Optimisation of coal blending for coke-making at an integrated steel plant

Daniel Charters
u17000174

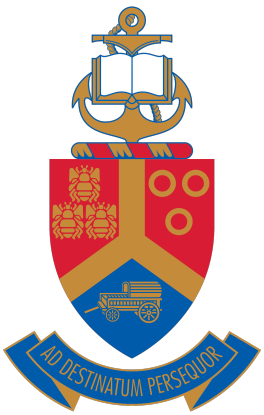
A project report in partial fulfilment of the requirements for the degree

Baccalareus (Industrial Engineering)

in the

Faculty of Engineering, Built Environment, and Information Technology

University of Pretoria



07 October 2020

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

"The goal is to hit the sweet spot of maximum optimisation, where foolish risk is balanced with excessive caution."

— Steven J. Bowen

Abstract

Title:	Optimisation of coal blending for coke-making at an integrated steel plant
Student name:	Daniel J. Charters
Student number:	u17000174
Supervisor:	Prof. Johan W. Joubert

This project is on the optimisation of coal blending for coke-making at the ArcelorMittal South Africa Newcastle Works integrated steel plant. The problem of blending different coals for coke-making is arguably one of the most critical processes in steelmaking. It has direct implications on the blast furnace operation and efficiency, and subsequently, the entire steelmaking process. The coke that is produced must be done so at minimum cost; and because of its role in the blast furnace as a fuel source, the coke must also have the highest possible quality and strength. The blending process is, however, not that simple; there is an aspect of uncertainty that is introduced into the problem through the randomly distributed coal parameters.

The problem was identified as a multi-objective, stochastic blending problem, which must achieve the lowest possible daily cost with the highest strength. The solution development to this problem involves stochastic and multi-objective programming techniques. The specific stochastic programming intervention used is chance-constrained programming, with both the scenario and approximation approaches being used. Goal programming and preemptive optimisation are the chosen multi-objective programming interventions.

The two deliverables of the project are firstly, an optimisation model that accurately captures the essence of the coal blending for coke-making problem, and secondly, a solution set that includes the models that all follow a different intervention to handling multi-objectivity and uncertainty. To achieve these deliverables, a methodology was followed that involved an in-depth literature review, model formulation and exploratory data analysis, model development and evaluation, as well as model validation.

The multi-objective and stochastic blending problem is a common model developed in literature, but there is no literature for its application to coke-making; similar problems were reviewed and used to formulate a multi-objective stochastic model that captures the essence of the problem. This model was then developed further, using different interventions to convert the multi-objective stochastic model to equivalent deterministic models. The results of the data analysis were used as the backbone for these models. This resulted in eight different models as potential solutions to the problem; these models make up the solution set. Of these, the model which makes use of preemptive optimisation and expected values for the randomly distributed coal parameters produces the best results.

The blend currently in use has a daily cost of R8 836 283, a strength value of 56.73 units, and a reliability of 62.38%; the proposed solution to the problem outperforms this blend in all three aspects. It has a daily cost of R8 774 988, a strength of 68.72, and a reliability of 83.74%. The proposed solution will save R22.4 million annually, while delivering coke with higher strength and quality, as well as a solution with far greater reliability. These results are validated through pilot-scale testing of the blend, and Monte Carlo simulation.

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

Acknowledgements

It would be amiss not to acknowledge the contributing parties that played a role in the project's success: firstly, Prof. Johan W. Joubert, at the University of Pretoria, for his guidance and supervision throughout the project; secondly, all the staff at the Newcastle Works that contributed their time and effort to the project; and lastly, my peers at the University of Pretoria, for their comments and recommendations that improved the project report. The project would not have been possible without any of you.

Contents

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

List of Figures

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

List of Tables

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

Acronyms

AMSA ArcelorMittal South Africa

CCP Chance-constrained Programming

CRI Coke reactivity index

CSR Coke strength after reaction

EDA Exploratory Data Analysis

I10 Irsid 10 index

I40 Irsid 40 index

LP Linear Programming

M10 Micum 10 index

M40 Micum 40 index

MILP Mixed Integer Linear Programming

MINLP Mixed Integer Non-linear Programming

PDF Probability Distribution Function

Chapter 1

Introduction

The project was completed at ArcelorMittal South Africa (AMSA) Newcastle Works and focussed on the optimisation of the blending of coals for the coke-making process.

1.1 Company background

AMSA Newcastle Works is an integrated steel plant, meaning the plant has facilities to process raw materials and to manufacture output. Newcastle Works encompasses all the processes needed to transform raw materials into market grade steel. The Newcastle plant is one of four AMSA plants across South Africa and is part of the ArcelorMittal group, the world's leading steel producer.

1.2 Coke's importance in steelmaking

Coke is a hard, strong and porous material made from coal. It has a high carbon content and very few impurities. It is an essential industrial product, as it is a fuel source, with it mainly being used in smelting processes. The most common smelting process it is used in is a blast furnace.

In the integrated steelmaking process, a blast furnace transforms three raw materials; iron ore, limestone and coke, into molten iron. This molten iron is then later converted to steel. The blast furnace is a crucial element of an integrated steel plant, as it is the primary link between the raw materials and finished products. Simply put, without the blast furnace, there is no integrated steel plant.

Due to the importance of the blast furnace, the raw materials combined in the furnace are also critical. Of the raw materials, coke is the most important because it is the primary fuel source in a blast furnace, and it also produces agents for molten iron production, two of which are reducing and carburising agents. Due to the role of coke, it must have high enough strength and permeability to support the blast furnace load and allow the passage of hot blast air for combustion. It also has to allow for chemical reactions to take place within the furnace to produce molten iron. Thus, ensuring coke has both physical and chemical properties of a certain level is of utmost importance in steelmaking (Sarna, 2015).

Vasko et al. (2005) describe how coke impacts the blast furnace directly, and emphasise coke quality as an essential aspect for blast furnace productivity and quality. Using a high-quality coke will result in a lower coke consumption and ultimately lower steel production costs. Numerous chemical and physical properties of coke will directly affect its quality in the blast furnace, but of these properties, coke strength is by far the most important.

There is no other suitable material that can replace coke in the blast furnace, and coke is unarguably the most critical raw material for effective blast furnace operation, and subsequently, the integrated steelmaking process (Babich and Senk, 2019).

1.3 Coke-making process

Vasko et al. (2005) describe the coke-making process as the carbonisation of coals at high temperature in the absence of oxygen, with the coking process having the following summarised steps:

1. Multiple coal types are selected and blended according to a set ratio to ensure that certain coke qualities (both physical and chemical) are met.
2. The mixture of blended coals is then placed in a coke oven for the coking cycle to start. The coke oven is where the conversion from coal to coke occurs.
3. The mixture is baked in the coke oven for approximately 20 hours, with steadily increasing temperatures, until the end temperature is around 1100°C and most of the impurities from the coal have been eliminated.
4. At the end of the coking cycle, when the coal has entirely transformed to coke, it is pushed out of the oven, cooled and transferred to the blast furnace.

AMSA Newcastle Works follows the same process for the production of their coke, and the coal blending process used is discussed below.

1.4 AMSA Newcastle's coal blending process

Typically, a blending problem requires the determination of the most economical blend of raw materials that meets specific requirements in a final product (Ashayeri et al., 1994). The problem posed in this project is an example of a blending problem.

Currently, nine possible coal types are available for the blend, seven of which are expensive, import coals. Because there is no single coal type that possesses all the required coking qualities, the blending of multiple coal types occurs. The current coal blend (which is used as a benchmark) is comprised of 70% imported coal, and the remaining 30% is local coal. The reason for the high proportion of imported coal is due to quality requirements, as the available local coal, although cheaper, is poor coking coal. This blend has a daily cost of R8 838 424, and it does not meet all of the coke quality requirements. This provides the opportunity to develop a method that optimises the coal blending for the coke-making process, while ensuring all coke quality requirements are met and, consequently, providing economic benefit for not only AMSA Newcastle Works but for the AMSA group in general.

The method that optimises the coal blending process will have to take more than one objective into account. Vasko et al. (2005) emphasise the fact that coke strength is of utmost importance for successful blast furnace operation; thus, it should be considered as an objective of the optimisation process. AMSA's only objective for their coal blending has been on cost minimisation, although they attempt to meet all the required quality measurements, it does not always prove successful. Their system in place does not allow for the simultaneous optimisation of both cost and coke strength, as it is done manually by using expert knowledge as the basis of blend selection.

Another downfall of the system that is currently in place is that the values used in modelling the coal blend are entirely deterministic. This approach could be acceptable if

the values used were always accurate, thus 100% correct at the time of blending, but this is not the case. It would be impractical as measurements would have to be taken every day, and then only after measurements have been processed could they be used to determine the optimal blend. However, this is not possible in the continuous coke-making process, both from a financial and logistical point of view. For this reason, the expected values of the coal parameters are estimated, by averaging recent readings for the particular coals, and these are the values used in the blend calculations. This approach could work if the coal qualities were to exhibit deterministic behaviour, to an extent, but this is not the case. Coal is a raw material, found in nature, so it is bound to exhibit some form of uncertainty in its parameters. Literature also highlights that coal parameters generally follow some form of a random probability distribution, rather than being entirely deterministic (Babic and Peric, 2011; Djeumou Fomeni, 2018; Shih and Frey, 1995). The coals that are used by AMSA are no exception. Each coal parameter has an element of uncertainty present. As an illustration of the uncertainty, consider the example of the ash content distribution in one of the coals, shown in Figure 1.1.

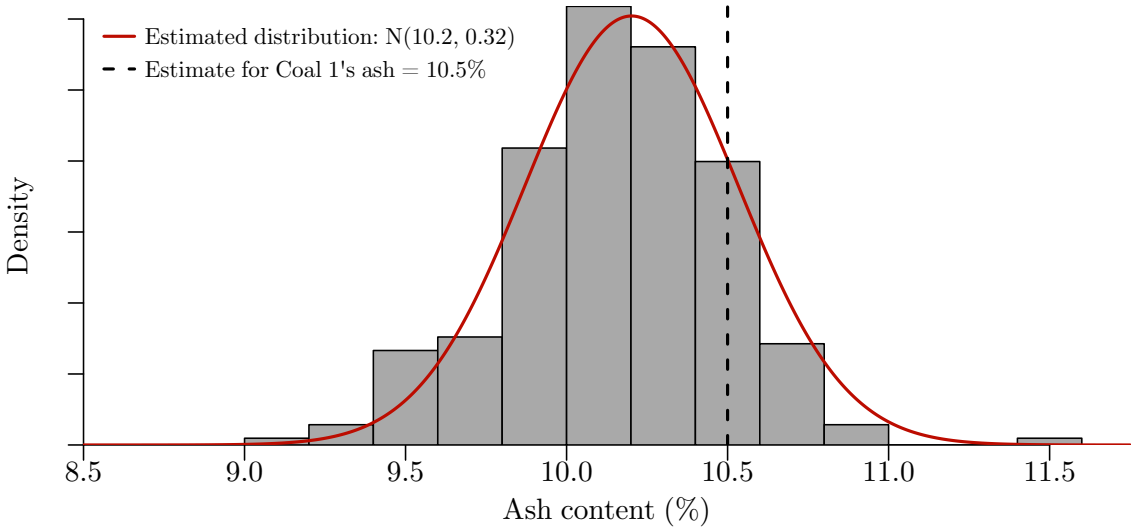


Figure 1.1: Distribution of ash content in Coal 1.

The probabilistic nature is evident, and this is the case for the other parameters and the other coal types. The distribution of this particular parameter is best estimated by $\sim \mathcal{N}(10.2, 0.32)$, which is a normal distribution with mean, μ, of 10.2% and standard deviation, σ, of 0.32%. Also shown in the figure is the expected value that AMSA use for Coal 1's ash content, which is 10.5%. This estimated value is roughly the 80^{th} percentile, meaning 80% of the time the value of the ash content in Coal 1 is below the estimated value. As but one example of all the parameters, the ash content has a direct implication on blast furnace productivity. A higher ash content means the resulting coke is a less effective fuel source in the furnace, as more ash in the blend means there is less coke to be used as fuel. The other parameters have similar implications as the ash content, and it is evident that ensuring the blend has the correct parameters is crucial. Comparing the estimated distribution with the current expected value emphasises the need to consider the uncertainty that is present in the ash content of the coals, and the same can be said about the other parameters too.

There is sufficient evidence to justify the inclusion of both uncertainty and multiple objectives into the model used in solving the problem of coal blending for coke-making. Uncertainty is present as a result of the raw materials used, which is coal, and the multi-objectivity stems from the problem being set in the context of an integrated steel plant, meaning the coke-making process should not be treated in isolation, but rather, as a pivotal role in the steel making process. This leads to the following problem statement:

> *The problem of blending coal optimally for the coke-making process at an integrated steel plant can be classified as a multi-objective, stochastic blending problem. The most effective techniques to deal with this type of problem are stochastic programming, specifically chance-constrained and two-stage recourse programming, as well as multi-objective optimisation and Monte Carlo simulation. This is consistent with the reviewed work, and in particular, that of Vasko et al. (2005), Sakalli et al. (2011), Babic and Peric (2011) and Djeumou Fomeni (2018).*

1.5 Research design

The solution to the problem of optimising the coal blending for coke-making with multiple objectives under uncertainty had two main deliverables:

1. An optimisation model that accurately captures the essence of the coal blending for coke-making problem at AMSA Newcastle Works — this model is stochastic and has multiple objectives.

2. A set of solutions, where each solution follows a different intervention to deal with the coke-making blending problem — the solution set is made up of prescriptive models which are the deterministic equivalents of the stochastic, multi-objective model.

The process of achieving these deliverables was an iterative one, and this is consistent with the literature.

1.6 Research methodology

To achieve the deliverables for this project, an operations research methodology was followed. The methodology followed was similar to that as outlined by Manson (2006).

1.6.1 Solution suggestion

As there is no exact formulation for the optimisation of coal blending for coke-making that takes uncertainty and multiple objectives into account, similar problems were reviewed, and from these, a solution method, as well as proposed solution development and solution evaluation methods, were identified. This is done in the second chapter of this report, the *Literature review*. Reviewing literature highlighted the need for an optimisation model that takes uncertainty and multiple objectives into account for the blending of coals for coke-making.

The problem in this project was identified as a multi-objective, stochastic blending problem. The suggested solution to the problem is a stochastic, multi-objective optimisation model, and it was formulated as such. This model captures the essence of the

coal blending for coke-making problem, and the formulation of this model is in the third chapter of this report, which is *Model formulation.*

The available process data were analysed to be able to formulate the suggested solution. This was to gain a better understanding of the problem, how the coke-making process fits in and affects other processes in the integrated steel plant, and lastly, to understand and model the uncertainty in the raw materials in the coking process. R (R Core Team, 2020) was used for the analysis.

1.6.2 Solution development

This is the step where the different interventions to deal with multi-objectivity and stochasticity, in the form of prescriptive models, were explored and developed as part of the solution set. Each model in the solution set followed a different intervention to approximate the uncertain blending problem. These models were subsequently formulated as the associated deterministic equivalents of the stochastic model.

The models formulated, as the solution set, were developed using state-of-practise mathematical modelling techniques including stochastic programming, multi-objective optimisation, Mixed Integer Linear Programming (MILP) and Mixed Integer Non-linear Programming (MINLP) techniques. The different interventions, to handle the uncertainty, used to obtain the equivalent deterministic models were the *expected value* approach, the *chance-constrained* approach (in particular the *scenario* and *approximation* approaches). The different interventions followed, to handle the multiple objectives, were *goal programming*, *preemptive optimisation* and the *weighted sums of objectives* approach. Combinations of the interventions to handle multi-objectivity and stochasticity were also used.

The result of this step in the methodology was the set of solutions, for the coal blending for coke-making problem, that deal with multi-objectivity and stochasticity. All the developed models are in the fourth chapter of this report, *Model development.*

1.6.3 Solution evaluation

This step in the methodology links to the penultimate chapter of the report, *Model evaluation.* All the models in the solution set were solved and compared with each other and the benchmark blend. Various statistical and analytical techniques were used for model testing, including *Monte Carlo simulation.* The possible solutions were compared by cost and quality, leading to a final version of the formulation being chosen as the solution. The chosen solution provided the optimal blend of coal with the highest possible quality and the lowest possible cost, while also being reliable.

The chosen solution was evaluated through pilot testing, which was done on a pilot coke oven at the Newcastle Works. The proposed blend, which was a solution to the chosen model, was used in a pilot coke oven, which recreated the entire coking process but on a smaller scale. The proposed blend and its resulting coke were both tested in a laboratory; with the proposed blend producing coke with a strength that is 21% better than the current one, with the reliability of the blend also being improved by 34%. The proposed solution produces a blend with not only better quality, but it is also R61 295 cheaper on a daily basis.

This provided significant insight into the reliability and performance of the models, and it converted a theoretical solution into a physical one. This testing allowed for not only theoretical validation and evaluation but the model's actual performance was also evaluated and validated. This form of evaluation is consistent with the literature.

1.6.4 Conclusion

The steps in the methodology were iterative and required revision of the proposed solutions and hypotheses based on the model evaluations. From this iterative process, a *'good-enough'* solution was chosen, which was presented as the final solution of this project. It was presented to AMSA as well. Further research into the expansion and enhancement of the project for future application was done, and recommendations were also made. The conclusion and recommendations are in the final chapter of this report, the *Conclusion*.

Chapter 2

Literature review

This chapter reviews the work that addresses the coal blending for coke-making problem, and similar problems to it. The review looks at how those problems were solved, as well as how those solutions and methods used can be applied to the problem in this project.

2.1 The blending problem

The blending problem, as defined by Ashayeri et al. (1994), is one of the oldest and best-known problems in Operations Research relating to optimisation. Approaching the problem of blending coal for the coke-making process as a blending problem is suitable, as raw materials need to be blended to obtain particular minimum or maximum characteristics in the final product.

Blending problems are usually formulated and solved using Linear Programming (LP), and this approach has been applied to numerous industries, including agriculture, process manufacturing, metallurgy and the coal and food industries, among others (Djeumou Fomeni, 2018).

Blending problems are divided into two broad categories: single-blend and multi-blend problems. Multi-blend problems relate to the formulation of several blends simultaneously and single-blend problems, as the name suggests, formulate a single blend. Both problems have constraints for the input raw materials (Ashayeri et al., 1994). In the context of the coke-making process, single-blend blending problems will be considered as there is only one possible output at any given time.

The classical blending problem model decides what the mix of ingredients is that best fulfils specific output requirements (Rardin, 2016). Blending problems, in their most basic form, have five components: sets, ingredient decision variables, composition constraints, ratio constraints and an objective function. There are generally two sets in a blending problem, a set of ingredients and a set of parameters. Sets are just all the possible ingredients and parameters. Ingredient decision variables are the principal decision variables for the blending problem; they are used to answer the question: *"How much of this ingredient should be included in the blend?"* The amount of each ingredient to include in the final blend is what the model will return as the "decision". Composition constraints enforce either upper or lower limits, or both, on each of the properties of the resulting blend. The properties of the final blend are determined by a ratio of the properties of the respective individual ingredients in the blend. Ratio constraints constrain the minimum and maximum amount of each ingredient allowed in the blend. They are also used for conditional constraints like, for example: *"If ingredient 1 is in the blend, ingredient 2 cannot be in the blend"* or *"If ingredient 1 is in the blend, ingredient 2 can only make*

up 25% of the blend." Additional binary constraints are included to be able to do this. Lastly, the objective function is what the model will try to either maximise or minimise. In blending problems, the objective is usually to minimise the cost of the blend; but it can be to maximise quality, for example. It is dependant on the problem. Below is a formulation of the classical blending problem, in its basic sense.

Let $\boldsymbol{I}$ denote the set of ingredients and $\boldsymbol{J}$ be the set of parameters. Then let

$x_i \triangleq$ the amount of ingredient $i \in \boldsymbol{I}$ included in the blend.

The model parameters are defined as:

$c_i \triangleq$ the cost (ZAR) of ingredient $i \in \boldsymbol{I}$.
$p_{i,j} \triangleq$ the value of parameter $j \in \boldsymbol{J}$ of ingredient $i \in \boldsymbol{I}$.
$L_j^{\min} \triangleq$ the lower limit for parameter $j \in \boldsymbol{J}$ in the blend.
$L_j^{\max} \triangleq$ the upper limit for parameter $j \in \boldsymbol{J}$ in the blend.
$b_i \triangleq \begin{cases} 1 & \text{if ingredient } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases}$

With all the model parameters defined, the model can be formulated.

$$\min z = \sum_{i \in \boldsymbol{I}} x_i c_i \tag{2.1}$$

subject to:

$$\sum_{i \in \boldsymbol{I}} x_i \geq n_1 \tag{2.2}$$

$$L_j^{\min} \leq \frac{\sum_{i \in \boldsymbol{I}} x_i p_{i,j}}{\sum_{i \in \boldsymbol{I}} x_i} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{2.3}$$

$$x_i \leq M(1 - y_i) \qquad \forall i \in \boldsymbol{I} \tag{2.4}$$

$$b_i \geq 1 - My_i \qquad \forall i \in \boldsymbol{I} \tag{2.5}$$

$$n_2 \leq \sum_{i \in \boldsymbol{I}} b_i \leq n_3 \tag{2.6}$$

$$x_i \geq 0 \qquad \forall i \in \boldsymbol{I} \tag{2.7}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{2.8}$$

$$y_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{2.9}$$

The objective function in (2.1) seeks to minimise the total cost of the blend by minimising the summation of each ingredient's cost in the blend. Constraint (2.2) ensures that the total amount of ingredients in the blend is above a specific, positive number, n_1. This number is generally a daily demand, or a set value (like a mass, or a number of objects, for example) which the blend will always have to be greater than or equal to. Constraint (2.3) ensures that the final blend parameter limits are adhered to for all parameters, and it is a composition constraint. The limits, $L_j^{\min}$ and $L_j^{\max}$, are known values, and it is common to see that only one of the limits applies in each instance. The parameter value is calculated as a weighted average based on the amount of ingredient i in the blend. Constraints (2.4) and (2.5) are the binary constraints, and these ensure that if ingredient i is in the blend, b_i is equal to one, and this is for all values of i. M and y are standard variables included in mathematical modelling, with M representing a very large number and y being a binary linking variable as a result of the *if-then* conditional constraint. Constraint (2.6) ensures

that the number of ingredients is above a certain level, n_2 and below another, n_3. This is a ratio constraint. Lastly, constraint (2.7) ensures that the ingredient amount will always be positive or zero, and constraints (2.8) and (2.9) ensure that b_i, for all i values, and y, are only either 0 or 1.

The classic blending problem has evolved with its application in many different industries, and the classical LP blending problem may not be suitable to solve all blending problems as it may not capture the entire essence of the problem. This has lead to the development of models suited for specific problems (Djeumou Fomeni, 2018). The problems will still be formulated using the blending problem approach, but they will be solved using different interventions or approaches that are far more capable, depending on the problem.

Blending problems can be formulated to include non-linear constraints and to use mixed-integer programming, which includes integer and continuous variables in the formulation. This allows for the adaptation of the problem to suit more complex cases (Ashayeri et al., 1994).

It is often the case where there is some form of variability in the problem, such as raw material variability or uncertain demand, for example. Blending models can also deal with this uncertainty and have been applied to multiple industries, as is seen in the work of Sakalli and Baykoc (2011), Sakalli et al. (2011), Jareonkitpoolpol et al. (2018) and Sakalli and Birgören (2019), among others. Shih and Frey (1995) and Djeumou Fomeni (2018) identified that coal properties exhibit variability, and as such, the decisions about coal blending for coke-making must deal with uncertainty.

Although cost minimisation is the typical objective of a blending problem, there may be more than one objective of the model; for example, quality maximisation and cost minimisation, as shown by Djeumou Fomeni (2018). This gives rise to the idea of multi-objective programming, which can be incorporated into the typical blending problem, as shown by Shih and Frey (1995), Babic and Peric (2011) and Sakalli and Birgören (2019), among others. Due to coke's pivotal role in the integrated steelmaking process, cost minimisation for coke-making cannot be treated in isolation, but rather the effective plant performance along with cost minimisation of the integrated process must be a priority.

The preceding paragraphs illustrate that the typical blending problem can be enhanced to incorporate non-linearity, uncertainty and multiple objectives. This is key in the coke-making process, due to the nature of the problem, as it includes uncertainty, non-linearity and the need for multiple objectives. Modelling blending problems under uncertainty and with multiple objectives are discussed in the subsections that follow.

2.2 Blending problems with uncertainty

Some uncertainty is almost always present in a process, and if not taken into account, it may cause violations of the specified limits of a model. Formulating a model with uncertainty, if it is present in the process, is essential if the model aims to capture the essence of the problem accurately (Sakalli et al., 2011). The approach of modelling under uncertainty, when it is assumed some probabilistic distribution can describe the uncertain variables, is known as stochastic programming.

Numerous authors have formulated a blending problem using stochastic programming; among them are Shih and Frey (1995), Sakalli and Baykoc (2011), Sakalli et al. (2011), Jareonkitpoolpol et al. (2018) and Sakalli and Birgören (2019). All of these authors apply an approach known as Chance-constrained Programming (CCP) to account for variability present in their particular problems.

When using CCP, uncertainty can be included in the model by integrating reliability criteria. CCP incorporates 'chance constraints' which include the measure of the reliability, i.e. the probability with which the constraints must be met (Shih and Frey, 1995).

In typical blending problems, the parameters of the raw materials are modelled as deterministic values, but this has limitations to real-world applications as parameters are very rarely deterministic (Shih and Frey, 1995). When modelling under uncertainty using stochastic programming, the authors fit distributions to the specific parameters of the raw materials. The data from these distributions are then used in the model, with the parameters treated as random variables as opposed to deterministic ones. As the parameters are now random variables, the constraints can only be realised with a minimum probability (the reliability), and not with certainty, as is the case with deterministic programming.

CCP is predicated on linear programming, but the probabilistic constraints must still be converted to their equivalent deterministic ones to be solved (Shih and Frey, 1995). The methods to convert the probabilistic natures of a stochastic programming model to equivalent deterministic ones are illustrated by Sakalli et al. (2011) and Nemirovski and Shapiro (2006), they are also illustrated with examples by Joubert (2019). When formulating a model that will take uncertainty into account, the authors first formulate a deterministic model, which is then converted to include the random variables, with the result being a stochastic model. The CCP approach is used to convert the stochastic programming model to its equivalent deterministic model. Although the initial deterministic model may be linear, the resulting deterministic model (the one converted from the stochastic model) may be non-linear. Many authors solve both models (the initial and the resulting deterministic one) to draw comparisons between the two and to evaluate the performance of the models.

