provides a single value in $[0, 1]$. The mapping that assigns each value of the joint distribution function to its corresponding value in the ordered pair of marginal distributions (taking into account their correlations) is a function and this function is a copula. This means that in essence the copula encodes all the necessary correlation information to arrive at the correct multivariate distribution for a given set of marginals.

In order to formally define copulas, we require the notions of n -increasing and grounded functions. We give these informally here and only show the case for 2 dimensions. For the formal definition of these terms, we refer to [33].

Definition 17 (n -increasing Function). A function C is **n -increasing** if the function is non-decreasing along any of its dimensions. This is the multidimensional equivalent of a monotonous function. In the 2-dimensional case:

$$\forall u_1, u_2, v_1, v_2 \in [0, 1] \text{ with } u_1 \leq u_2, v_1 \leq v_2 : C(u_1, v_1) \leq C(u_2, v_2) \quad (2.28)$$

Definition 18 (Grounded Function). A **grounded** function C has the property $C(\mathbf{t}) = 0$ for all $\mathbf{t}$ where at least one value t_k has the minimal value of its domain. For copulas, this minimal value is always 0 as all dimensions are defined in $[0, 1]$. This means a 2-dimensional function on $[0, 1] \times [0, 1]$ is grounded if:

$$C(u, 0) = C(0, v) = 0 \quad (2.29)$$

With these, we can define the notion of a subcopula and finally, the copula itself.

Definition 19 (Subcopula). An **n -subcopula** is a function C' with the following properties:

1. $\text{Dom } C' = S_1 \times S_2 \times \dots \times S_n$ where each S_k is a subset of $[0, 1]$ containing 0 and 1.
2. C' is grounded and n -increasing.
3. C' has (one-dimensional) margins $C'_k, k = 1, 2, \dots, n$ which satisfy:

$$C'_k(u) = u \forall u \in S_k$$

Notably, for all $\mathbf{u}$ in $\text{Dom}C'$, $0 \leq C'(\mathbf{u}) \leq 1$, so that the range of C' is also a subset of $[0, 1]$.

Definition 20 (Copula). An n -**copula** is an n -subcopula C with domain $[0, 1]^n$.

As initially stated, copulas allow us to separate the correlation information of a multivariate distribution from its marginal distributions. The key reason for the prevalence of copulas in statistical applications is that this construction is universal as proved by Abe Sklar in the correspondingly named Sklar's Theorem [44]. It states that any multivariate distribution can be expressed as a set of marginal distributions and a copula function:

Theorem 6 (Sklar's Theorem). Let $H(x_1, \dots, x_d)$ be a multivariate CDF with marginal CDFs $(F_1(x_1), \dots, F_d(x_d))$, we can construct a copula C with:

$$H(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)) \quad (2.30)$$

If $(F_1(x_1), \dots, F_d(x_d))$ are continuous then C is unique.

Copulas find applications in various types of statistical modeling and simulation and have been widely studied. As a result, there exist many named types of copulas and copula families with different useful properties. Probably the most notable of these are Archimedian copulas which are associative, have explicit formulations and allow the modeling of multivariate distributions of any dimension with a single parameter for the dependence between marginals.

In contrast, the Gaussian copula does not have these pleasant properties but is still the copula of choice for the modeling and simulations performed in this thesis. As the name suggests, it is constructed from a multivariate Gaussian distribution.

Definition 21 (Gaussian Copula). Let $R \in [-1, 1]^{d \times d}$ be a correlation matrix, then the **Gaussian copula** C_R^{Gauss} is defined as:

$$C_R^{\text{Gauss}}(u) = \Phi_R(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_D)) \quad (2.31)$$

Where Φ^{-1} is the inverse CDF of the standard Gaussian distribution and Φ_R is the joint CDF of a multivariate Gaussian with mean vector $\mathbf{0}$ and covariance matrix equal to R .

The downside of this copula is that it does not admit a simple analytical formula. It can, however, be bounded above and below and approximated numerically.

The main strength of the Gaussian copula is that it can be defined for any correlation matrix or its equivalent covariance matrix. This means that for any set of marginal distributions (e.g. for some renewable generators), we can define the correlations we want them to have and define the corresponding Gaussian copula to construct their multivariate distribution.

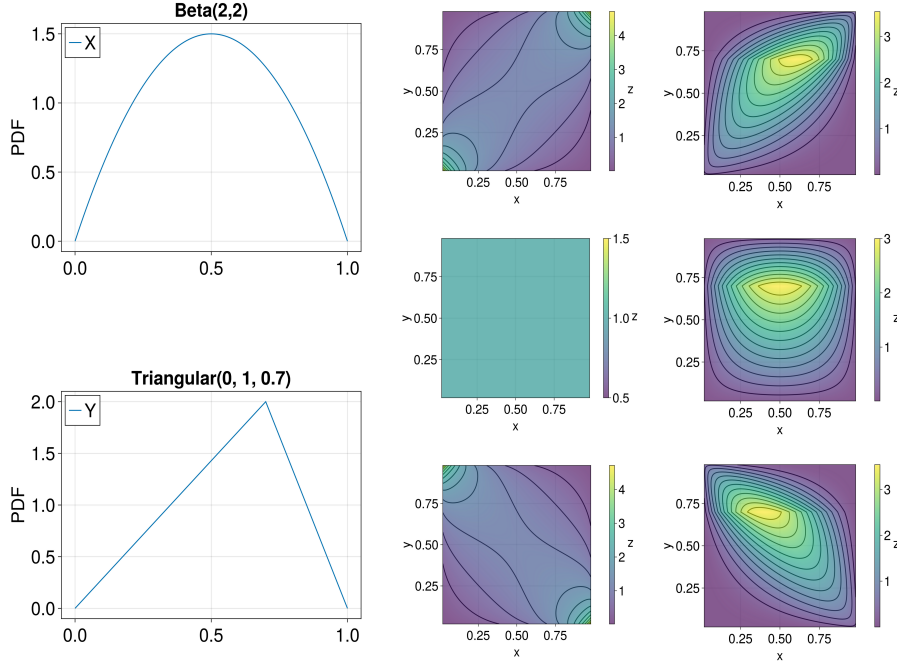


Figure 2.4: A Beta distribution and a triangle distribution (left) correlated using different Gaussian copulas (middle) and their resulting combined distribution (right). The top copula models a positive correlation of 0.5, the middle models independence and the bottom models negative correlation of -0.5 .

Example 3. Let us put all these building blocks together. Figure 2.4 shows two marginal distributions, Beta and triangle distributed, with different correlations modeled by copulas. The top copula is constructed with a correlation coefficient of 0.5, the middle has correlation 0 and the bottom has correlation -0.5 . This results in the expected skew of the combined distribution of the marginals towards the correlation. Computing the Pearson correlation of the resulting bivariate distributions yields the desired results with values of $\{0.49, 0.01, -0.49\}$ for the positive, independent, and negative correlations, respectively.

2.5 Measuring Differences in Distributions

For the analysis of stochastic processes, it is useful to evaluate the difference between different outcome distributions. For this purpose, various distances have been developed. The following gives a brief overview of some of these based on [38] and [11]. Note that some but not all of these distances are metrics, formally. To avoid confusion with metrics and metric spaces, we will always refer

to the probability distances as distances.

Abbreviation	Metric	State Space
D	Discrepancy	metric
H	Hellinger distance	measurable
I	Relative entropy	measurable
K	Kolmogorov metric	$\mathbb{R}$
L	Lèvy metric	$\mathbb{R}$
P	Prokhorov metric	metric
S	Separation distance	countable
TVD	Total Variation Distance	measurable
W	Wasserstein metric	$\mathbb{R}$ or metric
χ^2	χ^2 distance	measurable

Table 2.1: Probability distances covered in [11] with their associated state spaces. This can mean **any** metric space, measurable space, countable space, or the real numbers.

Table 2.1 shows the distances covered in [11]. A big distinguishing factor between distances is the state space they are defined for. Some distances are defined only on metric spaces or – in the case of the real numbers – a single metric space. Others are defined on any measurable space.

In the evaluation chapter, we are specifically interested in distributions of stochastic resources. This type of distribution of discrete entities with no associated numerical value is called a categorical distribution and is neither continuous nor metric. But it is measurable, as any finite or countably infinite set is a measurable set with the power set as its algebra. Of the measurable space distances shown, the most suitable for such distributions is the Total Variation Distance (TVD) as our state space is neither continuous nor metric.

Definition 22. TVD The **Total Variation Distance (TVD)** for two PMFs P and Q is defined as:

$$\delta_{TVD}(P, Q) = \frac{1}{2} \sum_x |P(x) - Q(x)| \quad (2.32)$$

And in the continuous case for PDFs P and Q :

$$\delta_{TVD}(P, Q) = \frac{1}{2} \int_x |P(x) - Q(x)| dx \quad (2.33)$$

The TVD takes values in $[0, 1]$ and is half of the L^1 -norm. The Hellinger distance would be equally suitable as TVD and Hellinger distance are topologically equivalent. The χ^2 distance and relative entropy are not used because they are not symmetric in their inputs (and thus not metrics).

2.6 Monte Carlo Simulation

The following gives a short overview of Monte Carlo methods as they apply to Stochastic Resource Optimization. This section is largely based on the book by Kalos and Whitlock [22] with some probability theory by Blitzenstein and Hwang [3].

What is Monte Carlo? Monte Carlo methods are a class of algorithms that use random sampling to obtain numerical results on problems that are otherwise hard to compute or probabilistic in nature. Their main applications are in optimization (through estimators) and numerical integration, e.g. for functions that do not have an analytical formulation like the Gaussian copula. In Stochastic Resource Optimization, we use Monte Carlo methods to simulate and approximate multivariate distributions of sets of stochastic resources (e.g. renewable generators). A necessary prerequisite to do this with a computer is the ability to collect random samples from an arbitrary distribution using a (Pseudo) Random Number Generator that provides standard uniform random values. This is possible because of the universality of the uniform, also called the fundamental theorem of simulation [3]:

Theorem 7 (Universality of the Uniform). Let F be a CDF which is continuous and strictly increasing. This ensures that the inverse function F^{-1} exists as a function: $(0, 1) \rightarrow \mathbb{R}$. Then the following holds:

1. Let $U \sim \text{Unif}(0, 1)$ and $X = F^{-1}(U)$. Then X is a random variable with CDF F .
2. Let X be a random variable with CDF F . Then $F(X) \sim \text{Unif}(0, 1)$.

The first part of the theorem states that we can simulate any invertible CDF F using samples from a standard uniform distribution. The second part states that if we plug a random variable X into its own CDF, we obtain a standard uniform distribution.

With this, we can define the Monte Carlo estimator G of an arbitrary integral of function f with sample size N :

Definition 23 (Monte Carlo Estimator). Let $\{X_1, X_2, \dots, X_N\}$ be uniform random variables with support in $[a, b]$. Define G_N as:

$$G_N = \frac{b-a}{N} \sum_{i=1}^N f(X_i) \quad (2.34)$$

Then:

$$\mathbb{E}(G_N) = \int_a^b f(x) dx \quad (2.35)$$

This property can be generalized from uniform random variables to other distributions. Let $\text{PDF}(X_i)$ denote the PDF of random variable X_i . This is to

avoid confusion between the integral of the function to approximate (f) and the usual notation for the PDF f_i in this case. Then:

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{\text{PDF}(X_i)} \quad (2.36)$$

$$\mathbb{E}(F_N) = \int_a^b f(x) dx \quad (2.37)$$

This requires that $\text{PDF}(X_i) \neq 0$ for all $x : |f(x)| > 0$.

Example 4. Assume we want to approximate the well known Gaussian integral:

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi} \approx 1.77245$$

Figure 2.5 shows the accuracy of approximations of this integral using uniform random variables $\text{Unif}(-5, 5)$ (G_N) and Gaussian variables $N(0, 1)$ (F_N) for different sample sizes. The uniform bounds do not match the exact integral bounds but the bulk of the function in question is centered around the origin so this still provides a reasonable approximation. The results are shown as the Mean Absolute Error (MAE) from the true value over 100 runs. In general, a Monte Carlo estimator converges faster if the probability distribution of the sampled data is more similar to the function under integration. In this case, the normally distributed random variables lead to much faster convergence to the true value.

Example 5. Consider the example shown in Figure 2.6. These distributions will serve as a running example in future sections. In the top part, we have three truncated distributions on the interval $[0, 20]$ based on a Gaussian, Beta, and Weibull distribution. These could represent forecasts of power generation of three different generators. If we want to analytically compute the sum of these three distributions, we have to solve a messy double integral. Instead, we can approximate the sum distribution using random sampling. The results of this sampling are shown in the bottom plot. We could then fit an appropriate distribution to the sampled data and work with the resulting parameters. In this case, we show a Gaussian fit for the given data which is not perfectly accurate but may be close enough depending on the purpose.

Sample Size A natural followup question for Monte Carlo methods is: How many random samples do we need to take for an accurate approximation? In theory, more samples are always better and by the law of large numbers we must converge to the true value of whatever we are approximating, eventually. In practice, we are bounded by computation time. One possible bound for the sample size can be calculated using Chebychev's inequality [3].

Definition 24 (Chebychev Inequality). Let X be a random variable with mean μ and variance σ^2 , then for any $a > 0$:

$$\mathbb{P}(|X - \mu| \geq a) \leq \frac{\sigma^2}{a^2} \quad (2.38)$$

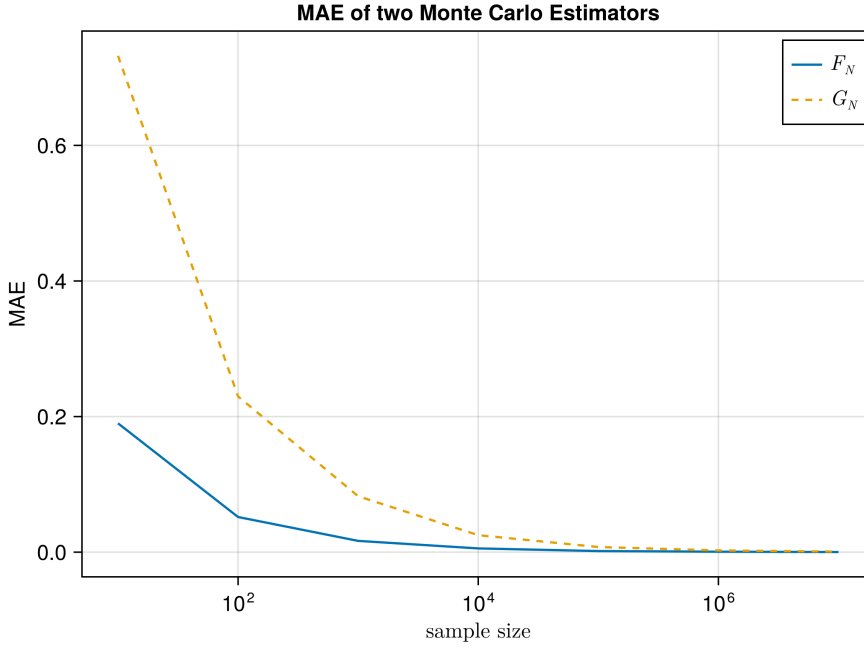


Figure 2.5: Mean Absolute Error (MAE) of two estimators for the Gaussian integral using uniform random variables (G_N) and normally distributed random variables (F_N).

If the variance exists for our Monte Carlo estimator, this means that:

$$\mathbb{P}(|G - \mathbb{E}(G)| \geq \left[\frac{\text{var}(G)}{\delta} \right]^{\frac{1}{2}}) \leq \delta \quad (2.39)$$

where δ is any positive number. This can provide a rough estimate for the number of samples to bound the probability of the samples being a large distance from the actual mean. More precise estimates exist when more assumptions can be made on the quantity to be approximated, but these are not discussed here.

For a quick estimate of a probability bound, the less tight Markov inequality can also be useful.

Definition 25 (Markov Inequality). For any random variable X and constant $a > 0$:

$$\mathbb{P}(|X| \geq a) \leq \frac{\mathbb{E}|X|}{a} \quad (2.40)$$

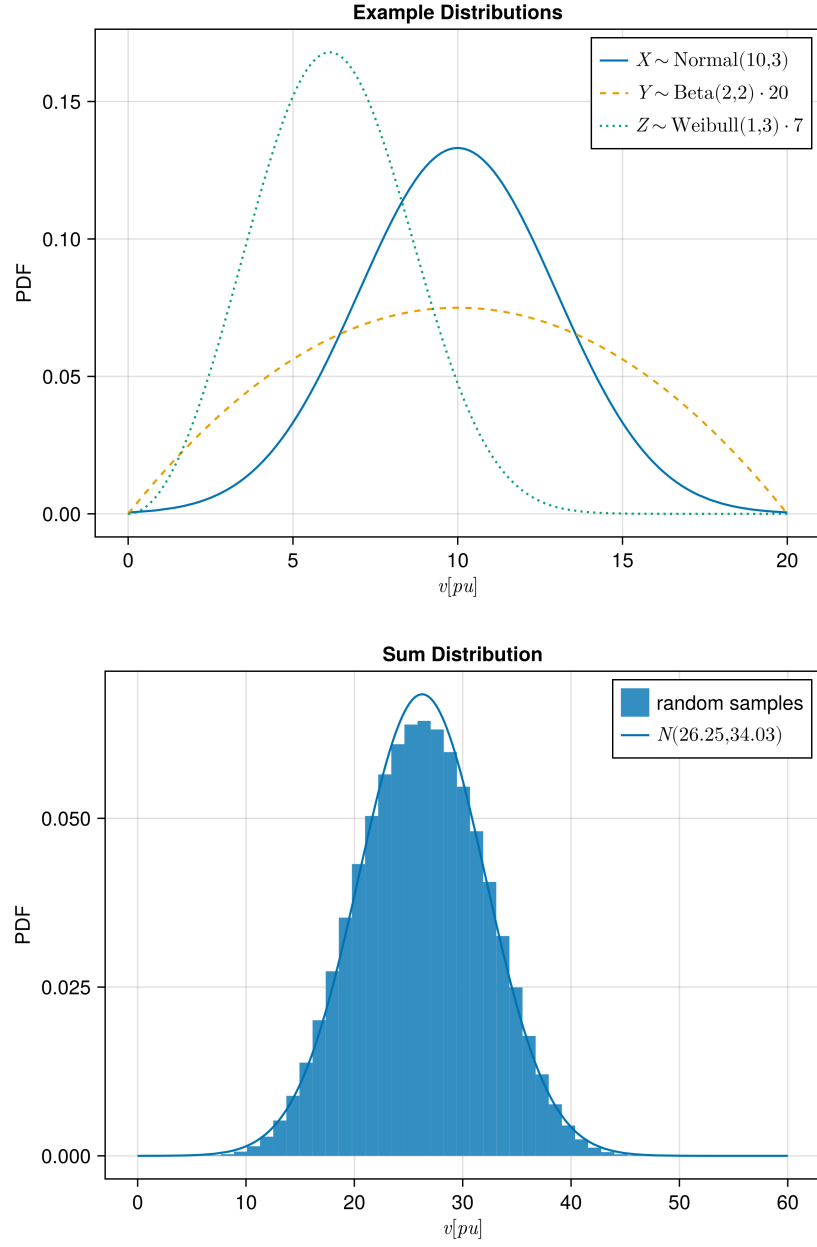


Figure 2.6: Running Example: Three independent Gaussian, Beta, and Weibull distributions and the distribution of their sum. The sum distribution is approximated using random samples with a fitted Gaussian distribution.

2.7 Optimization

This section provides a brief overview of stochastic optimization based on Shapiro et al. [42]. It also provides a distinction from robust optimization [13] as well as connection points to combinatorial optimization [28] as they apply to Stochastic Resource Optimization.

Two-stage Formulations As the name implies, Stochastic Optimization (SO) is a class of optimization problems in the presence of random inputs. Stochastic problems naturally arise in domains with incomplete knowledge or any sort of future prediction involved. A wide variety of stochastic programming formulations with different properties has been developed over the years. A common class are two-stage stochastic programs, consisting of a first stage uncertain optimization problem and a second stage deterministic one.

Definition 26 (Two-Stage Linear Program). A **two-stage linear program** is given by two intertwined optimization problems. The first-stage problem is:

$$\min_{x \in R^n} c^\top x + \mathbb{E}(Q(x, \xi)) \quad (2.41)$$

$$\text{s. t. } Ax = b, x \geq 0 \quad (2.42)$$

And $Q(x, \xi)$ is the optimal value of the second-stage problem with data ξ :

$$\min_{y \in R^m} q^\top y \quad (2.43)$$

$$\text{s. t. } Tx + Wy = h, y \geq 0 \quad (2.44)$$

$$\xi := (q, h, T, W) \quad (2.45)$$

Here, x contains the first stage decision variables, decided with the available uncertain information and y contains the decision variables of the deterministic second stage optimization problem. Some or all of the contents of ξ are random, and the goal is to minimize the expectation in regard to their distribution.

Chance Constraints The previous problem formulation limits the randomness to the objective function. What happens if we want to restrict a property $f(X)$ for an input quantity into our optimization modeled by the random variable X to not exceed some threshold τ ? In classical optimization we could do this with a hard constraint:

$$f(X) \leq \tau \quad (2.46)$$

However, if any outcome of X makes this condition infeasible, then the entire optimization problem becomes inconsistent. The most common way to address this in SO is by using a chance constraint:

Definition 27 (Chance Constraint). Let p be the acceptable probability of failure and τ a threshold quantity. Then they define a **chance constraint** on

the function $f(X)$ of random variable X as:

$$\mathbb{P}(f(X) > \tau) \leq p \quad (2.47)$$

$$\Leftrightarrow \mathbb{P}(f(X) \leq \tau) \geq 1 - p \quad (2.48)$$

Note that the probability formulation here also directly matches the cumulative distribution function (CDF) or survival function (1-CDF) of the random variable $f(X)$.

With this, we can then write the general formulation of a chance constrained program.

Definition 28 (Chance Constrained Program).

$$\min f(x, \xi) \text{ s.t. } g(x, \xi) = 0, \mathbb{P}(h(x, \xi) \geq 0) \geq p \quad (2.49)$$

Here, f is the objective function, g represents a set of equality constraints and h represents a set of inequality constraints.

Assumptions and Solutions All stochastic programming formulations make the key assumption that the uncertain quantities they describe are (accurately) modeled by random variables. This is a strong prerequisite and accurate models based on random variables can be difficult to solve. Many practical solutions to stochastic programs rely on equivalent deterministic relaxations of the problem. Even then, solutions suffer from the curse of dimensionality (i.e. the exponential growth of the solution space in the number of decision variables). This makes Monte Carlo simulations an attractive alternative to closed form solutions to these problems.

Robust Optimization Stochastic optimization requires uncertain quantities to be modeled by known or estimated random variables. In contrast, Robust Optimization makes no such assumption. Instead, RO assumes that the uncertain data is limited to a set of outcomes called an **uncertainty set**. This leads to computationally tractable models for many problem classes. An uncertain linear problem can be formulated as an RO problem as follows:

$$\min_x \{c^\top x : Ax \leq d\} (c, A, d) \in U \quad (2.50)$$

Here, $c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, d \in \mathbb{R}^m$ are the uncertain coefficients and U is the uncertainty set. Just as in stochastic optimization, the decision variables x have to be decided before any outcomes of the uncertain variables are known. The key difference is that the set of possible outcomes is limited and the selection of the uncertainty set is entirely in the responsibility of the user. Because of this, there are no chance constraints in RO formulations. Any constraints are hard constraints on the acceptable set of outcomes.

