Decomposition and duality based approaches to Stochastic Integer Programming

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

Jeffrey Christiansen
BSc Adv. Hons., Monash University

School of Science
College of Science, Engineering and Health
RMIT University

June 2018

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Jeffrey Christiansen

June 21, 2018

Acknowledgements

First and foremost, I would like to thank my supervisors Andrew Eberhard (whose door always seems to be open) and Natashia Boland (who battled through various inconveniences and injuries) for their excellent guidance and steadfast support. For the same reasons, my heartfelt thanks to Brian Dandurand, who acted as my unofficial third supervisor (despite this appearing nowhere in his job description). I couldn't have asked for a better team.

The research contained in this thesis has been carried out within a larger collaborative research project. Besides Andrew, Natashia and Brian, the other members of this project were Fabricio Oliveira and Jeff Linderoth, both of whom made substantial contributions to both our research and my education as a researcher. James Luedtke joined forces with us for our first paper and also merits mention here. The contributions made to the research in this thesis by the aforementioned collaborators is documented in the Preface on the following page.

Thanks are also due to the friendly support staff at the National Computational Infrastructure (NCI) and the (unfortunately now retired) V3 Alliance (VPAC) for their technical assistance. Without their trouble-shooting expertise the experimental component of this research would have taken much longer, in terms of both implementation and execution.

Finally, I'm very grateful to my family, friends and colleagues for their encouragement and support. I expect to speak with many of you personally, but in particular I'd like to thank the various people I've lived with over the last four years for their understanding when deadlines became pressing, the RMIT Optimisation group for broadening my horizons (particularly Vera Roshchina, who organised the group for several years), and my office companions and fellow travellers on the PhD track for always being ready to have a laugh.

Preface

Some of the material in this thesis is based on published multi-author papers.

- Chapter 4 is based on [16]. The other co-authors of this paper are Natashia Boland, Brian Dandurand, Andrew Eberhard, Jeff Linderoth, James Luedtke and Fabricio Oliveira.
- Chapter 5 is based on [17]. The other co-authors of this paper are Brian Dandurand, Fabricio Oliveira, Andrew Eberhard and Natashia Boland.
- Chapter 6 is based on [93]. The other co-authors of this paper are Fabricio Oliveira, Brian Dandurand and Andrew Eberhard.

From these sections and chapters, the experimental results in Sections 4.3 and 5.3 and the theoretical results developed in Section 6.2 were completed by Jeffrey Christiansen. The remainder of these chapters is the result of equal collaboration between the authors of each respective paper. The work in the chapters not previously listed was also completed by Jeffrey Christiansen.

Contents

List of Figures

List of Tables

Abstract

Stochastic Integer Programming is a variant of Linear Programming which incorporates integer and stochastic properties (i.e. some variables are discrete, and some properties of the problem are randomly determined after the first-stage decision). A Stochastic Integer Program may be rewritten as an equivalent Integer Program with a characteristic structure, but is often too large to effectively solve directly. In this thesis we develop new algorithms which exploit convex duality and scenario-wise decomposition of the equivalent Integer Program to find better dual bounds and faster optimal solutions. A major attraction of this approach is that these algorithms will be amenable to parallel computation.

Chapter 1

Introduction

Optimisation is a field of mathematics which provides tools for making optimal decisions in a given situation. Decision problems are classified by how they can be mathematically represented. This classification determines which algorithms can be effectively and efficiently applied to any given problem.

A powerful modelling technique in optimisation is Linear Programming, which allows us to solve decision problems which can be represented as Linear Programs. A Linear Program consists of a linear objective function which must be minimised or maximised subject to a set of linear constraints. We can represent a generic linear program with n decision variables and m linear inequality constraints in the following form:

$$\begin{aligned} \zeta^{LP} = & \min_x c_1 x_1 + \cdots + c_n x_n \\ & \text{s.t. } a_{1,1} x_1 + \cdots + a_{1,n} x_n \leqslant b_1 \\ & \quad \vdots \\ & \quad a_{m,1} x_1 + \cdots + a_{m,n} x_n \leqslant b_m \end{aligned} \tag{1.1}$$

To abbreviate this formulation, define

$$x := \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad c := \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}, \quad \text{and} \quad b := \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

as the vectors of decision variables, objective coefficients and linear constraint constants respectively, and

$$A := \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{bmatrix}$$

as the matrix of linear constraint coefficients. We can now represent (1.1) using matrix algebra in the condensed form:

$$\begin{aligned} \zeta^{LP} &= \min_x c^T x \\ \text{s.t. } & Ax \leqslant b \end{aligned} \tag{1.2}$$

We will use similar abbreviations when formulating optimisation problems in subsequent chapters. In fact, since additional indices to distinguish variable vectors will frequently be required, from this point onward the notation x_i will not denote the ith variable in the vector x unless this is specifically indicated in the text.

Linear Programming is a valuable tool for modelling decision problems but has several limitations. We will focus on two of these limitations. First, a Linear Program cannot model discrete decisions (such as a yes/no decision); all decision variables in a Linear Program must be continuous. Second, a Linear Program cannot model a problem in which some of the problem data is unavailable when the first decision needs to be made. Discrete decisions and imperfect knowledge are common properties of real-life decision problems, and a more general framework than Linear Programming is necessary to model these problems accurately.

Stochastic Mixed-Integer Linear Programming, typically shortened to Stochastic Integer Programming, generalises the concept of Linear Programs to account for these limitations. Incorporating binary and integer variables into a standard Linear Program formulation is a straightforward process, although solving the resulting non-convex optimisation problem is frequently difficult. On the other hand, there are several valid approaches to incorporating a stochastic element into a Linear Program so that the initial decisions must be made in conditions of uncertainty. The research in this thesis focuses on a scenario-based approach in which initial decisions must be made before the outcome scenario is known, then recourse decisions are made in reaction to the outcome scenario.

A well-known algorithm for solving scenario-based Stochastic Programming problems is the Progressive Hedging algorithm of Rockafellar and Wets [98]. This algorithm calculates what the best initial decision would be if the outcome scenario were known beforehand (for each possible outcome scenario), then tries to find an intermediate solution which is reasonable for all outcome scenarios by alternately updating the decision variables, a set of averaged consensus variables, and the dual multipliers.

The Progressive Hedging algorithm is an effective approach for non-integer Stochastic Programs

in that it is guaranteed to converge to the global optimal solution. The algorithm may also be applied to Stochastic Integer Programs but the guarantee of optimal convergence does not apply; since the averaging operation is not well defined on an integer set this outcome is not surprising.

The main contributions of this thesis fall into two major classes. First, we will define algorithms for Stochastic Integer Programming which improve on Progressive Hedging and more generally the state of the art in Stochastic Integer Programming, either in the direction of calculating dual bounds or in finding high-quality feasible solutions. These algorithms are based on a similar alternating-update framework, but with structures and goals that are better suited to an integer programming environment. Second, we will explore the underlying theory which motivates and justifies the construction of these algorithms.

The main chapters of this thesis are structured as follows.

Chapter 2 formalises and elaborates on the background material presented in the Introduction. This chapter also contains a review of the current literature in stochastic integer optimisation and related fields of mathematics.

Chapter 3 describes the Frank-Wolfe method, several of its generalisations, and their applications. Section 3.2 extends existing knowledge by demonstrating that these Frank-Wolfe-type methods can be applied to solving the convex hull relaxation of integer programs. This technique is employed in Chapters 4 and 5. Section 3.3 demonstrates that the Frank-Wolfe method can be applied to non-smooth optimisation problems.

Chapters 4 and 5 use a modified version of the Progressive Hedging algorithm to compute the ordinary Lagrangian dual bound of a Stochastic Integer Program directly. The primary algorithmic difficulty in this approach is in solving subproblems over the convex hull of the feasible set. The main results of Chapters 4 and 5 are included in the published papers [16] and [17] respectively.

Chapter 6 takes the alternate approach of constructing penalty functions which result in strong augmented Lagrangian duality. The primary algorithmic difficulty in this approach is in dealing with the restrictions which this places on the penalty function (in particular, non-differentiability). Having done this, an alternating update algorithm is used to obtain high-quality primal feasible solutions. The main results of Chapter 6 are included in the published paper [93].

Chapter 7 proposes an modified version of the algorithm developed in Chapter 6 which has stronger theoretical properties when applied to a Stochastic Mixed-Integer Program with only in-

teger variables in the first stage.

Chapter 8 summarises and concludes the results presented in the thesis and outlines potential directions for future research.

Chapter 2

Background and Literature Review

2.1 Introduction

2.1.1 Integer Linear Programming

A **Mixed Integer Linear Program** (MILP), typically abbreviated to **Mixed Integer Program** (MIP) or **Integer Program** (IP), may be written in the following general form:

$$\begin{aligned} \zeta^{IP} &= \min c^T x \\ \text{s.t. } & Ax \leqslant b \\ & x \geqslant 0 \\ & x \in \mathbb{R}^{n-q} \times \mathbb{Z}^q \end{aligned} \tag{2.1}$$

This MIP has n decision variables, represented by the vector x. $n - q$ of these variables are continuous, and the remaining q are discrete or integer variables. The vector c has the same length as x and defines the coefficients for each variable in the linear objective function. A is a $l \times n$ matrix and b is a vector of length l, which together define the constraints on the decisions variables as a set of l inequalities.

A particular choice of decision variables x is called a solution. If a solution satisfies all of the conditions of the problem, then it is a feasible solution. If a feasible solution yields the best possible objective value across all feasible solutions, then it is an optimal solution. In this case the objective is being minimised, so the best possible objective value is the smallest.

Equation (2.1) may appear restrictive since it does not account for differences in objective and constraint type, or for different variable bounds. However, it is possible to reformulate any

integer linear program with equalities, negative variables, or a maximisation objective in the form of Equation (2.1), and so any results proved for Equation (2.1) may be generalised to other forms.

In practice it is often more intuitive to construct a MIP using a combination of equalities and inequalities, and with a variety of variable bounds.

2.1.2 Stochastic Programming

The representation of a problem as a MIP assumes that all decisions may be made in advance with full knowledge of what effect they will have. Unfortunately this is not always the case. In many practical problems it is necessary to make decisions before everything about the problem is known with certainty. For example, when scheduling train networks the exact number of passengers each day and the occurrence of any mechanical faults are not known with certainty when the timetable is arranged. In agriculture, the weather conditions which will occur in a particular year are not known when the crops are planted.

In both of these examples and in many other practical problems the future conditions are not known with certainty. It is possible to model this uncertainty with a probability distribution. For example, the probability distributions associated with the set of outcomes corresponding, for example, to high or low passenger traffic, good or bad weather, and the frequency of mechanical faults may be estimated using previous data, which allows a partially informed decision to be made.

Different possible outcomes are modelled as separate **scenarios**, each of which is assigned a probability of occurring. Stochastic programs aim to find the decision which has the best average outcome across all scenarios.

There are some problems which arise when applying the answer obtained by solving a stochastic program to the real world. First, the assignment of probabilities to outcome scenarios is often made without knowledge of the exact actual probabilities of all outcomes; an informed guess must be made. Second, it may not be practical to examine all possible outcome scenarios; the number of outcome scenarios may be prohibitively large, or infinite in the case of a continuous random distribution. This second problem is addressed by taking a representative sample of the set of possible outcomes as the scenario space for the purposes of obtaining a tractable stochastic program.

These issues mean that the optimal solution of a stochastic program will not necessarily be exactly "optimal" in the real world. Therefore, algorithms for stochastic programs frequently prioritise

finding a "good" solution quickly over finding an "optimal" solution.

The work presented herein will generally employ a decision stage based approach to modelling the stochastic elements of a given problem. These models may have two or more stages. A decision problem with two stages will be referred to as a **two-stage** stochastic program. A problem with more than two decision stages will be referred to as a **multi-stage** stochastic program.

In a two-stage stochastic program, the **first stage decisions** or **initial decisions** are made before the outcome scenario is known. After the outcome scenario is determined, the **second stage decisions** or **recourse decisions** can be made. For example, in the hypothetical train scheduling problem above, if a mechanical fault occurs it may be possible to reschedule other trains to cover the gap.

A multi-stage stochastic program is structured similarly. However, instead of fully unveiling the final outcome scenario in a single step, it is revealed in several discrete steps. Recourse decisions may be made at each of these steps. The structure of the outcome scenarios is typically represented with a tree structure.

An alternative approach to modelling stochastic aspects of a problem is **chance-constrained programming**. In a chance-constrained program, one or more constraints must be satisfied with a given probability (or degree of certainty). A chance constraint may be written in the form

$$P\left(f(x) \leqslant b\right) \geqslant p,$$

which has the meaning "x must be chosen such that the probability that the constraint $f(x) \leqslant b$ is satisfied is greater than or equal to p".

Chance constraints are more appropriate to some classes of problems than the two- or multi-stage approach, and can more directly tackle problems involving continuous random distributions.

Chance-constrained programming is outside the scope of this project. More detail about this approach may be found in, e.g. [84].

2.1.3 Stochastic Programming Formulation

Two-stage stochastic programs have a much simpler structure than multi-stage stochastic programs. For the purposes of simplicity and clarity, the following introductory explanation of decision-stage based Stochastic Program problem formulation will focus on the two-stage case.

Two-stage stochastic programs are modelled with two sets of variables, corresponding to the first-stage and second-stage decisions. Since the first-stage decisions must be made before determination of the random variable, there is a single set of **first-stage variables** across all possible scenarios. Since the second-stage decisions may be made after the random outcome is observed, and may therefore respond to it, there are different sets of **second-stage variables** for each scenario.

A **Two-stage Stochastic Mixed Integer Linear Program** (SMILP), abbreviated here to **Stochastic Integer Program** (SIP), may be written in the following general form:

$$\begin{aligned} \zeta^{SIP} = & \min_x c^T x + \sum_{s \in S} p_s [Q(x,s)] \\ & \text{s.t. } Ax \leqslant b \\ & \quad x \geqslant 0 \\ & \quad x \in \mathbb{R}^{n-q} \times \mathbb{Z}^q \end{aligned} \tag{2.2}$$

where

$$\begin{aligned} Q(x,s) = & \min_y d_s^T y \\ & \text{s.t. } W_s y \leqslant h_s - T_s x \\ & \quad y \geqslant 0 \\ & \quad y \in \mathbb{R}^{m-r} \times \mathbb{Z}^r \end{aligned} \tag{2.3}$$

In an SIP, the second-stage problem for each possible scenario is modelled as a MIP, with the first-stage decision used as a parameter. The expected value of the second-stage problem across all scenarios is added to the first-stage problem as a penalty or incentive in the objective function.

The set S is the set of all possible outcome scenarios s. The expected value of the second-stage problem, given a first stage decision, is found by taking the average value of the second-stage decision problem across all possible scenarios $s \in S$, weighted by the probability p_s of each scenario s. We will always assume that $\sum_{s \in S} p_s = 1$.

This IP has n first-stage decision variables, represented by the vector x, and m second-stage decision variables, represented by the vector y. A full solution to the two-stage SIP consists of a single first-stage decision and multiple second-stage decisions, each of which corresponds to an outcome scenario. A particular choice of decision variables for each stage and outcome scenario is sometimes referred to as a **policy**. This term is used because we cannot make all of the decisions required in the problem which the stochastic program represents at a single point in time, but we

can initially define a policy which chooses decisions when and as appropriate. To be feasible a policy must satisfy all constraints on both the first- and second-stage variables.

In the context of SIP formulations such as (2.2) our problem data is represented as follows. c and d are vectors of length n and m respectively which define the objective functions for the first-stage and second-stage problems. A is a matrix and b is a vector which together define the first-stage constraints $Ax \leqslant b$. As the second-stage decision y has not yet been made in the first stage, the first-stage constraints do not refer to those decision variables. Similarly W_s and T_s are matrices and h_s is a vector which together define the second-stage constraints $W_s y \leqslant h_s - T_s x$ for each scenario s. Within the $Q(x, s)$ subproblem the first-stage decision has already been made and is treated as a constant, so the $T_s x$ term is written on the right-hand side of the second-stage constraints in this formulation.

This formulation does not give any explicit guarantees that the second-stage problems $Q(x, s)$ will be bounded or feasible for all possible first-stage decisions x. To correct this we may enforce an additional **recourse** condition. ζ^{SIP} as defined in (2.2) and (2.3) has **relatively complete recourse** if $Q(x, s) < \infty$ for all x which satisfy the first-stage constraints $Ax \leqslant b$ and $x \geqslant 0$. The stronger condition of **complete recourse** holds if $W_s y \leqslant z$ for all scenarios s and vectors z of real numbers with the appropriate dimension.

The formulation given in (2.2) for ζ^{SIP} has the disadvantage that most of the complexity is concealed within the $Q(x, s)$ function. To address this problem, ζ^{SIP} can be reformulated as a single **deterministic equivalent** MIP as follows:

$$\begin{aligned}
\zeta^{SIP} = \min_{x, y_1, \ldots, y_{|S|}} \; & c^T x + \sum_{s \in S} p_s \left[d_s^T y_s \right] \\
\text{s.t. } & Ax \leqslant b \\
& T_s x + W_s y_s \leqslant h_s \quad \forall s \in S \\
& x \geqslant 0, y \geqslant 0 \\
& x \in \mathbb{R}^{n-q} \times \mathbb{Z}^q \\
& y \in \mathbb{R}^{m-r} \times \mathbb{Z}^r
\end{aligned} \tag{2.4}$$

This is an ordinary MIP and in principle may be solved using techniques for solving ordinary mixed integer programs. For real world problems, since there is a separate set of variables and constraints for each scenario, the MIP is invariably very large and therefore exceptionally difficult to solve in

this way.

However, this MIP does have a particular structure imposed by its origins as an SIP, which allows specialised algorithms to solve it much more easily. The principle behind these algorithms is to separate, or **decompose**, the problem into smaller subproblems.

Algorithms based on **stage-wise decomposition**, alternately named **primal decomposition**, separate the variables and constraints based on which decision stage they correspond to. The L-shaped method (see e.g. [71]) is a standard stage-wise decomposition method. Stage-wise decomposition methods, like Benders decomposition, are often strongly based on duality and are thus open to further development with the application of new duality methods for Integer Programming, but are less well adapted to parallel decomposition.

Algorithms based on **scenario-wise decomposition**, alternately named **dual decomposition**, separate the variables and constraints based on which outcome scenario they correspond to. Lagrangian duality gives a theoretical basis for scenario-wise decomposition on which many algorithms have been constructed. Since there are a large number of scenarios in an SIP, and the subproblems corresponding to each scenario are not strongly dependent on each other, scenario-wise decomposition offers a great deal of scope for parallel computation. As such, scenario-wise decomposition methods will be a primary focus of this research.

2.1.4 Introduction to Lagrangian Duality

In some constrained optimisation problems, a particular subset of the constraints make the problem significantly more difficult to solve. Conversely, if these constraints could be removed from the problem it would be easier to solve. These constraints generally provide important information about the problem, so simply ignoring them is not practical and will not result in a useful answer. However, it is possible to move this information from the constraint into the objective using Lagrangian duality.

Consider a general MIP of the form:

$$\begin{aligned} \zeta^{IP} = & \min_x c^T x \\ & \text{s.t. } Qx = r \\ & x \in X \end{aligned} \tag{2.5}$$

where the term $x \in X$ represents the "easy" constraints as well as bounds and integrality of the variables. For the purposes of this discussion the s constraints represented by $Qx = r$ are considered difficult constraints. In the context of duality, the x variables are referred to as **primal variables**.

The **Lagrangian** corresponding to ζ^{IP} is:

$$L(x, \lambda) = c^T x + \lambda^T (Qx - r) \tag{2.6}$$

The corresponding **Lagrangian dual function** is

$$\begin{aligned} \zeta^{LR}(\lambda) = & \min_x L(x, \lambda) \\ & \text{s.t. } x \in X \end{aligned} \tag{2.7}$$

and the corresponding **Lagrangian dual problem** is

$$\zeta^{LD} = \max_\lambda \zeta^{LR}(\lambda) \tag{2.8}$$

The vector λ of **dual variables** has length s. In the case where x consists of only continuous variables, the initial program is convex and the Lagrangian dual is a **strong dual**, meaning that $\zeta^{LD} = \zeta^{IP}$ (e.g. [100, Theorem 3.27]).

If x contains integer variables the problem is no longer convex, and a **duality gap** between ζ^{IP} and ζ^{LD} may occur. ζ^{LD} is still guaranteed to be a **weak dual** to ζ^{IP} — it yields a lower bound on the value of ζ^{IP}, which provides information and may contribute to finding the solution for ζ^{IP} via another method such as Branch-and-Bound.

Definition 2.1 *Given a set X, The **convex hull** of X, denoted* $\text{conv}(X)$*, is the smallest convex set containing X. A constructive definition will be given in Section 2.3.2.*

A primal characterisation of the Lagrangian dual problem is

$$\begin{aligned} \zeta^{LD} = & \min_x c^T x \\ & \text{s.t. } Qx = r \\ & x \in \text{conv}(X) \end{aligned} \tag{2.9}$$

(see e.g. [46]). This is a primal problem which has the same optimal value as the Lagrangian dual problem. Furthermore it is a continuous programming problem, as taking the convex hull of the

feasible set X removes the integer restriction on the decision variables. In practice it is challenging to calculate $\text{conv}(X)$ and model it with linear constraints.

The **augmented Lagrangian** corresponding to ζ^{IP} is:

$$L_\rho(x, \lambda) = c^T x + \lambda^T (Qx - r) + \psi^\rho(Qx - r) \tag{2.10}$$

The corresponding **augmented Lagrangian dual function** is

$$\begin{aligned} \zeta_\rho^{LR+}(\lambda) = &\min_x L_\rho(x, \lambda) \\ &\text{s.t. } x \in X \end{aligned} \tag{2.11}$$

and the corresponding **augmented Lagrangian dual problem** is

$$\zeta_\rho^{LD+} = \max_\lambda \zeta_\rho^{LR+}(\lambda) \tag{2.12}$$

The augmenting term $\psi^\rho(Qx - r)$ is intended to penalise decisions which violate the difficult constraints $Qx = r$. The **augmenting function** $\psi^\rho(\cdot)$ has the properties $\psi^\rho(0) = 0$ and $\psi^\rho(\boldsymbol{u}) > 0$ for all $\boldsymbol{u} \neq 0$, and determines the character of the penalty for violation of the hard constraints. The **penalty parameter** ρ is a strictly positive scalar parameter which determines the size of the penalty. The typical augmenting function used in most applications of augmented Lagrangian duality is:

$$\psi^\rho(\boldsymbol{u}) = \frac{\rho}{2} \|\boldsymbol{u}\|_2^2 \tag{2.13}$$

Similarly to the ordinary Lagrangian dual, the augmented Lagrangian dual is a strong dual to ζ^{IP} if the decision variables x are continuous (e.g. [100, Theorem 4.30]). Under some conditions, the augmented Lagrangian dual may also be strong for an integer problem; these conditions are discussed further in Section 2.2.1. Otherwise, it is a weak dual. The added term improves the convergence properties of algorithms which employ Lagrangian duality, although in general it has the undesirable side effect of destroying separability present in the original Lagrangian.

A primal characterisation of the augmented Lagrangian dual problem, due to Feizollahi et al. [37], is

$$\begin{aligned} \zeta_\rho^{LD+} = &\min_x c^T x + \rho\omega \\ &\text{s.t. } Qx = r \\ &(x, \omega) \in \text{conv}(S_\psi) \end{aligned} \tag{2.14}$$

where the set S_ψ is defined as follows:

$$S_\psi := \left\{(x, \omega) \in \mathbb{R}^{n+1} : \psi^\rho(Qx - r) \leqslant \omega, x \in X\right\} \tag{2.15}$$

2.1.5 Applying Lagrangian Duality to Stochastic Programming

To apply Lagrangian duality to Equation 2.4, it is necessary to express the difficult part of the problem (that the first-stage decisions must be the same for all scenarios) as a constraint. To do this, we separate the first-stage variables x into separate vectors x_s for all $s \in S$ and impose a **non-anticipativity constraint** to ensure that the first-stage decisions remain identical for all scenarios. The non-anticipativity constraint may be expressed in several ways, for example:

- Linking the first-stage decisions sequentially, i.e. $x_1 = x_2$, $x_2 = x_3$, ..., $x_{|S|-1} = x_{|S|}$.
- Linking one first-stage decision to all of the others, i.e. $x_1 = x_2$, $x_1 = x_3$, ..., $x_1 = x_{|S|}$.
- Linking all of the first-stage decisions to a new **non-anticipativity** or **consensus** variable $\bar{x}$, i.e. $\bar{x} = x_1$, $\bar{x} = x_2$, ..., $\bar{x} = x_{|S|}$.

For the sake of condensing notation, when describing an SIP, let $\boldsymbol{x}$ represent all of the first-stage decision vectors $\{x_s : s \in S\}$. Similarly let $\boldsymbol{y}$ represent all of the second-stage decision vectors $\{y_s : s \in S\}$, and $\boldsymbol{\lambda}$ represent all of the dual variable vectors $\{\lambda_s : s \in S\}$.

In the context of this formulation, a decision policy is composed of a first- and second-stage decision for each possible outcome scenario. If a policy satisfies the non-anticipative constraints, it is described as **non-anticipative** or **implementable**. If a policy satisfies all of the other constraints on the first and second stage variables it is described as **admissible**. A policy is **feasible** for ζ^{SIP} if and only if it is both implementable and admissible, since this means that it satisfies all of the constraints on the problem. If no other feasible policy results in a better objective value, then it is an optimal policy.

The following representation of ζ^{SIP} uses the consensus decision variable approach to represent the non-anticipativity constraints.

$$
\begin{aligned}
\zeta^{SIP} = & \min_{\boldsymbol{x},\boldsymbol{y},\bar{x}} \sum_{s\in S} p_s \left[c^T x_s + d_s^T y_s\right] \\
& \text{s.t. } Ax_s \leqslant b \quad \forall s \in S \\
& \quad T_s x_s + W_s y_s \leqslant h_s \quad \forall s \in S \\
& \quad x_s = \bar{x} \quad \forall s \in S \\
& \quad x_s \geqslant 0,\ y_s \geqslant 0 \quad \forall s \in S \\
& \quad x_s \in \mathbb{R}^{n-q} \times \mathbb{Z}^q \quad \forall s \in S \\
& \quad y_s \in \mathbb{R}^{m-r} \times \mathbb{Z}^r \quad \forall s \in S
\end{aligned}
\tag{2.16}
$$

The constraints here which define the feasibility of first- and second-stage decisions for a particular scenario may be abbreviated as $(x_s, y_s) \in K_s$, where

$$
K_s := \left\{(x,y) \mid Ax \leqslant b, T_s x + W_s y_s \leqslant h_s, x \in \mathbb{R}^{n-q} \times \mathbb{Z}^q, y \in \mathbb{R}^{m-r} \times \mathbb{Z}^r\right\}.
$$

This facilitates a more compact representation of the SIP:

$$
\begin{aligned}
\zeta^{SIP} = & \min_{\boldsymbol{x},\boldsymbol{y},\bar{x}} \sum_{s\in S} p_s \left[c^T x_s + d_s^T y_s\right] \\
& \text{s.t. } (x_s, y_s) \in K_s \quad \forall s \in S \\
& \quad x_s = \bar{x} \quad \forall s \in S
\end{aligned}
\tag{2.17}
$$

The **Lagrangian** corresponding to ζ^{SIP} is:

$$
L(\boldsymbol{x}, \boldsymbol{y}, \bar{x}, \boldsymbol{\lambda}) = \sum_{s\in S} p_s L_s(x_s, y_s, \bar{x}, \lambda_s), \tag{2.18}
$$

where

$$
L_s(x_s, y_s, \bar{x}, \lambda_s) = (c^T x_s + d_s^T y_s) + \lambda_s^T (x_s - \bar{x}). \tag{2.19}
$$

Note that the p_s scaling factor also applies to the dual multiplier term in this formulation. Since the dual variables are free we can multiply each of them by a scaling factor such as p_s without loss of generality. The corresponding **Lagrangian dual function** is

$$
\begin{aligned}
\zeta^{LR}(\boldsymbol{\lambda}) = & \min_{\boldsymbol{x},\boldsymbol{y},\bar{x}} L(\boldsymbol{x}, \boldsymbol{y}, \bar{x}, \boldsymbol{\lambda}) \\
& \text{s.t. } (x_s, y_s) \in K_s
\end{aligned}
\tag{2.20}
$$

and the corresponding **Lagrangian dual problem** is

$$
\zeta^{LD} = \max_{\boldsymbol{\lambda}} \zeta^{LR}(\boldsymbol{\lambda}). \tag{2.21}
$$

The primal characterisation of the Lagrangian dual corresponding to (2.9) is:

$$\begin{aligned} \zeta^{SIP} = &\min_{\boldsymbol{x},\boldsymbol{y},\bar{x}} \sum_{s\in S} p_s \left[c^T x_s + d_s^T y_s\right] \\ &\text{s.t. } (x_s, y_s) \in \operatorname{conv} K_s \quad \forall s \in S \\ &x_s = \bar{x} \quad \forall s \in S \end{aligned} \tag{2.22}$$

Define an augmenting function $\psi^\rho(\boldsymbol{u}) : \mathbb{R}^{n|S|} \to \mathbb{R}$ as follows:

$$\psi^\rho(\boldsymbol{u}) \equiv \psi^\rho(u_1, \ldots, u_s, \ldots, u_{|S|}) := \sum_{s\in S} p_s \psi_s^\rho(u_s) \tag{2.23}$$

The **augmented Lagrangian** corresponding to ζ^{SIP}, using the augmenting function (2.23), is:

$$L^\rho(\boldsymbol{x}, \boldsymbol{y}, \bar{x}, \boldsymbol{\lambda}) = \sum_{s\in S} p_s L_s^\rho(x_s, y_s, \bar{x}, \lambda_s), \tag{2.24}$$

where

$$L_s^\rho(x_s, y_s, \bar{x}, \lambda_s) = (c^T x_s + d_s^T y_s) + \lambda_s^T (x_s - \bar{x}) + \psi_s^\rho(x_s - \bar{x}). \tag{2.25}$$

The corresponding **augmented Lagrangian dual function** is

$$\begin{aligned} \zeta_\rho^{LR+}(\boldsymbol{\lambda}) = &\min_{\boldsymbol{x},\boldsymbol{y},\bar{x}} L^\rho(\boldsymbol{x}, \boldsymbol{y}, \bar{x}, \boldsymbol{\lambda}) \\ &\text{s.t. } (x_s, y_s) \in K_s \end{aligned} \tag{2.26}$$

and the corresponding **augmented Lagrangian dual problem** is

$$\zeta_\rho^{LD+} = \max_{\boldsymbol{\lambda}} \zeta_\rho^{LR+}(\boldsymbol{\lambda}). \tag{2.27}$$

Each of the constraints in Equation 2.26 contain only variables from a single scenario, and therefore may be separated by scenario. The augmented Lagrangian in the objective of (2.26) is not generally separable by scenario due to the augmenting term. However, if $\bar{x}$ is fixed the resulting objective and problem ζ_ρ^{LD+} is separable by scenario and much more tractable. This observation motivates algorithms which solve the overall problem by iteratively solving ζ_ρ^{LD+} for a fixed $\bar{x}$, then using the result to generate an improved value for $\bar{x}$. Similar approaches are possible for other non-anticipativity conditions.

Example 2.2 *Consider the following SIP with two first-stage variables x^1 and x^2 and a single second-stage variable y:*

$$\begin{aligned} \zeta^{SIP} = & \min_{x,y} x^1 + \frac{1}{2}(1000y_1) + \frac{1}{2}(1000y_2) \\ & s.t. \; -y_1 \leqslant x^1 - x^2 \\ & \quad -y_2 \leqslant x^1 + x^2 - 1 \\ & \quad x^1, x^2, y_1, y_2 \in \{0,1\} \end{aligned} \tag{2.28}$$

The set of feasible decisions in each scenario is

$$K_1 := \left\{(x^1, x^2, y_1) \in \{0,1\}^3 \mid -x^1 + x^2 - y_1 \leqslant 0\right\},$$

$$K_2 := \left\{(x^1, x^2, y_2) \in \{0,1\}^3 \mid -x^1 - x^2 - y_2 \leqslant -1\right\}.$$

Given these definitions, we can rewrite this problem in the form

$$\begin{aligned} \zeta^{SIP} = & \min_{x,y} x^1 + \frac{1}{2}(1000y_1) + \frac{1}{2}(1000y_2) \\ & s.t. \; (x^1, x^2, y_1) \in K_1 \\ & \quad (x^1, x^2, y_1) \in K_2 \end{aligned} \tag{2.29}$$

We can think of this as playing a (quite unfair) game as follows. You choose to pay a dollar ($x^1 = 1$) or not ($x^1 = 0$). In either case, choose heads ($x^2 = 1$) or tails ($x^2 = 0$), then flip a coin.

- *In scenario 1 the coin is tails: if you did not pay and you chose heads (so $x^1 - x^2 = 0 - 1 = -1$) then you must set $y_1 = 1$ and pay the penalty of 1000 dollars. If you paid the dollar and/or you chose tails (so $x^1 - x^2 \geqslant 0$) then you may set $y_1 = 0$ and pay nothing further.*

- *In scenario 2 the coin is heads: if you did not pay and you chose tails (so $x^1 + x^2 - 1 = 0 + 0 - 1 = -1$) then you must set $y_2 = 1$ and pay the penalty of 1000 dollars. If you paid the dollar and/or you chose tails (so $x^1 + x^2 - 1 \geqslant 0$) then you may set $y_2 = 0$ and pay nothing further.*

Obviously the best move is to pay the dollar (after which the heads/tails choice is irrelevant); this yields an objective value of 1 in all outcome scenarios and hence $\zeta^{SIP} = 1$.

We can rewrite in terms of non-anticipativity constraints as follows:

$$\begin{aligned}
\zeta^{SIP} = &\min_{x,\bar{x},y} \frac{1}{2}(x_1^1 + 1000y_1) + \frac{1}{2}(x_2^1 + 1000y_2) \\
&s.t.\ (x_1^1, x_1^2, y_1) \in K_1 \\
&\quad (x_1^1, x_1^2, y_1) \in K_2 \\
&\quad x_1^1 = \bar{x}^1, \quad x_2^1 = \bar{x}^1 \\
&\quad x_1^2 = \bar{x}^2, \quad x_2^2 = \bar{x}^2
\end{aligned} \tag{2.30}$$

The Lagrangian dual bound can be calculated via the dual problem definition as in (2.21) *or the convex hull-based primal representation as in* (2.22)*. Both will be demonstrated below.*

The Lagrangian dual problem corresponding to (2.30) *is*

$$\begin{aligned}
\zeta^{LD} = &\max_{\lambda} \min_{x,\bar{x},y} \frac{1}{2}(x_1^1 + 1000y_1) + \frac{1}{2}(x_2^1 + 1000y_2) + \\
&\frac{1}{2}\left[\lambda_1^1(\bar{x}^1 - x_1^1) + \lambda_2^1(\bar{x}^1 - x_2^1)\right] + \frac{1}{2}\left[\lambda_1^2(\bar{x}^2 - x_1^2) + \lambda_2^2(\bar{x}^2 - x_2^2)\right] \\
&s.t.\ (x_1^1, x_1^2, y_1) \in K_1 \\
&\quad (x_1^1, x_1^2, y_1) \in K_2
\end{aligned}$$

with the implied dual feasibility condition $\lambda_1^1 = -\lambda_2^1$ *and* $\lambda_1^2 = -\lambda_2^2$*. If we let* $\lambda^1 = \lambda_1^1 = -\lambda_2^1$ *and* $\lambda^2 = \lambda_1^2 = -\lambda_2^2$ *then* ζ^{LD} *can be rewritten in the form*

$$\begin{aligned}
\zeta^{LD} = &\max_{\lambda} \min_{x,\bar{x},y} \frac{1}{2}((1-\lambda^1)x_1^1 - \lambda^2 x_1^2 + 1000y_1) + \frac{1}{2}((1+\lambda^1)x_2^1 + \lambda^2 x_2^2 + 1000y_2) \\
&s.t.\ (x_1^1, x_1^2, y_1) \in K_1 \\
&\quad (x_1^1, x_1^2, y_1) \in K_2
\end{aligned}$$

By substituting all of the feasible points and eliminating those which are dominated by other feasible points, we can rewrite ζ^{LD} *as:*

$$\zeta^{LD} = \max_{\lambda} \left[\frac{1}{2}\min\left\{0, -\lambda^2 + 1000, 1-\lambda^1, 1-\lambda^1-\lambda^2\right\} + \frac{1}{2}\min\left\{1000, \lambda^2, 1+\lambda^1, 1+\lambda^1+\lambda^2\right\}\right]$$

A feasible solution to this maximisation problem is $\lambda^1 = 0$ *and* $\lambda^2 = 1$*, which results in*

$$\frac{1}{2}\min\left\{0, -\lambda^2 + 1000, 1-0, 1-0-1\right\} + \frac{1}{2}\min\left\{1000, 1, 1+0, 1+0+1\right\} = \frac{1}{2}.$$

Since the Lagrangian dual problem is a maximisation problem, $\frac{1}{2}$ *is a* **lower** *bound on the Lagrangian dual bound.*

Now consider the primal representation based on the convex hull of the feasible region. Since $(0,0,0)$ *and* $(1,1,0)$ *are in* K_1, $(\frac{1}{2},\frac{1}{2},0)$ *is in* $\operatorname{conv} K_1$. *Since* $(0,1,0)$ *and* $(1,0,0)$ *are in* K_2, $(\frac{1}{2},\frac{1}{2},0)$ *is in* $\operatorname{conv} K_2$.

Therefore $x_1^1 = x_2^1 = \bar{x}^1 = \frac{1}{2}$, $x_1^2 = x_2^2 = \bar{x}^2 = \frac{1}{2}$ *and* $y_1 = y_2 = 0$ *is a feasible solution of the convex hull representation of the Lagrangian dual of* ζ^{SIP}. *This solution has objective value* $\frac{1}{2}$. *Since the convex hull representation of the Lagrangian dual is a minimisation problem,* $\frac{1}{2}$ *is an* **upper** *bound on the Lagrangian dual bound.*

Since $\frac{1}{2}$ *is an upper and lower bound on the Lagrangian dual bound, it must in fact be the Lagrangian dual bound, resulting in a duality gap of* $1 - \frac{1}{2} = \frac{1}{2}$ *between the primal optimal solution and the dual bound.*

2.2 Current Literature in Lagrangian Duality

Overviews of Lagrangian duality may be found in [12, 100]. Some theory with particular relevance to the developments of later chapters is discussed in this section.

Algorithms for convex optimisation which employ Lagrangian duality include the subgradient method, cutting plane method, bundle methods, and alternating direction method of multipliers (among many variants). These methods and the associated literature will be discussed in Sections 2.3 and 2.5.

2.2.1 Exact Augmented Lagrangian Duality for Integer Variables

The augmented Lagrangian dual is not in general a strong dual for an integer problem, so it only provides a bound on the optimal solution rather than the exact optimal value. However, under some conditions on the problem, the augmenting function and the penalty parameter, the augmented Lagrangian dual is strong. Boland and Eberhard [18] showed that the augmented Lagrangian dual applied under the following conditions is a strong dual:

- The augmenting function ψ is of the form $\psi(\boldsymbol{u}) = \phi(\|\boldsymbol{u}\|)$, where ϕ is a convex, monotonically increasing function, $\phi(0) = 0$ and for some $\delta > 0$ the following conditions hold:

$$\liminf_{a\to+\infty} \frac{\phi(a)}{a} \geqslant \delta$$

and **diam** $\{a|\phi(a) \leqslant \delta\}$ approaches 0 from above as δ approaches 0 from above.

- One of the following conditions holds:
  - The feasible set of the LP relaxation of the problem does not contain a lineality space.
  - The constraints on the feasible set are rational and the norm used in the augmenting function is the infinity norm.
  - The convex hull of the feasible set is bounded.
- One of the following conditions holds:
  - The penalty parameter goes to infinity.
  - The feasible set is finite and discrete, and the penalty parameter is sufficiently large (but finite).

Feizollahi et al. [37] generalised this result under the condition that the penalty parameter goes to infinity, for general mixed-integer linear programs, and for augmenting functions ψ which satisfy the weaker conditions of being proper, non-negative, lower semi-continuous and level-bounded.

Feizollahi et al. also showed that in the alternate case where the penalty parameter is restricted to finite values, the above result holds even when the feasible set contains infinitely many feasible points. Furthermore, if the penalty function is proper, non-negative and bounded below by the infinity norm in a neighbourhood of the origin, the augmented Lagrangian dual is exact even if the feasible set is not discrete (some of the variables are continuous). In particular, these conditions on the augmenting function are satisfied by the use of any norm as an augmenting function.

2.2.2 Semi-Lagrangians

Instead of directly applying Lagrangian relaxation to an equality constraint in a mixed-integer linear program, in some cases it is preferable to reformulate the constraint into a different form first. Semi-Lagrangian relaxation, as proposed by Beltran et al. [9], includes a redundant inequality constraint $Qx \leqslant r$ to accompany the complicating equality constraint $Qx = r$. When Lagrangian relaxation is applied to the equality constraint, the inequality constraint remains as an explicit constraint on the problem. An equivalent approach is to reformulate the difficult constraint $Qx = r$ into two

sub-constraints $Qx \leqslant r$ and $Qx \geqslant r$, and then apply Lagrangian relaxation to only one of these two sub-constraints.

Under some conditions, semi-Lagrangian relaxation results in a strong dual even for integer programs, given that the problem has non-negative coefficients and the non-integrality constraints on the variables are polyhedral. However, this comes at the cost of retaining an inequality constraint in the problem to be solved. At worst the resulting problem will be no easier to solve than the original MIP.

To put the semi-Lagrangian approach to effective use, the structure of the problem and algorithm employed must allow the resulting primal integer programs (which include the scenario-linking inequalities) to be solved easily. This is accomplished by choosing values for the dual variables which allow the problems to be simplified. The method for choosing suitable dual variables varies from problem to problem. For example, consider the p-median problem with the following formulation:

$$\begin{aligned}
\zeta^* = &\min_{x,y} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \\
\text{s.t. } &\sum_{i=1}^{m} x_{ij} = 1, \forall j, \\
&\sum_{i=1}^{m} y_i = p, \\
&x_{ij} \leqslant y_i, \forall i, j, \\
&x_{ij}, y_i \in \{0, 1\}
\end{aligned}$$

Beltran et al. [9] gradually applied semi-Lagrangian relaxation to this problem using the following procedure (with appropriate termination conditions):

1. Apply a full Lagrangian relaxation to the first constraint ($\sum_{i=1}^{m} x_{ij} = 1, \forall j$) and second constraint ($\sum_{i=1}^{m} y_i = p$) and solve the associated dual problem

$$\begin{aligned}
(\boldsymbol{\lambda}', \boldsymbol{\mu}') = &\operatorname*{argmax}_{\boldsymbol{\lambda}, \boldsymbol{\mu}} \min_{x,y} \sum_{j=1}^{n} \left[\sum_{i=1}^{m} c_{ij} x_{ij} + \lambda_j \left(\sum_{i=1}^{m} x_{ij} - 1 \right) + \mu_j \left(\sum_{i=1}^{m} y_i - p \right) \right] \\
&\text{s.t. } x_{ij} \leqslant y_i, \forall i, j, \\
&\quad x_{ij}, y_i \in \{0, 1\}
\end{aligned} \tag{2.31}$$

   using a cutting plane method.