The CCP approach has been successful in solving blending problems under uncertainty. The initial deterministic models generally produce a lower cost, but the reliability of the stochastic model and the robustness of CCP is often desirable (Sakalli et al., 2011).

The CCP approach is an applicable method for solving the coal blending for coke-making problem as there is known variation in the coal parameters and it has successful implementation to very similar problems, including fertilizer blending under uncertainty (Jareonkitpoolpol et al., 2018), brass casting under uncertainty (Sakalli et al., 2011) and also in coal blending for power stations (Shih and Frey, 1995).

An approach to stochastic programming that is not often applied to the blending problem, but is certainly worth looking into, is that of two-stage recourse programming. The concept of recourse is the ability to take corrective action after an uncertain event has occurred, and a two-stage recourse situation is when some decisions are made before the uncertainty is realised, which is the first stage, and then other decisions are made when the values of the random variables are known, and these would be the recourse decisions (Jensen, 2004). Using a two-stage recourse approach as an intervention when dealing with uncertainty has definite benefits, and one significant benefit is the ability to include the cost of violating a constraint into the model. One of the challenges in using this intervention is quantifying the cost of the recourse, and because the problem in this project is in an integrated steel plant and part of a continuous process, the cost of violating constraints would be challenging to quantify. A suggested solution to the difficulty in quantifying the recourse cost would be to have a fixed recourse cost, and this can be calculated from processes that are downstream from coke-making. This intervention is beneficial in the sense that it can take corrective action, whereas CCP might be too conservative and costly. The difficulty of this intervention is noted, but it is attempted as an intervention as part of the solution set.

2.3 Multi-objective blending problems

In a real-world decision-making environment, it is necessary to consider the possibility that the model will have more than one objective function, and that these objectives may be competing against each other and will require trade-offs (Shih and Frey, 1995). The typical blending problem has the objective of minimising the cost of the blend but, practically, problems almost always have more than one measure of a solution's merit (Rardin, 2016).

Examples of blending problems, similar to that of the coal blending for coke-making problem, that have multiple objectives are illustrated by Djeumou Fomeni (2018), Sakalli and Birgören (2019) and Babic and Peric (2011).

Djeumou Fomeni (2018) develops a multi-objective optimisation model for the blending of tea with the objectives of cost minimisation and improved quality. He solves the multiple objectives by combining them, by a weighted sum, to form a composite objective function. He then runs a Monte Carlo simulation with 10^4 iterations that randomly generates two different weights to be used in the composite objective function. All efficient points of the composite objective function are plotted, and from this, he can visualise and determine the efficient frontier, which is a collection of all the efficient points. An efficient point is any feasible solution to the multi-objective model. In this case, the frontier is highly non-linear, which allows the decision-maker to make competitive cost and quality trade-offs. This method has potential for the coke-making process as the relationship between coke quality (in terms of blast furnace operation) and coke price is not fully understood, especially in the context of an integrated steel plant. This method should develop an efficient frontier for that relationship.

Sakalli and Birgören (2019) apply a similar approach to the joint optimisation of quality and cost in brass casting. Their model takes uncertainty into account as well. They first solve the model for the objective of cost minimisation to find the optimal cost. In the second model, which has the objective to maximise quality, the cost of the blend is modelled as an equality constraint, which Rardin (2016) refers to as *preemptive optimisation*. Parametric programming is applied to run the model at different costs above the optimal cost; this then produces cost and quality combinations that are feasible solutions. These combinations are plotted against each other to produce an efficient (or Pareto-optimal) frontier. This process can be useful, but it will be tedious to generate a large enough dataset for the coke-making process to produce an efficient frontier. One is bound to get many infeasible solutions because of the equality constraint; it does, however, give full control to the decision-maker.

The work done by Babic and Peric (2011) uses the most common approach in multi-objective blending problems. They apply goal programming to determine the optimal blend of livestock feed with three objectives. Goal programming is an approach that aims to achieve specific goals for objective functions rather than maximising or minimising those functions. These goals are typically derived from the solution of the individual blending problems. Goal programming is the most straightforward multi-objective optimisation technique to use and implement, which is why it is the most widely used method (Rardin, 2016). In the case of this example, there are three objective functions that are all equally weighted to start. The blending problem is solved for each of the individual objective functions, and then a second model is established where the three goal functions (the objectives) are reformulated into constraints with deviation variables. Each constraint is then an equality constraint that is equal to the goal determined by the initial models for each objective (or by the decision-maker, assuming the goals are reasonable). This second model has the objective of minimising the sum of the deviations, ensuring that each of the

optimal objectives is as close to their initial value as possible. This model can be adapted to ensure that an efficient solution is obtained.

Babic and Peric (2011) also apply preemptive goal programming by weighting the objectives, which solves for each goal in a specific order. They develop and solve the second model with different scenarios by changing the weightings. This is achieved by altering the second model's objective function to include very large values that are multiplied by the deviation variables of each initial objective. The most important deviations to be minimised are multiplied by the largest values (as it is a minimisation problem). This approach gives the decision-maker the best possible solution from all perspectives.

All three of the above-mentioned multi-objective optimisation approaches can be used in the coke-making blending problem, but it is not yet known what the best approach is. The coke-making blending problem, because it is in an integrated steel plant, will have to include multiple objectives that relate to cost and quality optimisation. Quality optimisation, in the form of coke strength, is critical for efficient blast furnace operation. The problem will thus be formulated with multiple objectives.

2.4 Blending problems specifically for coke-making

After reviewing the literature, the only two papers that discuss the optimal blending of coals specifically for the coke-making process are those by Vasko et al. (2005) and Simons (1997) (cited in Vasko et al. (2005)). Both of these papers approach the problem as a blending problem, and they formulate a single-objective, deterministic LP model.

It is suggested that any mathematical model used for the optimisation of coal blending, specifically for coke-making, must, at a minimum, consider the following six constraints:

1. A minimum coke strength.
2. A maximum allowable coke oven wall pressure.
3. A maximum or minimum (dependant on the parameter) on specific chemical and physical properties of coke, with particular emphasis on volatile matter, sulphur percentage and ash content.
4. Limiting the amount of certain coals in the blend (plant-specific).
5. A minimum number of different coals as input (for logistic reasons, as well as allowing blend changes if needed, again, this is plant-specific).
6. A maximum number of different coals used to ensure that the plant can handle all the coals effectively.

A significant amount of literature for applications of real-world blending problems recognise the need to include uncertainty into the model, as it is inherent in almost every input raw material. The need to include multiple objectives is also prevalent as most processes do not happen in isolation, so their cost minimisation cannot be the only objective. Both of these points are emphasised in the coal blending for coke-making process as it is part of an integrated process, where cost and quality are of utmost importance, and uncertainty is introduced through the variability in raw materials.

2.5 Monte Carlo simulation

All the models discussed above would need to be evaluated after they are solved, as would any model. An appropriate way to evaluate a model would be to use Monte Carlo

simulation (Jareonkitpoolpol et al., 2018). Using this method allows for the comparison of all the models, and it will be a link between the real-world problem and the models solved, which would incorporate uncertainty.

Monte Carlo simulation makes use of probability distributions, for any uncertain factor, and builds models of the possible results by repeatedly sampling out of the Probability Distribution Function (PDF) of all the different uncertain factors. Using PDFs is a realistic way of describing uncertainty in the parameters of the model, so the distributions that are estimated are not only used in building and solving the models but also in testing the models (Bonate, 2001). There are several advantages of using Monte Carlo simulation for model testing when compared to using single-point estimate analysis. Among them are the fact that the results are probabilistic, meaning not what might happen is shown, but also the likelihood of that happening. The results are graphical, which helps communicate results; and a large number of possible scenarios are depicted too, which allows for scenario analysis.

Monte Carlo simulation can be applied to the coal blending problem in this project as a means of evaluating models, along with physical testing of the blends in the pilot oven. This allows for the testing and comparison of the different interventions followed to solve the stochastic, multi-objective model, as well as to be able to see how the problem is affected under various, random conditions.

2.6 Coal parameters relevant in coke-making

Literature highlights specific parameters of coal, and the parameters of the resulting coke, that are important and need to be considered when developing models for the coal blending (Vasko et al., 2005). ArcelorMittal South Africa (AMSA) specify twelve different parameters that need to be considered for the blend optimisation; five of which are from the incoming coal, and the remaining seven are the resulting coke parameters. In reality, there are many more variables that can be considered, but the most important are:

1. Phosphorus content
2. Potassium oxide content
3. Ash content
4. Sulphur content
5. Volatile content
6. Coke reactivity index (CRI)
7. Coke strength after reaction (CSR)
8. Wall pressure
9. Micum 40 index (M40)
10. Micum 10 index (M10)
11. Irsid 40 index (I40)
12. Irsid 10 index (I10)

The first five parameters listed are determined and measured in the incoming coal. These parameters behave in a non-linear fashion in the coke-oven, but AMSA has, over the years and through expert knowledge, quantified this relationship and set limits for these

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

parameters in the blend. All five of these are measured as a percentage of the total mass of the coal. Phosphorus, potassium oxide, ash and sulphur content are self-explanatory as they are just the amount of that particular element in the coal. Volatile content refers to impurities, and components that are liberated from the coal at high temperatures (such as in a coke oven). The volatile content has a direct influence on coal-to-coke yield.

The other parameters are those that are found in the resulting coke. They are only measured during, and after the coking process, and as such, historical data is of particular importance for these elements. The CRI is a measure of the coke's reactivity with carbon dioxide, and a more inert coke is desirable for blast furnace operation as there would be less weight loss. The CSR refers to the coke's "hot" strength, meaning the strength of the coke after the reaction with carbon dioxide. A higher CSR is desirable as it translates to higher quality coke and better blast furnace operation. Both CSR and CRI are measured as a percentage of the total mass. Wall pressure is a measure of the force exerted on the walls of the coke oven during the coking process as a result of the reactions in the oven. Wall pressure is measured in kilopascal (kPa), and it is desirable to have low pressure to ensure safe operation and oven longevity.

The remaining four parameters are found as a result of the combined Half-Micum and Irsid test. This Half-Micum test involves taking a sample of coke and placing it in a tumble drum, which is then rotated 100 times, after which the sample is screened. The percentage of the sample remaining that have a diameter greater than 40mm is the M40 reading, while the proportion that has a diameter smaller than 10mm is the M10 value. This process is then repeated with the coke remaining from the Half-Micum test, and the Irsid test is done. This test is the same as before, but instead of 100 rotations of the drum, there are 500. Similar to the Micum indices, I40 is a measure of the proportion of coke remaining with a diameter greater than 40mm, and I10 is the proportion with a diameter of less than 10mm. Higher M40 and I40 values, and lower M10 and I10 values correlate to higher coke strength. It can be noted that a combination of numerous parameters determines coke strength.

All of these parameters have specific limits that AMSA has predefined for the project. These limits, along with the information gained from the literature review, are used in the formulation of the model for optimising the blending of coals for coke-making in the next chapter.

Chapter 3

Model formulation

The coal blending for coke-making problem has been described as a multi-objective, stochastic blending problem. Based on the literature, the best way to formulate this problem would be by using an adaptation of the classical blending problem, making use of stochastic programming and multi-objective optimisation interventions. The model formulated in this chapter accurately captures the essence of the problem in this project.

3.1 Stochastic multi-objective model formulation

The model is formulated in such a way that it incorporates the uncertainty that is present in the different parameters of each coal type. The objective of this model is to minimise the cost of the blend while simultaneously maximising the strength of the output coke, and there are numerous constraints that the model is subject to, which ArcelorMittal South Africa (AMSA) Newcastle Works prescribes. The formulation of the stochastic, multi-objective model is as follows.

The sets are defined by letting:

$\boldsymbol{I}$ be the set of coals, such that $i \in \boldsymbol{I} = \{1, \ldots, 9\}$; where $i = 1$ is Coal 1, etc.
$\boldsymbol{J}$ be the set of coal parameters, such that $j \in \boldsymbol{J} = \{1, \ldots, 12\}$; and

$$\boldsymbol{J} = \begin{cases} 1 & \text{Phosphorus content} \\ 2 & \text{Potassium oxide content} \\ 3 & \text{Ash content} \\ 4 & \text{Sulphur content} \\ 5 & \text{Volatile content} \\ 6 & \text{Coke reactivity index (CRI)} \\ 7 & \text{Coke strength after reaction (CSR)} \\ 8 & \text{Wall pressure} \\ 9 & \text{Micum 40 index (M40)} \\ 10 & \text{Micum 10 index (M10)} \\ 11 & \text{Irsid 40 index (I40)} \\ 12 & \text{Irsid 10 index (I10)} \end{cases}$$

The decision variable is defined as

$x_i \triangleq$ the amount (tons) of coal type $i \in \boldsymbol{I}$ in the blend.

The model parameters are defined as

$$
\begin{aligned}
c_i &\triangleq \text{the cost (R/ton) of coal type } i \in \boldsymbol{I}.\\
\widetilde{p}_{i,j} &\triangleq \text{the randomly distributed parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}.\\
s_i &\triangleq \text{the expected strength measure of coal type } i \in \boldsymbol{I} \text{ after coking.}\\
L_j^{\min} &\triangleq \text{the lower limit for parameter } j \in \boldsymbol{J} \text{ in the blend.}\\
L_j^{\max} &\triangleq \text{the upper limit for parameter } j \in \boldsymbol{J} \text{ in the blend.}\\
b_i &\triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;}\\ 0 & \text{otherwise.}\end{cases}\\
d_i &\triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I};\\ 0 & \text{otherwise.}\end{cases}\\
t &\triangleq \text{the total amount (tons) of coal in the blend.}
\end{aligned}
$$

The model is then formulated as follows:

$$\min z_c = \sum_{i \in \boldsymbol{I}} x_i c_i \tag{3.1}$$

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{3.2}$$

subject to:

$$t = 2\,996 \tag{3.3}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{3.4}$$

$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i \widetilde{p}_{i,j}}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{3.5}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{3.6}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{3.7}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{3.8}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{3.9}$$

$$x_9 \leq 60 \tag{3.10}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{3.11}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{3.12}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{3.13}$$

Objective function (3.1) seeks to minimise the total cost of the blend, while (3.2) has the aim of maximising the strength of the output coke. Constraints (3.3) and (3.4) ensure that the total amount of coal (in tons) in the blend is exactly 2 996, this is the daily amount of coal that AMSA uses. The constraint (3.5) ensures that the parameter limits for the blend are adhered to for all the parameters. The level of each parameter is calculated as a weighted average based on the amount of the coal type in the blend. The parameters, $\widetilde{p}_{i,j}$, are randomly distributed. Constraints (3.6)–(3.8) are the conditional binary constraints that ensure that if the coal type is in the blend, it will have a minimum amount of 30 tons, this is the case for all the coal types. The variable t is used rather than the standard M, as the most any given coal type can be is $t = 2\,996$, so it will suffice in the formulation.

The constraint (3.9) ensures that the total number of coal types in the blend is between, and including, four and eight. This is a requirement from AMSA. Constraint (3.10) limits the amount of Coal 9 in the blend to 60 tons. The reason for this is that Coal 9 is waste coal, and the parameters are not entirely known. Waste coal is the coal that is gathered which has fallen out of trucks, off conveyors, leaked from the silos, etc. Constraint (3.11) ensures that the amount of coal type $i \in \boldsymbol{I}$ is either zero, or positive, and that is it is an integer (due to logistical reasons). Lastly, the constraints (3.12) and (3.13) ensure the binary variables are only either 0 or 1.

To understand the uncertainty that is present in the formulation, consider constraint (3.5), which limits the parameters between their upper and lower bounds, with the parameter $\widetilde{p}_{i,j}$ being randomly distributed. To illustrate the uncertainty, consider the example where $j = 3$, thus the parameter is the ash content. There is only one limit for the ash content in the blend, and it is an upper limit of 10%. Constraint (3.5) can be explicitly rewritten for the ash content parameter, and it is as follows:

$$\sum_{i \in \boldsymbol{I}} \frac{x_i \widetilde{p}_{i,3}}{t} \leq 10 \tag{3.14}$$

The adapted constraint (3.14) ensures that the ash content in the final blend does not exceed 10%. The numerator of (3.14) can be expanded further as:

$$x_1\widetilde{p}_{1,3} + x_2\widetilde{p}_{2,3} + x_3\widetilde{p}_{3,3} + x_4\widetilde{p}_{4,3} + x_5\widetilde{p}_{5,3} + x_6\widetilde{p}_{6,3} + x_7\widetilde{p}_{7,3} + x_8\widetilde{p}_{8,3} + x_9\widetilde{p}_{9,3}$$

The expansion above shows each coal type $i \in \boldsymbol{I}$ multiplied by its corresponding randomly distributed ash parameter $\widetilde{p}_{i,3}$. This means that every coal type's ash content is randomly distributed, as is the case which can be seen in Figure 1.1 in Chapter 1, where the distribution of $\widetilde{p}_{1,3}$ is plotted. The random distribution of all the other coal's ash content, $\widetilde{p}_{i,3}$, can be seen in Figure 3.1, as well as how distributions compare with one another, and where on the distribution the upper limit of the ash content lies. The case is similar for all other parameters $j \in \boldsymbol{J}$, and this is where the stochastic nature of the problem arises.

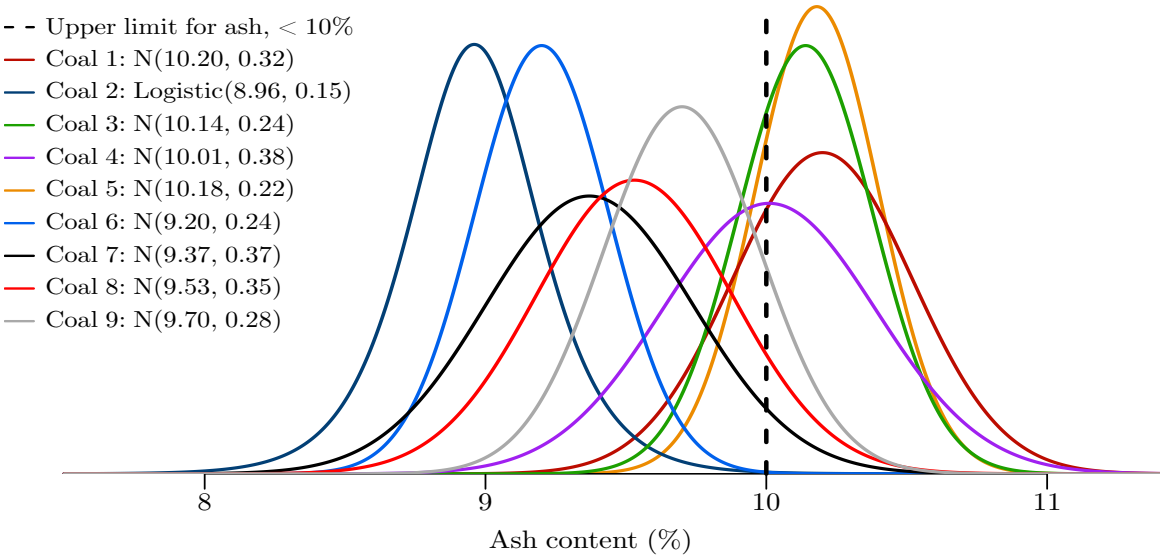


Figure 3.1: Probability distributions of ash content in all coals.

Table 3.1: Coal cost and coke strength.

Coal No.	Cost (R/ton)	Strength
1	3908	91.31
2	4098	76.03
3	1718	10.78
4	3962	69.70
5	3983	90.81
6	3420	88.99
7	3881	65.14
8	3255	60.69
9	0	56.74

Table 3.2: Parameter limits.

Parameter	Lower limit ($L_j^{\min}$)	Upper limit ($L_j^{\max}$)
Phosphorus (%)	$-\infty$	0.05
Potassium oxide (%)	$-\infty$	0.16
Ash (%)	$-\infty$	10
Sulphur (%)	$-\infty$	1
Volatiles (%)	$-\infty$	27
CRI (%)	$-\infty$	28
CSR (%)	59	∞
Wall pressure (kPa)	$-\infty$	8.5
M40 (%)	64	∞
M10 (%)	$-\infty$	10
I40 (%)	42.5	∞
I10 (%)	$-\infty$	24.5

3.1.1 Input data needed for formulation

To be able to formulate and solve the models there is data that are needed as input into the model. The cost and expected strength of each coal type are given in Table 3.1. Cost is represented by c_i and strength is s_i in the formulation. The strength is a combination of numerous other parameters, and although those parameters are randomly distributed, the strength will be assumed to be a deterministic value in the objective function of the models, and the value of strength is unitless.

The lower limits, $L_j^{\min}$, and upper limits, $L_j^{\max}$, for each of the twelve parameters can be found in Table 3.2. These limits play a vital role in the model as they are what bound the coal parameters. These limits are determined by AMSA and are constant throughout the report.

Lastly, the randomly distributed parameters used in the model, $\widetilde{p}_{i,j}$, were calculated as a result of data analysis and can be find in Tables A.1 – A.3 in the Appendix.

3.2 Data analysis for model formulation

All the data that were used in developing solutions to the stochastic, multi-objective model are analysed in this section. The method and approach used to analyse the data is discussed, with examples and a summary of the analysis provided as well.

3.2.1 The raw data

The raw data that were analysed were obtained from AMSA, and the data are from 2000 onwards. The data in its raw form were in separate, very messy Excel spreadsheets, so the first step in the analysis was to gather all the necessary observations. As only nine coal types are available for the blend, only those observations were kept. Only the twelve parameters of interest were included as well.

The data are split into two groups: incoming coal data and pilot oven data. The incoming coal data are gathered after the coal arrives at the plant, and the coal parameters measured are the phosphorus, potassium oxide, ash, sulphur and volatile content. The parameters in the pilot oven data that are of importance are CRI, CSR, M40, M10, I40, I10 and the wall pressure. The data included in the pilot oven results come from what are known as 100% trials, where only one specific coal type is put through the coking cycle in a pilot oven. This is done to get accurate readings for the coke parameters of the specific coal type. It must be noted that there are significantly fewer observations of the pilot oven data, and the reason for this is that 100% trials are not done very often. The spread of data across the different coal types is not uniform either, as coals have been in use for different periods, with some coal types having as little as 17 observations, this made analysis tricky.

As the raw data were in Excel, the initial step was done in Excel as well, which involved cleaning and sorting the data. Once all the relevant data were grouped, cleaned and sorted, it was read into R (R Core Team, 2020), where analysis could begin.

3.2.2 Exploratory data analysis

Exploratory Data Analysis (EDA) is an approach to analysing data to get information about its main characteristics, and data visualisation is a key tool in this approach. This was the approach followed to analyse the data, with the primary objective of estimating the probability distributions of the parameters. There were three necessary steps followed in the EDA, the first was hypothesising the distribution, or its family, the second step is estimating the parameters of the distribution and then lastly, evaluating the quality of the fit, which was done visually and with a *goodness of fit* test. The functions available in R (R Core Team, 2020) were used for these steps; and for reproducibility, a set seed of 20200824 was used throughout the project. What follows is an example of the process used to estimate the Probability Distribution Function (PDF) that best describes the distribution of sulphur content in Coal 4. Finding a PDF that is a good estimate for the distribution of the random variables is an essential step for the modelling interventions that deal with stochasticity.

As data visualisation is a key step in EDA, the data were plotted as a histogram. This was done to get a feel for the data and to hypothesise what distribution it could follow. An example of this is found in Figure 3.2, which is a plotted as a histogram. From this plot, it was hypothesised that a normal distribution could be a good fit for the data.

To reinforce the hypothesis, the data were plotted on a Cullen and Frey graph, shown in Figure 3.3. The Cullen and Frey graph is a standard data analysis tool which is used to

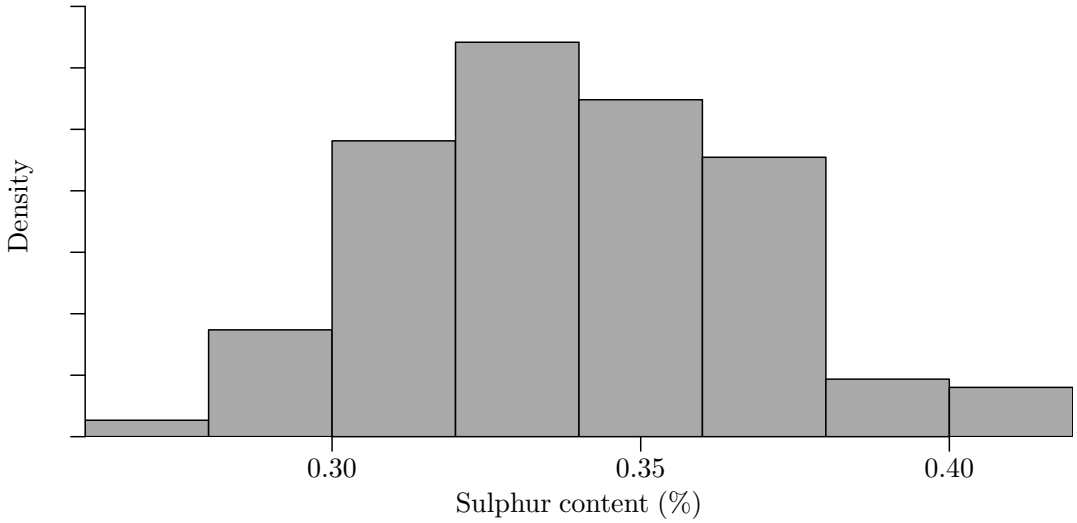


Figure 3.2: Distribution of sulphur content in Coal 4.

see the possible distributions that the data could follow. This graph is the square of the skewness of the data plotted against the kurtosis of the data. Skewness is a measure of the lack of symmetry, and the higher the square of the skewness, the less symmetrical the data are. Common symmetric distributions include the normal and uniform distributions. Kurtosis is a measure of tailedness, meaning the data either have a heavy, or a light tail when compared with a standard normal distribution. A higher kurtosis value indicates that the data are more heavy-tailed. A normal distribution has a skewness value of 0 and a kurtosis value of around 3. Also plotted on the Cullen and Frey graph are 1 000 bootstrapped values, which allows for better estimation of the distribution. Bootstrapping is a statistical tool used for resampling a single dataset to recreate many datasets, and this allows for estimations to be made about the data's parameters.

From the Cullen and Frey graph, it would appear the normal distribution is a good estimate. This is based on the observed values and the bootstrapped values. The square of the skewness is between 0 and 1, which indicates a reasonably symmetrical distribution. Furthermore, the kurtosis is grouped between 2 and 4, which is a seemingly good approximation for a normal distribution.