Incorporating Chance Constraints Choosing the correct uncertainty set for a problem is one of the most important decisions for a robust optimization formulation. They describe an inherent trade-off between robustness and tractability. One way to reduce the size of the uncertainty set is to allow a constraint to almost never be violated. In the absence of chance constraints, this led to the development of safe hard approximations of chance constraints:

$$x \models a(\zeta)^\top x \leq d \forall \zeta \in U, \text{ then } x \models P_\zeta(a(\zeta)^\top x \leq d) \geq 1 - \epsilon \quad (2.51)$$

That is, if x satisfies the hard constraint $a(\zeta)^\top x \leq d$, then it also satisfies the corresponding chance constraint. This type of formulation is used for **distributionally robust** models, meaning models where the underlying probability distribution is itself uncertain.

Advantages and Disadvantages of RO The main advantage of RO over SO is that it requires no assumptions on the underlying probability distributions. This means that in many cases, uncertain outcomes can be formulated as robust problems' safe outcomes and yield tractable solutions. The main downside of this approach is that it (necessarily) tends towards very conservative solutions, leaving much of the solution space unexplored. For the purposes of this work, we will be focusing on stochastic formulations of the studied optimization problem. However, both robust and hybrid approaches of stochastic and robust optimization are sensible candidates for future works.

Combinatorial Optimization The previous examples cover optimization problems with continuous solution spaces. Combinatorial optimization is the class of optimization problems with finite, discrete solution spaces. It covers many useful and famous problems including Integer Linear Programming, graph-based problems like the Traveling Salesman Problem as well as search and sorting problems. The closest relative to Stochastic Resource Optimization is the class of knapsack problems.

Knapsack Problems The classic definition of the knapsack problem is to decide which items from a given set to include in a knapsack of limited size.

Definition 29 (Knapsack Problem). Given a set of items $\{1, \dots, n\}$ with weights $\{w_1, \dots, w_n\}$ and values $\{v_1, \dots, v_n\}$ and a knapsack weight limit W :

$$\max_{x \in \{1, \dots, n\}^n} \sum_{i \in x} v_i \quad (2.52)$$

$$\text{s.t. } \sum_{i \in x} w_i \leq W \quad (2.53)$$

This problem is difficult in general, being NP-hard, but surprisingly easy for an NP-hard problem as it admits an optimal pseudopolynomial time algorithm and a Fully Polynomial Time Approximation Scheme (FPTAS). The FPTAS is

an algorithm that solves the problem up to a factor of $1 - \epsilon$ of optimality in polynomial time.

Many variants of the knapsack problem exist, including stochastic versions. Most of these stochastic knapsack problems assume some part of the problem is uncertain with known probability distributions. This could be the weight or value of the items or even the size limit of the knapsack. Yang et al. [54] studied a variant that is very closely related to Stochastic Resource Optimization:

Definition 30 (Correlated Stochastic Knapsack). There are n items that take random size $s_i \in \mathbb{N}$ with probability $p_i(s_i)$ as well as a reward function $R_i : \mathbb{N} \rightarrow \mathbb{N}$ with $r_i = R_i(s_i)$. R_i is bounded with a maximum value M and assumed to be a non-decreasing function. Finally, there is a budget (or weight limit) $B \in \mathbb{N}$. The task is to maximize the expected reward while holding the budget limits.

This formulation of the problem makes a number of important assumptions. Yang et al. show that under these assumptions, the problem admits a pseudo-polynomial time approximation algorithm:

1. Item sizes are discrete.
2. Reward functions are bounded and non-decreasing.
3. Total profit is a lattice-submodular function.
4. Items are put in the knapsack one by one and the size (and reward) of an item become known as soon as it is put in. If it overflows, the item is not put in and the problem halts.

A submodular function is a function from the power set of a set to the real numbers $f : 2^S \rightarrow \mathbb{R}$ that satisfies the following diminishing returns property for all $T, U \subseteq S$:

$$f(T \cap U) + f(T \cup U) \leq f(T) + f(U) \quad (2.54)$$

A lattice-submodular function is a submodular function where the set S is a lattice (i.e. a set with specific properties). Notably, profits in the Stochastic Resource Optimization problem of this thesis are also submodular.

Knapsacks and their stochastic variants see use in a wide range of applications. In energy systems, deterministic knapsacks have been applied to energy allocation problems [23] and appliance scheduling [43]. Applications on grid-related problems are less common but exist for example in the energy routing in microgrids [34]. The remaining chapters of this work outline an attempt to apply a stochastic knapsack formulation to energy system planning problems. The core elements – the knapsack items – of this approach are Stochastic Resources. Their definition and properties are the subject of the next chapter.

Chapter 3

Stochastic Resources

This chapter introduces SR – a high level abstraction for value-generating stochastic processes. Stochastic Resources combine an arbitrary univariate distribution with a corresponding cost function. They are mainly designed to describe uncertain loads or generators where the value outcome also influences its cost. For example, the total expected electricity consumption for a household and its corresponding cost for a day could be described by a Stochastic Resource. We first provide the formal definition for the SR model with its expressiveness and limitations. We then elaborate on the important computation of sums of stochastic resources under various conditions – a critical operation for the optimization tasks outlined in the following chapters.

3.1 Definition

A Stochastic Resource combines a random variable with real-valued support with an arbitrary cost function.

Definition 31 (Stochastic Resource). A **Stochastic Resource** consists of a univariate random variable X with real-valued support $V \in [v_{min}, v_{max}]$ where v_{min} and v_{max} are the lowest and highest amount of value the resource can generate and a cost function $c : V \rightarrow \mathbb{R}$ giving the purchase cost of each generated amount of value. The random variable of the cost $c(X)$ is referred to as C_X (functions of random variables are random variables).

The distribution is not required to be supported on the whole interval $[v_{min}, v_{max}]$ and can be a discrete distribution. The purpose of Stochastic Resources is to reason about the distributions of sums of random variables where each variable assigns a cost to generated values.

Example 6. A standard 6-sided die becomes a Stochastic Resource if each of its rolls is associated with a cost value. Its cost random variable C_{D6} could be simulated by replacing the 6 digits on the die with their respective cost value.

Example 7. In the energy domain, the day-ahead forecast for the total generation of a wind generator or photovoltaic unit for a given hour with a known feed-in tariff is a Stochastic Resource. Here, energy generation provides the value distribution and the feed-in tariff describes a linear cost (or, in this case, remuneration) function.

Expressiveness The Stochastic Resource model is deliberately simple. First, stochastic resources are limited to univariate distributions. As shown in the preliminaries, any multivariate distribution can be expressed by its marginal distributions and a Copula. So on the distribution side there is no loss of generality. The main drawback is that a single univariate distribution for a resource can naturally only forecast a single quantity – e.g. a single time point or the total generation over a time period. For the general practice of modeling predictions as a series of error distributions over time, this means each time step would be its own set of resources.

Cost functions only take the generation of their respective resource into account. This is a relevant restriction because, in many cases, real cost functions depend on multiple factors. This would be useful for example to model dynamic pricing depending on the total generation of a whole set of resources (e.g. lower prices when PV output is very high). Technically, any set of resources with such a cost function can be unified into a single Stochastic Resource. In practice, this would lead to large and unwieldy multivariate resources that defeat the purpose of modeling them as separate entities.

For the requirements listed in the introduction, this means the Stochastic Resource model is capable of incorporating any forecast method (FR 2) that provides a probability distribution as its output. In practice, this includes methods that do not explicitly provide a distribution. Their outputs can be treated as if they originated from a random distribution for the purposes of Monte Carlo based approaches. Large distances between resources (FR 1) and their correlations (FR 3) are both includable using copulas. This covers all the requirements pertaining to the chosen model, meaning the SR model is generally suitable for the outlined task.

3.2 Sums of Stochastic Resources

Section 2.1 introduced the concept of sums of random variables and functions of random variables. This section expands on this to define the summation of stochastic resources – a key operation for the optimization problem in the next chapter. The core point laid out in this section is that the sum of two stochastic resources is itself a Stochastic Resource. Assume we have two stochastic resources R_1 and R_2 with random variables X_1, X_2 , distributions f_1, f_2 and cost functions c_1, c_2 . We can compute a Stochastic Resource $R_{1+2} = R_1 + R_2$ with distribution f_{1+2} and cost function c_{1+2} . We refer to the distribution of values of their sum f_{1+2} as the **sum value distribution** and the corresponding distribution of costs $c_{1+2}(X_{1+2})$ as the **sum cost distribution**. The remainder

of this section is concerned with computing the sum value distribution and the combined cost function.

3.2.1 Sum Costs

An important question to answer before we define sums of stochastic resources is what the sum of their costs should be. Costs can be arbitrary real-valued functions of the value generation and each resource has its own cost function. This means that the sum of two resources no longer has a single cost value associated with one value. The cost of each value differs based on how much of it is contributed by each constituent resource.

Example 8. Consider the simple case of two discrete independent stochastic resources with value generation $V \in \{0, 1, 2, 3\}$ and

$$c_1(v) = 2v$$

$$c_2(v) = 3v$$

If we arrive at the outcome $V_{1+2} = 3$, there are now 4 possible cost values:

V_1	V_2	$c_1 + c_2$
0	3	9
1	2	8
2	1	7
3	0	6

In countably infinite or continuous cases, there are of course infinitely many combinations. This means there is not a single unique sum cost value for $(c_1 + c_2)(3)$.

In this work, we instead define $c_{1+2}(v)$ as the *expected cost* of outcome v . This reduces the output back to a single value representing the mean cost of the outcome. An important advantage of this approach is that we can subsequently treat the sum of two resources as another Stochastic Resource (with the cost function now being replaced by an expected cost function). The expected cost is the chosen representation of the cost sum of two stochastic resources in this work.

Example 9. Consider the same two resources as in the above example with $V \in \{0, 1, 2, 3\}$ to be equally likely. In this case, each outcome in the value sum table is equally likely, and the sum cost function becomes:

$$c_{1+2}(3) = \frac{9 + 8 + 7 + 6}{4} = 7.5$$

The independence of the involved random variables and the linearity of the cost functions means this average holds for every value:

$$c_{1+2}(x) = \frac{2+3}{2}x = 2.5x$$

Later, in Example 13, we show that this is not always the case.

3.2.2 Independent Resource Sums

Independent random distributions are the simplest case for summation. Here, the convolution theorem applies, meaning summation can be converted into multiplication using the Fourier transform. As we will show, this significantly simplifies (and accelerates) the summation of stochastic resources.

General Formulation Let R_1 and R_2 be independent stochastic resources with value variables V_1 and V_2 or PDF f_1 and f_2 in the continuous case. Since values of stochastic resources are described by random variables, their value sum is the sum of their respective random variables. The sum of two independent random variables is given by their convolution. If R_1 and R_2 are discrete resources, this is given by the probability P of each event:

$$\mathbb{P}(V_1 + V_2 = t) = \sum_x \mathbb{P}(V_1 = t - x) \mathbb{P}(V_2 = x) \quad (3.1)$$

And in the continuous case by the probability density function:

$$f_{1+2}(t) = \int_{-\infty}^{\infty} f_1(t - x) f_2(x) dx \quad (3.2)$$

The expected cost function of the resource sum can be computed using a weighted sum:

$$c_{1+2}(v) = \frac{1}{\mathbb{P}(V_1 + V_2 = v)} \sum_x \mathbb{P}(V_1 = x) \mathbb{P}(V_2 = v - x) (c_1(x) + c_2(v - x)) \quad (3.3)$$

Here, $\mathbb{P}(V_1 = x) \mathbb{P}(V_2 = v - x)$ describes the parts of $f_{1+2}(v)$ (from the convolution). The total sum is normalized by the probability of getting v from the sum. This means that if an outcome accounts for a third of the probability of the outcome, then its cost is weighted at one third as well. This formulation generalizes to the continuous case using probability density functions:

$$c_{1+2}(v) = \frac{1}{f_{1+2}(v)} \int_{-\infty}^{\infty} f_1(x) f_2(v - x) (c_1(x) + c_2(v - x)) dx \quad (3.4)$$

Note. It is important to remember that $c(v)$ and by extension $c_{1+2}(v)$ are not PDFs and do not describe the cost distribution of a Stochastic Resource. For the distribution of cost values, one must compute $c(X)$. In many cases, we are not particularly concerned with the cost distribution, or the cost function of a resource sum for that matter. For the general formulation of the Stochastic Resource Optimization problem introduced in the next chapter, we only require knowledge of the expected cost of a set of resources. This is much easier to compute, as the expected cost of a set of resources is simply the sum of their constituent expected costs, regardless of their dependence or independence. In all cases, **expectation of sum is sum of expectations** as per Theorem 1.

FFT Resource Sums Let us first consider the complexity of computing the sum of two independent discrete resources. Their value sum is the sum of two random variables given by their convolution. In the naive way, this requires N operations executed N times for a complexity of $\mathcal{O}(N^2)$. Since this operation is ubiquitous both in statistics and digital signal processing, many ways have been developed to speed it up. The most common is to make use of the convolution theorem combined with the Fast Fourier Transform (FFT). Let $f * g$ denote the convolution of f and g and $\mathcal{F}(f)$ the Fourier transform of f , then the convolution theorem states:

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g) \quad (3.5)$$

$$\mathcal{F}(f \cdot g) = \mathcal{F}(f) * \mathcal{F}(g) \quad (3.6)$$

That is, convolution in the time (or in our case value) domain is equivalent to multiplication in the frequency domain. For discrete values, the Fourier transform can be computed with an FFT algorithm in $\mathcal{O}(n \log n)$ time. To achieve the equivalent computation of the direct convolution, we need to compute:

$$\sum_x f_1(t-x)f_2(x) = f_{1+2} = \text{IFFT}(\text{FFT}(f_1) \cdot \text{FFT}(f_2)) \quad (3.7)$$

Here, IFFT is the inverse FFT. This entire operation is again in $\mathcal{O}(n \log n)$. In practice, this FFT convolution requires more operations than the direct convolution for small input sizes. Consequently, there exists a size limit for which the direct method is still faster. This depends on implementation. For example, a 2011 test run of a Matlab implementation showed that FFT convolution was faster for inputs starting at length $N = 64$ ¹.

This approach can be extended to the expected cost function of the resource sum. To achieve this, we need to consider both the Fourier transform of the value distributions and the transform of the value distributions weighted by their costs to get:

$$\text{FFT}(c_{1+2}) = \text{FFT}(f_1 \cdot c_1) \cdot \text{FFT}(f_2) + \text{FFT}(f_2 \cdot c_2) \cdot \text{FFT}(f_1) \cdot \frac{1}{f_{1+2}} \quad (3.8)$$

By the same reasoning as above, this ultimately results in a computational complexity in $\mathcal{O}(n \log n)$, though the constant overhead for each FFT call grows as well.

Both the value sum and the cost sum can be extended to n -ary variants to compute the sum of multiple resources. For a set of n resources, we get:

$$f_\Sigma = \text{IFFT}\left(\prod_{i=1}^n \text{FFT}(f_i)\right) \quad (3.9)$$

$$\text{FFT}(c_\Sigma) = \sum_{i=1}^n \left(\text{FFT}(f_i \cdot c_i) \prod_{k \neq i} \text{FFT}(f_k) \right) \cdot \frac{1}{f_\Sigma} \quad (3.10)$$

¹https://ccrma.stanford.edu/~jos/sasp/FFT_versus_Direct_Convolution.html

With this, we can provide a bound on the computational complexity of the sum of n resources. We know the following complexity bounds for the individual operations, assuming all resources have size m :

Operation	Runtime
$\text{FFT}(f)$	$\mathcal{O}(m \log m)$
$\text{IFFT}(F)$	$\mathcal{O}(m \log m)$
$f_1 \cdot f_2$	$\mathcal{O}(m)$
$\prod_{i=1}^n f_i$	$\mathcal{O}(n^2 m)$
$\sum_{i=1}^n \prod_{i=1}^n f_i$	$\mathcal{O}(n^3 m)$

Combining it all, we get:

$$c_\Sigma \in \mathcal{O}(n^3 m + n^2 m + 2nm(\log m + \log nm + 1.5)) \quad (3.11)$$

$$\Rightarrow c_\Sigma \in \mathcal{O}(n^3 m) \quad (3.12)$$

In practice, this means computing resource sums exactly becomes computationally infeasible quickly, even in the simplest case of independent discrete resources. Additionally, the operations result in ever smaller probability bins on the resulting sum resources. On real hardware, this results in rounding errors and necessary corrections at the edges of the distributions. These errors can be accounted for and corrected but impose an additional overhead on the computation.

For our purposes, dealing mostly with continuous distributions and potentially large numbers of resources, approximations of resource sums are generally preferable. For smaller sets of discrete resources, initial benchmarks should be conducted to determine the best method.

Independent Continuous Sums Unlike discrete random variables, the convolution of continuous random variables (and by extension the sum cost function of stochastic resources) is difficult to compute in general. The corresponding integral does not necessarily have a closed form solution. The most common way to address this is to discretize the random variables and treat them as discussed for discrete random variables.

For the convolution of continuous random variables with finite support, Agrawal and Elmaghraby [1] suggest discretizing the distributions using Chebyshev's points. This necessarily changes the exact sum distribution but preserves various moments (key properties of probability distributions) of the true sum distribution. An advantage of this approach compared to other methods is the prevention of support shrinkage during discretization.

This still does not solve the issue of a multiplicatively increasing number of buckets with each new convolution. If the used random variables are sufficiently well-behaved, then computing their convolution analytically should be preferable. The following example gives the convolution and sum cost for two uniformly distributed resources.

Example 10. Consider two independent resources R_1, R_2 with value distributions $V_1 \sim \text{Unif}(0, 5)$ and $V_2 \sim \text{Unif}(0, 5)$ and cost functions $c_1(x) = x^3$ and $c_2(x) = 5x$. The convolution of these two uniform distributions forms a symmetrical triangular distribution on $(0, 10)$:

$$f_{1+2}(v) = \begin{cases} \frac{v}{25} & \text{if } 0 < v \leq 5 \\ \frac{10-v}{25} & \text{if } 5 < v < 10 \\ 0 & \text{otherwise} \end{cases}$$

Solving the piecewise integral for the sum cost function yields:

$$c_{1+2}(v) = \begin{cases} \frac{v^4}{100} + \frac{v^2}{10} & \text{if } 0 < v \leq 5 \\ -\frac{v(v^3 - 20v^2 + 160v - 600)}{100} & \text{if } 5 < v < 10 \\ 0 & \text{otherwise} \end{cases}$$

The results of this example are shown in Figure 3.1. Because of the triangular sum probability density, the resulting sum cost function is a piecewise function with the two pieces meeting at $v = 5$.

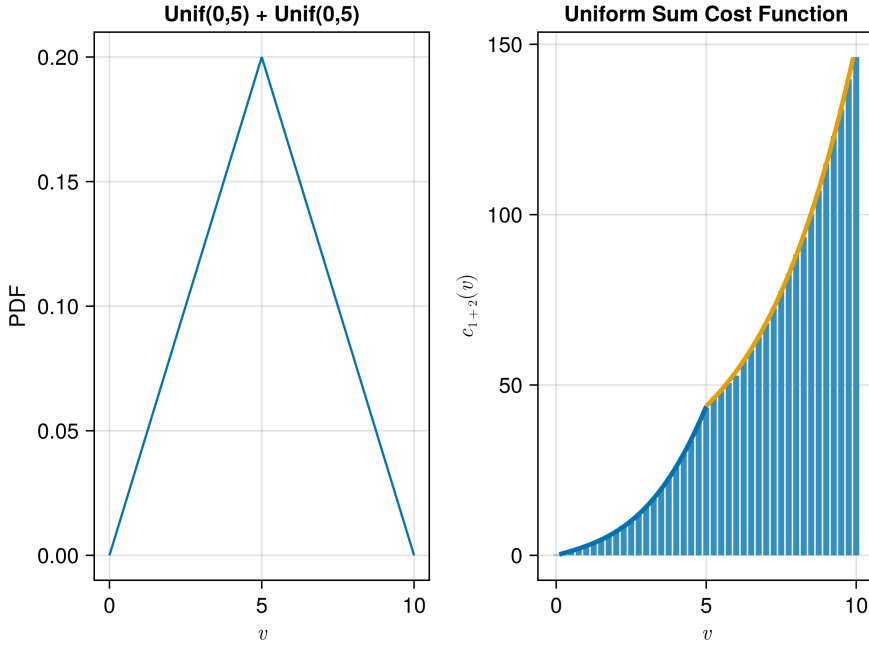


Figure 3.1: Sum of two uniform distributions and their resulting sum cost function. The lines on the right show the piecewise analytical results for the sum cost while the bars show summed random samples from the base distributions.

Example 11. As the previous example illustrates, sums of Stochastic Resources yield unwieldy integrals even for the simplest of cases and become outright analytically unsolvable quickly. Let us return to the sum of random variables shown in Example 5, that is scaled and truncated Gaussian (X), Beta (Y), and Weibull (Z) distributions on the interval $[0, 20]$. Units in the example are in synthetic cost units cu and value units vu . Assume the Beta distribution represents a PV generator, the Weibull is a wind generator and the Gaussian distribution is a hydropower facility. The PV generator has a linear cost of $c_X(v) = 12vcu/vu$ while the wind and hydropower generators follow dynamic pricing schemes:

$$c_Y(v) = (15 - 6 * \frac{1}{1 + e^{-5v+35}})vcu/vu$$

$$c_Z(v) = (\frac{32-v}{2})vcu/vu$$

The cost of the wind plant is $15cu/vu$ for low values, then quickly drops to $9cu/vu$ for values greater than $v = 7vu$. Similarly, the hydropower cost starts at $16cu/vu$ at $v = 0vu$ and drops linearly to $6cu/vu$ at $v = 20vu$.

Figure 3.2 shows the distributions cost functions at the top and the cost functions at the bottom while Figure 3.3 shows the approximated sum distribution and sum cost function. This also shows some limitations of numerical approximation. While the sum cost function is theoretically defined on the whole interval $[0, 20]$, values above 50 are practically almost impossible because of the low probability for values above 10 in the Weibull distribution. Consequently, this part of the cost function is not represented in the data and not part of the polynomial fit.

3.2.3 Dependent Resource Sums

Where independent resources were (in principle) quick and easy to sum, correlations can complicate this task significantly. In general, their sums require knowledge of the multivariate distribution of the involved resources and – for an analytical solution – that an integral over this distribution can be solved.