2. Add the constraint $\sum_{i=1}^{m} x_{ij} \leqslant 1 \ \forall j$ (i.e. use a semi-Lagrangian relaxation for the first constraint) to (2.31) and solve again, using the dual variables $(\boldsymbol{\lambda}', \boldsymbol{\mu}')$ obtained from the previous step as a starting point. This problem is solved with a general MIP solver (such as CPLEX). Denote the optimal choice of dual variables for this problem as $(\boldsymbol{\lambda}'', \boldsymbol{\mu}'')$.

3. Add the constraints $\sum_{i=1}^{m} x_{ij} \leqslant 1 \ \forall j$ and $\sum_{i=1}^{m} y_i \leqslant p$ to (2.31) (i.e. a full semi-Lagrangian relaxation) and solve again, using the dual variables $(\boldsymbol{\lambda}'', \boldsymbol{\mu}'')$ obtained from the previous step as a starting point.

The motivation for this approach as opposed to solving the full semi-Lagrangian dual problem directly is that the dual information obtained at each step makes the subsequent problems considerably easier. In particular, when solving the partial semi-Lagrangian dual problem, any variable x_{ij} whose cost c_{ij} exceeds the corresponding dual variable λ_j may be eliminated from consideration. The solution to the full semi-Lagrangian dual problem is typically close to that of the partial semi-Lagrangian problem, which allows a relatively easy search for the optimal solution.

Different problems require different approaches to choosing appropriate dual variables for semi-Lagrangian relaxation. For example, when solving Uncapacitated Facility Location problem instances it is instead ideal to keep small as many dual variables as possible, since this eliminates a large number of variables from the corresponding subproblem [10].

A further refinement of semi-Lagrangian duality is to replace the complicating inequality constraint with a surrogate constraint; that is, replace the set of equalities $Q_i x = R_i$ for $i \in I$ with a single inequality $\sum_{i \in I} Q_i x = \sum_{i \in I} R_i$. In combination with the (previously redundant) added inequalities, this replacement does not weaken the formulation, and when Lagrangian duality is applied to the single surrogate constraint only a single dual variable results. This approach is considered in [85, 64].

2.3 Convex Optimisation

2.3.1 Convexity Definitions

Definition 2.3 *A set $C \in \mathbb{R}^n$ is said to be a* ***convex set*** *if for all points x and y in C, and for all α in $[0, 1]$, the point $\alpha x + (1 - \alpha) y$ is also in C.*

In informal terms, this means that any straight line connecting two points in C is itself contained in C.

Definition 2.4 *The set of* ***extended real numbers*** $\mathbb{R} \cup \{+\infty, -\infty\}$ *is denoted* $\mathbb{R}_\infty$.

Definition 2.5 *The* ***epigraph*** *of a function* $f : X \to \mathbb{R}_\infty$*, denoted* $\operatorname{epi} f$*, is the set of points lying on or above the graph of the function:*

$$\operatorname{epi} f = \{(x, \mu) \mid x \in X, \mu \in \mathbb{R}_\infty, \mu \geqslant f(x)\}$$

A function is convex if and only if its epigraph is a convex set.

Definition 2.6 *A function on a convex set* $X \in \mathbb{R}^n$*,* $f : X \to \mathbb{R}_\infty$ *is said to be a* ***convex function*** *if for all points* x *and* y *in* X*, and for all* α *in* $[0, 1]$*, the following condition holds:*

$$f(\alpha x + (1 - \alpha)y) \leqslant \alpha f(x) + (1 - \alpha)f(y)$$

Equivalently a function is convex if and only if its epigraph is a convex set.

In informal terms, this means that any straight line connecting two points on the graph of f lies entirely above f.

A convex optimisation problem has the form

$$\zeta = \min_{x \in X} f(x) \tag{2.32}$$

where X is a convex set and f is a convex function. Unless otherwise stated we will assume that the feasible set X is closed and that ζ is bounded below (and therefore an optimal solution exists). The properties of convex sets and functions are used to design algorithms specifically for these problems.

2.3.2 Convex Analysis, Duality and Optimality Conditions

For an overview of convex analysis see e.g [97]. Some basic definitions are reproduced here.

Definition 2.7 *The* ***convex hull*** *of a set* $X \subset \mathbb{R}^n$*, denoted* $\operatorname{conv}(X)$*, is the set of all points which may be obtained as a convex combination of points in* X *i.e. if* x *is in* $\operatorname{conv}(X)$ *then there exist*

points $x_1, \ldots, x_n$ *in* X *and weights* $\lambda_1, \ldots, \lambda_n$ *satisfying* $\sum_{i=1}^{n} \lambda_i = 1$ *and* $\lambda_i \in [0,1] \quad \forall i \in (1, \ldots, n)$ *such that*

$$x = \sum_{i=1}^{n} \lambda_i x_i$$

This is also the smallest convex set containing X*.*

Definition 2.8 *The* ***effective domain*** *of a convex function* $f : X \to \mathbb{R}_\infty$ *is the set of all points* $x \in X$ *such that* $f(x) < +\infty$*.*

Definition 2.9 *A convex function* $f : X \to \mathbb{R}_\infty$ *is said to be* ***proper*** *if* $f(x) < +\infty$ *for some* $x \in X$ *(i.e. the effective domain of* f *is non-empty) and* $f(x) > -\infty$ *for all* $x \in X$*.*

Definition 2.10 *The* ***characteristic function*** *of a set* $X \subseteq \mathbb{R}^n$ *is denoted* $\delta_X : \mathbb{R}^n \to \mathbb{R}_\infty$*, where*

$$\delta_X(x) = \begin{cases} 0 & x \in X \\ +\infty & x \notin X \end{cases}$$

Definition 2.11 *Assume* V *is a vector space. The set of linear functionals on this vector space is said to be the corresponding* ***dual*** *(vector)* ***space****, and is denoted* V^**.*

The results presented in later chapters will frequently only consider the special case where the vector space V is a finite dimensional Euclidean space, meaning that $V^* = V$. In these cases the distinction between the vector space and its dual is unimportant. However, some definitions and cited results will use the more general notation.

Definition 2.12 *A function* $f : X \to \mathbb{R}_\infty$ *is said to be* ***lower semi-continuous*** *at* $x_0 \in X$ *if*

$$f(x_0) \leqslant \liminf_{x \to x_0} f(x)$$

Consider as an example the function $\delta_{\{0\}} : \mathbb{R} \to \mathbb{R}_\infty$ defined by

$$\delta_{\{0\}}(x) = \begin{cases} 0 & x = 0 \\ +\infty & x \neq 0. \end{cases}$$

This function is convex and lower semi-continuous but is not differentiable or continuous at 0. The existence of convex functions which are not differentiable means that the theory of ordinary derivatives and gradients is frequently not applicable in convex analysis. Instead, an analogous concept which leverages the convexity property is used.

Definition 2.13 *The* ***subgradient*** *of a convex function $f : X \to \mathbb{R}_\infty$ is denoted ∂f, and is defined as follows:*

$$\partial f(x_0) := \{x^* \mid x^* \in X^*, f(x) - f(x_0) \geqslant \langle x^*, (x - x_0)\rangle \; \forall x \in X\} \tag{2.33}$$

The condition $f(x) - f(x_0) \geqslant \langle x^*, (x - x_0)\rangle \; \forall x \in X$ is referred to as the ***subgradient inequality****.*

Note that there exist more general definitions of the subgradient which do not rely on convexity or the global behaviour of f, although they are generally equal to the convex subgradient when the latter exists. We will use the definition presented above unless the text specifically indicates otherwise.

If X is a finite dimensional real vector space, a consequence of the subgradient inequality is that the hyperplane with normal vector x^* which touches epi f at x_0 is a supporting hyperplane of epi f.

Definition 2.14 *An optimisation problem of the form*

$$\min_{x \in X} f(x) \tag{2.34}$$

is said to be convex if its objective function f is a convex function and its feasible region X is a convex set.

Remark 2.15 *Optimsation problems with integer variables are almost always non-convex, since their feasible regions are a discrete set of points or hyperplanes.*

The basic definition of the globally optimal solution to an optimisation problem (that no better feasible point exists) is difficult to evaluate directly for a given point. Based on the structure of convex optimisation problems we can define an equivalent **optimality criterion** which is more easily evaluated. This will be particularly useful when we define termination conditions for the algorithms presented in later chapters.

If a convex optimisation problem of the form (2.34) is unconstrained (i.e. $X = \mathbb{R}^n$) and smooth, its optimal points x_0 are those which satisfy $0 = \nabla f(x_0)$. Similarly, if the optimisation problem is unconstrained and non-smooth its optimal points x_0 satisfy $0 \in \partial f(x_0)$; to see this, substitute $x^* = 0$ into the subgradient inequality to obtain

$$f(x) - f(x^0) \geqslant 0 \quad \forall x \in X.$$

If the convex optimisation problem contains constraints a more sophisticated optimality criterion is required. In addition we will need some form of constraint qualification condition to exclude "unreasonable" constraint sets and objective functions.

Definition 2.16 *The **normal cone** to a convex set X at a point $\hat{x}$ is denoted $N_X(\hat{x})$ and is defined as follows:*

$$N_X(x) := \{x^* \mid \langle x - \hat{x}, x^* \rangle \leqslant 0 \quad \forall x \in X\}$$

Theorem 2.17 *[100, Theorem 3.34] Consider a convex optimisation problem*

$$\begin{aligned} \zeta^{IP} = & \min_x f(x) \\ & s.t.\ g_i(x) \leqslant 0 \quad \forall i = 1, \ldots, m \\ & h_j(x) = 0 \quad \forall j = 1, \ldots, p \\ & x \in X \end{aligned} \tag{2.35}$$

where f and every g_i are convex functions, every h_j is an affine function, and X is a convex and closed set in $\mathbb{R}^n$. Assume that the following qualification conditions are satisfied:

- *The **Slater condition** holds i.e. there exists a point x such that $g_i(x) = 0$ for all $i = 1, \ldots, m$ and $h_j(x) = 0$ for all $j = 1, \ldots, p$.*
- *The objective function f is continuous at some feasible point.*

Then, if $\hat{x}$ is a feasible point such that the conditions

$$0 \in \partial f(\hat{x}) + \sum_{i=1}^{m} \hat{\lambda}_i \partial g_i(\hat{x}) + \sum_{j=1}^{p} \hat{\mu}_j \nabla h_j(\hat{x}) + N_X(\hat{x}) \tag{2.36}$$

(where $N_X(\hat{x})$ is the normal cone of X at $\hat{x}$) and

$$\hat{\lambda}_i g_i(\hat{x}) = 0 \quad \forall i = 1, \ldots, m \tag{2.37}$$

are satisfied for some $\hat{\lambda} \in \mathbb{R}^m_+$ and $\hat{\mu} \in \mathbb{R}^p$, then $\hat{x}$ is an optimal solution of (2.35). Conversely, if $\hat{x}$ is an optimal solution of (2.35), the criteria (2.36) and (2.37) must be satisfied for some $\hat{\lambda} \in \mathbb{R}^m_+$ and $\hat{\mu} \in \mathbb{R}^p$.

The conditions given in (2.36) and (2.37) are a generalisation of the **Karush-Kuhn-Tucker (KKT) conditions** to non-smooth optimisation.

A variety of methods for solving convex optimisation problems are explored in the following sections.

2.3.3 Alternating Direction Method of Multipliers

The **Alternating Direction Method of Multipliers** (ADMM) is a primal-dual algorithm for solving convex optimisation problems [44, 48]. ADMM is based on the idea of iteratively solving an optimisation problem with one set of variables fixed, then another, with the end goal of obtaining the optimal decision for all variables.

ADMM proceeds on the assumption that the convex problem may be expressed in the form

$$\begin{aligned} \zeta^{IP} = & \min f(x) + g(y) \\ & \text{s.t. } Ax + By = c \\ & x \in X \\ & y \in Y \end{aligned}$$

where f and g are convex functions and X and Y are convex polyhedral sets.

The augmented Lagrangian with penalty parameter ρ corresponding to the relaxation of the constraint $Ax + By = c$ is:

$$L_\rho(x, y, \lambda) = f(x) + g(y) + \lambda^T(Ax + By - c) + \frac{\rho}{2} \|Ax + By - c\|_2^2$$

The ADMM algorithm is outlined below.

Initialise Initialise the decision variables x^0 and y^0. Set the dual variables λ^0 to zero. Set $k = 1$.

Step 1 Update x:

$$x^k \in \operatorname*{argmin}_{x \in X} L_\rho(x, y^{k-1}, \lambda^{k-1})$$

Step 2 Update y:

$$y^k \in \operatorname*{argmin}_{y \in Y} L_\rho(x^k, y, \lambda^{k-1})$$

Step 3 Update dual variables λ:

$$\lambda^k \leftarrow \lambda^{k-1} + \rho(Ax^k + By^k - c)$$

Step 4 Check for convergence; if the method has not yet converged, set $k = k + 1$ and return to Step 1.

The choice of penalty parameter ρ involves a tradeoff; typically when the penalty parameter is larger convergence is swifter (in terms of number of iterations) but each individual subproblem is more difficult to solve.

When applied to a stochastic integer program ADMM is equivalent to the Progressive Hedging algorithm of Rockafellar and Wets [98]. In this context, the minimisation in step 1 is separable into smaller subproblems, and that in step 2 has a closed-form representation, which aids in the practical application of the algorithm. Progressive Hedging is discussed further in Section 2.5.1.

The Alternating Direction Method of Multipliers is a special case of Douglas-Rachford splitting [43]. Eckstein and Bertsekas [32] demonstrated that Douglas-Rachford splitting is itself an application of the proximal point algorithm, and applied generalisations of the proximal point algorithm involving variable step lengths and approximate solving of subproblems to ADMM.

Another possible generalisation of ADMM is to vary the penalty parameter at each iteration and/or by the variable to which each penalty pertains [70]. Lenoir and Mahey [75] proposed a variety of methods for altering the penalty parameter at each iteration to improve the rate of convergence. Computational results indicated that these methods did not perform significantly better than choosing a "good" static penalty parameter, but they did remove the need to choose a good initial parameter.

Two theoretical and practical overviews of the Alternating Direction Method of Multipliers are given by Boyd et al. [21] and Eckstein and Yao [34].

2.3.4 Frank-Wolfe Method

Methods of feasible directions for solving convex optimisation problems approximate the feasible set by finding feasible points. Since the feasible set is convex, any convex combinations of these feasible points must also be feasible. A simple example of a feasible directions method is the Frank-Wolfe method [41] (also known as the conditional gradient method). The Frank-Wolfe method assumes that the objective function f is differentiable, so its gradient ∇f exists.

The Frank-Wolfe method always converges to the optimal solution of a convex optimisation problem which satisfies the conditions for its use. At any given iteration, $\hat{x}$ is the current candidate solution, and $\bar{x}$ may be chosen as an extreme point of the feasible set. (The feasible set is assumed to be bounded.)

Since the developments in this thesis make extensive use of the Frank-Wolfe method and its generalisations, they are reviewed in greater detail in Chapter 3.

2.3.5 Subgradient Method

Gradient projection methods for convex optimisation problems (of the form of Equation 2.32), with a continously differentiable objective function, have the following basic steps:

Step 1 Calculate the direction of steepest descent. Terminate if the current point is a stationary point.

Step 2 Take a step in the direction of steepest descent, and project the result onto the feasible set to obtain a feasible point.

Step 3 Take a step in the direction of this feasible point. Return to Step 1.

The length of each step is determined by an arbitrary parameter which typically decreases over time.

If the objective function is not continuously differentiable, it may not be possible to find a direction of steepest descent. In this case, instead of finding the gradient of the function at a point, we can instead take a subgradient of the function and take a step in that direction. A method of this form is called a **subgradient method**.

At any given iteration, the chosen subgradient may not be a direction of descent at all. However, convergence is guaranteed by the property that for sufficiently small step sizes, the distance from the optimal solution set must decrease at every step ([12], Proposition 6.3.1). (The idea of the proof is that the angle between any subgradient and any optimal point must be smaller than a right angle.)

Summaries of the properties of subgradient methods may be found in [12, 103].

2.3.6 Cutting Plane Method

Calculating a subgradient of a convex function f at a point x is analogous to finding a supporting hyperplane to the function which comes into contact with the graph of the convex function at x. A convex function may be approximated from below by finding a number of supporting hyperplanes

and finding the pointwise maximum of these hyperplanes. Cutting plane methods use this to generate successively closer approximations to the convex objective function, with the following basic steps:

Initialise Choose a starting point by some heuristic. Set $k = 1$.

Step 1 Find a subgradient at the current point, and the supporting hyperplane which corresponds to this subgradient. Add this hyperplane to the set of supporting hyperplanes.

Step 2 Choose a new point by solving the problem

$$\zeta = \min_{x \in X} F^k(x)$$

where F^k is an approximation of f at the current step k generated by taking the maximum of the supporting hyperplanes.

Step 3 Set $k = k + 1$ and return to Step 1.

If the objective function $f(x)$ is polyhedral then the cutting plane method converges to an optimal solution in a finite number of steps, since the approximation of the objective function becomes exact with a finite number of cutting planes. Otherwise, the cutting plane method is only guaranteed to converge in the limit as the number of iterations goes to infinity [12].

Cutting plane methods were first developed by Cheney and Goldstein [25] and by Kelley [66]. Summaries of the properties of cutting plane methods may be found in [12, 100]. The basic cutting plane method has largely been superseded by bundle-type methods, considered in the next section.

2.3.7 Bundle Method

Bundle methods are a general class which differ from the above methods in that they do not necessarily change the current point at every iteration; they build up information about the problem until a sufficiently descending step is found, and only then actually take that step and change the current point.

The term "bundle" refers to the bundle of information which is accumulated over time as the method runs. The cutting plane also stores a "bundle" of information in the form of the set of

cutting planes. However, bundle methods place more emphasis on curating the bundle, to limit its size and to maximise the useful information stored given that limitation.

The information used by subgradient, cutting plane and bundle methods can be represented as a set of triples (x_i, f_i, s_i) where x_i is a point in the decision space, f_i is the value of the objective function at x_i and s_i is a subgradient of f at x_i. In bundle methods, this set is curated by **selecting** which pieces of information to keep and which to discard, and by **compressing** multiple pieces of information into a smaller form (while retaining as much useful information as possible).

A bundle method has the following basic steps:

Initialise Choose a starting point x_1 by some heuristic, and find a subgradient at the current point. Add this information to the bundle. Set $k = 1$, and set x_1 as the current stability centre $\hat{x}$.

Step 1 Choose the best next point x_{k+1}, based on the current approximation of the objective function (which is derived from the bundle) and a stability term (to stop the algorithm from moving too far from $\hat{x}$, based on limited information).

Step 2 Perform an optimality test on x_{k+1}. If it is suffiently close to optimal, terminate.

Step 3 Find a subgradient at x_{k+1}. Add the information obtained thereby to the bundle.

Step 4 Perform a progress test to determine whether x_{k+1} should replace the current stability centre $\hat{x}$.

Step 5 Select which items in the bundle of information should be discarded and/or compressed.

Step 6 Set $k = k + 1$ and return to Step 1.

A specific implementation of the bundle method must define

- A means of approximating the objective based on the information bundle,
- A stability term for the problem,
- A progress test for updating the stability centre,
- A test for (sufficient) optimality, and

- An algorithm to determine which of the pieces of information in the bundle (subgradients at points) should be retained, compressed or discarded.

As an example, the proximal bundle method is a commonly used bundle method based on combining bundle ideas with the proximal point method. The proximal point method consists of iteratively finding the best point (with respect to the objective) which is in close proximity to the current point (which is enforced by adding a term $|x_k - x_{k+1}|^2$, using some norm $|\cdot|$, which penalises a choice of new point x_{i+1} if it is far away from the current point x_i).

Therefore, the proximal bundle method employs a proximal term of the form $|x_{k+1} - \hat{x}|^2$ to enforce proximity of new points x_{k+1} to the current stability centre $\hat{x}$. One possible implementation builds cutting planes (as in the cutting plane method) using the subgradients obatined in Step 4. In this case, the solution to the update problem in Step 2 has a closed-form solution. Furthermore, it is easy to determine which cutting planes are currently not affecting the selection of x_{k+1} and can be discarded [29].

The progress test considers the difference between the expected objective value at x_{k+1} based on the current approximation to the objective, and the actual objective value calculated at that point; if the approximation is good, it is "safe" to take a step. The optimality test also takes into consideration the distance in the decision space between $\hat{x}$ and x_{k+1}; if the approximation is good and the best point is close, the algorithm should be very close to the optimal point.

Summaries of the properties of subgradient methods may be found in e.g. [60, 29]. The proximal bundle method is considered in detail in e.g. [73, 69, 40].

2.4 SIP Reformulations and Benchmark Instances

2.4.1 Scenario Clustering

Instead of fully decomposing an SIP with n outcome scenarios into n separate subproblems, it is possible to partially decompose the problem by incorporating multiple scenarios (and the non-anticipativity constraints which link them) in a single subproblem. The resulting subproblems are more difficult, but the larger number of non-anticipativity constraints which are enforced explicitly can improve the convergence rate of the overall algorithm employed to solve the SIP.

An obvious problem to consider is how to choose which scenarios to cluster together. Crainic et

al. [27] experimented with clusters of similar scenarios, dissimilar scenarios, and scenarios chosen at random (with similarity measured by either properties of the constraints or distance of anticipative solutions corresponding to each scenario). They found that, when applying Progressive Hedging to network design problems, clustering similar scenarios resulted in the best performance, followed by clustering dissimilar scenarios. Either clustering strategy performed better than clustering scenarios at random, but any clustering method outperformed the no-clustering approach.

Scenario clustering has also been tested with encouraging computational results in combination with the subgradient method and cutting plane method [35] (in which scenarios were clustered with similar scenarios) and applied to multi-stage stochastic problems [36] (in which scenarios were clustered based on the stage in which they diverged).

2.4.2 Test Problems for Stochastic Programming

SIPLIB (A **S**tochastic **I**nteger **P**rogramming Test Problem **LIB**rary) [3] provides a variety of benchmark problems for stochastic integer programming. SIPLIB contains the following problem sets:

- **DCAP** - dynamic capacity acquisition and allocation under uncertainty, with mixed-integer first-stage and pure binary second-stage variables [4]
- **EXPUTIL** - expected utility knapsack problem, with pure binary variables
- **MPTSP** - multi-path travelling salesman problem, with pure binary first- and second-stage variables
- **PROBPORT** - chance constrained portfolio optimisation, with continuous first-stage and pure binary second-stage variables
- **SEMI** - SIP related to planning semiconductor tool purchases, with mixed-integer first-stage and continuous second-stage variables
- **SMKP** - stochastic multiple knapsack problem, with pure binary first- and second-stage variables
- **SIZES** - SIP related to product substitution applications, with mixed-integer first- and second-stage variables

- **SSLP** - SIP related to server location under uncertainty, with pure binary first-stage and mixed-binary second-stage variables [92]
- **VACCINE** - vaccine allocation problem, with continuous variables and employing joint chance constraints

Bodur et al. [15] presented two sets of benchmark stochastic integer programming problems. Both the first set of capacitated facility location problems (CAP) variables and second set of stochastic network interdiction problems (SNIP) have binary first-stage and continous second-stage variables.

For the purposes of benchmarking, stochastic integer programs are sometimes generated from problems with only continuous decision variables by arbitrarily limiting some variables to be integer-valued only.

Multi-stage extensions of some SIZES, DCAP and SEMI problems have been created by Zenarosa et al. [115], and may be found at [114].

In the computational experiments performed throughout this work, we will generally refer to the CAP, DCAP and SSLP problem sets.

2.5 Stochastic Programming Algorithms

2.5.1 Progressive Hedging

The Progressive Hedging (PH) algorithm for continuous stochastic problems was originally formalised by Rockafellar and Wets [98]. Both the current name and operation of the algorithm are based on what Rockafellar and Wets term the **principle of progressive hedging** in optimisation under uncertainty. The essence of this principle is that by finding a solution to a stochastic problem without enforcing non-anticipativity, and then gradually demanding greater conformity to the non-anticipativity constraints, a good solution will be found.

The Progressive Hedging algorithm is outlined below. A more formal definition of the algorithm is deferred until the introduction of Chapter 4.

Initialise Determine the starting values for the primal variables x_s^0, y_s^0 and dual variables λ_s^0 for all $s \in S$. Set the consensus variables $\bar{x}^0 = \sum_{s \in S} p_s x_s^0$. Choose a value for the penalty parameter

ρ. Set $k = 1$.

The starting values for the primal variables x_s and y_s are obtained by finding the optimal anticipative decisions for each scenario. It suffices to initially set the dual variables to zero.

Step 1 Update x and y for each scenario:

$$(x_s^k, y_s^k) \in \operatorname*{argmin}_{(x,y)\in X_s} L_\rho(x, y, \bar{x}^{k-1}, \lambda^{k-1}) \quad \forall s \in S$$

Note that since $\bar{x}^{k-1}$ is constant the parts of the augmented Lagrangian which were non-separable are now constant and can be ignored. Therefore, this minimisation is separable into subproblems, each of which contains the variables and constraints from one scenario. This greatly decreases the computational difficulty of this step.

Step 2 Update $\bar{x}$:

$$\bar{x}^k \in \operatorname*{argmin}_{\bar{x}} L_\rho(x^k, y^k, \bar{x}, \lambda^{k-1})$$

Step 3 Update dual variables λ for each scenario:

$$\lambda_s^k = \lambda_s^{k-1} + p_s \rho^k (x_s^k - \bar{x}) \quad \forall s \in S$$

Step 4 Check for convergence; if the method has not yet convereged, set $k = k + 1$ and return to Step 1.

Progressive Hedging is guaranteed to converge to the optimal primal solution when applied to a convex stochastic problem even if the subproblems used to update the primal variables are not solved exactly [98]. In particular it is possible to use a computationally simple heuristic to obtain an approximate solution to the subproblems quickly, rather than use a slower method to obtain an unnecessarily precise solution. This idea has been applied to control problems related to fishery management [57]. In practice, finding a more accurate solution to the subproblems can actually slow the convergence of the Progressive Hedging algorithm, not only in terms of computation time but also with respect to the number of iterations [65].

Since the feasible region of a SIP is non-convex by virtue of the integrality restriction, the above convergence result does not apply in the SIP context. In an intuitive sense, taking the probability-weighted average of the first-stage decisions is problematic for an integer problem. If any first-stage

variables are integer-valued, their average is unlikely to be an integer itself. Even if all of the first-stage variables are integer-valued, the integer-valued second-stage variables cause the second-stage problem (2.3) to be non-convex, which means that the objective function of ζ^{SIP} (as defined in (2.2)) is non-convex.

Progressive Hedging may be used to solve the continuous relaxation of an SIP with relative ease, since this is simply an ordinary continuous stochastic program. An intuitive approach to solving the SIP (and integer programs in general) is to first solve their linear relaxation and then examine feasible integer points with values near to the fractional optimal point of the relaxation. The notion of finding a "nearby point" is not always practical, especially in the case of a binary program, and in general there is no guarantee of finding an optimal or even feasible integer point in this way. Nevertheless, this approach has been applied to some problems with success [77].

The first application of Progressive Hedging to general multi-stage stochastic integer programming was made by Løkketangen and Woodruff [78], using tabu search to solve the integer programming subproblems.

Since the Progressive Hedging algorithm is not guaranteed to converge to the optimal primal solution when applied to integer problems, solving the subproblems approximately is attractive in this context also. Progressive Hedging has been used in this way to solve stochastic lot-sizing problems, solving the scenario subproblems with a dynamic programming algorithm only guaranteed to yield an optimal solution in the initialisation step [54].

Penalty Parameters

The value of the penalty parameter is especially important when applying Progressive Hedging to integer stochastic problems, since for these problems the parameter determines not only the rate of convergence and difficulty of the subproblems (as in continuous stochastic programming), but also the quality of the solution found by the algorithm [88, 77].

A downside of the conventional approach of choosing a single penalty parameter which applies to every non-anticipative constraint is that the decision variables may not all have the same scale. For example, if a problem contains a binary variable and another variable with a large range of possible values, failure to achieve consensus with respect to the binary variable will be penalised to a lesser degree, even if its value is a more important component of the policy. This downside can

be overcome by choosing separate penalty parameters for the non-anticipativity constraints which pertain to each decision variable.

Watson and Woodruff [112] proposed the rule

$$\rho(i) = \frac{c_i}{\max_{s \in S}(x_s^0) - \min_{s \in S}(x_s^0) + 1}$$

for choosing a penalty corresponding to each integer first-stage variable x_i, based on the anticipative first-stage decisions x_s^0 obtained for each scenario in the initialisation step. They proposed the similar rule

$$\rho(i) = \frac{c_i}{\max(\sum_{s \in S} p_s \left| x_s^0 - \bar{x}^0 \right|, 1)}$$

for continuous variables.

In addition, the results covered in the previous section regarding penalty parameters for ADMM are also applicable to Progressive Hedging.

Variable Fixing

Once the Progressive Hedging algorithm has obtained consensus as to the optimal value of a given decision variable, that variable may be "fixed" to that value to decrease the computational difficulty of the subproblems to be solved.

The difficult part of this procedure is to find a reliable heuristic to determine that the optimal value of the variable has in fact been reached. If the heuristic is too conservative, then the improvement in performance will be small. If the heuristic is too aggressive it may lock variables prematurely to a non-optimal value.

One possible approach is to wait until the value of all integer variables has achieved consensus, then fix the integer values and determine the value of the continuous variables by directly solving the IP-equivalent problem [78, 77]. Since the resulting problem is a continuous LP it can be computationally tractable despite its large size.

Alternatively, variables which have not yet achieved consensus may be "slammed" by fixing them to a consensus value which seems reasonable based on the present value of the variables. This method is easiest to apply to problems where all of the bounds on the variables are either from above or below, since in this case it is possible to guarantee admissibility of the slammed variable by taking the most pessimistic value for the variable [112].

Obtaining Lower Bounds

A downside of Progressive Hedging as applied to SIPs is that despite being a Lagrangian dual-based approach it does not directly yield lower bounds. Gade et al. [45] showed that the dual variables obtained at each step may be used to compute a lower bound for the SIP. The lower bounds are computed by solving the following mixed integer linear program for each scenario s:

$$\begin{aligned} D_s(\lambda) = &\min_{x_s, y_s} p_s(c^T x_s + d_s^T y_s) + \lambda_s^T x_s \\ &\text{s.t. } (x_s, y_s) \in K_s \end{aligned}$$

The sum $\sum_{s \in S} D_s(\lambda)$ is a lower bound for the SIP. This mixed-integer linear program is of comparable complexity to the mixed-integer quadratic program solved for each scenario in each Progressive Hedging step, and requires a similar amount of computational effort to solve.

The lower bounds obtained by this method, using the dual variables from successive steps of Progressive Hedging, are not guaranteed to converge to the optimal Lagrangian dual value ζ^{LD} when Progressive Hedging is applied to an integer problem. However, computational experiments show that in some cases tight bounds can be obtained by this method [45]. In particular, when the penalty parameter is chosen to be small the quality of the lower bound tends to improve. Unfortunately, this low penalty parameter also causes the method to converge very slowly. The developments in Chapters 4 and 5 of this work address this weakness of the Progressive Hedging algorithm.

2.5.2 Dual Decomposition

The Dual Decomposition method for stochastic integer programs was proposed by Carøe and Schultz [22]. It is based on a branch-and-bound framework, using the (non-augmented) Lagrangian dual to obtain dual bounds (lower bounds for a minimisation problem). Each node of the branching tree corresponds to an IP. Assuming that we are solving a minimisation problem, the method proceeds as follows:

Initialise Initialise the set of problems $\mathcal{P}$ by adding the original IP equivalent problem (Equation 2.4).

Step 1 Choose (and remove) a problem P from $\mathcal{P}$. If $\mathcal{P}$ is empty then the best known solution $(\hat{z}, \hat{x})$ (if any) is optimal; terminate.

Step 2 *(Compute lower bound)* Solve the corresponding Lagrangian dual problem to P. If P is infeasible return to Step 1.

If the optimal objective value of the Lagrangian dual problem is greater than the best known solution's objective value $\hat{z}$, return to Step 1; the assumptions made by the bounds on this problem cannot result in a better solution than what we already have.

Step 3 *(Compute upper bound)* Otherwise, consider the solution found for the Lagrangian dual problem.

- If the first-stage decisions $x_1, ..., x_{|S|}$ across all scenarios are identical, then compute the second-stage decisions and objective value which corresponds to this first-stage decision. If the objective value is smaller than $\hat{z}$ (hence an improvement), replace $(\hat{z}, \hat{x})$ with the objective value and first-stage decision obtained here. Remove any problems in $\mathcal{P}$ with known lower bounds which are worse (greater) than $\hat{z}$ and return to Step 1.
- If the first-stage decisions $x_1, ..., x_{|S|}$ across all scenarios are not identical, then compute their average and round the integer variables by some heuristic to obtain an integer decision $\bar{x}$. If this rounded decision is feasible then determine whether it is an improvement on our existing best known solution as above, and remove any problems from $\mathcal{P}$ which are now known to result in suboptimal solutions. In any case proceed to Step 4.

Step 4 *(Branching step)* Select a first-stage variable x^i which did not acheive consensus across all scenarios. (Its average value is $\bar{x}^i$.) Add two new problems to $\mathcal{P}$ by taking the current P and adding one of two constraints to it:

- If x^i is an integer variable, add the constraints $x^i_s \leqslant \lfloor \bar{x}^i \rfloor \ \forall s \in S$ to one problem and $x^i_s \geqslant \lfloor \bar{x}_i \rfloor + 1 \ \forall s \in S$ to the other.
- If x^i is a continuous variable, add the constraints $x^i_s \leqslant \bar{x}^i - \varepsilon \ \forall s \in S$ to one problem and $x^i_s \geqslant \bar{x}^i + \varepsilon \ \forall s \in S$ to the other, where ε is a small constant to ensure disjoint subdomains.

The optimal value of both of these problems is bounded below by the lower bound on P, since they are a strictly more constrained version of P. Once this is done, return to Step 1.

The idea behind the Dual Decomposition algorithm is to force consensus of first-stage decision variables which are not reaching consensus across all variables by adding constraints to the problem which force the variables in one direction or another from their "natural" average. The algorithm can determine which constraints might result in optimal solutions (and therefore should be retained), and which constraints might result in suboptimal solutions (and therefore should be removed from consideration), by computing bounds on the resulting problems.

As it uses the non-augmented Lagrangian dual, the calculation of the Lagrangian dual and thus the lower bound calculation may be made in parallel easily. Unlike Progressive Hedging, the Dual Decomposition method is also guaranteed to find an optimal decision to an SIP; eventually the constraints added in each branching step will constrain the decision variables to their optimal values. However, Dual Decomposition yields much slower initial improvement in (lower and upper) bounds than Progressive Hedging.

The computationally-difficult part of the Dual Decomposition algorithm is solving the Lagrangian dual problem in Step 3. The original implementation by Carøe and Schultz [22] used a proximal bundle method [68] to solve the dual problem. The proximal bundle method may partially be used in parallel in that the subproblems may be solved separately by scenario; if a non-anticipativity constraint linking the first-stage variables to a consensus variable $\bar{x}$ is used, the master problem of the algorithm may also be solved in parallel [79]. Guo et al. [50] employed Progressive Hedging to find reasonable lower bounds quickly (which in some cases allows a given branch to be ruled out immediately) and to find a good starting point for the proximal bundle method.

Other methods of applying branching methods to stochastic integer programming are also possible. Lulli and Sen [81] applied a branch-and-price method to SIPs, using a column generation algorithm to solve the primal problem directly and obtain bounds, rather than using the Lagrangian dual. Though they appear different the two approaches to solving the problem and obtaining bounds are closely related [79]. Another branch-and-bound algorithm for problems with integer second stage was proposed by Ahmed et al. [4].

2.5.3 Diagonal Quadratic Approximation

The Diagonal Quadratic Approximation (DQA) method for multistage stochastic (non-integer) programs was first proposed by Mulvey and Ruszczyński [86, 87]. The principle of the DQA method is similar to Progressive Hedging in that the augmented Lagrangian is employed to separate the SIP into scenario-wise subproblems. The primary distinguishing feature of DQA is that the augmented Lagrangian is approximated by a simpler, separable function.

Consider the quadratic term of the augmented Lagrangian:

$$\frac{\rho}{2} \|x_i - x_j\|_2^2 \tag{2.38}$$

This term considers cross-products of the form $x_i^T x_j$, which combines two different variables x_i and x_j in a non-separable quadratic term. This cross-product term may be approximated around $\tilde{x}_i$ and $\tilde{x}_j$ as follows:

$$x_i^T x_j \approx x_i^T \tilde{x}_j + \tilde{x}_i^T x_j - \tilde{x}_i^T \tilde{x}_j \tag{2.39}$$

Since $\tilde{x}_i$ and $\tilde{x}_j$ are constants this approximation is linear and separable in x. The augmented Lagrangian also contains terms of the form $(Q_i x_i)^T Q_i x_i$; these terms are still quadratic but their sum is separable. This principle can be used to approximate the augmented Lagrangian in the context of other applications besides stochastic optimisation.

When this approximation is employed, the augmented Lagrangian for a stochastic program becomes separable by scenario. Therefore, the Progressive Hedging-type approach of using a non-anticipativity variable $\bar{x}$ and alternating between minimisation over the decision variables and non-anticipativity variables in an iterataive manner is unnecessary, and so sequential non-anticipativity constraints (i.e. $x_1 = x_2$, $x_2 = x_3$, etc.) are used instead. However, since the accuracy of the approximation (2.39) is dependent on the proximity of x to a chosen fixed point $\tilde{x}$, an iterative series of minimisations bringing $\tilde{x}$ closer to the optimal decision is still necessary.

To accomodate a multistage problem, define $x_i(t)$ as the decision variables corresponding to outcome scenario i in stage t, and define $\nu(i,t)$ as the outcome scenario which scenario i is linked to by the non-anticipativity constraint in stage t. Define $\tilde{L}^\rho(\boldsymbol{x}, \tilde{\boldsymbol{x}}, \boldsymbol{\lambda})$ as the approximation of the augmented Lagrangian. Given an initial state $(\boldsymbol{x}^0, \boldsymbol{\lambda}^0)$ for the primal and dual variables, a penalty parameter ρ, and an update parameter $\alpha \in (0, 1]$, the $\tilde{x}$ updates of the DQA method are accomplished as follows:

Initialise Set $k = 1$ and $\tilde{\boldsymbol{x}}^k = \boldsymbol{x}^0$.

Step 1 Minimise $\tilde{L}^\rho(\boldsymbol{x}, \tilde{\boldsymbol{x}}^k, \boldsymbol{\lambda})$ over $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ with respect to feasibility constraints on $\boldsymbol{x}$, to obtain a new feasible decision $\boldsymbol{x}^k$. Note that this can be solved independently for each scenario.

Step 2 If $\boldsymbol{x}^k$ is sufficiently close to $\tilde{\boldsymbol{x}}^k$, terminate.

Step 3 Update $\tilde{\boldsymbol{x}}^k$:

$$\tilde{\boldsymbol{x}}^{k+1} = \tilde{\boldsymbol{x}}^k + \alpha(\boldsymbol{x}^k - \tilde{\boldsymbol{x}}^k)$$

Step 4 Set $k = k + 1$ and return to Step 1.

Step 1 is solved using a primal-dual barrier method. A more sophisticated implementation of the DQA method also proposed in [87] integrates the consensus update in Step 3 with the barrier method; in effect the consensus update is not forced to wait until near-optimality of $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ is achieved in each iteration.

Like Progressive Hedging, the DQA method can be applied in parallel, with each of a group of computational nodes being responsible for its own scenarios and their associated subproblems. Since the non-anticipativity constraints link scenarios together as pairs instead of linking all scenarios to a single consensus decision, each computational node need only communicate with the subset of nodes with which its scenarios are paired. Scenario clustering may be used to combine all of the scenario-wise subproblems assigned to a node into a single larger subproblem.

Chapter 3

The Frank-Wolfe Method and Generalisations

3.1 Introduction

3.1.1 The Frank-Wolfe Method

This chapter introduces the Frank-Wolfe method and its more sophisticated generalisations, and develops theory related to these algorithms for reference in subsequent chapters. Section 3.2 demonstrates how the Frank-Wolfe method and its variants can be applied to solving the convex hull relaxation of an integer program. Section 3.3 explores a generalisation of the Frank-Wolfe method to non-smooth optimisation.

The Frank-Wolfe method (sometimes referred to as the conditional gradient method) was initially proposed by Frank and Wolfe in [41] for quadratic programming problems, and was generalised by Holloway [61] for general convex programming problems.

Given a problem of the form

$$\zeta^{CP} = \min_{x} \{f(x) \mid x \in X\}, \tag{3.1}$$

where f is a convex, continuous and differentiable function whose gradient ∇f is known, and the feasible set X is closed and convex, the Frank-Wolfe method consists of the following steps:

Initialise Find a feasible solution $\hat{x}^0 \in X$ for Equation 3.1. Set $k = 1$.

Step 1 Set $\xi^k \in \arg\min_{x \in X} \left\{\nabla f(\hat{x}^{k-1})(x - \hat{x}^{k-1})\right\}$. If $\nabla f(\hat{x}^{k-1})(\xi^k - \hat{x}^{k-1}) \geqslant 0$, the algorithm terminates; we cannot find a better point than $\hat{x}^{k-1}$.

Step 2 Set $t^k \in \arg\min_{0 \leqslant \tau \leqslant 1} \left\{ f((1-\tau)\hat{x}^{k-1} + \tau\xi^k) \right\}$.

Step 3 Set $\hat{x}^k = (1-t^k)\hat{x}^k + t^k\xi^k$.

Step 4 Set $k = k+1$ and return to Step 1.

The termination condition in Step 1 may be informally interpreted as "there exists no feasible point in a direction of descent from the current point". Formally, by reference to the Karush-Kuhn-Tucker conditions in Theorem 2.17, $\hat{x}^{k-1}$ is an optimal solution of (3.1) if and only if

$$-\nabla f(\hat{x}^{k-1}) \in N_X(\hat{x}^{k-1}) \tag{3.2}$$

i.e. $\langle x - \hat{x}^{k-1}, -\nabla f(\hat{x}) \rangle \leqslant 0$ for all $x \in X$.

Furthermore, since f is a convex function, the hyperplane with gradient ∇f which intersects the graph of f at $\hat{x}^{k-1}$ minorizes f; therefore,

$$f(\hat{x}^{k-1}) + \nabla f(\hat{x}^{k-1})(\xi^k - \hat{x}^{k-1})$$

(i.e. the minimum of this hyperplane over X) is a lower bound on ζ^{CP} for all k.