Another technique to visually estimate the PDF is to plot bootstrapped distributions of the estimated PDF and to compare it with the empirical probability distribution of the data. This plot shows two things; how different the estimated distribution would be if we were to resample out of the data 1 000 times and get different parameters, and how the empirical distribution of the data compares, graphically, to the estimated distribution. This plot can be seen in Figure 3.4.

From these plots, it would seem that a normal distribution is a reasonable estimate of the distribution of sulphur content in Coal 4. The bootstrapped distributions are somewhat condensed around the estimated PDF, which can be expected based on the bootstrapped values of the Cullen and Frey graph and the central limit theorem. The empirical distribution fits in reasonably well with bootstrapped ones, which could indicate that the estimated PDF is a decent fit for the data.

The next step would be to estimate the parameters of the distribution, and in this case, it would be mean, μ, and the standard deviation, σ of the sulphur content. This is

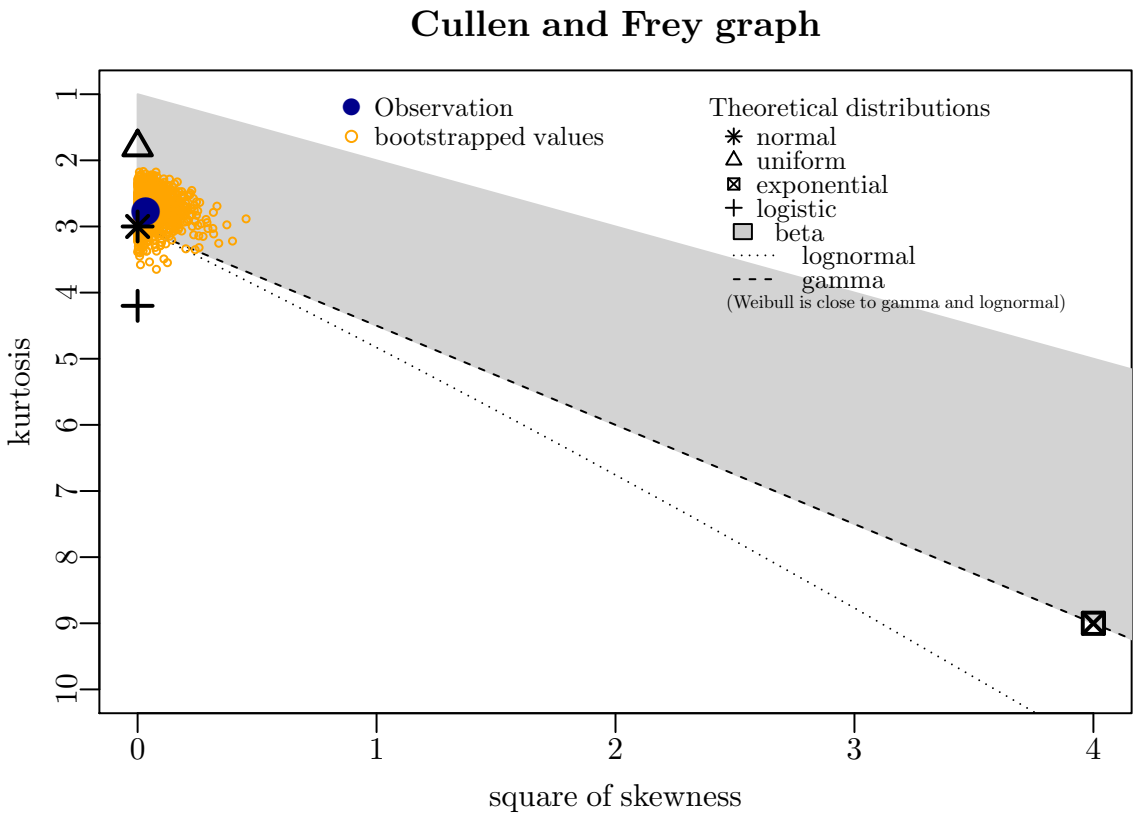


Figure 3.3: Cullen and Frey graph of sulphur content in Coal 4.

a relatively straightforward procedure. The estimated PDF is $\sim \mathcal{N}(\mu = 0.34, \sigma = 0.03)$.

To quantify the estimation and to formally allow interpretation of the fitted distribution from a hypothesis test, a *Chi-squared (χ^2) test* was used. The χ^2 test, as a goodness of fit test, is used to determine how well the estimated distribution fits the empirical distribution. It is a hypothesis test, with the null hypothesis being that there is no significant difference between the estimated and the empirical distributions; and the alternative hypothesis is that there is a significant difference between the estimated and empirical distributions. A significance level (α) of 0.05 was used, and the test was carried out in R (R Core Team, 2020) using the built-in functionality. The statistic of interest from this test is the probability value. If the probability value is less than the significance level, the null hypothesis is rejected, and if it is higher than the significance level, the null hypothesis is not rejected. In the case of the sulphur content in Coal 4, the probability value from the χ^2 test was 0.097, which is higher than α, resulting in the null hypothesis failing to be rejected. It can then be assumed that $\sim \mathcal{N}(\mu = 0.34, \sigma = 0.03)$ is a decent approximation of the distribution of the sulphur content in Coal 4. This fitted distribution can be seen in Figure 3.5.

It must be noted that the χ^2 test is not the only goodness of fit test, with the *Kolmogorov-Smirnov*, *Anderson-Darling* and the *Shapiro-Wilk* tests also being used to quantify and interpret the results of a hypothesis test. Probability-probability and quantile-quantile plots are also used as visual assessments of the goodness of fit.

This process paints an interesting picture of the sulphur content in Coal 4, and although not a perfect fit, the normal distribution $\sim \mathcal{N}(0.34, 0.03)$ seems like a good estimate for

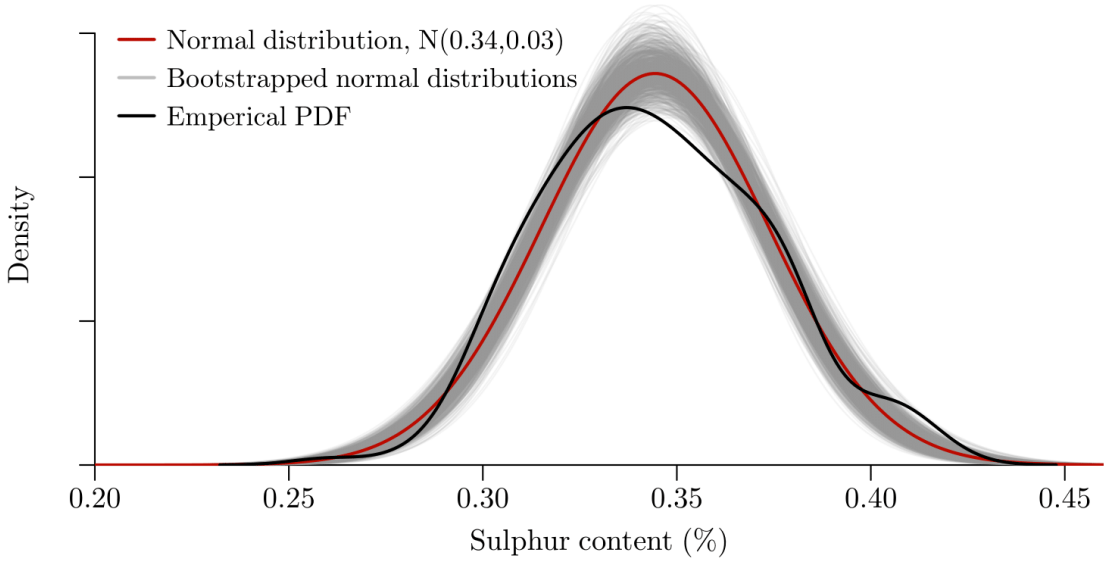


Figure 3.4: Empirical PDF of Coal 4's sulphur content compared to bootstrapped normal distributions.

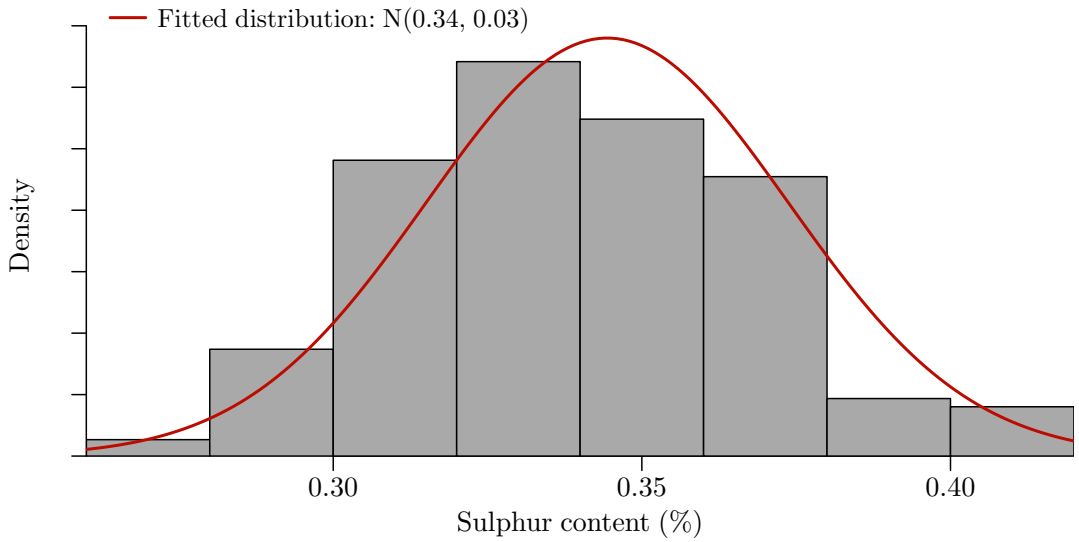


Figure 3.5: Distribution and fitted distribution of sulphur content in Coal 4.

the distribution. This was but one example of how the estimated probability distributions of the parameters in all the coal types were arrived at by using EDA and formal hypothesis tests. A summary of the other parameters is given in the next section.

3.2.3 Summary of data

After analysing the available data by following the EDA approach, each parameter of every coal type was estimated with a probability distribution. Some of these distributions are not used in the various interventions that are used to solve the coal blending problem, but it is worth noting that they are present in the process, and as such, are part of the stochastic model too.

It was significantly more challenging to estimate the distributions for the coke parameters as there is far less data for these. In the case of Coal 9, which is waste coal, all parameters have been assumed to be normally distributed, and both the mean and standard deviation are calculated by taking a weighted average of the other coal's expected values.

The parameters of the incoming coals can be found in Table A.3. There are five different distribution types present in the incoming coal parameters: where $\mathcal{N}(\mu, \sigma)$ is a normal distribution with mean of μ and standard deviation of σ; $\mathcal{U}(min, max)$ is a uniform distribution with minimum and maximum as parameters; $\ln\mathcal{N}(\mu, \sigma)$ is a lognormal distribution with parameters μ as the log of the mean, and σ as the log of the standard deviation; $Logis(\mu, s)$ is a logistic distribution with parameters μ as the location and s as the scale; and lastly the Weibull distribution, $Weib(k, \lambda)$, where k is the shape parameter and λ is the scale parameter.

The coke parameters, in a general sense, are less likely to follow a known distribution, thus estimating distributions for them was challenging. The Gamma distribution, $\Gamma(\alpha, \beta)$, with parameters α as the shape and β as the rate, is a lot more common in the coke parameters, particularly in the wall pressure. The coke parameter probability distribution estimates can be found in Table A.1 and Table A.2.

The data that were gathered, analysed and interpreted were used in conjunction with the multi-objective, stochastic model for optimising the coal blending for coke-making. The data were used in different ways, but the two most important uses of the data were its application to stochastic programming, and for calculating more accurate expected values when using the expected value approach as an intervention to deal with uncertainty.

The model formulated in this chapter captures the essence of the coal blending problem, but that is of little use if it is not solved. The different interventions to solve the stochastic, multi-objective model, and the formulation of the models that use those interventions to solve the problem, are discussed in the next chapter.

Chapter 4

Model development

To develop the solution set and actually solve the problem, the stochastic model formulated in the previous chapter must be converted to the deterministic equivalent models, while also solving for both objectives. To formulate the models, different interventions to handle uncertainty and multiple objectives were followed, and this chapter includes the models that were formulated using the interventions. There are two main approaches followed when dealing with uncertainty; the *expected value approach* and Chance-constrained Programming (CCP). The interventions used for the multi-objectivity are *goal programming*, *weighted sum of objectives* and *preemptive optimisation*.

This chapter is laid out such that the first models formulated are the *expected value* models, with the different multi-objective interventions, followed by the CCP models, again with the multi-objective interventions. Brief results from the models are also included, but they are interpreted and compared in the next chapter. The results that are included in this chapter are based around cost and strength. Also included are explanations of the relevant interventions, and how they are applied with reference to the multi-objective stochastic model. The interventions to handle multiple objectives are discussed in the beginning (as all the models will include multiple objectives), while the interventions to deal with uncertainty are discussed in the relevant subsections.

Goal programming

Goal programming is an extension of Linear Programming (LP), with the same basic techniques being used to formulate the model, and it is the most common technique for dealing with multi-objective problems, mainly because of its ease of implementation (Rardin, 2016). Goal programming was developed to solve problems with multiple objectives, and this helps decision making where there are competing objectives (Rifai, 1996). The objective of the goal programming model does not optimise the objectives directly, as in a LP, but it rather attempts to minimise the deviations from the goal by the realised results. Rifai (1996) highlight the steps needed to formulate a goal programming model:

1. Identify goals and express the multiple objective functions as individual constraints, including the deviational variables.
2. Formulate a new objective function for the goal programming model which aims to minimise the sum of all the deviational variables. Normalising the deviational variables must be done if the goals have different units (conversion to percentage is the most common).
3. Assign a hierarchy of importance to the goals by multiplying the respective deviational variables with a priority factor. This priority factor is preemptive

and reflects the desired importance of each of the goals. This is referred to as *preemptive goal programming* by Winston (2003), and this can be done by manually assigning weights or by means of a Monte Carlo simulation, as shown by Djeumou Fomeni (2018). If using the Monte Carlo simulation approach, a Pareto optimal frontier will be developed.

4. Add or subtract very small multiples of the original objectives from the new objective function. This ensures that the goal programming model yields an efficient point.
5. Formulate and solve the model as one would in the case of LP, MILP or MINLP.

Based on the above steps, the goal programming intervention for the stochastic multi-objective model is formulated as follows, and this method will be used when formulating models for the solution set:

The initial objective functions are:

$$\min z_c = \sum_{i \in \boldsymbol{I}} x_i c_i \tag{4.1}$$

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{4.2}$$

These objective functions are then formulated into constraints. All the previous constraints, equations (3.3) – (3.13), will still be included in the formulation.

$$\sum_{i \in \boldsymbol{I}} x_i c_i - d_c = G_c \tag{4.3}$$

$$\sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} + d_s = G_s \tag{4.4}$$

There are four new variables introduced into the model, with d_c and d_s being the deviational variables for *cost* and *strength*, respectively. G_c is the goal value for the cost minimisation objective, similarly for G_s as the strength maximisation objective. The values of G_c and G_s are found by solving for the respective objectives individually, with no regard for the other. G_c is the lowest possible cost of the blend that meets all the requirements, similarly for G_s being the highest strength of the blend. By rearranging the above equations, and normalising them such that the deviational variables are both in terms of percentage, the resulting normalised variables are:

$$d_c^N = \frac{d_c}{G_c} \tag{4.5}$$

$$d_s^N = \frac{d_s}{G_s} \tag{4.6}$$

Where d_c^N is the normalised cost deviational variable, similarly for strength, and the calculation of these are included as additional constraints. The single objective function, which seeks to minimise the deviation from the goals, is as follows:

$$\min z_d = w_c d_c^N + w_s d_s^N - 10^{-12} z_c + 10^{-7} z_s \tag{4.7}$$

The variables w_c and w_s are the priority factors, or weights, for the cost and strength goals, respectively. These can be changed, either manually or through simulation, to develop different scenarios. By subtracting a very small multiple of the original cost

objective, z_c, and adding a very small multiple of the original strength objective, z_s, we ensure that the model returns an efficient point. The different values that the respective objective functions are multiplied by are to ensure that they are in the same value range and can be added together. This objective function will be subject to all of the original constraints of the multi-objective stochastic model, but will also include the additional ones formulated above, and this approach is followed throughout. The next intervention to deal with multi-objectivity discussed is the *weighted sum of objectives* approach.

Weighted sum of objectives

A relatively rudimentary approach to dealing with multiple objectives is that of the weighted sum of objectives method. Using this method, the multiple objective functions are combined into a single, composite objective function which is to be maximised by summing the original objectives that are to be maximised with positive weights, and those which were to be minimised with negative weights. The weights are simply multiplied with the original objectives, and can be any range of numbers. The signs orient the objectives, while the weights reflective the relative importance of the objectives (Rardin, 2016). The objectives generally differ in units, and as such would need to be normalised (again, normally to percentages).

This method is a common one in multi-objective optimisation, and it is an independent method as well as a component of other methods, such as in goal programming (Marler and Arora, 2010). Much the same as the third step above, where priority factors are assigned to the goals, the weights can be user assigned to compare different scenarios, or a Monte Carlo simulation approach can also be followed where changing weights are simulated to produce a Pareto frontier, as done by Djeumou Fomeni (2018). Once formulated as a single objective model, it can be solved and if the single objective model produces an optimal solution, the solution is an efficient point of the multi-objective model (Rardin, 2016). These solutions are known as a Pareto optimal solutions as it is unlikely that they optimise all the objectives, and one objective cannot be improved without leaving the other objectives worse off, hence the need for the different weightings as some objectives would be more important than others.

Similar to above, the steps to formulate a weighted sum of objective model are as follows:

The initial objective functions are:

$$\min z_c = \sum_{i \in \boldsymbol{I}} x_i c_i \tag{4.8}$$

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{4.9}$$

These objective functions are then combined into one objective function, and this function maximises the combination. It is formed by subtracting the cost minimisation objective from the strength maximisation objective:

$$\max z_{c,s} = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} - \sum_{i \in \boldsymbol{I}} x_i c_i \tag{4.10}$$

But, similar to the goal programming approach, these objectives are in different units, so they have to be normalised, and the resulting objective function is:

$$\max z_{c,s} = w_s \frac{\sum_{i \in \boldsymbol{I}} x_i s_i}{t G_s} - w_c \frac{\sum_{i \in \boldsymbol{I}} x_i c_i}{G_c} \tag{4.11}$$

The goal variables, G_c and G_s are found by first solving the model for each individual objective, and the weight variables, w_c and w_s, are reflect the relative importance of each objective. These weights can be changed manually or through Monte Carlo simulation. This objective function will still be subject to all the constraints from the original model, which are equations (3.3) – (3.13), and the model will be solved accordingly.

Preemptive optimisation

The last of the interventions to handle multi-objectivity is that of preemptive optimisation. The preemptive process is when the objectives are solved in turn, trying to improve one without worsening the others. The objectives are rarely of equal importance, and in preemptive optimisation the most important objective is solved for first (as the only objective of the model). The optimal objective value for that model is then modelled as a constraint in the next model, where the most important objective is set equal to the optimal solution attained from its model; this model is then solved for the second most important objective, and so forth. By doing this, the objectives are solved for in the order of their relative importance and it is assured that solving one objective will not worsen another.

It is up to the decision maker to determine the order and to generate the different scenarios which can be compared. But, as with *goal programming* and *weighted sum of objectives*, a Monte Carlo simulation can be applied to preemptive optimisation, and this works particularly well when there are only two objectives. This approach can be used to solve for the model by incrementally changing the value of one of the objectives — which is a constraint in the model — and solving for the other objective. This, once again, produces a Pareto frontier of non-dominated points. This method is applied by Sakalli and Birgören (2019) for a blending problem, and it is also known as parametric programming.

In the context of this problem, there are only two objective functions, and of those, cost minimisation is considered to be the most important for AMSA, thus it would be solved for first, resulting in the goal value, G_c. This objective will then be modelled as a constraint in the second model, which seeks to maximise strength, and it would be as follows:

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{4.12}$$

subject to:

$$\sum_{i \in \boldsymbol{I}} x_i c_i \leq G_c \tag{4.13}$$

This objective function would still also be subject to all the other constraints, as in the previous two methods. By using the above approach one would ensure that the lowest possible cost is met first, and then with that cost, the maximum strength is met. This can be done in the opposite way, where strength is first maximised and then cost minimised. This gives the decision maker two possible scenarios; a cost optimal solution, or an optimal strength solution, as strength and cost are competing objectives. A useful way to generate more scenarios, which allows the decision maker more options, would be to incrementally increase the value of G_c and solve the model multiple times for the changing value, which would be done by a Monte Carlo simulation. Assuming the same objective as before, strength maximisation which is subject to the cost equality constraint, the objective function would still be to maximise the strength, but now it is subject to a changing cost constraint which is as follows:

$$\sum_{i \in \boldsymbol{I}} x_i c_i \leq G_c + C_{increment} \tag{4.14}$$

The new variable, $C_{increment}$, is the increment that the cost objective will be increased by with each iteration, and it is chosen by the decision maker. The value of the increment will remain the same, the value of G_c would increase incrementally each time the model is solved, therefore shifting the goal. Doing this will generate different scenarios for the decision maker to choose from, and it will produce a Pareto frontier.

The three interventions discussed are the most commonly used ones in a multi-criteria decision making setting, and they are all relatively straightforward to implement, as shown. They will all be used to develop the solution set in conjunction with the *expected value* and CCP models.

4.1 Expected value models

When dealing with a stochastic problem, where decision making must be done under uncertainty, the expected value approach is an intervention to dealing with the uncertainty. This approach results in an equivalent deterministic model, subject to all the same constraints as the original model. It is a simple method to use, and it often produces favourable objective function values, but with little reliability. The reliability of the expected value models depends on the problem and the circumstances; it is dependent on with how much certainty the model parameters are known.

In the case of the coal blending for coke-making problem, the quality parameters can be determined with some certainty if all the coals are tested before they are used, meaning their respective qualities would be known with *certainty*. This is logistically impossible as the testing process is a long and expensive one, and the coal types are ordered based on the amount that will be used, so blend modelling happens well before the coals are actually physically blended together for the coke-making process. So, as an approach to dealing with the uncertainty, the expected value of each of the coal parameters can be used in place of the Probability Distribution Function (PDF). This simplifies the problem, but it can be very effective if used at the correct time with the correct information.

For the expected values of the different parameters there are two different approaches; one is to use the values that ArcelorMittal South Africa (AMSA) assign to the parameters as their expected values, and the other is to estimate the expected value of the parameters

based on their PDF. Both of these methods are used in conjunction with the multi-objective interventions. They will both have the same model, with the only difference being the data used for the coal parameters. The randomly distributed coal parameters, $\widetilde{p}_{i,j}$, will become expected values as follows:

$$E(\widetilde{p}_{i,j}) = p^A_{i,j} \tag{4.15}$$

or

$$E(\widetilde{p}_{i,j}) = p^E_{i,j} \tag{4.16}$$

where $p^A_{i,j}$ is the value that AMSA expects for parameter $j \in \boldsymbol{J}$ in coal type $i \in \boldsymbol{I}$, and $p^E_{i,j}$ is the estimated expected value for parameter $j \in \boldsymbol{J}$ in coal type $i \in \boldsymbol{I}$ based on the PDF's found in Tables A.1 – A.3. Using equations (4.15) and (4.16) in place of $\widetilde{p}_{i,j}$ in the model the expected value modules can be formulated.

4.1.1 AMSA's expected value model

This model, much the same as the original, multi-objective stochastic model, is formulated in such a way as to minimise the total cost of the blend while simultaneously maximising blend strength. The model is subject to the same constraints as the original model. These constraints ensure the required coke parameters, as well as the other constraints specific to AMSA, are met. This model is a deterministic equivalent of the original, stochastic model. It still has the two objectives, and the models that follow a multi-objective intervention are formulated in subsequent subsections.

The cost and strength input data for the model can be found in Table 3.1. The parameter limits are the same throughout, and they can be found in Table 3.2, and the expected values that AMSA uses for the different parameters can be found in Table B.1 in Appendix B. The model is then formulated by letting $\boldsymbol{I}$ be the set of coals, such that $i \in \boldsymbol{I} = \{1, \ldots, 9\}$, and $\boldsymbol{J}$ be the set of coal parameters, such that $j \in \boldsymbol{J} = \{1, \ldots, 12\}$.

The model parameters are defined as follows:

$$\begin{aligned}
x_i &\triangleq \text{ the amount (tons) of coal type } i \in \boldsymbol{I} \text{ in the blend.} \\
c_i &\triangleq \text{ the cost (R/ton) of coal type } i \in \boldsymbol{I}. \\
s_i &\triangleq \text{ the expected strength measure of coal type } i \in \boldsymbol{I} \text{ after coking.} \\
p^A_{i,j} &\triangleq \text{ the AMSA estimated parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
L^{\min}_j &\triangleq \text{ the lower limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
L^{\max}_j &\triangleq \text{ the upper limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
b_i &\triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases} \\
d_i &\triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases} \\
t &\triangleq \text{ the total amount (tons) of coal in the blend.}
\end{aligned}$$

The model is then formulated:

$$\min z_c = \sum_{i \in \boldsymbol{I}} x_i c_i \tag{4.17}$$

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{4.18}$$

subject to:

$$t = 2\,996 \tag{4.19}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{4.20}$$

$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^A}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{4.21}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{4.22}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{4.23}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{4.24}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{4.25}$$

$$x_9 \leq 60 \tag{4.26}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{4.27}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{4.28}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{4.29}$$

The model constraints are much the same as before. The objective functions (4.17) and (4.18) seek to minimise cost and maximise strength, respectively. Constraints (4.19) and (4.20) ensure the total amount of coal in the blend is exactly 2 996. The adapted constraint (4.21) ensures that the parameter limits are adhered to. The parameter values that AMSA expect are used in this formulation as $p_{i,j}^A$, and this converts the stochastic model to its deterministic form. Constraints (4.22) – (4.24) ensure that if any coal type is in the blend, then a minimum amount of 30 tons is needed (this is for logistical reasons). To ensure that a certain minimum and maximum number of coal types are in the blend constraint (4.25) is used. Constraint (4.26) only allows a maximum amount of 60 tons of waste coal in the blend. For logistical reasons, all amounts of coal in the blend must be an integer, and this is what constraint (4.27) enforces, and lastly, constraints (4.28) and (4.29) ensure the binary conditions are met.