General Formulation Let R_1 and R_2 be continuous dependent stochastic resources with PDF f_1 and f_2 , respectively and joint distribution function $f_{1,2}$. As established in Section 3.2, we can write the PDF of the value sum of R_1 and R_2 as:

$$f_{1+2}(t) = \int_{-\infty}^{\infty} f_{1,2}(x, t-x)dx$$

For the expected cost $c_{1+2}(v)$, the general idea of equation 3.4 does not change. The only difference is that the constituent parts of the weighted probabilities now must be derived from the joint distribution function:

$$c_{1+2}(v) = \frac{1}{f_{1+2}(v)} \int_{-\infty}^{\infty} f_{1,2}(x, v-x)(c_1(x) + c_2(v-x))dx \quad (3.13)$$

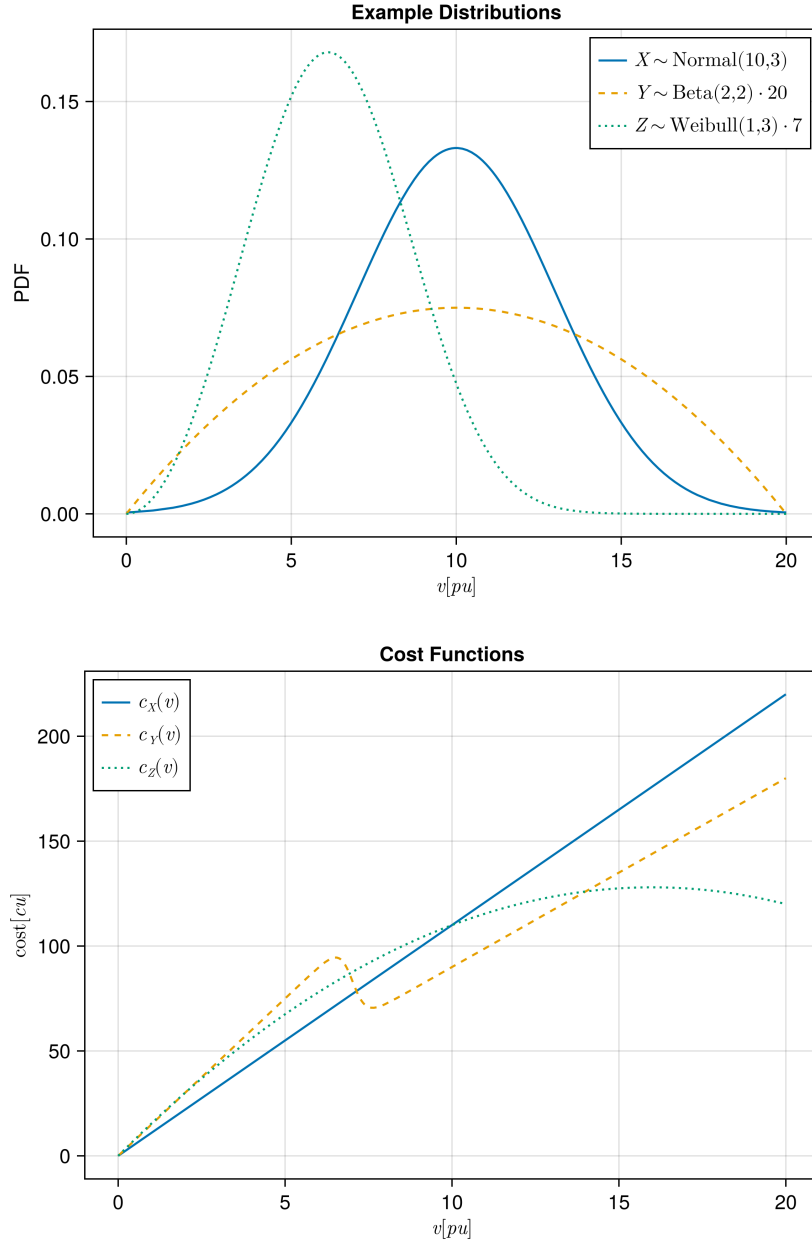


Figure 3.2: Running example distributions and their cost functions. Costs are chosen to simulate that wind (Z) and PV (Y) costs are reduced in times of high wind speeds and solar incandescence, respectively.

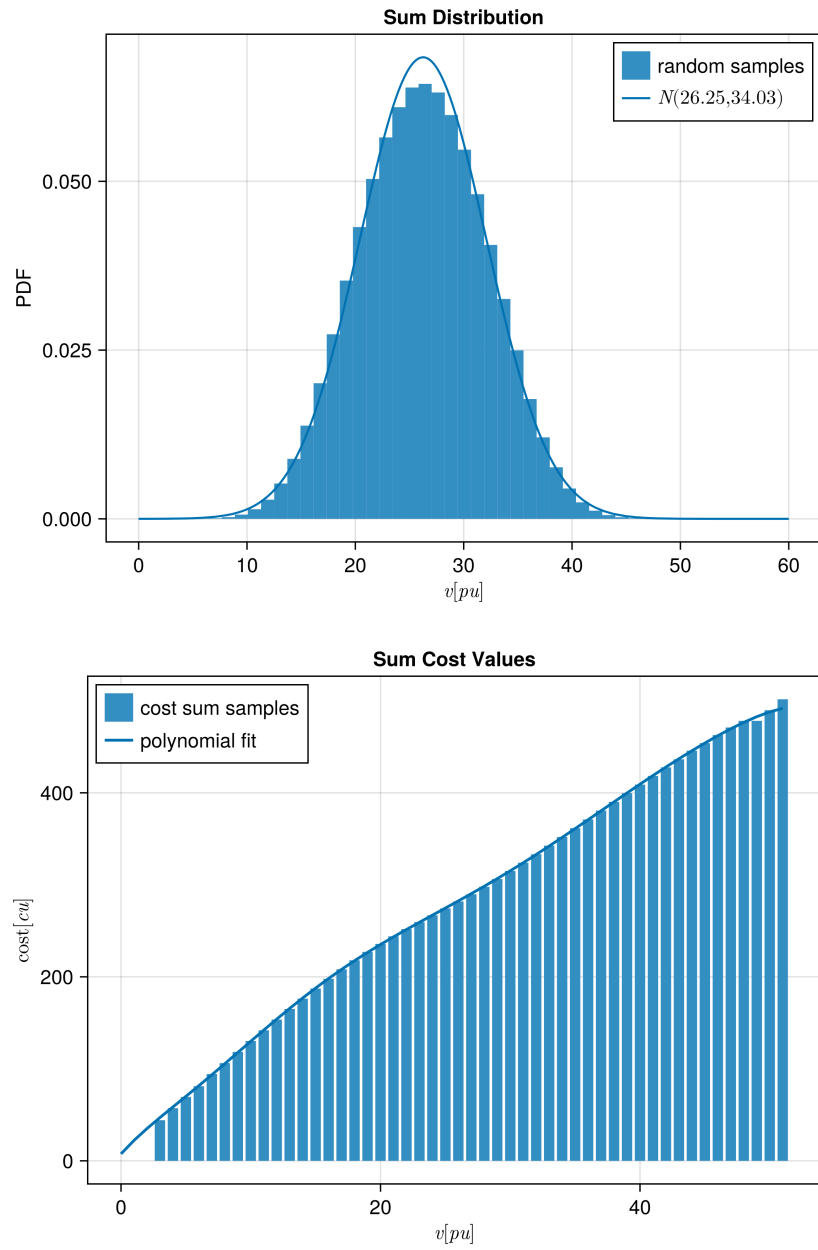


Figure 3.3: Running example sum distribution and sum cost function with a polynomial fit.

The necessity of the joint distribution function complicates the computation as the resource sum is no longer a convolution that can be easily replaced by a Fourier transform and multiplication. In the case of discrete random variables, the run-time is still polynomial in the size and number of resources but prohibitively slow in practice. Most real world sums of dependent random variables rely on various Monte Carlo estimators. This is likely also the best practical solution in most cases for dependent resource sums. The following example shows a simple sum of two copula-correlated resources using a Monte Carlo estimation.

Example 12. Consider two stochastic resources R_1 and R_2 with the Beta and Triangular value distributions from 2.4, scaled by a factor of 5 and correlated using a Gaussian copula with correlation coefficient 0.5. We can assign them cost functions $c_1(v) = v$ and $c_2(v) = \sqrt{v}$. Their bivariate distribution and cost functions are shown in the top half of Figure 3.4. Instead of solving the integral for their sum cost function, we can use Monte Carlo sampling to estimate it. The bottom half of the figure shows the samples of their sum value distribution and the resulting sum cost function with a polynomial fit.

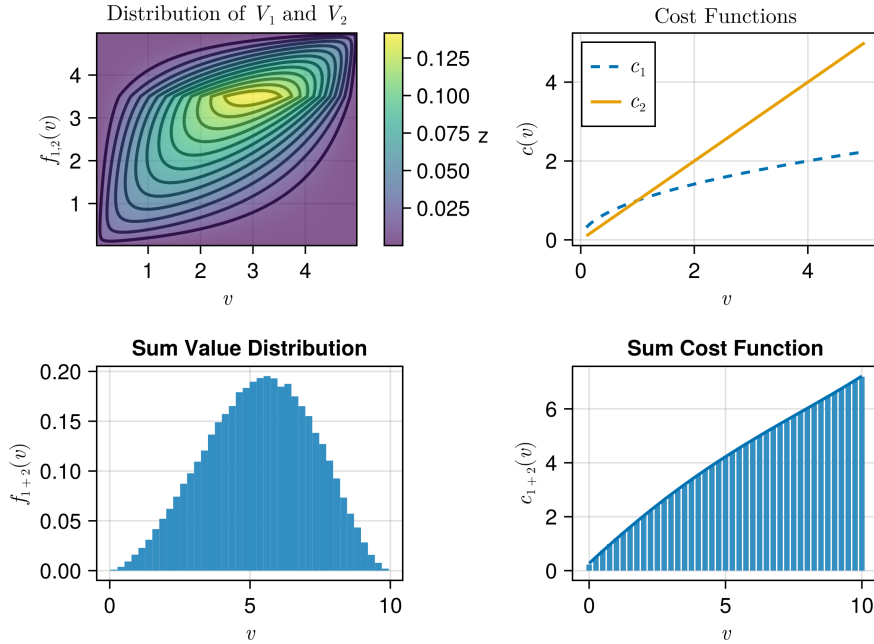


Figure 3.4: Two copula correlated resources and their estimated sum distribution and sum cost functions using Monte Carlo sampling.

Correlation, Symmetry, and Linearity When computing the sum of two resources, computation can be significantly simplified if cost functions are linear

and the underlying multivariate value distribution is symmetrical. However, if this is not the case, then different correlation coefficients can influence the sum cost function. In practice, this means that in most cases the sum cost function is difficult to compute. Even if the marginal distributions are modeled using symmetrical distributions, the multivariate distribution of multiple resources is unlikely to be symmetrical. The following example illustrates this point.

Example 13 (Correlation and Symmetry Affect Sum Costs). Consider the case of a bivariate Gaussian distribution with mean $(0, 0)$ and covariance Σ :

$$f(x; \Sigma) = \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}x^\top \Sigma^{-1}x\right)$$

Assume the two marginal distributions are negatively correlated with

$$\Sigma = \begin{pmatrix} 1.0 & -0.5 \\ -0.5 & 1.0 \end{pmatrix}$$

and we want to calculate the expected sum cost at $v = 3$. The inverse of Σ is:

$$\Sigma^{-1} = \begin{pmatrix} 4/3 & 2/3 \\ 2/3 & 4/3 \end{pmatrix}$$

And its determinant is:

$$|\Sigma| = 0.75$$

We can compute the matrix multiplications in the distribution to get:

$$\begin{aligned} f_{1+2}(v) &= \int_{-\infty}^{\infty} \frac{1}{2\pi\sqrt{0.75}} \exp\left(-\frac{4}{6}(x^2 - vx + v^2)\right) dx \\ f_{1+2}(3) &\approx 0.00221592 \operatorname{erf}((2x - 3)/\sqrt{6}) \\ &= 0.0332377 \end{aligned}$$

Antiderivatives of this form are well-known to not be expressible by elementary functions. This is why it is written here in terms of the error function **erf**.

Similarly, with $c_1(v) = 3v$ and $c_2(v) = 2v$, for the cost function we get:

$$\begin{aligned} c_{1+2}(v) &= \frac{1}{f_{1+2}(v)} \int_{-\infty}^{\infty} \frac{1}{2\pi\sqrt{0.75}} \exp\left(-\frac{4}{6}(x^2 - vx + v^2)\right) (c_1(x) + c_2(v - x)) dx \\ c_{1+2}(3) &\approx \frac{0.0044317}{0.0332377} = 7.5 \end{aligned}$$

Notably, this is the same expected cost as if the two marginal distributions were independent. We can confirm this by using the same steps for:

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

This yields:

$$c_{1+2}(3) \approx \frac{0.0332377}{0.0044317} = 7.5$$

Which is of course the same as using the knowledge of independence and linearity of c_1 and c_2 directly and calculating:

$$c_{1+2}(3) = \frac{c_1(3) + c_2(3)}{2} = \frac{9 + 6}{2} = 7.5$$

The expected cost does not change with different correlations here because of the symmetry of the bivariate Gaussian distribution and the linearity of both cost functions. If we instead set $c_1(x) = x^2$ or $c_1(x) = 2x^2$, this no longer holds as we can show for

$$\Sigma = \begin{pmatrix} 1 & cov \\ cov & 1 \end{pmatrix}$$

with $cov \in \{-0.5, 0, 0.5\}$.

cov	$c_1(x) = 3x$	$c_1(x) = x^2$	$c_1(x) = 2x^2$
-0.5	7.5	6	9
0	7.5	5.75	8.5
0.5	7.5	5.5	8

The squared cost function breaks the symmetry of the cost random variable C . Negatively correlated resources result in higher costs because the average distance of the two outcome values increases, resulting in a disproportionate effect of the squared cost. An approximation of the expected cost functions for each case is shown in Figure 3.5. Note that the expected cost for the linear case would also differ with different covariance if we choose an asymmetrical distribution.

3.3 Summary

In this chapter, we defined Stochastic Resources – structures that combine a univariate random variable with an arbitrary cost function. Stochastic Resources act as a model for uncertain distributed energy resources. The bulk of the chapter was concerned with computing their sums. This is a key operation for the following chapter on the Stochastic Resource Optimization problem. We have shown that the sum of two Stochastic Resources is itself a Stochastic Resource. This sum can be computed efficiently using direct methods or Fast Fourier Transforms in the discrete and independent case – or using analytical methods in the continuous case. However, when correlations and more complicated base distributions are involved, both of these methods quickly become computationally infeasible. For the optimization problem, this means in many cases the most effective approach is to approximate sums of resources using numerical methods.

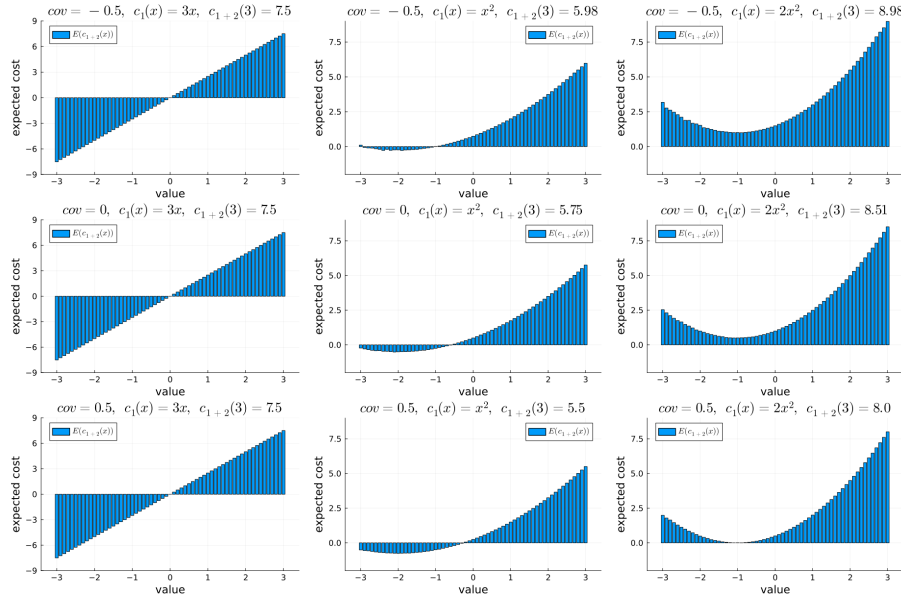


Figure 3.5: Expected cost functions of Example 13 approximated by random sampling. Different covariance values change the expected cost of a value if the cost functions are not linear.

Chapter 4

Stochastic Resource Optimization

The previous chapter introduced stochastic resources as a general model for uncertain distributed energy resources. In this chapter, we investigate how to make informed decisions about which resources to use for a given task. For example, we may own a distributed set of photovoltaic generators and want to decide if we can offer a certain amount of power at a given time tomorrow within a reasonable amount certainty (e.g. 95%). We may not require all our generators for this task and there may be cost differences (both real costs and opportunity costs) between the generators to consider. This chapter covers problems with independent as well as correlated resources. As we will show, if all correlation information is known, the problem can be solved somewhat efficiently using discretization and metaheuristic algorithms. When not all correlations are known, we must work with some assumptions which negatively influence the solution quality.

4.1 Problem Formulation

Stochastic Resource Optimization (SRO) is the process of deciding for a given set of stochastic resources,

- whether the set can provide a given amount of value with a given probability and
- find the most cost-efficient subset of resources to meet the target.

In this way, SRO can help to make unit commitment decisions for uncertain energy resources. A general overview of the process is given in Figure 4.1.

Definition 32 (Stochastic Resource Optimization). Given a set of resources R , a target value v_{target} (or v_t) and a target probability p_{target} (or p_t), the

Stochastic Resource Optimization Problem (SRO) is defined as:

$$\arg \min_{s \in 2^R} \mathbb{E}(C_s) \quad (4.1)$$

$$\text{subject to } 1 - F_{X_s}(v_{\text{target}}) \geq p_{\text{target}} \quad (4.2)$$

Here, X_s is the sum distribution of the resource set s with sum cost distribution C_s and F_{X_s} is the cumulative distribution function of X_s .

Note. The expected cost $\mathbb{E}(C)$ may not exist for any random variable. The addition "if it exists" is left implicit for any mentions of expected values. We assume that in all practical applications a suitable expected value can be approximated even it may not be analytically defined.

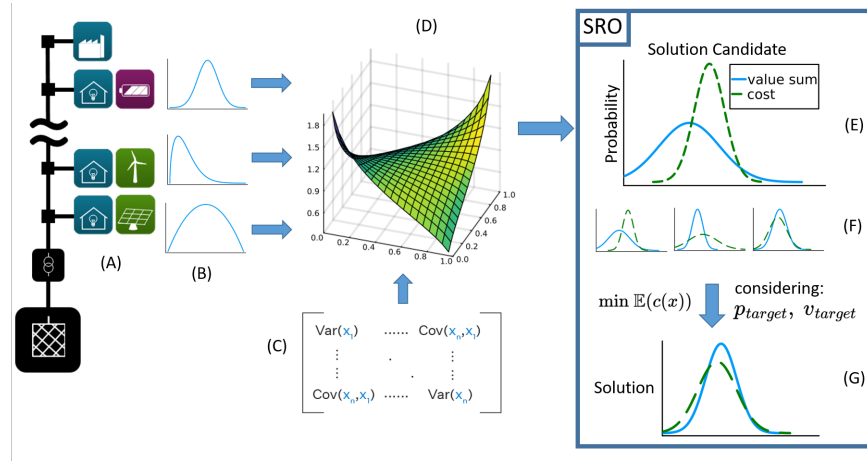


Figure 4.1: General overview of the function of SRO taken from [41] (own work). A subset of resources (A, B) together with their correlations (C) (given here by a Gaussian copula (D)) form a solution candidate (E). Each solution candidate is evaluated by its sum value distribution and its expected cost (F) until the best solution candidate (G) is found.

Example 14. Recall the running example 11 consisting of three resources with truncated Gaussian, Beta, and Weibull distributions. We chose the cost functions there deliberately so that the costs of different resources is the lowest at different points of the value generation range. Let us assume we have 4 of each of these resources and all resources are independent. The following table shows which resource combinations would be selected by an SRO solver for different target values at a target probability of 90%. Notably, the Gaussian and Beta resources are generally picked before the Weibull resources because the generation for the Weibull distribution is skewed to the left where its cost function is also higher than the other two types of resources.

p_{target}	v_{target}	Gaussian	Beta	Weibull	total
0.9	10	1	0	1	2
0.9	20	0	3	0	3
0.9	30	1	3	0	4
0.9	40	3	2	0	5
0.9	50	1	4	2	7
0.9	60	2	4	2	8
0.9	70	4	4	1	9
0.9	80	3	4	4	11
0.9	90	4	4	4	12
0.9	100	-	-	-	-

Table 4.1: SRO solutions for different target values at target probability 90% for 4 sets of the running example resources. At $v_{target} = 100$ the problem becomes infeasible and no resource set is returned.

Why minimize $\mathbb{E}(C_s)$ instead of $c_s(v_{target})$? That is, why minimize the expected cost of the sum cost function and not the expected cost of the target value for the resource subset? SRO was developed assuming that once a resource is selected, all its generated value will be bought. Minimizing the expected cost accounts for the possibility of other total values being generated. This simplifies the summation process – as illustrated in 3.2 – by directly working with the cost distributions.

Minimizing cost at exactly the target value is preferable when no buying guarantee exists. In [41], both a *buy all value* and *buy only necessary value* market type were considered. The latter severely complicates the optimization process. As shown in Example 3.5, computation of the expected cost function of a set of resources can be difficult to impossible.

In a Monte Carlo based solver, the minimal sum cost for the target value has to be determined for any simulated outcome of the random variables. While the two market schemes result in different costs, the performance of the investigated heuristic solvers was unaffected. Since this is expected to be the case for any resources with well-behaved cost functions, later works focus on minimizing $\mathbb{E}(C_s)$ as the goal.

Notably, the buy-necessary market trivializes the resource decision process if there are no up-front (or opportunity) costs to selecting a resource (i.e. $c(0) = 0$ for all resources). In this case, selecting all resources every time is optimal because it maximizes the probability of supplying the target value while any excess generation can simply be ignored. The problem then reduces to an optimization problem on how much value to buy from each resource once the outcomes are determined.

4.2 Categorization

SRO has strong similarities and some key differences to existing optimization problem formulations. For a stochastic optimization problem it has a very simple formulation with only a single chance constraint and an easily computable objective function (if the expected cost of each resource is known, this is just a simple sum over the subset). The main difficulty is the result of being combinatorial. As such, it is best thought of as a particular stochastic knapsack variant.

SRO as a Knapsack Problem SRO bears close resemblance to the correlated stochastic knapsack problem by Yang et al. [54] from Definition 30. The key differences are:

1. In SRO, how much value was generated by a resource does not become known on selection.
2. SRO allows continuous and unbounded values while in [54] they are discrete.
3. SRO allows arbitrary cost functions while in [54] they are non-decreasing.

The one assumption that remains is that both SRO and the stochastic correlated knapsack problem have submodular objective functions. The unbounded values and arbitrary cost functions are especially detrimental as many approximation scheme results for knapsack variants rely on some bounds to exploit.

Following the complexity results for the summation of stochastic resources in Section 3.2, it is easy to see that SRO with discrete and independent resources is in NP as the resource sum can be computed in polynomial time. Additionally, the 0-1 knapsack problem can be encoded into an SRO problem. This yields the following result.

Theorem 8. Stochastic Resource Optimization is strongly NP-hard.

Proof. Recall the 0-1 knapsack problem introduced in Definition 29: It consists of a set of n items with positive weights w_i^K and values v_i^K for $i \in \{1, \dots, n\}$. The aim is to find the subset of items that maximizes the sum of values while having a sum weight of at most $W > 0$.

Each of these items can be encoded into a Stochastic Resource with a single value $v_i^{SRO} = -w_i^K$, $\mathbb{P}(X_i = v_i^{SRO}) = 1$ and $c_i(v_i^{SRO}) = -v_i^K$. Then set $p_{target} = 1$ and $v_{target} = -W$. Now, minimizing the expected cost in the SRO problem corresponds to maximizing the value in the knapsack problem and if more items are selected than the knapsack capacity W allows, then the chance constraint is violated, making the solution infeasible. Thus, the solution set to the SRO problem is the same as the set of items that solve the knapsack problem. This encoding can be done in polynomial time.

Wojtczak [49] showed that the decision variant of the 0-1 knapsack problem with rational weights and profits is strongly NP-complete. Since we did not

restrict the profits or values in the previous steps in any way, the decision variant of SRO is strongly NP-hard and since the optimization variant solves the decision variant, it too is strongly NP-hard. $\square$

This result means that the general formulation of SRO does not admit a pseudo-polynomial time algorithm. Since its objective function is polynomially bounded, it can also not have a fully polynomial time approximation scheme unless $P=NP$. Note that for this result we used a very simple variant of SRO with degenerate probability distributions and independent resources. Naturally there are also subclasses of SRO that are easy to solve. These mostly rely on restrictions on the resource distributions that allow the exponential search space to be reduced. One such condition is discussed in the next section.