The Frank-Wolfe method has a worst-case convergence rate of $O(1/k)$ (e.g. [42]).

To utilise the Frank-Wolfe method we need to be able to solve the minimisation problems in Steps 1 and 2. Since $\nabla f(\hat{x})(x - \hat{x})$ is affine with respect to x for a fixed $\hat{x}$, and X is convex, the step 1 update is typically not very difficult. However, $f((1-\tau)\hat{x} + \tau\xi)$ is merely convex with respect to τ for fixed $\hat{x}$ and ξ (it need not be affine or smooth), so even though its feasible set $[0,1]$ is very simple in structure the Step 2 update may not be easy to solve exactly, depending on the structure of f.

In early iterations of the algorithm our t^k updates are based on gradient information at points which are not necessarily close to optimal. This gradient information is therefore only an approximate guide to the location of the optimal point; therefore, it is unsurprising that the line search in Step 2 may be performed approximately as well without compromising the convergence properties of the Frank-Wolfe method. The rules used for these approximations generally make use of the gradient information at $\hat{x}^{k-1}$ and global properties of the objective function. The approximation schemes used in ordinary gradient descent algorithms are generally applicable to the Frank-Wolfe method as well.

An example of such an approximation scheme is the Armijo rule [6]. The Armijo rule chooses a step length based on the accuracy of the gradient information. If the gradient information remains reliable over a long step length then a long step will be chosen. Conversely, if the gradient information becomes inaccurate over a long step length then a shorter step will be chosen. This process is formalised in the context of the Frank-Wolfe method as follows. At step 2 in iteration k of the Frank-Wolfe method we wish to choose a step length t^k from $\hat{x}^{k-1}$ in the direction $d^k = \xi^k - \hat{x}^{k-1}$ based on the gradient $\nabla f(\hat{x}^{k-1})$. Replace the minimisation over τ with the following procedure:

Initialise Set an initial step length $0 < s \leqslant 1$, a step-size multiplier $0 < \beta < 1$, and a parameter $0 < \gamma < 1$ which determines the required accuracy of the gradient projection. Set $m = 0$.

Step 1 Set $\tau = \beta^m s$ (this is the trial step length).

Step 2 If $f(\hat{x}^{k-1}) - f(\hat{x}^{k-1} + \tau d^k) \geqslant -\gamma\tau\langle\nabla f(\hat{x}^{k-1}), d^k\rangle$ then set $t^k = \tau$ and terminate.

Step 3 Set $m = m + 1$ and return to Step 1.

In fact, it is not necessary to use any information about the objective function or the progress of the algorithm when choosing t^k. For example, if we skip step 2 entirely and initialise the step sizes using the rule

$$t^k = \frac{2}{k+2}$$

for all k, the Frank-Wolfe method converges with the same worst-case rate of $O(1/k)$ [42].

3.1.2 Simplicial Decomposition Method

A limitation of the original Frank-Wolfe method is that it retains minimal information from one iteration to the next. In particular, the direction of the $\hat{x}^k$ step is limited to the direction of the most recently found feasible point ξ^k. A more sophisticated algorithm which makes better use of previous calculations is the simplicial decomposition method (SDM) as given in [61, 110].

SDM applied to (3.1) consists of the following steps:

Initialise Find a feasible solution $\hat{x}^0 \in X$. Let $\xi^0 = \hat{x}^0$. Set $k = 1$.

Step 1 Set $\xi^k \in \arg\min_{x \in X} \left\{\nabla f(\hat{x}^{k-1})(x - \hat{x}^{k-1})\right\}$. If $\nabla f(\hat{x}^{k-1})(\xi^k - \hat{x}^{k-1}) \geqslant 0$, the algorithm terminates; we cannot find a better point than $\hat{x}^{k-1}$.

Step 2 Set $(t_0^k, t_1^k, \ldots, t_k^k) \in \arg\min_{\tau_0,\tau_1,\ldots,\tau_k} \left\{ f(\tau_0^k \xi^0 + \tau_1^k \xi^1 + \cdots + \tau_k^k \xi^k) \mid \tau_0^k + \tau_1^k + \cdots + \tau_k^k = 1 \right\}$.

Step 3 Set $\hat{x}^k = t_0^k \xi^0 + t_1^k \xi^1 + \cdots + t_k^k \xi^k$.

Step 4 Set $k = k + 1$ and return to Step 1.

A generalisation of the Frank-Wolfe method is to retain the feasible points $\bar{x}$ from each iteration, and in step 3 find $\hat{x}$ by searching over the convex hull of all (or a subset of) the previously generated extreme points together with the previous $\hat{x}$, rather than only the most recent extreme point [61].

The major practical disadvantage of SDM, as opposed to the basic Frank-Wolfe method, is the increased complexity of the minimisation problem in Step 2. Instead of minimising a potentially complicated convex function over a single variable, SDM must minimise over a larger number of variables, and the number of these variables grows larger with more iterations. The restricted simplicial decomposition method is a more sophisticated version of SDM which retains only the "important" points from previous iterations, quantified by their weights $(t_0^k, t_1^k, \ldots, t_k^k)$ in the convex combination used to generate the next candidate solution $\hat{x}^k$ [56, 109]. Once the size of the set of retained extreme points reaches some chosen limit, any further extreme points replace the currently least important one. This restricts the subproblem in Step 3 of the algorithm to a manageable size.

As with the Frank-Wolfe method, the convergence properties of SDM may be preserved (or partially preserved) even when the $(t_0^k, t_1^k, \ldots, t_k^k)$ update is not calculated exactly. In fact, if we set

$$t_i^k = \frac{1}{k}$$

for all i and k, SDM converges with rate $O(\ln(k)/k)$ [42].

3.2 Solving over the Convex Hull of Integer Programs

The main application of Frank-Wolfe type methods in this thesis is to solve problems of the form

$$\min_x \{ f(x) \mid x \in \operatorname{conv}(\mathfrak{X}) \} \tag{3.3}$$

where f is a convex, continuous and differentiable function, and $\mathfrak{X}$ is of the form

$$\mathfrak{X} := \left\{ x \in \mathbb{R}^{n-m} \times \mathbb{Z}^m \mid Ax \leqslant b \right\} \tag{3.4}$$

where A and b are defined such that $\mathfrak{X}$ and hence $\operatorname{conv}(\mathfrak{X})$ are bounded sets.

Problems of the form (3.3) arise naturally from applying the primal characterisation (2.9) of the Lagrangian dual problem to integer programs. The primary difficulty in solving (3.3) is that while the feasible set $\operatorname{conv}(\mathfrak{X})$ is convex and polyhedral, finding the linear inequalities which explicitly represent $\operatorname{conv}(\mathfrak{X})$ is a difficult task. The naive approach of enumerating the extremal points of $\mathfrak{X}$ is impractical since in general the number of such points is very large. In this section we will demonstrate that the Frank-Wolfe method can be applied to this problem with good results; in effect the Frank-Wolfe method enumerates only a few of the extremal points, meaning that it is computationally practical, but optimality of the final solution is still guaranteed.

The key insight which makes the Frank-Wolfe class of methods well suited to this class of problems is that to find the minimum of $f(x)$ over $\operatorname{conv}(\mathfrak{X})$ it is not necessary to construct a complete explicit representation of $\operatorname{conv}(\mathfrak{X})$. Finding some extremal points of $\mathfrak{X}$ which are close to the optimal point is sufficient, and the required number of points is typically manageable.

This application also has good theoretical convergence characteristics; as we will see, the simplicial decomposition method in particular is guaranteed to find the optimal solution in finite time when applied to (3.3). The underlying principle behind this guarantee is that (in the worst-case scenario) SDM is guaranteed to construct a complete explicit representation of $\operatorname{conv}(\mathfrak{X})$ eventually. In practice the method generally terminates well before this point.

SDM applied to (3.3) consists of the following steps:

Initialise Find a feasible solution $\hat{x}^0 \in \operatorname{conv}(\mathfrak{X})$. Let $\xi^0 = \hat{x}^0$. Set $k = 1$.

Step 1 Set $\xi^k \in \arg\min_{x \in \mathfrak{X}} \left\{ \nabla f(\hat{x}^{k-1})(x - \hat{x}^{k-1}) \right\}$. If $\nabla f(\hat{x}^{k-1})(\xi^k - \hat{x}^{k-1}) \geqslant 0$, the algorithm terminates; we cannot find a better point than $\hat{x}^{k-1}$.

Step 2 Set $(t_0^k, t_1^k, \ldots, t_k^k) \in \arg\min_{\tau_0, \tau_1, \ldots, \tau_k} \left\{ f(\tau_0^k \xi^0 + \tau_1^k \xi^1 + \cdots + \tau_k^k \xi^k) \mid \tau_0^k + \tau_1^k + \cdots + \tau_k^k = 1 \right\}$.

Step 3 Set $\hat{x}^k = t_0^k \xi^0 + t_1^k \xi^1 + \cdots + t_k^k \xi^k$.

Step 4 Set $k = k + 1$ and return to Step 1.

In Step 1 we substitute a minimisation over $\mathfrak{X}$ for the original minimisation over $\operatorname{conv}(\mathfrak{X})$. This is permissible since the set of extremal points of $\operatorname{conv}(\mathfrak{X})$ in the direction $-\nabla f(\hat{x}^{k-1})$ must always

include a point in $\mathfrak{X}$, by the definition of the convex hull. Since we have an explicit representation for $\mathfrak{X}$, Step 1 can be evaluated using a MIP solver.

For the sake of convenience, in the following proofs we will use the abbreviation

$$V^k = \left\{\xi^0, \xi^1, \ldots, \xi^k\right\}.$$

Using this notation we can combine Steps 2 and 3 to obtain

$$\hat{x}^k \in \min_x \left\{f(x) \mid x \in \operatorname{conv}(V^k)\right\}.$$

In the case where $\mathfrak{X}$ contains only integer variables (i.e. $n = m$) the proof of finite convergence is straightforward since $\mathfrak{X}$ is a finite set. As such we will consider this case first and then generalise to the mixed-integer case.

Lemma 3.1 *At each iteration k of the simplicical decomposition method applied to* (3.3)*, exactly one of the following is true:*

- ξ^k *is not in* $\operatorname{conv}(V^{k-1})$ *and hence* $\operatorname{conv}(V^{k-1}) \subsetneq \operatorname{conv}(V^k)$.
- $\hat{x}^{k-1}$ *is optimal with respect to the original problem* (3.3).

Proof. Since $\hat{x}^{k-1} \in \min_x \left\{f(x) \mid x \in \operatorname{conv}(V^{k-1})\right\}$, and f is a convex function, the gradient $\nabla f(\hat{x}^{k-1})$ at this point satisfies the condition $\langle \nabla f(\hat{x}^{k-1}), x - \hat{x}^{k-1}\rangle \geqslant 0 \ \forall\ x \in \operatorname{conv}(V^{k-1})$ (this is a necessary condition for optimality). Therefore:

- If $\langle \nabla f(\hat{x}^{k-1}), \xi^k - \hat{x}^{k-1}\rangle < 0$ then ξ^{h+1} is not in $\operatorname{conv}(V^{k-1})$ and hence $\operatorname{conv}(V^{k-1}) \subsetneq \operatorname{conv}(V^k)$.
- If $\langle \nabla f(\hat{x}^{k-1}), \xi^k - \hat{x}^{k-1}\rangle \geqslant 0$ then

  $$\langle \nabla f(\hat{x}^{k-1}), \xi^k - \hat{x}^{k-1}\rangle \geqslant 0 \quad \forall\, \xi \in \mathfrak{X}$$

  and hence

  $$\langle \nabla f(\hat{x}^{k-1}), \xi^k - \hat{x}^{k-1}\rangle \geqslant 0 \quad \forall\, \xi \in \operatorname{conv}(\mathfrak{X}).$$

  The optimality of x^h with respect to (3.3) follows from the definition of the gradient.

Since exactly one of the conditions on $\langle \nabla f(\hat{x}^{k-1}), \xi^k - \hat{x}^{k-1}\rangle$ must be satisfied, this completes the proof. ■

Proposition 3.2 *The simplicical decomposition method applied to* (3.3) *finds the optimal solution in a finite number of iterations, under the assumption that* $\mathfrak{X}$ *is a finite set of discrete points.*

Proof. By Lemma 3.1, at each iteration k of SDM we either expand the inner approximation $\text{conv}(V^k)$ by adding a new point $\xi^k \in \mathfrak{X} \backslash V^{k-1}$ to V^k, or we find the optimal solution. Since $\mathfrak{X}$ is a finite set, eventually the possible points to add to V^k will be exhausted and so the optimal solution must be found instead. ■

When $\mathfrak{X}$ contains non-discrete variables and hence is not finite in size, we will need to take a different approach based on exhausting the faces of $\text{conv}(\mathfrak{X})$.

Definition 3.3 *Let* C *be a convex set. A convex subset* F *of* C *is called a* ***face*** *of* C *if for every* $x \in F$ *and every* $y, z \in C$ *such that* x *is a convex combination of* y *and* z *we have* $y, z \in F$*. We will represent this relation as* $F \lhd C$*.*

Observe that C itself and the empty set are both faces of C according to this definition. In some circumstances we will want to exclude these.

Definition 3.4 *Let* C *be a convex set. A face* F *of* C *is called a* ***proper face*** *of* C *if it is a non-empty strict subset of* C*.*

We will make use of the following standard definitions.

Definition 3.5 *The* ***affine hull*** *of a set* $X \subset \mathbb{R}^n$*, denoted* $\text{aff}(X)$*, is the set of all points which may be obtained as an affine combination of points in* X *i.e. if* x *is in* $\text{aff}(X)$ *then there exist points* $x_1, \ldots, x_n$ *in* X *and weights* $\lambda_1, \ldots, \lambda_n$ *satisfying* $\sum_{i=1}^n \lambda_i = 1$ *such that*

$$x = \sum_{i=1}^n \lambda_i x_i.$$

(This is also the smallest affine set containing X*.)*

Note that this definition differs from that of a convex hull in that the weights need not be in the interval $[0, 1]$.

Definition 3.6 *The* ***relative interior*** *of a set* $X \subset \mathbb{R}^n$*, denoted* $\text{ri}(X)$*, is its interior with respect to its affine hull i.e. if* x *is in* $\text{ri}(X)$ *then there exists a neighbourhood of* x *such that any point in the intersection of that neighbourhood and* $\text{aff}(X)$ *is also in* X*.*

Definition 3.7 *The* ***lineality space*** *of a convex set* $X \subset \mathbb{R}^n$*, denoted* $\operatorname{lin}(X)$ *is the largest subspace of* $\mathbb{R}^n$ *such that*

$$x + \operatorname{lin}(X) \subset X \quad \forall x \in X.$$

In particular if X *is an affine subspace then* $\operatorname{lin} X$ *is the same subspace translated such that it includes the origin.*

Lemmas 3.8 and 3.9 are well-known results relating to the properties of faces.

Lemma 3.8 *A convex set is the union of the relative interiors of its faces (including non-proper faces). Furthermore, these relative interiors are all disjoint sets.*

Lemma 3.9 *Polyhedral sets have a finite number of faces. Furthermore, all of these faces are themselves polyhedral sets, and can be represented in the form* $H \cap \operatorname{conv}(\mathfrak{X})$*, where* H *is a supporting hyperplane of* $\operatorname{conv}(\mathfrak{X})$*.*

We can now work towards the main result. Lemma 3.10 establishes the intuitively obvious result that linear optimisation over a set defined by mixed-integer linear constraints must produce an extreme point of the convex hull of the feasible region.

Lemma 3.10 *Let* $\mathfrak{X} \subset \mathbb{R}^n$ *be defined as in* (3.4)*. Let* $X = \operatorname{conv} \mathfrak{X}$*. Then for all* $d \in \operatorname{lin} \operatorname{aff} \mathfrak{X} \backslash \{0\}$ *and all* $x^\star \in \operatorname{argmin}_x \{\langle d, x\rangle \mid x \in \mathfrak{X}\}$*,* $x^\star \notin \operatorname{ri} X$*.*

Proof. (Note that $\operatorname{aff} \mathfrak{X} = \operatorname{aff} X$.) Assume for the purposes of contradiction that $x^\star \in \operatorname{ri} X$. By definition there exists an open ball $B_\epsilon(x^\star)$ of radius $\epsilon > 0$ such that every point in the intersection of $B_\epsilon(x^\star)$ and $\operatorname{aff} X$ is also in X. In particular this includes the point $x^\star - \epsilon d/2 \|d\|$.

However, since $x^\star \in \operatorname{argmin}_x \{\langle d, x\rangle \mid x \in \mathfrak{X}\}$ there are no points in the intersection of $\mathfrak{X}$ and the halfspace $\{x \mid \langle d, x\rangle < \langle d, x^\star\rangle\}$. Since X is the convex hull of $\mathfrak{X}$ there are also no points in the intersection of X and that halfspace. This is a contradiction. ■

Lemma 3.11 demonstrates that any face F of $\operatorname{conv}(\mathfrak{X})$ does not extend beyond the convex hull of the points of $\mathfrak{X}$ which are contained in F.

Lemma 3.11 *Let* $\mathfrak{X} \subset \mathbb{R}^n$ *be defined as in* (3.4)*. Let* F *be a face of* $\operatorname{conv}(\mathfrak{X})$*. Then*

$$F = \operatorname{conv}(\mathfrak{X} \cap F).$$

Proof. Since it is a face of a polyhedral set, F can be expressed as the intersection of $\operatorname{conv}(\mathfrak{X})$ with one of its supporting hyperplanes $H = \left\{x \mid \alpha^T x = \beta\right\}$ (see Lemma 3.9). Furthermore $\alpha^T x \leqslant \beta$ for all $x \in \operatorname{conv}(\mathfrak{X})$.

By definition, any point $x_0 \in F(\subseteq \operatorname{conv}(\mathfrak{X}))$ may be expressed as the convex combination of a set of points $x_1, \ldots, x_n$ in $\mathfrak{X}$:

$$x_0 = \lambda_1 x_1 + \cdots + \lambda_n x_n, \sum_{i=1}^{n} \lambda_i = 1$$

Multiply both sides by α^T:

$$\alpha^T x_0 = \lambda_1 \alpha^T x_1 + \cdots + \lambda_n \alpha^T x_n, \sum_{i=1}^{n} \lambda_i = 1$$

If $x_0 \in F$ then $\alpha^T x_0 = \beta$. Since $\alpha^T x \leqslant \beta$ for all $x \in \mathfrak{X}$ we have

$$\beta = \alpha^T x_i$$

for all $i \in \{1, \ldots, n\}$, and so $x_i \in H \cap \mathfrak{X} \subseteq H \cap \operatorname{conv}(\mathfrak{X}) = F$ for all $i \in \{1, \ldots, n\}$. Therefore every point in F may be expressed as a convex combination of points in $\mathfrak{X} \cap F$, and so $F \subseteq \operatorname{conv}(\mathfrak{X} \cap F)$. $\operatorname{conv}(\mathfrak{X} \cap F) \subseteq F$ follows from the observation that since F is a convex set, the convex hull of a subset of F is itself a subset of F. ■

Proposition 3.12 establishes that if the termination conditions of the simplicial decomposition method are not satisfied, then the next vertex $\xi^\star$ will expand the inner approximation $\operatorname{conv} V$ by 'exploring' a new face of X. Since X has a finite number of faces to explore, the desired convergence result follows directly in Proposition 3.13.

Proposition 3.12 *Let $\mathfrak{X} \subset \mathbb{R}^n$ be defined as in* (3.4)*. Let V be a finite subset of $\mathfrak{X}$. Let $X = \operatorname{conv}(\mathfrak{X})$. For all $\hat{x} \in \operatorname{conv} V$, $\xi^\star \in \mathfrak{X}$ and $d \in \operatorname{lin\,aff} \mathfrak{X} \backslash \{0\}$ which satisfy the following conditions:*

$$\langle d, \xi - \hat{x} \rangle \geqslant 0 \ \forall \xi \in \operatorname{conv}(V) \tag{3.5}$$

$$\min_x \left\{ \langle d, x - \hat{x} \rangle \mid x \in \mathfrak{X} \right\} < 0 \tag{3.6}$$

$$\xi^\star \in \operatorname*{argmin}_x \left\{ \langle d, x \rangle \mid x \in \mathfrak{X} \right\}, \tag{3.7}$$

there exists a proper face $F \lhd X$ which satisfies the following conditions:

$$\xi^\star \in \operatorname{ri} F \tag{3.8}$$

$$\operatorname*{proj}_{\operatorname{lin\,aff} F} d = 0 \tag{3.9}$$

$$F \cap \operatorname{conv} V = \varnothing \tag{3.10}$$

Proof. Since $d \in \operatorname{lin\,aff} \mathfrak{X} \backslash \{0\}$ and precondition (3.7) holds, the preconditions of Lemma 3.10 are satisfied; therefore, $\xi^\star$ is not in the relative interior of X. By 3.8, $\xi^\star$ is in the relative interior of a (non-empty) face $F \neq X$. By definition this means F is a proper face which satisfies (3.8). We will show that this face also satisfies (3.9) and (3.10).

Since $\xi^\star$ is in F it is also in $\mathfrak{X} \cap F$. Since $\mathfrak{X} \cap F$ is a subset of $\mathfrak{X}$, by precondition (3.7) we have

$$\xi^\star \in \operatorname*{argmin}_x \{\langle d, x\rangle \mid x \in \mathfrak{X} \cap F\}.$$

Since taking the convex hull relaxation of a linear IP does not change its optimal value, and any point in $\mathfrak{X} \cap F$ is also in $\operatorname{conv}(\mathfrak{X} \cap F)$, we have

$$\operatorname*{argmin}_x \{\langle d, x\rangle \mid x \in \mathfrak{X} \cap F\} \subseteq \operatorname*{argmin}_x \{\langle d, x\rangle \mid x \in \operatorname{conv}(\mathfrak{X} \cap F)\}.$$

By Lemma 3.11 we have $\operatorname{conv}(\mathfrak{X} \cap F) = F$ and hence

$$\operatorname*{argmin}_x \{\langle d, x\rangle \mid x \in \operatorname{conv}(\mathfrak{X} \cap F)\} = \operatorname*{argmin}_x \{\langle d, x\rangle \mid x \in F\}.$$

Since $\langle d, x\rangle = \langle \operatorname{proj}_{\operatorname{lin\,aff} F} d, x\rangle$ for all x in F we have

$$\operatorname*{argmin}_x \{\langle d, x\rangle \mid x \in F\} = \operatorname*{argmin}_x \left\{\left\langle \operatorname*{proj}_{\operatorname{lin\,aff} F} d, x\right\rangle \mid x \in F\right\}$$

Therefore we have

$$\xi^\star \in \operatorname*{argmin}_x \left\{\left\langle \operatorname*{proj}_{\operatorname{lin\,aff} F} d, x\right\rangle \mid x \in F\right\}.$$

Since $\xi^\star \in \operatorname{ri} F$ we must have $\operatorname{proj}_{\operatorname{lin\,aff} F} d = 0$ and hence F satisfies (3.9).

We will show that F satisfies (3.10) by contradiction. Assume that there exists a point q in the intersection of F and $\operatorname{conv} V$. Since $q \in F$ we know that $q - \xi^\star \in \operatorname{lin\,aff} F$. Since $\operatorname{proj}_{\operatorname{lin\,aff} F} d = 0$ we have

$$\langle d, q\rangle = \langle d, \xi^\star\rangle \tag{3.11}$$

Since $\langle d, -\hat{x}\rangle$ is constant with respect to x we have

$$\operatorname*{argmin}_x \{\langle d, x\rangle \mid x \in \mathfrak{X}\} = \operatorname*{argmin}_x \{\langle d, x - \hat{x}\rangle \mid x \in \mathfrak{X}\}$$

and therefore by precondition (3.6) we have

$$\xi^\star \in \operatorname*{argmin}_x \{\langle d, x - \hat{x}\rangle \mid x \in \mathfrak{X}\} \tag{3.12}$$

By (3.11), (3.12) and (3.7) we have

$$\langle d, q - \hat{x}\rangle = \langle d, \xi^\star - \hat{x}\rangle = \min_x \{\langle d, x - \hat{x}\rangle \mid x \in \mathfrak{X}\} < 0.$$

Since $q \in \operatorname{conv} V$ this contradicts precondition (3.5). Therefore the intersection of F and $\operatorname{conv} V$ must be empty, and so F satisfies (3.10). ■

Proposition 3.13 *The simplicical decomposition method applied to* (3.3) *finds the optimal solution in a finite number of iterations.*

Proof. By Lemma 3.1, at each iteration k of SDM we either expand the inner approximation $\operatorname{conv}(V^k)$ by adding a new point $\xi^k \in \mathfrak{X} \backslash V^{k-1}$ to V^k, or we find the optimal solution. Furthermore, at each iteration we know by construction that either $\hat{x}^{k-1}$ is an optimal solution for (3.3), or the following conditions are satisfied:

- $\hat{x}^{k-1} \in \operatorname{conv}(V^{k-1})$
- $\xi^k \in \mathfrak{X}$
- $\nabla f(\hat{x}^{k-1}) \in \operatorname{lin\,aff} \mathfrak{X} \backslash \{0\}$
- $\langle \nabla f(\hat{x}^{k-1}), \xi^k - \hat{x}^{k-1}\rangle \geqslant 0 \ \forall \xi \in \operatorname{conv}(V^{k-1})$
- $\min_x \left\{\langle \nabla f(\hat{x}^{k-1}), x - \hat{x}^{k-1}\rangle \mid x \in \mathfrak{X}\right\} < 0$
- $\xi^k \in \operatorname{argmin}_x \left\{\langle \nabla f(\hat{x}^{k-1}), x\rangle \mid x \in \mathfrak{X}\right\}$

Therefore, by Proposition 3.12, at each iteration k which precedes the discovery of an optimal solution there exists a proper face F^k such that

- $\xi^k \in \operatorname{ri} F^k$ and hence $F^k \cap \operatorname{conv} V^k \neq \varnothing$, but
- $F^k \cap \operatorname{conv} V^{k-1} = \varnothing$.

Let $\mathfrak{F}^k$ denote the set of faces of $\text{conv}(\mathfrak{X})$ which have non-zero intersection with V^k. At least one face F^k is added to $\mathfrak{F}^k$ at each iteration which precedes the discovery of an optimal solution. Since $\text{conv}(\mathfrak{X})$ is a polyhedral set it has a finite number of faces, so eventually an optimal solution must be discovered. ■

Remark 3.14 *An attractive property of this convergence proof is that it can be easily generalised to the case where f may change from one iteration to the next. We will make use of this observation in Section 4.2.4.*

3.3 Frank-Wolfe Method for Non-Smooth Optimisation

The Frank-Wolfe method and its generalisations studied in the previous sections utilise the gradient of the objective function to find an appropriate search direction at each iteration. As such, these methods can only be applied to optimisation problems with smooth objective functions. In this section we propose a generalisation of the Frank-Wolfe type methods to non-smooth convex optimisation.

Definition 3.15 *The **limit inferior** of a sequence of numbers $\{x_k\}$ as $k \to \infty$ is defined by*

$$\liminf_{k\to\infty} \{x_k\} := \lim_{k\to\infty} \left(\inf_{m \geqslant k} x_m \right).$$

The limit inferior of a sequence of sets $\{X_k\}$ as $k \to \infty$ is defined by

$$\liminf_{k\to\infty} X_k := \{x \mid \exists \{x_k\} \ \ s.t.\ x_k \in X_k \, \forall k,\ x_k \to x\}$$

Assume E is a subspace of a metric space X, Y is an ordered metric space, and a is a limit point of E. The limit inferior of $f : E \to Y$ as $x \to a$ is defined by

$$\liminf_{x\to a} f(x) := \lim_{\epsilon \downarrow 0} \left(\inf \{ f(x) \mid x \in E \cap B_\epsilon(a) \backslash \{a\} \} \right)$$

where $B_\epsilon(a)$ is the metric ball of radius ϵ around a.

*The definition of the **limit superior*** $(\limsup)$ *of a sequence of numbers or a function follows directly by analogy. The limit superior of a sequence of sets requires $x^k \to x$ and $x^k \in X_{n_k}$ for all k, where $\{X_{n_k}\}$ is a subsequence of $\{X_k\}$.*

Definition 3.16 *The* ***directional derivative*** *of f at x in the direction of d is defined by*

$$f'(x;d) := \liminf_{\tau \downarrow 0} \frac{f(x+\tau d) - f(x)}{\tau}.$$

Since f is continuous the directional derivative has a single value even when f is non-smooth at x.

Given a problem of the form

$$\zeta^{CP} = \min_x \{f(x) \mid x \in X\}, \tag{3.13}$$

where f is a convex, continuous, but not necessarily differentiable function whose subgradient ∂f and directional derivatives $f'(x;d)$ are known, and the feasible set X is closed and convex, the non-smooth Frank-Wolfe method consists of the following steps:

Initialise Find a feasible solution $\hat{x}^0 \in X$ for Equation 3.13. Set $k = 1$.

Step 1 Set $s^k \in \operatorname{argmin}\left\{\|s\| \mid s \in \partial f(\hat{x}^{k-1})\right\}$.

Step 2 Set $\xi^k \in \arg\min_{x \in X}\left\{s^k(x - \hat{x}^{k-1})\right\}$ and $d^k = \xi^k - \hat{x}^{k-1}$.

Step 3 Set $t^k \in \arg\min_{0 \leqslant \tau \leqslant 1}\left\{f(\hat{x}^{k-1} + \tau d^k)\right\}$.

Step 4 Set $\hat{x}^k = \hat{x}^k + t^k d^k$.

Step 5 If $\min\left\{f'(\hat{x}^k; d) \mid d \in X - \left\{x^k\right\}\right\} \geqslant 0$, terminate; $\hat{x}^k$ is a local minimum (and hence a global minimum by virtue of convexity of f and X).

Step 6 Set $k = k + 1$ and return to Step 1.

As with the normal Frank-Wolfe method, we can generalise the step length update in Step 3 to utilise a heuristic instead, for example a generalised version of the Armijo rule:

Initialise Set an initial step length $0 < s \leqslant 1$, a step-size multiplier $0 < \beta < 1$, and a parameter $0 < \gamma < 1$ which determines the required accuracy of the gradient projection. Set $m = 0$.

Step 1 Set $\tau = \beta^m s$ (this is the trial step length).

Step 2 If $f(\hat{x}^{k-1}) - f(\hat{x}^{k-1} + \tau d^k) \geqslant -\gamma\tau\nabla f(\hat{x}^{k-1})^T d^k$ then set $t^k = \tau$ and terminate.

Step 3 Set $m = m + 1$ and return to Step 1.

Remark 3.17 *Unlike the corresponding step from the ordinary Frank-Wolfe method, Step 1 of the non-smooth Frank-Wolfe method (in which we calculate the minimal subgradient of the objective function f at a point $\hat{x}^{k-1}$) is not necessarily trivial. In practice we might use a bundle method to generate increasingly accurate approximations of the minimal subgradient, while incorporating a guarantee of descent with respect to ξ^k. In the theoretical results which follow we will concentrate on the simpler case in which Step 1 can be solved exactly.*

We will now show that the non-smooth Frank-Wolfe method using the generalised Armijo rule, applied to (3.13), converges to an optimal solution. The following definitions and proofs are analogous to those given for the conditional gradient method in [12, Section 2.2].

Definition 3.18 $\bar{x} \in X$ *is* ***stationary*** *for* (3.13) *if*

$$f'(\bar{x}; d) \geqslant 0 \quad \forall\, d \in X - \{\bar{x}\}$$

When X is closed and convex, stationarity is a necessary condition for optimality of $\bar{x}$. It is also a sufficient condition if f is convex.

Definition 3.19 *The bounded direction sequence $\{d^k\}$ is* ***subgradient-related*** *to the solution sequence $\{\hat{x}^k\}$ if for any subsequence $\{\hat{x}^k\}_{k \in K}$ which converges to a non-stationary point, the corresponding sequence $\{d^k\}_{k \in K}$ satisfies the following condition:*

$$\limsup_{k \in K, k \to \infty} f'(\hat{x}^k; d^k) < 0 \tag{3.14}$$

Remark 3.20 *Especially in a non-differentiable setting,* (3.14) *may be satisfied even if $\{\hat{x}^k\}_{k \in K}$ converges to a stationary point.*

The following lemma does not make any assumptions upon the manner in which the solution sequence $\{\hat{x}^k\}$ is generated. We will assume that this sequence converges to some point $\bar{x}$ without loss of generality; since X is compact, we can replace $\{\hat{x}^k\}$ with a convergent subsequence if necessary. Therefore, this result can be applied to the non-smooth Frank-Wolfe method using the generalised Armijo rule.

Lemma 3.21 *The direction sequence $\{d^k\}$ and solution sequence $\{\hat{x}^k\}$ generated by the non-smooth Frank-Wolfe method are subgradient-related.*

Proof. Assume that $\{\hat{x}^k\}$ converges to a non-stationary point $\bar{x}$. By construction of d^k, the sequence $\{d^k\}$ is bounded. Furthermore, for all $k \geqslant 0$ we have

$$f'(\hat{x}^k; d^k) \leqslant f'(\hat{x}^k; d), \quad \forall\, d \in X - \{\hat{x}^k\}$$

Take the limit:

$$\limsup_{k \in K, k \to \infty} f'(\hat{x}^k; d^k) \leqslant \limsup_{k \in K, k \to \infty} f'(\hat{x}^k; d), \quad \forall\, d \in X - \{\hat{x}^k\} \tag{3.15}$$

Since f is locally Lipschitz continuous, by [26, Proposition 2.1.1] the function $(\hat{x}, d) \mapsto f'(\hat{x}; d)$ is upper semi-continuous and so we have

$$\limsup_{k \in K, k \to \infty} f'(\hat{x}^k; d) \leqslant f'(\bar{x}; d) \tag{3.16}$$

for any fixed $d \in X - \{\hat{x}^k\}$. Take $\bar{d} \in \operatorname{argmin}_d \left\{ f'(\bar{x}; d) \mid d \in X - \{\hat{x}^k\} \right\}$. By the assumed nonstationarity of $\bar{x}$ we have that

$$f'(\bar{x}; \bar{d}) < 0. \tag{3.17}$$

Therefore, by combining inequalities (3.15), (3.16) and (3.17) we have

$$\limsup_{k \in K, k \to \infty} f'(\hat{x}^k; d^k) < 0.$$

This is the required condition to demonstrate that $\{d^k\}$ is subgradient-related to $\{\hat{x}^k\}$. ■

Proposition 3.22 *Assume that x and d are chosen such that $f'(x, d) < 0$, and that $-\infty < f'(x, d) < \infty$. Then the generalised Armijo rule algorithm terminates after a finite number of iterations.*

Proof. The termination condition ofthe generalised Armijo rule is:

$$f(x + \tau d) - f(x) \leqslant \tau \gamma f'(x; d)$$

Assume for the purposes of contradiction that this condition is never satisfied, i.e.

$$f(x + \tau d) - f(x) - \tau \gamma f'(x; d) > 0$$

Rearrange and take the limit to obtain

$$\lim_{\tau \downarrow 0} \frac{f(x+\tau d) - f(x)}{\tau} - \gamma f'(x;d) \geqslant 0$$

and so

$$(1-\gamma) f'(x;d) \geqslant 0$$

Since $\gamma < 1$ we have

$$f'(x;d) \geqslant 0$$

which contradicts our initial condition that $f'(x,d) < 0$. Therefore the termination condition of the generalised Armijo rule must be satisfied by a sufficiently small τ, which the update step of the algorithm will eventually generate. ■

Proposition 3.23 *Assume that ζ^{CP} as defined in* (3.13) *is bounded below. Then the solution sequence $\{\hat{x}^k\}$ generated by the non-smooth Frank-Wolfe method with the generalised Armijo rule satisfies the condition*

$$\lim_{k \to \infty} \left(f(\hat{x}^k) - f(\hat{x}^{k-1}) \right) = 0$$

and $\lim_{k \to \infty} t^k f'(\hat{x}^k; d^k) = 0$.

Proof. By the convexity of f we have

$$f(\hat{x}^k + t^k d^k) \geqslant f(\hat{x}^k) + t^k f'(\hat{x}^k; d^k)$$

for all $k \geqslant 1$. Since $t^k > 0$ this is equivalent to

$$f'(\hat{x}^k; d^k) \leqslant \frac{f(\hat{x}^k + t^k d^k) - f(\hat{x}^k)}{t^k}. \tag{3.18}$$

If $\left\{t^k\right\}$ is generated according to the generalised Armijo rule we have

$$\frac{f(\hat{x}^k + t^k d^k) - f(\hat{x}^k)}{t^k} \leqslant \gamma f'(\hat{x}^k; d^k). \tag{3.19}$$

Since $\gamma \in (0,1)$, if $f'(\hat{x}^k; d^k) < 0$ then

$$f(\hat{x}^k + t^k d^k) < f(\hat{x}^k).$$

The update step for $\hat{x}^k$ preserves the property that

$$f(\hat{x}^k) \leqslant f(\hat{x}^{k-1} + t^{k-1}d^{k-1}) < f(\hat{x}^{k-1}). \tag{3.20}$$

i.e. the sequence $\{f(\hat{x}^k)\}$ is monotonically decreasing. By assumption f is bounded below over X, so $\{f(\hat{x}^k)\}$ is also bounded below. Therefore we have $\lim_{k\to\infty} f(\hat{x}^k) = \bar{f} > -\infty$ for some $\bar{f}$ and so

$$\lim_{k\to\infty} \left(f(\hat{x}^k) - f(\hat{x}^{k-1})\right) = 0, \tag{3.21}$$

as required.

By combining (3.19) (with $k \to k-1$) and (3.20) we have

$$f(\hat{x}^k) - f(\hat{x}^{k-1}) \leqslant t^{k-1}\gamma f'(\hat{x}^{k-1}; d^{k-1}).$$

for all $k \geqslant 2$. Take the limit of both sides:

$$\lim_{k\to\infty}(f(\hat{x}^k) - f(\hat{x}^{k-1})) \leqslant \gamma \lim_{k\to\infty} t^k f'(\hat{x}^{k-1}; d^{k-1}) = \gamma \lim_{k\to\infty} t^k f'(\hat{x}^k; d^k).$$

Since γ is an arbitrary positive constant, by (3.21) $\lim_{k\to\infty} t^k f'(\hat{x}^k; d^k)$ is bounded below by 0. Since t^k is strictly positive for all k and $f'(\hat{x}^k; d^k)$ is strictly negative for all k, $\lim_{k\to\infty} t^k f'(\hat{x}^k; d^k)$ is bounded above by 0. Therefore

$$\lim_{k\to\infty} t^k f'(\hat{x}^k; d^k) = 0,$$

as required. ■

It is well known [99] that the limits correspond to the Kuratowski(–Painlevé) limit of the epigraph multifunction, giving rise to the following definitions:

$$\begin{aligned} \operatorname{epi}(\text{e-ls}_{v\to w} f_v) &:= \liminf_{v\to w} \operatorname{epi} f_v\,, \\ \operatorname{epi}(\text{e-li}_{v\to w} f_v) &:= \limsup_{v\to w} \operatorname{epi} f_v\,. \end{aligned} \tag{3.22}$$

(Recall that epi denotes the epigraph, as in Definition 2.5.)

Definition 3.24 *Assume X is a vector space with dual space X^*. The* ***convex conjugate*** *of a function $f : X \to \mathbb{R}_\infty$ at a point $x^* \in X^*$ is given by*

$$f^*(x^*) := \sup_{x\in X} \{\langle x, x^*\rangle - f(x)\}\,.$$

The ***convex biconjugate*** *of a function $f : X \to \mathbb{R}_\infty$ at a point $x \in X$ is the convex conjugate of its convex conjugate, and is denoted f^{**}.*

We need the following result which is a special case of the general result found in [113].

Theorem 3.25 *[113, Theorem 3.4] Suppose $\{f_k\}_{k\in\mathbb{N}}$ is a family of locally uniformly bounded, closed convex, functions. Then*

$$\text{e-ls}_{k\to\infty} f_k = (\text{e-li}_{k\to\infty} f_k^*)^*. \tag{3.23}$$

Corollary 3.26 *Suppose $\{C_k\}_{k\in\mathbb{N}}$ is a family of closed convex, uniformly bounded sets and $C := \liminf_{k\to\infty} C_k$. Then*

$$\min_{\{d^k\to\bar{d}\}} \liminf_{k\to\infty} \delta^*_{C_k}(d^k) = \text{e-li}_{k\to\infty}\, \delta^*_{C_k}(\bar{d}) = \delta^*_C(\bar{d}). \tag{3.24}$$

Proof. The first identity corresponds to the well-known equivalent formulation of the epi-limit infimum which may be found in e.g. [7, Theorem 1.13]. We note that

$$\begin{aligned}
\text{epi}[\text{e-ls}_{k\to\infty}\delta_{C_k}(\cdot)] &= \liminf_{k\to\infty} \text{epi}\, \delta_{C_k}(\cdot) \\
&= \liminf_{k\to\infty} [C_k \times [0,+\infty)] \\
&= [\liminf_{k\to\infty} C_k] \times [0,+\infty)] = \text{epi}\, \delta_{\liminf_{k\to\infty} C_k}(\cdot) = \text{epi}\, \delta_C(\cdot).
\end{aligned}$$

The second identity now follows from Theorem 3.25 applied to the indicator function $x \mapsto \delta_{C_k}(x)$.

■

In the following result we will use the notation

$$\partial_{d^k} f(x^k) := \{z \in \partial f(x^k) \mid \langle z, d^k\rangle = f'(x^k, d^k)\}.$$

Corollary 3.27 *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a closed, convex function and $x_k \to \bar{x}$ along with $d^k \to \bar{d}$. Suppose in addition that we have $\liminf_{k\to\infty} \partial_{d^k} f(x^k) \neq \emptyset$. Then*

$$\liminf_{k\to\infty} f'(x^k, d^k) \geqslant \text{e-li}_{k\to\infty} \delta^*_{\partial_{d^k} f(x^k)}(\bar{d}) = f'(\bar{x}, \bar{d}). \tag{3.25}$$

Proof. First note that for any d we have

$$f'(x^k, d) = \sup\{\langle z, d\rangle \mid z \in \partial f(x^k)\} = \delta^*_{\partial f(x^k)}(d) \geqslant \delta^*_{\partial_{d^k} f(x^k)}(d).$$

As f is closed, convex (finite valued) we have $\{\partial f(x^k)\}_{k\in\mathbb{N}}$ locally uniformly bounded and so we may apply Corollary 3.26 to obtain

$$\begin{aligned}
\liminf_{k\to\infty} f'(x^k, d^k) &\geqslant \min_{\{d'\to\bar{d}\}} \liminf_{k\to\infty} \delta^*_{\partial_{d^k} f(x^k)}(d') = \text{e-li}_{k\to\infty} \delta^*_{\partial_{d^k} f(x^k)}(\bar{d}) \\
&= \delta^*_{\liminf_{k\to\infty} \partial_{d^k} f(x^k)}(\bar{d}).
\end{aligned}$$

As f is semi-smooth and regular it is directionally upper semi-continuous by [105, Corollary 2.2 and Proposition 2.3]. Hence we have $\langle z, \bar{d}\rangle = f'(\bar{x}, \bar{d})$ for any $z \in \liminf_{k\to\infty} \partial_{d^k} f(x^k)$ from which the result follows. ■

Proposition 3.28 *The limit point $\bar{x}$ of a convergent subsequence $\{\hat{x}^k\}_{k\in K}$ of the solution sequence $\{\hat{x}^k\}$ generated by the non-smooth Frank-Wolfe method using the generalised Armijo rule is stationary (and therefore optimal) with respect to the convex problem defined in* (3.13).