This model is used as a base when formulating the models that follow, and all the constraints and variables above are present in the next models too. Any new variables or constraints will be highlighted where applicable. Because the model is now in its deterministic form it can be solved for the multiple objectives, and the first intervention used is goal programming.

Goal programming using AMSA's expected values

This model is formulated using the goal programming approach to the base model which has been formulated in the previous subsection. Because the goal values of each objective have to be set, the model is be solved for each of the two objectives first, and that is used to determine the values of G_c and G_s. Those are subsequently formulated as constraints with the deviational variables, d_c and d_s. The deviational variables are normalised and make up the new objective function. Equations (4.3) – (4.7) are used in this formulation. As only a few things change between the different models formulated in this section, only what changes, or what is new is shown in the formulation. The full model can be found in Appendix C.1. The model is formulated as before, and we start by defining the new variables:

$G_c \triangleq$ the goal for the the cost objective.
$G_s \triangleq$ the goal for the the strength objective.
$d_c \triangleq$ the cost deviational variable.
$d_s \triangleq$ the strength deviational variable.
$d_c^N \triangleq$ the normalised cost deviational variable.
$d_s^N \triangleq$ the normalised strength deviational variable.
$w_c \triangleq$ the cost weighting variable.
$w_s \triangleq$ the strength weighting variable.

The model is then formulated with the new objective function:

$$\min z_d = w_c d_c^N + w_s d_s^N - 10^{-12} z_c + 10^{-7} z_s \tag{4.30}$$

This objective function is subject to these additional constraints:

$$\sum_{i \in \boldsymbol{I}} x_i c_i - d_c = G_c \tag{4.31}$$

$$\sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} + d_s = G_s \tag{4.32}$$

$$d_c^N = \frac{d_c}{G_c} \tag{4.33}$$

$$d_s^N = \frac{d_s}{G_s} \tag{4.34}$$

The objective function (4.30) seeks to minimise the weighted sum of the deviations from the goals for cost and strength, while ensuring an efficient point is reached. Constraints (4.31) and (4.32) are the soft constraints that bring the deviational variables into the model. Because the cost goal is the lowest possible attainable cost the actual cost will always be greater than the goal, which is why the deviational variable is subtracted, the opposite is true for the strength deviational variable. Constraints (4.33) and (4.34) normalise the deviation variables. The remaining constraints are as before.

When solving the models there are three things that are important to note; the total cost, the total strength and the amount of each of the coal types $i \in \boldsymbol{I}$ that are in the blend. To start with, one must solve for cost minimisation and strength maximisation independently, as the results from these will be used in the goal programming model as the goal values, G_c and G_s. Solving for cost minimisation and strength maximisation, the model produces two independent solutions; one is the *cost optimal blend* and the other is the *strength optimal blend*. These solutions can be found in Table 4.1. It must be noted that the cost optimal blend is significantly cheaper than the current blend being used by AMSA, while also having a better strength value, it is R485 364 cheaper than the current blend, and the strength value is 5.26 units higher. A full comparison of the different blends will be done in the next chapter.

From these results one gets a starting point for the goal programming model, as initial goals are needed, and the optimal cost and strength values are used as the goals. Once the goals have been set, which are constraints (4.31) and (4.32), a Monte Carlo simulation approach is used to randomly generate values for w_c and w_s, which determines the hierarchy of importance of each of the objective functions.

The weights that are generated are uniformly distributed between 0 and 100, and they are integers. By sampling for the w_c and w_s, and solving the model using Monte Carlo simulation, with 10^4 iterations, we develop various scenarios in an approach known as

Table 4.1: Cost and strength optimal solutions using AMSA's expected values.

			Amount of coal type i in blend (tons)								
Blend type	Cost	Strength	1	2	3	4	5	6	7	8	9
Cost optimal	8 353 060	61.98	0	0	1 002	32	0	1 902	0	0	60
Strength optimal	11 143 322	87.96	1 231	413	0	30	0	1 322	0	0	0

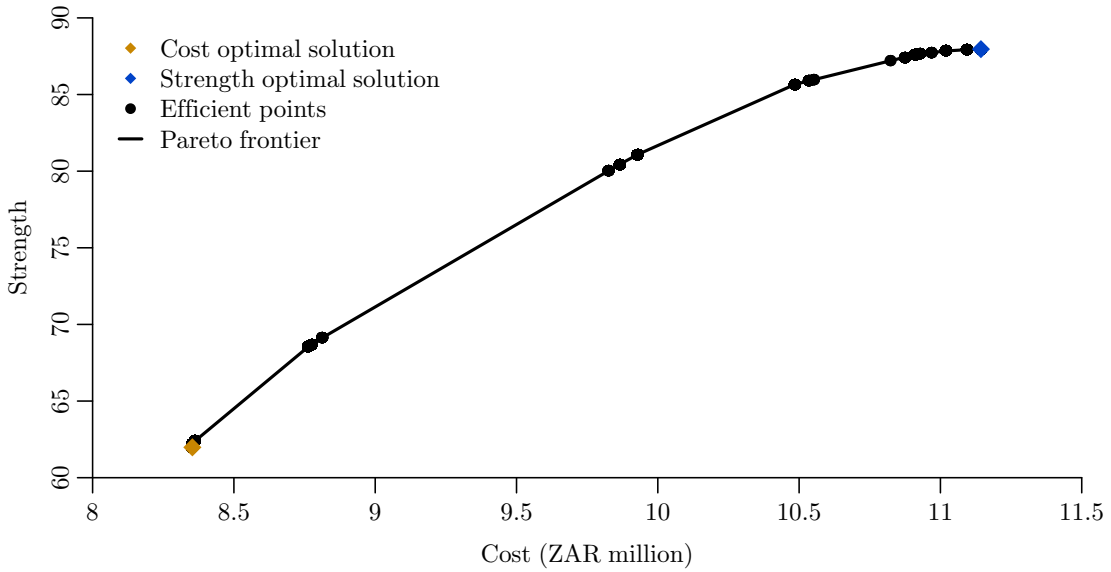


Figure 4.1: Pareto frontier and optimal solutions of the AMSA expected value goal programming model.

preemptive goal programming. These scenarios would be in the form of model solutions, and they are dependant on the weighting of each of the objectives. This results in a set of efficient points of the model, and a Pareto optimal frontier can be formed from these points, this can be seen in Figure 4.1. Plotted in this figure are all the efficient points, the cost and strength optimal solutions, and the Pareto frontier that is formed.

Interestingly enough, there are only a relatively small number of different scenarios that emerge through the simulation of the different weights. The details of each of these scenarios can be found in in Table D.1 in Appendix D. There are a total of 21 different scenarios that occur, and they seemingly occur in small clusters around the *elbows* of the frontier curve.

What this curve shows is that there is a non-linear relationship between the cost of the blend and the strength of the blend, however, there are different segments in which the relationship seems linear, particularly towards the left of the curve. This is of particular importance as these linear relationships will determine how cost is related to strength. Across the entire curve there is a directly proportional relationship between cost and strength — by increasing the cost of the blend the strength of the blend will increase. In the first segment of the curve, from around R8 300 000 to R8 850 000, the gradient of the frontier curve is the steepest, meaning that in this cost range the strength of the blend can be increased for a cheaper price, when compared to other sections of the frontier. The gradient in this range is approximately 64 000, this means that to increase the strength

by one unit, the marginal cost is approximately R64 000. Compare this with the next cost range, which is roughly R8 850 000 to R9 900 000, where the marginal cost is roughly R93 000. This is of particular importance to AMSA as it shows the relationship between cost and strength; and it will aid in determining a blend to use based on both *cost* and *strength* as the decision maker can just change the weightings according to the scenario they would want. The Pareto frontier will be further explored with the preemptive optimisation model and the weighted sum of objective model.

Weighted sum of objectives using AMSA's expected values

The model that will be formulated in this subsection is very similar to that which is formulated in the previous subsection. Rather than minimising the deviations from the goal functions, the model will now try to maximise a composite objective function which is a weighted sum of the cost minimisation and strength maximisation objective functions. It is noted that this intervention is generally used as an aid in goal programming, but it is attempted as an independent intervention here.

The model is mostly the same as the goal programming model, with the exclusion of constraints (4.31) – (4.34), and the variables related to deviation. There are no new variables or constraints introduced, and the objective function of the model is as follows:

$$\max z_{c,s} = w_s \frac{\sum_{i \in \boldsymbol{I}} x_i s_i}{t G_s} - w_c \frac{\sum_{i \in \boldsymbol{I}} x_i c_i}{G_c} \tag{4.35}$$

The objective function (4.35) seeks to maximise a composite sum of the two objectives of the model. This objective function is normalised by dividing the objectives by their goal values, G_s and G_c, which were determined in the previous model formulation. The weights, w_s and w_c, as before, are the values that either the decision maker determines, or are the simulated values. These weights assigned the priority in which the objective function must be solved. The full model formulation can be found in Appendix C.2.

The results from this model, both when solving for cost and strength independently and when developing an optimal frontier through simulation, are almost identical to the goal programming model. As such, this intervention will not be used as an independent intervention going forward seeing as though the results are practically the same. Goal programming is chosen over this intervention as it gives more control to the decision maker. It was, however, interesting to note the similarities in the results, and the results from solving this model, as done in the goal programming intervention, can be found in Table D.2 in Appendix D. The final intervention to deal with multi-objectivity is that of preemptive optimisation.

Preemptive optimisation using AMSA's expected values

This approach is based around the decision maker setting one of the objective functions to be a certain value, they *preempt* the value of the most important objective function. By preempting the objective function value it ensures that the most important objective is met, and the other objectives are met as close as possible to their optimal value without worsening the other preempted objective.

This approach is useful as it gives a lot of control of the solution to the modeller as they would choose what the value of one objective would be, and the solution developed accordingly. This can be very useful to generate different scenarios in a *"what-if"* approach to the problem, where the decision maker can ask the question: *"If the cost of the blend is this, what would the strength be?"* This will be handy when working with a fixed budget,

for example. The easiest way to do this would be to develop different scenarios through simulating the value of the preempted objective, rather than the decision maker having to make up scenarios to test.

The preemptive optimisation model is much the same as the original expected value model, with the objective functions remaining in the same form, the cost is just modelled as a constraint, and there are two new variables introduced. The variable introduced for the simulation of the preempted objective is $C_{increment}$, which is the increment that the cost is increased by every time the model is solved. The variable G_c is again used here as the staring value of the cost objective, and it would be the cost optimal solution. Basically, the model will be solved 10^4 times, each time with a slightly higher cost than before. This should produce an efficient frontier and give the decision maker many different scenarios to choose from, with very little manual effort. The model is then formulated as follows:

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{4.36}$$

subject to:

$$\sum_{i \in \boldsymbol{I}} x_i c_i \leq G_c + C_{increment} \tag{4.37}$$

The objective function (4.36) seeks to maximise the strength of the blend, and it is subject to all the same constraints as the original expected value model. There is an additional constraint, (4.37), which sets the desired preempted value of the cost objective. The right hand side can be changed to any value that the decision maker desires, but it is set like this for simulating the values in a for loop with 10^4 iterations. The full model formulation can be found in Appendix C.3.

The result of the Monte Carlo simulation is 10 000 different solutions to the model. This gives the decision maker more than enough different solutions to choose from, depending on the strength and cost values that are desired. The results of the simulation are plotted in Figure 4.2.

What can be seen from this plot is a Pareto frontier for the objective values. The frontier is made up of 1 514 Pareto optimal solutions, which is when the solution is at such a stage that it cannot be improved without worsening one of the preference criterion (either cost or strength). This means that the decision maker has 1 514 different Pareto efficient solutions to chose from, and as Rardin (2016) refers to, these would be the "best" feasible solutions to the multi-objective model. The remaining points on the plot would be dominated points, which are points below the efficient frontier. Although these points are still feasible solutions to the model, they are not as good as the non-dominated efficient points.

Also shown in the plot is is the solution that AMSA are currently using, with a cost of R8 838 424 and a strength value of 56.73. This point falls well below the efficient frontier, and even the other dominated points; this highlights how inefficient the current blend is. At the current cost AMSA are incurring, the blend should have a minimum strength of 69.2, substantially better than the current value of 56.73.

The Pareto frontier that is shown in this plot is similar to the one which was developed through goal programming, shown in Figure 4.1. There are seemingly different *'segments'* of the curve where the relationship seems linear, and once again, the segment with the steepest gradient is the first section, roughly between R8 300 000 and R8 850 000, and this is where the most favourable relationship between cost and strength occurs. This cost range is the most important to AMSA, and as such, it is explored further by plotting

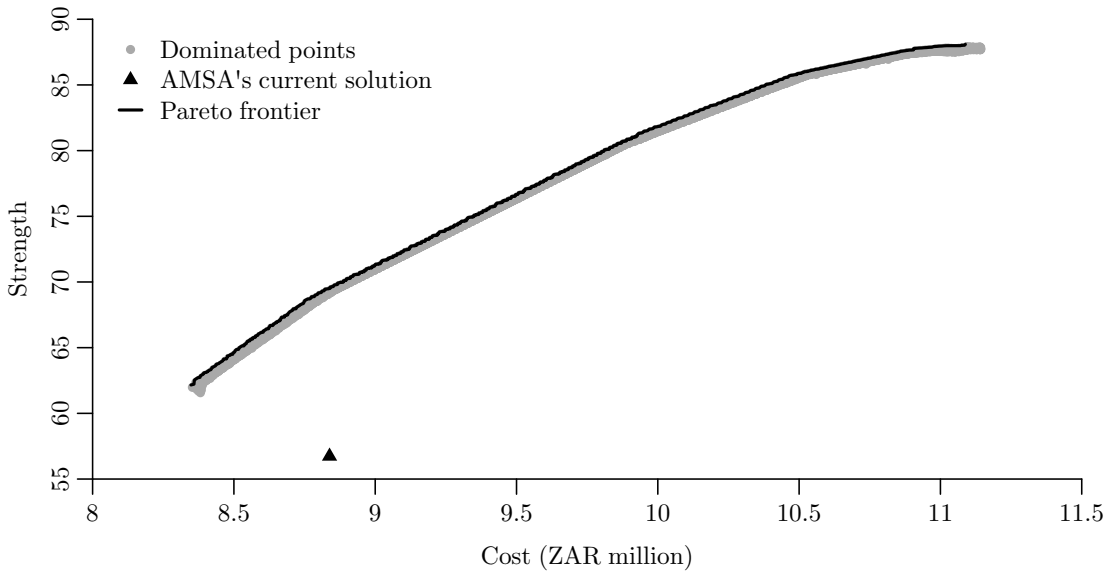


Figure 4.2: Pareto frontier of the AMSA expected value preemptive optimisation model.

a similar figure as before, but now focussing on the specific cost range with the most favourable cost-strength relationship. This plot can be seen in Figure 4.3.

In this price range there are 139 efficient points, out of a total of 1 712, to the model, and these points would be the best for AMSA to use, as they are almost all cheaper than the current solution, with all of them having a higher strength, and as such, they must be considered when determining the optimal blend. This plot highlights the directly proportional relationship between cost and strength; as cost increases so too does strength. We want the lowest cost and highest strength, which is an inversely proportional relationship, however, the relationship is directly proportional. This is where the competition between the cost and strength objectives comes from. Also highlighted in the figure is AMSA's current solution, which is woefully inefficient when compared to the Pareto optimal points plotted, and even the dominated points of the model. This means that the model's worst scenario, when comparing *both* cost and strength, is better than the best solution that is currently available.

The three interventions to handle multiple objectives in expected value models using AMSA's expected values have all been successful, showing positive results, with some solutions providing up to R485 364 in daily savings, which equates to a possible R177 million annual saving. Numerous scenarios that the decision makers can choose from were also developed. The similarity in results of the weighted sum of objectives approach to goal programming is noted, and as such, it will not be used going forward in developing models as part of the solution set. The goal programming and preemptive optimisation interventions will be used in the next section, where the estimated expected values of the parameters are used rather than the values that AMSA expect.

4.1.2 Estimated expected value model

Much the same as the expected value models that were formulated using AMSA's expected values for the coal parameters, the models in this section are formulated using expected values as an intervention to deal with the uncertainty present. The expected values used are the *estimated* expected values, and these are estimated from the PDF of each parameter

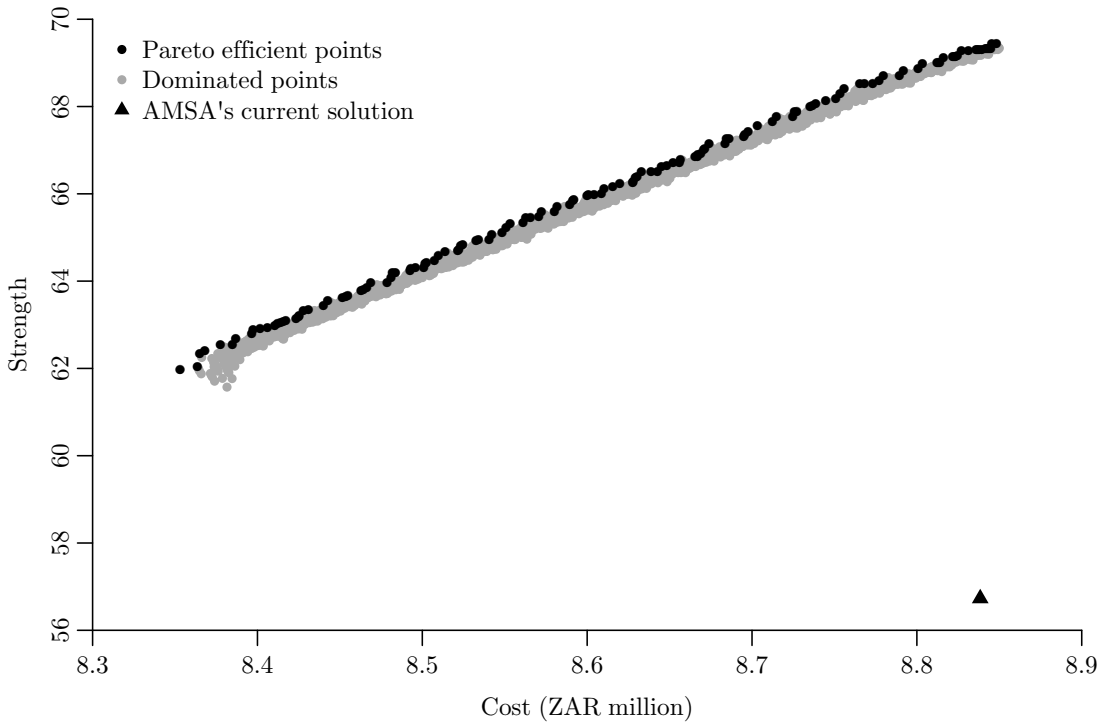


Figure 4.3: Simulated solutions with a cost lower than R8.85 million.

$j \in \boldsymbol{J}$ for coal type $i \in \boldsymbol{I}$. This is an alternate approach to determining the expected values of the parameters, and it is possibly more accurate as it makes use of historical data to estimate the parameters. The expected value of the randomly distributed parameter is calculated as $E(\widetilde{p}_{i,j}) = p^E_{i,j}$, and it is the median value based on the PDF of the respective coal type and parameter. The estimated parameters are used as data in the models formulated in this section, and can be found in Table B.2 in Appendix B. The data for the cost and strength are as before.

The two interventions to deal with multi-objectivity in this section are *goal programming* and *preemptive optimisation*. These interventions are applied to the estimated expected value model, which is formulated by letting $\boldsymbol{I}$ be the set of coals, such that $i \in \boldsymbol{I} = \{1, \ldots, 9\}$, and $\boldsymbol{J}$ be the set of coal parameters, such that $j \in \boldsymbol{J} = \{1, \ldots, 12\}$.
The model parameters are defined as follows:

$$
\begin{aligned}
x_i &\triangleq \text{ the amount (tons) of coal type } i \in \boldsymbol{I} \text{ in the blend.} \\
c_i &\triangleq \text{ the cost (R/ton) of coal type } i \in \boldsymbol{I}. \\
s_i &\triangleq \text{ the expected strength measure of coal type } i \in \boldsymbol{I} \text{ after coking.} \\
p^E_{i,j} &\triangleq \text{ the estimated parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
L^{\min}_j &\triangleq \text{ the lower limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
L^{\max}_j &\triangleq \text{ the upper limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
b_i &\triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases} \\
d_i &\triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases} \\
t &\triangleq \text{ the total amount (tons) of coal in the blend.}
\end{aligned}
$$

The model is then formulated:

$$\min z_c = \sum_{i \in \boldsymbol{I}} x_i c_i \tag{4.38}$$

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{4.39}$$

subject to:

$$t = 2\,996 \tag{4.40}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{4.41}$$

$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^E}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{4.42}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{4.43}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{4.44}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{4.45}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{4.46}$$

$$x_9 \leq 60 \tag{4.47}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{4.48}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{4.49}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{4.50}$$

All the constraints are as before, with the only difference being constraint (4.42), which uses Equation (4.16) as an estimate of the randomly distributed parameters. This model is used as the base model for the next two subsections. The model is now a deterministic equivalent of the stochastic multi-objective model, and the two interventions to handle the multi-objectivity follow.

Goal programming using estimated expected values

This model is formulated using the goal programming approach as is done for the goal programming model that makes use of AMSA's expected values, the only difference between that model and this one is the parameters used, the formulation is the same. This model makes use of the estimated expected values by taking the median of all the respective PDF's. The entire model formulation is in Appendix C.4.

As in the previous expected value model formulation, the solution is found by first determining the goals of the respective objective functions. This is done by solving the model independently for cost and strength, to find the cost and strength optimal solutions, and the values generated there are used as G_c and G_s. The results from generating a cost optimal and strength optimal solutions can be found in Table 4.2.

The value of G_c is 10 069 798 and G_s is 85.88. These are worse when compared to the cost optimal and strength optimal solutions of the expected value model that uses AMSA's values. This is attributed to the fact that the estimated expected values are, in general, worse than the AMSA expected values. Worse meaning the parameters with upper limits are higher and the parameters with lower limits are lower.

Table 4.2: Cost and strength optimal solutions using estimated expected values.

			Amount of coal type i in blend (tons)								
Blend type	Cost	Strength	1	2	3	4	5	6	7	8	9
Cost optimal	10 069 798	69.88	1 783	471	682	0	0	0	0	0	60
Strength optimal	11 622 322	85.88	1 424	978	0	0	32	562	0	0	0

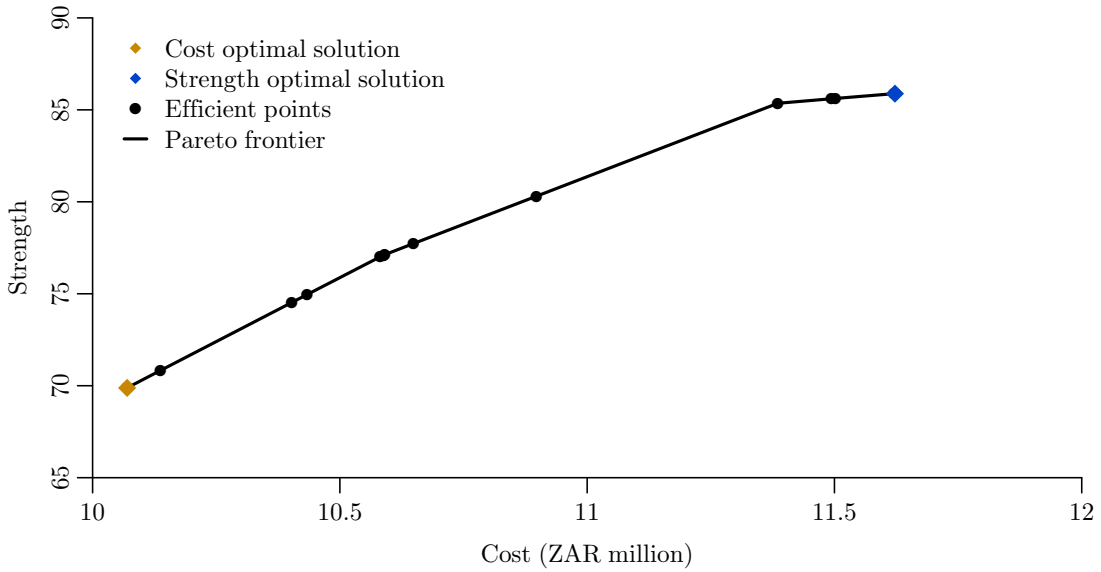


Figure 4.4: Pareto frontier and optimal solutions of the estimated expected value goal programming model.

As before, a Monte Carlo simulation, where the weights, w_c and w_s, are randomly generated, with values being uniform between 0 and 100. There are 10^4 iterations of the simulation, and the result is different scenarios for cost and strength which are dependant on the weight of each objective function. The resulting scenarios can be seen in Table D.3 in Appendix D. Using these scenarios, one can plot them to determine the Pareto frontier, and this can be seen in Figure 4.4.

The curve has a similar shape to that of the AMSA expected value goal programming model, and the directly proportional relationship between cost and strength is visible here too. There also seems to be definite sections in this curve where the cost-strength relationship is linear, and there are three different segments, each with different gradients. The first segment is the cost range under around R10.6 million, the second is up to approximately R11.4 million, with the last segment being from R11.4 million upwards. Again, the first segment would be the most important for AMSA to focus on as it has the steepest gradient, which means this is the cost region where it is cheapest to increase strength by one unit. It would cost an approximate R130 000 to increase the strength by one unit in this cost range, compared to an approximate R220 000 in the next cost range. This curve is useful as it shows the cost-strength relationship and it aids the decision maker in making a decision about the blend based on both *cost* and *strength*. The next intervention for the estimated expected value model is that of preemptive optimisation.