4.3 Independent SRO

The main difficulty in solving SRO instances lies in exponential number of possible resource combinations. Consider the simplest case of discrete and independent resources. As we have shown in the previous chapter, we can compute the sum of any subset of such resources in polynomial time. If all resource combinations have to be checked this way, then the problem is completely dominated by the 2^n possible combinations. There is no way around this in the general case, but there are resource sets for which the problem becomes significantly easier.

Sorting Resources If we can establish an ordering of the resource set such that each resource is strictly better for the given problem than the previous, then the number of resource combinations to check reduces to $\mathcal{O}(n)$ after sorting. One such order emerges if a resource is strictly better in all possible problems.

Definition 33. A Stochastic Resource R_1 **dominates** a resource R_2 , written as $R_1 \succeq R_2$ if for all x :

$$1 - F_1(x) \geq 1 - F_2(x) \quad (4.3)$$

$$E(c_1(R_1)) \leq E(c_2(R_2)) \quad (4.4)$$

This notion of domination is very strong as it requires R_1 to have strictly higher probability to generate at least any given value x and a lower expected cost. The first part ensures that R_1 always contributes more to the chance constraint while the second ensures it is always cheaper to take.

Theorem 9. An SRO problem with resources $\{R_1, R_2, \dots, R_n\}$ with a total domination order $R_i \succeq R_{i+1} \forall i$ can be solved in at most n subset checks.

Proof. Without loss of generality, we can assume exactly one resource has been selected for the problem. This works because we can replace any selected subset of resources with the resource created from their sum. Assume that there exists a resource R_{i+e} that achieves a better solution than R_i . By definition, $E(c_i(R_i)) \leq E(c_{i+e}(R_{i+e}))$. Therefore, the only way R_{i+e} achieves a better

solution than R_i is if R_{i+e} fulfills the chance constraint, while R_i does not. But we know that $1 - F_{i+e}(x) \leq 1 - F_i(x)$. Therefore, no such resource R_{i+e} exists and the problem solution must contain R_i and the resources $R_j, j < i$. $\square$

We do not require a stronger notion on the cost side of the relation because SRO optimizes for expected cost, which is the sum of the expected costs of each resource in the solution.

Example 15. Independent and identically distributed (i.i.d.) resource sets always contain a full domination order. Alternatively, we could hold the cost function fixed across all resources and shift their distributions left or right along the value range while maintaining a domination order.

This condition quickly breaks if resources don't follow very specific (or identical) distributions. Figure 4.2 shows the example of two Gaussian distributions with the same mean but different variances which makes their survival functions intersect, which means the first condition of domination can not be met.

Even in the absence of a complete domination order, sorting resources by their expected cost and checking subsets in this order may be a quick heuristic for many problem instances. This is the equivalent of a greedy solver in a classical knapsack problem which is part of an experiment in Chapter 5.

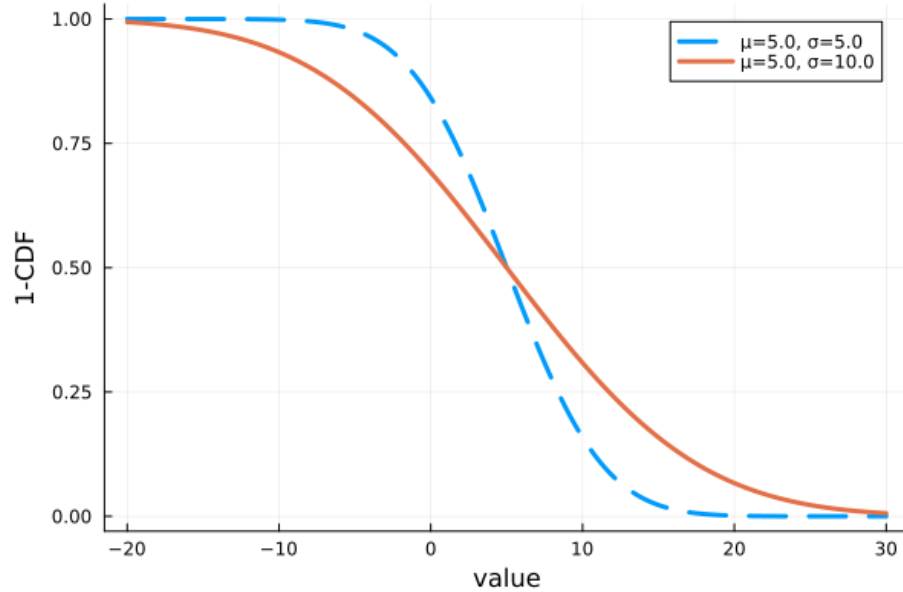


Figure 4.2: Two Gaussian distribution survival functions intersecting because of different variances.

4.4 Implementation in Software

SRO was developed in tandem with the software package `SRO.jl`¹ in Julia². This section gives an overview of the implementation at the time of writing this thesis. The corresponding version of the package is tagged in the git repository.

4.4.1 Data Model

The data model closely follows the general formulation of SRO. It consists of four main structures. Three of these encapsulate the general SRO problem formulation and solution: `SROResource` models a single resource with a univariate value distribution and a corresponding cost function. `SROProblem` encapsulates a single optimization problem instance consisting of a set of resources, a target value and probability, and the correlations of the resources given by a `CorrelationDict`. Lastly, `SROSolution` gives a general output structure for any solver implementation. A solution contains a copy of the problem to solve, the subset of resources returned by the solver, given by their indices, and the expected cost of the solution.

The `CorrelationDict` is the only main structure with more involved logic. Its keys are tuples of indices, representing a specific subset of resources and the corresponding value is the object that generates multivariate samples for these resources according to their correlations. This multivariate object can be of any type, most commonly a multivariate distribution. The only requirement is that it implements the `Base.rand` to allow generating sample data.

Since correlation information can be quite fractured, the dictionary can contain many overlapping subsets of indices. When adding new elements, this can lead to the same subset of resources being mapped to two different distributions. E.g. if the dictionary has two distributions for the tuples $(1, 2, 4)$ and $(1, 2, 3)$, and we want to sample random data for the tuple $(1, 2)$, then both of these entries are valid sources for the data. The only exception to this rule is if a tuple is a strict subset of another tuple. In this case the smaller tuple gets removed from the dictionary when the larger one is added. This abstraction ensures that the SRO implementation can handle arbitrary resource distributions with any combination of available correlation information.

4.4.2 Monte Carlo SRO

The central solver implemented so far is a Monte Carlo based solver traversing the full set of resource combinations. It works by sampling all entries in the `CorrelationDict` `n_samples` times. It then constructs all subsets of the problem resources with known correlation information, (i.e. the sets in the correlation dictionary as well as their combination) and does an exhaustive search on this set. For each subset, the chance constraint fulfillment is estimated from the samples and the expected total cost is calculated from the expected costs of

¹<https://github.com/jsager621/SRO.jl/>

²<https://julialang.org/>

the resources. In practice, this estimation process is very simple. We sample the sum distribution of a set of resources and calculate the ratio of samples bigger than v_{target} to the amount of samples.

We can formalize this in terms of a Monte Carlo estimator. Recall that F_N is a Monte Carlo estimator of the integral of an arbitrary function f using random variables $\{X_1, X_2, \dots, X_N\}$:

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{\text{PDF}(X_i)} \quad (4.5)$$

In the case of SRO, we are interested in approximating the chance constraint of the optimization:

$$1 - F_{X_s}(v_{target}) \geq p_{target} \quad (4.6)$$

Assume we want to do this for two stochastic resources with random variables V_1 and V_2 for their value generation. We approximate $1 - F_{X_{1+2}}(v_{target})$ to evaluate the chance constraint. We can set:

$$\text{ind}(x) = \begin{cases} 1 & \text{if } x \geq v_{target} \\ 0 & \text{else} \end{cases} \quad (4.7)$$

$$f(x) = f_{1+2}(x) \text{ind}(x) \quad (4.8)$$

We use the (known) multivariate distribution $f_{1,2}$ to directly sample V_{1+2} . Let $X_i \sim V_{1+2}$, then:

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{f_{1+2}} = \frac{1}{N} \sum_{i=1}^N \text{ind}(X_i) \quad (4.9)$$

From the properties of a Monte Carlo estimator, we arrive at:

$$\mathbb{E}(F_N) = \int_a^b f(x) dx = \mathbb{P}(V_{1+2} \geq v_{target}) = 1 - F_{X_{1+2}}(v_{target}) \quad (4.10)$$

The main challenge for this approach is the exponential number of sets to check. Consequently, any method to quickly eliminate infeasible subsets and speed up the chance constraint estimation drastically impacts the run time. The current implementation reduces the number of subsets in three ways. First, an initial pass of resources is done by their expected value generation to estimate the minimum number of resources necessary to achieve the target value. Subsets smaller than this cutoff are eliminated. Second, the estimation of the chance constraint is avoided for subsets with higher expected cost than the best feasible set so far. This is a simple sum calculation over a maximum of n resources, making it much faster than the chance constraint estimation. Lastly, if the resources generate only positive values, the Markov inequality (see Definition 25) is checked to remove subsets that can not fulfill the chance constraint.

This last check could be expanded using the Chebyshev or Chernoff inequalities as well. However, these checks require the sampled data to estimate their bounds which should not be much faster (if at all) than directly checking the chance constraint from the data.

Different SRO Solvers in this Thesis Most of the solvers discussed in the following chapters use the Monte Carlo method described above to estimate the SRO chance constraint (unless explicitly stated otherwise). Their difference are based on which subset of the power set of resources they consider for evaluation. Whenever we mention a **full** solver, this is an exhaustive search of the entire power set. Meanwhile, a greedy algorithm may just consider subsets in a specific order. A metaheuristic solver would instead have a certain logic for its selection of subsets to check.

The planned SRO evaluation paper [21] contains a small benchmark comparison of this solver against a decomposition-based approach (using the same solver on smaller subsets) and a greedy algorithm. The results of the comparison are shown in Figure 4.3. The decomposition-based method was significantly faster than the full solver but requires grouped correlation information to achieve a solution in two solver runs on smaller problems. This achieved comparable costs to the full solver but with higher failure magnitudes when the lossy correlation grouping produced bad estimates of the chance constraint. The greedy solver is the fastest with the trade-off of high-cost solutions only marginally better than simply selecting all resources to begin with.

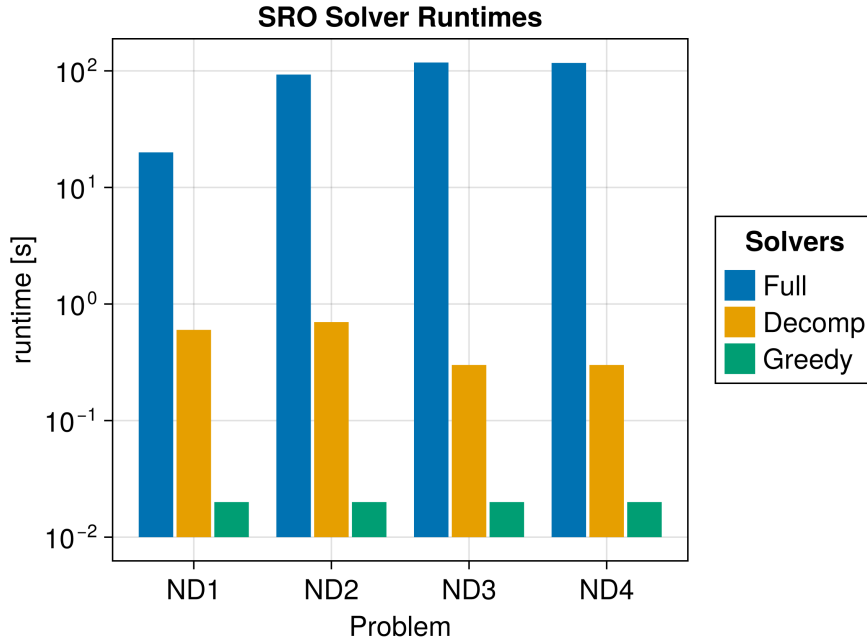


Figure 4.3: Runtime comparison of SRO solvers on problems with 21 resources. The full solver and greedy solver worked on the full resource sets in each problem while the decomposition solver had access to correlation groups to solve the problem in two smaller full solver passes.

4.5 Summary

This chapter introduces Stochastic Resource Optimization – the problem of selecting the best Stochastic Resources for a given target. We showed that SRO is closely related to stochastic knapsack problems and can essentially be considered a knapsack problem itself. SRO is generally difficult to solve, being NP-hard but can be solved efficiently under specific strict circumstances that allow sorting Stochastic Resources by their quality.

The biggest hurdle for SRO, as with most knapsack problems, is the explosion of possible combinations as the number of Stochastic Resources increases. Sums of sets under evaluation can be relatively efficiently approximated using Monte Carlo methods but reducing the amount of subsets to check for a solution is much more difficult. Additionally, accurate correlation models for each Stochastic Resource may not be available.

The following chapter addresses the quality of SRO solutions. First, it investigates problems with full correlation information – using metaheuristic algorithms to deal with the amount of combinations. Second, it shows a study of how solution quality degrades as correlation information becomes more scarce or inaccurate.

Chapter 5

SRO Solution Quality

This chapter describes a number of simulation-based experiments to evaluate the properties of SRO. As we have demonstrated in Chapter 3, sums of stochastic resources can vary depending on their correlation properties. Ideally, the full correlation information of all resources involved in a problem is known and accurately modeled. In reality, correlation information is difficult to model accurately. In many cases it is only available between some resources, or missing entirely.

Here, we evaluate SRO problems under differing levels of correlation information. The first section shows how SRO can be solved efficiently using simple meta-heuristic algorithms when all correlation information is known. Then we show that solution quality degrades greatly as correlation information becomes only partially available. Finally, we conduct a sensitivity analysis and show that SRO solutions are stable under small variations in the inputs.

5.1 SRO with Full Correlation Information

Let us start with the best case scenario for SRO. That is, problems with accurate and complete prediction models for all resources and their correlations. These results were published as part of [41] (own work).

As shown in Chapter 3, under the above assumptions, Stochastic Resources can be summed efficiently using a combination of discretization and fast Fourier transforms. All solvers in this section are based on Monte Carlo methods as described in Subsection 4.4.2. Our results show that simple metaheuristics like a Binary Particle Swarm Optimization (BPSO) [29] can achieve close to optimal results that significantly outperform the baseline heuristic – selecting all resources.

We also compare the original version of SRO where all generated value on selected resources must be bought to a variant where only value up to the target value is bought. These two variants are referred to as **buy-all** and **buy-necessary** in the following. Here, the results indicate that the different *market*

types do not change the relative performance of the investigated algorithms.

5.1.1 Methodology

The first aim of this experiment is to establish the suitability of the SRO formulation for energy system problems. We generate a number of energy resources to approximate different types of distributed energy resources. SRO problems based on these resources are then solved by different algorithms and the results are compared by their cumulative costs. The baseline for comparison is given by an oracle solver as the lower bound and simply selecting all available resources as the upper bound.

Cumulative cost is chosen as the metric for comparison here because all algorithms under investigation provide feasible SRO solutions close to the bounds of the chance constraint. Consequently, failure metrics such as failure rate or failure magnitude of the chance constraint would not yield much information. The remaining point of comparison is the selection of optimal resources to minimize the target function. Since all algorithms are compared on the same sets of problems, the cumulative cost and the mean costs achieved by the algorithms hold the same relative values.

The second aim is to establish if the market type influences the algorithm choice for SRO. The established formulation of SRO minimizes expected cost under the assumption that all generated values apply their cost (i.e. are bought or allocated). Alternatively, value might only be bought up to the target value, starting with the cheapest resource. Notably, this second formulation becomes trivial if the cost for 0 generation starts at 0 in all resources – simply selecting all resources yields the optimal solution. To account for the opportunity cost of planning in a given resource for the task, costs start at $c_{sel} > 0$, called the selection cost.

5.1.2 Problem Generation

SRO resources are generated based on Weibull, Beta, and Gaussian distributions to approximate wind generation, photovoltaic generation, and battery storage, respectively. Each distribution is scaled to a randomized mean $\mu \in [50, 150]$, standard deviation $\sigma \in [10, 50]$ and lastly truncated to the interval $v \in [0, 250]$. All resources are modeled with linear costs with a randomized cost factor $c \in [2, 5]$. The experiment considers two size tiers for problems with $n \in \{10, 30\}$ resources. The target values were set to $p_{target} = 0.7$ and $v_{target} = n \frac{v_{mean}}{2}$ where v_{mean} is the mean of means of generated marginals. Correlations between resources are assigned randomly using a Gaussian copula with a randomly generated correlation matrix. This means that both the marginal distributions and their correlations differ between problems.

For each distribution type (Weibull, Beta, Gaussian), each market type (buy-all, buy-necessary) and each number of resources (10, 30), we generate 100 SRO problems for a total of $3 \cdot 2 \cdot 2 = 12$ problem sets. Each problem set is randomly

instantiated (within the bounds outlined above) and solved 100 times to compare the cumulative differences between algorithms.

5.1.3 Compared Algorithms

The study compares four algorithms (detailed descriptions below):

- A full solver optimizing over the complete solution space.
- Taking a limited number of random samples for each subset size of the resource set.
- A binary particle swarm optimization (BPSO).
- A simple 1+1 evolutionary algorithm (EVO) with a probability to flip any particular resource of $p_{flip} \in \{0.3, 0.6, 0.8\}$.

These algorithms are selected to represent a simple heuristic (size sampling), exhaustive search, and metaheuristics (BPSO, Evolutionary Algorithm (EVO)). The expected outcome is that the simple heuristic is fastest but provides the highest cost results while the exhaustive search becomes prohibitively slow quickly but achieves optimal results. Ideally, the metaheuristics would be much faster than an exhaustive search while providing close to optimal costs.

The baseline for the comparison is given by two algorithms:

- An oracle that constructs optimal solutions by knowing the realized values of each resource.
- Selecting all possible resources every time.

Selecting all resources is chosen as the upper bound because it guarantees that the chance constraint holds if possible. The oracle solver should universally achieve lower cumulative costs than the other algorithms because it does not have to account for the chance constraint in its selection.

All non-oracle algorithms used a Monte-Carlo strategy with $n_{samples} = 1000$ samples to estimate the resource distributions. This value was chosen based on a pre-study evaluation of the run-times and estimation accuracy of the algorithms.

Full Solver (full) As the name implies, this solver checks every possible subset of resources and returns the feasible solution with the lowest expected cost. The algorithm is formally described in Subsection 4.4.2. This provides the optimal solution but becomes slow as the number of resources increases. This algorithm is only present in the simulated problems with 10 resources and not in the larger 30-resource problems where simulation times were prohibitively long on the available hardware.

Size-based Sampling (by_size) One question we would like to answer in our analysis is whether the amount of chosen resources is a stronger predictor of the final cost than which resources were selected in particular. To answer this, the size-based sampling algorithm ensures that each possible subset size of the solution space is randomly sampled. The best solution encountered in this search is returned. The amount of samples taken per size is set by the $n_{subsets}$ parameter. To ensure parity with the other algorithms, we set $n_{subsets} = n_{resources}$, so the algorithm takes a total of $n_{resources}^2$ samples.

Binary Particle Swarm Optimization (BPSO) In Particle Swarm Optimization (PSO) [24] a set of agents (particles) randomly explores the parameter space in search of an optimal solution. Each particle has a position x and a velocity v in the solution space. Each position has a corresponding solution quality (in this case, cost) evaluation. A number $n_{particles}$ of particles explore the solution space for n_{steps} time steps. In each cycle the particle moves, updates its velocity and evaluates its new position. The velocity update is influenced by the solutions encountered by other (or neighboring) particles. Regular PSO is defined for continuous parameter spaces. We use a variant of BPSO which works on discrete spaces by Lee et al. [29] with the following velocity and position update functions:

$$v(t+1) = wv(t) + a_1R_1(l_{best} - x) + a_2R_2(g_{best} - x) \quad (5.1)$$

$$x(t+1) = \begin{cases} 0 & \text{if } R_{pos} > S(v(t+1)) \\ 1 & \text{else} \end{cases} \quad (5.2)$$

Here, v is the particles' velocity at a given time step, x is its position, w is an inertia weight, a_1, a_2 are acceleration constants, R_1, R_2 are uniformly distributed random variables in $[0, 1]$, l_{best} and g_{best} are the best solution encountered by the particle and the best solution encountered by the whole swarm, respectively. R_{pos} is a uniformly distributed random variable in $[0, 1]$ and S is the sigmoid function:

$$S(v) = \frac{1}{1 + e^v} \quad (5.3)$$

In the simulations, we run BPSO with $n_{particles} = n_{runs} = n_{resources}$. These values were chosen because they provided quick results and good solution qualities on the evaluated problem sets during a pre-study evaluation. In both cases, we set $a_1 = a_2 = 1.43$ and $w = 0.69$ based on analysis from [29].

1+1 Evolutionary Algorithm (EVO) The 1+1 evolutionary algorithm works by randomly changing a candidate solution from the solution space and memorizing the best solution encountered until a termination condition is met. In our case, the algorithm has two parameters: n_{steps} sets the number of random samples taken until termination and p_{flip} gives the probability of changing the value of each element of the boolean selection vector (i.e. the currently evaluated element of the solution space). We set $n_{steps} = n_{resources}^2$ (so the amount

p_{target}	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
full	0.23	0.33	0.43	0.52	0.62	0.71	0.81	0.91
by_size	0.28	0.38	0.47	0.56	0.66	0.75	0.84	0.92
BPSO	0.26	0.36	0.45	0.55	0.64	0.73	0.82	0.92
evo	0.30	0.39	0.48	0.57	0.65	0.74	0.83	0.92

Table 5.1: Success rates of implemented algorithms compared to the target probability.

of samples is equivalent to the other algorithms) and compare three variants of this algorithm with $p_{flip} \in \{0.3, 0.6, 0.8\}$.

Oracle Solvers Oracle solvers operate on resources whose realized values (and associated costs) are known. The oracle solver varies between the two investigated market types. If all generated value has to be bought (buy-all variant), then the oracle decision problem becomes a regular 0-1 knapsack problem:

$$\min \sum_i x_i (c_i(v_i)) \quad (5.4)$$

$$\text{subject to } \sum_i x_i v_i \geq v_{target} \quad (5.5)$$

In the buy-necessary variant, we have to introduce a second stage to the optimization problem. Once a set of resources is selected, value has to be bought in order from lowest to highest cost. Since we restricted the stochastic resources to linear cost functions that are strictly larger than 0, this can be achieved by simply sorting them by their cost factor and taking generated value in this order.

5.1.4 Results

As an initial sanity check, Table 5.1 shows the achieved success rates of the approximation-based algorithms against the set target probability. The achieved success rates closely match the target probability in all cases, so all algorithms provide feasible SRO solutions.

The resulting cumulative costs for the problems with 10 resources are shown in Figure 5.1. The full set of result plots is attached in Appendix B. Of the fast algorithms, BPSO performs best across all scenarios. Size-based sampling and evolutionary algorithms with $p_{flip} \in \{0.3, 0.6\}$ show similar performance with slightly higher costs than BPSO. Notably, selecting all resources may achieve lower costs than other algorithms in the buy-necessary market if the higher selection cost is compensated by having resources with lower cost factors.

In the larger problem sets of 30 resources, the heuristic results closely resemble the results for smaller sets. Here, BPSO performs best with size-based

sampling, and the two evolutionary algorithms with $p_{flip} \in \{0.3, 0.6\}$ slightly worse.

In total, the results indicate that some SRO problems with distributions that are typical for energy system prediction models can be efficiently solved by metaheuristic algorithms. This is based on the assumption that the prediction models and their underlying correlations are accurately modeled. In the following sections, we investigate what happens when this assumption is relaxed.