Proof. To arrive at a contradiction, assume that $\bar{x}$ is non-stationary. Since $\{d^k\}$ is subgradient-related to $\{\hat{x}^k\}$ (as per Lemma 3.21) we have that

$$\limsup_{k\in K, k\to\infty} f'(\hat{x}^k; d^k) < 0. \tag{3.26}$$

From the structure of the $\hat{x}^k$ update step it follows that $\lim_{k\to\infty} t^k = 0$. Via Proposition 3.23 we have $\lim_{k\to\infty} t^k f'(x^k, d^k) = 0$ and due to (3.26) we then must have $\lim_{k\to\infty} t^k = 0$.

Define $\bar{t}^k = t^k/\beta$. Recall that $\beta \in (0, 1)$ is the Armijo rule step length multiplier; assuming the initial step length was too long, $\bar{t}^k$ is the smallest candidate step length which did *not* satisfy the Armijo rule criterion at step k. Since $\{t^k\}$ converges to zero there exists some $\bar{k} \in K$ such that $\bar{t}^k \leqslant 1$ for all $k \geqslant \bar{k}$. By directional upper semi-continuity of $\{\partial f(x^k)\}_{k\in K}$ there exists a subset $K' \subseteq K$ and $z_k \in \partial_{d^k} f(x^k)$ for which $\lim_{k\in K', k\to\infty} z_k = z \in \liminf_{k\in K', k\to\infty} \partial_{d^k} f(x^k) \neq \varnothing$. For notational simplcity we rename this subsequence index set K' to be K.

Therefore, based on the condition in the generalised Armijo rule, we have

$$\gamma f'(\hat{x}^k; d^k) < \frac{f(\hat{x}^k + \bar{t}^k d^k) - f(\hat{x}^k)}{\bar{t}^k}$$

which implies

$$(\gamma - 1) f'(\hat{x}^k; d^k) < \frac{f(\hat{x}^k + \bar{t}^k d^k) - f(\hat{x}^k)}{\bar{t}^k} - f'(\hat{x}^k; d^k)$$

Note that (3.26) and $\gamma \in (0, 1)$ implies

$$\liminf_{k\in K, k\to\infty} (\gamma - 1) f'(\hat{x}^k; d^k) > 0.$$

Therefore we have

$$0 < \liminf_{k\in K, k\to\infty} \left(\frac{f(\hat{x}^k + \bar{t}^k d^k) - f(\hat{x}^k)}{\bar{t}^k} - f'(\hat{x}^k; d^k) \right).$$

Next note that

$$\begin{aligned}\liminf_{k\in K,k\to\infty}\left(\frac{f(\hat{x}^k+\bar{t}^k d^k)-f(\hat{x}^k)}{\bar{t}^k}-f'(\hat{x}^k;d^k)\right) &+ \liminf_{k\in K,k\to\infty} f'(\hat{x}^k;d^k)\\ &\leqslant \liminf_{k\in K,k\to\infty}\left(\frac{f(\hat{x}^k+\bar{t}^k d^k)-f(\hat{x}^k)}{\bar{t}^k}\right)\end{aligned}$$

and so we have

$$\begin{aligned}0 &< \liminf_{k\in K,k\to\infty}\left(\frac{f(\hat{x}^k+\bar{t}^k d^k)-f(\hat{x}^k)}{\bar{t}^k}-f'(\hat{x}^k;d^k)\right)\\ &\leqslant \liminf_{k\in K,k\to\infty}\left(\frac{f(\hat{x}^k+\bar{t}^k d^k)-f(\hat{x}^k)}{\bar{t}^k}\right)-\liminf_{k\in K,k\to\infty} f'(\hat{x}^k;d^k)\\ &\leqslant \liminf_{k\in K,k\to\infty}\left(\frac{f(\hat{x}^k+\bar{t}^k d^k)-f(\hat{x}^k)}{\bar{t}^k}\right)-f'(\bar{x},\bar{d}),\end{aligned} \tag{3.27}$$

having applied Corollary 3.27 to obtain the last inequality.

By the Mean Value Theorem for subgradients (e.g. [26, Theorem 2.3.7]) applied to f on the interval $[0,\bar{t}^k]$ there exists $\hat{t}^k \in (0,\bar{t}^k)$ for all $k \geqslant 0$ such that

$$\frac{f(\hat{x}^k+\bar{t}^k d^k)-f(\hat{x}^k)}{\bar{t}^k} \in \left\{s^T d^k \mid s \in \partial f(\hat{x}^k+\hat{t}^k d^k)\right\}$$

and so

$$\frac{f(\hat{x}^k+\bar{t}^k d^k)-f(\hat{x}^k)}{\bar{t}^k} \leqslant \max\left\{s^T d^k \mid s \in \partial f(\hat{x}^k+\hat{t}^k d^k)\right\}. \tag{3.28}$$

By the upper semi-continuity of ∂f, for each $\epsilon > 0$ there exists a $\delta > 0$ such that

$$\partial f(\hat{x}^k+\hat{t}^k d^k) \subset \partial f(\bar{x}) + \epsilon B(0;1) \quad \forall\, \hat{x}^k+\hat{t}^k d^k \in \bar{x}+\delta B(0;1)$$

for k sufficiently large and hence we have

$$\max\left\{s^T d^k \mid s \in \partial f(\hat{x}^k+\hat{t}^k d^k)\right\} \leqslant f'(\bar{x},\bar{d})+\epsilon,$$

for $k \in K$ sufficiently large. Thus for any $\varepsilon > 0$ we have

$$\liminf_{k\in K,k\to\infty}\left(\frac{f(\hat{x}^k+\bar{t}^k d^k)-f(\hat{x}^k)}{\bar{t}^k}\right)-f'(\bar{x},\bar{d}) \leqslant \epsilon.$$

As ϵ is arbitrary we have

$$\liminf_{k\in K,k\to\infty}\left(\frac{f(\hat{x}^k+\bar{t}^k d^k)-f(\hat{x}^k)}{\bar{t}^k}\right)-f'(\bar{x},\bar{d}) \leqslant 0 \tag{3.29}$$

which contradicts (3.27). ■

Remark 3.29 *The name 'conditional subgradient method' is avoided in this section since an algorithm with this name based on the structure of the subgradient method was proposed by Larsson et al. in [72]. This differs from the non-smooth Frank-Wolfe method stated above in that conditions are imposed on the choice of subgradient* s^k *in Step 1 based on the feasible set, but the step direction* ξ^k *is defined directly as the negative of this subgradient rather than performing the minimisation in Step 2. Under some conditions this may result in a step to a non-feasible point, which must be projected back onto the feasible set.*

Chapter 4

Calculating Dual Bounds with Frank-Wolfe-based Progressive Hedging

4.1 Introduction

In this chapter we will consider the problem of calculating high-quality dual bounds for two-stage SIP problems, as represented by (2.4) and repeated here for reference:

$$\begin{aligned} \zeta^{SIP} = &\min_{x,y} c^T x + \sum_{s \in S} \left[p_s d_s^T y_s \right] \\ &\text{s.t. } (x, y_s) \in K_s \quad \forall s \in S \end{aligned} \tag{4.1}$$

The set of feasible decisions K_s for scenario s is defined as follows:

$$K_s := \left\{ (x, y) \mid Ax \leqslant b, T_s x + W_s y_s \leqslant h_s, x \in \mathbb{R}^{n-q} \times \mathbb{Z}^q, y \in \mathbb{R}^{m-r} \times \mathbb{Z}^r \right\}$$

As in (2.16) we can reformulate this problem using non-anticipativity constraints as follows:

$$\begin{aligned} \zeta^{SIP} = &\min_{\boldsymbol{x}, \boldsymbol{y}, \bar{x}} \sum_{s \in S} p_s \left[c^T x_s + d_s^T y_s \right] \\ &\text{s.t. } (x_s, y_s) \in K_s \quad \forall s \in S \\ &x_s - \bar{x} = 0 \quad \forall s \in S \end{aligned} \tag{4.2}$$

With the exception of the non-anticipativity constraints $x_s - \bar{x} = 0$ this problem is separable by scenario s. By using Lagrangian relaxation to remove this constraint we can move the non-separable terms to the objective, where they can be dealt with more easily.

The Lagrangian corresponding to the relaxation of the non-anticipativity constraints in (4.2) is

$$L(\boldsymbol{x}, \boldsymbol{y}, \bar{x}, \boldsymbol{\lambda}) = \sum_{s \in S} p_s \left[(c^T x_s + d_s^T y_s) + \lambda_s^T (x_s - \bar{x}) \right] \tag{4.3}$$

In order to guarantee that the Lagrangian dual problem is bounded the dual feasibility condition $\sum_{s\in S} p_s\lambda_s = 0$ must be enforced. Under this condition the $\sum_{s\in S} p_s\lambda_s^T\bar{x}$ term is equal to zero and the Lagrangian may be expressed in the form

$$L(\boldsymbol{x}, \boldsymbol{y}, \bar{x}, \boldsymbol{\lambda}) = \sum_{s\in S} p_s \left[(c^T + \lambda_s^T)x_s + d_s^T y_s\right]. \tag{4.4}$$

The corresponding Lagrangian dual function is

$$\begin{aligned} \zeta^{LR}(\boldsymbol{\lambda}) = &\min_{\boldsymbol{x},\boldsymbol{y},\bar{x}} L(\boldsymbol{x}, \boldsymbol{y}, \bar{x}, \boldsymbol{\lambda}) \\ &\text{s.t. } (x, y_s) \in K_s \quad \forall s \in S \end{aligned} \tag{4.5}$$

and the corresponding Lagrangian dual problem is:

$$\zeta^{LD} = \max_{\lambda} \zeta^{LR}(\boldsymbol{\lambda}) \tag{4.6}$$

The inclusion of a maximisation over the dual variables makes (4.6) difficult to tackle directly. A well-known primal characterisation of ζ^{LD} (incorporating only primal variables) is:

$$\begin{aligned} \zeta^{LD} = &\min_{(\boldsymbol{x},\boldsymbol{y},\bar{x})} \sum_{s\in S} p_s \left[c^\top x_s + d_s^\top y_s\right] \\ &\text{s.t. } (x_s, y_s) \in \text{conv}(K_s) \quad \forall s \in S \\ &x_s = \bar{x} \quad \forall s \in S \end{aligned} \tag{4.7}$$

Since the feasible region has been convexified this is essentially a continuous Stochastic Program.

The optimal value of the Lagrangian dual problem ζ^{LD} is a high-quality dual bound for the corresponding SIP instance. However, calculating this optimal value is not necessarily trivial. The current chapter and Chapter 5 will develop methods for solving the primal characterisation of ζ^{LD} given in (4.7). The primary challenge facing these methods is that no explicit representation is available for the set $\text{conv}(K_s)$.

In this chapter we will also require the augmented Lagrangian dual. The **augmented Lagrangian** corresponding to ζ^{SIP}, using the augmenting function (2.23), is:

$$L^\rho(\boldsymbol{x}, \boldsymbol{y}, \bar{x}, \boldsymbol{\lambda}) = \sum_{s\in S} p_s L_s^\rho(x_s, y_s, \bar{x}, \lambda_s), \tag{4.8}$$

where

$$L_s^\rho(x_s, y_s, \bar{x}, \lambda_s) = (c^T x_s + d_s^T y_s) + \lambda_s^T(x_s - \bar{x}) + \frac{\rho}{2}\left\|x_s - \bar{x}\right\|_2^2. \tag{4.9}$$

4.2 Algorithm Design and Theory

4.2.1 Algorithm Background

This chapter covers the theoretical development and computational application of a novel algorithm, called Frank-Wolfe-based Progressive Hedging (FW-PH), which is used to compute high-quality Lagrangian bounds for SIPs efficiently and with a high potential for parallelization. The fundamental idea behind FW-PH is to apply the Progressive Hedging (PH) algorithm to the convex hull relaxation (4.7). Since this is itself a continuous stochastic program, the Progressive Hedging algorithm is guaranteed to converge to optimality. Although $\text{conv}(K_s)$ is only defined implicitly, we can get around this by applying a Frank-Wolfe (FW) type method to solve the PH subproblems.

FW-PH has two important advantages over the direct approach of applying PH to the integer stochastic program (4.2) and calculating dual bounds as in [45].

1. FW-PH requires less computational effort. To apply PH directly to a SIP, we must solve a quadratic mixed-integer program for each scenario in each iteration for the primal variable update, and calculating a dual bound requires an additional mixed-integer program to be solved for each scenario. By contrast, FW-PH requires only a single mixed-integer program for each scenario at each iteration, to both calculate a dual bound and update the primal variables.

2. Under reasonable assumptions, the sequence of dual bounds obtained by FW-PH is guaranteed to converge to the optimal Lagrangian dual bound ζ^{LD}. The dual bounds calculated in [45] have no such guarantee.

4.2.2 Convergence of Progressive Hedging

Psuedocode for the Progressive Hedging algorithm is given in Algorithm 4.1. The algorithm is given in a general form which is applicable to the original SIP (4.2) and the convex hull relaxation (4.7).

The termination condition on Line 16 of Algorithm 4.1 is motivated by the squared norm primal residuals $\left\|x_s^k - \bar{x}^k\right\|_2^2$ for all $s \in S$ which quantifies consensus and the dual residual $\left\|\bar{x}^k - \bar{x}^{k-1}\right\|_2^2$ which quantifies convergence. If all of these residuals are small then $\bar{x}^k$ is close to satisfying the necessary and sufficient conditions for optimality [21, Section 3.3]. We can determine whether all

Algorithm 4.1 PH applied to problem (4.2) ($D_s = K_s$) or (4.7) ($D_s = \mathrm{conv}(K_s)$).

Precondition: $\sum_{s \in S} p_s \lambda_s^0 = 0$
function PH(λ^0, ρ, k_{max}, ϵ)
  for $s \in S$ **do**
    $(x_s^0, y_s^0) \in \mathrm{argmin}_{x,y} \left\{ (c + \lambda_s^0)^\top x + d_s^\top y \mid (x, y) \in D_s \right\}$
  end for
  $\phi^0 \leftarrow \sum_{s \in S} p_s \left[(c + \lambda_s^0)^\top x_s^0 + d_s^\top y_s^0 \right]$
  $\bar{x}^0 \leftarrow \sum_{s \in S} p_s x_s^0$
  $\lambda_s^1 \leftarrow \lambda_s^0 + \rho(x_s^0 - \bar{x}^0)$ for all $s \in S$
  for $k = 1, \ldots, k_{max}$ **do**
    for $s \in S$ **do**
      $\phi_s^k \leftarrow \min_{x,y} \left\{ (c + \lambda_s^k)^\top x + d_s^\top y \mid (x, y) \in D_s \right\}$
      $(x_s^k, y_s^k) \in \mathrm{argmin}_{x,y} \left\{ L_s^\rho(x, y, \bar{x}^{k-1}, \lambda_s^k) \mid (x, y) \in D_s \right\}$
    end for
    $\phi^k \leftarrow \sum_{s \in S} p_s \phi_s^k$
    $\bar{x}^k \leftarrow \sum_{s \in S} p_s x_s^k$
    if $\sqrt{\sum_{s \in S} p_s \left\| x_s^k - \bar{x}^{k-1} \right\|_2^2} < \epsilon$ **then**
      return $(x^k, y^k, \bar{x}^k, \lambda^k, \phi^k)$
    end if
    $\lambda_s^{k+1} \leftarrow \lambda_s^k + \rho(x_s^k - \bar{x}^k)$ for all $s \in S$
  end for
  return $(x^{k_{max}}, y^{k_{max}}, \bar{x}^{k_{max}}, \lambda^{k_{max}}, \phi^{k_{max}})$
end function

of the residuals are small in a single step by considering their sum:

$$\left\|\bar{x}^k - \bar{x}^{k-1}\right\|_2^2 + \sum_{s\in S} p_s \left[\left\|x_s^k - \bar{x}^k\right\|_2^2\right] = \sum_{s\in S} p_s \left[\left\|x_s^k - \bar{x}^k\right\|_2^2 + \left\|\bar{x}^k - \bar{x}^{k-1}\right\|_2^2\right]$$

By the definition of the squared 2-norm we can rewrite this as:

$$\sum_{s\in S} p_s \left[\left\|x_s^k - \bar{x}^k\right\|_2^2 + \left\|\bar{x}^k - \bar{x}^{k-1}\right\|_2^2\right] = \sum_{s\in S}\sum_{i=1}^{n} p_s \left[(x_{s,i}^k - \bar{x}_i^k)^2 + (\bar{x}_i^k - \bar{x}_i^{k-1})^2\right]$$

where $x_{s,i}^k$, $\bar{x}_i^k$ and $\bar{x}_i^{k-1}$ are the ith elements of their respective vectors. Since the definition of $\bar{x}^k$ in line 15 implies that the dual feasibility condition $\sum_{s\in S} p_s(x_{s,i}^k - \bar{x}_i^k) = 0$ is satisfied, we can write

$$\begin{aligned}
&\sum_{s\in S}\sum_{i=1}^{n} p_s \left[(x_{s,i}^k - \bar{x}_i^k)^2 + (\bar{x}_i^k - \bar{x}_i^{k-1})^2\right] \\
&= \sum_{s\in S}\sum_{i=1}^{n} p_s \left[(x_{s,i}^k - \bar{x}_i^k)^2 + 2(x_{s,i}^k - \bar{x}_i^k)(\bar{x}_i^k - \bar{x}_i^{k-1}) + (\bar{x}_i^k - \bar{x}_i^{k-1})^2\right] \\
&= \sum_{s\in S}\sum_{i=1}^{n} p_s \left[(x_{s,i}^k - \bar{x}_i^k) + (\bar{x}_i^k - \bar{x}_i^{k-1})\right]^2 \\
&= \sum_{s\in S} p_s \left\|x_s^k - \bar{x}^{k-1}\right\|_2^2 .
\end{aligned}$$

The following proposition addresses the convergence of PH applied to problem (4.7).

Proposition 4.1 *Assume that problem* (4.7) *is feasible with* $\mathrm{conv}(K_s)$ *bounded for each* $s \in S$*, and let Algorithm 4.1 be applied to problem* (4.7) *(so that* $D_s = \mathrm{conv}(K_s)$ *for each* $s \in S$*) with tolerance* $\epsilon = 0$ *for each* $k \geqslant 1$*. Then, the limit* $\lim_{k\to\infty} \lambda^k = \lambda^*$ *exists, and furthermore,*

1. $\lim_{k\to\infty} \sum_{s\in S} p_s(c^\top x_s^k + q_s^\top y_s^k) = \zeta^{LD}$,
2. $\lim_{k\to\infty} \phi(\lambda^k) = \zeta^{LD}$,
3. $\lim_{k\to\infty}(x_s^k - \bar{x}^k) = 0$ *for each* $s \in S$,

and each limit point $(((x_s^*, y_s^*)_{s\in S}, \bar{x}^*)$ *is an optimal solution for* (4.7)*.*

Proof. Since the constraint sets $D_s = \mathrm{conv}(K_s)$, $s \in S$, are bounded, and problem (4.7) is feasible, problem (4.7) has an optimal solution $((x_s^*, y_s^*)_{s\in S}, z^*)$ with optimal value ζ^{LD}. The feasibility of problem (4.7), the linearity of its objective function, and the bounded polyhedral structure of its constraint set $D_s = \mathrm{conv}(K_s)$, $s \in S$, imply that the hypotheses for PH convergence to

the optimal solution are met (See Theorem 5.1 of [98]). Therefore, $\{\lambda^k\}$ converges to some λ^*, $\lim_{k\to\infty}\sum_{s\in S} p_s(c^\top x_s^k + q_s^\top y_s) = \zeta^{LD}$, $\lim_{k\to\infty}\phi(\lambda^k) = \zeta^{LD}$, and $\lim_{k\to\infty}(x_s^k - z^k) = 0$ for each $s \in S$ all hold. The boundedness of each $D_s = \text{conv}(K_s)$, $s \in S$, furthermore implies the existence of limit points $((x_s^*, y_s^*)_{s\in S}, z^*)$ of $\{((x_s^k, y_s^k)_{s\in S}, z^k)\}$, which are optimal solutions for (4.7). ■

Note that the convergence in Proposition 4.1 applies to the continuous problem (4.7) but *not* to the mixed-integer problem (4.2). In problem (4.2), the constraint sets K_s, $s \in S$, are not convex, so there is no guarantee that Algorithm 4.1 will converge when applied to (4.2).

4.2.3 Applying the Simplicial Decomposition Method

To use Progressive Hedging (as in Algorithm 4.1) to solve (4.7) requires a method for solving the subproblem

$$(x_s^k, y_s^k) \in \operatorname*{argmin}_{x,y} \left\{ L_s^\rho(x, y, \bar{x}^{k-1}, \lambda_s^k) : (x, y) \in \text{conv}(K_s) \right\} \tag{4.10}$$

appearing in Line 12 of the algorithm.

As we saw in Section 3.2, the Frank-Wolfe method and its generalisations are well suited to solving a problem of this form. In this chapter we will use the simplicial decomposition method (SDM) to solve these subproblems.

The application of SDM to solve problem (4.10), i.e., to minimise $L_s^\rho(x, y, \bar{x}, \lambda_s)$ over $(x, y) \in \text{conv}(K_s)$, for a given $s \in S$, is presented in Algorithm 4.2. Here, t_{max} is the maximum number of iterations and $\tau > 0$ is a convergence tolerance. Γ^t is the bound gap used to measure closeness to optimality, and ϕ_s is used to compute a Lagrangian bound as described in the next section. The inner approximation to $\text{conv}(K_s)$ at iteration $t \geqslant 1$ takes the form $\text{conv}(V_s^t)$, where V_s^t is a finite set of points, with $V_s^t \subset \text{conv}(K_s)$. The points added by Algorithm 4.2 to the initial set, V_s^0, to form V_s^t, are all in K_s: here $\mathcal{V}(\text{conv}(K_s))$ is the set of extreme points of $\text{conv}(K_s)$ and, of course, $\mathcal{V}(\text{conv}(K_s)) \subseteq K_s$.

Observe that

$$\nabla_{(x,y)} L_s^\rho(x, y, \bar{x}, \lambda_s)\big|_{(x,y)=\left(x_s^{t-1}, y_s^{t-1}\right)} = \begin{bmatrix} c + \lambda_s + \rho(x_s^{t-1} - z) \\ q_s \end{bmatrix} = \begin{bmatrix} c + \widehat{\lambda}_s \\ q_s \end{bmatrix},$$

and so the optimization at Line 5 minimises the gradient approximation to $L_s^\rho(x, y, \bar{x}, \lambda_s)$ at the point (x_s^{t-1}, y_s^{t-1}). Since this is a linear objective function, optimization over $\mathcal{V}(\text{conv}(K_s))$ can be

Algorithm 4.2 SDM applied to problem (4.10).

1: Precondition: $V_s^0 \subset \text{conv}(K_s)$ and $z = \sum_{s \in S} p_s x_s^0$
2: **function** SDM(V_s^0, x_s^0, λ_s, z, t_{max}, τ)
3: **for** $t = 1, \ldots, t_{max}$ **do**
4: $\widehat{\lambda}_s^t \leftarrow \lambda_s + \rho(x_s^{t-1} - z)$
5: $(\widehat{x}_s, \widehat{y}_s) \in \text{argmin}_{x,y} \left\{ (c + \widehat{\lambda}_s^t)^\top x + q_s^\top y \mid (x, y) \in \mathcal{V}(\text{conv}(K_s)) \right\}$
6: **if** $t = 1$ **then**
7: $\phi_s \leftarrow (c + \widehat{\lambda}_s^t)^\top \widehat{x}_s + q_s^\top \widehat{y}_s$
8: **end if**
9: $\Gamma^t \leftarrow -[(c + \widehat{\lambda}_s^t)^\top (\widehat{x}_s - x_s^{t-1}) + q_s^\top (\widehat{y}_s - y_s^{t-1})]$
10: $V_s^t \leftarrow V_s^{t-1} \cup \{(\widehat{x}_s, \widehat{y}_s)\}$
11: $(x_s^t, y_s^t) \in \text{argmin}_{x,y} \{ L_s^\rho(x, y, z, \lambda_s) \mid (x, y) \in \text{conv}(V_s^t) \}$
12: **if** $\Gamma^t \leqslant \tau$ **then**
13: **return** $(x_s^t, y_s^t, V_s^t, \phi_s)$
14: **end if**
15: **end for**
16: **return** $(x_s^{t_{max}}, y_s^{t_{max}}, V_s^{t_{max}}, \phi_s)$
17: **end function**

accomplished by optimization over K_s (see, e.g., [90], Section I.4, Theorem 6.3). Hence Line 5 requires a solution of a single-scenario MILP.

The value of ϕ_s for the first iteration in particular ($t = 1$) is used to construct a dual bound for the overall SIP in Proposition 4.2.

The (x_s^t, y_s^t)-update at Line 11 can be accomplished by expressing (x, y) as a convex combination of the finite set of points, V_s^t, where the weights $a \in \mathbb{R}^{|V_s^t|}$ in the convex combination are now also decision variables. That is, the Line 11 problem is solved with a solution to the following convex continuous quadratic subproblem

$$(x_s^t, y_s^t, a) \in \underset{x,y,a}{\text{argmin}} \left\{ \begin{array}{c} L_s^\rho(x, y, z, \lambda_s) \text{s.t.} (x, y) = \sum_{(\widehat{x}^i, \widehat{y}^i) \in V_s^t} a_i (\widehat{x}^i, \widehat{y}^i), \\ \sum_{i=1,\ldots,|V_s^t|} a_i = 1, \text{ and } a_i \geqslant 0 \text{ for } i = 1, \ldots, |V_s^t| \end{array} \right\}. \tag{4.11}$$

For implementational purposes, the x and y variables may be substituted out of the objective of problem (4.11), leaving a as the only decision variable, with the only constraints being nonnegativity of the a components and the requirement that they sum to 1.

The convergence properties of the simplicial decomposition method (see Section 3.1.2) guarantee that an arbitrarily near-optimal solution can be found for the augmented Lagrangian 4.10 which comprises the main Progressive Hedging update step. However, the convergence speed of SDM may be slow. Furthermore, it will frequently be an inefficient use of computational resources to solve the

augmented Lagrangian calculation exactly or nearly exactly. If the choice of dual variables is poor (which is especially likely in the early iterations), an exact solution to the augmented Lagrangian is not much more useful than an approximate one. This motivates the development of the FW-PH method found in the next section.

4.2.4 FW-PH Method

The FW-PH method obtains an approximate solution to the augmented Lagrangian by running a small number of SDM iterations for each update step. This has the advantage of greatly reducing the number of MILP subproblems to be solved in each PH iteration. The standard PH convergence proof relies on exact updates and therefore is not directly applicable to FW-PH. The remainder of this section will demonstrate that despite this obstacle, FW-PH is still guaranteed to converge to the optimal Lagrangian dual bound.

The FW-PH algorithm is stated in pseudocode-form in Algorithm 4.3. Similar to Algorithm 4.1, the parameter ϵ is a convergence tolerance, and k_{max} is the maximum number of (outer) iterations. The parameter t_{max} is the maximum number of (inner) SDM iterations in Algorithm 4.2.

The parameter $\alpha \in \mathbb{R}$ affects the initial linearization point $\tilde{x}_s$ of the SDM method. Any value $\alpha \in \mathbb{R}$ may be used, but the use of $\tilde{x}_s = (1-\alpha)z^{k-1} + \alpha x_s^{k-1}$ in Line 6 is a crucial component in the efficiency of the FW-PH algorithm, as it enables the computation of a valid dual bound, ϕ^k, at *each* iteration of FW-PH without the need for additional MILP subproblem solutions. Specifically, we have the following result.

Proposition 4.2 *Assume that the precondition $\sum_{s\in S} p_s\lambda_s^0 = 0$ holds for Algorithm 4.3. At each iteration $k \geqslant 1$ of Algorithm 4.3, the value, ϕ^k, calculated at Line 9, is the value of the Lagrangian relaxation $\phi(\cdot)$ evaluated at a Lagrangian dual feasible point, and hence provides a finite lower bound on ζ^{LD}.*

Proof. Since $\sum_{s\in S} p_s\lambda_s^0 = 0$ holds and, by construction, $0 = \sum_{s\in S} p_s(x_s^0 - z^0)$, we have $\sum_{s\in S} p_s\lambda_s^1 = 0$ also. We proceed by induction on $k \geqslant 1$. At iteration k, the problem solved for each $s \in S$ at Line 5 in the first iteration ($t = 1$) of Algorithm 4.2 may be solved with the same optimal value by exchanging $\mathcal{V}(\text{conv}(K_s))$ for K_s; this follows from the linearity of the objective function. Thus,

Algorithm 4.3 FW-PH applied to problem (4.7).

1: **function** FW-PH($(V_s^0)_{s\in S}$, $(x_s^0, y_s^0)_{s\in S}$, λ^0, ρ, α, ϵ, k_{max}, t_{max})
2: $\quad\bar{x}^0 \leftarrow \sum_{s\in S} p_s x_s^0$
3: $\quad\lambda_s^1 \leftarrow \lambda_s^0 + \rho(x_s^0 - \bar{x}^0)$, for $s \in S$
4: $\quad$**for** $k = 1, \ldots, k_{max}$ **do**
5: $\quad\quad$**for** $s \in S$ **do**
6: $\quad\quad\quad\widetilde{x}_s \leftarrow (1-\alpha)z^{k-1} + \alpha x_s^{k-1}$
7: $\quad\quad\quad[x_s^k, y_s^k, V_s^k, \phi_s^k] \leftarrow \mathrm{SDM}(V_s^{k-1}, \widetilde{x}_s, \lambda_s^k, z^{k-1}, t_{max}, 0)$
8: $\quad\quad$**end for**
9: $\quad\quad\phi^k \leftarrow \sum_{s\in S} p_s \phi_s^k$
10: $\quad\quad\bar{x}^k \leftarrow \sum_{s\in S} p_s x_s^k$
11: $\quad\quad$**if** $\sqrt{\sum_{s\in S} p_s \left\|x_s^k - \bar{x}^{k-1}\right\|_2^2} < \epsilon$ **then**
12: $\quad\quad\quad$**return** $((x_s^k, y_s^k)_{s\in S}, \bar{x}^k, \lambda^k, \phi^k)$
13: $\quad\quad$**end if**
14: $\quad\quad\lambda_s^{k+1} \leftarrow \lambda_s^k + \rho(x_s^k - \bar{x}^k)$, for $s \in S$
15: $\quad$**end for**
16: $\quad$**return** $\left((x_s^{k_{max}}, y_s^{k_{max}})_{s\in S}, \bar{x}^{k_{max}}), \lambda^{k_{max}}, \phi^{k_{max}}\right)$
17: **end function**

an optimal solution computed at Line 5 may be used in the computation of $\phi_s(\widetilde{\lambda}_s^k)$ carried out in Line 7, where

$$\begin{aligned}\widetilde{\lambda}_s^k &:= \widehat{\lambda}_s^1 = \lambda_s^k + \rho(\widetilde{x}_s - z^{k-1}) = \lambda_s^k + \rho((1-\alpha)z^{k-1} + \alpha x_s^{k-1} - z^{k-1}) \\ &= \lambda_s^k + \alpha\rho(x_s^{k-1} - z^{k-1}).\end{aligned}$$

By construction, we have at each iteration $k \geqslant 1$ in Algorithm 4.3 that

$$\sum_{s\in S} p_s(x_s^{k-1} - z^{k-1}) = 0 \quad \text{and} \quad \sum_{s\in S} p_s \lambda_s^k = 0,$$

which establishes that $\sum_{s\in S} p_s \widetilde{\lambda}_s^k = 0$. Thus, $\widetilde{\lambda}^k$ is feasible for the Lagrangian dual problem, so that $\phi(\widetilde{\lambda}^k) = \sum_{s\in S} p_s \phi_s^k$, and, since each ϕ_s^k is the optimal value of a bounded and feasible mixed-integer linear program, we have $-\infty < \phi(\widetilde{\lambda}^k) < \infty$. ■

We establish convergence of Algorithm 4.3 for any $\alpha \in \mathbb{R}$ and $t_{max} \geqslant 1$. For the special case where we perform only one iteration of SDM for each outer iteration ($t_{max} = 1$), we require the additional assumption that the initial scenario vertex sets share a common point. More precisely, we require the assumption

$$\bigcap_{s\in S} \mathbf{proj}_x(\mathrm{conv}(V_s^0)) \neq \varnothing \tag{4.12}$$

which can, in practice, be effectively handled through appropriate initialization, under the standard assumption of relatively complete recourse. We describe one initialization procedure in Section 4.3.

Proposition 4.3 *Let the convexified separable deterministic equivalent SIP* (4.7) *have an optimal solution, and let Algorithm 4.3 be applied to* (4.7) *with* $k_{max} = \infty$, $\epsilon = 0$, $\alpha \in \mathbb{R}$, *and* $t_{max} \geqslant 1$. *If either* $t_{max} \geqslant 2$ *or* (4.12) *holds, then* $\lim_{k\to\infty} \phi^k = \zeta^{LD}$.

Proof. First note that for any $t_{max} \geqslant 1$, the sequence of inner approximations $\text{conv}(V_s^k)$, $s \in S$, will stabilise, in that, for some threshold $0 \leqslant \bar{k}_s$, we have for all $k \geqslant \bar{k}_s$

$$\text{conv}(V_s^k) =: \overline{D}_s \subseteq \text{conv}(K_s). \tag{4.13}$$

This follows from the conclusion of Proposition 3.12 that each expansion of the inner approximations $\text{conv}(V_s^k)$ takes the form $V_s^k \leftarrow V_s^{k-1} \cup \{(\hat{x}_s, \hat{y}_s)\}$, where $(\hat{x}_s, \hat{y}_s)$ is in the relative interior of a previously unexplored face of $\text{conv}(K_s)$. By a similar argument to Proposition 3.13, since each polyhedron $\text{conv}(K_s)$, $s \in S$ has only a finite number of such faces, the stabilization (4.13) must occur at some $\bar{k}_s < \infty$.

Case 1: $t_{\max} \geqslant 2$

The stabilizations (4.13), $s \in S$, are reached at some iteration $\bar{k} := \max_{s\in S}\left\{\bar{k}_s\right\}$. Noting that $\overline{D}_s = \text{conv}(V_s^k)$ for $k > \overline{k}$ we must have

$$(x_s^k, y_s^k) \in \operatorname*{argmin}_{x,y} \left\{ L_s^\rho(x, y, \bar{x}^{k-1}, \lambda_s^k) : (x, y) \in \text{conv}(K_s) \right\}. \tag{4.14}$$

Otherwise, due to Lemma 3.1, the call to SDM on Line 7 must return $V_s^k \supsetneq V_s^{k-1}$, contradicting the finite stabilization (4.13). Therefore, the $k \geqslant \bar{k}$ iterations of Algorithm 4.3 are identical to Algorithm 4.1 iterations, and so Proposition 4.1 implies that $\lim_{k\to\infty} x_s^k - \bar{x}^k = 0$, $s \in S$, and $\lim_{k\to\infty} \phi(\lambda^k) = \zeta^{LD}$. By the continuity of $\lambda \mapsto \phi_s(\lambda)$ for each $s \in S$, we have $\lim_{k\to\infty} \phi^k = \lim_{k\to\infty} \sum_{s\in S} p_s \phi_s(\lambda_s^k + \alpha(x_s^{k-1} - z^{k-1})) = \lim_{k\to\infty} \sum_{s\in S} p_s \phi_s(\lambda_s^k) = \lim_{k\to\infty} \phi(\lambda^k) = \zeta^{LD}$ for all $\alpha \in \mathbb{R}$.

Case 2: $t_{max} = 1$ and (4.12) holds

We have at each iteration $k \geqslant 1$ the optimality

$$(x_s^k, y_s^k) \in \operatorname*{argmin}_{x,y} \left\{ L_s^\rho(x_s, y_s, \bar{x}^{k-1}, \lambda_s^k) \mid (x_s, y_s) \in \text{conv}(V_s^k) \right\}.$$

When stabilisation occurs as in (4.13), the iterations $k \geqslant \bar{k}$ of Algorithm 4.3 are identical to PH iterations applied to the restricted problem

$$\min_{x,y,\bar{x}} \left\{ \sum_{s\in S} p_s(c^\top x_s + q_s^\top y_s) \mid (x_s, y_s) \in \overline{D}_s,\ \forall s \in S, x_s = \bar{x},\ \forall s \in S \right\}. \tag{4.15}$$

We have initialised the sets $(V_s^0)_{s\in S}$ such that $\cap_{s\in S}\mathbf{proj}_x \operatorname{conv}(V_s^0) \neq \varnothing$, so since the inner approximations to $\operatorname{conv}(K_s)$ only expand in the algorithm, $\cap_{s\in S}\mathbf{proj}_x(\overline{D}_s) \neq \varnothing$. Therefore, problem (4.15) is a feasible and bounded linear program, and so the PH convergence described in Proposition 4.1 with $D_s = \overline{D}_s$, $s \in S$, holds for its application to problem (4.15). That is, for each $s \in S$, we have 1) $\lim_{k\to\infty} \lambda_s^k = \lambda_s^*$ and $\lim_{k\to\infty}(x_s^k - \bar{x}^k) = 0$; and 2) for all limit points $((x_s^*, y_s^*)_{s\in S}, \bar{x}^*)$, we have the feasibility and optimality of the limit points, which implies $x_s^* = \bar{x}^*$ and

$$\min_{x,y} \left\{ (c + \lambda_s^*)^\top (x - x^*) + q_s^\top (y - y^*) \text{s.t.} (x, y) \in \overline{D}_s \right\} = 0 \tag{4.16}$$

Next, for each $s \in S$ the compactness of $\operatorname{conv}(K_s) \supseteq \overline{D}_s$, the continuity of the minimum value function

$$\lambda \mapsto \min_{x,y} \left\{ (c + \lambda)^\top x + q_s^\top y \text{s.t.} (x, y) \in \overline{D}_s \right\},$$

and the limit $\lim_{k\to\infty} \widetilde{\lambda}_s^{k+1} = \lim_{k\to\infty} \lambda_s^{k+1} + \alpha\rho(x_s^k - \bar{x}^k) = \lambda_s^*$, together imply that

$$\lim_{k\to\infty} \min_{x,y} \left\{ (c + \widetilde{\lambda}_s^{k+1})^\top (x - x^k) + q_s^\top (y - y^k) \text{s.t.} (x, y) \in \overline{D}_s \right\} = 0. \tag{4.17}$$

Recall that $\widetilde{\lambda}_s^k = \lambda_s^k + \rho\alpha(x_s^{k-1} - z^{k-1})$ is the $t = 1$ value of $\widehat{\lambda}_s^t$ defined in Line 4 of Algorithm 4.2. Thus, for $k + 1 > \bar{k}$, we have due to the stabilization (4.13) that

$$\begin{aligned} \min_{x,y} \left\{ (c + \widetilde{\lambda}_s^{k+1})^\top (x - x^k) + q_s^\top (y - y^k) \text{s.t.} (x, y) \in \overline{D}_s \right\} = \\ \min_{x,y} \left\{ (c + \widetilde{\lambda}_s^{k+1})^\top (x - x^k) + q_s^\top (y - y^k) \text{s.t.} (x, y) \in \operatorname{conv}(K_s) \right\} \end{aligned} \tag{4.18}$$

If equality (4.18) does not hold, then the inner approximation expansion $\overline{D}_s \subset \operatorname{conv}(V_s^{k+1})$ must occur, since a point $(\widehat{x}_s, \widehat{y}_s) \in \operatorname{conv}(K_s)$ that can be strictly separated from $\overline{D}_s$ would have been discovered during the iteration $k + 1$ execution of Algorithm 4.2, Line 5, $t = 1$. The expansion $\overline{D}_s \subset \operatorname{conv}(V_s^{k+1})$ contradicts the finite stabilization (4.13), and so (4.18) holds. Therefore, the equalities (4.17) and (4.18) imply that

$$\lim_{k\to\infty} \min_{x,y} \left\{ (c + \widetilde{\lambda}_s^{k+1})^\top (x - x^k) + q_s^\top (y - y^k) \text{s.t.} (x, y) \in \text{conv}(K_s) \right\} = 0. \tag{4.19}$$

Our argument has shown that for all limit points (x_s^*, y_s^*), $s \in S$, the following stationarity condition is satsfied:

$$(c + \lambda_s^*)^\top (x - x_s^*) + q_s^\top (y - y_s^*) \geqslant 0 \quad \forall (x, y) \in \text{conv}(K_s), \tag{4.20}$$

which together with the feasibility $x_s^* = \bar{x}^*$, $s \in S$ implies that each limit point $((x_s^*, y_s^*)_{s\in S}, \bar{x}^*)$ is optimal for problem (4.7) and λ^* is optimal for the dual problem (4.6).

Thus, for all $t_{max} \geqslant 1$, we have shown $\lim_{k\to\infty}(x_s^k - \bar{x}^k) = 0$, $s \in S$, and $\lim_{k\to\infty} \phi(\lambda^k) = \zeta^{LD}$. By a similar argument to Case 1 (drawing on the continuity of $\lambda \mapsto \phi_s(\lambda)$) we can show that $\lim_{k\to\infty} \phi^k = \zeta^{LD}$.

■

While using a large value of t_{max} more closely matches Algorithm 4.3 to the original PH algorithm as described in Algorithm 4.1, we are motivated to use a small value of t_{max} since the work per iteration is proportional to t_{max}. Specifically, each iteration requires solving $t_{max}|S|$ MILP subproblems, and $t_{max}|S|$ continuous convex quadratic subproblems. (For reference, Algorithm 4.1 applied to problem (4.2) requires the solution of $|S|$ MIQP subproblems for each λ update and $|S|$ MILP subproblems for each Lagrangian bound ϕ computation.)

4.3 Computational Results

4.3.1 Preliminary Information

We performed computations using a C++ implementation of Algorithm 4.1 ($D_s = K_s$, $s \in S$) and Algorithm 4.3 using CPLEX 12.5 [63] as the solver for all subproblems. For reading SMPS files into scenario-specific subproblems and for their interface with CPLEX, we used modified versions of the COIN-OR [1] Smi and Osi libraries. The computing environment is the Raijin cluster maintained by Australia's National Computing Infrastructure (NCI) and supported by the Australian Government [89]. The Raijin cluster is a high performance computing (HPC) environment which has 3592 nodes (system units), 57472 cores of Intel Xeon E5-2670 processors with up to 8 GB PC1600 memory per core (128 GB per node). All experiments were performed in a serial setting using a single node and one thread per CPLEX solve.