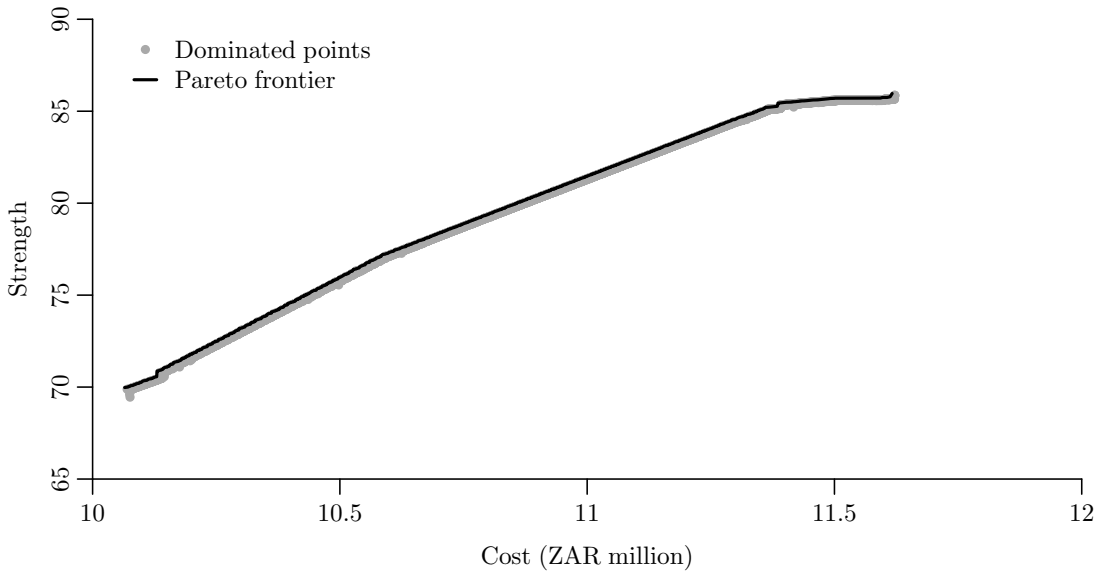


Figure 4.5: Pareto frontier of the estimated expected value preemptive optimisation model.

Preemptive optimisation using estimated expected values

The model which is formulated here follows the same steps when doing preemptive optimisation for the expected value model using AMSA's estimates. The difference in the models lie in the parameters, as they are now the estimated expected values. The entire formulation of this model can be found in Appendix C.5.

The cost objective is still considered the most important, and that is why it is solved for first, and then its value preempted after that. As before, the model will be solved 10^4 times, each time with an incrementally increasing value for the cost objective. In the first iteration of the model, the cost objective is set equal to R10 069 798, and each iteration after that it increases by R155. Doing this will generate different scenarios, each with a different cost and strength. When plotted, these scenarios form a Pareto frontier, and this can be seen in Figure 4.5

This curve has much the same shape as that in Figure 4.4. All of the points plotted are feasible solutions to the model, but there are 1 277 non-dominated solutions, which are the "best" feasible solutions to the model. They are the points that provide the best compromise between the conflicting objectives of cost minimisation and strength maximisation.

To explore the cost range that would be most important to AMSA, that which is below R10.6 million, we can plot all the feasible solutions to the model in that range. This is shown in Figure 4.6

In this cost range, there are 463 Pareto efficient, or non-dominated, points, which leaves the decision maker with the 463 "best" solutions to the model. There are two very distinct sections in the data plotted in this figure, as there is a substantial jump in the strength, for a very small increase in cost. At this jump in the data the strength can be increased by 0.32 units for only R900. The reason for this jump must be explored as it could possibly be replicated in other situations to increase the strength for a much lower cost.

Using the estimated expected values for the coal parameters is seemingly more expensive and produces less desirable solutions when compared to using the values that AMSA

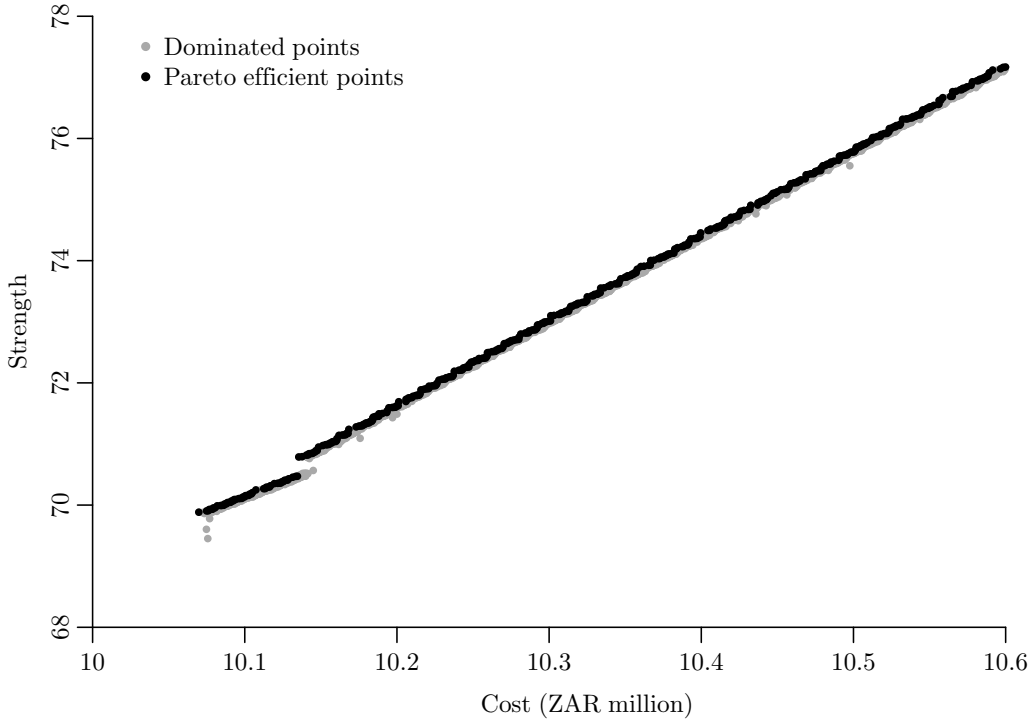


Figure 4.6: Solutions to the estimated expected value preemptive optimisation model under R10.6 million.

expect. This may be the case, but using the estimated expected values is more conservative, and it possibly paints a more honest picture of the problem as there would be less risk of violating the constraints, this is because historical data is used to estimate the values. The balance between risk and caution must found, and the reliability and accuracy of these models will be evaluated in the next chapter.

Solving the model by using the expected value approach produced some very positive and interesting results. Using the expected values of the parameters is a rather rudimentary approach to a problem that has inherent uncertainty present. By ignoring the uncertainty one fails to fully capture the essence of the problem, and it is for this reason that the models are reformulated using stochastic programming interventions as the uncertainty will be taken into account. This is done in the next section.

4.2 Stochastic programming models

Stochastic programming is an approach to mathematical modelling and optimisation that considers uncertainty. As highlighted previously, the different coal parameters are all randomly distributed, meaning they are uncertain. Previously, either the estimated expected values or the AMSA expected values were used in the model as deterministic values, which made the model deterministic. In this section, stochastic programming is used to convert the stochastic model to its deterministic equivalents, while still making use of the multi-objective interventions of goal programming, and preemptive optimisation.

It is computationally very demanding [1] to solve the models in this section, and as such,

[1] On a device with an Intel Core i7 processor and 8GB RAM, one instance of the stochastic model takes upwards of 8 hours to solve using LINGO version 18.0.

the multi-objective interventions are used, but in a much simpler way. In the previous sections Monte Carlo simulation was used to solve the model thousands of times, but due to computational limitations, the number of iterations will be drastically reduced, this is just to illustrate how the different interventions work. The multi-objective interventions are still needed, as the model still has the two objectives of cost minimisation and strength maximisation. They are applied in such a way that shows that multi-objective and stochastic programming interventions are not mutually exclusive interventions, but rather, they can be used together to solve problems that have *both* uncertainty and multiple objectives, like the problem of coal blending for coke-making.

There are two main interventions to stochastic programming; CCP and recourse programming. Recourse programming is ideal for a two-stage problem, and the fundamental idea of it is the concept of recourse. Recourse is the ability to take corrective action after a random event has happened or after uncertainty has been realised. The benefits of recourse programming are noted, but the problem at this stage of the project is only a one-stage problem, with no recourse available, and as such, this intervention is not used. Recourse programming will be used when the project is taken further, as it will then include more than one stage.

The second intervention in stochastic programming, CCP, is based around ensuring the probability of meeting a constraint is above a certain level. If the problem is in the form below, the CCP approach can be used:

$$\min f(x) \quad \text{subject to} \quad \Pr\left(F(x,\tilde{\xi}) \leq 0\right) \geq 1-\alpha \tag{4.51}$$

This is the general form that Nemirovski and Shapiro (2006) propose for CCP. The objective function is arbitrary, but it must be a real valued, convex function; and the constraint that it is subject to is $F(x,\tilde{\xi}) \leq 0$. The constraint is dependant on the decision variable, x, and the random vector, $\tilde{\xi}$, which can be described by some PDF. The stochastic nature of the problem arises from $\tilde{\xi}$ being randomly distributed. $\Pr\left(A\right)$ denotes the probability of event A happening, and $\alpha \in (0,1)$. The constraints says that we must ensure that the constraint $F(x,\tilde{\xi}) \leq 0$ is met with a probability value of at least $1-\alpha$, and it known as a *chance constraint*.

The problem in this project can be formulated into the form as shown in equation (4.51). Because all the coal parameters are randomly distributed and can be described by some PDF, the constraint that ensures the parameters are above or below a certain level can be formulated in the same manner as above, and this puts the model in the general form that is required. Objective functions (3.1) and (3.2), and constraint (3.5) can be formulated as follows, with $\boldsymbol{I}$ being the set of coals, such that $i \in \boldsymbol{I} = \{1,\ldots,9\}$, and $\boldsymbol{J}$ being the set of coal parameters, such that $j \in \boldsymbol{J} = \{1,\ldots,12\}$:

$$\min z_c = \sum_{i\in\boldsymbol{I}} x_i c_i \tag{4.52}$$

$$\max z_s = \sum_{i\in\boldsymbol{I}} \frac{x_i s_i}{t} \tag{4.53}$$

subject to:

$$\Pr\left(L_j^{\min} \leq \sum_{i\in\boldsymbol{I}} \frac{x_i \widetilde{p}_{i,j}}{t} \leq L_j^{\max}\right) \geq 1-\alpha \qquad \forall j \in \boldsymbol{J} \tag{4.54}$$

The objective functions (4.52) and (4.53) are still cost minimisation and strength maximisation, respectively. These objective functions are still subject to all the other

constraints, along with constraint (4.54), which says that the probability of the blend parameter $j \in \boldsymbol{J}$ being adhered to is $1 - \alpha$.

Because the model is now in the correct general form it can be converted to a deterministic equivalent by using one of two approaches; either the *approximation* approach or the *scenario* approach.

4.2.1 Chance-constrained Programming: Scenario approach

One of the approaches to CCP, which is proposed by Nemirovski and Shapiro (2006) and illustrated by Joubert (2019), is the scenario approximation approach. This approach is based on Monte Carlo sampling techniques where a sample, $\xi^1, \ldots, \xi^N$, of N independent realisations of the random vector ξ is generated. In the case of this problem, the sample would be $p_{i,j}^1, \ldots, p_{i,j}^N$ for all the coal types $i \in \boldsymbol{I}$ and repeated for each of the parameters $j \in \boldsymbol{J}$. N, which is the sample size, is calculated as follows:

$$N = \frac{2n}{\alpha} \ln\left(\frac{12}{\alpha}\right) + \frac{2}{\alpha} \ln\left(\frac{2}{\delta}\right) + 2n \tag{4.55}$$

Where $1 - \alpha$ is the probability level the constraint must be met with, $1 - \delta$ is the probability that the optimal solution to the problem is feasible, and n is the number of random variables in the single constraint. In our case, we want to be 95% certain that the blend parameter is adhered to, 99% sure that the optimal solution is feasible, and there are 9 random variables (as there are 9 coal types) in each of the constraints. This means $\alpha = 0.05$, $\delta = 0.01$ and $n = 9$, leading to a sample size of 2 203 for this problem, and this is the sample size for all 12 parameters. These values are used throughout.

By using the scenario approach, the general form in (4.51) becomes:

$$\min f(x) \quad \text{subject to} \quad F(x, \xi^\nu) \leq 0, \nu = 1, \ldots, N \tag{4.56}$$

Using this converted form of the general form, the multi-objective stochastic model can be formulated as a deterministic equivalent. We start by letting $\boldsymbol{I}$ being the set of coals, such that $i \in \boldsymbol{I}$, $\boldsymbol{J}$ being the set of coal parameters, such that $j \in \boldsymbol{J}$, and $\nu = 1, \ldots, N$, which is the set of samples. The model is then formulated:

$$\min z_c = \sum_{i \in \boldsymbol{I}} x_i c_i \tag{4.57}$$

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{4.58}$$

subject to:

$$L_j^{l\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^\nu}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J}, \nu = 1, \ldots, N \tag{4.59}$$

This model is now deterministic, and it has the objectives of cost minimisation, (4.57), and strength maximisation, (4.58). It is still subject to all the original constraints, along with constraint (4.59). This constraint ensures that the blend parameter $j \in \boldsymbol{J}$ is adhered to for all N scenarios by sampling from the sample set $p_{i,j}^1, \ldots, p_{i,j}^N$. What this means is that there is no longer one constraint per parameter, there are now N constraints that are the same, but the parameter values, $p_{i,j}$, in those constraints are sampled from $p_{i,j}^1, \ldots, p_{i,j}^N$, and this is illustrated below with $N = 2\,203$.

$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^1}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J}$$
$$\vdots \qquad\qquad \vdots$$
$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^{2203}}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J}$$

Here the constraint is repeated 2 203 times for all 12 of the different parameters, $j \in \boldsymbol{J}$, and $p_{i,j}$ is sampled from the sample set $p_{i,j}^1, \ldots, p_{i,j}^{2203}$ for all parameters $j \in \boldsymbol{J}$. The sample is developed by randomly generating N values from each coal type's PDF, and this is repeated for all the different parameters.

Because the model has been converted to a deterministic equivalent of the original stochastic model, it can now be formulated with the same multi-objective programming interventions as before. Both the preemptive optimisation and goal programming interventions are applied to the scenario approximation approach model.

Goal programming for scenario approach

By using the same steps as for the expected value models one can formulate a goal program for the model generated through the scenario approach. The model is formulated by letting $\boldsymbol{I}$ being the set of coals, such that $i \in \boldsymbol{I}$, $\boldsymbol{J}$ being the set of coal parameters, such that $j \in \boldsymbol{J}$, and $\nu = 1, \ldots, N$, which is the set of samples. There are additional variables that are defined for the model:

$G_c \triangleq$ the goal for the the cost objective.
$G_s \triangleq$ the goal for the the strength objective.
$d_c \triangleq$ the cost deviational variable.
$d_s \triangleq$ the strength deviational variable.
$d_c^N \triangleq$ the normalised cost deviational variable.
$d_s^N \triangleq$ the normalised strength deviational variable.
$w_c \triangleq$ the cost weighting variable.
$w_s \triangleq$ the strength weighting variable.

The model is then formulated with goal programming objective function:

$$\min z_d = w_c d_c^N + w_s d_s^N - 10^{-12} z_c + 10^{-7} z_s \tag{4.60}$$

This objective function is subject to these additional constraints:

$$\sum_{i \in \boldsymbol{I}} x_i c_i - d_c = G_c \tag{4.61}$$
$$\sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} + d_s = G_s \tag{4.62}$$
$$d_c^N = \frac{d_c}{G_c} \tag{4.63}$$
$$d_s^N = \frac{d_s}{G_s} \tag{4.64}$$
$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^\nu}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J}, \nu = 1, \ldots, N \tag{4.65}$$

Table 4.3: Cost and strength optimal solutions using scenario approach.

			Amount of coal type i in blend (tons)								
Blend type	Cost	Strength	1	2	3	4	5	6	7	8	9
Cost optimal	10 961 528	74.15	95	1 703	212	159	0	731	0	36	60
Strength optimal	11 217 900	74.53	149	1 787	228	139	0	693	0	0	0

Table 4.4: Goal programming scenarios using scenario approach.

				Amount of coal type i in blend (tons)								
w_c	w_s	Cost	Strength	1	2	3	4	5	6	7	8	9
1	1	11 006 702	74.49	133	1 772	213	114	0	704	0	0	60
2	1	10 962 317	74.16	96	1 704	212	158	0	731	0	35	60
1	2	11 007 326	74.49	134	1 773	213	113	0	703	0	0	60

Objective function (4.60) seeks to minimise the normalised deviations from each objective function. In the objective function is the addition and subtraction of very small multiples of the original objective function, this is done to ensure that an efficient solution is reached. Constraints (4.61) and (4.62) bring the deviational variables into the model while setting the goals of each objective function, the deviational variables are then normalised in constraints (4.63) and (4.64). Constraint (4.65) is the scenario approximation constraint, and it ensures the blend parameter $j \in \boldsymbol{J}$ is adhered to for all N scenarios by sampling from the sample set $p^1_{i,j}, \ldots, p^N_{i,j}$, and in this case, $N = 2\,203$. All the other constraints still remain for model, found in Appendix C.7, which can now be solved.

As a starting point in solving goal programs, the goals have to be determined, and this is done by solving the model for each objective function independently. The results of this can be seen in Table 4.3.

It can be seen that these solutions are less desirable than before, as the cost is significantly higher in the cost optimal solution, and the strength is a lot lower when compared to previous solutions. This is attributed to the fact that the scenario approach is often very conservative, and as a result, generally produces less desirable solutions as a trade-off for reliability (Nemirovski and Shapiro, 2006).

Nonetheless, these values are used as the goal values, G_c and G_s, when solving for the goal programming model. Only three different weighting scenarios will be used, due to computational limitations. One scenario is when the objectives are treated with equal importance, the second scenario is when cost is treated as twice as important as strength, and vice versa for the third. The results of this can be seen in Table 4.4.

As before, one could use a Monte Carlo simulation to simulate the weight values, and from this, develop a Pareto frontier and determine what the efficient points are. A similar approach can be applied when using preemptive optimisation with the scenario approximation approach, which follows.

Preemptive optimisation for the scenario approach

As illustrated, multi-objectivity and uncertainty are not mutually exclusive events, and neither are the interventions to deal with them. In this section, preemptive optimisation

is combined with the scenario approach to deal with uncertainty and multi-objectivity simultaneously. The full model can be found in Appendix C.8.

The steps followed are the same as before when formulating a preemptive optimisation model, with the only difference being that no Monte Carlo simulation, to incrementally increase the preempted value of one of the objectives, will be done here. The cost objective is considered to be the most important, and as such, it will be the preempted value in the model. Shown in Table 4.5 are the results of five preempted cost values. If one wanted to determine an efficient frontier, and which solutions in the cost range would be the "best" to use, more points would be needed, and this could easily be achieved through incrementally changing the preempted cost value, as was done with the expected value models, but that is not the focus here.

Table 4.5: Preemptive optimisation scenarios using scenario approach.

		Amount of coal type i in blend (tons)								
Cost	Strength	1	2	3	4	5	6	7	8	9
10 961 528	74.15	95	1 703	212	159	0	731	0	36	60
11 183 318	74.40	125	1 742	231	137	0	731	30	0	0
10 970 872	74.19	101	1 715	212	154	0	724	0	30	60
11 021 320	74.48	136	1 772	215	113	0	705	0	0	55
11 217 900	74.53	149	1 787	228	139	0	693	0	0	0

The scenario approximation approach is a very conservative approach to the problem, as is evident by the results produced from the models. It is however, very useful in the sense that it can take, as input into the model, any distribution function and use it in the model, as it merely samples out of that distribution. This is particularly beneficial for this problem as there are a number of "not-so-nice" distributions, such as the lognormal, Weibull and Gamma distributions. An alternate approach to the scenario approach in CCP is the approximation approach.

4.2.2 Chance-constrained Programming: Approximation approach

This approach aims to approximate the chance-constrained problem, and the approximation is based on analytical bounding of the probability of the randomly distributed constraint, and this is discussed by Nemirovski and Shapiro (2006). Joubert (2019) also discusses and illustrates this approach.

It follows from the scenario approach in that there must be a chance constraint in the general form of (4.56) to be able to use this approach. The constraint to the model, shown below, is in the general form, which means this intervention can be used.

$$\Pr\left(L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i \widetilde{p}_{i,j}}{t} \leq L_j^{\max}\right) \geq 1 - \alpha \qquad \forall j \in \boldsymbol{J} \tag{4.66}$$

Once a constraint is in the general form, it can be approximated by:

$$f_0(x) + \sum_j \tilde{\xi}_j{}' f_j(x) \leq 0 \tag{4.67}$$

Where $f_0(x)$ is the deterministic component of random vector $\tilde{\xi}$, and $\tilde{\xi}_j{}'$ is the remaining stochastic component of the random vector $\tilde{\xi}$. It is imperative that the remaining

random variables, $\tilde{\xi_j}'$, have zero means; this implies that only distributions that are centred at their mean can be used in this approach (Nemirovski and Shapiro, 2006). Distributions that are centred at their mean are generally the "nice", rotationally invariant distributions; such as the uniform, normal and logistic distributions, for example.

This approach involves splitting the stochastic variables into their deterministic components so that the remaining stochastic components are rotationally invariant around zero (Joubert, 2019). And this is given by the general form of:

$$f_0(x) + \Omega\left(\sum_j \sigma_j^2 f_j^2(x)\right)^{\frac{1}{2}} \leq 0 \tag{4.68}$$

Where σ_j^2 is the variance of the randomly distributed parameter. Ω is what is known as the *safety factor*, and it is calculated as follows:

$$\Omega = \sqrt{2\ln\left(\frac{m}{\alpha}\right)} \tag{4.69}$$

Where $1-\alpha$ is the desired probability that the constraint is met with, and m is the number of times the random variable appears. In the case of this problem, we are working with an α value of 0.05 and $m = 1$, thus $\Omega = 2.448$.

The general form in (4.68) can be applied to the chance constraint in (4.66) with random variable $\tilde{p}_{i,j}$. The resulting constraint would look as follows:

$$tL_j^{\min} \leq \sum_{i\in\boldsymbol{I}} x_i E(\widetilde{p}_{i,j}) + \Omega\left(\sum_{i\in\boldsymbol{I}} x_i^2\sigma_{i,j}^2\right)^{\frac{1}{2}} \leq tL_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{4.70}$$

The deterministic component of the random variable $\widetilde{p}_{i,j}$ is $E(\widetilde{p}_{i,j})$, which would be the expected value of the distribution. The stochastic component is $\sigma_{i,j}^2$, which is the variance of the random variable $\tilde{p}_{i,j}$ for all coal types $i \in \boldsymbol{I}$ and parameters $j \in \boldsymbol{J}$. The variable t has been removed from the bottom of the fraction, but this is merely for readability. This constraint says that the sum of all the deterministic components plus the square root of the sum of the variance multiplied by the safety factor, summed over all the coal types $i \in \boldsymbol{I}$ and for all parameters $j \in \boldsymbol{J}$, should be between the upper and lower bounds for the specific parameter. This is how one would apply the approximation approach to CCP for this problem.

It is highlighted by both Joubert (2019) and Nemirovski and Shapiro (2006) that this method can only be used when the random variable, in our case $\widetilde{p}_{i,j}$, is symmetrical, or rotationally invariant around the mean, but this is not the case for all the $\widetilde{p}_{i,j}$ distributions. There are five parameters that follow a gamma distribution, one follows a Weibull distribution and four are estimated by a lognormal distribution; none of these are rotationally invariant, and as such, cannot be directly used in the formulation of approximation approach model. The proposed workaround is to set the variance of these specific parameters equal to zero, which would be the stochastic component, but then use an expected value other than the mean for their deterministic component. The $1-\alpha$ percentile will be used as the expected deterministic component, $E(\widetilde{p}_{i,j})$, of the parameters with asymmetrical distributions. The α percentile will be used for parameters who have a lower bound though.

For the parameters whose distributions are rotationally invariant, the deterministic component, $E(\tilde{p}_{i,j})$, will just be the expected value (so, the mean for the uniform and

normal distributions, and the location for the logistic distributions). The stochastic components will be the variance of the distributions, which are calculated as follows for the different distributions:

- Normal distribution: σ^2
- Uniform distribution: $\frac{1}{12}(b-a)^2$
- Logistic distribution: $\frac{1}{3}(s^2\pi^2)$

Using the adapted form of the general form, the multi-objective stochastic model can be formulated as a deterministic equivalent. We start by letting $\boldsymbol{I}$ being the set of coals, such that $i \in \boldsymbol{I}$, $\boldsymbol{J}$ being the set of coal parameters, such that $j \in \boldsymbol{J}$. The new model parameters are defined as:

$$\begin{aligned} E(\widetilde{p}_{i,j}) &\triangleq \text{ the deterministic component of parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\ \sigma^2_{i,j} &\triangleq \text{ the stochastic component of parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \end{aligned}$$

The model is then formulated:

$$\min z_c = \sum_{i \in \boldsymbol{I}} x_i c_i \tag{4.71}$$

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{4.72}$$

subject to:

$$tL_j^{\min} \leq \sum_{i \in \boldsymbol{I}} x_i E(\widetilde{p}_{i,j}) + \Omega \left(\sum_{i \in \boldsymbol{I}} x_i^2 \sigma^2_{i,j} \right)^{\frac{1}{2}} \leq tL_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{4.73}$$

The two objective functions remain the same, the only additional aspect of the model is constraint (4.73). This constraint is the approximation approach constraint, and it ensures that the limits of all the parameters $j \in \boldsymbol{J}$ are adhered to, and are met with 95% certainty. The remaining constraints are as they were before, with the full formulation found in Appendix C.9 The data used in this formulation can be found in Tables B.3 and B.4 in Appendix B. This model is now a deterministic equivalent of the stochastic, multi-objective model, and it can now be solved. As with the scenario approach, both preemptive optimisation and goal programming will be used in conjunction with this model to develop solutions to the problem.

Goal programming for the approximation approach

As with the expected value models, and the scenario approach model, one can apply the concept of goal programming to the approximation approach model. The full model formulation can be found in Appendix C.10. The model is similar to the goal programming for the scenario approach model, with the only difference being constraint (4.65) is replaced with (4.73).

Like any goal programming model, one needs to set goals for the different objective functions, and this is done by solving for each objective independently. The results are the cost optimal and strength optimal solutions, and these are found in Table 4.6

The solutions generated through this approach are somewhat more desirable, as they have a lower cost and higher strength, when compared with the solutions developed using

Table 4.6: Cost and strength optimal solutions using approximation approach.

			Amount of coal type i in blend (tons)								
Blend type	Cost	Strength	1	2	3	4	5	6	7	8	9
Cost optimal	10 739 184	76.07	52	1 370	116	0	0	1042	0	356	60
Strength optimal	11 698 249	81.65	436	1 782	0	0	55	723	0	0	0

Table 4.7: Goal programming scenarios using approximation approach.