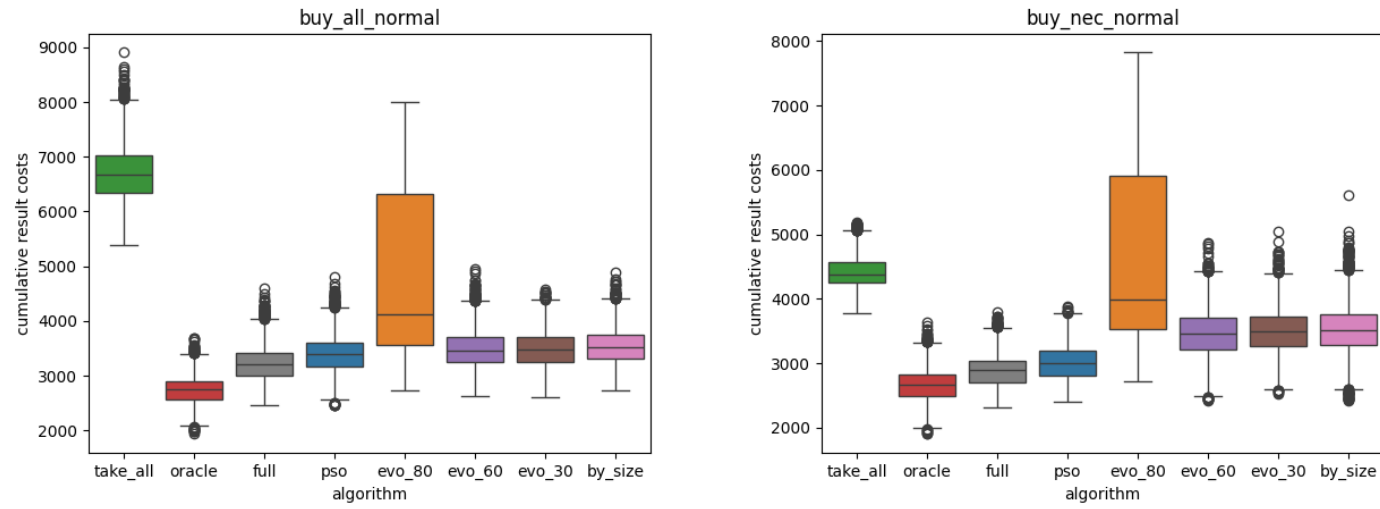


Figure 5.1: Cumulative costs for 10 resource scenarios with Gaussian distributed resources for buy-all problems (left) and buy-necessary problems (right) as published in [41].

5.2 SRO with Partial Correlation Information

Establishing complete or even partially correct correlation information is difficult to achieve, especially as the physical distance between energy resources increases. In this section, we investigate what happens to the failure rate and failure magnitude of an SRO problem under different levels of available correlation information. This study was done as part of the SRO evaluation paper currently in preparation for submission [21]. The results show that assuming independence between correlated resources can drastically increase the failure rate of SRO solutions.

5.2.1 Methodology

To determine the influence of correlation information, we compare four resource sets – three fully synthetic, one with correlations based on a study – with different target parameters under varying levels of correlation knowledge. The true multivariate distribution of the resources is known for evaluation. We then separate the resources into disjoint subsets of the same size. Correlation within a group remain known (from the multivariate distribution), but different groups are treated as independent by the solver. The randomization procedure for this is shown in Figure 5.2.

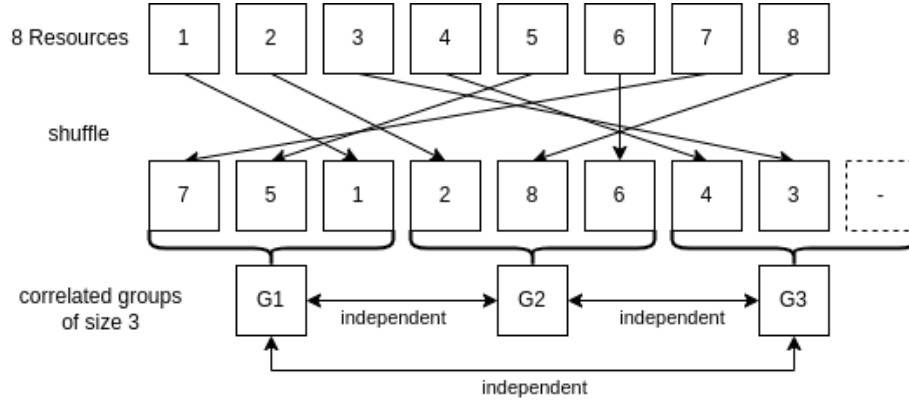


Figure 5.2: Reordering procedure with 8 resources and group size 3. The set of resources is randomly shuffled, then groups are assigned in shuffled order. If the group size does not exactly divide the number of resources, the last group contains the remainder. Correlation information within groups is preserved, but independence is assumed across groups.

In this setup, a subset size of 1 would mean all resources are treated as independent, subset size 2 means sets of 2 resources have known correlations, and so on. As the subset size increases to a maximum of $n_{resources}$, more correlation information is known to the solver.

The **evaluation criteria** for this experiment, grouped by subset size, are the following (detailed below):

- problem failure rates
- mean problem failure magnitude
- mean cost of successful subsets

We denote the probability of a set of resources R_{sol} to achieve the target value as:

$$P_{succ} = \mathbb{P}\left(\sum_{R_{sol}} v \geq v_t\right) \quad (5.6)$$

A problem solution counts as a **failure** if the resource set R_{sol} misses the target probability by more than half a percentage point:

$$P_{succ} < p_t - 0.005 \quad (5.7)$$

The problem **failure magnitude** M_F is the distance of the achieved success probability from the target probability:

$$M_F = \begin{cases} p_t - P_{succ}, & \text{if } P_{succ} < p_t. \\ 0, & \text{otherwise.} \end{cases} \quad (5.8)$$

5.2.2 Problem Sets

This analysis is conducted on four problem sets. The first three are constructed to capture different configurations of positive and negative correlations between sets of 12 resources. The fourth is a set of 12 North Sea wind parks with correlations modeled after the analysis of Hjelmeland et al. [15].

Dummy This problem set consists of 12 resources with 4 Gaussian, 4 Weibull, and 4 Beta distributions. The three resource groups have a correlation coefficient of 0.5 internally and are independent to the rest. This means a Gaussian resource is moderately correlated with other Gaussian resources but uncorrelated to the Weibull or Beta distributed resources. Resources are scaled to have the same expected generation and each group of resources has linear cost functions with the same ascending cost factors. The aim of this dummy set is to establish a baseline performance for moderate correlations.

Extreme Dummy and Positive/Negative Dummy Similar to the previous problem set, the extreme dummy set consists of 12 resources with groups of 4 being internally correlated with a correlation coefficient of 0.7. Here, all resources have a Gaussian distribution. Cost functions are assigned as in the previous set. The same linear cost functions with ascending cost factors are used within each set of 4 resources.

For the positive/negative dummy set, the same resources are combined with different correlation coefficients. The group-internal coefficient remains at 0.7, but now resources of different groups are negatively correlated with a coefficient of -0.3 .

North Sea Correlation The North Sea model is based on a study of the correlations of Norwegian wind parks [15]. 12 wind parks with a combined nominal power of 12.8 GW are modeled as resources with Beta distributed values with $\alpha = \beta = 2$. Resources are then scaled to the corresponding buildout size of the wind park. Since the Beta distribution is defined on $[0, 1]$, the scaled distribution is then defined on $[0, P_{max}]$ where P_{max} is the buildout size of the wind park in question. The chosen wind parks are listed in Table 5.2. The three largest wind parks in the study were not chosen for this set because their buildout is significantly larger than the others, so they would dominate the results.

The correlation factors from the study are transformed into a covariance matrix. We then use the covariance matrix to construct a Gaussian Copula to model the resource correlations. Costs for the wind parks are linear with cost factors from 2.0 up to 3.1 in ascending order of buildout size, making larger resources more expensive in this setup.

Wind Park/Region	buildout [GW]
Stadthavet	1.8
Olderveggen	0.3
Frøyagrunnene	0.2
Sandskallen - Sørøya nord	0.9
Vannøya Nordøst	0.5
Auvær	0.4
Nordmela	1.1
Gimsøy Nord	0.9
Træna fjorden - Selvær	0.7
Træna Vest	2.7
Nordøyan - Ytre Vikna	0.5
Frøyabanken	2.8

Table 5.2: North Sea wind parks with modeled correlations from [15] with their buildout in gigawatts.

SRO Problems The SRO problems based on the above resource sets are listed in Table 5.3. The target values and probabilities were chosen so that at least one feasible solution (and ideally more) exists in all cases.

Dummy			Extreme			Pos/Neg			North Sea		
pro	v_t	p_t	pro	v_t	p_t	pro	v_t	p_t	pro	$v_t[MW]$	p_t
D1	30	0.95	E1	30	0.80	P1	40	0.80	NC1	1000	0.95
D2	30	0.90	E2	35	0.75	P2	45	0.75	NC2	2000	0.85
D3	30	0.75	E3	40	0.70	P3	50	0.70	NC3	3000	0.75
D4	40	0.85	E4	45	0.65	P4	55	0.65	NC4	4000	0.65
D5	40	0.75							NC5	5000	0.55
D6	50	0.5									

Table 5.3: Targets for the dummy (left), extreme dummy (middle) and positive/negative dummy (right) problem sets.

5.2.3 Simulation Setup

This section describes the setup of the performed simulations in detail. To evaluate a given set of problems, each problem is solved multiple times. This is necessary to establish the distribution of solutions provided by the SRO solver as its sampling-based implementation does not yield deterministic results.

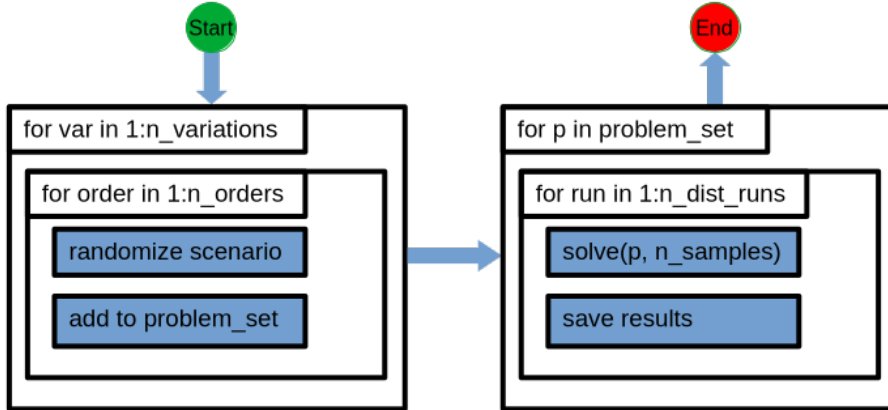


Figure 5.3: Simulation control flow for the correlation information analysis and sensitivity analysis. On the left side, SRO problem sets are generated by randomizing the order of the available resources and grouping them into subsets of different sizes (correlation information analysis). This step is repeated for the number of variations of targets, cost functions, and distributions in the sensitivity analysis. Once the problem sets are generated, they are passed to the right side where each problem set is solved `n_dist_runs` times to estimate the distribution of resources in the solution.

Figure 5.3 shows the main simulation loop for all experiments. First, a set of optimization problems is generated for the scenario. Here, we generate

`n_variations * n_orders` problems.

parameter	value
<code>n_resources</code>	{12, 21}
<code>n_samples</code>	1000
<code>n_dist_runs</code>	100
<code>n_orders</code>	10
<code>n_variations</code>	100

Table 5.4: Simulation Parameters. Values were chosen to cover non-trivial amounts of energy resources while keeping simulation times low. The limiting factor here is the sensitivity analysis, where the number of solver runs grows with `n_variations * n_dist_runs` and the time for each solver run grows with $2^{n_resources}$. With these values, the sensitivity simulation finishes in under 8 hours on an AMD Ryzen 7 5800X. Note that each additional resource doubles the execution time, so adding just 2 more resources would push the computation time above 24 hours and any significantly larger problem would run for multiple days or weeks.

The `n_orders` parameter controls how many permutations of the given set of resources are performed as described in Figure 5.2 to generate mutually independent sets for the **correlation analysis**. For the **sensitivity analysis**, this value is set to 1 as we are only interested in the full information cases in that experiment. Conversely, the `n_variations` parameter controls the number of slightly varied problem sets for the simulation and is only relevant for the sensitivity analysis.

Each problem in a problem set is solved `n_dist_runs` times to estimate the solution distribution provided by the Monte Carlo solver. The solver itself uses `n_samples` random samples to estimate resource subset quality in the solving process. The four scenarios are described in Subsection 5.2.2. All simulation parameters are listed in Table 5.4.

5.2.4 Results

Figure 5.4 shows the results of the failure rate analysis under different amounts of available correlation information. The figure is grouped by problem sets, namely the dummy (D), extreme dummy (E), extreme dummy with high positive and negative correlations (P) and North Sea Correlation (NC) problems.

The top of Figure 5.4 shows the failure rates of the different problem sets as correlation information increases. Constraint satisfaction is very low without full correlation information in the D, E, and N sets. Here, failure rates drop with increasing subset size. Only full correlation knowledge provides a high probability of constraint fulfillment. Even just assuming 2 resources to be independent (i.e. subset size 10) provides a significant rise in failure rate. Similarly, achieved costs in these 3 sets increase with increasing correlation information

as the solver has to select more resources to meet the chance constraint. Mean failure magnitudes with full correlation information are very low at about 1% in the D, P and NC scenarios and about 3% in the E scenario. This means that even when the solver fails to satisfy the chance constraint, it only misses it by a small margin on average.

The extreme scenario with groups of highly positively correlated resources that are negatively correlated to other groups is an outlier here. Assuming almost no correlation information (i.e. assuming independence) yields better results for failure rates than settings with more knowledge (except full knowledge). At the same time, the achieved costs are very consistent across knowledge levels. In the small amount of cases where the problems do fail, they tend to fail by very wide margins up to a mean of 0.41 for a knowledge subset size of 4.

There is a likely explanation for the low failure rates is that for this specific correlation setup: Because of the particular distributions and choice of target values in this scenario, assuming independence leads to very similar chance constraint estimates and low failure rates – the target values cluster around the center of the distribution, where the survival functions of the original and modified sum distributions intersect. At the same time, if the subsets with known correlation happen to exactly match the positively or negative correlated resource groups, the solver makes very wrong assumptions about the distributions resulting in large failures. This would explain why failure magnitudes are very high specifically on the even subset sizes of 2, 4, and 6 but low for a subset size of 5 as this can never exactly match the correlated resources.

Failure magnitudes generally decrease with increasing correlation information in all scenarios. Another interesting observation is that in the D scenario, failure magnitudes start low but then spike to a maximum of 0.28 at almost complete correlation information (subset size 10), then fall off completely to about 0.01 once full information is available. This means that for this scenario, having almost no correlation information leads to high problem failure rates but low failure magnitudes. As correlation information increases, the failure rate drops, but the remaining failure cases create much larger deviations.

The main takeaway of this experiment is that a lack of accurate correlation information can strongly impact the solution quality of SRO problems. In some cases, the fulfillment of the chance constraint is unaffected. In others, even small missing pieces of information lead to large failure magnitudes on the chance constraint.

5.3 Sensitivity Analysis

As part of the evaluation paper in preparation for submission [21], the same problem sets used in the correlation information analysis were used to conduct a sensitivity analysis to gauge how stable SRO solutions are under small changes in the input resources or target. This analysis is based on the same problem sets as the ones used in the correlation information section.

5.3.1 Methodology

For the sensitivity analysis, we compare the variation of the solution sets under small variations of the cost functions of resources, the target value of the problem and the random distribution of the resources.

To achieve this, we randomize the cost function, target value or resource distributions before each simulation run and solve the problem multiple times to estimate the solution set distribution. We then compare the TVD of the resource distributions achieved by the solver. As stated in Definition 22, the TVD for a discrete distribution for two probability mass functions P and Q is:

$$\delta_{TVD}(P, Q) = \frac{1}{2} \sum_x |P(x) - Q(x)| \quad (5.9)$$

For cost variations, the cost factor of the linear cost function $c(x) = m * x$ was randomly increased by up to 0.9 in steps of 0.1. The original cost functions have cost factors of $m \in \{2, 3, 4, 5\}$ and the randomization could at most set the cost to $m + 0.9$. For target variations, the target value of the problems was randomly increased or decreased by up to 10% of the original target value. The distribution variations were done for resources with Beta distributed resources with default parameters of $Beta(\alpha = 2, \beta = 2)$. The parameter $\beta \in [2, 5]$ is uniformly randomized. Then, the new distribution is normalized to have the same mean as the original. This skews the distribution toward the left.

5.3.2 Results

The following summarizes the results of the sensitivity analysis. The full results are listed in Table C.2 in the appendix. The mean and maximum values of the Total Variation Distance (TVD) for each problem set are shown in Figure 5.5. The main metric of comparison is the TVD between pairs of resource distributions. The table lists the maximum, mean, and median TVD for these pairs as well as the percentage of pairs where both resource distributions contain feasible solutions and the number of pairs with a TVD of zero (i.e. the exact same resource distribution). Problem sets are named by their origin: D{1-6} for dummy set problems, E{1-4} for extreme dummy problems, P{1-4} for extreme dummy problems with high positive and negative correlations and N{1-5} for variations on the North Sea model problems, used for the distribution variation.

For cost variations, the TVD is generally low, with a maximum value of 0.16 in problem set D3 which also has the highest mean value of 0.07. Notably, problems D1 and D4 show low solvability rates and high amounts of zero values for the solvable pairs. This indicates that there are generally few feasible solutions and many results select most of the resources as the solution candidate. A similar effect occurs in P1, P2 and P4 which are always solvable but mostly by selecting all available resources, resulting in a TVD of 0.

The target variations show a similar trend with low solution rates in D1 and D4 and a maximum TVD of 0.25 in D3. Problems generally contain more zeros than in the cost variations. This is likely due to the fact that target variations

change the feasible set of resources but cost variations do not. In the case of P1 and P4, the amount of zeroes actually decreases as randomly reducing the target value makes more resource sets feasible, although in P4 specifically, not all variations are solvable anymore. Additionally, the mean TVD for target variations is higher than to the corresponding problem set with cost variations in all cases. A possible explanation for this is that cost functions were only randomized within resource groups, but the cost order of resources remained unchanged. I.e. resource 1 would get cost factors between 2.0 and 2.9 and resource 2 between 3.0 and 3.9. Thus, a unit of value in resource 1 is always cheaper than in resource 2.

For the distribution variations, the mean TVD is consistently low across all problem sets with values between 0.03 and 0.06. The highest value occurs in NC1 with a maximum of 0.17. Notably, all problem pairs contain feasible solutions and no two resource distributions are identical. This indicates, that the amount of feasible solutions in these problem sets is larger than in the dummy scenarios as a result of the larger resource set.

5.4 Summary

In this chapter, we investigate how the solution of SRO problems changes under varying levels of correlation information.

When correlation information is complete and accurate, SRO can be efficiently solved using fast meta-heuristic algorithms. This is the case for the standard buy-all market type of SRO and the buy-necessary market type in the study. Of course, this is limited to sufficiently well-behaved problem instances as the ones constructed here. As shown in the previous chapter, there must exist problem instances that can not be solved this easily as SRO is strongly NP-hard.

However, even for these simple problem instances, solution quality degrades rapidly if correlation information becomes sparse or inaccurate. In problem instances containing 12 Stochastic Resources, missing correlation information of two resources leads to large increases in problem failure rates in three out of four constructed scenarios. A notable caveat to this is that the experiment assumes independence for missing correlation information. Depending on the underlying correlation structure, this can lead to reduced failure rates as one of the scenarios shows.

Lastly, SRO solutions are generally stable under small variations in the problem inputs. The sensitivity analysis shows that the Total Variation Distance of solution resource sets is smaller than 0.3 for the investigated cost, target, and distribution variations. This also means the implemented Monte Carlo-based SRO solver reliably produces equivalent solutions over multiple runs.

In the next chapter, we present two case studies on how an SRO formulation may be applied to energy system problems.

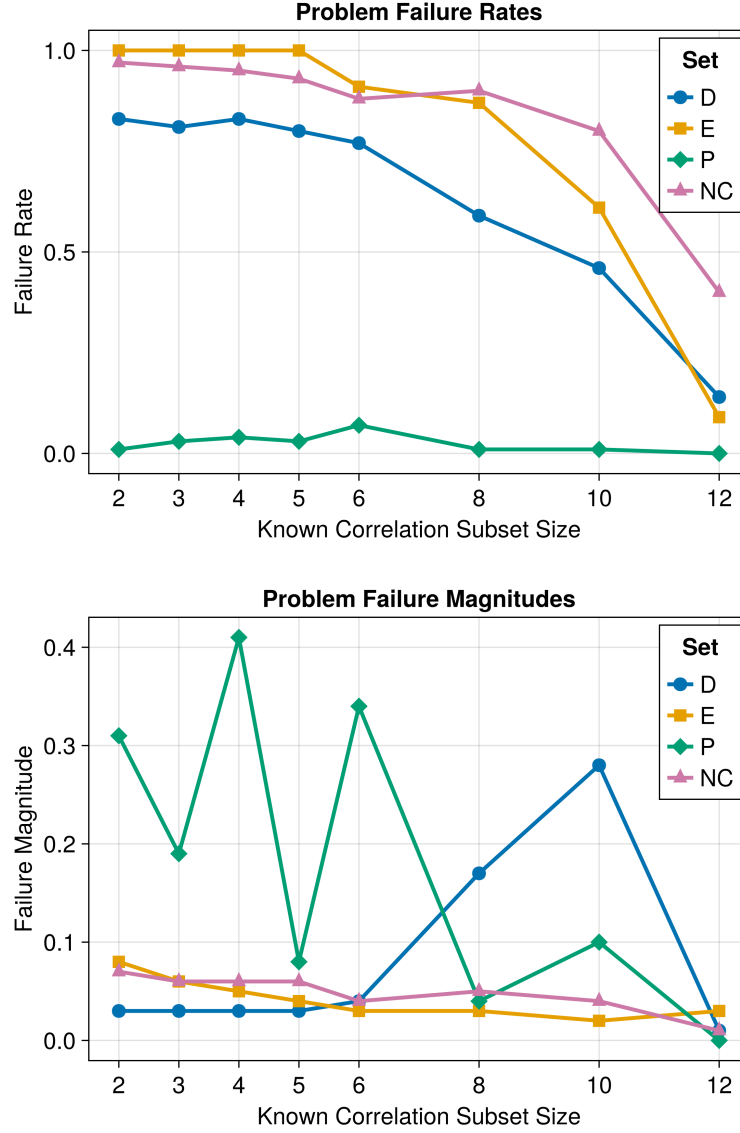


Figure 5.4: Results of the correlation information experiment: Failure rate and magnitude of different problem sets with increasing correlation information where a subset size of 12 indicates full information. The sets are: dummy (D), extreme dummy (E), extreme dummy with high positive and negative correlations (P) and North Sea correlation (NC). Lines between data points are purely for easier visual parsing and are not an indication of assumed values in other subset sizes.