In the experiments with Algorithms 4.1 and 4.3, we set the convergence tolerance at $\epsilon = 10^{-3}$. For Algorithm 4.3, we set $t_{max} = 1$. Also, for all experiments performed, we set $\lambda^0 = 0$. In this case, convergence of our algorithm requires that (4.12) holds, which can be guaranteed during the initialization of the inner approximations $(V_s^0)_{s\in S}$. Under the standard assumption of *relatively complete resource*, i.e., the assumption that for all $x \in X$ and $s \in S$ there exists y_s such that $(x, y_s) \in K_s$, a straightforward mechanism for ensuring this assumption is to solve the recourse problems for any fixed $\widehat{x} \in X$. Specifically, for each $s \in S$, let

$$\widehat{y}_s \in \arg\min_y \{q_s^\top y \mid (\widehat{x}, y) \in K_s\},$$

and initialise V_s^0 for each $s \in S$ so that $\{(\widehat{x}, \widehat{y}_s)\} \in V_s^0$. Observe also that this initialization corresponds to a technique for computing a feasible solution to the original problem (4.1), which is independently useful for obtaining an *upper bound* on ζ^{SIP}.

For the computational experiments, we run the following initialization to obtain $(V_s^0)_{s\in S}$ and $(x_s^0, y_s^0)_{s\in S}$ that are input into Algorithm 4.3:

Algorithm 4.4 Initialization step for FW-PH

```
Precondition: Problem (4.1) has relatively complete recourse
function FW-PH-Initialization(λ^0)
    for s ∈ S do
        (x_s^0, y_s^0) ← argmin_{x,y} {(c + λ_s^0)^⊤ x + q_s^⊤ y | (x, y) ∈ K_s}
        V_s^0 ← {(x_s^0, y_s^0)}
        if s ≠ 1 then
            ȳ_s ← argmin_y {q_s^⊤ y | (x_1^0, y) ∈ K_s}
            V_s^0 ← V_s^0 ∪ {(x_1^0, ȳ_s)}
        end if
    end for
    return (V_s^0, (x_s^0, y_s^0))_{s∈S}
end function
```

If problem (4.1) does not have relatively complete recourse, then any means to compute a feasible solution to (4.1) may be employed to initialise each V_s^0, $s \in S$, in a way to satisfy (4.12).

Two sets of Algorithm 4.3 experiments correspond to variants considering $\alpha = 0$ and $\alpha = 1$. Computations were performed on eight problems: the CAP instance 101 and 102 with the first 250 scenarios (CAP-101-250 and CAP-102-250), the DCAP instance DCAP233_500 and DCAP243_500 with 500 scenarios, and the SSLP instances 5-25-50, 5-25-100, 10-50-100, and 15-45-15 (which encode

the number of servers, clients, and scenarios, respectively). For a more detailed description of the structure of these instances see Section 2.4.2. For each problem, computations were performed for different penalty values $\rho > 0$. The penalty values used in the experiments for the SSLP-5-25-50 instance were chosen to include those penalties that are tested in a computational experiment with PH whose results are depicted in Figure 2 of [45]. For the other problem instances, the set of penalty values ρ tested is chosen to capture a reasonably wide range of performance potential for both PH and FW-PH. All computational experiments were allowed to run for a maximum of two hours in wall clock time.

4.3.2 Numerical Results

	Gap(%)			# Iterations			Time		
		FW-PH			FW-PH			FW-PH	
ρ	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$
20	0.05	0.17	0.09	509	398	445	T	T	T
100	0.01	0.00	0.00	178	446	440	1975.91	T	T
500	0.07	0.00	0.00	540	92	93	T	931.84	986.83
1000	0.15	0.00	0.00	544	127	130	T	1345.04	1425.90
2500	0.34	0.00	0.00	581	259	274	T	3087.30	3276.03
5000	0.66	0.00	0.00	33	473	468	293.03	T	T
7500	0.99	0.00	0.00	28	18	19	225.66	138.80	170.14
15000	1.59	0.00	0.00	545	28	33	T	246.65	283.53

(a) CAP-101-250; absolute percentage gap based on the known optimal value 733827.32

	Gap (%)			# Iterations			Time		
		FW-PH			FW-PH			FW-PH	
ρ	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$
20	0.47	0.46	0.49	422	426	412	T	T	T
100	0.01	0.00	0.00	219	408	405	3343.29	T	T
500	0.08	0.00	0.00	48	46	46	757.09	524.11	540.72
1000	0.13	0.00	0.00	24	25	24	297.34	271.72	286.68
2500	0.29	0.00	0.00	13	16	16	151.72	160.46	171.43
5000	0.61	0.00	0.00	14	18	18	156.90	170.86	188.87
7500	0.93	0.00	0.00	17	22	23	187.08	224.37	237.81
15000	1.91	0.00	0.00	22	39	42	228.26	450.64	436.41

(b) CAP-102-250; absolute percentage gap based on the known optimal value 788996.97

Table 4.1: FW-PH result summary for CAP problem instances: dual bounds.

Tables 4.1–4.3 provide a summary indicating the quality of the Lagrangian bounds ϕ computed at the end of each experiment for the eight problems with varying penalty parameter ρ. In each

	Gap (%)			# Iterations			Time		
		FW-PH			FW-PH			FW-PH	
ρ	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$
2	0.13	0.12	0.12	2234	576	570	T	T	T
5	0.22	0.09	0.09	2367	561	559	T	T	T
10	0.23	0.07	0.08	2583	592	573	T	T	T
20	0.35	0.07	0.07	2539	572	567	T	T	T
50	1.25	0.06	0.06	2721	578	580	T	T	T
100	1.29	0.06	0.06	2755	428	438	T	4016.29	4492.36
200	2.58	0.06	0.06	2667	256	262	T	1707.97	1848.49
500	2.58	0.07	0.07	2839	244	246	T	1799.88	1569.58

(a) DCAP-233-500; absolute percentage gap based on the best known upper bound 1737.73

	Gap (%)			# Iterations			Time		
		FW-PH			FW-PH			FW-PH	
ρ	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$
2	0.14	0.18	0.18	1710	558	577	T	T	T
5	0.20	0.13	0.13	2108	570	562	T	T	T
10	0.29	0.11	0.11	2110	562	559	T	T	T
20	0.52	0.10	0.10	2233	570	577	T	T	T
50	0.70	0.10	0.10	2355	578	579	T	T	T
100	1.32	0.09	0.09	2504	393	395	T	3744.33	3849.53
200	1.40	0.10	0.09	2568	244	261	T	1866.03	1854.85
500	2.11	0.10	0.10	2486	180	165	T	983.41	884.66

(b) DCAP-243-500; absolute percentage gap based on the known optimal value 2167.51

Table 4.2: FW-PH result summary for DCAP problem instances: dual bounds.

of these tables, the first column lists the values of the penalty parameter ρ, while the following are presented for PH and FW-PH (for both $\alpha = 0$ and $\alpha = 1$) computations in the remaining columns: 1) the absolute percentage gap $\left|\frac{\zeta^* - \phi}{\zeta^*}\right| * 100\%$ between the computed Lagrangian bound ϕ and some reference value ζ^* that is either a known optimal value for the problem, or a known best upper bound thereof (column "Percentage Gap"); 2) the total number of dual updates ("# Iterations"); and 3) the indication of whether the algorithm terminated due to the time limit, indicated by letter "T", or the satisfaction of the convergence criterion $\sqrt{\sum_{s \in S} p_s \left\| x_s^k - \bar{x}^{k-1} \right\|_2^2} < \epsilon$, indicated by the time elapsed in seconds (column "Time").

The following observations can be made from the results presented in Tables 4.1–4.3. First, for small values of the penalty ρ, there is no clear preference between the bounds ϕ generated by PH and FW-PH. However, for higher penalties, the bounds ϕ obtained by FW-PH are consistently of better quality (i.e., higher) than those obtained by PH, regardless of the variant used (i.e. $\alpha = 0$ or

	Gap (%)			# Iterations			Time		
		FW-PH			FW-PH			FW-PH	
ρ	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$
1	0.30	0.00	0.00	105	115	116	225.80	150.63	151.52
2	0.73	0.00	0.00	51	56	56	107.85	71.56	72.07
5	0.91	0.00	0.00	25	26	27	51.77	33.43	34.88
15	3.15	0.00	0.00	12	16	17	22.00	20.59	21.95
30	6.45	0.00	0.00	12	18	18	18.44	23.29	24.00
50	9.48	0.00	0.00	18	25	26	21.00	34.37	37.89
100	9.48	0.00	0.00	8	45	45	7.95	62.20	67.77

(a) SSLP-5-25-50; absolute percentage gap based on the known optimal value -121.60

	Gap (%)			# Iterations			Time		
		FW-PH			FW-PH			FW-PH	
ρ	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$
1	0.16	0.00	0.00	82	97	90	385.08	266.05	248.92
2	0.45	0.00	0.00	42	43	44	196.76	119.57	121.30
5	1.06	0.00	0.00	18	21	22	83.66	58.29	61.62
15	2.96	0.00	0.00	13	15	16	51.40	42.50	46.35
30	6.21	0.00	0.00	19	24	23	56.58	70.47	64.26
50	7.91	0.00	0.00	3123	38	36	T	113.21	107.54
100	7.91	0.00	0.00	27	74	70	44.60	223.73	216.66

(b) SSLP-5-25-100; absolute percentage gap based on the known optimal value -127.37

	Gap (%)			# Iterations			Time		
		FW-PH			FW-PH			FW-PH	
ρ	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$
1	0.57	0.22	0.22	130	234	234	T	T	T
2	0.63	0.03	0.03	131	226	227	T	T	T
5	1.00	0.00	0.00	104	218	219	4885.74	T	T
15	2.92	0.00	0.00	33	45	118	1012.11	1463.75	3949.99
30	4.63	0.00	0.00	18	21	22	413.28	618.52	619.85
50	4.63	0.00	0.00	11	26	27	202.47	759.83	756.59
100	4.63	0.00	0.00	9	43	45	106.76	1302.04	1271.27

(c) SSLP-10-50-100; percentage gap based on the known optimal value -354.19

	Gap (%)			# Iterations			Time		
		FW-PH			FW-PH			FW-PH	
ρ	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$	PH	$\alpha = 0$	$\alpha = 1$
1	2.85	2.15	2.17	224	304	300	T	T	T
2	2.21	1.00	1.00	193	272	272	T	T	T
5	1.21	0.01	0.03	181	180	178	7021.35	T	T
15	4.13	0.00	0.00	421	84	86	T	5022.34	4986.36
30	7.89	0.00	0.00	35	66	68	424.76	1873.24	1866.31
50	7.89	0.00	0.00	23	67	65	257.40	992.90	1020.19
100	7.89	0.00	0.00	6	69	62	32.25	562.65	428.18

(d) SSLP-15-45-15; percentage gap based on the known optimal value -253.60

Table 4.3: FW-PH result summary for SSLP problem instances : dual bounds.

$\alpha = 1$). This tendency is clearly illustrated by the results of the DCAP experiments in Table 4.2, where the absolute percentage gap of the Lagrangian lower bound with the known optimal value was 0.06% with $\rho = 200$ for FW-PH ($\alpha = 0$), while it was 2.58% for the same value of ρ for PH. This improvement is consistently observed for the other problems and the other values of ρ that are not too close to zero. Also, FW-PH did not terminate with suboptimal convergence or display cycling behavior for any of the penalty values ρ in any of the problems considered. In the case of the SSLP-5-25-50 problem instance, the FW-PH algorithm terminated due to convergence in all cases. The percentage gaps suggest that the convergence for PH was suboptimal, while it was optimal for FW-PH. Moreover, it is possible to see from these tables that the quality of the bounds ϕ obtained using FW-PH were not as sensitive to the value of the penalty parameter ρ as obtained using PH.

The FW-PH ($\alpha = 0$) versus PH convergence profiles for the experiments performed are given in Figures 4.1–4.8, in which we provide plots of wall time versus Lagrangian bound values based on profiling of varying penalty. Since the $\alpha = 1$ variant of FW-PH behaved very similarly to the $\alpha = 0$ variant in all cases, the plots of the $\alpha = 1$ variant have been omitted for brevity. The time scales for each plot have been chosen to highlight the most interesting features. The trend of the Lagrangian bounds is depicted with solid lines for FW-PH with $\alpha = 0$ and with dashed lines for PH.

As seen in the plots of Figures 4.1–4.8, the Lagrangian bounds ϕ generated with PH tend to converge suboptimally, often displaying cycling, for large penalty values. In terms of the quality of the bounds obtained, while there is no clear winner when low penalty ρ values are used, for large penalties, the quality of the bounds ϕ generated with FW-PH is consistently better than for the bounds generated with PH, regardless of the α value. This last observation is significant because the effective use of large penalty values ρ in methods based on augmented Lagrangian relaxation tends to yield the most rapid early iteration improvement in the Lagrangian bound; this point is most clearly illustrated in Figure 4.5, which depicts the results of the SSLP-5-25-50 experiments.

Although finding primal feasible solutions is not the stated purpose of FW-PH, in practice the algorithm can be easily modified to generate these solutions by adding one of the following heuristics:

- Record the vertices generated in the final SDM step; these points are integer feasible.
- Perform a single Progressive Hedging-style iteration at the end of FW-PH to generate an

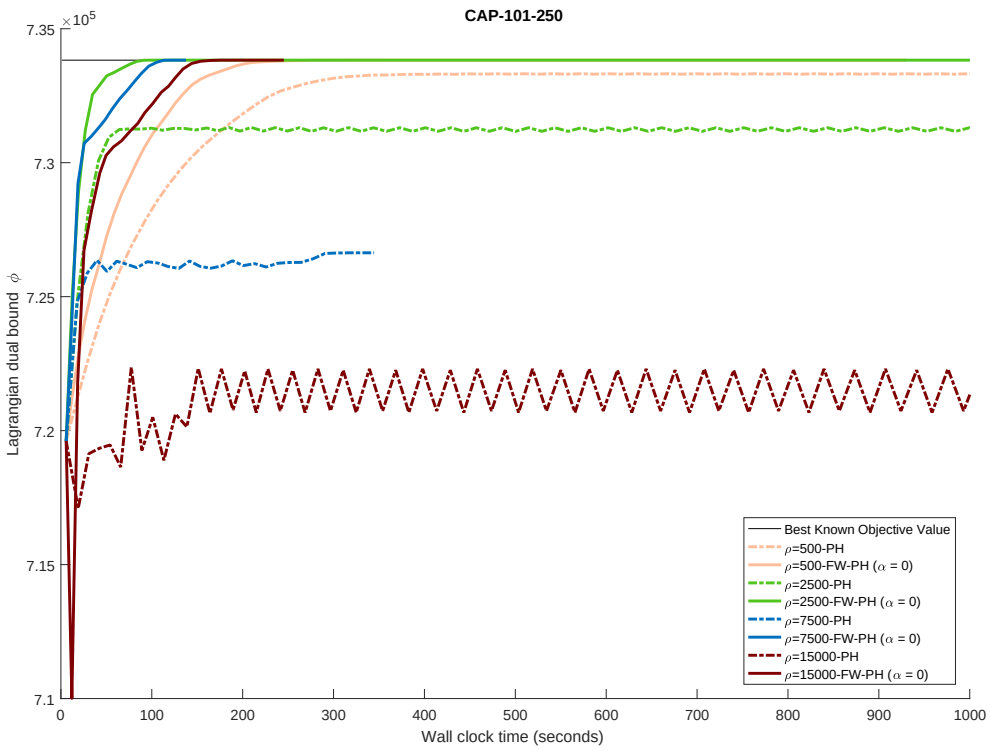


Figure 4.1: PH and FW-PH convergence profiles for CAP-101-250 ($\alpha = 0$)

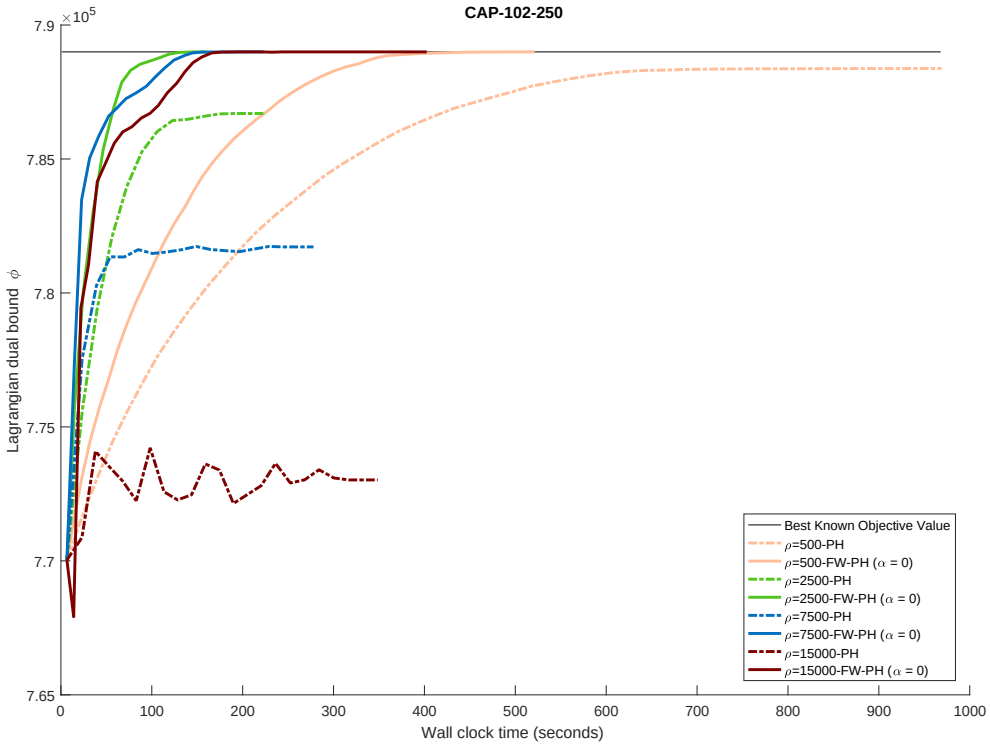


Figure 4.2: PH and FW-PH convergence profiles for CAP-101-250 ($\alpha = 0$)

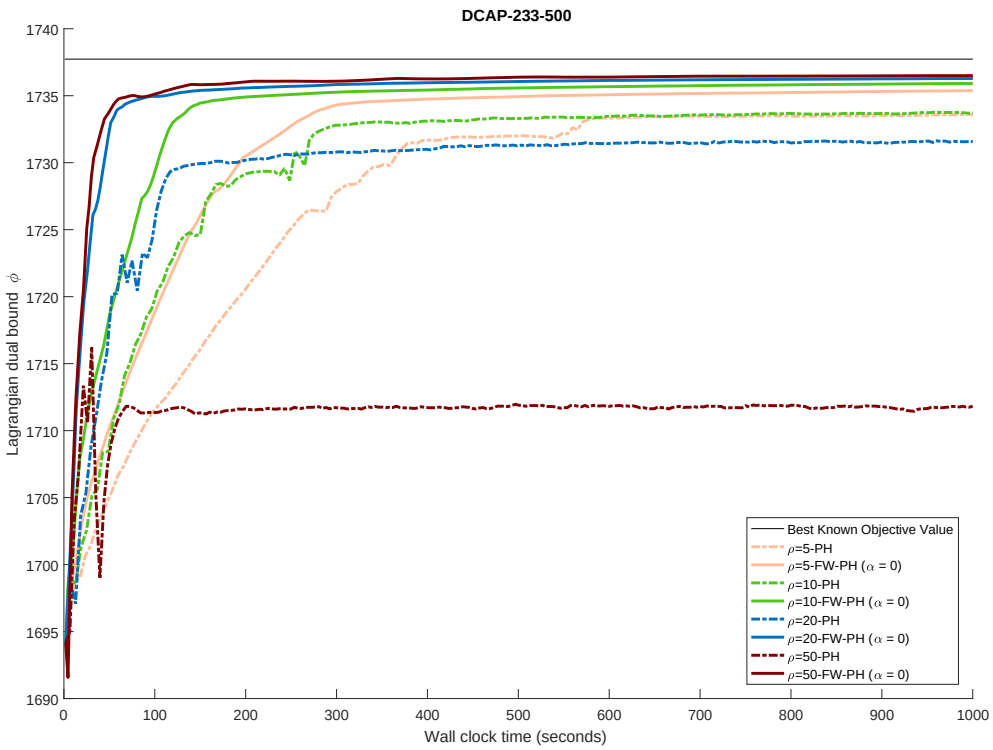


Figure 4.3: PH and FW-PH convergence profiles for DCAP-233-500 ($\alpha = 0$)

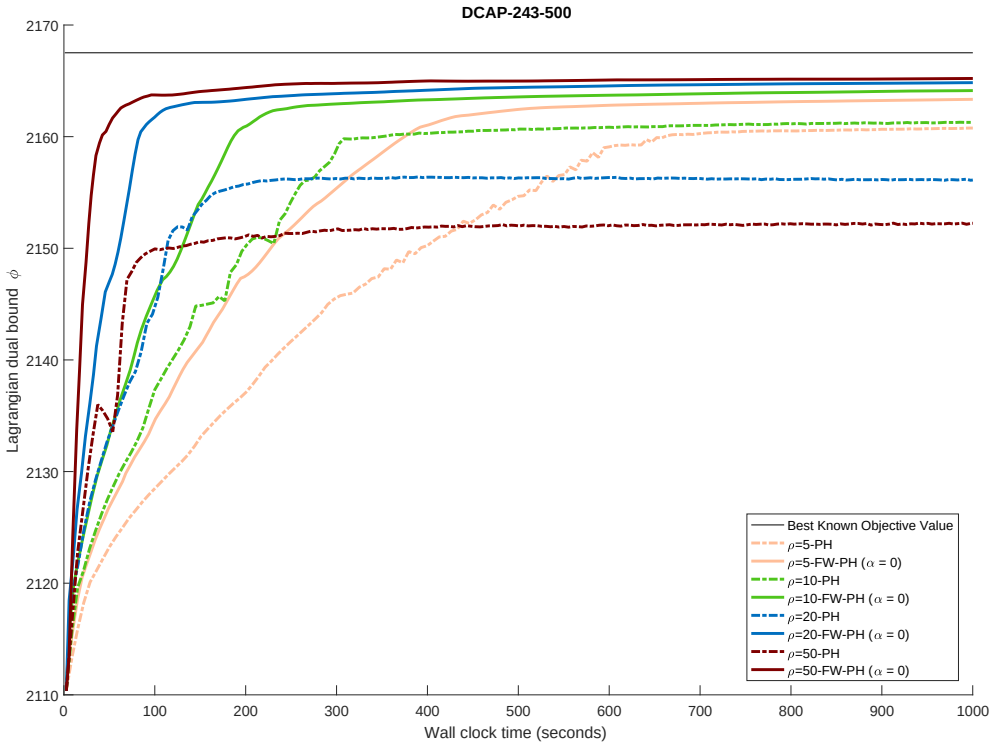


Figure 4.4: PH and FW-PH convergence profiles for DCAP-233-500 ($\alpha = 0$)

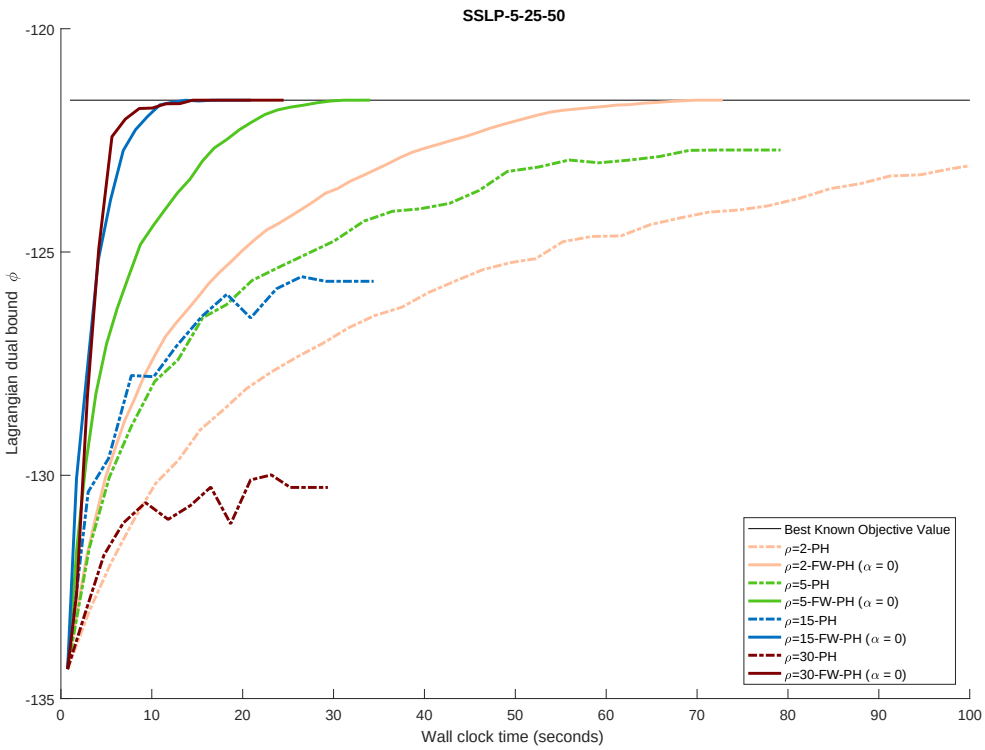


Figure 4.5: PH and FW-PH convergence profiles for SSLP-5-25-50 ($\alpha = 0$)

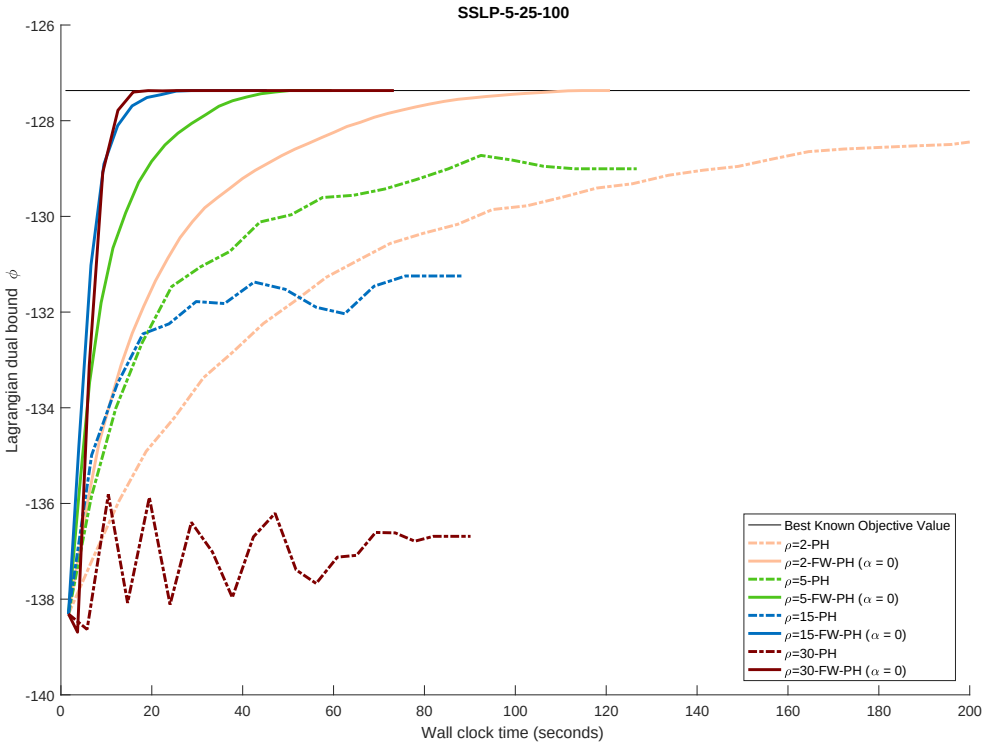


Figure 4.6: PH and FW-PH convergence profiles for SSLP-5-25-50 ($\alpha = 0$)

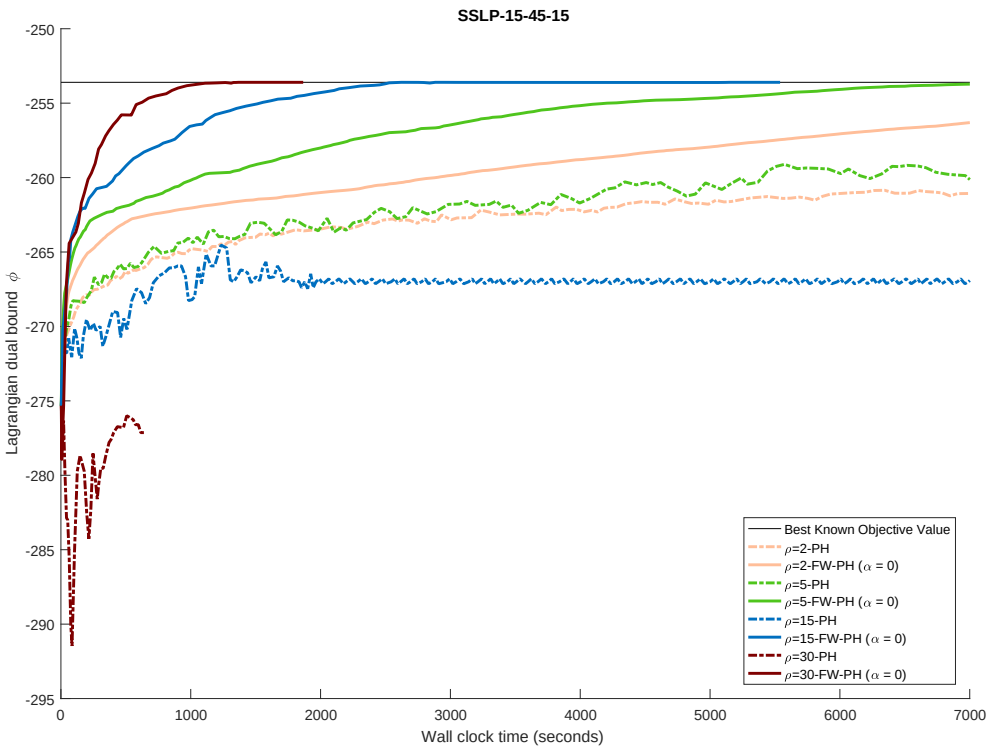


Figure 4.7: PH and FW-PH convergence profiles for for SSLP-10-50-100 ($\alpha = 0$)

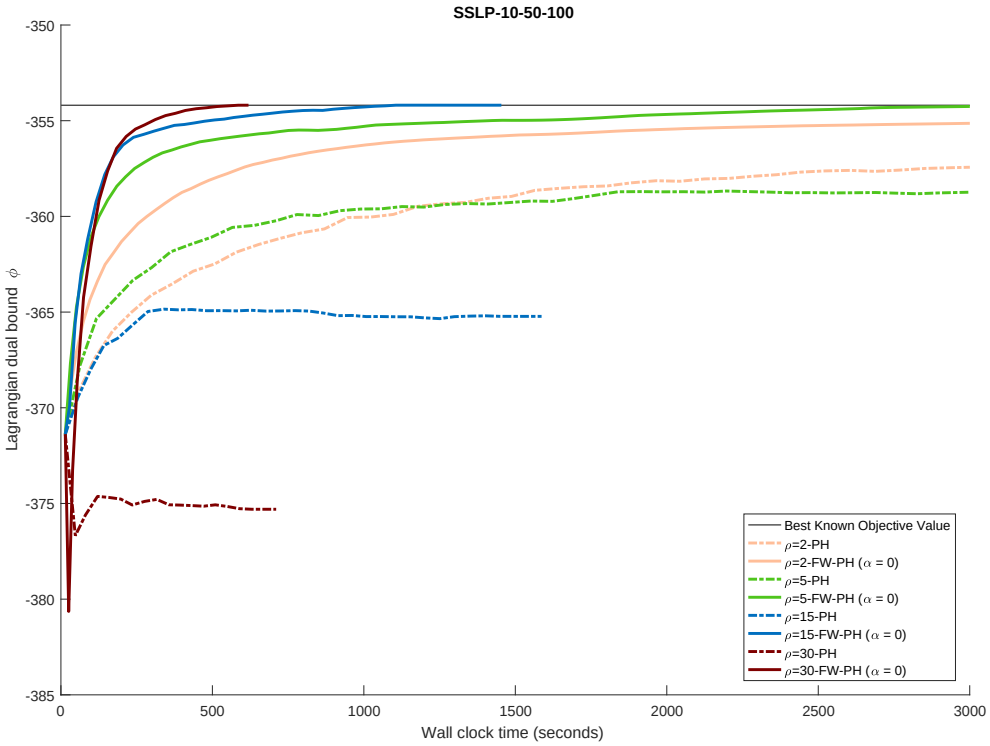


Figure 4.8: PH and FW-PH convergence profiles for for SSLP-10-50-100 ($\alpha = 0$)

integer feasible point.

These heuristics are not guaranteed to find a high-quality feasible solution. However, small-scale testing indicated that both approaches are promising and worthy of future examination.

Remark 4.4 *It may seem counter-intuitive that in some cases (particularly when applied to the DCAP class of instances) the PH algorithm completes more iterations than FW-PH, despite the fact that PH is solving a more difficult class of subproblem than FW-PH at each iteration (QMIP vs MIP). However, in these cases it is generally the case that the PH algorithm eventually cycles at some distance from the optimal dual bound (and therefore is solving potentially easy QMIPs in which only the cycling variables are contributing to complexity), whereas the FW-PH algorithm asymptotically converges to the optimal dual bound (and therefore is solving potentially difficult and ill-conditioned MIPs). Additional experimental work would be required to investigate this further.*

4.4 Conclusions

This chapter has presented an alternative approach to compute Lagrangian dual bounds for SIPs that combines ideas from Progressive Hedging (PH) and the Frank-Wolfe (FW) based methods. This approach is motivated by the weaknesses of previous methods for retrieving dual bounds from PH and PH-like algorithms; in general these previous methods require additional computational effort and the dual bounds they retrieve are not necessarily of high quality. In general the direct application of PH to SIPs is handicapped by the non-convexity of the SIPs.

By applying Progressive Hedging to the primal characterisation of the Lagrangian dual problem, which is itself a convex stochastic program, the issue of non-convexity is side-stepped. This approach is specifically directed towards finding Lagrangian dual bounds, as opposed to finding the primal optimal solution. The primary difficulty with this approach is that the PH update subproblems can no longer be computed directly, since the convex hull relaxation appearing in the primal characterisation is not explicitly defined. Section 4.2.3 shows that a FW-like method such as the simplicial decomposition method can be used to solve the update subproblems. However, using SDM to solve these subproblems exactly at each PH iteration is likely to be very slow.

A more sophisticated algorithm, designated as the FW-PH algorithm, is proposed in Section 4.2.4. FW-PH has good convergence properties under general assumptions on how the Lagrangian

bounds are computed and on the number of SDM iterations used at each step. Furthermore, under mild assumptions on the initialization of the algorithm, FW-PH convergence only requires the solution of a MILP subproblem and a continuous convex quadratic subproblem for each iteration and each scenario. FW-PH is versatile enough to handle a wide range of SIPs with integrality restrictions in any stage, while providing rapid improvement in the Lagrangian bound in the early iterations that is consistent across a wide range of penalty parameter values.

The numerical results for FW-PH, presented in Section 4.3, are encouraging as they suggest that the proposed FW-PH method applied to SIP problems usually outperforms the traditional PH method with respect to how quickly the quality of the generated Lagrangian bounds improves. This is especially true with the use of larger penalty values. For all problems considered and for all but the smallest penalties considered, the FW-PH method displayed better performance over PH in terms of the quality of the final Lagrangian bounds at the end of the allotted wall clock time.

The improved performance of FW-PH over PH for large penalties is significant because it is the effective use of large penalties enabled by FW-PH that yields the most rapid initial dual improvement. This last feature of FW-PH would be most helpful in its use within a branch-and-bound or branch-and-cut framework for providing strong lower bounds (in the case of minimisation). In addition to being another means to compute Lagrangian bounds, PH would still have a role in such frameworks as a heuristic for computing a primal feasible solution to the SIP, thus providing (in the case of minimisation) an upper bound on the optimal value.

Several opportunities to extend the FW-PH algorithm and its implementation are immediately apparent. The approach taken in this chapter of performing exactly one SDM iteration per PH iteration is unnecessarily limiting; a more sophisticated approach would vary the frequency of SDM updates based on their projected utility. The assumption that every extreme point is retained in the inner approximation of the feasible region is necessary for the convergence proof given in this chapter, but is likely unnecessary for convergence in practice. Furthermore, while we have observed that the structure of the FW-PH algorithm is amenable to parallel computation, the computational results presented in Section 4.3 do not explore the benefits of doing so. These potential improvements are pursued in Chapter 5.

Chapter 5

Simplicial Decomposition-based Augmented Lagrangian Method

5.1 Introduction and Background

5.1.1 Problem Formulation

In this chapter we will consider the problem of calculating high-quality dual bounds for a more general class of MIP having the form

$$\zeta^* := \min_{x,z} \{f(x) : Qx = z, x \in X, z \in Z\}, \tag{5.1}$$

where f is convex and continuously differentiable, $Q \in \mathbb{R}^{q \times n}$ is a block-diagonal matrix determining linear constraints $Qx = z$, $X \subset \mathbb{R}^n$ is a closed and bounded set, and $Z \subset \mathbb{R}^q$ is a linear subspace. The vector $x \in X$ of decision variables is derived from the original decisions associated with a problem, while the vector $z \in Z$ of auxiliary variables are introduced to effect a decomposable structure in (5.1). In particular, the decomposable structure takes the form: 1) $X = \prod_{i=1}^m X_i$ with $X_i \subset \mathbb{R}^{n_i}$ closed and bounded and $\sum_{i=1}^m n_i = n$; 2) $f(x) = \sum_{i=1}^m f_i(x_i)$ where $f_i : \mathbb{R}^{n_i} \mapsto \mathbb{R}$ are convex and differentiable for $i = 1, \ldots, m$; 3) Q has block diagonal structure with block diagonal components denoted as $Q_i \in \mathbb{R}^{q_i \times n_i}$, $i = 1, \ldots, m$ where $\sum_{i=1}^m q_i = q$, so that after setting $z = (z_i)_{i=1,\ldots,m}$, where for each $i = 1, \ldots, m$, $z_i \in \mathbb{R}^{q_i}$, we may write $Qx = z$ as $Q_i x_i = z_i$, $i = 1, \ldots, m$. This decomposable structure is implicitly present throughout the chapter, although explicit referral to it is typically avoided where it is not needed.

Assumption 5.1 *Problem* (5.1) *is feasible with finite optimal value.*

The structure of (5.1) is sufficiently general to include two-stage SIP problems. In particular, (5.1) represents a two-stage SIP when the problem data of (5.1) is defined as follows:

- f is a linear function in x.
- $q_i = q_j$ and $n_i = n_j$ for all i and j in $1, \ldots, m$, and furthermore $n > q$. q_i is the number of first-stage variables, while $n_i - q_i$ is the number of second-stage variables.
- Construct Q_i by horizontal concatenation of the identity matrix of size q_i with the zero matrix of size $q_i \times (q_i - n_i)$. This links the first-stage variables one-to-one with variables in z.
- X is restricted by the linear constraints of the SIP on first- and second-stage variables.
- $Z = \{(z_1, \ldots, z_{q_1}, z_{q_1+1}, \ldots, z_{m \times q_1}) \mid z_j = z_{i \times q_1 + j} \quad \forall i = 1, \ldots, m-1, \quad \forall j \in 1, \ldots, q_1\}$. The definition of this linear subspace enforces the non-anticipativity constraint.

By a similar (albeit more complicated) construction process it can be shown that multi-stage SIP problems can be represented by (5.1) as well. Stochastic programs in which f is convex but non-linear, or in which X is a general compact (not necessarily convex) set, can also be represented by (5.1).

We develop a solution approach to solving the following relaxation of (5.1),

$$\zeta^{CLD} := \min_{x,z} \{f(x) : Qx = z, x \in \operatorname{conv}(X), z \in Z\} \tag{5.2}$$

and its Lagrangian dual problem due to the relaxation of $Qx = z$,

$$\zeta^{CLD} = \sup_{\omega} \phi^C(\omega), \tag{5.3}$$

which is based on the dual function

$$\phi^C(\omega) := \min_{x} \left\{ f(x) + \omega^\top (Qx - z) : x \in \operatorname{conv}(X),\ z \in Z \right\}. \tag{5.4}$$

When f is linear, then $\phi^C(\omega) = \phi(\omega)$ where

$$\phi(\omega) := \min_{x,z} \left\{ f(x) + \omega^\top (Qx - z), x \in X, z \in Z \right\}. \tag{5.5}$$

(That is, when f is linear, the role of X and $\operatorname{conv}(X)$ are interchangeable.) Consequently, when f is linear, $\zeta^{CLD} = \zeta^{LD} := \sup_\omega \phi(\omega)$. However, when f is nonlinear, then in general, $\zeta^{CLD} \leqslant \zeta^{LD}$.

Strict inequality is demonstrated with the following example. Let $f : \mathbb{R}^2 \mapsto \mathbb{R}$ be defined by $f(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$, $X = \{0,1\} \times \{0,1\}$, and let $Qx = z$ be defined to model the constraints $x_1 - z_1 = 0$ and $x_2 - z_2 = 0$ where $Z = \{(z_1, z_2) : z_1 = z_2\} \subset \mathbb{R}^2$. We see trivially that $\zeta^{CLD} = 0$, which is verified with the saddle point $x_1^* = x_2^* = z_1^* = z_2^* = 0.5$ and $\omega^* = (0,0)$. However, $\zeta^{LD} = 0.5$, which is verified with either of the saddle points $x_1^* = x_2^* = z_1^* = z_2^* = 0$ and $\omega^* = (0,0)$, or $x_1^* = x_2^* = z_1^* = z_2^* = 1$ and $\omega^* = (0,0)$. Thus, $\zeta^{CLD} < \zeta^{LD}$.

Given that X is compact and f is continuous, in order for $-\infty < \phi^C(\omega)$ to hold, it is necessary and sufficient that the dual feasibility assumption

$$\omega \in Z^\perp := \left\{ \upsilon \in \mathbb{R}^q : \upsilon^\top z = 0 \text{ for all } z \in Z \right\} \tag{5.6}$$

is maintained either by assumption or by construction. Under condition (5.6), the z term in definition (5.4) vanishes, and we may compute

$$\phi^C(\omega) = \min_x \left\{ f(x) + \omega^\top Qx : x \in \operatorname{conv}(X) \right\}.$$

Consequently, ϕ^C becomes separable as

$$\phi^C(\omega) = \sum_{i=1}^m \phi_i^C(\omega_i),$$

where $\phi_i^C(\omega_i) := \min_x \left\{ f_i(x_i) + \omega_i^\top Q_i x_i : x_i \in \operatorname{conv}(X_i) \right\}$ and $\omega = (\omega_1, \ldots, \omega_m) \in \mathbb{R}^{q_1} \times \cdots \times \mathbb{R}^{q_m}$ has a block structure compatible with the block diagonal structure of Q.