				Amount of coal type i in blend (tons)								
w_c	w_s	Cost	Strength	1	2	3	4	5	6	7	8	9
1	1	11 176 864	79.80	0	1 763	35	0	0	1 138	0	0	60
2	1	10 739 247	76.07	42	1 373	114	0	0	1 048	0	359	60
1	2	11 383 466	81.08	199	1 732	0	0	126	879	0	0	60

the scenario approach. There is also a larger range for both the cost and strength, which means there could be more efficient points in the range of feasible solutions to the model. The goal values are $G_c = 10\,739\,184$ and $G_s = 81.65$, and these are used in the goal programming model.

Like with the scenario approach, only a select few scenarios will be found, this is done in place of Monte Carlo simulation to generate random weights. This does mean that no Pareto frontier will be found, but it does show that the goal programming approach can be used in conjunction with CCP to find feasible solutions to a stochastic multi-objective problem. Three scenarios are used, and these determine the weights of the objectives in the goal programming model. The first scenario is when no priority is given to any of the objectives, they are both weighted the same, the second when cost is considered to be twice as important as strength, and vice versa for the third. The solutions to these different scenarios are given in Table 4.7.

These results can be considered more favourable as those produced through goal programming with the scenario approach as the cost is generally lower, and strength higher. In the scenario approach it would cost an average of R675 000 to increase the strength by one unit, whereas using the approximation approach it would cost an average of R172 000. The difference and similarities between the scenario and approximation approaches can be further studied by simulating more values for the weights, it is computationally taxing, but it may be very beneficial as it would allow better insight into both approaches.

Preemptive optimisation for the approximation approach

The final model that is developed is one that uses preemptive optimisation and the approximation approach. The full formulation of the model can be found in Appendix C.11. The same is done here as what was done for the scenario approach; five different cost values are enforced onto the model to see what maximum strength can be returned for the given cost. The same cost values that were used in the scenario approach will be used here too, this is to allow for a more direct comparison of the two. The results can be seen in Table 4.8.

It can be seen from the table that the approximation has far better solutions when

Table 4.8: Preemptive optimisation scenarios using approximation approach.

		Amount of coal type i in blend (tons)								
Cost	Strength	1	2	3	4	5	6	7	8	9
10 961 528	78.09	42	1 608	106	0	0	1 120	0	60	60
11 183 318	79.83	64	1 739	40	0	0	1 093	0	0	60
10 970 872	78.17	31	1 622	104	0	0	1 130	0	49	60
11 021 320	78.56	36	1 652	89	0	0	1 123	0	36	60
11 217 900	80.06	55	1 744	31	0	36	1 070	0	0	60

compared with the scenario approach in Table 4.5. For the same cost, the strengths are consistently higher, some even up to 5.5 units higher. The spread of the different coal types is also interesting to note, as it shows which coal types each of the models favour, for example, the approximation approach model favours coal type 6 when compared to the scenario approach, whereas the scenario approach model favours coal type 3. If more scenarios are developed this type of observation can be studied in further detail, and can be used as an aid in decision making.

This concludes the model development. In total, eight different models were formulated using a combination of different interventions for both uncertainty and multi-objectivity. These models were also solved, and they provided some interesting solutions to the problem; some more favourable than others, but each solution unique in the sense that the different interventions focussed on and used different approaches to the problem. It was also highlighted that stochastic programming and multi-objective programming can be used simultaneously to solve a blending problem. The focus of this chapter was to develop a solution set, and this was done, as all nine models provided more than one solution to the problem — this will aid the decision-maker in blend formulation, and allows for a better understanding of the cost-strength relationship, particularly in the different cost ranges.

In the next chapter, the solutions from the models will be evaluated and validated, with an emphasis being placed on comparison between the current blend, and the modelled solutions. Monte Carlo simulation will be used to test the model reliability and performance. Selected blends will be physically tested at AMSA, and this will be a gauge of model performance too. The testing and evaluation will lead to a refined solution set, and a final solution to the problem of coal blending for coke-making at an integrated steel plant.

Chapter 5

Model evaluation

In this penultimate chapter, the models that were previously formulated are evaluated, and from this, the solution set is finalised. Also, a model will be chosen as the recommended solution that AMSA can use for the problem of coal blending for coke-making, and this model undergoes testing at the pilot coke oven. The models are evaluated on three main criteria; *blend cost*, *blend strength* and *solution reliability*.

For the sake of evaluating the models, a set blend for each had to be determined; this was done by solving the models with given criteria. The criteria of relevance are that of the weightings — used for *goal programming* — and the preempted values, which are used for the *preemptive optimisation* models. In discussion with AMSA, it was determined that the *cost* objective is twice as important as the *strength* objective, and the cost values that are preempted are also chosen according to the model being used. The values are then used to solve the models for a single, optimal instance.

Eight models are solved, along with the current model being used, and when using the relevant weightings of the objectives and the preempted values, the nine subsequent blends are determined and can be found in Table 5.1, with the model numbers explained in Table 5.2. These are the blends that are used to evaluate the models, and these models make up the solution set. To start the evaluation, the reliability of the models are evaluated.

Table 5.1: All modelled blends.

			Amount of coal type i in blend (tons)								
Model No.	Cost	Strength	1	2	3	4	5	6	7	8	9
1	8 836 283	56.73	0	180	899	330	210	748	0	569	60
2	8 354 308	62.23	0	0	1 001	0	30	1 905	0	0	60
3	8 774 988	68.72	0	0	741	0	0	2 165	0	30	60
4	10 069 798	69.88	1 783	471	682	0	0	0	0	0	60
5	10 200 000	71.60	1 738	489	613	0	40	56	0	0	60
6	10 962 317	74.17	96	1 704	212	158	0	731	0	35	60
7	10 970 872	74.19	101	1 715	212	154	0	724	0	30	60
8	10 739 247	76.07	42	1 373	114	0	0	1 048	0	359	60
9	10 970 871	78.17	31	1 622	104	0	0	1 130	0	49	60

Table 5.2: Model number keys.

Model No.	Corresponding model
1	Current model
2	AMSA's expected value goal programming model
3	AMSA's expected value preemptive optimisation model
4	Estimated expected value goal programming model
5	Estimated expected value preemptive optimisation model
6	Scenario approach goal programming model
7	Scenario approach preemptive optimisation model
8	Approximation approach goal programming model
9	Approximation approach preemptive optimisation model

5.1 Model reliability

The reliability of a model can be described as the degree with which the resulting blend can be depended on to be within the set specifications, and how consistent the results are. In our problem, meeting the 12 blend parameters is used as a measure of model reliability. A model will be considered to be 100% reliable if all 12 of its parameters are within the limits (which can be found in Table 3.2, for reference).

To determine whether or not the modelled blend's parameters are within the limits is a relatively simple process, the set blends (which are found in Table 5.1) are used with changing inputs, which would be the randomly distributed parameters of the individual coals. This is done in a Monte Carlo simulation fashion, where the individual coal parameters are sampled out of their Probability Distribution Function (PDF) and are then plugged into the model. This is done for every coal type and every coal parameter and is repeated 10^4 times for each model. This results in 10^4 different instances for each model, and each instance represents a scenario where the input coal parameters are different, this helps us calculate the model's reliability.

If the parameter of the blend is within the limits, that particular parameter is assigned a value of 1, and if not within the limits, it is assigned a value of 0. This leads to there being 12 values, either 1 or 0, which correspond to each coal parameter. These values are then summed together and divided by 12 to give a reliability percentage for each instance of the model being solved. For example, if a modelled instance had 9 of the parameters within the limits, and the remaining 3 not, that particular instance will have a reliability percentage of 75%. The average reliability of all the models, over the 10^4 instances, can be found in Table 5.3. The results are somewhat predictable, with the Chance-constrained Programming (CCP) models having very high reliability and the expected value models having lower reliability.

The measure of a model's reliability, however, does not solely lie with whether or not the target is met; it also depends on how well the target is met and how predictable the reliability and the 12 parameter values are. With this in mind, the reliability of the models is further tested, and three additional factors are considered; the spread of each model's reliability, how much each parameter value deviates from the limit and the variance of each parameter value in the blends.

To calculate the spread of the reliability, the ranges are found — the variance is not

Table 5.3: Average and range of each model's reliability.

Model No.	Reliability	Range
1	0.6238	0.0674
2	0.7062	0.0772
3	0.8374	0.0800
4	0.7836	0.1058
5	0.8381	0.0948
6	0.9998	0.0044
7	0.9998	0.0046
8	0.9992	0.0081
9	0.9993	0.0074

used as the reliabilities do not seem to follow any known distributions — and this is a straightforward calculation. The results can also be found in Table 5.3.

The next test of reliability is how well the target is met. The need for this can be seen in Figure 5.1, which is just one example that shows the distribution of the volatile percentage produced by the nine different models. What the plot shows is that although most of the instances are below the upper limit, some are better than others. Consider Model number 9, this model produces an expected volatile percentage of around 24.5%, and almost every single instance will have a volatile percentage within the limit of 27%. Compare that to Model 8, which has an expected value of 25%, which is still within the limit, however, on average, the instances using that model will be slightly 'worse' when compared to Model 9, and this is attributed to the location of the distribution relative to the upper limit value. It is clear that, on average, Model 9 meets the volatiles target better than when compared to Model 8, and this comparison method is applied to all the models for the different parameters.

To determine how well a model hits the target, the deviation from the limit is calculated. This is done by finding the average deviation from the limit of each parameter value for all the instances of the model. The further below a limit (in the case of an upper limit) the better for an instance, and similarly for when there is a lower limit, the further above the limit the instance value is, the better. The results can be found in Table E.1 in Appendix E.

The deviation values across the different parameters for a model are of varying scales, and to be able to add them without distorting the values, all deviations must be normalised. This is done by dividing each parameter deviation value by the highest value from all the models for each parameter. The normalised values are then summed together for each model to get a sum of the deviations, this value is then normalised again, following the same approach, with the result being found in Table 5.4. Normalising the values is beneficial as it allows us to compare the different models in the same units, as well as allowing us to add different values that were previously in varying scales.

The final factor that is considered when testing reliability is that of the variance, or spread, of each parameter. Similarly to before, having parameter values whose distribution is less spread out makes it more predictable. Consider Figure 5.1 again; note that Models 1 and 3 have similar expected values (28.2% and 27.6% respectively), but Model 3's data is far more spread out when compared to Model 1. This would imply that Model 1 has a more

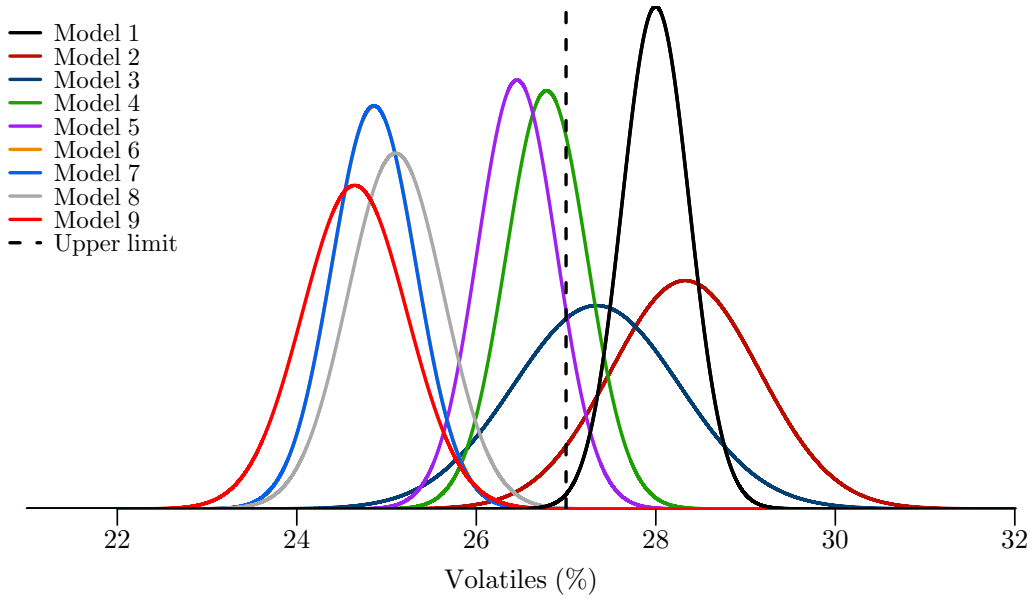


Figure 5.1: Distribution of volatiles percentage in all modelled blends.

predictable volatile value when compared to Model 3. Again, this comparison is carried out for each model across all 12 parameters. The variance of each of the distributions for the parameters is easily calculated as all the modelled parameters are normally distributed, which is concurrent with the *central limit theorem*. The results of these calculations can be found in Table E.2 in Appendix E.

As the same with the deviation values, the variance values are all normalised, following the same method. Each model's total normalised variance is calculated, and then that value is normalised again, with the result being shown in Table 5.4.

The four factors used to evaluate model reliability have all been calculated; they can now be used to determine a final 'reliability' value for each of the models. To be able to add and compare the different factors, they would need to be in the same units and within the same scale; the easiest and most effective way to achieve this is through normalisation. The deviation and variance values have already been normalised, and the same method is applied to normalise the reliability and the range values, with the results being shown in Table 5.4.

Now that all the factors are in the same scaled units, they can easily be added together. Rather than summing the values for each model, a more intuitive approach is followed; because positive deviation from the limit is desirable, it is considered to be twice as important as both the range and variance values. Similarly for reliability, it is considered the most important factor and is twice as important as the deviation, and four times as important as the range and variance. Because we want a model that has better predictability, a small variance and a small range is desirable. This means that because we normalised those values on a maximum, we would need to subtract both the range value and the variance value from the other two factors. This calculation results in a weighted total; the value of this total is rather arbitrary, it is merely used to differentiate and to draw comparisons between the different models, and more importantly, to be able to rank the models on their reliability. The higher a model's weighted total is, the more 'reliable' the model is considered to be, in comparison with the others. The resulting normalised values and the weighted total can be found in Table 5.4.

Table 5.4: Reliability of models with all factors normalised.

	Reliability values		Parameter values		
Model No.	Average	Range	Deviation	Variance	Weighted total
1	0.62394	0.6371	0.1794	0.7214	1.496
2	0.70639	0.7300	0.2926	0.9888	1.692
3	0.83759	0.7564	0.4769	1.0000	2.548
4	0.78379	1.0000	0.3421	0.9013	1.918
5	0.83826	0.8957	0.3912	0.8692	2.371
6	1.00000	0.0416	0.9137	0.8095	4.976
7	0.99998	0.0431	0.9151	0.8123	4.975
8	0.99943	0.0768	0.8997	0.7815	4.939
9	0.99957	0.0702	1.0000	0.8562	5.072

Based on the above calculations and observations, and by using the values in Table 5.4, one can conclude that Model 9 is the most reliable, with Model 1 being the least reliable. The fact that the current model, which is Model 1, is the least reliable, along with having low strength and a suboptimal cost, emphasises the need to intervene and develop a different solution to the problem. The next subsection covers the steps and calculations that are used to refine the solution set and the process of determining the chosen model as the final solution.

5.2 Model selection

The process outlined in this section deals with the final steps of the model evaluation, which are refining the solution set and determining the optimal model to present as a solution to the problem. Three factors are taken into account in this process; they are the model reliability, modelled blend cost and the modelled blend strength. The reliability of the models was already determined in the previous section, so cost and strength will be the focus for now.

Because the blends from each model have already been determined, the cost of the blend will remain constant throughout all the instances of the model being used, this makes evaluating the models on cost simple, as the cost is a static value. Because we want to add the three different factors together, we again have to normalise the values. This is done similarly to before, with the cost being normalised by taking the lowest cost and dividing it by the modelled blend cost; this ensures the models with the lowest cost have the highest cost factors for comparisons. The result of the normalisation can be seen in Table 5.5.

Similarly for strength, which is also a static value, the strength factor is calculated by normalising the strength value of each model. Strength is normalised by taking each model's strength and dividing by the highest strength values of the different models. The reliability value that was previously calculated is also included as a factor, which has also been normalised. These results can be found in Table 5.5.

The three factors that will be used for model selection have now been calculated and normalised, which means a weighted total can be calculated, which is shown in Table 5.5.

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

Table 5.5: Normalised factors for model selection.

Model No.	Cost	Strength	Reliability	Weighted total
1	0.9455	0.7257	0.2950	5.5283
2	1.0000	0.7961	0.3336	5.9258
3	0.9521	0.8791	0.5023	6.0688
4	0.8296	0.8939	0.3782	5.4846
5	0.8190	0.9160	0.4674	5.5755
6	0.7621	0.9488	0.9811	5.9272
7	0.7615	0.9491	0.9809	5.9250
8	0.7779	0.9731	0.9738	6.0317
9	0.7615	1.0000	1.0000	6.0460

The overall blend strength is considered to be twice as important as reliability, and as before, the cost of the blend is twice as important as strength. This means the cost is four times as important as the reliability. The weighted total for each model is calculated accordingly, and once again, the actual value is of no importance, the value merely allows one to compare and rank the different models. A higher weighted total means a model is superior based on cost, strength and reliability.

Based on the table above, the best model is Model 3, which is the preemptive optimisation model using AMSA's expected values. The current model, Model 1, is second last, and the only reason it is not last is because cost carriers a significantly greater weight and it has a relatively low cost. The refined solution set, with the rankings of the models, from best to worst, can be found in Table 5.6.

Table 5.6: Refined solution set with relative model rankings.

Ranking	Model	Model No.
1	AMSA's expected value preemptive optimisation model	3
2	Approximation approach preemptive optimisation model	9
3	Approximation approach goal programming model	8
4	Scenario approach goal programming model	6
5	AMSA's expected value goal programming model	2
6	Scenario approach preemptive optimisation model	7
7	Estimated expected value preemptive optimisation model	5
8	Current model	1
9	Estimated expected value goal programming model	4

Because the cost and strength values of the models remain static for every instance of that model, comparing the costs and strengths between the models alone is rather rudimentary. A different approach is used to validate the static comparisons between the models, and it involves a retrospective type analysis of the cost and strength values in a two-stage type problem. In this analysis, the inputs are sampled, just as before; however, there is no set blend which the model has to use. This means that the model is solved for

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

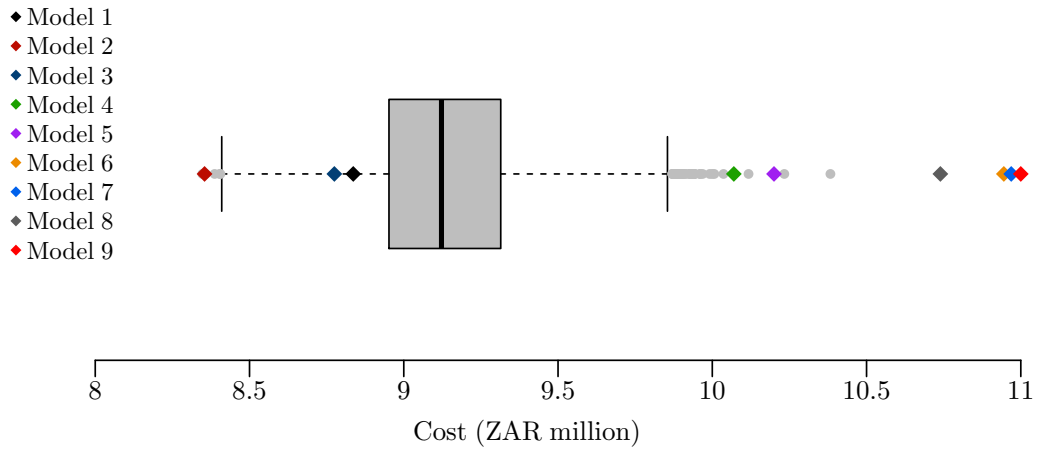


Figure 5.2: Box plot of retrospective analysis with cost of all modelled blends.

an optimal blend, producing an optimal cost and strength each time. This approach is highly impractical in real life as the model is solved under the assumption that the values of the inputs are fully known with 100% accuracy; this is never the case. Nevertheless, what this approach does do is give an idea of how the models perform based on cost and strength when compared to the perfect scenario (the perfect scenario is when all inputs are *fully known* and 100% *accurate*, and the best solution for those inputs is found every time). This type of analysis allows the decision-maker to answer the questions: *"If the inputs are known with certainty, what is the optimal solution every time? Moreover, how do the nine models, and the solutions they give, compare to the optimal solutions over all the instances?"*

By using Monte Carlo simulation, the inputs to the model are sampled out of the PDFs of each coal parameter, and the model is solved 10^4 times. Only one model is used to get solutions, it is an expected value goal programming model, with all the constraints the same as before; there is no need to use any stochastic models, as it is assumed the inputs are known with certainty at the time of solving the model, as it is a retrospective type analysis. This approach is like simulating the question *"What if?"* many times, which in this case is answered 10^4 times. The result of this analysis is two plots, which show where the modelled costs and strengths would lie in comparison to the simulated values, which are plotted as a box plot, with the cost being shown in Figure 5.2, and strength being shown in Figure 5.2.

The plots paint an interesting picture, particularly with regards to cost. There are three models, Models $1-3$, that outperform almost all the instances of the retrospective model. Model 1 has a lower cost 99.97% of the time, while Model 3 has a lower cost 92.35% of the time. The current model has a lower cost 88% of the time. This plot also highlights the poor performance of the other models based on cost.

The strength plot also has some valuable information in that the modelled values seem more evenly spread out across the simulated solutions. As an example, Model 8 has a higher strength 89.25% of the time, while Model 3 only has a higher strength 17.5% of the time. The current blend has a higher strength just 3.75×10^{-6} of the time. It is also interesting to note that no model is far superior in terms of strength. In terms of cost, Model 2 outperformed almost every instance and other the models, but there is not a model like that for strength.

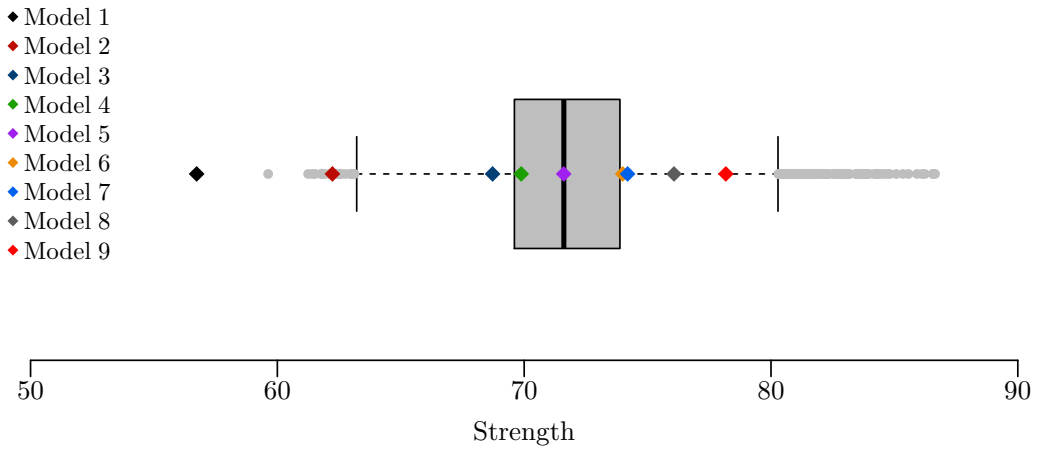


Figure 5.3: Box plot of retrospective analysis with strength of all modelled blends.

What these plots show is how well the respective models stack up when compared to the "perfect scenario" each time. Model 3 was of particular interest in the figures as it is, based on the factors previously calculated, the preferred solution. These plots illustrate that this model performs very well in terms of cost, with it being higher than the simulated value very few times. The strength, however, was not as good, but it still does relatively well when compared to models in the same cost range. All of this, along with the calculations performed above, validate that Model 3 is the chosen final solution to the problem. The preemptive optimisation model, which uses AMSA's expected values, is validated in the next section.

5.3 Model validation

This section deals with the validation of the chosen model as a solution to the problem. This validation is done by testing the actual blend in the pilot coke oven, which recreates the coking process, but on a smaller scale. Ideally, all the blends would have been tested in the pilot oven, but due to cost limitations, only the one blend could be tested. The validation is more related to the physical testing of the blend, as it has already been proven to work theoretically in the previous two sections. The physical testing allows us to prove that the model *actually* works.

To test the blend that is produced by Model 3, $450kg$ of coal is loaded into the pilot oven. This $450kg$ blend is a scaled version of the Model 3 blend found in Table 5.1. The results from the pilot oven testing can be found in Table 5.7.

The results in this table validate the model performance, and they prove that the model works. The model, in this particular instance, has a 100% reliability, meaning every parameter limit is adhered to. Although this is desirable, it will not always be the case; there is an estimated 5.15% chance that Model 3 is 100% accurate — the average reliability of this model is expected to be 83.74%, which is still very good. Most of the actual values from the blend are more favourable than the expected values, and this could be a contributing factor to the higher than average reliability. Only parameters 2, 6 and 8 are worse than the expected values, but they are still within the limits.

The validation was a resounding success, as it shows that the model works on both

Table 5.7: Pilot oven results for Model 3 validation.

Parameter j	Limit	Expected value	Actual value	Reliability
1	≤ 0.05	0.046	0.046	1
2	≤ 0.16	0.12	0.13	1
3	≤ 10	9.45	8.80	1
4	≤ 1	0.84	0.73	1
5	≤ 27	27.34	26.60	1
6	≤ 28	23.91	24.80	1
7	≥ 59	58.20	64.40	1
8	≤ 8.5	5.47	6.19	1
9	≥ 64	66.99	72.80	1
10	≤ 10	6.65	5.00	1
11	≥ 42.5	46.09	54.40	1
12	≤ 24.5	18.73	16.90	1
			Total reliability	100%

a theoretical and an actual level. The chosen solution to the problem, the preemptive optimisation model, using AMSA's expected values, is compared to the current model being used in Table 5.8 below.

Table 5.8: Comparison between chosen and current models.

Model No.	Cost	Strength	Average reliability
1	8 836 283	56.73	0.6238
3	8 774 988	68.72	0.8374
Improvement	61 295	11.99	0.2136

It can be seen that the preferred solution (Model 3) outperforms the current model (Model 1) in all three areas of interest, and in particular, the strength and reliability of the blends. Using Model 3, AMSA will save approximately R22.4 million annually; and with that reduced cost they can improve the average strength of the blend by almost 12 units, which is a 21% improvement. The reliability is also increased by 21.36%, which is an improvement of 34% on the current model. Not only will the blend have a higher strength and be more reliable than the one currently in use, but it is also cheaper, and it is for this reason that Model 3 is the chosen solution to the problem.