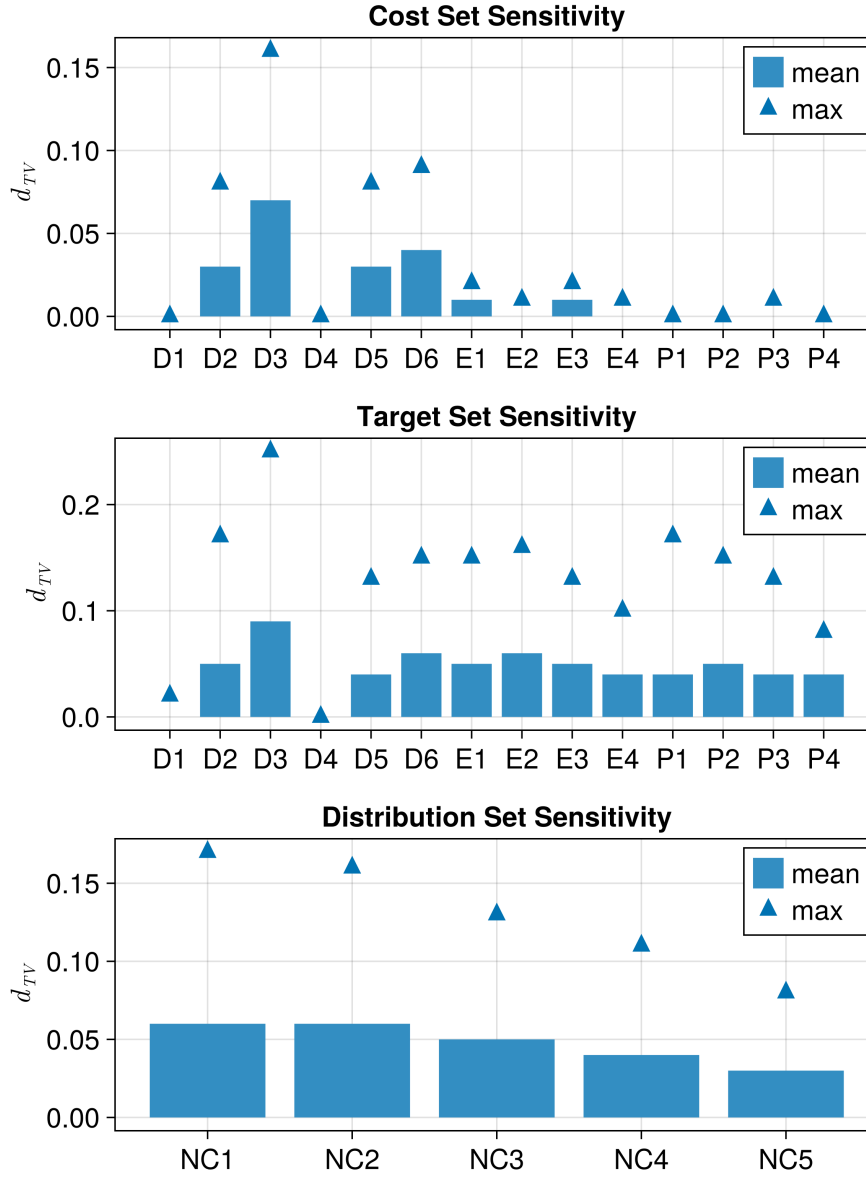


Figure 5.5: Mean and maximum TVD in the sensitivity analysis problem sets. Mean TVD is consistently below 0.1 and maximum values only exceed 0.2 for problem D3 in the target set.

Chapter 6

Case Studies

This chapter shows two case studies that apply SRO to energy domain problems. The first is an idealized congestion management scenario where SRO is used to select the ideal energy resources to counteract an induced grid congestion. This scenario assumes complete and accurate correlation information. The second case study investigates how SRO problem instances behave for a real-world correlation model on North Sea wind parks. Since this second study also includes correlation information on multiple levels of granularity, it also compares the performance of a decomposition-based solver for SRO for solution quality and solving time. As a side effect, these studies show that the implemented SRO structures and solvers are capable of incorporating different levels of correlation information.

6.1 Idealized Congestion Management

This case study was performed as part of [41] (own work). The aim is twofold. First, to establish that SRO solvers will select intuitively sensible resources where such intuition exists. Second, to apply SRO to a common energy system use-case (congestion management) with mixed distributed energy resources. The scenario assumes an accurate prediction model with full correlation information.

6.1.1 Scenario Grid and Idealized Day

The example power grid consists of a single residential power line with 13 load taps and a single transformer rated for a peak load of 150kVA. 12 of these load taps are occupied by households with energy resources (4 batteries, 4 photovoltaic units, 4 wind generators) that form the basis of the SRO problem. The final tap is an adjustable load used to induce congestions in the system at specific times. This grid mirrors a scaled down "Kerber Freileitung 1" grid

implemented in pandapower¹ and developed by Kerber [25]. Figure 6.1 shows a shortened representation of the grid with renewable generators and batteries on prosumer households along the line and the adjustable load (represented by a factory) at the end.

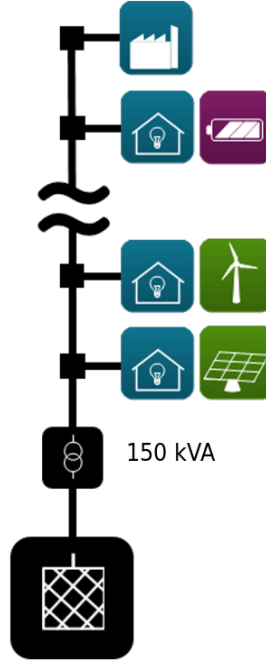


Figure 6.1: Congestion scenario grid. The power line contains 4 units each of wind generation, PV and battery storage and an adjustable load on the final node. All units are scaled to peak outputs of 6kW to 9kW to match the 150 kVA transformer. In the scenario, energy resources are used exclusively to counteract household loads exceeding 150 kW at the transformer.

The scenario is evaluated on a single idealized day. Household loads, shown in Figure 6.2, follow a generic curve with two peaks, one around noon and one in the evening. At four specific times (7:00, 12:00, 16:30, 20:00), congestions of a fixed size (115% load) are induced in the system. These times correspond to times of different combined availability of wind and PV generators as shown in Figure 6.3. The availability for each point in time is given as a relative output value $r \in [0, 1]$ that scales the realized power output of the resource. The day is modeled to be consistently sunny with a 90% PV output peak at noon and very windy in the morning with wind speeds falling off over time. Battery storage is used as the backup flexibility of the system and is set to a constant 80% possible output.

¹<https://pandapower.readthedocs.io/en/v2.4.0/networks/kerber.html>

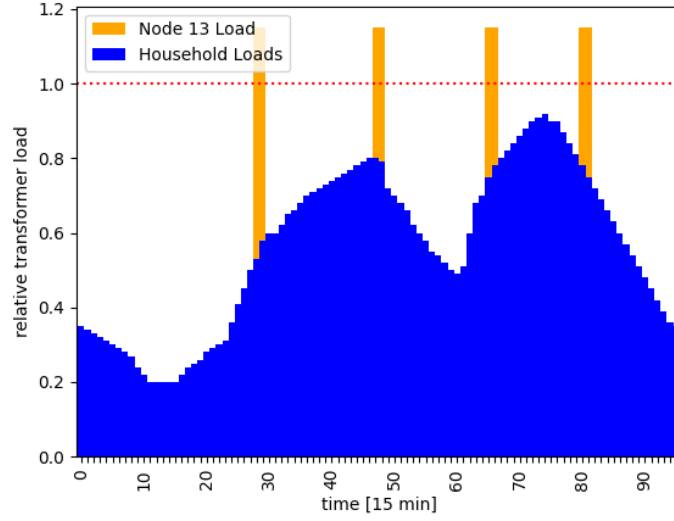


Figure 6.2: Idealized aggregated load curve from [41]. The blue bars are the aggregated base load of the scenario. The orange bars are artificially induced congestions at 4 time points during the day.

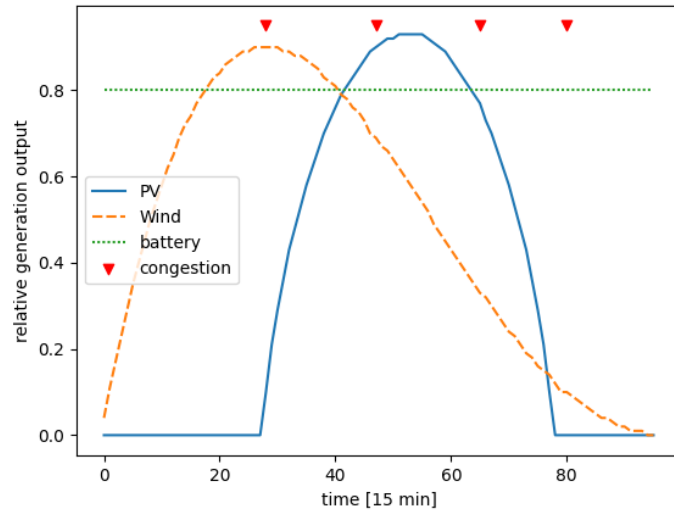


Figure 6.3: Relative outputs of the different types of generators used in the example scenario from [41].

Expected Result Hypothesis The aim of this scenario is to counteract congestions using renewable energy sources and batteries. For the purposes of the scenario, these resources are only used during the times of induced congestions and do not influence the net load of the grid otherwise. Given the relative availability and cost of each resource at the four congestion times, we have the following expectations:

- Wind and PV are preferred over batteries whenever possible (because their use is cheaper).
- The relative availability of PV and wind during different times of the scenario day is reflected in their selection at different times.

Type	$corr$	σ_{rel}	c_{sel}	c_{perw}	r_{p1}	r_{p2}	r_{p3}	r_{p4}
PV	0.8	0.1	50	0.1	0.1	0.93	0.77	0.1
Wind	0.6	0.2	100	0.1	0.9	0.7	0.35	0.1
Battery	0	0.05	300	0.3	0.8	0.8	0.8	0.8

Table 6.1: Parameters for generator instantiation on the congestion management system. The parameters are (in order): the correlation coefficient $corr$ between units of the same type, the standard deviation of the distribution of the generation σ_{rel} , the selection cost c_{sel} and cost per unit of power c_{perw} , and the relative output of the generator at the times of the four induced congestions r_{p1} to r_{p4} .

Resources and Problems Table 6.1 shows the chosen parameters to instantiate the generators. Each generator type has 4 units with maximum output powers of $P_{max} \in \{6kW, 7kW, 8kW, 9kW\}$. These resources are labeled $\{w, pv, b\}_1$ to $\{w, pv, b\}_4$ in order of these power values. Weather dependent units of the same type are correlated with a coefficient of 0.8 and 0.6 for PV and wind, respectively. These values are chosen to ensure very high correlation of PV generators and somewhat high correlation of wind generators, assumed from their co-location. Generation between different types as well as for all batteries is assumed to be independent.

The instantiated values are randomly selected from a truncated Gaussian distribution around $P_{max} \cdot r_p$ of the given unit with the given standard deviation σ_{rel} relative to this value. This plays the role of a prediction based on a forecast error in this scenario. PV and Wind generators have relative standard deviations of 10% and 20%, respectively. Battery outputs have lower relative standard deviation of 5% and are independent. The output levels follow the curves described in Figure 6.3 and have the given values for the four congestions induced in the day. Each resource has a linear cost function with selection cost $c(0) = c_{sel}$ and cost factor $c(x) = c_{sel} + v \cdot c_{perw}$.

All congestions create an SRO problem with $v_{target} = 22.5kW$ and $p_{target} = 0.8$. The target value matches the total overload at the time of congestion (which

is the same in all four induced congestions). The target probability is chosen to be reasonably high but not overly restrictive as target probabilities above 0.9 would require most of the available resources without showing the differences at different times of the day. Each problem is evaluated twice, once with the standard SRO formulation (buy-all) and once with a setup where only up to v_{target} of the generated value is actually bought (buy-necessary). The problems are labeled $p1$ to $p4$ in order on the time of day and are given the subscript a (buy-all) and n (buy-necessary) for their respective market type.

problem	res_{oracle}	c_{oracle}	res_{BPSO}	c_{BPSO}
$p1_a$	$w_{2,3,4}$	2562	$pv_{2,3}, w_{1,3,4}, b_1$	4393
$p1_n$	w_{2-4}	2550	pv_2, w_{2-4}, b_1	2900
$p2_a$	pv_{2-4}	2445	$pv_{1,2}, w_{1,3,4}$	3243
$p2_n$	pv_{2-4}	2400	$pv_{1,2,4}, w_3$	2500
$p3_a$	$pv_{2-4}, w_{1,2}$	2627	$pv_{1,3,4}, w_{1,3,4}$	2913
$p3_n$	pv_{1-4}	2450	pv_{1-3}, w_{2-4}	2700
$p4_a$	$pv_{1-4}, w_{1-4}, b_{1-3}$	7189	pv_1, b_{1-4}	8657
$p4_n$	$pv_{1-4}, w_{1-4}, b_{1-3}$	7124	$pv_{1-4}, w_3, b_{3,4}$	7236

Table 6.2: Simulation results for the four induced congestions on the example grid showing the selected resources by each solver and their total achieved costs. Resources are referenced by their type, w for wind, pv for PV and b for battery with their rated outputs $\{6, 7, 8, 9\}kW$ indexed with 1 to 4.

Results The four congestions were solved using an all-knowing oracle solver and a fast Binary Particle Swarm Optimization metaheuristic – the same algorithms described in 5.1. The results are listed in Table 6.2. The results show that both solvers tend to the renewable energy sources at times of high availability and only fall back to the more expensive batteries when strictly necessary. For example, in problems 1 to 3 at the times of high wind and high PV availability, respectively, the oracle solver uses only these renewable energy resources.

Figure 6.4 shows an illustration of the results for BPSO in the buy-all. Naturally, the metaheuristic solver has to select more resources in its solutions to account for the uncertainty of generation that is not an issue for the oracle. Consequently, it has to select one battery (the smallest) to fulfill the chance constraint in problem 1 but manages problem 2 and 3 purely on renewable resources. The increased amount of selected resources also means it achieves consistently higher costs, most notably much higher costs in $p1_a$.

In summary, the results indicate that SRO will choose the intuitively sensible renewable energy resources at times of high availability as outlined in the hypothesis. This makes it a suitable formulation as part of a congestion management scheme with uncertain energy resources. Of course, it can not provide a complete solution alone. For a full congestion management, the problem would need to be expanded to the operation planning of the resources at various time

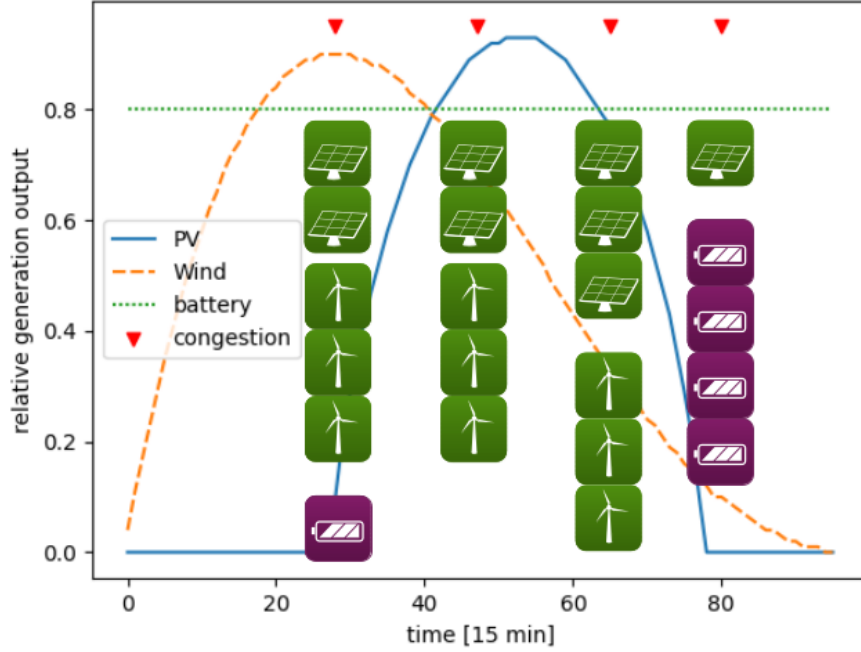


Figure 6.4: Representation of the resources selected by BPSO for the buy-all market. The image shows how many of each type of generator were chosen at each induced congestion together with the relative maximum output of each resource at the time.

points and taking into account all the relevant physical limitations.

6.2 Correlated North Sea Scenario

The contents of this section are in preparation for submission [21]. To study the applicability of the SRO model to a real world scenario with spatially distant resources, we created a scenario of North Sea wind parks based on [15]. The paper provides a detailed correlation analysis of Norwegian wind parks and correlation coefficients for North Sea regions in Germany, Denmark, the Netherlands, and the UK. We use these to test the performance of a Monte-Carlo-based full SRO solver in comparison to a greedy solver and a decomposition-based approach.

6.2.1 Methodology

The aim of this case study is to determine how the use of aggregated correlated regions affects the solution quality. To achieve this, we run the same sets of problems on both correlation resolutions, then compare the problem failure rate

of the resource sets. Importantly, the failure rates are determined by the fine-grained correlation values in both cases.

The analysis compares 3 algorithms:

- a full solver based on Monte-Carlo sampling
- a greedy algorithm based on the mean cost of each resource
- a decomposition solver

The **full solver** checks all possible resource combinations. This is the procedure described in the Monte Carlo SRO subsection 4.4.2.

The **greedy solver** sorts resources by their expected cost and only considers subsets in this order. Other than that, it works exactly like the full solver. To compare, we run the greedy solver both in ascending and descending expected cost order. The expected result is that the greedy solver in ascending cost order achieves lower average costs than in descending order and both achieve much higher costs than the full solver.

Lastly, the **decomposition solver** uses the fact that we have correlation information for wind power regions as well as the underlying wind parks. We use this to break the problem down into two smaller runs of the full solver. First, we solve for the relevant set of regions to achieve a feasible solution and then run full solver only on the lower level subset of wind parks of these regions. An example of a decomposition solver run is shown in Figure 6.5. The procedure breaks a single SRO solver run with 21 resources down into two solver runs with 6 and 5 resources, respectively, in this case. This achieves a drastic reduction in computation time as the solving time grows exponentially in the number of resources.

The algorithms are compared by:

- the problem failure rate
- the mean failure magnitude
- the mean cost of solutions
- the run-time of the algorithm

6.2.2 Problem Set

The problem set consists of all Norwegian wind parks listed in [15] except for the 3 wind parks around Stadt (Stadthavet, Olderveggen, Frøyagrunnene) as these were not mapped to one of the coarser correlation regions and thus lack the higher granularity correlation view for this case study. Additionally, it models wind power regions in the Netherlands (NL), United Kingdom (UK), Denmark (DK) and Germany (DE) with the correlations provided for these regions listed in [15]. The full set of wind parks is shown in Table 6.3. From these, we construct four SRO problems listed in Table 6.4.

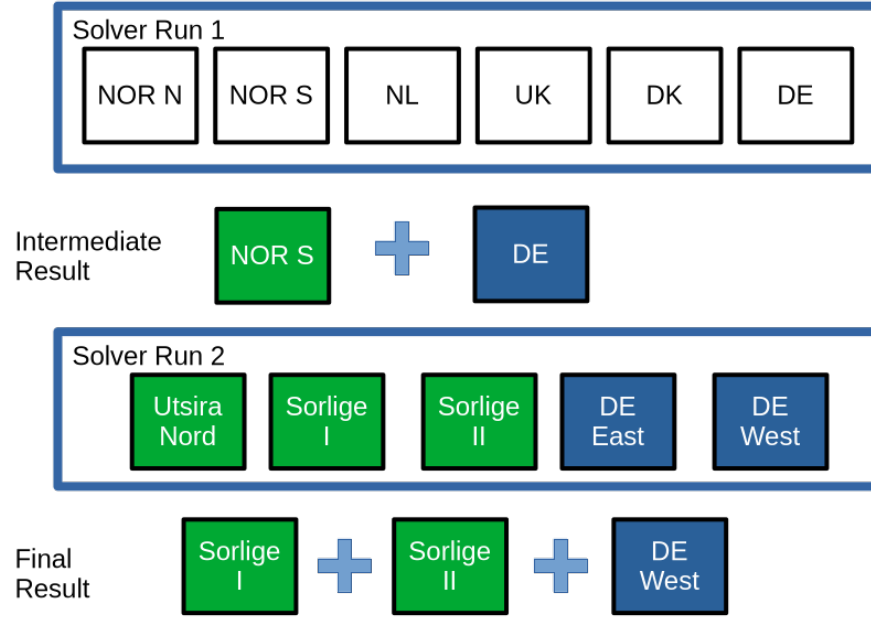


Figure 6.5: Decomposition Solver. Problems are solved in two steps. First the input resources are the high level regions. The resulting group of regions is then expanded into their constituent lower level resources which form the basis of the second solver run. The result of this second run is the final output of the solver. In this example, the computation consists of solving one SRO problem with 6 resources and one problem with 5 resources. This has a much lower expected computation time than directly solving for all 21 resources as the computation time grows exponentially in the number of resources. Each individual solver run uses the full solver.

The aim of this problem set is to determine the effectiveness of solving SRO as a two-stage problem, selecting regions first, then only solving the detailed model for the selected regions. If any region can be ruled out by such a decomposition solver it would significantly reduce computation times. Costs in this setup are linear and uniform across all resources because we are only interested in the effect of the decomposition approach on failure rates.

6.2.3 Results

Table 6.5 shows the results of the solver comparison and Figure 6.6 shows the average run times and costs of the used solvers. In three of the four problems, the decomposition solver achieves slightly lower costs and failure rates but has much higher failure magnitudes if the solution fails. This may seem like a contradiction at first as the full solver searches the full resource power set and

Wind Park/Region	buildout [GW]	source	region
Sandskallen - Sørøya nord	0.9	[15]	NOR N
Vannøya Nordøst	0.5	[15]	NOR N
Auvær	0.4	[15]	NOR N
Nordmela	1.1	[15]	NOR N
Gimsøy Nord	0.9	[15]	NOR N
Træna fjorden - Selvær	0.7	[15]	NOR N
Træna Vest	2.7	[15]	NOR N
Nordøyan - Ytre Vikna	0.5	[15]	NOR N
Frøyabanken	2.8	[15]	NOR N
Utsira Nord	3.5	[15]	NOR S
Sorlige Nordsjø I	4.8	[15]	NOR S
Sorlige Nordsjø II	9.0	[15]	NOR S
NL north	2.35	[31]	NL
NL west	2.35	[31]	NL
UK north	5.3	[9]	UK
UK west	5.3	[9]	UK
UK east	5.3	[9]	UK
DK east	1.35	[36]	DK
DK west	1.35	[36]	DK
DE east	1.8	[12]	DE
DE west	7.4	[12]	DE

Table 6.3: North Sea and Baltic wind parks with modeled correlations from [15] with their assumed buildout in gigawatts. Estimates for other wind park areas are from the listed sources. In the case of NL, UK, and DK sets, the total estimated buildout was equally split into the regions. The DE report gives numbers for both Baltic (DE east) and North Sea (DE west) buildout. Each wind park is part of a region with coarser correlation information being available between regions.

p	$v_t[MW]$	p_t
ND1	10000	0.90
ND2	15000	0.80
ND3	20000	0.75
ND4	25000	0.60

Table 6.4: North Sea Decomposition (ND) scenarios. Each scenario is encoded as an SRO problem p with a target value v_t and target probability p_t . Each problem uses the same stochastic resources – the different North Sea wind parks of the model.

should provide the optimal solution in all cases. However, in the evaluation,

we only consider the cost of problem runs that produced feasible solutions to the problem within the failure rate bound. This bound was set to a maximum threshold of 0.05. This means that in practice, the full solver can (and does) have higher failure rates (but lower failure magnitudes) than the decomposition solver and the mean costs compare different problem instances. Consequently, the cost comparison between these two solvers is of limited use. The issue is not present in the greedy solver variants as these almost always fulfill the chance constraint.