Given that X is closed and bounded (thus so is $\operatorname{conv}(X)$), and (5.2) is assumed to be feasible, then in order to guarantee that the maximum in (5.3) is realised for some $\omega^* \in Z^\perp$, we assume a constraint qualification such as Slater's condition. In other words, we assume that there exists (x^*, z^*) such that $x^* \in \operatorname{int}(\operatorname{conv}(X))$ and $Qx^* = z^*$, where $\operatorname{int}(\cdot)$ returns the topological interior of the set argument. If $\operatorname{conv}(X)$ is polyhedral, then even this Slater's condition is not required.

5.1.2 Method Overview

The method developed in this chapter is an iterative solution approach to solving problem (5.3) subject to the following challenges:

Implementability: The set X is not convex (for example, it may have mixed-integer constraints as part of its definition). Consequently the augmented Lagrangian method is not supported by the theory of proximal point methods.

Efficiency of parallelisation: The solution approach should be amenable to efficient parallel computation, in the sense of maximising the computational work that can be parallelised, the memory usage that can be distributed, and minimising the amount of parallel communication.

Our method is theoretically applicable to calculating Lagrangian dual bounds of any problem which can be expressed in the form of (5.1) and satisfies Assumption 5.1. However, to implement the method in practice we will need to have an solver available to evaluate its subproblems.

Assumption 5.2 *The gradient of f is known for all $x \in X$. Furthermore, a solver is available for the following problems:*

- *Minimise an affine function over X.*
- *Minimise the sum of f and a quadratic function over the closed convex hull of a nonempty subset of X.*

Remark 5.1 *In the case where* (5.1) *represents a SIP, Assumption 5.2 is trivially satisfied: f is an affine function and X is defined by linear constraints and integrality constraints (so we can minimise an affine function over X with a standard MILP solver).*

Throughout this chapter we will assume that the problem of interest (5.1) satisfies Assumptions 5.1 and 5.2.

For the Lagrangian dual problem (5.3), we note that the objective function ϕ is concave, even when f and X are not convex. We can apply a subgradient method (see e.g. [103]; in textbooks [12, 102]) for solving (5.3) in an efficiently parallelizable manner. Such an approach is proposed in [22]. However, it is preferable to make use of structural features of (5.3) that allow for smoothing or regularization, so that better convergence properties are realised. For this reason, we consider alternative developments based on proximal point methods that are modified to address both of the above two challenges.

5.1.3 Background

As a starting point, we consider the classical augmented Lagrangian method based on proximal point methods. The augmented Lagrangian (AL) method (also known as the method of multipliers) is

developed from proximal point methods, and references include [58, 94, 11, 12].

The AL method typically has favorable convergence properties as a dual solution approach for convex problems (linear convergence rate under certain assumptions, see [95, 11] and references cited therein). However, two issues arise: 1) the set X is not convex, and so current theories of convergence are not applicable; and 2) the primal subproblem associated with each iteration of the AL method is not separable due to the augmented Lagrange term, making efficient parallel implementations difficult to develop.

This chapter introduces modifications to the AL method that address both of these issues. In order to introduce computational tractability in light of the possible nonlinearity of f and the nonconvexity of X, the modified AL method solves an alternative dual problem that can provide a weaker dual bound than that provided by the value of (5.3). In the case when f is linear, the alternative dual problem is equivalent to (5.3). This matter is explained in more detail in Section 5.2.5. The method that results from these modifications is most naturally compared with the proximal bundle method. The proximal bundle method initially appeared in [74], and for a survey with history, see [29]. Use of inexact oracles for computing $\phi(\omega)$ and elements of the subdifferential set $\partial\phi(\omega)$ are studied in [29, 30, 52] and references therein.

In its dual form, the bundle method may be referred to as the stabilized column generation method [5] or the proximal simplicial decomposition method [14]. In implementation, the developed algorithm more closely resembles the latter dual form.

For parallelisation of the proximal bundle method, see [39] and [79]. The approach developed in this chapter is most naturally compared with [79], as both approaches address the manner in which the same continuous master problem is approximately solved. The approach of [39] uses a substantially different parallel computational paradigm based on subspace optimization. This approach, in which solution subspaces are assigned to processors based on periodically updated global state information, is not necessarily based on the problem's decomposable structure.

The proximal bundle method approach requires modification for efficient parallelisation. This matter is addressed in [79], where a solution to the continuous master problem is obtained by primal dual interior point methods that exploit the decomposable structure present in the augmented Lagrangian term. We provide and analyze an alternative approach based on the use of:

1. the simplicial decomposition method (SDM) [61, 110, 12, 13], which provides an alternative framework to the proximal bundle method to address the implementability of the proximal point method while allowing for the possibility that f is nonlinear; and

2. nonlinear block Gauss-Seidel (GS) method [59, 111, 49, 108, 20] to approximate the solutions to the continuous master problem.

For a more detailed description of the simplicial decomposition and block Gauss-Seidel methods, see Sections 3.1.2 and 6.3.1 respectively. Motivated by its constituent parts, the algorithm we develop is referred to as SDM-GS-ALM.

In an iteration of SDM-GS-ALM, the analog to the continuous master problem is not solved to (near) exactness; instead, approximate solutions based on possibly just one nonlinear block GS iteration are used. Due to the underlying need for convexification of the non-relaxed constraint set, implementability requires that the nonlinear block GS method must be integrated with the SDM so that optimal convergence of the resulting iterations can be established. In this way, a serious step condition similar to that used in proximal bundle methods is eventually satisfied after a finite number of such integrated SDM-GS iterations, and analogous dual optimal convergence of our approach is recovered even with the deviations from the proximal bundle method. In summary, we algorithmically integrate the AL method, the SDM, nonlinear block GS iterations, and the proximal bundle method serious step condition. A convergence analysis is also provided for SDM-GS-ALM. Such an integration allows for a considerable improvement in parallel efficiency with respect to maximising the computational work that can be parallelised, the memory usage that can be distributed, and minimising the amount of parallel communication.

Other methods developed in the past that are related to aspects of our contribution include the following. In terms of approximating within the AL method, we include reference to [31, 33], where the research goal of developing implementable approximation criteria is addressed. The separable augmented Lagrangian (SALA) method [51], which is an application of the alternating direction method of multipliers (ADMM) [48, 44, 21] with a form of resource allocation decomposition and incorporates separability into the AL method. Other approaches to introducing separability into the AL method include [23, 107]. Jacobi iterate approaches applied within either a proximal bundle method or an AL method framework are considered in [86, 101]; the accelerated distributed aug-

mented Lagrangian method (ADAL) developed in [23] is like a Jacobi-iterate analogue of ADMM with supporting convergence analysis. Other approaches to incorporating separability are found in the alternating linearization approaches [67, 76] and the predictor corrector proximal multiplier (PCPM) methods [24, 55]. All of these methods provide implementable mechanisms for approximating primal subproblem solutions and effecting parallelism in a setting where X is convex. However, they are not practically implementable in our setting where X is not convex and its convex hull $\text{conv}(X)$ is not given beforehand in a computationally useful closed-form description.

In papers such as [45, 38], ADMM is applied directly to the primal problem (5.1). In both works, it is acknowledged that ADMM is not theoretically supported in optimal convergence due to the lack of convexity of X. Nevertheless, [45] reports the potential for Lagrangian dual bounds to be recovered at each iteration of ADMM even though it is applied to (5.1). In [38], where ADMM is applied to nonconvex decentralised unit commitment problems, heuristic improvements to ADMM are introduced to address the lack of convexity due to the mixed-integer constraints. In contrast to both of these approaches, where ADMM is applied directly to the primal problem (5.1), the approach developed in this chapter and the preceding chapter both resemble ADMM but with application to a primal characterization of the dual problem. In these two approaches, the challenge of not having an explicit form for this primal characterization is addressed.

The FW-PH algorithm developed in the previous chapter is structureally similar to the SDM-GS-ALM algorithm developed in this chapter. While the algorithms differ only slightly in terms of functionality, there are substantial differences in the motivation and the convergence analysis. The convergence analysis of FW-PH interfaces with the convergence analysis for ADMM, which is most naturally developed in the context of the theory of maximal monotone operators and Douglas-Rachford splitting methods [32, 34], or as the proximal decomposition of the graph of a maximal monotone operator [82]. In contrast, the convergence analysis of SDM-GS-ALM naturally reflects its synthesis of SDM, the nonlinear block GS method, the proximal bundle method, and the AL method. The convergence analysis of SDM-GS-ALM follows under more general assumptions than that for FW-PH. In particular, the convergence analysis of SDM-GS-ALM allows for trimming of the inner approximations, and it does not require the warm-starting required by FW-PH. The most important difference in functionality is due to the influence of ideas from proximal bundle methods in SDM-GS-ALM, where updates of ω are taken conditionally at each iteration, while such updates

are taken unconditionally at each iteration of FW-PH. We shall see that these conditional updates help to mitigate performance problems that arise due to the seemingly inevitable use of suboptimal algorithm parameters.

5.2 Algorithm Design and Theory

5.2.1 Augmented Lagrangian Method

Algorithm 5.1 provides a general framework for an AL method with approximate subproblem solutions that uses the bundle method's serious step condition (SSC). This framework will be useful to guide the developments presented next, in which we discuss optimality conditions and convergence properties of the algorithm. The convergence proof of Algorithm 5.1 is based on the convergence proofs of the proximal bundle method such as found in Chapter 7 of [100]. In the following we denote the augmented Lagrangian (AL):

$$\begin{aligned} L_\rho(x, z, \omega) &:= f(x) + \omega^\top (Qx - z) + \frac{\rho}{2}\|Qx - z\|^2 \\ &= f(x) + \omega^\top Qx + \frac{\rho}{2}\|Qx - z\|^2 \end{aligned} \tag{5.7}$$

where the AL relaxes $Qx = z$ and with the second equality following from $\omega \in Z^\perp$ and $z \in Z$.

In the proximal bundle method, the dual function ϕ is approximated by a cutting plane model function that majorizes ϕ. Instead we use the following approximation $\widehat{\phi} : \mathbb{R}^q \times \mathbb{R}^n \times \mathbb{R}^q \mapsto \mathbb{R}$ of ϕ^C centered at (x^k, z^k), $k \geqslant 0$, to replace the cutting plane model:

$$\widehat{\phi}(\omega, x^k, z^k) := L_\rho(x^k, z^k, \omega) + \frac{\rho}{2}\left\|Qx^k - z^k\right\|_2^2.$$

The $\frac{\rho}{2}\left\|Qx^k - z^k\right\|_2^2$ term is a duplicate of a term inside the $L_\rho(x^k, z^k, \omega)$ expression; this is required to guarantee that, under appropriate conditions, $\widehat{\phi}$ can be used as an upper bound on ϕ^C (see Lemma 5.4).

The convex hull $\mathrm{conv}(X)$ is not known explicitly, and so ϕ^C cannot be evaluated directly. Consequently, we will also require a tractable minorization of ϕ^C. For $x^k \in \mathrm{conv}(X)$, $k \geqslant 0$, define $\check{\phi}(\omega, x^k)$ as follows:

$$\check{\phi}(\omega, x^k) := \min_x \left\{ f(x^k) + \nabla_x f(x^k)^\top (x - x^k) + \omega^\top Qx : x \in X \right\}. \tag{5.8}$$

To justify that for any $x^k \in \operatorname{conv}(X)$ this function minorizes ϕ^C, observe that $f(x) \geqslant f(x^k) + \nabla_x f(x^k)^\top (x - x^k)$ for all x; furthermore, since the objective term $f(x^k) + \nabla_x f(x^k)^\top (x - x^k) + \omega^\top Qx$ is linear in terms of x, its minimum with respect to x over $\operatorname{conv}(X)$ is identical to its minimum over X.

In Algorithm 5.1, a proximal bundle method-like serious step condition is used in Line 9 that makes use of $\widehat{\phi}$ and $\check{\phi}$ in place of the cutting plane model and ϕ, respectively. Proposition 5.9 will demonstrate that any sequence of solutions converging to the argmin in Line 5 must eventually satisfy this serious step condition, unless dual optimality (i.e. the Line 8 condition) has already been achieved. The Line 6 condition is an optimality condition for z which we enforce for any accepted approximate solution to the Line 5 subproblem.

The inputs f, Q, X, and Z specify the data associated with problem (5.1); $\rho > 0$ is the AL term coefficient; ω^0 is an initial dual solution; $\gamma \in (0, 1)$ is the parameter of the serious step condition of Line 9; and $\epsilon > 0$ is a tolerance for termination. Algorithm 5.1 will be given a specific implementation in the form of SDM-GS-ALM in Section 5.2.5.

Algorithm 5.1 A general approximated ALM using a bundle method SSC.

1: **Preconditions:** $\omega^1 \in Z^\perp$, $\gamma \in (0, 1)$.
2: **function** ApproxALM(f, Q, X, Z, ρ, ω^1, γ, ϵ, k_{max})
3: **for** $k = 1, 2, \ldots, k_{max}$ **do**
4: Solve approximately
5: $(x^k, z^k) \in \operatorname{argmin}_{x,z} \left\{ L_\rho(x, z, \omega^k) : x \in \operatorname{conv}(X), z \in Z \right\}$ such that
6: 1) $z^k \in \operatorname{argmin}_z \left\{ \left\| Qx^k - z \right\|_2^2 : z \in Z \right\}$ and
7: 2) either
8: $\widehat{\phi}(\omega^k, x^k, z^k) - \check{\phi}(\omega^k, x^{k-1}) \leqslant \epsilon$ or
9: $0 < \gamma \leqslant \frac{\check{\phi}\left(\omega^k + \rho(Qx^k - z^k), x^k\right) - \check{\phi}(\omega^k, x^{k-1})}{\widehat{\phi}(\omega^k, x^k, z^k) - \check{\phi}(\omega^k, x^{k-1})}$
10: **if** $\widehat{\phi}(\omega^k, x^k, z^k) - \check{\phi}(\omega^k, x^{k-1}) \leqslant \epsilon$ **then**
11: **return** (x^k, z^k, ω^k)
12: **else**
13: set $\omega^{k+1} \leftarrow \omega^k + \rho(Qx^k - z^k)$
14: **end if**
15: **end for**
16: **return** (x^k, z^k, ω^{k+1})
17: **end function**

Remark 5.2 *Under the assumption that* $\operatorname{conv}(X)$ *is not known beforehand by any characterization,*

direct evaluation of ϕ^C or any of its subgradients at any $\omega \in Z^\perp$ is not possible. This dual function is not used in the proximal bundle method and is only treated indirectly in the current development.

In addition to generating a sequence $\{\omega^k\}$ of dual solutions to (5.3), our algorithm will also generate a sequence of primal solutions $\{(x^k, z^k)\}$ to (5.2), and so reference to (5.2) will be useful. In applying the AL method to problem (5.2), the continuous master problem for fixed $\omega \in Z^\perp$ takes the form

$$\zeta_\rho^{AL}(\omega) := \min_{x,z} \{L_\rho(x, z, \omega), x \in \operatorname{conv}(X), z \in Z\}. \tag{5.9}$$

Lemma 5.3 *For any optimal solution ω^* to problem (5.3), we have $\zeta_\rho^{AL}(\omega^*) = \zeta^{CLD}$. Additionally, any optimal solution (x^*, z^*) to problem (5.9) with $\omega = \omega^*$ is also optimal for problem (5.2).*

Proof. We specialise developments in, e.g., Section 4 of [96] or Section 6.4.3 of [100]. In the following, we begin assuming $\overline{\omega} \in Z^\perp$ as an arbitrary fixed vector to show:

$$\max_{\omega \in Z^\perp} \left\{ (\omega - \overline{\omega})^\top Qx - \frac{1}{2\rho} \|\omega - \overline{\omega}\|_2^2 \right\} = \frac{\rho}{2} \min_{z \in Z} \|Qx - z\|_2^2. \tag{5.10}$$

By the uniqueness of the projection onto a subspace, $z \in \operatorname{argmin}_{z \in Z} \|Qx - z\|_2^2$ is the unique $z \in Z$ for which $(Qx - z) \in Z^\perp$. Moreover, the optimality condition for left-hand side problem in (5.10): $0 \in \partial_\omega \left\{ (\omega - \overline{\omega})^\top Qx - \frac{1}{2\rho} \|\omega - \overline{\omega}\|_2^2 + \delta_{Z^\perp}(\omega) \right\}$ dictate that $(\omega - \overline{\omega}) \in \rho(Qx + Z)$ (where we have used, for $\omega \in Z^\perp$, $\partial \delta_{Z^\perp}(\omega) = N_{Z^\perp}(\omega) = Z$, where $\delta_{Z^\perp}$ denotes the indicator function of a convex set $Z^\perp$ and $N_{Z^\perp}(\omega)$ the normal cone at ω). Hence $(\omega - \overline{\omega}) = \rho(Qx - z)$ for some $z \in Z$. Furthermore, $\omega - \overline{\omega} \in Z^\perp$ must hold, and so $z \in Z$ must be chosen so that $\rho(Qx - z) \in Z^\perp$ also. Consequently, from our first observation, this $z \in Z$ must be the unique solution of the right-hand side of (5.10). Evaluating the objective on the left-hand side of (5.10) (observing that we must have $\rho(Qx - z) \in Z^\perp$) establishes the claimed equality. We may compute:

$$\max_{\omega \in Z^\perp} \phi^C(\omega) - \frac{1}{2\rho} \|\omega - \overline{\omega}\|_2^2 \tag{5.11}$$

$$= \max_{\omega \in Z^\perp} \min_x \left\{ f(x) + \omega^\top Qx - \frac{1}{2\rho} \|\omega - \overline{\omega}\|_2^2 : x \in \operatorname{conv}(X) \right\}$$

$$= \min_x \left\{ \begin{array}{l} f(x) + \overline{\omega}^\top Qx \\ + \max_{\omega \in Z^\perp} \left\{ (\omega - \overline{\omega})^\top Qx - \frac{1}{2\rho} \|\omega - \overline{\omega}\|_2^2 \right\} : x \in \operatorname{conv}(X) \end{array} \right\} \tag{5.12}$$

$$= \min_x \left\{ f(x) + \overline{\omega}^\top Qx + \frac{\rho}{2} \min_{z \in Z} \left\{ \|Qx - z\|_2^2 \right\} : x \in \operatorname{conv}(X) \right\}$$

$$= \min_{x,z} \{L_\rho(x, z, \overline{\omega}), x \in \operatorname{conv}(X), z \in Z\}.$$

The switching of min and max in (5.12) is justified by the Sion min-max theorem [104] along with the convexity of f, $\text{conv}(X)$, Z and $\text{conv}(X)$ assumed compactness. In substituting $\overline{\omega} = \omega^*$, the value of the left-hand side maximisation problem (5.11) is clearly ζ^{CLD}, while the same substitution on the right-hand side (5.10) yields the value $\zeta_\rho^{AL}(\omega^*)$, from which we see that $\zeta^{CLD} = \zeta_\rho^{AL}(\omega^*)$. To prove the last claim, we note that $L_\rho(x^*, z^*, \omega^*) = \zeta^{CLD}$ implies that $\|Qx^* - z^*\|_2^2 = 0$. Otherwise, $\phi^C(\omega^*) < \zeta^{CLD}$, contradicting the dual optimality of ω^*. Thus, (x^*, z^*) is feasible and optimal for problem (5.2). ■

The approximation $\widehat{\phi}$ satisfies the following bounding relationship.

Lemma 5.4 *For each (x^k, z^k), $k \geqslant 0$, such that the z-optimality condition is satisfied:*

$$z^k \in \operatorname*{argmin}_z \left\{ \|Qx^k - z\|_2^2 : z \in Z \right\}, \tag{5.13}$$

we have for each $\omega \in Z^\perp$

$$\widehat{\phi}(\omega, x^k, z^k) \geqslant \phi^C\left(\omega + \rho(Qx^k - z^k)\right). \tag{5.14}$$

Proof. Via convexity of the term $\|Qx - z\|_2^2$ over $(x, z) \in \text{conv}(X) \times Z$, we may write the following inequalities that hold for $(x, z) \in \text{conv}(X) \times Z$ and a fixed $\omega \in Z^\perp$. Via the subgradient inequality:

$$\begin{aligned}
L_\rho(x, z, \omega) &\geqslant f(x) + \omega^\top Qx + \frac{\rho}{2}\|Qx^k - z^k\|_2^2 \\
&\qquad + \rho(Qx^k - z^k)^\top(Qx - z) - \rho(Qx^k - z^k)^\top(Qx^k - z^k) \\
&= f(x) + \omega^\top Qx - \frac{\rho}{2}\|Qx^k - z^k\|_2^2 + \rho(Qx^k - z^k)^\top(Qx - z) \\
\implies L_\rho(x, z, \omega) + \frac{\rho}{2}\|Qx^k - z^k\|_2^2 &\geqslant f(x) + \left[\omega + \rho(Qx^k - z^k)\right]^\top Qx \qquad (5.15) \\
&\geqslant \min_x \left\{ f(x) + \left(\omega + \rho(Qx^k - z^k)\right)^\top Qx : x \in \text{conv}(X) \right\}. \qquad (5.16)
\end{aligned}$$

Note that the term $-\rho(Qx^k - z^k)^\top z$ vanishes due to the optimality condition associated with (5.13). Inequality (5.14) follows from the inequalities (5.15)–(5.16) once the substitution $(x, z) = (x^k, z^k)$ and the definition of $\widehat{\phi}(\omega, x^k, z^k)$ are applied to the left-hand side of (5.15). ■

Observe that, due to the linearity of the objective function with respect to x in (5.8), the use of constraint sets X and $\text{conv}(X)$ are interchangeable, and so in evaluating $\check{\phi}$, an explicit description of $\text{conv}(X)$ is not required. Furthermore, from the definition of ϕ^C, the convexity of f over $\mathbb{R}^n$, and the interchangeability of X and $\text{conv}(X)$ in (5.8), it is clear that for all $x^k \in \mathbb{R}^n$, $k \geqslant 0$, we have

$\phi^C(\omega) \geqslant \check{\phi}(\omega, x^k)$. This is not only true in principle but also practically if one can provide a oracle that returns an extremal point of $\mathrm{conv}(X)$ when minimising a linear function over X. When the non-convexity is entirely due to the presence of integer restrictions on variables, MIP or MINLP solvers can provide such an oracle. Later in Section 5.2.5 we shall see that the class of problems that is amenable to the final implementable algorithm is dictated, in practice, by the user's ability to provide such an oracle. Moreover, when f is linear, we have $\phi^C(\omega) \equiv \check{\phi}(\omega, x^k)$ for all x^k, $k \geqslant 0$; the two functions collapse into the same function with the centering at x^k of the latter function now irrelevant.

The first important property of $(\omega, x) \mapsto \check{\phi}(\omega, x)$ is its continuity.

Lemma 5.5 *Let X be compact, and f be continuously differentiable. Then $(\overline{\omega}, \overline{x}) \mapsto \check{\phi}(\overline{\omega}, \overline{x})$ is continuous over $(\overline{\omega}, \overline{x}) \in Z^\perp \times \mathbb{R}^n$.*

Proof. From (5.8), compute

$$\begin{aligned}\check{\phi}(\overline{\omega}, \overline{x}) &= f(\overline{x}) - \nabla_x f(\overline{x})^\top \overline{x} + \min_x \left\{ \left[\nabla_x f(\overline{x}) + \overline{\omega}^\top Q \right] x + \delta_{\mathrm{conv}(X)}(x) \right\} \\ &= f(\overline{x}) - \nabla_x f(\overline{x})^\top \overline{x} - \delta^*_{\mathrm{conv}(X)} \left(- \left[\nabla_x f(\overline{x}) + \overline{\omega}^\top Q \right] \right).\end{aligned}$$

where $\delta_{\mathrm{conv}(X)}(x) := \begin{cases} 0, & \text{if } x \in \mathrm{conv}(X); \\ \infty, & \text{otherwise.} \end{cases}$ is the indicator function on the set $\mathrm{conv}(X)$ and $\delta^*_{\mathrm{conv}(X)}$ is the conjugate function [95] of $\delta_{\mathrm{conv}(X)}$. As $\mathrm{conv}(X)$ is convex and compact, we see that $\delta^*_{\mathrm{conv}(X)}(\cdot)$ has domain $\mathbb{R}^n$ and is thus continuous over $\mathbb{R}^n$ (e.g., Lemma 2.91 of [100]), yielding the intended conclusion. ■

The second property of $\check{\phi}$ is its limiting behavior as the solutions (x^k, z^k) approach certain critical values.

Lemma 5.6 *Let the sequence $\left\{(x^k, z^k)\right\} \subset \mathrm{conv}(X) \times Z$ satisfy the z-optimality condition* (5.13) *for each $k \geqslant 1$. If, for some fixed $\omega \in Z^\perp$, the sequence $\left\{(x^k, z^k)\right\}$ converges optimally in the sense that*

$$\lim_{k\to\infty} (x^k, z^k) = (x^*, z^*) \in \operatorname*{argmin}_{x,z} \left\{ L_\rho(x, z, \omega) : x \in \mathrm{conv}(X), z \in Z \right\},$$

then

$$\lim_{k\to\infty} \check{\phi}(\omega + \rho(Qx^k - z^k), x^k) = L_\rho(x^*, z^*, \omega) + \frac{\rho}{2} \left\| Qx^* - z^* \right\|_2^2. \tag{5.17}$$

Proof. We begin by writing the necessary (and sufficient) conditions associated with the optimality $(x^*, z^*) \in \operatorname{argmin}_{x,z} \{L_\rho(x, z, \omega) : x \in \operatorname{conv}(X), z \in Z\}$:

$$\begin{bmatrix} \nabla f(x^*) + [\omega + \rho(Qx^* - z^*)]^\top Q \\ -\rho(Qx^* - z^*) \end{bmatrix} \begin{bmatrix} x - x^* \\ z - z^* \end{bmatrix} \geqslant 0 \quad \text{for all } x \in \operatorname{conv}(X), z \in Z.$$

Since $z^k \in \operatorname{argmin}_z \left\{ \left\| Qx^k - z \right\|_2^2 : z \in Z \right\}$ for each $k \geqslant 1$, we have $Qx^k - z^k \in Z^\perp$, and so $Qx^* - z^* \in Z^\perp$ also. Thus, we can simplify the consideration of the above displayed necessary conditions to consider the x block only:

$$\left[\ \nabla f(x^*) + [\omega + \rho(Qx^* - z^*)]^\top Q \ \right]^\top \left[\ x - x^* \ \right] \geqslant 0 \quad \text{for all } x \in \operatorname{conv}(X),$$

which implies

$$\min_x \left\{ \left[\ \nabla f(x^*) + [\omega + \rho(Qx^* - z^*)]^\top Q \ \right]^\top \left[\ x - x^* \ \right] : x \in \operatorname{conv}(X) \right\} = 0.$$

In terms of $\check{\phi}(\omega + \rho(Qx^* - z^*), x^*)$, the above equality is re-written as:

$$\begin{aligned} \check{\phi}(\omega + \rho(Qx^* - z^*), x^*) &= f(x^*) + \omega^\top Qx^* + \rho \left\| Qx^* - z^* \right\|_2^2 \\ &= L_\rho(x^*, z^*, \omega) + \frac{\rho}{2} \left\| Qx^* - z^* \right\|_2^2, \end{aligned}$$

where the equality $(Qx^* - z^*)^\top z^* = 0$ is utilised. The continuity of $(\overline{\omega}, \overline{x}) \mapsto \check{\phi}(\overline{\omega}, \overline{x})$ established in Lemma 5.5 gives the desired conclusion. ■

We use Lemmas 5.5 and 5.6 to develop the proximal bundle method-like serious step condition (SSC) that makes use of $\widehat{\phi}$ and $\check{\phi}$ in place of the cutting plane model and ϕ, respectively. Defining $\widetilde{\omega}^k := \omega^k + \rho(Qx^k - z^k)$, consider the following modified serious step condition:

$$\gamma \leqslant \frac{\check{\phi}(\widetilde{\omega}^k, x^k) - \check{\phi}(\omega^k, x^{k-1})}{\widehat{\phi}(\omega^k, x^k, z^k) - \check{\phi}(\omega^k, x^{k-1})} \leqslant 1, \tag{5.18}$$

where $\gamma \in (0, 1)$ is the SSC parameter. The upper bound of (5.18) is satisfied automatically since $\widehat{\phi}(\omega^k, x^k, z^k) \geqslant \phi^C(\widetilde{\omega}^k) \geqslant \check{\phi}(\widetilde{\omega}^k, x^k)$ holds by Lemma 5.4 and the definition of $\check{\phi}$. However, the satisfaction of the lower bound is conditional on γ.

Remark 5.7 *Throughout this chapter, we shall always assume or construct z^k such that the z-optimality condition* (5.13) *is satisfied for each $k \geqslant 0$. Due to the necessary conditions of optimality associated with* (5.13) *and that Z is a linear subspace, we have $(Qx^k - z^k)^\top z = 0$ for all $z \in Z$. It*

immediately follows that if $\omega^k \in Z^\perp$*, then* $\widetilde{\omega}^k = \omega^k + \rho(Qx^k - z^k) \in Z^\perp$ *also. Thus, the satisfaction of the* z*-optimality condition* (5.13) *guides the generation of* $\{\omega^k\}$ *so that if* $\omega^0 \in Z^\perp$*, then* $\omega^k \in Z^\perp$ *is always maintained for each* $k \geqslant 1$*.*

Under certain circumstances, the denominator of the ratio displayed in (5.18) can be zero. The following lemma states that this never happens when ω^k is *not* dual optimal with respect to the dual problem (5.3).

Lemma 5.8 *For any* $\omega \in Z^\perp$ *that is not dual optimal with respect to the dual problem* (5.3) *and* $(x, z) \in \operatorname{conv}(X) \times Z$*, we have*

$$\widehat{\phi}(\omega, x, z) - \phi^C(\omega) > 0. \tag{5.19}$$

Consequently, at any iteration k*, the denominator of the ratio displayed in* (5.18) *cannot be zero when* ω^k *is not dual optimal.*

Proof. By the definition of $\widehat{\phi}$, we have

$$\widehat{\phi}(\omega, x, z) - \phi^C(\omega) \geqslant L_\rho(x^*, z^*, \omega) + \frac{\rho}{2}\|Qx - z\|_2^2 - \phi^C(\omega)$$

for all $\operatorname{conv}(X)$ and $z \in Z$, where

$$(x^*, z^*) \in \operatorname*{argmin}_{x,z} \{L_\rho(x, z, \omega) : x \in \operatorname{conv}(X), z \in Z\}.$$

(That is, we substitute $L_\rho(x, z, \omega)$ from the definition of $\widehat{\phi}$ with $L_\rho(x^*, z^*, \omega)$ to get the inequality.) Now $L_\rho(x^*, z^*, \omega) - \phi^C(\omega) > 0$ when ω is not dual optimal. Otherwise, if $L_\rho(x^*, z^*, \omega) = \phi^C(\omega)$, then $Qx^* = z^*$ must hold, and (x^*, z^*, ω) is a Lagrangian saddle point for problem (5.2) with respect to the Lagrangian relaxation of the constraint $Qx = z$. This contradicts the non-dual optimality of ω. Thus, the strict inequality (5.19) is established.

In the context of (5.18) at iteration k, noting that $\phi^C(\omega^k) \geqslant \check{\phi}(\omega^k, x^{k-1})$, we substitute $(x, z) = (x^k, z^k)$ and $\omega = \omega^k$ in the strict inequality (5.19) and so the denominator in (5.18) is positive when ω^k is not dual optimal. ■

From Lemma 5.6, we have the following result regarding the satisfaction of condition (5.18).

Proposition 5.9 *Let the sequence* $\{(x^k, z^k)\} \subset \operatorname{conv}(X) \times Z$ *satisfy*

$$z^k \in \operatorname*{argmin}_z \left\{\left\|Qx^k - z\right\|_2^2 : z \in Z\right\}$$

for each $k \geqslant 1$*. Furthermore, let* $\omega \in Z^{\perp}$ *and* $\omega \notin \operatorname{argmax}_{\omega} \phi(\omega)$*. If the sequence* $\left\{(x^k, z^k)\right\}$ *converges optimally in the sense that*

$$\lim_{k\to\infty} (x^k, z^k) = (x^*, z^*) \in \operatorname*{argmin}_{x,z} \left\{L_\rho(x, z, \omega) : x \in \operatorname{conv}(X), z \in Z\right\},$$

then condition (5.18) *must be satisfied after a finite number of iterations.*

Proof. For all $(x^k, z^k) \in \operatorname{conv}(X) \times Z$ with $z^k \in \operatorname{argmin}_z \left\{\left\|Qx^k - z\right\|_2^2\right\}$, we have

$$\begin{aligned}\widehat{\phi}(\omega, x^k, z^k) &= L_\rho(x^k, z^k, \omega) + \frac{\rho}{2}\left\|Qx^k - z^k\right\|_2^2 \\ &\geqslant \phi^C(\omega + \rho(Qx^k - z^k)) \geqslant \check{\phi}(\omega + \rho(Qx^k - z^k), x^k),\end{aligned}$$

where the first inequality follows from the definition of $\widehat{\phi}$ and Lemma 5.4, and the second inequality follows readily from the definition of $\check{\phi}$, the subgradient inequality and the interchangeability of X and $\operatorname{conv}(X)$. By the assumption that ω is not dual optimal, the denominator of (5.18) cannot be zero by Lemma 5.8. It follows from the convergence in (5.17) implied by Lemma 5.6 that the ratio in (5.18) must approach 1, and so condition (5.18) must be satisfied after a finite number of iterations. ■

Consequently, unless the current ω^k is already dual optimal, there cannot be an infinite number of null-steps when using condition (5.18). Recall that we use k to count only serious steps.

Proposition 5.10 *Assume that problem* (5.3) *has an optimal dual solution* ω^**, and that for each* $k \geqslant 1$*,* $\phi^C(\omega^k) < \phi^C(\omega^*)$*. Also, assume that* ρ *and* γ *may vary with each iteration, defined by sequences* $\{\rho_k\}$ *and* $\{\gamma_k\}$ *such that* $\rho_k > 0$ *and* $\gamma_k \in (0, 1)$*, bounded strictly away from zero for all* $k \geqslant 1$*, and* $\rho_k\left(\frac{1-\gamma_k}{\gamma_k}\right) = c > 0$ *for all* k*. If the sequence* $\left\{\omega^k\right\}$ *of dual updates is generated with Algorithm 5.1 with* $\epsilon = 0$ *and* $k_{max} = \infty$*, then* $\left\{\omega^k\right\}$ *converges, and* $\lim_{k\to\infty} \check{\phi}(\omega^k, x^{k-1}) = \zeta^{CLD}$ *(and consequently* $\lim_{k\to\infty} \phi^C(\omega^k) = \zeta^{CLD}$*). Furthermore,*

$$\lim_{k\to\infty} \widehat{\phi}(\omega^k, x^k, z^k) = \zeta^{CLD},$$

and all limit points $(\bar{x}, \bar{z})$ *of the sequence* $\left\{(x^k, z^k)\right\}$ *are optimal for problem* (5.2)*.*

Proof. Let ω^* be any dual optimal solution for problem (5.3). For each iteration $k \geqslant 1$, write the

following two relations:

$$\begin{aligned}\left\|\omega^{k+1}-\omega^*\right\|_2^2 &= \left\|\omega^k-\omega^*+\rho_k(Qx^k-z^k)\right\|_2^2\\ &= \left\|\omega^k-\omega^*\right\|_2^2+2\rho_k(Qx^k-z^k)^\top(\omega^k-\omega^*)+\rho_k^2\left\|Qx^k-z^k\right\|_2^2, \qquad (5.20)\\ \text{and}\quad \phi^C(\omega^*) &\leqslant L_{\rho_k}(x^k,z^k,\omega^*)=L_{\rho_k}(x^k,z^k,\omega^k)+(\omega^*-\omega^k)^\top(Qx^k-z^k)\\ &\Longrightarrow (\omega^k-\omega^*)^\top(Qx^k-z^k)\leqslant L_{\rho_k}(x^k,z^k,\omega^k)-\phi^C(\omega^*). \qquad (5.21)\end{aligned}$$

Substituting the inequality (5.21) into equality (5.20), we have

$$\begin{aligned}\left\|\omega^{k+1}-\omega^*\right\|_2^2 &\leqslant \left\|\omega^k-\omega^*\right\|_2^2\\ &\quad+2\rho_k\left[L_{\rho_k}(x^k,z^k,\omega^k)-\phi^C(\omega^*)\right]+\rho_k^2\left\|Qx^k-z^k\right\|_2^2 \qquad (5.22)\\ &= \left\|\omega^k-\omega^*\right\|_2^2+2\rho_k\left[\breve{\phi}(\omega^k,x^{k-1})-\phi^C(\omega^*)\right]\\ &\quad+2\rho_k\left[L_{\rho_k}(x^k,z^k,\omega^k)+\frac{\rho_k}{2}\left\|Qx^k-z^k\right\|_2^2-\breve{\phi}(\omega^k,x^{k-1})\right]. \qquad (5.23)\end{aligned}$$

By assumption, for each $k\geqslant 1$, we have $\phi^C(\omega^k)<\phi^C(\omega^*)$, so by Lemma 5.8 and $\epsilon=0$, the Line 8 condition of Algorithm 5.1 never holds. Therefore the alternate Line 9 termination condition, which is equivalent to the satisfaction of condition (5.18), must be satisfied for each $k\geqslant 1$. Rewriting (5.18), with the substitution $\widetilde{\omega}^k=\omega^{k+1}$, as

$$L_{\rho_k}(x^k,z^k,\omega^k)+\frac{\rho_k}{2}\left\|Qx^k-z^k\right\|_2^2-\breve{\phi}(\omega^k,x^{k-1})\leqslant\frac{\breve{\phi}(\omega^{k+1},x^k)-\breve{\phi}(\omega^k,x^{k-1})}{\gamma_k} \qquad (5.24)$$

and substituting (5.24) into (5.23), we have

$$\begin{aligned}\left\|\omega^{k+1}-\omega^*\right\|_2^2 &\leqslant \left\|\omega^k-\omega^*\right\|_2^2+2\rho_k\left[\breve{\phi}(\omega^k,x^{k-1})-\phi^C(\omega^*)\right]\\ &\qquad+\frac{2\rho_k}{\gamma_k}\left[\breve{\phi}(\omega^{k+1},x^k)-\breve{\phi}(\omega^k,x^{k-1})\right]\\ &\leqslant \left\|\omega^k-\omega^*\right\|_2^2+2\rho_k\left[\breve{\phi}(\omega^{k+1},x^k)-\phi^C(\omega^*)\right]\\ &\qquad+2\rho_k\left(\frac{1-\gamma_k}{\gamma_k}\right)\left[\breve{\phi}(\omega^{k+1},x^k)-\breve{\phi}(\omega^k,x^{k-1})\right] \qquad (5.25)\end{aligned}$$

From (5.25), we make the following three inferences: 1) that $\left\{\left\|\omega^k-\omega^*\right\|\right\}$ is bounded, 2) that $\sum_{k=1}^{\infty}\left[\phi^C(\omega^*)-\phi^C(\omega^k)\right]$ is finite, and 3) that $\left\{\omega^k\right\}$ converges. To establish these inferences, we rearrange terms and sum the inequality (5.25) from $k=\ell,\ldots,N$ for some integers $1\leqslant\ell\leqslant N$ to

get

$$
\begin{aligned}
&2\sum_{k=\ell}^{N}\rho_k\left[\phi^C(\omega^*)-\breve{\phi}(\omega^{k+1},x^k)\right]+\left\|\omega^{N+1}-\omega^*\right\|_2^2\\
&\quad\leqslant\left\|\omega^{\ell}-\omega^*\right\|_2^2+2\rho_N\left(\frac{1-\gamma_N}{\gamma_N}\right)\breve{\phi}(\omega^{N+1},x^N)-2\rho_\ell\left(\frac{1-\gamma_\ell}{\gamma_\ell}\right)\breve{\phi}(\omega^{\ell},x^{\ell-1})\\
&\quad+2\sum_{k=\ell}^{N-1}\left[\rho_k\left(\frac{1-\gamma_k}{\gamma_k}\right)-\rho_{k+1}\left(\frac{1-\gamma_{k+1}}{\gamma_{k+1}}\right)\right]\breve{\phi}(\omega^{k+1},x^k)\\
&\quad\leqslant\left\|\omega^{\ell}-\omega^*\right\|_2^2+2c\left[\phi^C(\omega^*)-\breve{\phi}(\omega^{\ell},x^{\ell-1})\right],
\end{aligned}
\tag{5.26}
$$

where the last inequality follows from the assumption that $\rho_k\left(\frac{1-\gamma_k}{\gamma_k}\right)=c$ for all k and the bounding relationships implied by the optimality of ω^*:

$$
\phi^C(\omega^*)>\phi^C(\omega^{k+1})\geqslant\breve{\phi}(\omega^{k+1},x^k)
\tag{5.27}
$$

Noting that each summand $\phi^C(\omega^*)-\breve{\phi}(\omega^{k+1},x^k)$ in the summation on the left-hand side of (5.26) is nonnegative so we have immediately from (5.26) that $\sum_{k=1}^{\infty}\rho_k\left[\phi^C(\omega^*)-\breve{\phi}(\omega^k,x^{k-1})\right]<\infty$ and $\left\{(\omega^k-\omega^*)\right\}$ is bounded, establishing the first two inferences from (5.25). The validity of the first two inferences imply the boundedness of $\left\{\omega^k\right\}$ and the convergence $\lim_{k\to\infty}\breve{\phi}(\omega^k,x^{k-1})=\phi^C(\omega^*)$, respectively. The boundedness of $\left\{\omega^k\right\}$ implies the existence of limit points, while the convergence $\lim_{k\to\infty}\breve{\phi}(\omega^k,x^{k-1})=\phi^C(\omega^*)$ implies that all such limit points are dual optimal. It is straightforward from (5.27) that $\lim_{k\to\infty}\phi^C(\omega^k)=\phi^C(\omega^*)$ also.

To establish the third assertion, that $\left\{\omega^k\right\}$ in fact converges, we drop the summation from the left-hand side of (5.26),

$$
\left\|\omega^{N+1}-\omega^*\right\|_2^2\leqslant\left\|\omega^{\ell}-\omega^*\right\|_2^2+2c\left[\phi^C(\omega^*)-\breve{\phi}(\omega^{\ell},x^{\ell-1})\right],
\tag{5.28}
$$

and note that the above analysis holds independent of the choice of dual optimal ω^*. Since it was just shown that $\left\{\omega^k\right\}$ has limit points, and that all such limit points are dual optimal, we now specify ω^* to be one of these limit points. We then choose an appropriate ℓ for any $\varepsilon>0$ so that the right-hand side of (5.28) is arbitrarily small, i.e.,

$$
\left\|\omega^{N+1}-\omega^*\right\|_2^2\leqslant\varepsilon
$$

for all $N\geqslant\ell$. Thus, $\lim_{k\to\infty}\omega^k=\omega^*$, and it is clear that the limit point ω^* of $\left\{\omega^k\right\}$ is in fact unique.