Along with the final, single solution, there is also a more refined solution set of 8 new models, which have been ranked from best to worst based on cost, strength and reliability. This solution set gives the decision-maker different options to choose from, depending on what the desired outcome is, or what the circumstances are.

Having achieved the two main deliverables of the project; a multi-objective, stochastic model that accurately captures the essence of the problem, and a set of solutions, with one preferred solution to the problem, the project can be concluded. Recommendations and the conclusion follow in the final chapter.

Chapter 6

Conclusion

The problem of coal blending for coke-making at an integrated steel plant has proven to be a difficult one to solve; not only are there multiple competing objectives, there is also an element of uncertainty. This makes the decision making for the optimal blend highly complicated, and poor decision making previously has led to a suboptimal blend of coals. This has adverse effects on the cost and strength of the blend, as well as the reliability of the solution in use at the moment at ArcelorMittal South Africa Newcastle Works. This poor performance has left the door open for a better, more modern solution to the coal blending problem to be introduced, and this new solution is the topic of this report.

The solution to overcome the problem of the suboptimal blend made use of state-of-practice mathematical modelling techniques, including multi-objective and stochastic programming. The blending problem is prevalent in Operations Research, but application of this problem specifically for coke-making has been minimal. There are only two known papers that attempt to optimise the coal blending process for coke-making; they are by Vasko et al. (2005) and Simons (1997), which is cited by Vasko et al. (2005). These papers, however, do not take the uncertainty that is inherent to the problem, as highlighted by the Exploratory Data Analysis (EDA), into account, nor do they consider the simultaneous optimisation of both cost and strength. It is for that reason that multi-objective and stochastic programming techniques were researched in-depth, and applied to the problem to develop eight mathematical models, each of which followed a different intervention to the problem.

This project brought together different mathematical modelling techniques, ranging from simple Linear Programming (LP), to more complex techniques, like Chance-constrained Programming (CCP) and Monte Carlo simulation, and applied them to the coke-making process. There is sufficient evidence and motivation for the need to use both multi-objective and stochastic programming techniques, and in this regard, the project departs from other research in the coal blending for coke-making process. There are many counterparts for the multi-objective stochastic blending problem found in literature, but this is the first known paper that uses these techniques in the coke-making process.

To this end, the available techniques were taken advantage of, and it was shown that the optimisation models developed produce favourable results for AMSA. All the developed models produce blends with higher strengths and reliabilities than the current solution, and two models produce a lower cost as well; these models make up the solution set that was presented to AMSA. There is one model that is chosen as the preferred solution to the problem — the preemptive optimisation model that uses the values expected by AMSA. This model will save approximately R22.4 million annually, with drastically increased strengths and reliability; these are all critical to efficient blast furnace opera-

tion, and subsequently, the entire integrated steelmaking process. This was validated both theoretically and physically. To ensure optimal solutions going forward, AMSA will have to continually check and validate the models in the solution set and make adjustments as required. Because data were the backbone of the project, the datasets used will have to be continuously updated with the ever-changing coal parameters, as this will ensure greater accuracy, and with this, the models will have to be rerun and solved often.

One limitation of the project was the fact that it was only considered as a one-stage problem, with no recourse available. This was mainly done to simplify the problem, and there was insufficient knowledge on the intricacies of the coke-making process to consider it as a two-stage problem. A recommendation for when the project is taken further is to expand it to be a two-stage problem; this will allow recourse programming to be used. The benefits of recourse programming are undeniable, and it would be of particular use for this problem as it would be less conservative than CCP. It is hypothesised that recourse programming will ultimately produce even more favourable results, as there will not be needless and excessive caution in the decision-making process because there is recourse available.

Another recommendation would be to implement a machine learning approach to predict the value of the coal parameters, as it is hypothesised that it can be done — this would have substantial benefit as there would be less uncertainty to consider, and again there would be no need to be excessively cautious. Coal parameter prediction has been partially successful, shown by Chelgani et al. (2011), however, it has yet to be applied to the coke-making industry. This, combined with recourse programming, could see AMSA Newcastle Works improve their coal blending process significantly.

Nonetheless, the project was a great success; some of the endless possibilities of operations research, and in particular mathematical modelling for optimisation, were demonstrated and applied. The results are positive and insightful, and more control over the coal blending process has been given to the decision-makers through modelling. The only step that remains in the project is full-scale implementation at AMSA Newcastle Works, and beyond; this is extremely exciting, and there is much anticipation around this project's future.

Bibliography

Ashayeri, J., van Eijs, A., and Nederstigt, P. (1994). Blending modelling in a process manufacturing: A case study. *European Journal of Operational Research*, 72(3):460–468.

Babic, Z. and Peric, T. (2011). Optimization of livestock feed blend by use of goal programming. *International Journal of Production Economics*, 130(2):218–223.

Babich, A. and Senk, D. (2019). *New trends in coal conversion: Combustion, Gasifications, Emissions and Coking*. Woodhead Publishing, Cambridge, England.

Bonate, P. (2001). A brief introduction to monte carlo simulation. *Clinical Pharmacokinetics*, 40(1):15–22.

Chelgani, S., Dehghan, F., and Hower, J. (2011). Estimation of some coal parameters depending on petrographic and inorganic analyses by using genetic algorithm and adaptive neuro-fuzzy inference systems. *Energy Exploration and Exploitation*, 29(4):479–494.

Djeumou Fomeni, F. (2018). A multi-objective optimisation approach for the blending problem in the tea industry. *International Journal of Production Economics*, 205:452–465.

Jareonkitpoolpol, A., Ongkunaruk, P., and Janssens, G. (2018). Determination of the optimal blending problem of organic-chemical fertilizer under uncertainty. *Soil Use and Management*, 34(4):449–460.

Jensen, P. (2004). Operation research and methods: Stochastic programming. https://www.me.utexas.edu/~jensen/ORMM/computation/unit/stoch_program/recourse.html#top. Accessed 23 May 2020.

Joubert, J. (2019). *Operations research course notes (BOZ 780)*. University of Pretoria.

Manson, N. (2006). Is operations research really research? *ORiON: The Journal of ORSSA*, 22(2):155–180.

Marler, R. and Arora, J. (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 41(2):853–862.

Nemirovski, A. and Shapiro, A. (2006). Convex approximations of chance constrained programs. *SIAM Journal on Optimisation*, 17(4):969–996.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rardin, R. (2016). *Optimization in Operations Research*. Pearson, Hoboken, New Jersey.

Rifai, A. (1996). A note on the structure of the goal-programming model: assessment and evaluation. *International Journal of Operations and Production Management*, 16(1):40–49.

Sakalli, U. and Baykoc, O. (2011). An optimization approach for brass casting blending problem under aleatory and epistemic uncertainty. *International Journal of Production Economics*, 133:708–718.

Sakalli, U., Baykoc, O., and Birgören, B. (2011). Stochastic optimization for blending problem in brass casting industry. *Annals of Operations Research*, 186:141–157.

Sakalli, U. and Birgören, B. (2019). Joint optimization of quality and cost in brass casting using stochastic programming. *Engineering Optimization*, 0(0):1–13.

Sarna, S. (2015). Selection of coal for inclusion in coal blend in coke making. https://www.ispatguru.com/selection-of-coal-for-inclusion-in-coal-blend-in-coke-making. Accessed 22 May 2020.

Shih, J. and Frey, H. (1995). Coal blending optimization under uncertainty. *European Journal of Operations Research*, 83(3):452–465.

Vasko, F., Newhart, D., and Strauss, A. (2005). Coal blending models for optimum cokemaking and blast furnace operation. *Journal of the Operational Research Society*, 56(3):235–243.

Winston, W. (2003). *Operations Research: Applications and Algorithms*. Cengage, Boston, Massachusetts.

Appendix A

Estimated probability distributions of parameters

Table A.1: Estimated probability distributions of coke parameters.

Coal No.	CRI (%)	CSR (%)	Wall pressure (kPa)
1	$\mathcal{N}(15.15, 1.85)$	*Logis* (70.01, 1.45)	$\mathcal{N}(9.23, 2.62)$
2	$\mathcal{N}(20.53, 3.31)$	$\mathcal{U}(60.1, 75.5)$	Γ (12.71, 4.28)
3	*Logis* (47.8, 2.04)	*Logis* (24.7, 2.55)	Γ (8.13, 2.48)
4	$\mathcal{U}(19.3, 26.5)$	$\mathcal{U}(58.4, 68.7)$	Γ (9.09, 3.04)
5	*Logis* (15.58, 0.83)	$\mathcal{U}(60.1, 78.5)$	$\mathcal{N}(23.12, 3.16)$
6	Γ (123.2, 7.88)	*Logis* (69.85, 1.08)	Γ (4.14, 0.66)
7	$\mathcal{N}(21.11, 1.82)$	$\mathcal{N}(61.03, 2.11)$	$\mathcal{N}(3.81, 1.49)$
8	$\mathcal{N}(22.19, 2.49)$	*Logis* (59.92, 1.92)	$\mathcal{N}(3.44, 1.00)$
9	$\mathcal{N}(27.91, 2.76)$	$\mathcal{N}(53.24, 3.72)$	$\mathcal{N}(5.43, 1.83)$

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

Table A.2: Estimated probability distributions of Micum and Irsid parameters.

Coal No.	M40 (%)	M10 (%)	I40 (%)	I10 (%)
1	$\mathcal{N}(80.4, 2.52)$	$\mathcal{N}(5.81, 0.76)$	$\mathcal{U}(52.7, 65.8)$	$\mathcal{N}(17.73, 1.78)$
2	$\mathcal{N}(74.67, 3.17)$	*Logis* (6.61, 0.63)	$\mathcal{N}(51.9, 3.64)$	$\mathcal{N}(20.2, 2.22)$
3	$\mathcal{N}(30.56, 6.34)$	$\mathcal{N}(7.98, 1.69)$	$\ln\mathcal{N}(2.0, 0.39)$	*Logis* (18.92, 2.78)
4	$\mathcal{N}(73.76, 3.45)$	*Logis* (7.56, 0.74)	$\mathcal{N}(51.14, 4.52)$	$\mathcal{N}(20.49, 2.46)$
5	$\mathcal{N}(81.09, 2.44)$	$\mathcal{U}(3.4, 7.6)$	$\mathcal{N}(62.32, 4.20)$	$\mathcal{N}(18.07, 2.42)$
6	*Logis* (79.64, 1.55)	$\mathcal{N}(6.20, 0.91)$	$\mathcal{N}(59.36, 4.05)$	*Logis* (18.66, 1.28)
7	$\mathcal{N}(70.01, 3.94)$	$\mathcal{N}(6.58, 0.80)$	$\mathcal{N}(45.64, 5.67)$	$\mathcal{N}(19.24, 2.54)$
8	$\mathcal{N}(66.92, 7.68)$	*Logis* (6.58, 0.43)	$\mathcal{N}(41.38, 8.81)$	$\mathcal{N}(19.0, 2.25)$
9	$\mathcal{N}(61.23, 4.99)$	$\mathcal{N}(6.94, 1.23)$	$\mathcal{N}(38.98, 4.83)$	$\mathcal{N}(19.07, 3.23)$

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

Table A.3: Estimated probability distributions of incoming coal parameters.

Coal No.	Phosphorus (%)	Potassium oxide (%)	Ash (%)	Sulphur (%)	Volatiles (%)
1	$\mathcal{N}(0.07, 0.01)$	$\mathcal{N}(0.19, 0.02)$	$\mathcal{N}(10.20, 0.32)$	$\mathcal{N}(0.90, 0.06)$	$\mathcal{N}(23.78, 0.71)$
2	$\ln\mathcal{N}(-4.12, 0.31)$	$Logis\,(0.1, 0.01)$	$Logis\,(8.96, 0.15)$	$\ln\mathcal{N}(-0.6, 0.06)$	$\mathcal{N}(24, 0.6)$
3	$\mathcal{N}(0.005, 0.002)$	$\mathcal{N}(0.12, 0.02)$	$\mathcal{N}(10.14, 0.24)$	$\mathcal{N}(1.05, 0.07)$	$\mathcal{N}(36.46, 0.5)$
4	$\mathcal{N}(0.06, 0.007)$	$\mathcal{N}(0.13, 0.03)$	$\mathcal{N}(10.01, 0.38)$	$\mathcal{N}(0.34, 0.03)$	$\mathcal{N}(20.52, 0.51)$
5	$\mathcal{N}(0.07, 0.008)$	$\mathcal{N}(0.19, 0.01)$	$\mathcal{N}(10.18, 0.22)$	$\mathcal{N}(1.02, 0.05)$	$\mathcal{N}(20.31, 0.51)$
6	$Logis\,(0.06, 0.005)$	$\mathcal{N}(0.12, 0.01)$	$\mathcal{N}(9.2, 0.24)$	$\mathcal{N}(0.62, 0.05)$	$Weib\,(23.95, 24.74)$
7	$\mathcal{N}(0.04, 0.01)$	$\mathcal{N}(0.26, 0.04)$	$\mathcal{N}(9.37, 0.37)$	$\ln\mathcal{N}(-0.32, 0.08)$	$\mathcal{N}(27.85, 0.59)$
8	$\mathcal{U}(0.03, 0.05)$	$\mathcal{N}(0.27, 0.03)$	$\mathcal{N}(9.53, 0.35)$	$\mathcal{N}(0.75, 0.07)$	$Logis\,(28.07, 0.31)$
9	$\mathcal{N}(0.04, 0.006)$	$\mathcal{N}(0.15, 0.02)$	$\mathcal{N}(9.70, 0.28)$	$\mathcal{N}(0.77, 0.06)$	$\mathcal{N}(27.99, 0.71)$

Appendix B

Data used in formulations

Table B.1: AMSA's expected values for coal parameters.

	Parameter j											
Coal i	1	2	3	4	5	6	7	8	9	10	11	12
1	0.042	0.20	10.5	0.96	23.3	13.3	74.0	10.9	76.4	5.3	55.7	17.7
2	0.022	0.08	8.80	0.59	23.8	14.5	76.1	2.47	80.4	6.4	56.8	19.8
3	0.005	0.09	10.2	1.11	36.3	42.0	31.1	3.66	38.7	6.0	10.2	21.2
4	0.058	0.15	9.70	0.35	20.5	19.9	71.5	4.14	75.3	6.3	52.1	19.4
5	0.069	0.18	10.1	1.01	20.6	13.0	76.1	31.8	83.9	4.8	58.1	15.2
6	0.066	0.11	9.20	0.63	22.3	13.5	73.8	8.22	80.7	6.2	62.2	18.0
7	0.042	0.24	9.10	0.71	26.7	17.0	67.8	5.24	72.6	7.1	48.9	20.7
8	0.006	0.20	7.20	0.87	32.4	21.5	61.9	3.55	54.2	7.6	26.6	19.6
9	0.050	0.17	10.8	0.83	24.1	24.9	57.1	4.00	67.9	8.3	43.1	16.6

Table B.2: Estimated expected values for coal parameters.

	Parameter j											
Coal i	1	2	3	4	5	6	7	8	9	10	11	12
1	0.069	0.19	10.2	0.90	23.8	15.2	70.0	9.23	80.4	5.81	59.3	17.7
2	0.016	0.10	9.0	0.57	24.0	20.5	67.8	2.89	74.7	6.61	51.9	20.2
3	0.005	0.12	10.1	1.05	36.5	47.8	24.7	3.15	30.6	7.98	7.4	18.9
4	0.061	0.13	10.0	0.34	20.5	22.9	63.6	2.88	73.8	7.56	51.1	20.5
5	0.068	0.19	10.2	1.02	20.3	15.6	69.3	23.11	81.1	5.50	62.3	18.1
6	0.060	0.12	9.2	0.62	24.4	15.6	15.7	5.78	79.6	6.21	59.4	18.7
7	0.043	0.26	9.4	0.73	27.8	21.1	61.0	3.81	70.0	6.58	45.6	19.2
8	0.040	0.27	9.5	0.75	28.1	22.2	59.9	3.44	66.9	6.54	41.4	19.0
9	0.040	0.15	9.7	0.77	28.0	27.9	53.2	5.43	61.2	6.94	39.0	19.1

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

Table B.3: Deterministic components for approximation approach.

	Parameter j											
Coal i	1	2	3	4	5	6	7	8	9	10	11	12
1	0.070	0.19	10.20	0.90	23.78	15.2	70.0	9.23	80.40	5.81	59.25	17.73
2	0.027	0.10	8.96	0.63	24.00	20.5	67.8	4.46	74.67	6.61	51.90	20.20
3	0.005	0.12	10.14	1.05	36.46	47.8	24.7	5.37	30.56	7.98	3.89	18.92
4	0.060	0.13	10.01	0.34	20.52	22.9	63.6	4.79	73.76	7.56	51.14	20.49
5	0.070	0.19	10.18	1.02	20.31	15.6	69.3	23.12	81.09	5.50	62.32	18.07
6	0.060	0.12	9.20	0.62	25.90	18.0	69.9	12.05	79.64	7.71	59.32	18.66
7	0.040	0.26	9.37	0.83	27.85	21.1	61.0	3.81	70.01	6.58	45.64	19.24
8	0.040	0.27	9.53	0.75	28.07	22.2	59.9	3.44	66.92	6.58	41.38	19.00
9	0.040	0.15	9.70	0.77	27.99	27.9	53.2	5.43	61.23	6.94	38.98	19.07

Table B.4: Stochastic components for approximation approach.

	Parameter j											
Coal i	1	2	3	4	5	6	7	8	9	10	11	12
1	0.00	0.000	0.10	0.004	0.50	3.42	6.92	6.86	6.35	0.58	14.30	3.17
2	0.00	0.000	0.07	0.000	0.36	10.96	19.76	0.00	10.05	1.31	13.25	4.93
3	0.00	0.000	0.06	0.005	0.25	13.69	21.39	0.00	40.20	2.86	0.00	25.43
4	0.00	0.001	0.14	0.001	0.26	4.32	8.84	0.00	11.90	1.66	20.43	6.05
5	0.00	0.000	0.05	0.003	0.26	2.27	28.21	9.99	5.95	1.47	17.64	5.86
6	0.00	0.000	0.06	0.003	0.00	0.00	3.84	0.00	7.90	0.83	16.40	5.39
7	0.00	0.002	0.14	0.000	0.35	3.31	4.45	2.22	15.52	0.64	32.15	6.45
8	0.00	0.001	0.12	0.005	0.32	6.20	12.13	1.00	58.98	0.61	77.62	5.06
9	0.00	0.000	0.08	0.004	0.50	7.62	13.84	3.35	24.90	1.51	23.33	10.43

Appendix C

Entire model formulations

The models that are in this appendix are the full formulations of all the different models formulated in the report.

C.1 Goal programming model using AMSA's expected values

This model is the full formulation of the goal programming model where the parameters values used are the ones AMSA expects. This is the full model for that which is partially formulated in Subsection 4.1.1. We start by letting $\boldsymbol{I}$ be the set of coals, such that $i \in \boldsymbol{I} = \{1, \ldots, 9\}$, and $\boldsymbol{J}$ be the set of coal parameters, such that $j \in \boldsymbol{J} = \{1, \ldots, 12\}$.

The model parameters are defined as follows:

$$
\begin{aligned}
x_i &\triangleq \text{the amount (tons) of coal type } i \in \boldsymbol{I} \text{ in the blend.} \\
c_i &\triangleq \text{the cost (R/ton) of coal type } i \in \boldsymbol{I}. \\
s_i &\triangleq \text{the expected strength measure of coal type } i \in \boldsymbol{I} \text{ after coking.} \\
p^A_{i,j} &\triangleq \text{the AMSA estimated parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
L^{\min}_j &\triangleq \text{the lower limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
L^{\max}_j &\triangleq \text{the upper limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
b_i &\triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases} \\
d_i &\triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases} \\
t &\triangleq \text{the total amount (tons) of coal in the blend.} \\
G_c &\triangleq \text{the goal for the the cost objective.} \\
G_s &\triangleq \text{the goal for the the strength objective.} \\
d_c &\triangleq \text{the cost deviational variable.} \\
d_s &\triangleq \text{the strength deviational variable.} \\
d^N_c &\triangleq \text{the normalised cost deviational variable.} \\
d^N_s &\triangleq \text{the normalised strength deviational variable.} \\
w_c &\triangleq \text{the cost weighting variable.} \\
w_s &\triangleq \text{the strength weighting variable.}
\end{aligned}
$$

The model is then formulated:

$$\min z_d = w_c d^N_c + w_s d^N_s - 10^{-12} z_c + 10^{-7} z_s \tag{C.1}$$

subject to:

$$\sum_{i \in \boldsymbol{I}} x_i c_i - d_c = G_c \tag{C.2}$$

$$\sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} + d_s = G_s \tag{C.3}$$

$$d_c^N = \frac{d_c}{G_c} \tag{C.4}$$

$$d_s^N = \frac{d_s}{G_s} \tag{C.5}$$

$$t = 2\,996 \tag{C.6}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{C.7}$$

$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^A}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{C.8}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{C.9}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{C.10}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{C.11}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{C.12}$$

$$x_9 \leq 60 \tag{C.13}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{C.14}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.15}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.16}$$

C.2 Weighted sum of objective model using AMSA's expected values

This model is the full formulation of the weighted sum of objective model where the parameters values used are the ones AMSA expects. This is the full model for that which is partially formulated in Subsection 4.1.1. We start by letting $\boldsymbol{I}$ be the set of coals, such that $i \in \boldsymbol{I} = \{1, \ldots, 9\}$, and $\boldsymbol{J}$ be the set of coal parameters, such that $j \in \boldsymbol{J} = \{1, \ldots, 12\}$.

The model parameters are defined as follows:

$x_i \triangleq$ the amount (tons) of coal type $i \in \boldsymbol{I}$ in the blend.
$c_i \triangleq$ the cost (R/ton) of coal type $i \in \boldsymbol{I}$.
$s_i \triangleq$ the expected strength measure of coal type $i \in \boldsymbol{I}$ after coking.
$p^A_{i,j} \triangleq$ the AMSA estimated parameter $j \in \boldsymbol{J}$ of coal type $i \in \boldsymbol{I}$.
$L^{\min}_j \triangleq$ the lower limit for parameter $j \in \boldsymbol{J}$ in the blend.
$L^{\max}_j \triangleq$ the upper limit for parameter $j \in \boldsymbol{J}$ in the blend.
$b_i \triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases}$
$d_i \triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases}$
$t \triangleq$ the total amount (tons) of coal in the blend.
$G_c \triangleq$ the goal for the the cost objective.
$G_s \triangleq$ the goal for the the strength objective.
$w_c \triangleq$ the cost weighting variable.
$w_s \triangleq$ the strength weighting variable.

The model is then formulated:

$$\max z_{c,s} = w_s \frac{\sum_{i \in \boldsymbol{I}} x_i s_i}{t G_s} - w_c \frac{\sum_{i \in \boldsymbol{I}} x_i c_i}{G_c} \tag{C.17}$$

subject to:

$$t = 2\,996 \tag{C.18}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{C.19}$$

$$L^{\min}_j \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p^A_{i,j}}{t} \leq L^{\max}_j \qquad \forall j \in \boldsymbol{J} \tag{C.20}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{C.21}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{C.22}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{C.23}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{C.24}$$

$$x_9 \leq 60 \tag{C.25}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{C.26}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.27}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.28}$$

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

C.3 Preemptive optimisation model using AMSA's expected values

This model is the full formulation of the preemptive optimisation model where the parameters values used are the ones AMSA expects. This is the full model for that which is partially formulated in Subsection 4.1.1. We start by letting $\boldsymbol{I}$ be the set of coals, such that $i \in \boldsymbol{I} = \{1, \ldots, 9\}$, and $\boldsymbol{J}$ be the set of coal parameters, such that $j \in \boldsymbol{J} = \{1, \ldots, 12\}$.

The model parameters are defined as follows:

$$
\begin{aligned}
x_i &\triangleq \text{the amount (tons) of coal type } i \in \boldsymbol{I} \text{ in the blend.} \\
c_i &\triangleq \text{the cost (R/ton) of coal type } i \in \boldsymbol{I}. \\
s_i &\triangleq \text{the expected strength measure of coal type } i \in \boldsymbol{I} \text{ after coking.} \\
p^A_{i,j} &\triangleq \text{the AMSA estimated parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
L^{\min}_j &\triangleq \text{the lower limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
L^{\max}_j &\triangleq \text{the upper limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
b_i &\triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases} \\
d_i &\triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases} \\
t &\triangleq \text{the total amount (tons) of coal in the blend.} \\
G_c &\triangleq \text{the goal for the the cost objective.} \\
C_{increment} &\triangleq \text{the cost increment that the cost objective value increases by each iteration of the loop.}
\end{aligned}
$$

The model is then formulated:

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{C.29}$$

subject to:

$$\sum_{i \in \boldsymbol{I}} x_i c_i \leq G_c + C_{increment} \tag{C.30}$$

$$t = 2\,996 \tag{C.31}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{C.32}$$

$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^A}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{C.33}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{C.34}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{C.35}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{C.36}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{C.37}$$

$$x_9 \leq 60 \tag{C.38}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{C.39}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.40}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.41}$$

C.4 Goal programming model using estimated expected values

This model is the full formulation of the goal programming model where the parameters values used are the ones estimated expected values, which are estimated by finding the median value of each of the Probability Distribution Functions (PDF's). This is the full model for that which is partially formulated in Subsection 4.1.2. We start by letting $\boldsymbol{I}$ be the set of coals, such that $i \in \boldsymbol{I} = \{1, \ldots, 9\}$, and $\boldsymbol{J}$ be the set of coal parameters, such that $j \in \boldsymbol{J} = \{1, \ldots, 12\}$.