For all solving methods, the runtime is strongly influenced by the amount of solution candidates that can be dismissed without evaluating the chance constraint. For the decomposition solver, runtime additionally depends heavily on the chosen regions as this influences the amount of resources for the final solver run. In all cases, decomposition leads to at least a 4 times speedup compared to directly solving for all resources. The failure magnitudes for the greedy and full solvers are negligible. The decomposition solver produces some larger failures (up to a mean of 9% for ND2) due to the information discrepancy between the decomposition and detailed correlation models.

p	solver	failure rate	$\overline{mag}$	$\overline{cost}$	runtime [s]	
ND1	dec	0.39	0.03	31975	0.6	± 1.6
	full	0.54	0.01	30903	20	± 9
	grd	0.0	-	33322	0.02	$\pm 10^{-4}$
	grd ⁻¹	0.0	0.004	39509	-	-
ND2	dec	0.36	0.09	38746	0.7	± 1
	full	0.62	0.02	39697	93	± 6
	grd	0.0	-	43925	0.02	$\pm 10^{-4}$
	grd ⁻¹	0.0	-	44297	-	-
ND3	dec	0.42	0.08	49062	0.3	± 0.4
	full	0.47	0.01	50463	118	± 3
	grd	0.0	-	53218	0.02	± 0.05
	grd ⁻¹	0.05	0.02	53041	-	-
ND4	dec	0.36	0.05	52276	0.3	± 0.4
	full	0.51	0.01	54236	117	± 3
	grd	0.0	-	60336	0.02	$\pm 10^{-4}$
	grd ⁻¹	0.11	0.01	55437	-	-

Table 6.5: Comparison of results for full SRO solver, decomposition solver and greedy solver on the 21 resource North Sea setup.

In contrast, the greedy algorithm is two to three orders of magnitude faster and significantly overachieves the chance constraint for all problems, leading to a failure rate of 0.0. The achieved success rates exceed the target probabilities by $\{2, 6, 3, 10\}$ percentage points for problems 1 to 4, respectively. This comes at the trade-off of significantly higher costs in the achieved solutions.

An analysis of the solution distributions shows that the greedy algorithm selects significantly more resources than the other two algorithms, using all but 1 resource for problem 3 and always using all resources for problem 4.

The apparent reason for this is twofold: First, all resources in the set use the same cost function. Thus, the expected cost of a resource is directly linked to its size, so the greedy algorithm starts with the smallest wind parks and moves up in size. Second, the greedy algorithm completely ignores correlations between the resources for the sorting. Running the same solver with a reversed sort order (grd^{-1}) shows similar cost results for problems 2 to 4, uses fewer resources, and occasionally fails to fulfill the chance constraint.

6.3 Summary

This chapter presented two case studies for SRO in energy domain problems.

In the first, we show that SRO produces intuitively sensible selections of distributed energy resources for a congestion management scenario. PV and wind generation are preferred by the solver at times of high generation and fallback to more expensive units (modeled as battery storage units here) only occurs when necessary.

The second uses SRO to select North Sea wind parks for different target values based on a real world study of their correlation properties. The results show that this correlation information can be easily incorporated into the implemented SRO structure and solver.

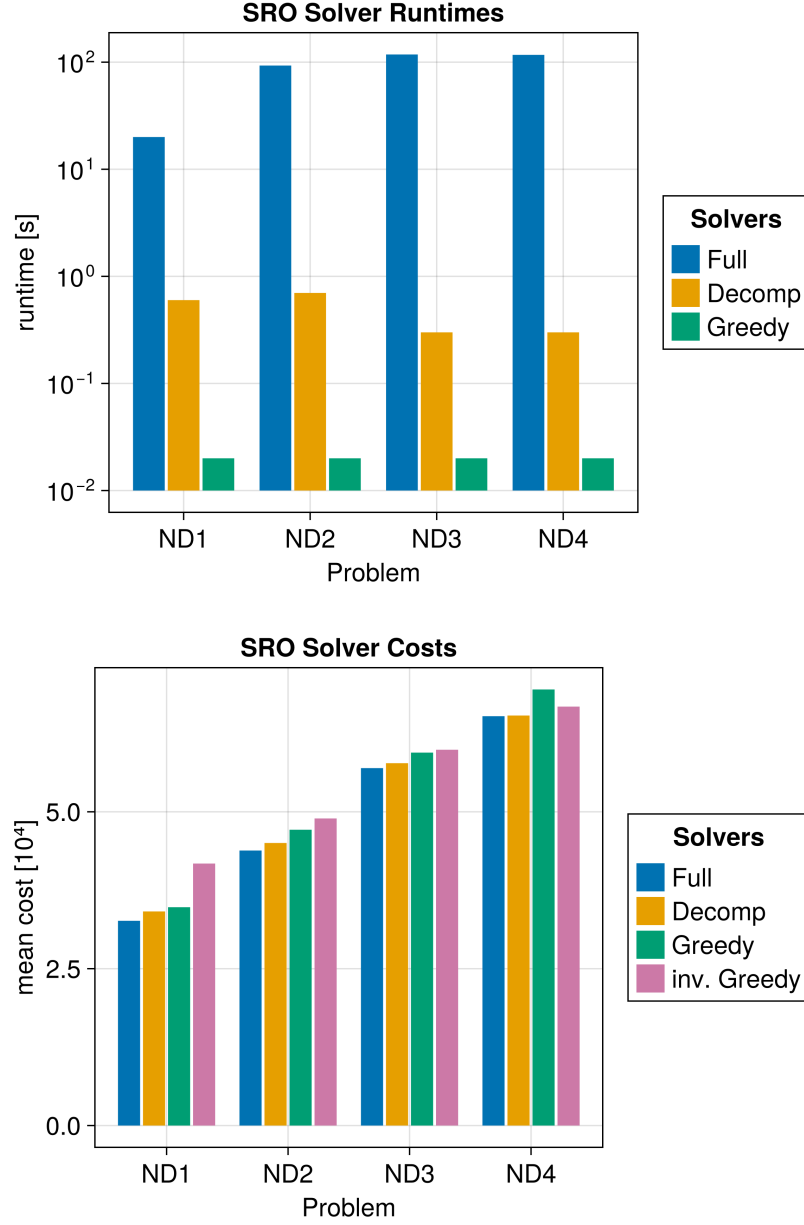


Figure 6.6: Run time (upper figure) and cost comparison (lower figure) of full, decomposition, and greedy solver on the North Sea decomposition scenarios ND1 – ND4. These were chosen because they are the problems with the largest amount of resources in this work. The inverted greedy solver time (upper figure) is not shown because it is by design the exact same as the regular greedy solver.

Chapter 7

Future Work and Conclusion

This chapter is split into two main parts. First, we give a summary of this thesis and evaluate the achievement of the stated goals in the introduction. Second, we outline some possible extensions and lines of research for SRO to be conducted in future work.

7.1 Summary

The main research question was how short term chance-constrained economic decisions can be made for widely distributed energy resources based on arbitrary forecast methods, considering their correlations and generation costs. This question was split into three sub-questions regarding the incorporation of widely distributed energy resources into the process (S-RQ1), decisions under full correlation information (S-RQ2) and decisions under partial correlation information (S-RQ3).

Model Applicability (S-RQ1, S-RQ2) The developed SRO model is compatible with any energy resource forecast that can be modeled as a random variable. On the implementation side, this is even more general and can technically admit any stochastic process as an input. As such, it can incorporate widely distributed energy resources based on arbitrary forecast methods. We have shown that Stochastic Resources can be used as a model in a small scale congestion management scenario as well as a large scale planning problem for North Sea wind parks.

Complete and Partial Correlation Information (S-RQ2, S-RQ3) The correlation dictionary struct enables the implemented software to incorporate any sets of correlated information – be they complete or partial. SRO solutions

are stable under small variations but absolutely require accurate and complete correlation information to provide reliable solutions. Under these conditions, SRO can provide a quick estimation of which resources to commit for a given task before forwarding the results to an economic dispatch solver for the detailed planning of each time step.

Solving Time (NFR 2) One of the non-functional requirements of the thesis was the efficient solvability of SRO instances. Our theoretical analysis shows that summation for discrete stochastic resources can be relatively fast. We also provide an avenue for solutions for continuous resources through Monte Carlo methods. This works both for sets of independent and correlated resources. The Stochastic Resource Optimization problem remains intractable in the general case for large sets of resources as the combinatorial factor dominates the runtime. A combination of fast (meta)heuristic solvers, decomposition, and slow full solvers can be incorporated into a usable any-time optimization scheme. Additionally, sets of resources that exhibit a domination order can be exhaustively solved in polynomial time.

In summary, this thesis provides answers to all the research questions outlined in the introduction. Further analysis for efficient solutions should be conducted if the SRO model is adopted and extended in the future. Some of these avenues of research are outlined in the following section.

7.2 Future Work

This section gives a brief overview of some extensions and further analyses that could be conducted for SRO in the future.

7.2.1 Theoretical Analysis

The theoretical analysis within this thesis is limited to some computational bounds for the summation of stochastic resources and a preliminary analysis of the complexity class of the optimization problem. There are two main points of interest for future analysis.

First is the investigation of relevant subclasses of stochastic resources for different applications. SRO is hard to solve in general, but many applications have quite limited probability distributions and simple cost functions. It would be worthwhile to isolate fragments of SRO, e.g. with linear cost functions and symmetrical (multivariate) distributions. These properties may be used to drastically reduce the complexity of the optimization problem and achieve optimal solutions for large sets of resources. Most variants of knapsack problems admit pseudo-polynomial algorithms or polynomial time approximation schemes under the right restrictions. Thus, a relevant research question for future work in the field of theoretical analysis would be: Are there subsets of SRO for which this is possible as well while still providing a useful model?

Second is further analysis of Monte Carlo based solvers for SRO. In particular, practical guidelines for bounds on the number of requirement samples for such solvers depending on the number of resources and their probability distributions. The matrix of random values is the single largest data structure and a key factor in the SRO solver. Linked with this is further development of fast, high quality (not necessarily optimal) solvers for SRO. So far we have only looked at full solvers, decomposition methods, and simple metaheuristic solvers. There is a large body of work on metaheuristics for combinatorial problems that is yet untapped for SRO. We would expect different algorithms to have different strength for varying categories of SRO problems. Thus, a resulting research question could be which metaheuristics methods are best suited for different subcategories of SRO problems.

7.2.2 Model Adaptations

The two main shortcomings of the SRO model have already been discussed in Chapter 3. They are the time point nature of stochastic resources and the limitation on cost functions to only take the value of the resource into account for which they are defined. For practical applications, the latter could be adapted to incorporate multivariate cost functions that consider the realizations of multiple (or all) resources in the set. This would, for example, allow modeling global energy prices dependent on the total generation in a neighborhood. One complication this would bring is that subsets investigated during optimization have to take into account the realizations of resources outside the subset for their expected costs. For the current implementation of the SRO solver, this would however only be a minor inconvenience as the sample data for all resources is present anyway.

The time point nature of stochastic resources is harder to address. One possible approach would be to allow time series prediction data as the value generator of a Stochastic Resource to directly map existing forecast methods. Without further study, it is unclear what the benefits of this would be over using separate sets of resources for each time step. In any case, it would considerably complicate the summation and optimization processes.

7.3 Conclusion

This thesis introduced the Stochastic Resource model and the subsequent optimization problem for the commitment of these resources. It is an attempt at a high-level commitment planning formalism for distributed energy resources in the absence of underlying high fidelity physical models. This is both a strength and a limitation. It enables SRO to aid decision-making on highly distributed energy resources based on their correlations and local forecasts. It also means that SRO alone can not consider possible grid constraints for moving power between these resources.

Ultimately, SRO can not (and is not meant to) remove the need for classical

unit commitment and economic dispatch planning based on all relevant physical details. Its main value lies in broader tasks like questions on total generation for larger regions. In this manner it can also assist in portfolio planning decisions (e.g. where to build or acquire a PV plant or wind park).

The main challenge for the application of SRO in practice is the availability of accurate correlation and prediction models. As our analysis has shown, even a prediction model that is perfectly accurate locally is of little use, when the correlations between energy resources are not adequately accounted for. One escape from this would be to shift towards a robust (instead of a stochastic) optimization approach which ultimately shifts the blame for inaccurate models to whoever signs off on the considered scenarios. This typically leads to safe results at the cost of less exploitation of the optimization space. Our prediction is that as forecast methods and correlation models improve in quality (as they undoubtedly will), stochastic optimization methods will be the preferable choice for stochastic problem-solving.

Bibliography

- [1] Mani K Agrawal and Salah E Elmaghraby. On computing the distribution function of the sum of independent random variables. *Computers & Operations Research*, 28(5):473–483, 2001.
- [2] Ioannis K Bazionis and Pavlos S Georgilakis. Review of deterministic and probabilistic wind power forecasting: Models, methods, and future research. *Electricity*, 2(1):13–47, 2021.
- [3] Joseph K Blitzstein and Jessica Hwang. *Introduction to probability*. Chapman and Hall/CRC, 2019.
- [4] Jiaxin Cao, Bo Yang, Shanying Zhu, Chao Ning, and Xinpeng Guan. Day-ahead chance-constrained energy management of energy hubs: a distributionally robust approach. *CSEE Journal of Power and Energy Systems*, 8(3):812–825, 2021.
- [5] Promphak Dawan, Kobsak Sriprapha, Songkiate Kittisontirak, Terapong Boonraksa, Nitikorn Junhuathon, Wisut Titiroongruang, and Surasak Niemcharoen. Comparison of power output forecasting on the photovoltaic system using adaptive neuro-fuzzy inference systems and particle swarm optimization-artificial neural network model. *Energies*, 13(2):351, 2020.
- [6] Brian C Dean, Michel X Goemans, and Jan Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Mathematics of Operations Research*, 33(4):945–964, 2008.
- [7] Amol Deshpande, Lisa Hellerstein, and Devorah Kletenik. Approximation algorithms for stochastic submodular set cover with applications to boolean function evaluation and min-knapsack. *ACM Transactions on Algorithms (TALG)*, 12(3):1–28, 2016.
- [8] Tao Ding, Qingrun Yang, Xiyuan Liu, Can Huang, Yongheng Yang, Min Wang, and Frede Blaabjerg. Duality-free decomposition based data-driven stochastic security-constrained unit commitment. *IEEE Transactions on Sustainable Energy*, 10(1):82–93, 2018.

- [9] Crown Estate. Uk offshore wind report 2024. *The Crown Estate: London*, 2024. <https://www.datocms-assets.com/136653/1747814298-osw-report-2024.pdf>, visited 2025-09-04.
- [10] Ricardo Fernández-Blanco, Yury Dvorkin, and Miguel A Ortega-Vazquez. Probabilistic security-constrained unit commitment with generation and transmission contingencies. *IEEE Transactions on Power Systems*, 32(1):228–239, 2016.
- [11] Alison L Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International statistical review*, 70(3):419–435, 2002.
- [12] Deutsche WindGuard GmbH. Status des offshore-windenergieausbaus in deutschland, 2024. https://www.wind-energie.de/fileadmin/redaktion/dokumente/publikationen-oeffentlich/themen/06-zahlen-und-fakten/20250204_Status_des_Offshore_Windenergieausbaus_Jahr_2024.pdf visited 2025-09-04.
- [13] Bram L Gorissen, İhsan Yanıkoğlu, and Dick Den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015.
- [14] Ge Guo, Luckny Zephyr, José Morillo, Zongjie Wang, and C Lindsay Anderson. Chance constrained unit commitment approximation under stochastic wind energy. *Computers & Operations Research*, 134:105398, 2021.
- [15] Martin Hjelmeland and Jonas Kristiansen Nøland. Correlation challenges for north sea offshore wind power: a norwegian case study. *Scientific Reports*, 13(1):18670, 2023.
- [16] Bri-Mathias Hodge, Debra Lew, and M Milligan. Short-term load forecasting error distributions and implications for renewable integration studies. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2013.
- [17] Bri-Mathias Hodge, Debra Lew, Michael Milligan, Emilio Gómez-Lázaro, Xiaoli Guo Larsén, Gregor Giebel, Hannele Holttinen, Samuli Sillanpää, Richard Scharff, Lennart Söder, et al. Wind power forecasting error distributions: An international comparison. In *11th International Workshop on Large-Scale Integration of Wind Power into Power Systems as well as on Transmission Networks for Offshore Wind Power Plants*, 2012.
- [18] Da Huo, Chenghong Gu, David Greenwood, Zhaoyu Wang, Pengfei Zhao, and Jianwei Li. Chance-constrained optimization for integrated local energy systems operation considering correlated wind generation. *International Journal of Electrical Power & Energy Systems*, 132:107153, 2021.
- [19] Da Huo, Chenghong Gu, Kang Ma, Wei Wei, Yue Xiang, and Simon Le Blond. Chance-constrained optimization for multienergy hub systems in a smart city. *IEEE Transactions on Industrial Electronics*, 66(2):1402–1412, 2018.

- [20] Kelachukwu J Iheanetu. Solar photovoltaic power forecasting: A review. *Sustainability*, 14(24):17005, 2022.
- [21] Sager Jens and Astrid Nieße. The impact of accurate correlation information on the solution quality and sensitivity of stochastic resource optimization - submission in preparation. unpublished, unpublished.
- [22] Malvin H Kalos and Paula A Whitlock. *Monte carlo methods*. John Wiley & Sons, 2009.
- [23] Theo Karaboghossian and Michele Zito. Easy knapsacks and the complexity of energy allocation problems in the smart grid. *Optimization Letters*, 12(7):1553–1568, 2018.
- [24] James Kennedy and Russell Eberhart. Particle swarm optimization (pso). In *Proc. IEEE international conference on neural networks, Perth, Australia*, volume 4, pages 1942–1948, 1995.
- [25] Georg Kerber and Rolf Witzmann. Statistische analyse von ns-verteilnetzen und modellierung von referenznetzen. *EW-Magazin für die Energiewirtschaft*, pages 22–26, 2008.
- [26] Anton J Kleywegt and Jason D Papastavrou. The dynamic and stochastic knapsack problem. *Operations research*, 46(1):17–35, 1998.
- [27] VY Kondaiah, B Saravanan, P Sanjeevikumar, and Baseem Khan. A review on short-term load forecasting models for micro-grid application. *The Journal of Engineering*, 2022(7):665–689, 2022.
- [28] Bernhard Korte and Jens Vygen. *Combinatorial optimization: theory and algorithms*. Springer, 2008.
- [29] Sangwook Lee, Sangmoon Soak, Sanghoun Oh, Witold Pedrycz, and Moongu Jeon. Modified binary particle swarm optimization. *Progress in Natural Science*, 18(9):1161–1166, 2008.
- [30] Miles Lubin, Yury Dvorkin, and Scott Backhaus. A robust approach to chance constrained optimal power flow with renewable generation. *IEEE Transactions on Power Systems*, 31(5):3840–3849, 2015.
- [31] T Macquart, D Kucukbahar, and B Prinsen. Dutch offshore wind market report. *Netherlands Enterprise Agency (RVO): Utrecht, The Netherlands*, 2023. <https://www.rvo.nl/sites/default/files/2023-05/Offshore-Wind-Market-Report.pdf>, visited 2025-09-04.
- [32] Afshin Najafi-Ghalelou, Mohsen Khorasany, and Reza Razzaghi. Risk-constrained scheduling of energy hubs: A stochastic p -robust optimization approach. *IEEE Systems Journal*, 16(4):5787–5798, 2022.
- [33] Roger B Nelsen. *An introduction to copulas*. Springer, 2006.

- [34] S Nethravathi and Venkatakirthiga Murali. A novel knapsack algorithm-based energy routing in a microgrid. *Distributed Generation & Alternative Energy Journal*, pages 641–668, 2023.
- [35] Ali Esmaeel Nezhad, Pedro HJ Nardelli, Subham Sahoo, and Farideh Ghanavati. Scheduling of energy hub resources using robust chance-constrained optimization. *IEEE Access*, 10:129738–129753, 2022.
- [36] ocean Offshore Coalition for Energy and Nature. Accelerating offshore wind and nature protection, 2025. https://offshore-coalition.eu/wp-content/uploads/2025/03/Denmark_Factsheet_FINAL_country-profile.pdf, visited 2025-09-04.
- [37] Chao Qin and Yuan Zeng. Sr-based chance-constrained economic dispatch for power systems with wind power. *IET Generation, Transmission & Distribution*, 13(13):2779–2788, 2019.
- [38] Svetlozar T Rachev, Lev B Klebanov, Stoyan V Stoyanov, and Frank Fabozzi. *The methods of distances in the theory of probability and statistics*, volume 10. Springer, 2013.
- [39] Line Roald, Sidhant Misra, Michael Chertkov, and Göran Andersson. Optimal power flow with weighted chance constraints and general policies for generation control. In *2015 54th IEEE conference on decision and control (CDC)*, pages 6927–6933. IEEE, 2015.
- [40] Line Roald, Frauke Oldewurtel, Bart Van Parys, and Göran Andersson. Security constrained optimal power flow with distributionally robust chance constraints. *arXiv preprint arXiv:1508.06061*, 2015.
- [41] Jens Sager and Astrid Nieße. Combinatorial chance-constrained economic optimization of distributed energy resources. *Energy Informatics*, 7(1):128, 2024.
- [42] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2021.
- [43] Amit Shewale, Anil Mokhade, Amruta Lipare, and Neeraj Dhanraj Bokde. Efficient techniques for residential appliances scheduling in smart homes for energy management using multiple knapsack problem. *Arabian Journal for Science and Engineering*, 49(3):3793–3813, 2024.
- [44] M Sklar. Fonctions de répartition à n dimensions et leurs marges. In *Annales de l’ISUP*, volume 8, pages 229–231, 1959.
- [45] Yanne MG Soares and Roberta AA Fagundes. Interval quantile regression models based on swarm intelligence. *Applied Soft Computing*, 72:474–485, 2018.

- [46] Tyler Summers, Joseph Warrington, Manfred Morari, and John Lygeros. Stochastic optimal power flow based on convex approximations of chance constraints. In *2014 Power Systems Computation Conference*, pages 1–7. IEEE, 2014.
- [47] Kaarthik Sundar, Harsha Nagarajan, Line Roald, Sidhant Misra, Russell Bent, and Daniel Bienstock. Chance-constrained unit commitment with n-1 security and wind uncertainty. *IEEE Transactions on Control of Network Systems*, 6(3):1062–1074, 2019.
- [48] Zhiwen Wang, Chen Shen, Feng Liu, Xiangyu Wu, Chen-Ching Liu, and Feng Gao. Chance-constrained economic dispatch with non-gaussian correlated wind power uncertainty. *IEEE Transactions on Power Systems*, 32(6):4880–4893, 2017.
- [49] Dominik Wojtczak. On strong np-completeness of rational problems. In *Computer Science–Theory and Applications: 13th International Computer Science Symposium in Russia, CSR 2018, Moscow, Russia, June 6–10, 2018, Proceedings 13*, pages 308–320. Springer, 2018.
- [50] Hongyu Wu, Mohammad Shahidehpour, Zuyi Li, and Wei Tian. Chance-constrained day-ahead scheduling in stochastic power system operation. *IEEE Transactions on Power Systems*, 29(4):1583–1591, 2014.
- [51] Zhi Wu, Pingliang Zeng, Xiao-Ping Zhang, and Qinyong Zhou. A solution to the chance-constrained two-stage stochastic program for unit commitment with wind energy integration. *IEEE Transactions on Power Systems*, 31(6):4185–4196, 2016.
- [52] Yue Xie, Oscar Harper, Hiran Assimi, Aneta Neumann, and Frank Neumann. Evolutionary algorithms for the chance-constrained knapsack problem. In *Proceedings of the genetic and evolutionary computation conference*, pages 338–346, 2019.
- [53] Xiaohe Yan, Chenghong Gu, Xin Zhang, and Furong Li. Robust optimization-based energy storage operation for system congestion management. *IEEE Systems Journal*, 14(2):2694–2702, 2020.
- [54] Sheng Yang, Samir Khuller, Sunav Choudhary, Subrata Mitra, and Kanak Mahadik. Correlated Stochastic Knapsack with a Submodular Objective. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms (ESA 2022)*, volume 244 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 91:1–91:14, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [55] Yue Yang, Wenchuan Wu, Bin Wang, and Mingjie Li. Chance-constrained economic dispatch considering curtailment strategy of renewable energy. *IEEE Transactions on Power Systems*, 36(6):5792–5802, 2021.