To prove the last assertion, the satisfaction of (5.18) is rewritten as

$$\begin{aligned}\check{\phi}(\omega^{k+1}, x^k) - \check{\phi}(\omega^k, x^{k-1}) &\leqslant \hat{\phi}(\omega^k, x^k, z^k) - \check{\phi}(\omega^k, x^{k-1}) \\ &\leqslant \frac{1}{\gamma_k}\left(\check{\phi}(\omega^{k+1}, x^k) - \check{\phi}(\omega^k, x^{k-1})\right).\end{aligned}$$

Due to the convergence $\lim_{k\to\infty} \check{\phi}(\omega^k, x^{k-1}) = \zeta^{CLD}$, we have on taking the limit as $k \to \infty$ of the last displayed inequalities that $\lim_{k\to\infty} \hat{\phi}(\omega^k, x^k, z^k) = \zeta^{CLD}$. Noting that $\hat{\phi}(\omega^k, x^k, z^k) = L_{\rho_k}(x^k, z^k, \omega^k) + \frac{\rho_k}{2}\left\|Qx^k - z^k\right\|_2^2$, it is clear that if $\limsup_{k\to\infty} \rho_k = \infty$, then we have

$$\lim_{k\to\infty} \left\|Qx^k - z^k\right\|_2^2 = 0$$

and

$$\lim_{k\to\infty} L_{\rho_k}(x^k, z^k, \omega^k) = \zeta^{CLD},$$

and so the limit points of $\left\{(x^k, z^k)\right\}$ must be feasible and furthermore optimal for (5.2). Now assume $0 < \limsup \rho_k < \infty$. In taking the limit points $(\bar{x}, \bar{z}, \omega^*)$ of the sequence $\left\{(x^k, z^k, \omega^k)\right\}$ and $\bar{\rho}$ of $\{\rho_k\}$, noting that the optimal value of problem (5.9) with $\omega = \omega^*$ is ζ^{CLD} by Lemma 5.3,

$$\zeta^{CLD} + \frac{\bar{\rho}}{2}\left\|Q\bar{x} - \bar{z}\right\|_2^2 \leqslant L_{\bar{\rho}}(\bar{x}, \bar{z}, \omega^*) + \frac{\bar{\rho}}{2}\left\|Q\bar{x} - \bar{z}\right\|_2^2 = \zeta^{CLD}.$$

From this, it follows that $\left\|Q\bar{x} - \bar{z}\right\|_2^2 = 0$ and $L_{\bar{\rho}}(\bar{x}, \bar{z}, \omega^*) = \zeta^{CLD}$, and so $(\bar{x}, \bar{z})$ must be feasible and furthermore optimal for (5.2). ■

5.2.2 Convergence Rate Analysis for Augmented Lagrangian Method

The proof of Proposition 5.10 allows for the following remarks on the rate-of-convergence associated with $\lim_{k\to\infty} \check{\phi}(\omega^k, x^{k-1}) = \zeta^{CLD}$. Note that each iteration k of Algorithm 5.1 corresponds to a serious step update of ω^k.

1. Let $\rho_k = \rho$ and $\gamma_k = \gamma$ for all $k \geqslant 1$, where $\rho > 0$ and $\gamma \in (0, 1)$ are constants. Then, from the proof of Proposition 5.10, we have

   $$\sum_{k=1}^{N}\left[\phi^C(\omega^*) - \check{\phi}(\omega^k, x^{k-1})\right] < \infty$$

   and since $\left\{\check{\phi}(\omega^k, x^{k-1})\right\}$ is monotonically non-decreasing, it is clear that

   $$\phi^C(\omega^*) - \check{\phi}(\omega^k, x^{k-1}) = o(1/k),$$

   where o is the Little-o notation.

2. Let $\rho_k = k\rho$ for some constant $\rho > 0$ and $\gamma_k = \frac{\rho_k}{c+\rho_k}$ for some constant $c > 0$ so that $\rho_k \left(\frac{1-\gamma_k}{\gamma_k}\right) = c$ is a constant. Then we have

$$\sum_{k=1}^{N} k \left[\phi^C(\omega^*) - \check{\phi}(\omega^k, x^{k-1})\right] < \infty,$$

$$\text{and so} \qquad \phi^C(\omega^*) - \check{\phi}(\omega^k, x^{k-1}) = o(1/k^2).$$

3. Let $\rho_k = \rho b^k$ for some constant $\rho > 0$ and $b > 1$, and as $\gamma_k = \frac{\rho_k}{c+\rho_k}$ for some constant $c > 0$ so that $\rho_k \left(\frac{1-\gamma_k}{\gamma_k}\right) = c$, a constant. Then we have

$$\sum_{k=1}^{N} b^k \left[\phi^C(\omega^*) - \check{\phi}(\omega^k, x^{k-1})\right] < \infty,$$

$$\text{and so} \qquad \phi^C(\omega^*) - \check{\phi}(\omega^k, x^{k-1}) = o(b^{-k}).$$

Since the number of null step updates per serious step is not fixed, and a null step does not require significantly less computational effort, these convergence results in terms of serious steps cannot be generalised to a convergence result in terms of total steps or runtime without some notion of how often null steps are taken.

Exploratory numerical tests did not conclusively reveal any clear and consistent pattern to the frequency of null steps as the algorithm progresses. However, for at least some combinations of instances and parameters the null step frequency increased significantly as the duality gap decreased. Therefore, the practical convergence behaviour of the algorithm in terms of runtime is likely worse than the above results would suggest.

5.2.3 Integration of the Simplicial Decomposition and Gauss-Seidel Methods

We consider the following general two-block problem

$$\min_{x,z} \{F(x,z) : x \in \operatorname{conv}(X), z \in Z\} \tag{5.29}$$

where $F : \mathbb{R}^n \times \mathbb{R}^q \mapsto \mathbb{R}$ is a continuously differentiable function, $\operatorname{conv}(X)$ and Z are closed convex sets, and $\operatorname{conv}(X)$ is also bounded. (Z can be more generally a convex set in this setting, not necessarily a linear (sub)space.) Additionally, we assume for each fixed $x \in \operatorname{conv}(X)$ that $z \mapsto F(x,z)$ is inf-compact. (That is, the set $\{z \in Z : F(x,z) \leqslant \ell\}$ is compact for all $x \in \operatorname{conv}(X)$ and $\ell \in \mathbb{R}$.) In the context of Algorithm 5.1, we would identify $F(x,z) = L_\rho(x,z,\omega)$ for a given ω.

Problem (5.29) is assumed to be feasible, bounded, and to have an optimal solution (x^*, z^*). We shall utilise the following two-block nonlinear Gauss-Seidel (GS) method with the x update approximated in a manner resembling an iteration of the SDM. As per Assumption 5.2, we assume the user provides an oracle to return an extremal point in $\operatorname{conv}(X)$ when minimising a linear function over X. This can be used to initialise the following and later algorithms.

Algorithm 5.2 An iteration of inner-approximated nonlinear Gauss-Seidel approach applied to problem (5.29).

1: Precondition: $\widetilde{x} \in \operatorname{conv}(X)$, $\widetilde{z} \in \operatorname{argmin}_z \{F(\widetilde{x}, z) : z \in Z\}$, $D \subseteq \operatorname{conv}(X)$
2: **function** SDM-GS(F, X, Z, D, $\widetilde{x}$, $\widetilde{z}$, t_{max})
3: **for** $t = 1, \ldots, t_{max}$ **do**
4: $\widetilde{x} \leftarrow \operatorname{argmin}_x \{F(x, \widetilde{z}) : x \in D\}$
5: $\widetilde{z} \leftarrow \operatorname{argmin}_z \{F(\widetilde{x}, z) : z \in Z\}$
6: **end for**
7: $\widehat{x} \in \operatorname{argmin}_x \left\{\nabla_x F(\widetilde{x}, \widetilde{z})^\top (x - \widetilde{x}) : x \in X\right\}$
8: Reconstruct D to be any set such that
9: $\{\widetilde{x} + \alpha(\widehat{x} - \widetilde{x}) : \alpha \in [0, 1]\} \subseteq D \subseteq \operatorname{conv}(X)$
10: Set $\Gamma \leftarrow -\nabla_x F(\widetilde{x}, \widetilde{z})(\widehat{x} - \widetilde{x})$
11: **return** $(\widetilde{x}, \widetilde{z}, D, \Gamma)$
12: **end function**

If the z block update of Line 5 is trivialised, such as by making it not actually appear in the definition of F, or by making Z a singleton set, then Algorithm 5.2 would be identical to SDM applied to problem (5.29) in which the z block of variables correspondingly does not play any role. On the other hand, if the x update (4) is replaced with an update based on an exact minimisation $\widetilde{x} \leftarrow \operatorname{argmin}_x \{F(x, \widetilde{z}) : x \in \operatorname{conv}(X)\}$ (so that the computations of Lines 7–10 and the returning of D and Γ can be skipped), then Algorithm 5.2 would be equivalent to a more traditional two-block nonlinear Gauss-Seidel method. Different forms of approximation of the x update, such as those resulting from gradient descent steps in x, are also considered in [53, 20].

Remark 5.11 *The main approach envisioned for constructing the inner approximation D on lines 8 and 9 is to take $D \leftarrow \operatorname{conv}(D \cup \{\widetilde{x}, \widehat{x}\})$. To implement this update of D, we need to save the points $\widehat{x}$ computed during previous calls to Algorithm 5.2.*

We assume in the following proposition that Algorithm 5.2 is applied iteratively in the sense that at iteration $k \geqslant 0$, we input $(\widetilde{x}, \widetilde{z}) = (x^k, z^k)$ and return $(\widetilde{x}, \widetilde{z}) = (x^{k+1}, z^{k+1})$. Furthermore,

at the same iteration k call of Algorithm 5.2, we set $d^{k+1} = \widehat{x} - \widetilde{x}$ where $\widehat{x}$ and $\widetilde{x}$ are set as in Line 9. This provides a reference sequence of directions $\{d^k\}$ necessary in the proof of the following proposition.

Proposition 5.12 *For problem* (5.29)*, let F be convex and continuously differentiable, and let* $\operatorname{conv}(X)$ *and Z be nonempty and convex, with* $\operatorname{conv}(X)$ *bounded and $z \mapsto F(x,z)$ inf-compact for each $x \in \operatorname{conv}(X)$. Then, for any $t_{max} \geqslant 1$, the sequence $\{(x^k, z^k)\}$ generated by iterations of Algorithm 5.2 has limit points $(\bar{x}, \bar{z})$, each of which are optimal for problem* (5.29)*.*

Proof. In light of the convexity and continuous differentiablity of F and the convexity of $\operatorname{conv}(X)$ and Z, it is sufficient to show that

$$\nabla_x F(\bar{x}, \bar{z})^\top (x - \bar{x}) \geqslant 0 \quad \text{for all } x \in \operatorname{conv}(X) \tag{5.30}$$

$$\text{and} \quad \nabla_z F(\bar{x}, \bar{z})^\top (z - \bar{z}) \geqslant 0 \quad \text{for all } z \in Z. \tag{5.31}$$

As $\nabla_z F(x^k, z^k)^\top (z - z^k) \geqslant 0$ for all $z \in Z$ holds for each $k \geqslant 1$ (this follows due to the optimality $z^k \in \operatorname{argmin}_z \{F(x^k, z) : z \in Z\}$ that holds by construction) the satisfaction of the latter condition (5.31) is trivially established for any limit points $(\bar{x}, \bar{z})$. It remains only to show the satisfaction of the x-stationarity condition (5.30). This may be established by using Proposition 3.2 of [20] combined with the last sentence of Remark 3.3 from the same reference; a more explicit proof that (5.30) holds is given in Section 5.2.4.

Note, for the sake of nontriviality, that $\nabla_x F(x^k, z^k)^\top (x - x^k) \geqslant 0$ for all $x \in X$ is assumed *not* to hold for any $k \geqslant 1$. Thus, with reference to the argument given in Section 5.2.4, the sequence of directions $\{d^k\}$ satisfy the the Direction Assumption (DA), prior to Algorithm 5.3. Also, the Gradient Related Assumption (GRA) referred to in Section 5.2.4 is satisfied for this same $\{d^k\}$, by Lemma 5.16 therein. Due to the construction of D in Line 9 and setting $(x^{k+1}, z^{k+1}) = (\widetilde{x}, \widetilde{z})$ after the termination of the for loop of Lines 3–6, we have given $\{d^k\}$ and any choice of $(\beta, \sigma) \in (0, 1)$ the satisfaction of the Sufficient Decrease Assumption (SDA), also referred to in Section 5.2.4. It then follows from Lemma 5.14 that limit points $(\bar{x}, \bar{z})$ of $\{(x^k, z^k)\}$ do exists, and that each of which satisfies the stationarity condition (5.30). ■

5.2.4 Establishing optimal convergence of SDM-GS

Given initial $(x^0, z^0) \in X \times Z \subset \mathbb{R}^n \times \mathbb{R}^q$, we consider the generation of the sequence $\{(x^k, z^k)\}$ with iterations computed using Algorithm 5.3, whose target problem is given by

$$\min_{x,z} \{F(x,z) : x \in X, z \in Z\}, \tag{5.32}$$

where $(x,z) \mapsto F(x,z)$ is convex and continuously differentiable over $X \times Z$, and sets X and Z are closed and convex, with X bounded and $z \mapsto F(x,z)$ is inf-compact for each $x \in X$.

We define the directional derivative with respect to x as

$$F'_x(x,z;d) := \lim_{\alpha \downarrow 0} \frac{F(x+\alpha d, z) - F(x,z)}{\alpha}.$$

Of interest is the satisfaction of the following local stationarity condition at $x \in X$:

$$F'_x(x,z;d) \geqslant 0 \quad \text{for all } d \in X - \{x\} \tag{5.33}$$

for any limit point $(x,z) = (\bar{x}, \bar{z})$ of some sequence $\{(x^k, z^k)\}$ of feasible solutions to problem (5.32). For the sake of nontriviality, we shall assume that the x-stationarity condition (5.33) never holds at $(x,z) = (x^k, z^k)$ for any $k \geqslant 0$. Thus, for each x^k, $k \geqslant 0$, there always exists a $d^k \in X - \{x^k\}$ for which $F'_x(x^k, z^k; d^k) < 0$.

Direction Assumptions (DAs): For each iteration $k \geqslant 0$, given $x^k \in X$ and $z^k \in Z$, we have d^k chosen so that 1) $x^k + d^k \in X$; and 2) $F'_x(x^k, z^k; d^k) < 0$.

Gradient Related Assumption (GRA): Given a sequence $\{(x^k, z^k)\}$ with $\lim_{k\to\infty}(x^k, z^k) = (\overline{x}, \overline{z})$, and a bounded sequence $\{d^k\}$ of directions, then the existence of a direction $\overline{d} \in X - \{\overline{x}\}$ such that $F'_x(\overline{x}, \overline{z}; \overline{d}) < 0$ implies that

$$\limsup_{k\to\infty} F'_x(x^k, z^k; d^k) < 0. \tag{5.34}$$

In this case, we say that $\{d^k\}$ is *gradient related* to $\{x^k\}$. This gradient related condition is similar to the one defined in [12]. The sequence of directions d^k is typically gradient related to $\{x^k\}$ by construction. (See Lemma 5.16.)

To state the last assumption, we require the notion of an Armijo rule step length $\alpha^k \in (0,1]$ given (x^k, z^k, d^k) and parameters $\beta, \sigma \in (0,1)$.

Algorithm 5.3 Computing an Armijo rule step length α^k at iteration k.

```
1: function ArmijoStep(F, x^k, z^k, d^k, β, σ)
2:     α^k ← 1
3:     while F(x^k + α^k d^k, z^k) − F(x^k, z^k) > α^k σ F'_x(x^k, z^k; d^k) do
4:         α^k ← β α^k
5:     end while
6:     return α^k
7: end function
```

Remark 5.13 *Under mild assumptions on F such as continuity that guarantee the existence of finite $F'_x(x, z; d)$ for all $(x, z, d) \in \{(x, z, d) : x \in X, d \in X - \{x\}, z \in Z\}$, we may assume that the while loop of Lines 3–5 terminates after a finite number of iterations. Thus, we have $\alpha^k \in (0, 1]$ for each $k \geqslant 1$.*

The last significant assumption is stated as follows.

Sufficient Decrease Assumption (SDA): For sequences $\{(x^k, z^k, d^k)\}$ and step lengths $\{\alpha^k\}$ computed according to Algorithm 5.3, we assume for each $k \geqslant 0$, that (x^{k+1}, z^{k+1}) satisfies

$$F(x^{k+1}, z^{k+1}) \leqslant F(x^k + \alpha^k d^k, z^k).$$

Lemma 5.14 resembles standard Armijo rule convergence proofs for algorithms which take iterative steps in a single variable, but incorporating the z-update required by our Gausss-Seidel approach requires special care.

Lemma 5.14 *For problem* (5.32)*, let $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \mapsto \mathbb{R}$ be convex and continuously differentiable, $X \subset \mathbb{R}^{n_x}$ convex and compact, and $Z \subseteq \mathbb{R}^{n_z}$ closed and convex. Furthermore, assume for each $x \in X$ that $z \mapsto F(x, z)$ is inf-compact. If a sequence $\{(x^k, z^k, d^k)\}$ satisfies the DA, the GRA, and the SDA for some fixed $\beta, \sigma \in (0, 1)$, then the sequence (x^k, z^k) has limit points $(\overline{x}, \overline{z})$, each of which satisfies the stationarity condition* (5.33)*.*

Proof. The existence of limit points $(\overline{x}, \overline{z})$ follows from the compactness of X, the inf-compactness of $z \mapsto F(x, z)$ for each $x \in X$, and the SDA. In generating $\{\alpha^k\}$ according to the Armijo rule as implemented in Lines 2–5 of Algorithm 5.3, we have

$$\frac{F(x^k + \alpha^k d^k, z^k) - F(x^k, z^k)}{\alpha^k} \leqslant \sigma F'_x(x^k, z^k; d^k). \tag{5.35}$$

By the DA, $F'_x(x^k, z^k; d^k) < 0$ and since $\alpha^k > 0$ for each $k \geqslant 1$ by Remark 5.13, we infer from (5.35) that $F(x^k + \alpha^k d^k, z^k) < F(x^k, z^k)$. By construction, we have $F(x^{k+1}, z^{k+1}) \leqslant F(x^k + \alpha^k d^k, z^k) < F(x^k, z^k)$.. By the monotonicity $F(x^{k+1}, z^{k+1}) < F(x^k, z^k)$ and F being bounded from below on $X \times Z$, we have $\lim_{k\to\infty} F(x^k, z^k) = \bar{F} > -\infty$. Therefore,

$$\lim_{k\to\infty} F(x^{k+1}, z^{k+1}) - F(x^k, z^k) = 0,$$

which implies

$$\lim_{k\to\infty} F(x^k + \alpha^k d^k, z^k) - F(x^k, z^k) = 0. \tag{5.36}$$

We assume for sake of contradiction that $\lim_{k\to\infty}(x^k, z^k) = (\overline{x}, \overline{z})$ does not satisfy the stationarity condition (5.33). By GRA, we have that $\{d^k\}$ is gradient related to $\{x^k\}$; that is,

$$\limsup_{k\to\infty} F'_x(x^k, z^k; d^k) < 0. \tag{5.37}$$

Thus, it follows from (5.35)–(5.37) that $\lim_{k\to\infty} \alpha^k = 0$.

Consequently, after a certain iteration $k \geqslant \bar{k}$, we can define $\{\bar{\alpha}^k\}$, $\bar{\alpha}^k = \alpha^k/\beta$, where $\bar{\alpha}^k \leqslant 1$ for $k \geqslant \bar{k}$, and so we have

$$\sigma F'_x(x^k, z^k; d^k) < \frac{F(x^k + \bar{\alpha}^k d^k, z^k) - F(x^k, z^k)}{\bar{\alpha}^k}. \tag{5.38}$$

Since F is continuously differentiable, the mean value theorem may be applied to the right-hand side of (5.38) to get

$$\sigma F'_x(x^k, z^k; d^k) < F'_x(x^k + \tilde{\alpha}^k d^k, z^k; d^k), \tag{5.39}$$

for some $\tilde{\alpha}^k \in [0, \overline{\alpha}^k]$.

Again, using the assumption $\limsup_{k\to\infty} F'_x(x^k, z^k; d^k) < 0$, and also the compactness of $X - X$, we take a limit point $\overline{d}$ of $\{d^k\}$, with its associated subsequence index set denoted by $\mathcal{K}$, such that $F'_x(\overline{x}, \overline{z}, \overline{d}) < 0$. Taking the limits over the subsequence indexed by $\mathcal{K}$, we have

$$\lim_{k\to\infty, k\in\mathcal{K}} F'_x(x^k, z^k; d^k) = F'_x(\overline{x}, \overline{z}; \overline{d})$$

and

$$\lim_{k\to\infty, k\in\mathcal{K}} F'_x(x^k + \tilde{\alpha}^k d^k, z^k; d^k) = F'_x(\overline{x}, \overline{z}; \overline{d}).$$

These two limits holds since 1) $(x, z) \mapsto F'_x(x, z; d)$ for each $d \in X - X$ is continuous and 2) $d \mapsto F'_x(x, z; d)$ is locally Lipschitz continuous for each $(x, z) \in X \times Z$ (e.g., Proposition 2.1.1 of [26]);

these two facts together imply that $(x, z; d) \mapsto F'_x(x, z; d)$ is continuous. Then, inequality (5.39) becomes in the limit as $k \to \infty$, $k \in \mathcal{K}$,

$$\sigma F'_x(\overline{x}, \overline{z}; \overline{d}) \leqslant F'_x(\overline{x}, \overline{z}; \overline{d}) \quad \Longrightarrow \quad 0 \leqslant (1-\sigma) F'_x(\overline{x}, \overline{z}; \overline{d}).$$

Since $(1-\sigma) > 0$ and $F'_x(\overline{x}, \overline{z}; \overline{d}) < 0$, we have a contradiction. Thus, $\overline{x}$ must satisfy the stationary condition (5.33). ■

Remark 5.15 *Noting that $F'_x(x^k, z^k; d^k) = \nabla_x F(x^k, z^k)^\top d^k$ under the assumption of continuous differentiability of F, one means of constructing $\{d^k\}$ is as follows:*

$$d^k \leftarrow \operatorname*{argmin}_d \left\{ \nabla_x F(x^k, z^k)^\top d : d \in X - \{x^k\} \right\}. \tag{5.40}$$

Lemma 5.16 *Given sequence $\{(x^k, z^k)\}$ with $\lim_{k\to\infty}(x^k, z^k) = (\overline{x}, \overline{z})$, let each d^k, $k \geqslant 1$, be generated as in* (5.40)*. Then $\{d^k\}$ is gradient related to $\{x^k\}$.*

Proof. By the construction of d^k, $k \geqslant 1$, we have

$$F'_x(x^k, z^k; d^k) \leqslant F'_x(x^k, z^k; d) \quad \forall\, d \in X - \{x^k\}.$$

Taking the limit, we have

$$\limsup_{k\to\infty} F'_x(x^k, z^k; d^k) \leqslant \limsup_{k\to\infty} F'_x(x^k, z^k; d) \leqslant F'_x(\overline{x}, \overline{z}; d) \quad \forall\, d \in X - \{\overline{x}\},$$

where the last inequality follows from the upper semicontinuity of the function $(x, z, d) \mapsto F'_x(x, z; d)$, which holds in our setting due, primarily, to Proposition 2.1.1 (b) of [26] given that F is assumed to be convex and continuous on $\mathbb{R}^n$. Taking

$$\overline{d} \in \operatorname*{argmin}_d \left\{ F'_x(\overline{3x}, \overline{z}; d) : d \in X - \{\overline{x}\} \right\},$$

we have by the assumed nonstationarity that $F'_x(\overline{x}, \overline{z}; \overline{d}) < 0$. Thus, $\limsup_{k\to\infty} F'_x(x^k, z^k; d^k) < 0$, and so GRA holds. ■

5.2.5 Implementing the Augmented Lagrangian Method using SDM-GS

The method SDM-GS-ALM is now stated as Algorithm 5.4, which uses Algorithm 5.2 as a subroutine to provide a practical implementation of Algorithm 5.1.

Remark 5.17 *At the return of Algorithm 5.2 in Line 8 of Algorithm 5.4, we have*

$$\begin{aligned}\Gamma &= -\nabla_x L_\rho(x^k, z^k, \omega^k)^\top (\widehat{x} - x^k) \\ &= -\left[\nabla_x f(x^k) + \left(\omega^k + \rho(Qx^k - z^k)\right)^\top Q\right]^\top (\widehat{x} - x^k)\end{aligned}$$

where $\widehat{x}$ is computed on Line 7 of Algorithm 5.2. One may verify the equality $(Qx^k - z^k)^\top z^k = 0$ due to $z^k \in \operatorname{argmin}_z \left\{\left\|Qx^k - z\right\|_2^2 : z \in Z\right\}$. Moreover using this value of Γ and the computation of $\widetilde{\phi}$ on Lines 4 and 12 one may show, using the fact that $\hat{x} \in \operatorname{argmin}_x\{\nabla_x L_\rho(x^k, z^k, \omega^k)^\top (x - x^k) : x \in X\}$, that

$$\widetilde{\phi} = L_\rho(x^k, z^k, \omega^k) + \frac{\rho}{2}\left\|Qx^k - z^k\right\|_2^2 - \Gamma = \check{\phi}\left(\omega^k + \rho(Qx^k - z^k), x^k\right).$$

Algorithm 5.4 A practical implementation of Algorithm 5.1 based on the use of SDM-GS iterations. (SDM-GS is given as Algorithm 5.2.)

1: **Preconditions:** $x^0 \in \operatorname{conv}(X)$, $z^0 \in Z$, $\omega^0 \in Z^\perp$, $D \subseteq \operatorname{conv}(X)$, $\gamma \in (0, 1)$.
2: **function** SDM-GS-ALM(f, Q, X, Z, D, ρ, x^0, z^0, ω^0, γ, ϵ, t_{max}, k_{max})
3: $(x^0, z^0, D, \Gamma) \leftarrow$ SDM-GS($L_\rho(\cdot, \cdot, \omega^0)$, X, Z, D, x^0, z^0, t_{max})
4: $\widetilde{\phi} \leftarrow L_\rho(x^0, z^0, \omega^0) + \frac{\rho}{2}\left\|Qx^0 - z^0\right\|_2^2 - \Gamma$
5: set $\omega^0 \leftarrow \omega^0 + \rho(Qx^0 - z^0)$, $\check{\phi}^0 \leftarrow \widetilde{\phi}$
6: **for** $k = 1, 2, \ldots, k_{max}$ **do**
7: Initialise $\omega^k \leftarrow \omega^{k-1}$, $\check{\phi}^k \leftarrow \check{\phi}^{k-1}$ ▷ (Default, null-step updates)
8: $(x^k, z^k, D, \Gamma) \leftarrow$ SDM-GS($L_\rho(\cdot, \cdot, \omega^k)$, X, Z, D, x^{k-1}, z^{k-1}, t_{max})
9: **if** $L_\rho(x^k, z^k, \omega^k) + \frac{\rho}{2}\left\|Qx^k - z^k\right\|_2^2 - \check{\phi}^k \leqslant \epsilon$ **then**
10: **return** $(x^k, z^k, \omega^k, \check{\phi}^k)$
11: **end if**
12: $\widetilde{\phi} \leftarrow L_\rho(x^k, z^k, \omega^k) + \frac{\rho}{2}\left\|Qx^k - z^k\right\|_2^2 - \Gamma$
13: $\eta_k \leftarrow \frac{\widetilde{\phi} - \check{\phi}^k}{L_\rho(x^k, z^k, \omega^k) + \frac{\rho}{2}\|Qx^k - z^k\|_2^2 - \check{\phi}^k}$
14: **if** $\eta_k \geqslant \gamma$ **then**
15: set $\omega^k \leftarrow \omega^k + \rho(Qx^k - z^k)$, $\check{\phi}^k \leftarrow \widetilde{\phi}$
16: **end if**
17: Possibly update ρ, e.g., $\rho \leftarrow \frac{1}{\min\{\max\{(2/\rho)(1-\eta_k), 1/(10\rho), 10^{-4}\}, 10/\rho\}}$ as in [69]
18: If ρ was updated, $\gamma \leftarrow \frac{\rho_k}{\rho_k + c}$
19: **end for**
20: **return** $(x^k, z^k, \omega^k, \check{\phi}^k)$
21: **end function**

Proposition 5.18 *Let $\left\{(x^k, z^k, \omega^k)\right\}$ be a sequence generated by Algorithm 5.4 applied to problem* (5.1) *with X compact, Z a linear subspace, $\omega^0 \in Z^\perp$, $\rho > 0$, $\gamma \in (0, 1)$, $\epsilon = 0$ and $k_{max} = \infty$. If there exists a dual optimal solution ω^* to the dual problem* (5.3)*, then either*

1. $\omega^k = \overline{\omega}$ is fixed and optimal for (5.3) *for $k \geqslant \bar{k}$ for some finite $\bar{k}$; or*

2. ω^k is never optimal for (5.3) *for any finite $k \geqslant 1$, but $\lim_{k\to\infty} \omega^k = \overline{\omega}$ is optimal,*

and the sequence $\{(x^k, z^k)\}$ has limit points $(\overline{x}, \overline{z})$, each of which are optimal for problem (5.2)*.*

Proof. In the first case, Algorithm 5.4 never takes serious steps for iterations $k \geqslant \bar{k} \geqslant 1$, and so with $\omega^k = \overline{\omega}$, optimal for (5.3) and fixed for $k \geqslant \bar{k}$, the Algorithm 5.4 iterations continue with the generation of $\{(x^k, z^k)\}$ as generated by iterations of SDM-GS (Algorithm 5.2). By Proposition 5.12, the sequence $\{(x^k, z^k)\}$ has limit points $(\overline{x}, \overline{z})$, each of which is optimal for problem (5.9) with $\omega = \overline{\omega}$. Then, by Lemma 5.3, $(\overline{x}, \overline{z})$ is also optimal for problem (5.2) since $\overline{\omega}$ is optimal for (5.3).

In the second case, where ω^k is never dual optimal for (5.2) for any finite $k \geqslant 1$, any serious step must be followed by a finite number of consecutive null-steps. We consider the subsequence indices $\{k_i\}_{i=1}^{\infty}$ where the update ω^{k_i+1} is obtained by a serious step. By Proposition 5.10, we have $\lim_{i\to\infty} \phi^C(\omega^{k_i+1}) = \zeta^{CLD}$, and accommodating the null steps in between, we have also $\lim_k \phi^C(\omega^k) = \zeta^{CLD}$. To prove the last claim, we note that $\omega^j = \omega^{k_{i+1}}$ for all integers j such that $k_i < j \leqslant k_{i+1}$ due to the taking of null steps. From Proposition 5.10, we have that $\lim_{i\to\infty} L_\rho(x^{k_i}, z^{k_i}, \omega^{k_i}) = \zeta^{CLD}$. By the continuity of $(x, z, \omega) \mapsto L_\rho(x, z, \omega)$, the convergences $\lim_{k\to\infty} \omega^k = \overline{\omega}$ and $\lim_{i\to\infty} Qx^{k_i} - z^{k_i} = 0$ (again, Proposition 5.10), we have $\lim_{i\to\infty} L_\rho(x^{k_i}, z^{k_i}, \omega^{k_{i+1}}) = \zeta^{CLD}$ also. Next, at each i, and integers j such that $k_i < j \leqslant k_{i+1}$, observe that

$$L_\rho(x^{k_i}, z^{k_i}, \omega^{k_{i+1}}) \geqslant L_\rho(x^j, z^j, \omega^j) \geqslant L_\rho(x^{k_{i+1}}, z^{k_{i+1}}, \omega^{k_{i+1}}).$$

In taking the limit of the above inequality as $i \to \infty$, it becomes evident that $\lim_{k\to\infty} L_\rho(x^k, z^k, \omega^k) = \zeta^{CLD}$ in the original sequence also. By the optimality of $\overline{\omega}$ for problem (5.3), we know from Lemma 5.3 that $\zeta_\rho^{AL}(\overline{\omega}) = \zeta^{CLD}$, and so each limit point $(\overline{x}, \overline{z})$ must be optimal for problem (5.9) with $\omega = \overline{\omega}$. Furthermore, by Lemma 5.3, $(\overline{x}, \overline{z})$ must also be optimal for problem (5.2). (These limit points exist furthermore, due to the compactness of $\operatorname{conv}(X)$ and the continuous and closed-form expression that the unique solution $z^k \in \operatorname{argmin}_z \left\{ \|Qx^k - z\|_2^2 : z \in Z \right\}$ has given $x^k \in \operatorname{conv}(X)$ when Z is a linear subspace.) ■

5.2.6 Parallelisation and Workload

The opportunities for parallelisation and distribution of the computational workload in SDM-GS-ALM, as stated in Algorithm 5.4, are not immediately apparent. This section explicitly indicates which update problems may be solved in parallel, and the nature of the required communication between the parallel computational nodes.

The bulk of computational work, parallelisation, and parallel communication occurs within the SDM-GS method stated in Algorithm 5.2, where for the problems of interest, the following decomposable structures apply: $X = \prod_{i=1}^{m} X_i$, $D = \prod_{i=1}^{m} D_i$, and $F(x, z) = \sum_{i=1}^{m} F(x_i, z)$. In the larger context of Algorithm 5.4, the subproblem of Line 4 in Algorithm 5.2 can be solved in parallel given fixed $\widetilde{z} \in Z$ and $\omega \in Z^{\perp}$ along the block indices $i = 1, \ldots, m$ as

$$\min_{x} \left\{ f_i(x) + (\omega_i)^{\top} Q_i x + \frac{\rho}{2} \left\| Q_i x - \widetilde{z}_i \right\|_2^2 : x \in D_i \right\}, \tag{5.41}$$

while the subproblem of Line 7 is solved as

$$\min_{x} \left\{ \nabla_x f_i(\widetilde{x}_i) + \left(\omega_i + \rho(Q_i \widetilde{x}_i - \widetilde{z}_i)\right)^{\top} Q_i x : x \in X_i \right\}.$$

Remark 5.19 *In the setting where problem* (5.1) *is a large-scale mixed-integer linear optimization problem, the subproblems of Line 4 are continuous convex quadratic optimization problems for each block $i = 1, \ldots, m$, which can be solved independently of one another and in parallel. In the same setting, the Line 7 subproblems are mixed-integer optimization problems for each block $i = 1, \ldots, m$, which can also be solved independently of one another and in parallel. Additionally, the reconstruction of D occurring in Line 9 can be done in parallel for each D_i along the indices $i = 1, \ldots, m$.*

Parallel communication is needed for the computation of the z update in Line 5 in Algorithm 5.2. In the larger context of Algorithm 5.4, this takes the form of solving

$$\min_{z} \left\{ \sum_{i=1}^{m} \left\| Q_i \widetilde{x}_i - z_i \right\|_2^2 : z \in Z \right\}.$$

This minimum corresponds to taking an average, which we can accomplish this by using a reduce type parallel communication. This type of communication applies an operator (in this case, summation) to the value in each processor. Dividing by m afterwards yields the desired average. The evaluation of the serious step condition through calculating η_k in Line 13 in Algorithm 5.4 also requires a reduce-sum type parallel communication. For implementation purposes, the computation

of these values, including the computation of Γ from the SDM-GS call, can be combined into one reduce-sum communication.

In total, each iteration of Algorithm 5.4 requires two reduce-sum type communications, one for computing the z-update of Line 5 Algorithm 5.2, and one combined reduce-sum communication to compute scalars associated with the Lagrangian bounds and the critical values for the termination conditions. The storage and updates of x^k and ω^k and D can also be done in parallel, while z^k and γ^k need to be computed and stored by every processor at each iteration k.

5.3 Computational Results

5.3.1 Preliminary Information

In this section, we present and examine the results of two computational tests with the following purposes:

Test 1: to demonstrate the effect of enforcing the serious step condition on the Lagrangian values;

Test 2: to compare the parallel speedup obtained with the use of two parallel implementations of SDM-GS-ALM (Algorithm 5.4) versus the parallel speedups reported in [79] for two other parallel approaches. Additionally, the final iteration Lagrangian bounds are compared between the different parallel implementations for each experiment.

Computational experiments were performed on instances from three classes of problems. The first class consists of the capacitated allocation problems (CAP) [15]. The second and third classes consist of the DCAP and SSLP problems from the Stochastic Integer Programming Test Problem Library (SIPLIB), which are described in detail in [91, 3] and accessible at [3]. These are all large-scale mixed-integer linear optimization problems, so the preceding observations for when f is linear apply.

Test 1 was conducted with a Matlab 2012b [83] serial implementation of Algorithm 5.4 using CPLEX 12.6.1 [62] as the solver. The computing environment was on an Intel® Core™ i7-4770 3.40 GHz processor with 8 GB RAM and on a 64-bit operating system. All experiments for Test 1 were run with maximum number of iterations $k_{max} = 20$.

The parallel experiments of Test 2 were conducted with a C++ implementation of Algorithm 5.4 using CPLEX 12.5 [63] as the solver and the message passing interface (MPI) for parallel communication. For reading SMPS files into scenario-specific subproblems and for their interface with CPLEX, we used modified versions of the COIN-OR [1] Smi and Osi libraries, either to instantiate appropriate C++ class instances of the subproblems directly, or to write scenario-specific MPS files from the SMPS file. The computing environment for the Test 2 experiments is the Raijin cluster maintained by Australia's National Computing Infrastructure (NCI) and supported by the Australian government [89]. The Raijin cluster is a high performance computing (HPC) environment which has 3592 nodes (system units), 57472 cores of Intel Xeon E5-2670 processors with up to 8 GB PC1600 memory per core (128 GB per node). All experiments were conducted using one thread per CPLEX solve.

5.3.2 Effects of the Serious Step Condition

The results of the Test 1 set of experiments are depicted in the plots of Figures 5.1–5.4. The use of different penalty parameter ρ values is differentiated by the use of different plot colors. The penalties are chosen so that the smallest penalties (in red) are near optimal in terms of the resulting computational performance, while the larger penalties are known beforehand to be too large for optimal performance. For testing purposes, this is the most interesting way to choose penalty values, as smaller (than optimal) penalty values yield very little difference in Lagrangian bound between the use of different SSC parameter values. Solid line and dashed line plots depict the Lagrange bounds due to the use of a more stringent SSC parameter value $\gamma = 0.5$ and a more lenient value for the SSC parameter $\gamma = 0.125$, respectively. The dotted line plots depict the Lagrangian values resulting from the non-use of the SSC, so that it evaluates true no matter what. The following observations are suggested from the results of the Test 1 experiments:

1. First, the most significant differences between the varied use of SSC occur when the penalty coefficient values are large. In this setting, it seems to be the case that the use of more stringent (i.e., larger) values of the SSC parameter γ has the effect of mitigating the destabilizing effect of having a penalty parameter ρ value that is too large. This is significant because the performance of iterative Lagrangian dual solution approaches based on (or related to)

proximal bundle methods is sensitive to the tuning of the ρ value, and the optimal tuning of such parameters is assumed to be unknown beforehand in practical applications. For this reason, any mechanism to mitigate the effect of having an unfavorable tuning of the penalty parameter is highly desirable.

2. As is the case for the proximal bundle method, information from the SSC test can be used to dynamically fine-tune the value of the penalty parameter ρ. For the convergence analysis culminating in Proposition 5.18 to remain valid, it is expected that if ρ does vary with iteration k, that it should stabilise to some positive value.

3. While not enforcing the SSC can adversely affect the growth trend in the Lagrangian bound, the use of a SSC parameter γ value that is too large can have a similar effect for the tail-end values. This is most clearly seen in the Figure 5.1 DCAP-233-500 $\rho = 50$ and $\rho = 100$ plots. In these plots, the growth in Lagrangian bound value is noticeably stunted in the tail-end iterations for the larger $\gamma = 0.5$ value as compared with the smaller $\gamma = 0.125$.