The model parameters are defined as follows:

$$
\begin{aligned}
x_i &\triangleq \text{the amount (tons) of coal type } i \in \boldsymbol{I} \text{ in the blend.} \\
c_i &\triangleq \text{the cost (R/ton) of coal type } i \in \boldsymbol{I}. \\
s_i &\triangleq \text{the expected strength measure of coal type } i \in \boldsymbol{I} \text{ after coking.} \\
p^E_{i,j} &\triangleq \text{the estimated expected value of parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
L^{\min}_j &\triangleq \text{the lower limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
L^{\max}_j &\triangleq \text{the upper limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
b_i &\triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases} \\
d_i &\triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases} \\
t &\triangleq \text{the total amount (tons) of coal in the blend.} \\
G_c &\triangleq \text{the goal for the the cost objective.} \\
G_s &\triangleq \text{the goal for the the strength objective.} \\
d_c &\triangleq \text{the cost deviational variable.} \\
d_s &\triangleq \text{the strength deviational variable.} \\
d^N_c &\triangleq \text{the normalised cost deviational variable.} \\
d^N_s &\triangleq \text{the normalised strength deviational variable.} \\
w_c &\triangleq \text{the cost weighting variable.} \\
w_s &\triangleq \text{the strength weighting variable.}
\end{aligned}
$$

The model is then formulated:

$$\min z_d = w_c d^N_c + w_s d^N_s - 10^{-12} z_c + 10^{-7} z_s \tag{C.42}$$

subject to:

$$\sum_{i \in \boldsymbol{I}} x_i c_i - d_c = G_c \tag{C.43}$$

$$\sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} + d_s = G_s \tag{C.44}$$

$$d_c^N = \frac{d_c}{G_c} \tag{C.45}$$

$$d_s^N = \frac{d_s}{G_s} \tag{C.46}$$

$$t = 2\,996 \tag{C.47}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{C.48}$$

$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^E}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{C.49}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{C.50}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{C.51}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{C.52}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{C.53}$$

$$x_9 \leq 60 \tag{C.54}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{C.55}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.56}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.57}$$

C.5 Preemptive optimisation model using estimated expected values

This model is the full formulation of the preemptive optimisation model where the parameters values used are the ones AMSA expects. This is the full model for that which is partially formulated in Subsection 4.1.1. We start by letting $\boldsymbol{I}$ be the set of coals, such that $i \in \boldsymbol{I} = \{1, \ldots, 9\}$, and $\boldsymbol{J}$ be the set of coal parameters, such that $j \in \boldsymbol{J} = \{1, \ldots, 12\}$.

The model parameters are defined as follows:

$x_i \triangleq$ the amount (tons) of coal type $i \in \boldsymbol{I}$ in the blend.
$c_i \triangleq$ the cost (R/ton) of coal type $i \in \boldsymbol{I}$.
$s_i \triangleq$ the expected strength measure of coal type $i \in \boldsymbol{I}$ after coking.
$p^E_{i,j} \triangleq$ the estimated expected value of parameter $j \in \boldsymbol{J}$ of coal type $i \in \boldsymbol{I}$.
$L^{\min}_j \triangleq$ the lower limit for parameter $j \in \boldsymbol{J}$ in the blend.
$L^{\max}_j \triangleq$ the upper limit for parameter $j \in \boldsymbol{J}$ in the blend.
$b_i \triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases}$
$d_i \triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases}$
$t \triangleq$ the total amount (tons) of coal in the blend.
$G_c \triangleq$ the goal for the the cost objective.
$C_{increment} \triangleq$ the cost increment that the cost objective value increases by each iteration of the loop.

The model is then formulated:

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{C.58}$$

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

subject to:

$$\sum_{i \in \boldsymbol{I}} x_i c_i \leq G_c + C_{increment} \tag{C.59}$$

$$t = 2\,996 \tag{C.60}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{C.61}$$

$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^E}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{C.62}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{C.63}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{C.64}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{C.65}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{C.66}$$

$$x_9 \leq 60 \tag{C.67}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{C.68}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.69}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.70}$$

C.6 CCP model

This model is the full formulation of the multi-objective chance-constrained model, which is partially formulated in Subsection 4.2.1. We start by letting $\boldsymbol{I}$ be the set of coals, such that $i \in \boldsymbol{I} = \{1, \ldots, 9\}$, and $\boldsymbol{J}$ be the set of coal parameters, such that $j \in \boldsymbol{J} = \{1, \ldots, 12\}$.

The model parameters are defined as follows:

$$
\begin{aligned}
x_i &\triangleq \text{ the amount (tons) of coal type } i \in \boldsymbol{I} \text{ in the blend.} \\
c_i &\triangleq \text{ the cost (R/ton) of coal type } i \in \boldsymbol{I}. \\
s_i &\triangleq \text{ the expected strength measure of coal type } i \in \boldsymbol{I} \text{ after coking.} \\
\widetilde{p}_{i,j} &\triangleq \text{ the randomly distributed parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
L_j^{\min} &\triangleq \text{ the lower limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
L_j^{\max} &\triangleq \text{ the upper limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
b_i &\triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases} \\
d_i &\triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases} \\
t &\triangleq \text{ the total amount (tons) of coal in the blend.}
\end{aligned}
$$

The model is then formulated:

$$\min z_c = \sum_{i \in \boldsymbol{I}} x_i c_i \tag{C.71}$$

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{C.72}$$

subject to:

$$t = 2\,996 \tag{C.73}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{C.74}$$

$$\Pr\left(L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i \widetilde{p}_{i,j}}{t} \leq L_j^{\max} \right) \geq 1 - \alpha \qquad \forall j \in \boldsymbol{J} \tag{C.75}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{C.76}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{C.77}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{C.78}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{C.79}$$

$$x_9 \leq 60 \tag{C.80}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{C.81}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.82}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.83}$$

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

C.7 Goal programming with scenario approach

This model is the full formulation of the goal programming scenario approach model, which is partially formulated in Subsection 4.2.1. The model is formulated by letting $\boldsymbol{I}$ being the set of coals, such that $i \in \boldsymbol{I}$, $\boldsymbol{J}$ being the set of coal parameters, such that $j \in \boldsymbol{J}$, and $\nu = 1, \ldots, N$, which is the set of samples.

The model parameters are defined as follows:

$$
\begin{aligned}
x_i &\triangleq \text{the amount (tons) of coal type } i \in \boldsymbol{I} \text{ in the blend.} \\
c_i &\triangleq \text{the cost (R/ton) of coal type } i \in \boldsymbol{I}. \\
s_i &\triangleq \text{the expected strength measure of coal type } i \in \boldsymbol{I} \text{ after coking.} \\
p^A_{i,j} &\triangleq \text{the AMSA estimated parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
L^{\min}_j &\triangleq \text{the lower limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
L^{\max}_j &\triangleq \text{the upper limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
b_i &\triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases} \\
d_i &\triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases} \\
t &\triangleq \text{the total amount (tons) of coal in the blend.} \\
G_c &\triangleq \text{the goal for the the cost objective.} \\
G_s &\triangleq \text{the goal for the the strength objective.} \\
d_c &\triangleq \text{the cost deviational variable.} \\
d_s &\triangleq \text{the strength deviational variable.} \\
d^N_c &\triangleq \text{the normalised cost deviational variable.} \\
d^N_s &\triangleq \text{the normalised strength deviational variable.} \\
w_c &\triangleq \text{the cost weighting variable.} \\
w_s &\triangleq \text{the strength weighting variable.}
\end{aligned}
$$

The model is then formulated:

$$\min z_d = w_c d^N_c + w_s d^N_s - 10^{-12} z_c + 10^{-7} z_s \tag{C.84}$$

subject to:

$$\sum_{i \in \boldsymbol{I}} x_i c_i - d_c = G_c \tag{C.85}$$

$$\sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} + d_s = G_s \tag{C.86}$$

$$d_c^N = \frac{d_c}{G_c} \tag{C.87}$$

$$d_s^N = \frac{d_s}{G_s} \tag{C.88}$$

$$t = 2\,996 \tag{C.89}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{C.90}$$

$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^{\nu}}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J}, \nu = 1, \ldots, N \tag{C.91}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{C.92}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{C.93}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{C.94}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{C.95}$$

$$x_9 \leq 60 \tag{C.96}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{C.97}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.98}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.99}$$

C.8 Preemptive optimisation with scenario approach

This model is the full formulation of the preemptive optimisation scenario approach model, found in Subsection 4.2.1. The model is formulated by letting $\boldsymbol{I}$ being the set of coals, such that $i \in \boldsymbol{I}$, $\boldsymbol{J}$ being the set of coal parameters, such that $j \in \boldsymbol{J}$, and $\nu = 1, \ldots, N$, which is the set of samples.

The model parameters are defined as follows:

$$
\begin{aligned}
x_i &\triangleq \text{ the amount (tons) of coal type } i \in \boldsymbol{I} \text{ in the blend.} \\
c_i &\triangleq \text{ the cost (R/ton) of coal type } i \in \boldsymbol{I}. \\
s_i &\triangleq \text{ the expected strength measure of coal type } i \in \boldsymbol{I} \text{ after coking.} \\
p_{i,j}^{E} &\triangleq \text{ the estimated expected value of parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
L_j^{\min} &\triangleq \text{ the lower limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
L_j^{\max} &\triangleq \text{ the upper limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
b_i &\triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases} \\
d_i &\triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases} \\
t &\triangleq \text{ the total amount (tons) of coal in the blend.} \\
G_c &\triangleq \text{ the goal for the the cost objective.}
\end{aligned}
$$

The model is then formulated:

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{C.100}$$

subject to:

$$\sum_{i \in \boldsymbol{I}} x_i c_i = G_c \tag{C.101}$$

$$t = 2\,996 \tag{C.102}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{C.103}$$

$$L_j^{\min} \leq \sum_{i \in \boldsymbol{I}} \frac{x_i p_{i,j}^{\nu}}{t} \leq L_j^{\max} \qquad \forall j \in \boldsymbol{J}, \nu = 1, \ldots, N \tag{C.104}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{C.105}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{C.106}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{C.107}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{C.108}$$

$$x_9 \leq 60 \tag{C.109}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{C.110}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.111}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.112}$$

C.9 Approximation approach model

This model is the full formulation of the multi-objective chance-constrained model, which is partially formulated in Subsection 4.2.2. We start by letting $\boldsymbol{I}$ be the set of coals, such that $i \in \boldsymbol{I} = \{1, \ldots, 9\}$, and $\boldsymbol{J}$ be the set of coal parameters, such that $j \in \boldsymbol{J} = \{1, \ldots, 12\}$.

The model parameters are defined as follows:

$$
\begin{aligned}
x_i &\triangleq \text{the amount (tons) of coal type } i \in \boldsymbol{I} \text{ in the blend.} \\
c_i &\triangleq \text{the cost (R/ton) of coal type } i \in \boldsymbol{I}. \\
s_i &\triangleq \text{the expected strength measure of coal type } i \in \boldsymbol{I} \text{ after coking.} \\
\widetilde{p}_{i,j} &\triangleq \text{the randomly distributed parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
L_j^{\min} &\triangleq \text{the lower limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
L_j^{\max} &\triangleq \text{the upper limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
b_i &\triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases} \\
d_i &\triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases} \\
t &\triangleq \text{the total amount (tons) of coal in the blend.} \\
E(\widetilde{p}_{i,j}) &\triangleq \text{the deterministic component of parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
\sigma_{i,j}^2 &\triangleq \text{the stochastic component of parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}.
\end{aligned}
$$

The model is then formulated:

$$\min z_c = \sum_{i \in \boldsymbol{I}} x_i c_i \tag{C.113}$$

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{C.114}$$

subject to:

$$t = 2\,996 \tag{C.115}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{C.116}$$

$$tL_j^{\min} \leq \sum_{i \in \boldsymbol{I}} x_i E(\widetilde{p}_{i,j}) + \Omega \left(\sum_{i \in \boldsymbol{I}} x_i^2 \sigma_{i,j}^2 \right)^{\frac{1}{2}} \leq tL_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{C.117}$$

$$x_i \leq tb_i \qquad \forall i \in \boldsymbol{I} \tag{C.118}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{C.119}$$

$$30 - x_i \leq td_i \qquad \forall i \in \boldsymbol{I} \tag{C.120}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{C.121}$$

$$x_9 \leq 60 \tag{C.122}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{C.123}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.124}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.125}$$

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

C.10 Goal programming model with approximation approach

This model is the full formulation of the goal programming scenario approach model, which is partially formulated in Subsection 4.2.2. The model is formulated by letting $\boldsymbol{I}$ being the set of coals, such that $i \in \boldsymbol{I}$, $\boldsymbol{J}$ being the set of coal parameters, such that $j \in \boldsymbol{J}$, and $\nu = 1, \ldots, N$, which is the set of samples.

The model parameters are defined as follows:

$x_i \triangleq$ the amount (tons) of coal type $i \in \boldsymbol{I}$ in the blend.
$c_i \triangleq$ the cost (R/ton) of coal type $i \in \boldsymbol{I}$.
$s_i \triangleq$ the expected strength measure of coal type $i \in \boldsymbol{I}$ after coking.
$p_{i,j}^A \triangleq$ the AMSA estimated parameter $j \in \boldsymbol{J}$ of coal type $i \in \boldsymbol{I}$.
$L_j^{\min} \triangleq$ the lower limit for parameter $j \in \boldsymbol{J}$ in the blend.
$L_j^{\max} \triangleq$ the upper limit for parameter $j \in \boldsymbol{J}$ in the blend.
$b_i \triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases}$
$d_i \triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases}$
$t \triangleq$ the total amount (tons) of coal in the blend.
$G_c \triangleq$ the goal for the the cost objective.
$G_s \triangleq$ the goal for the the strength objective.
$d_c \triangleq$ the cost deviational variable.
$d_s \triangleq$ the strength deviational variable.
$d_c^N \triangleq$ the normalised cost deviational variable.
$d_s^N \triangleq$ the normalised strength deviational variable.
$w_c \triangleq$ the cost weighting variable.
$w_s \triangleq$ the strength weighting variable.
$E(\widetilde{p}_{i,j}) \triangleq$ the deterministic component of parameter $j \in \boldsymbol{J}$ of coal type $i \in \boldsymbol{I}$.
$\sigma_{i,j}^2 \triangleq$ the stochastic component of parameter $j \in \boldsymbol{J}$ of coal type $i \in \boldsymbol{I}$.

The model is then formulated:

$$\min z_d = w_c d_c^N + w_s d_s^N - 10^{-12} z_c + 10^{-7} z_s \tag{C.126}$$

subject to:

$$\sum_{i \in \boldsymbol{I}} x_i c_i - d_c = G_c \tag{C.127}$$

$$\sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} + d_s = G_s \tag{C.128}$$

$$d_c^N = \frac{d_c}{G_c} \tag{C.129}$$

$$d_s^N = \frac{d_s}{G_s} \tag{C.130}$$

$$t = 2\,996 \tag{C.131}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{C.132}$$

$$tL_j^{\min} \leq \sum_{i \in \boldsymbol{I}} x_i E(\widetilde{p}_{i,j}) + \Omega \left(\sum_{i \in \boldsymbol{I}} x_i^2 \sigma_{i,j}^2 \right)^{\frac{1}{2}} \leq tL_j^{\max} \qquad \forall j \in \boldsymbol{J} \tag{C.133}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{C.134}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{C.135}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{C.136}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{C.137}$$

$$x_9 \leq 60 \tag{C.138}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{C.139}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.140}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.141}$$

C.11 Preemptive optimisation with approximation approach

This model is the full formulation of the preemptive optimisation scenario approach model, found in Subsection 4.2.1. The model is formulated by letting $\boldsymbol{I}$ being the set of coals, such that $i \in \boldsymbol{I}$, $\boldsymbol{J}$ being the set of coal parameters, such that $j \in \boldsymbol{J}$, and $\nu = 1, \ldots, N$, which is the set of samples.

The model parameters are defined as follows:

$$
\begin{aligned}
x_i &\triangleq \text{the amount (tons) of coal type } i \in \boldsymbol{I} \text{ in the blend.} \\
c_i &\triangleq \text{the cost (R/ton) of coal type } i \in \boldsymbol{I}. \\
s_i &\triangleq \text{the expected strength measure of coal type } i \in \boldsymbol{I} \text{ after coking.} \\
p^E_{i,j} &\triangleq \text{the estimated expected value of parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
L^{\min}_j &\triangleq \text{the lower limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
L^{\max}_j &\triangleq \text{the upper limit for parameter } j \in \boldsymbol{J} \text{ in the blend.} \\
b_i &\triangleq \begin{cases} 1 & \text{if coal type } i \in \boldsymbol{I} \text{ is in the blend;} \\ 0 & \text{otherwise.} \end{cases} \\
d_i &\triangleq \begin{cases} 1 & \text{if } b_i = 0, \text{ where } i \in \boldsymbol{I}; \\ 0 & \text{otherwise.} \end{cases} \\
t &\triangleq \text{the total amount (tons) of coal in the blend.} \\
G_c &\triangleq \text{the goal for the the cost objective.} \\
E(\widetilde{p}_{i,j}) &\triangleq \text{the deterministic component of parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}. \\
\sigma^2_{i,j} &\triangleq \text{the stochastic component of parameter } j \in \boldsymbol{J} \text{ of coal type } i \in \boldsymbol{I}.
\end{aligned}
$$

The model is then formulated:

$$\max z_s = \sum_{i \in \boldsymbol{I}} \frac{x_i s_i}{t} \tag{C.142}$$

subject to:

$$\sum_{i \in \boldsymbol{I}} x_i c_i = G_c \tag{C.143}$$

$$t = 2\,996 \tag{C.144}$$

$$\sum_{i \in \boldsymbol{I}} x_i = t \tag{C.145}$$

$$tL^{\min}_j \leq \sum_{i \in \boldsymbol{I}} x_i E(\widetilde{p}_{i,j}) + \Omega \left(\sum_{i \in \boldsymbol{I}} x_i^2 \sigma^2_{i,j} \right)^{\frac{1}{2}} \leq tL^{\max}_j \qquad \forall j \in \boldsymbol{J} \tag{C.146}$$

$$x_i \leq t b_i \qquad \forall i \in \boldsymbol{I} \tag{C.147}$$

$$b_i \leq t(1 - d_i) \qquad \forall i \in \boldsymbol{I} \tag{C.148}$$

$$30 - x_i \leq t d_i \qquad \forall i \in \boldsymbol{I} \tag{C.149}$$

$$4 \leq \sum_{i \in \boldsymbol{I}} b_i \leq 8 \tag{C.150}$$

$$x_9 \leq 60 \tag{C.151}$$

$$x_i \geq 0, \text{ and as integer} \qquad \forall i \in \boldsymbol{I} \tag{C.152}$$

$$b_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.153}$$

$$d_i \in \{0, 1\} \qquad \forall i \in \boldsymbol{I} \tag{C.154}$$

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

Appendix D

Scenarios

Included in this appendix are the results/scenarios that are generated by solving the various models.

Table D.1: Scenarios from goal programming model using AMSA's expected values.

			Amount of coal type i in blend (tons)								
Scenario	Cost	Strength	1	2	3	4	5	6	7	8	9
1	8 353 060	61.9811	0	0	1 002	32	0	1 902	0	0	60
2	8 354 310	62.2315	0	0	1 001	0	30	1 905	0	0	60
3	8 362 270	62.3931	30	0	995	0	0	1 911	0	0	60
4	8 763 940	68.5538	30	0	759	0	0	2 147	0	0	60
5	8 774 990	68.7171	0	0	741	0	0	2 165	0	30	60
6	8 813 410	69.1336	0	0	716	0	0	2 165	0	55	60
7	9 826 020	80.0245	0	0	56	0	0	2 154	0	726	60
8	9 865 810	80.4482	0	0	30	0	0	2 153	0	753	60
9	9 928 550	81.0846	30	0	0	0	0	2 135	0	771	60
10	10 485 300	85.6601	1 035	0	0	0	0	1 532	0	369	60
11	10 535 100	85.8909	1 060	45	0	0	0	1 505	0	326	60
12	10 552 300	85.9696	1 068	61	0	0	0	1 496	0	311	60
13	10 824 600	87.2079	1 196	313	0	0	0	1 352	0	75	60
14	10 875 400	87.4388	1 220	360	0	0	0	1 325	0	31	60
15	10 910 900	87.5961	1 235	394	0	0	0	1 307	0	0	60
16	10 914 800	87.6077	1 236	394	0	0	0	1 307	0	0	59
17	10 929 400	87.6413	1 235	396	0	0	0	1 310	0	0	55
18	10 969 100	87.7307	1 231	402	0	0	0	1 319	0	0	44
19	11 020 000	87.8429	1 226	410	0	0	0	1 330	0	0	30
20	11 094 300	87.9264	1 199	395	0	0	0	1 372	0	30	0
21	11 143 300	87.9635	1 231	413	0	30	0	1 322	0	0	0

Table D.2: Scenarios from weighted sum of objectives model.

			Amount of coal type i in blend (tons)								
Scenario	Cost	Strength	1	2	3	4	5	6	7	8	9
1	8 352 518	61.9875	0	0	1 002	31	0	1 903	0	0	60
2	8 354 308	62.2315	0	0	1 001	0	30	1 905	0	0	60
3	8 362 270	62.3931	30	0	995	0	0	1 911	0	0	60
4	8 763 942	68.5538	30	0	759	0	0	2 147	0	0	60
5	8 774 988	68.7171	0	0	741	0	0	2 165	0	30	60
6	8 813 413	69.1336	0	0	716	0	0	2 165	0	55	60
7	9 826 018	80.0245	0	0	56	0	0	2 154	0	726	60
8	9 865 815	80.4482	0	0	30	0	0	2 153	0	753	60
9	9 928 545	81.0846	30	0	0	0	0	2 135	0	771	60
10	10 485 315	85.6601	1 035	0	0	0	0	1 532	0	369	60
11	10 535 120	85.8909	1 060	45	0	0	0	1 505	0	326	60
12	10 552 347	85.9696	1 068	61	0	0	0	1 496	0	311	60
13	10 824 607	87.2079	1 196	313	0	0	0	1 352	0	75	60
14	10 875 445	87.4388	1 220	360	0	0	0	1 325	0	31	60
15	10 910 932	87.5961	1 235	394	0	0	0	1 307	0	0	60
16	10 914 840	87.6077	1 236	394	0	0	0	1 307	0	0	59
17	10 929 388	87.6413	1 235	396	0	0	0	1 310	0	0	55
18	10 969 124	87.7307	1 231	402	0	0	0	1 319	0	0	44
19	11 019 988	87.8429	1 226	410	0	0	0	1 330	0	0	30
20	11 094 292	87.9264	1 199	395	0	0	0	1 372	0	30	0
21	11 143 322	87.9635	1 231	413	0	30	0	1 322	0	0	0

Table D.3: Scenarios from goal programming model using estimated expected values.

			Amount of coal type i in blend (tons)								
Scenario	Cost	Strength	1	2	3	4	5	6	7	8	9
1	10 069 800	69.8839	1 783	471	682	0	0	0	0	0	60
2	10 136 800	70.8238	1 783	477	645	0	0	31	0	0	60
3	10 402 500	74.5219	1 782	503	499	0	0	152	0	0	60
4	10 433 400	74.9527	1 782	506	482	0	0	166	0	0	60
5	10 581 000	77.0015	1 781	521	401	0	0	233	0	0	60
6	10 588 500	77.1016	1 781	522	397	0	0	236	0	0	60
7	10 590 400	77.1226	1 780	523	396	0	0	237	0	0	60
8	10 648 500	77.7262	1 755	554	367	0	0	260	0	0	60
9	10 897 500	80.305	1 648	687	243	0	0	358	0	0	60
10	11 385 600	85.3568	1 438	948	0	0	0	550	0	0	60
11	11 493 800	85.6023	1 444	962	0	0	0	558	0	0	32
12	11 501 800	85.6203	1 445	963	0	0	0	558	0	0	30
13	11 620 200	85.8788	1 422	978	0	0	30	566	0	0	0
14	11 621 700	85.8811	1 425	978	0	0	30	563	0	0	0
15	11 622 300	85.8815	1 424	978	0	0	32	562	0	0	0

Appendix E

Model evaluation data

Table E.1: Average deviations from limits.

	Blend parameter j											
Model No.	1	2	3	4	5	6	7	8	9	10	11	12
1	0.01	0.006	0.30	0.10	-1.00	0.12	-5.79	3.07	-2.70	3.05	-3.49	5.45
2	0.01	0.039	0.46	0.08	-1.33	1.35	-4.62	3.09	-1.12	3.20	-0.66	5.76
3	0.00	0.038	0.55	0.16	-0.34	4.08	-0.79	3.03	2.99	3.35	3.59	5.77
4	0.00	0.001	0.01	0.02	0.22	4.29	-0.02	1.68	3.77	3.54	3.52	6.09
5	0.00	0.001	0.04	0.04	0.55	5.00	1.00	1.45	4.88	3.58	4.69	6.08
6	0.02	0.046	0.78	0.36	2.14	6.59	5.67	4.45	8.52	3.36	7.92	4.87
7	0.02	0.047	0.78	0.36	2.14	6.59	5.69	4.45	8.53	3.36	7.93	4.86
8	0.01	0.030	0.81	0.36	1.90	7.85	6.64	4.18	9.60	3.49	8.93	5.09
9	0.02	0.047	0.87	0.38	2.36	8.22	7.64	4.16	10.66	3.50	10.33	4.99

Table E.2: Variance of estimated parameter distributions.

	Blend parameter j											
Model No.	1	2	3	4	5	6	7	8	9	10	11	12
1	0.003	0.009	0.12	0.03	0.38	1.30	1.72	0.90	2.52	0.61	2.29	1.74
2	0.006	0.009	0.17	0.04	0.83	1.53	1.99	1.99	2.74	0.81	2.78	2.27
3	0.007	0.009	0.18	0.04	0.93	1.37	1.83	2.24	2.53	0.78	3.02	2.12
4	0.006	0.013	0.21	0.04	0.45	1.48	2.02	1.60	2.25	0.62	2.42	1.60
5	0.006	0.013	0.20	0.04	0.44	1.42	1.95	1.56	2.13	0.59	2.35	1.50
6	0.004	0.011	0.17	0.02	0.47	1.94	2.62	0.90	2.02	0.70	2.33	1.45
7	0.004	0.011	0.17	0.02	0.47	1.95	2.63	0.90	2.03	0.70	2.33	1.46
8	0.004	0.010	0.16	0.02	0.53	1.64	2.22	1.15	2.01	0.62	2.44	1.36
9	0.005	0.011	0.17	0.03	0.58	1.88	2.54	1.25	2.06	0.71	2.50	1.51

[git] • Branch: master @ b54f423; Author: Daniel Charters; Date: 2020-10-07 11:50:27 +0200 •

Appendix F

Plagiarism declaration

I, Daniel John Charters, a student at the University of Pretoria in the Department of Industrial and Systems Engineering, with student number u17000174, declare:

1. I understand what plagiarism is and I am aware of the University's policy in this regard.
2. I declare that this is my own original work.
3. Where other people's work has been used (either from a printed source, internet or any other source) this has been carefully acknowledged and referenced in accordance with departmental requirements.
4. I have not used another student's past work to hand in as my own.
5. I have not allowed, and will not allow, anyone to copy my work with the intention of handing it in as his/her own work.

Signed on October 7, 2020:

Daniel John Charters