- [56] Yao Zhang, Jianxue Wang, Bo Zeng, and Zechun Hu. Chance-constrained two-stage unit commitment under uncertain load and wind power output using bilinear benders decomposition. *IEEE Transactions on Power Systems*, 32(5):3637–3647, 2017.

List of Figures

2.1	Gaussian distributions with different means and standard deviations.	18
2.2	Beta distributions with different parameters.	19
2.3	Weibull distributions with different parameters.	20
2.4	Correlated Beta and Triangle Distributions	23
2.5	MAE of Monte Carlo Estimators	27
2.6	Running Example	28
3.1	Sum of Uniform Distributions	39
3.2	Running Example Cost Functions	41
3.3	Running Example Sum Polynomial Fit	42
3.4	Two copula correlated resources and their estimated sum distribution and sum cost functions using Monte Carlo sampling. . . .	43
3.5	Expected cost functions of Example 13	46
4.1	SRO Overview	48
4.2	Two Gaussian distribution survival functions intersecting because of different variances.	52
4.3	Solver Run Times	55
5.1	Cumulative Costs 10 Resources Gaussian	63
5.2	Reordering Procedure	64
5.3	Simulation Control Flow	67
5.4	Correlation Information Results	72
5.5	Mean and maximum TVD in the sensitivity analysis problem sets. Mean TVD is consistently below 0.1 and maximum values only exceed 0.2 for problem D3 in the target set.	73
6.1	Congestion Scenario Grid	76
6.2	Idealized aggregated load curve from [41]. The blue bars are the aggregated base load of the scenario. The orange bars are artificially induced congestions at 4 time points during the day. .	77
6.3	Relative outputs of the different types of generators used in the example scenario from [41].	77

6.4	Congestion Resource Selection by BPSO	80
6.5	Decomposition Solver	82
6.6	Algorithm Run Time Comparison	86
B.1	Cumulative Costs 10 Resources Gaussian	104
B.2	Cumulative Costs 10 Resources Weibull	105
B.3	Cumulative Costs 10 Resources Beta	106
B.4	Cumulative Costs 30 Resources Gaussian	107
B.5	Cumulative Costs 30 Resources Weibull	108
B.6	Cumulative Costs 30 Resources Beta	109

List of Tables

1.1	Research Questions	10
1.2	Artifacts and Chapters	11
1.3	Previous publications and their use in this work.	11
2.1	Probability Distances	24
4.1	Running Example SRO	49
5.1	Algorithm Success Rates	61
5.2	North Sea Wind Parks	66
5.3	Problem Set Targets	67
5.4	Simulation Parameters	68
6.1	Generator Parameters	78
6.2	Congestions Simulation Results	79
6.3	North Sea and Baltic Wind Parks	83
6.4	North Sea Decomposition scenarios	83
6.5	Comparison of full, decomposition, and greedy solvers	84
A.1	Publications	102
A.2	Publications in Preparation	102
C.1	SRO Problem Failure Rates	111
C.2	Sensitivity Results	112

Appendix A

Publications

Citation	Content
Jens Sager and Astrid Nieße. Combinatorial Chance-Constrained Economic Optimization of Distributed Energy Resources. <i>Energy Informatics</i> , 7(1):128, 2024. https://doi.org/10.1186/s42162-024-00430-3	SR and SRO definitions, solution quality under full correlation information, congestion case study
Jens Sager, Carsten Wegkamp, Astrid Nieße, and Bernd Engel. Effect of the Quantity and Penetration of Energy Resources on the Incentivization of Grid-Beneficial Behavior in Prosumer Households. In <i>ETG Kongress 2025; Voller Energie-heute und morgen.</i> , pages 860–865. VDE, 2025. https://ieeexplore.ieee.org/abstract/document/11201961	Study on inducing grid-beneficial behavior in prosumer households under varying levels of renewable penetration.
Carsten Wegkamp, Jens Sager, Bernd Engel, and Astrid Nieße. Investigation of Mechanisms for Grid-Serving Utilization of Prosumer Household Flexibility. <i>Economics of Energy & Environmental Policy</i> , 15(1), 2026.	Incentivation and interactive scheduling of prosumer households
Rico Schrage, Jens Sager, Jan Philipp Hörding, and Stefanie Holly. mango: A modular python-based agent simulation framework. <i>SoftwareX</i> , 27:101791, 2024. https://doi.org/10.1016/j.softx.2024.101791	Python simulation framework for Multi-Agent Systems
Jens Sager and Rico Schrage. Mango. jl: A julia-based multi-agent simulation framework. <i>Journal of Open Source Software</i> , 9(102):7098, 2024. https://doi.org/10.21105/joss.07098	Julia simulation framework for Multi-Agent Systems

Table A.1: Publications

Citation	Content
Sager Jens and Astrid Nieße. The Impact of Accurate Correlation Information on the Solution Quality and Sensitivity of Stochastic Resource Optimization - submission in preparation. unpublished	SRO under partial correlation information, sensitivity analysis, North Sea case study

Table A.2: Publications in Preparation

Appendix B

Full Information Results

The following is a collection of results from [41] showing the cumulative costs achieved by different solvers in the evaluated scenarios.

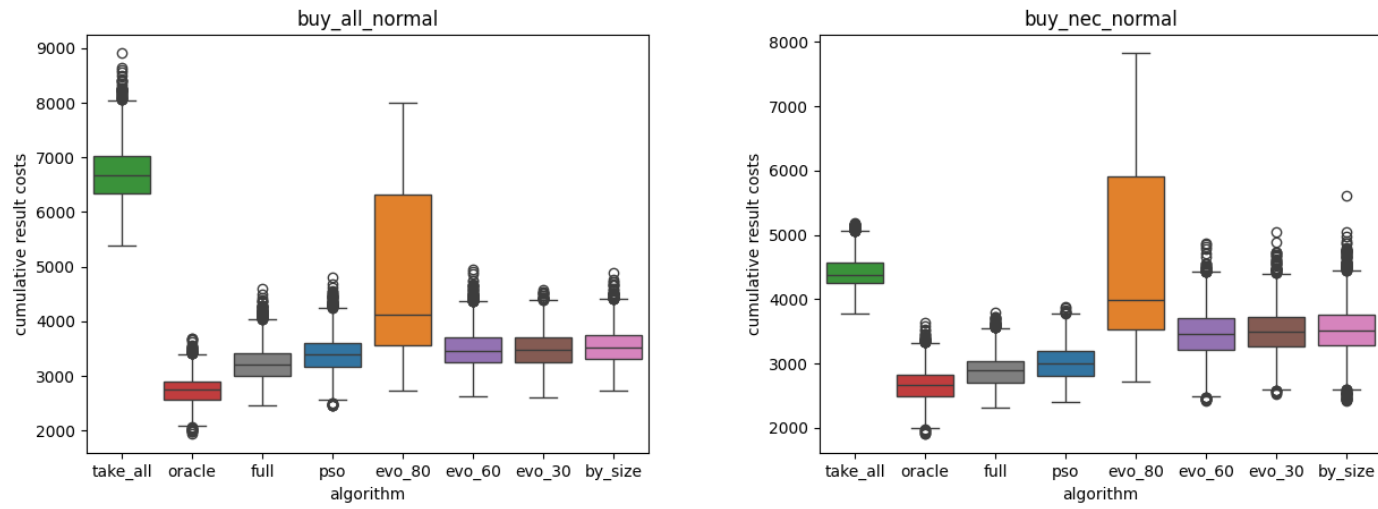


Figure B.1: Cumulative costs for 10 resource scenarios with Gaussian distributed resources for buy-all problems (left) and buy-necessary problems (right) as published in [41].

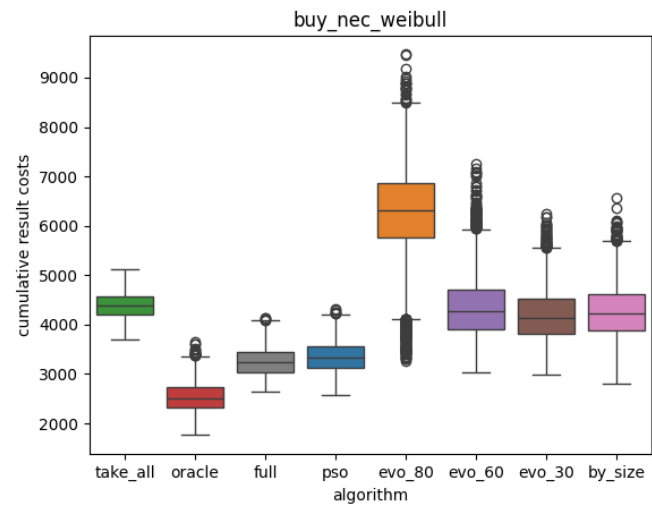
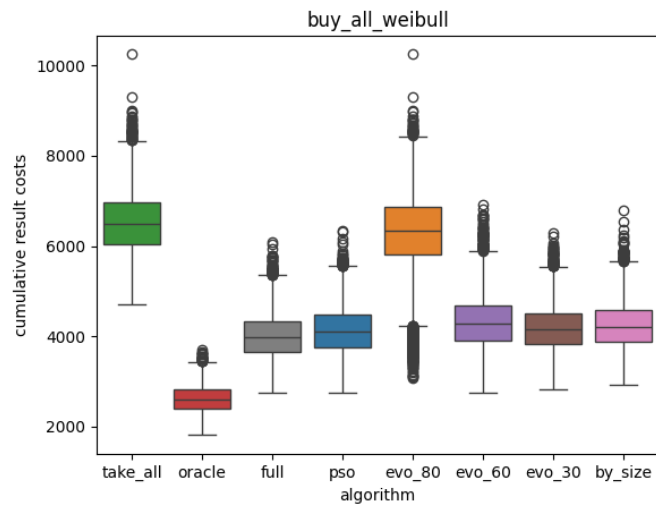


Figure B.2: Cumulative costs for 10 resource scenarios with Weibull distributed resources for buy-all problems (left) and buy-necessary problems (right) as published in [41].

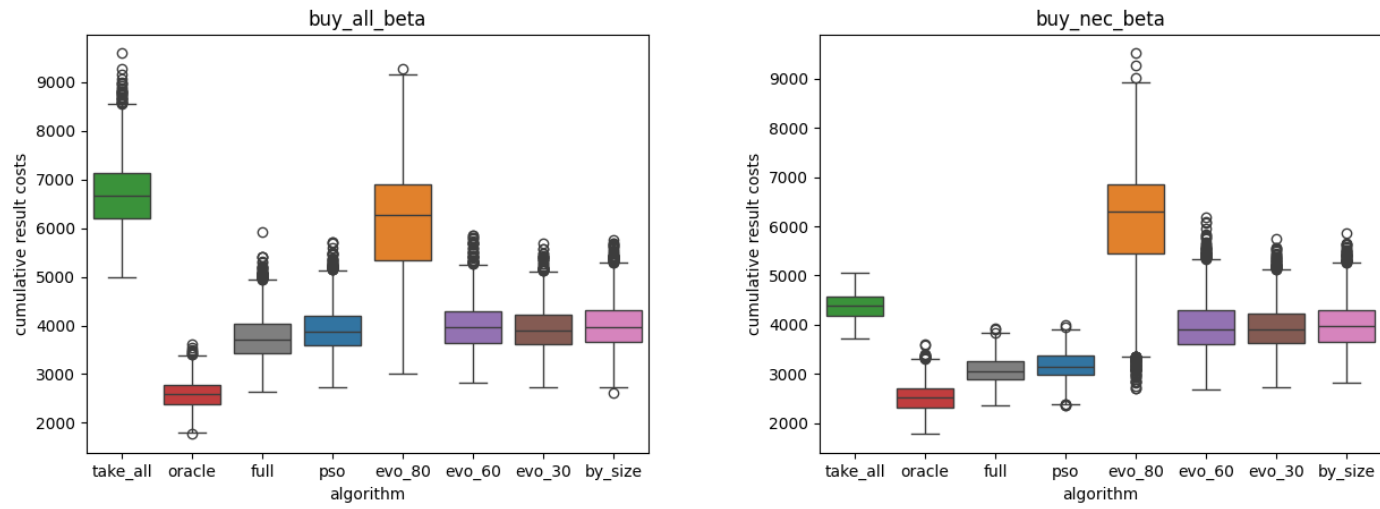


Figure B.3: Cumulative costs for 10 resource scenarios with Beta distributed resources for buy-all problems (left) and buy-necessary problems (right) as published in [41].

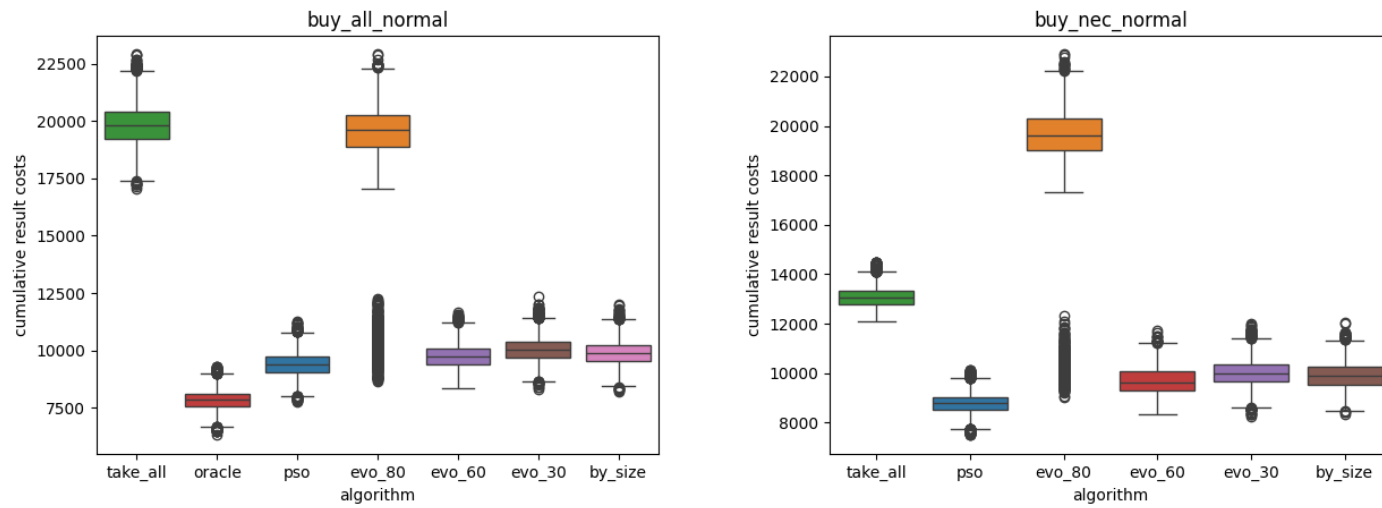


Figure B.4: Cumulative costs for 30 resource scenarios with Gaussian distributed resources for buy-all problems (left) and buy-necessary problems (right) as published in [41].

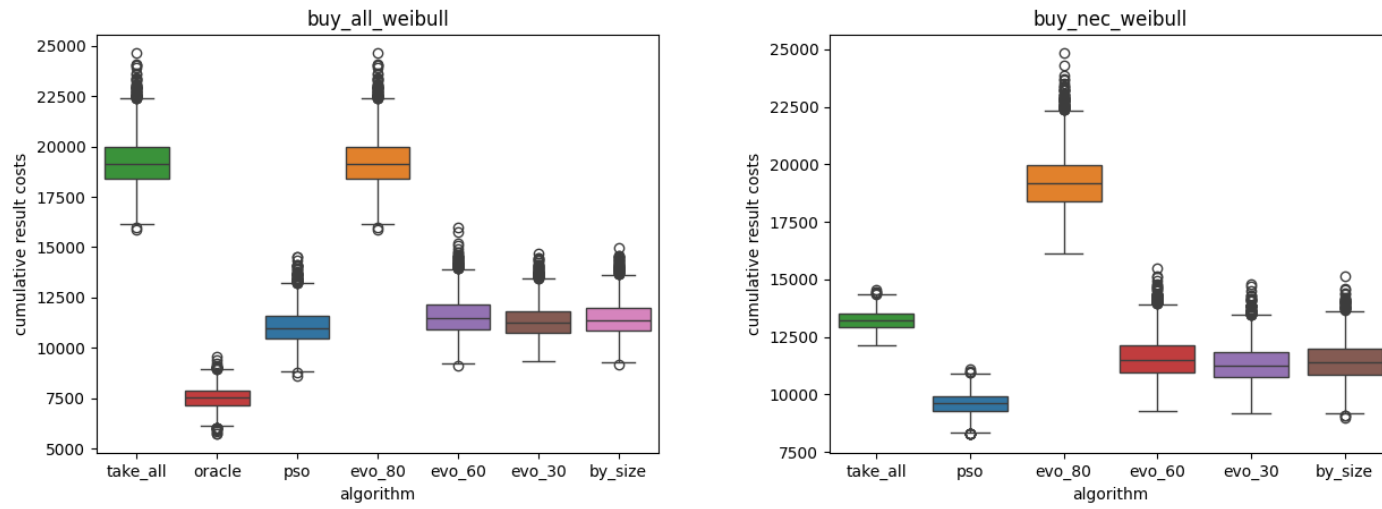


Figure B.5: Cumulative costs for 30 resource scenarios with Weibull distributed resources for buy-all problems (left) and buy-necessary problems (right) as published in [41].

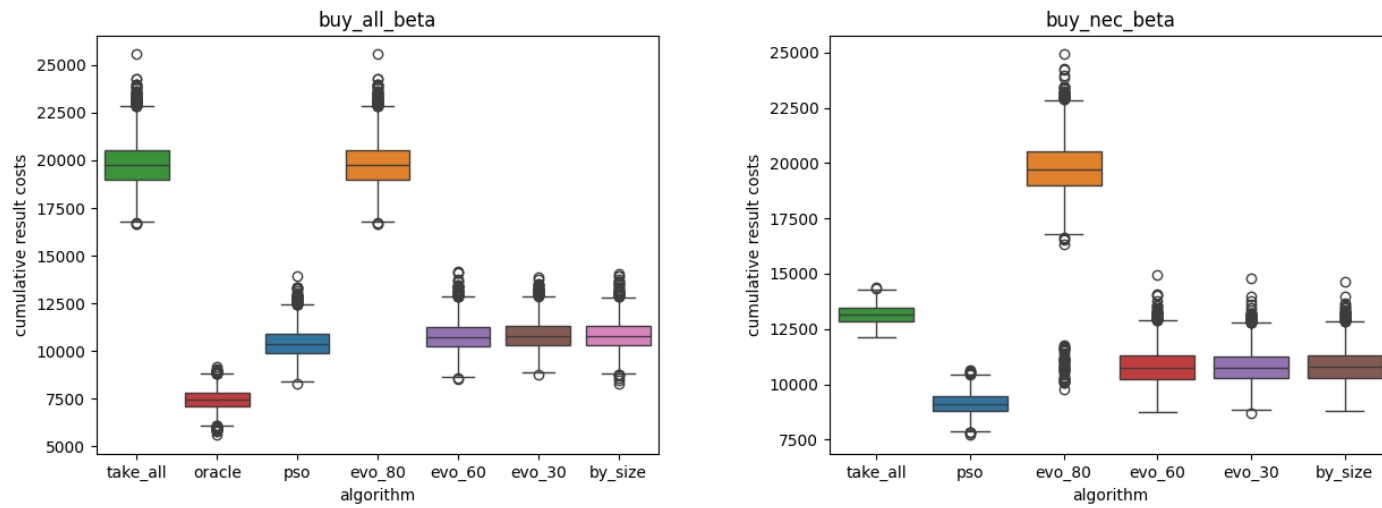


Figure B.6: Cumulative costs for 30 resource scenarios with Beta distributed resources for buy-all problems (left) and buy-necessary problems (right) as published in [41].

Appendix C

Partial Information and Sensitivity Results

size	failure rate				mean magnitude				mean cost			
	D	E	P	NC	D	E	P	NC	D	E	P	NC
2	0.83	1.0	0.01	0.97	0.03	0.08	0.31	0.07	154	143	188	9000
3	0.81	1.0	0.03	0.96	0.03	0.06	0.19	0.06	155	148	186	9100
4	0.83	1.0	0.04	0.95	0.03	0.05	0.41	0.06	156	155	183	9100
5	0.80	1.0	0.03	0.93	0.03	0.04	0.08	0.06	156	157	187	9200
6	0.77	0.91	0.07	0.88	0.04	0.03	0.34	0.04	154	163	181	9400
8	0.59	0.87	0.01	0.90	0.17	0.03	0.04	0.05	140	165	186	9400
10	0.46	0.61	0.01	0.80	0.28	0.02	0.10	0.04	135	175	186	9500
12	0.14	0.09	0.0	0.40	0.01	0.03	-	0.01	163	193	185	10000

Table C.1: Problem failure rates for SRO scenarios with different correlation information. Results are shown for the dummy (D), extreme dummy (E), extreme dummy with high positive and negative correlation (P) and North Sea (NC) scenarios. For each set, it lists the failure rate, mean failure magnitude and mean cost of the problem sets given correlation knowledge of subsets of different size. A subset size of 2 means the correlations of 6 sets of 2 resources are known, and every other correlation is assumed to be independent. A subset size of 12 means all correlation information is known.

cost set	D1	D2	D3	D4	D5	D6	E1	E2	E3	E4	P1	P2	P3	P4
d_{TV} max	0.0	0.08	0.16	0.0	0.08	0.09	0.02	0.01	0.02	0.01	0.0	<0.01	0.01	0.0
d_{TV} mean	0.0	0.03	0.07	0.0	0.03	0.04	0.01	<0.01	0.01	<0.01	0.0	<0.01	<0.01	0.0
d_{TV} median	0.0	0.03	0.07	0.0	0.02	0.04	0.01	<0.01	0.01	<0.01	0.0	0.0	<0.01	0.0
% solvable	22	100	100	3	100	100	100	100	100	100	100	100	100	100
% zeros	22	0.3	1.5	3	1.1	0.1	0.1	0.4	4.5	0	100	60	16	100
target set	D1	D2	D3	D4	D5	D6	E1	E2	E3	E4	P1	P2	P3	P4
d_{TV} max	0.02	0.17	0.25	0.0	0.13	0.15	0.15	0.16	0.13	0.10	0.17	0.15	0.13	0.08
d_{TV} mean	<0.01	0.05	0.09	<0.01	0.04	0.06	0.05	0.06	0.05	0.04	0.04	0.05	0.04	0.04
d_{TV} median	0.0	0.05	0.09	<0.01	0.04	0.06	0.04	0.05	0.04	0.04	0.03	0.05	0.05	0.03
% solvable	39	100	100	6	100	96	100	100	100	100	100	100	100	42
% zeros	17	3.8	0.1	3.8	14	14	0	0	1.9	3.7	19	24	29	14
distribution set	NC1	NC2	NC3	NC4	NC5									
d_{TV} max	0.17	0.16	0.13	0.11	0.08									
d_{TV} mean	0.06	0.06	0.05	0.04	0.03									
d_{TV} median	0.06	0.06	0.04	0.04	0.03									
% solvable	100	100	100	100	100									
% zeros	0	0	0	0	0									

Table C.2: Sensitivity results for problem sets with small changes in cost functions (top), target values (middle) and resource distribution (bottom). Problem sets are created from the dummy (D), extreme (E) and extreme positive/negative (P) sets for cost and target variations and from the North Sea correlation model (NC) for the distribution variations. For each unique pair of resource distributions, it shows the maximum (max), mean, and median total variation distance as well as the percentage of pairs where both problems were solved (% solvable) or had the exact same resource distribution (% zeros).