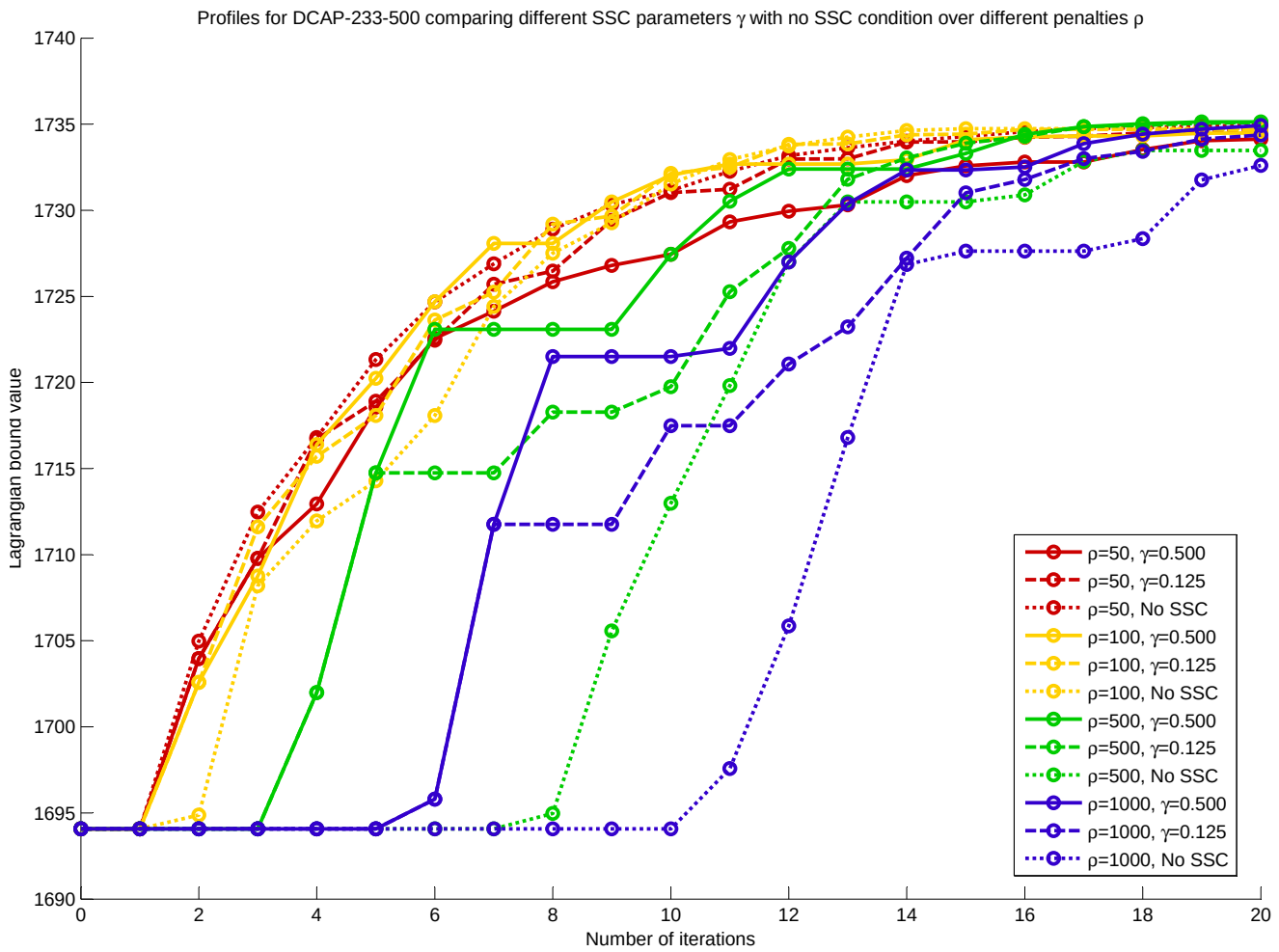


Figure 5.1: Applying SDM-GS-ALM to DCAP-233-500 using different penalties and parameterizations for the serious step condition

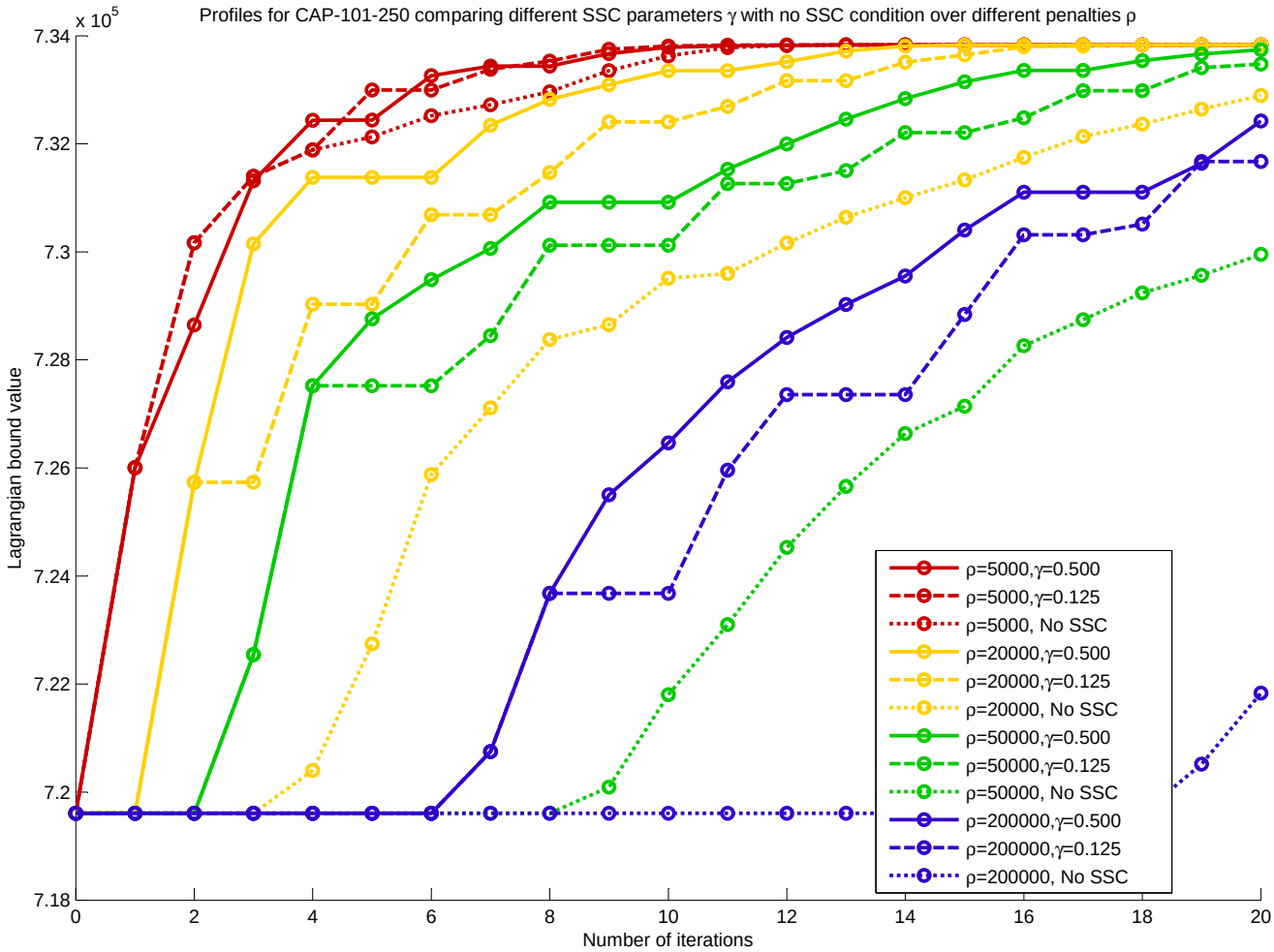


Figure 5.2: Applying SDM-GS-ALM to CAP-101-250 using different penalties and parameterizations for the serious step condition

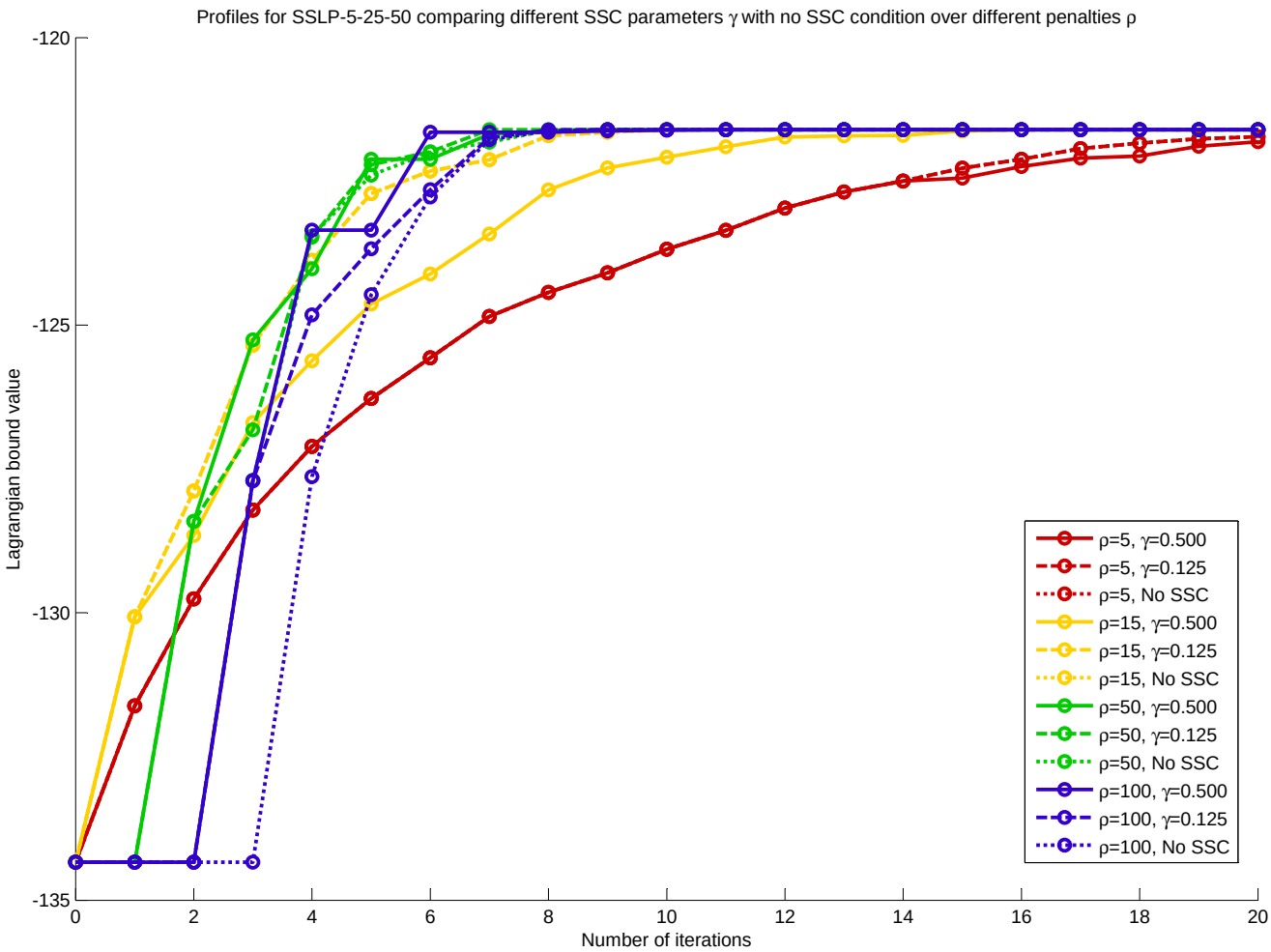


Figure 5.3: Applying SDM-GS-ALM to SSLP-5-25-50 using different penalties and parameterizations for the serious step condition

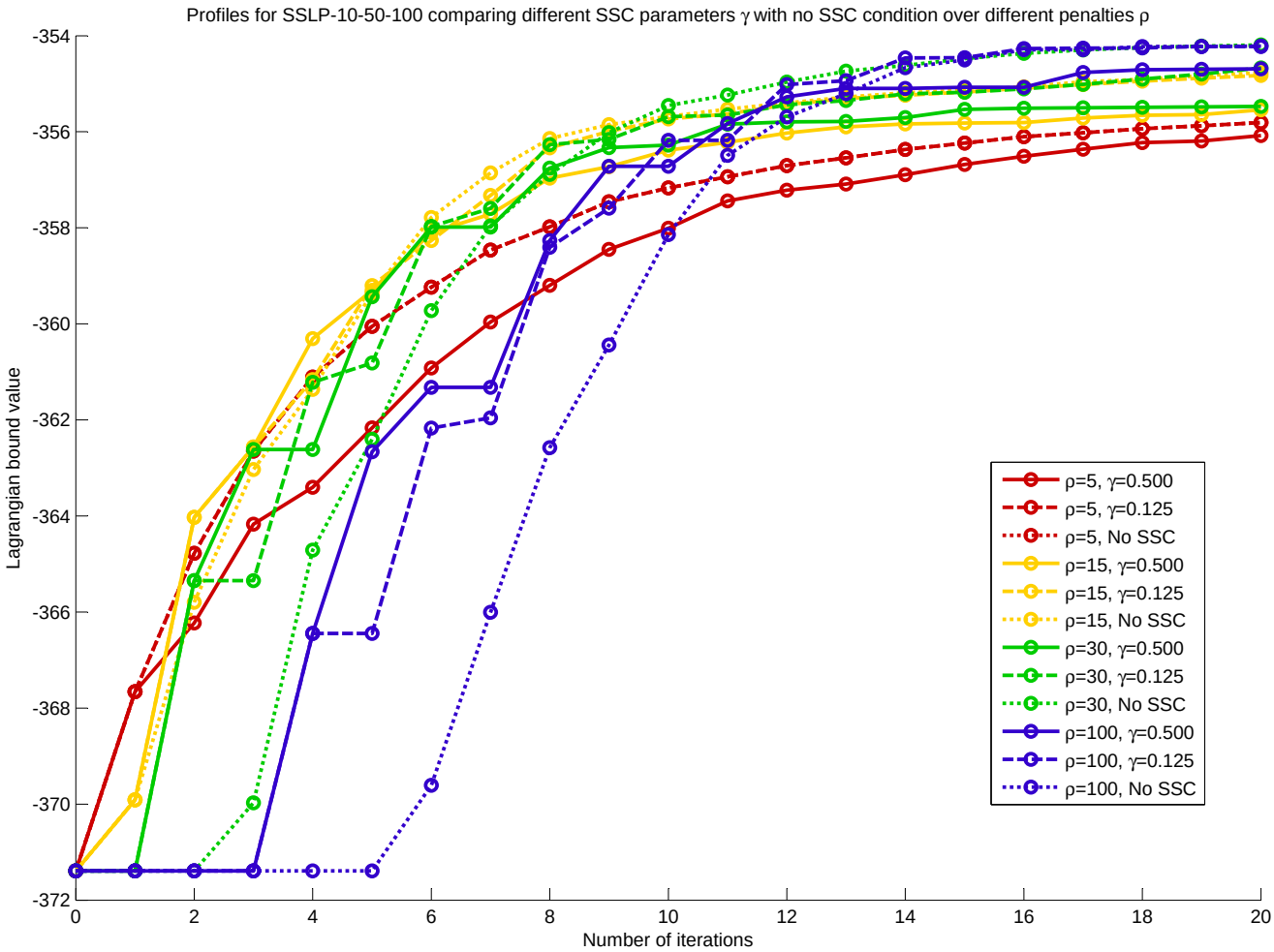


Figure 5.4: Applying SDM-GS-ALM to SSLP-10-50-100 using different penalties and parameterizations for the serious step condition

5.3.3 Benefits of Parallelisation

For the Test 2 experiments, we primarily compare the parallel speedup achieved with Algorithm 5.4 against that achieved with the enhancements to the proximal bundle method presented in [79]. Additionally, we compare the Lagrangian bound at the final iteration.

The enhancements in [79] use structure-exploiting primal-dual interior point solvers to improve the parallel efficiency of solving the proximal bundle method master problem. (The solution of this master problem is analogous to the approximated solution to problem (5.9) obtained by using the SDM-GS method in Algorithm 5.2.) The first solver is referred to by its acronym OOQP [47], while the second is PIPS-IPM [80].

In the experiments of Test 2, the underlying computing architecture and third-party software are inevitably different between our tests and those in [79]. Additionally, the termination criterion is necessarily different from that given in Step 2 of Figure 2 in [79] due to the differences in algorithms. In our tests, the termination criterion comes from Lines 9–11 of Algorithm 5.4 with $\epsilon = 10^{-6}$. We can nevertheless create a meaningful control in the tuning of the most important parameters affecting the performance of the algorithm.

1. As done in [79], we set the SSC parameter $\gamma = 0.1$, and we initialise the dual solution $\omega^0 = 0$.

2. In analogy to the possible trimming of cutting planes noted in [79], practical implementations of Algorithm 5.4 may judiciously trim the set D to improve performance. As all cuts are kept in the experiments of [79], so we also avoid trimming the expansion of D in our experiments, and so we just use the simple update rule $D \leftarrow \text{conv}(D \cup \{\widetilde{x}, \widehat{x}\})$ within Algorithm 5.2.

3. We use an update rule analogous to the one in [69] as is done in [79]. which takes the suggested form given in Line 17 of Algorithm 5.4. Initially, $\rho = 1$.

In Tables 5.1–5.2, the columns headed by OOQP and PIPS-IPM report the parallel speedup due to the use of $N = 1, 8, 16, 32$ processors, which are originally reported in Figure 2 of [79]. If, given the use of N processors, T_N denotes the total wall clock time (in seconds) divided by number of iterations, then we compute the parallel speedup as T_1/T_N. For the computational experiments with Algorithm 5.4, we compute each table entry T_1/T_N after taking, from five identically parameterised experiments, 1) the minimum T_1 value, and 2) the average T_N, $N > 1$, value. The column headed by SDM-GS1-ALM presents the parallel speedup values for the application of Algorithm 5.4 with $t_{max} = 1$. The column headed by SDM-GS5-ALM is analogous, with $t_{max} = 5$. The total wall clock time per iteration values used to compute the ratios T_1/T_N are provided in Tables 5.3–5.4, accounting for taking the minimum ($N = 1$) or average ($N > 1$) over the five experiments for each set of parameterizations associated with Algorithm 5.4. Also in Tables 5.1–5.2, the best Lagrangian bounds obtained for each combination of test problem and algorithm are reported.

For the two sets of experiments based on the application of Algorithm 5.4, a problem-specific maximum number of main loop iterations was set so as to make the tests as comparable with the tests in [79] as possible. These results are reported in Tables 5.3–5.4. For each entry (A, B) of these

tables, A provides the number of iterations at termination, and B provides the average wall clock time (in seconds) per iteration.

We draw the following conclusions from the results of the Test 2 experiments reported in Tables 5.1–5.2.

1. The improvement in parallel speedup (SDM-GS-ALM columns) over either OOQP or PIPS-IPM is evident for all problems except for the one with the fewest number of scenarios (SSLP 5-25-100).

2. Slightly inferior final Lagrange bounds reported for SDM-GS1-ALM ($t_{max} = 1$) are evident. This deficit is improved by using SDM-GS with $t_{max} = 5$, as done for the SDM-GS5-ALM experiments. But even these bounds are usually not as good as the bounds obtained with OOQP or PIPS-IPM; this is due to their more exact solving of the master problem instances. This suggests that as the iterations $k \geqslant 1$ increase, it is advantageous to solve the continuous master problem with SDM-GS iterations using larger t_{max} values.

3. Interestingly, parallel speedup is enhanced for SDM-GS5-ALM over SDM-GS1-ALM; although the latter yields lower average total wall clock time per iteration, the proportion of efficiently parallelizable work seems to increase in the former.

For Test 2, we also tested the performance of Algorithm 5.4 on the SSLP 10-50-2000 problem, which is of substantially larger scale than the other test problems considered in this chapter. Using $N = 1, 2, 4, 8, 16, 32, 64$ processors, we see very good speedup, which suggests the realised benefit of distributing the use of memory. We also see that for such large-scale problems, the additional cost in time of performing more inner loop Gauss-Seidel iterations (larger t_{max}) becomes marginal, since the cost of solving the mixed-integer linear subproblems takes a larger share of the computational time.

	Speedup for SSLP 5-25-100			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	5.54	5.23	4.38	4.78
16	8.89	8.55	6.61	7.07
32	11.69	11.94	8.19	8.89
Lagr. Value	-127.37	-127.37	-127.71	-127.58

	Speedup for SSLP 10-50-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	2.64	2.80	6.87	6.95
16	2.70	2.92	12.95	12.84
32	2.98	3.40	21.67	20.98
Lagr. Value	-349.14	-349.14	-349.48	-349.14

	Speedup for SSLP 10-50-2000	
No. Proc.	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00
2	2.34	2.34
4	4.81	4.83
8	9.29	9.25
16	18.69	18.48
32	34.63	35.10
64	60.59	60.93
Lagr. Value	-348.35	-347.75

Table 5.1: Comparing speedup and final best Lagrangian bound of SDM-GS-ALM, OOQP and PIPS-IPM for SSLP instances

	Speedup for DCAP 233-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	2.44	5.32	6.88	8.11
16	2.81	8.15	13.28	15.65
32	1.63	10.25	23.42	27.40
Lagr. Value	1736.68	1736.68	1734.99	1736.02

	Speedup for DCAP 243-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	2.85	5.71	6.51	7.61
16	3.59	5.85	12.28	14.44
32	1.98	6.44	21.99	25.25
Lagr. Value	2165.48	2165.50	2162.58	2164.48

	Speedup for DCAP 332-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	2.03	5.56	6.83	8.50
16	2.33	5.00	12.84	16.20
32	1.21	6.61	21.83	23.48
Lagr. Value	1587.44	1587.44	1584.77	1586.11

	Speedup for DCAP 342-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	1.00	1.00	1.00	1.00
8	2.45	3.78	7.16	8.25
16	2.71	4.36	12.95	15.49
32	1.84	4.64	22.41	26.93
Lagr. Value	1902.84	1903.21	1900.81	1901.90

Table 5.2: Comparing speedup and final best Lagrangian bound of SDM-GS-ALM, OOQP and PIPS-IPM for DCAP instances

	SSLP 5-25-100			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(8, 6.31)	(8, 6.33)	(8,3.22)	(8,3.32)
8	(8, 1.14)	(8, 1.21)	(8,0.74)	(8,0.69)
16	(8, 0.71)	(8, 0.74)	(8,0.49)	(8,0.47)
32	(8, 0.54)	(8, 0.53)	(8,0.39)	(8,0.37)

	SSLP 10-50-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(26, 3301)	(22, 2939)	(30, 168.80)	(29, 171.85)
8	(31, 1252)	(24, 1049)	(30, 24.58)	(29, 24.71)
16	(27, 1224)	(28, 1005)	(30, 13.04)	(30, 13.39)
32	(31, 1106)	(27, 865)	(30, 7.79)	(28, 8.19)

	SSLP 10-50-2000	
No. Proc.	SDM-GS1-ALM	SDM-GS5-ALM
1	(20, 840.69)	(20, 845.77)
2	(20, 359.63)	(20, 361.40)
4	(20, 174.83)	(20, 175.03)
8	(20, 90.51)	(20, 91.46)
16	(20, 44.98)	(20, 45.76)
32	(20, 24.27)	(20, 24.09)
64	(20, 13.87)	(20, 13.88)

Table 5.3: Comparing iteration count and runtime of SDM-GS-ALM, OOQP and PIPS-IPM for SSLP instances

	DCAP 233-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(68, 16.15)	(66, 12.71)	(68, 3.67)	(68, 5.21)
8	(68, 6.62)	(70, 2.39)	(68, 0.53)	(68, 0.64)
16	(68, 5.75)	(73, 1.56)	(68, 0.28)	(68, 0.33)
32	(68, 9.91)	(70, 1.24)	(68, 0.16)	(68, 0.19)
	DCAP 243-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(57, 14.37)	(57, 12.11)	(57, 3.72)	(57, 5.11)
8	(57, 5.04)	(58, 2.12)	(57, 0.57)	(57, 0.67)
16	(57, 4.00)	(59, 2.07)	(57, 0.30)	(57, 0.35)
32	(57, 7.26)	(59, 1.88)	(57, 0.17)	(57, 0.20)
	DCAP 332-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(82, 13.51)	(80, 9.45)	(82, 2.94)	(82, 4.85)
8	(82, 6.65)	(79, 1.70)	(81, 0.43)	(82, 0.57)
16	(82, 5.81)	(80, 1.89)	(81, 0.23)	(82, 0.30)
32	(82, 11.20)	(77, 1.43)	(82, 0.13)	(82, 0.21)
	DCAP 342-500			
No. Proc.	OOQP	PIPS-IPM	SDM-GS1-ALM	SDM-GS5-ALM
1	(59, 14.78)	(71, 12.07)	(59, 3.80)	(59, 5.57)
8	(59, 6.03)	(67, 3.19)	(59, 0.53)	(59, 0.68)
16	(59, 5.46)	(56, 2.77)	(59, 0.29)	(59, 0.36)
32	(59, 8.05)	(62, 2.60)	(59, 0.17)	(59, 0.21)

Table 5.4: Comparing iteration count and runtime of SDM-GS-ALM, OOQP and PIPS-IPM for DCAP instances

5.4 Conclusions

The developments in this chapter are motivated by the goal of improving the efficiency of parallelisation applied to iterative approaches for solving the Lagrangian dual problem of large scale optimization problems. These problems have nonlinear convex differentiable objective f, decomposable nonconvex constraint set X, and nondecomposable affine constraint set $Qx = z$ to which Lagrangian relaxation is applied. Problems of such a form include the split variable extensive form of mixed-integer linear stochastic programs as a special case. Implicitly, our approach refers to the convex hull $\text{conv}(X)$ of X, and the assumed lack of known description of $\text{conv}(X)$ needs to be addressed. Proximal bundle methods (alternatively in the form of the proximal simplicial decomposition method or stabilized column generation) are well-known for addressing the latter issue. In the former issue, that of exploiting the large scale structure to apply parallel computation efficiently, we develop a modified augmented Lagrangian (AL) method with approximate subproblem solutions that incorporates ideas from the proximal bundle method.

The approximation of subproblem solutions is based on an iterative approach that integrates ideas from the simplicial decomposition method (SDM) (for constructing inner approximations of $\text{conv}(X)$) and the nonlinear block Gauss-Seidel method. It is the latter Gauss-Seidel aspect that is primarily responsible for enhancing the parallel efficiency that is observed in the numerical experiments. While convergence analysis of the integrated SDM-GS approach may be derived from slight modifications to results in [20], for the sake of completeness and explicitness, Section 5.2.4 contains a proof of optimal convergence of SDM-GS as it is applied within our algorithm under a standard set of conditions. A distinction between so-called "serious" steps and "null" steps, in analogy to the proximal bundle method, is also recovered. Once these aspects are successfully integrated, then the contribution is complete, where the beneficial stabilization associated with proximal point methods and the ability to apply parallelisation more efficiently are both realised. The resulting algorithm developed in this chapter is referred to as SDM-GS-ALM, which has similar functionality to the alternating direction method of multipliers (ADMM).

We performed numerical tests of two sorts. In Test 1, we examined the impact of varying the serious step condition parameter. We found that parameterizations that effect more stringent serious step conditions seem to have the effect of mitigating the early iteration instability due to

penalty parameters that are too large. At the same time, the more stringent serious step condition parameterizations seemed to result in slower convergence to dual optimality in the tail-end. As is the case for proximal bundle methods, information obtained in the serious step condition tests may be used to beneficially adjust the proximal term penalty coefficient in early iterations.

In Test 2, we examined the efficiency of parallelisation, measured by the speedup ratio, due to the use of the SDM-GS-ALM, compared versus pre-existing implementations of the proximal bundle method that use structure exploiting primal dual interior point methods to improve parallel efficiency. We saw in these results a promising increase in parallel efficiency due to the use of SDM-GS-ALM, where the increase in parallel efficiency is attributed primarily to the successful incorporation of Gauss-Seidel iterations. The results of the last problem tested, SSLP 10-50-2000, additionally suggested a benefit due to the ability of SDM-GS-ALM to distribute not just the workload, but also the use of memory. The vector of auxiliary variables z is the only substantial block of data that needs to be stored and modified by all processors. In the context of stochastic optimization problems, this represents a modest communication bottleneck in proportion to the number of first-stage variables for two-stage problems, while for multistage problems, the amount of such data that must be stored by every processor and modified by parallel communication can increase exponentially with the number of stages.

Chapter 6

Penalty-based Gauss-Seidel Heuristic Method

6.1 Introduction

6.1.1 Problem Formulation

In this chapter we will develop and demonstrate an algorithm for finding high-quality primal solutions to two-stage stochastic mixed integer linear programs, as represented by (2.4) and repeated here for reference:

$$\begin{aligned} \zeta^{SIP} = & \min_{x,y} c^T x + \sum_{s \in S} \left[p_s d_s^T y_s \right] \\ & \text{s.t. } x \in X \\ & \quad y_s \in Y_s(x) \quad \forall s \in S \end{aligned} \tag{6.1}$$

The sets of feasible first-stage decisions X, and second-stage decisions $Y_s(x)$ for scenario s, are defined as follows:

$$X := \left\{ x \in \mathbb{R}^{n-q} \times \mathbb{Z}^q \mid Ax \leqslant b \right\}$$

$$Y_s(x) := \left\{ y \in \mathbb{R}^{m-r} \times \mathbb{Z}^r \mid T_s x + W_s y_s \leqslant h_s \right\}$$

This notation differs from that of the previous chapters in that the first- and second-stage feasible sets are separated.

Assumption 6.1 *ζ^{SIP} is feasible, the optimal value of ζ^{SIP} is bounded, and p_s, c, d_s, A, b, T_s, W_s, and h_s are vectors or matrices (as appropriate) of rational numbers.*

Similarly to (2.16) we can reformulate this problem using non-anticipativity constraints as follows:

$$\begin{aligned} \zeta^{SIP} = & \min_{\boldsymbol{x},\boldsymbol{y},z} \sum_{s\in S} p_s(c^T x_s + d_s^T y_s) \\ & \text{s.t. } x_s \in X \quad \forall s \in S \\ & \quad y_s \in Y_s(x_s) \quad \forall s \in S \\ & \quad x_s - z = 0 \quad \forall s \in S \end{aligned} \tag{6.2}$$

With the exception of the non-anticipativity constraints $x_s - z = 0$ this problem is separable by scenario s. A natural approach to solving this problem is to relax this constraint in some way so as to obtain full separability.

Since ζ^{SIP} contains integer variables, applying ordinary Lagrangian relaxation to the non-anticipativity constraints does not result in strong duality. Under some conditions, strong duality may be retrieved through augmented Lagrangian relaxation of these constraints. The augmented Lagrangian corresponding to ζ^{SIP} is

$$L_\rho(\boldsymbol{x}, \boldsymbol{y}, z, \boldsymbol{\lambda}) = \sum_{s\in S} \left[p_s(c^T x_s + d_s^T y_s) + \lambda_s^T (x_s - z) + \psi_s^\rho (x_s - z) \right] \tag{6.3}$$

where ψ is an appropriate penalty function dependent on the scenario s and penalty parameter ρ. The penalty parameter may be a scalar or a vector, depending on the penalty function. A linear increase in the components of ρ should result in a linear increase in the value of ψ. The properties of an 'appropriate' penalty function, as defined by Feizollahi et al. [37, Assumption 2], are stated in Assumption 6.2.

Assumption 6.2 *ψ satisfies the following requirements:*

- *ψ vanishes at zero (i.e. $\psi(0) = 0$)*
- *$\psi(u)$ is strictly positive for all non-zero u.*
- *ψ is lower semi-continuous.*
- *ψ is level-bounded.*
- $\lim_{\delta\to 0} \boldsymbol{diam}\{u \mid \psi(u) \leqslant \delta\} = 0$.

By making the variable substitution $\omega_s = \frac{\lambda_s}{p_s}$, (6.3) may be rewritten in the simpler form

$$L_\rho(\boldsymbol{x}, \boldsymbol{y}, z, \boldsymbol{\omega}) = \sum_{s \in S} \left[p_s(c^T x_s + d_s^T y_s + \omega_s^T (x_s - z)) + \psi_s^\rho (x_s - z) \right]. \tag{6.4}$$

In order to guarantee that the augmented Lagrangian dual problem is bounded the dual feasibility condition $\omega \in \Omega := \left\{ \omega \mid \sum_{s \in S} p_s^T \omega_s = 0 \right\}$ must be enforced. Under this condition the $\sum_{s \in S} p_s \omega_s^T z$ term is equal to zero and the augmented Lagrangian may be expressed in the form

$$L_\rho(\boldsymbol{x}, \boldsymbol{y}, z, \boldsymbol{\omega}) = \sum_{s \in S} p_s \left[(c^T + \omega_s^T) x_s + d_s^T y_s + \psi_s^\rho (x_s - z) \right]. \tag{6.5}$$

The corresponding augmented Lagrangian dual function is

$$\begin{aligned} \zeta_\rho^{LR+}(\boldsymbol{\omega}) = & \min_{\boldsymbol{x}, \boldsymbol{y}, z} L_\rho(\boldsymbol{x}, \boldsymbol{y}, z, \boldsymbol{\omega}) \\ & \text{s.t. } x_s \in X \quad \forall s \in S \\ & y_s \in Y_s(x_s) \quad \forall s \in S \end{aligned} \tag{6.6}$$

and the corresponding augmented Lagrangian dual problem is:

$$\zeta_\rho^{LD+} = \max_{\omega \in \Omega} \zeta_\rho^{LR+}(\boldsymbol{\omega}) \tag{6.7}$$

6.1.2 Conditions for Strong Duality

Under some conditions on the penalty parameter ρ and augmenting function ψ, the augmented Lagrangian dual is strong even when applied to problems with integer variables. For clarity some results from Feizollahi et al. [37], adapted to the problem structure considered in this chapter, are stated below.

Theorem 6.1 *[37, Theorem 2] Suppose that Assumption 6.1 on ζ^{SIP} as defined in* (6.1) *and Assumption 6.2 on the penalty function ψ used in* (6.7) *hold. Then* $\sup_{\rho > 0} \zeta_\rho^{LD+} = \zeta^{SIP}$.

Proof. Apply [37, Theorem 2] to (6.1). ■

Theorem 6.1 is theoretically interesting but its practical utility is limited, since it relies on taking the supremum of the Lagrangian dual over all $\rho > 0$. Typically this means that ρ must go to $+\infty$, which is computationally infeasible; very large penalty parameters generally lead to intractable numerical issues.

A more practically useful result may be obtained with more restrictive assumptions on the penalty function.

Assumption 6.3 *ψ satisfies the following requirements:*

1. $\psi(0) = 0$
2. $\psi(u) \geqslant \delta > 0, \forall u \notin V$
3. $\psi(u) \geqslant \gamma ||u||_\infty, \forall u \in V$

for some open neighbourhood V of 0 and positive scalars $\delta, \gamma > 0$.

Theorem 6.2 *[37, Theorem 5] Suppose that (6.1) satisfies Assumption 6.1. If $\psi : \prod_{s \in S} \mathbb{R}^{n_x} \mapsto \mathbb{R}$ is a summed augmenting function $\psi(u) := \sum_{s \in S} \psi_\rho^s(u_s)$ for (6.6) such that ψ satisfies Assumption 6.3, then there exists a finite penalty parameter ρ such that $\zeta_\rho^{LD+} = \zeta_\rho^{LR+}(\omega_{LP}) = \zeta^{SIP}$, for ω_{LP} (an optimal multiplier of the linear programming relaxation of the non-anticipativity conditions).*

Proof. Apply [37, Theorem 5] to (6.1). ■

Applying Theorem 6.2 to obtain strong duality only requires a finite choice of penalty parameter. However, the restriction imposed by Assumption 6.3 introduces computational difficulties of its own. In particular, the third part of Assumption 6.3 implies that ψ is non-smooth at zero, which means that we can no longer apply algorithms which require smoothness of the objective function to solve the minimisation problem in the augmented Lagrangian dual function.

Remark 6.3 *One may see with little difficulty that the proof of [37, Theorem 5] does not rely on the setting of $\omega = \omega_{LP}$. Indeed one can show that for any $\omega \in \Omega$ there still exists a finite penalty parameter such that Theorem 6.2 holds true. To see this, considering the notation of [37], one needs to replace $\bar{\lambda}_{LP}$ with a generic (dual feasible) λ and ζ^{LP} in [37, Theorem 5] with the Lagrangian dual function $\zeta^{LR}(\lambda)$ defined earlier in [37]. Also, adjust the definition of $\hat{\rho}$ in [37, Theorem 5] accordingly.*

Feizollahi et al. observed that any norm function satisfies the requirements of Assumption 6.3 and therefore that employing one as an augmenting function would result in strong duality as per Theorem 6.2. However, the class of norm functions is lacking in flexibility, since deviation from consensus in each direction is penalised equally. In Section 6.2 a more general class of penalty functions satisfying Assumption 6.3 will be defined.

6.2 Penalty Functions derived from Positive Bases

6.2.1 Positive Bases

We will use the concept of a **positive basis**, sometimes also called a **non-negative basis**, to define a more general class of penalty functions than the class of norm functions.

Definition 6.4 *A point $u \in \mathbb{R}^m$ is a* **positive combination** *of a set of vectors $N = \{\mathbf{n}_1, \ldots, \mathbf{n}_l\} \subset \mathbb{R}^m$ if there exists $\alpha_i \geqslant 0$ for $i = 1, \ldots, l$ such that $u = \sum_{i=1}^{l} \alpha_i \mathbf{n}_i$.*

Definition 6.5 *A set of vectors $N = \{\mathbf{n}_1, \ldots, \mathbf{n}_l\} \subset \mathbb{R}^m$ is* **positively independent** *if for all $i \in \{1, \ldots, l\}$, $\mathbf{n}_i$ is not a positive combination of $N \backslash \mathbf{n}_i$.*

Definition 6.6 *A set of vectors $N = \{\mathbf{n}_1, \ldots, \mathbf{n}_l\} \subset \mathbb{R}^m$* **positively spans** *a set $S \subset \mathbb{R}^m$ if all points in S are positive combinations of N.*

Definition 6.7 *A set of vectors $N = \{\mathbf{n}_1, \ldots, \mathbf{n}_l\} \subset \mathbb{R}^m$ is a* **positive basis** *for $\mathbb{R}^m$ if and only if N is positively independent and N positively spans $\mathbb{R}^m$.*

For purposes of brevity define $\{e_i\}_{i=1,\ldots,m}$ as the standard basis of $\mathbb{R}^m$, as follows:

$$
\begin{aligned}
e_1 &= (1, 0, \ldots, 0) \\
e_2 &= (0, 1, 0, \ldots, 0) \\
&\ldots \\
e_m &= (0, \ldots, 0, 1)
\end{aligned}
$$

The following sets of vectors are examples of positive bases for $\mathbb{R}^m$.

- The vertices of a m-simplex (generalised tetrahedron), centred at the origin.
- The set of vectors $\{+e_i\}_{i=1}^m \cup \{\sum_{i=1}^m -e_i\}$.
- The set of vectors $\{\pm e_i\}_{i=1}^m$.

We will require the following result:

Theorem 6.8 *([28, Theorem 3.1]) $\{\mathbf{n}_1, \ldots, \mathbf{n}_l\}$ positively spans $\mathbb{R}^m$ if and only if for every non-zero $u \in \mathbb{R}^m$ there exists an index i such that $u \cdot \mathbf{n}_i > 0$.*

6.2.2 Generalising the 1-Norm and ∞-Norm

The 1-norm and ∞-norm are particularly amenable to alternate representations because they are polyhedral and may be represented as the maxima of a finite number of hyperplanes. These two norms may be represented using the positive basis $\{\pm e_i\}_{i=1}^m$ in the following ways.

$$\|u\|_\infty = \max_{i=1,\ldots,m} \{\pm e_i^\top u\} \tag{6.8}$$

$$\|u\|_1 = \sum_{i=1}^m \max\{+e_i^\top u, 0\} + \sum_{i=1}^m \max\{-e_i^\top u, 0\} \tag{6.9}$$

$$\text{or alternatively } \|u\|_1 = \sum_{i=1}^m \max\left\{\nu_i^\top u : \nu_i \in \{+e_i, -e_i\}\right\}. \tag{6.10}$$

The representation of the 1-norm given by (6.10) cannot be generalised to an arbitrary positive basis, since it is dependent on each basis vector being matched with a negative multiple of itself. The representations of the ∞-norm and 1-norm given by (6.8) and (6.9) respectively may be generalised to an arbitrary positive basis $N = \{\mathbf{n}_1, \ldots, \mathbf{n}_l\}$ as follows:

$$\psi_\infty^N(u) := \max_{i=1,\ldots,l} \{\mathbf{n}_i^\top u\} \quad \text{and} \tag{6.11}$$

$$\psi_1^N(u) := \sum_{i=1}^l \max\{\mathbf{n}_i^\top u, 0\}. \tag{6.12}$$

For a general positive basis N, these functions are not norms because they are not necessarily absolutely homogeneous. As an illustrative example, the set of one-dimensional vectors $N = \{(-3), (1)\}$ is a positive basis for $\mathbb{R}^1$, but for $u > 0$ we have:

$$\psi_\infty^N((-1)u) = \max\{-3(-u), 1(-u)\} = 3u \neq u = \max -3(u), 1(u) = |-1|\, \psi_\infty^N(u)$$

and

$$\psi_1^N((-1)u) = \max\{-3(-u), 0\} + \max\{1(-u), 0\} = 3u$$
$$\neq u = \max\{-3(u), 0\} + \max\{1(u), 0\} = |-1|\, \psi_1^N(u)$$

However, $\psi_\infty^N(u)$ and $\psi_1^N(u)$ do share some useful properties with norms.

Lemma 6.9 *$\psi_\infty^N(u)$ and $\psi_1^N(u)$ are positively homogeneous, i.e. $\psi(\alpha u) = \alpha\psi(u)$ for all u and all $\alpha \geqslant 0$.*

Proof. Since α is non-negative it can be extracted as a constant factor from the maximums, as follows:

$$\psi_\infty^N(\alpha u) = \max_{i=1,\ldots,l}\{\alpha \mathbf{n}_i^\top u\} = \alpha \max_{i=1,\ldots,l}\{\mathbf{n}_i^\top u\} = \alpha\psi_\infty^N(u)$$

$$\psi_1^N(\alpha u) = \sum_{i=1}^{l} \max\{\alpha \mathbf{n}_i^\top u, 0\} = \alpha \sum_{i=1}^{l} \max\{\mathbf{n}_i^\top u, 0\} = \alpha\psi_1^N(u)$$

■

Lemma 6.10 *$\psi_\infty^N(u)$ and $\psi_1^N(u)$ are strictly positive for all non-zero u.*

Proof. By Theorem 6.8 at least one $\mathbf{n}_i^\top u$ term is positive for any non-zero u. Therefore, the maximisation term in $\psi_\infty^N(u)$ and at least one of the maximisation terms in $\psi_1^N(u)$ is always positive for any non-zero u. Since each of the maximisation terms in $\psi_1^N(u)$ is bounded below by zero this is sufficient to show that both functions are strictly positive for non-zero u. ■

Lemma 6.11 *$\psi_\infty^N(u)$ and $\psi_1^N(u)$ separate points, i.e. $\psi(u) = 0$ implies $u = 0$.*

Proof. Follows directly from Lemma 6.10 and the observation that ψ vanishes at zero, i.e. $\psi_\infty^N(0) = \psi_1^N(0) = 0$. ■

Lemma 6.12 *$\psi_\infty^N(u)$ and $\psi_1^N(u)$ are finite valued.*

Proof. The maximum or sum of a finite number of finite valued functions is finite valued. ■

Lemma 6.13 *$\psi_\infty^N(u)$ and $\psi_1^N(u)$ are sub-additive, i.e. $\psi(u) \leqslant \psi(u) + \psi(v)$.*

Proof.

$$\psi_\infty^N(u+v) = \max_{i=1,\ldots,l}\{\mathbf{n}_i^\top(u+v)\} \leqslant \max_{i=1,\ldots,l}\{\mathbf{n}_i^\top(u)\} + \max_{i=1,\ldots,l}\{\mathbf{n}_i^\top(v)\} = \psi_\infty^N(u) + \psi_\infty^N(v)$$

$$\psi_1^N(u+v) = \sum_{i=1}^{l} \max\{\mathbf{n}_i^\top(u+v), 0\} \leqslant \sum_{i=1}^{l} \max\{\mathbf{n}_i^\top(u), 0\} + \sum_{i=1}^{l} \max\{\mathbf{n}_i^\top(v), 0\} = \psi_1^N(u) + \psi_1^N(v)$$

■

Lemma 6.14 *$\psi_\infty^N(u)$ and $\psi_1^N(u)$ are coercive, i.e. $\psi(u) \to +\infty$ as $\|u\| \to +\infty$.*

Proof. Follows directly from positive homogeneity of $\psi_1^N(u)$ and $\psi_\infty^N(u)$. ■

In the next section we will see that this weaker set of properties is sufficient for $\psi_\infty^N(u)$ and $\psi_1^N(u)$ to yield strong duality when employed as Lagrangian dual penalty functions.

6.2.3 Strong Lagrangian Duality using Norm-like Penalties

The following results demonstrate that ψ_∞^N and ψ_1^N satisfy the conditions given in Assumption 6.3, and therefore than when they are employed as penalty functions Theorem 6.2 may be applied.

Lemma 6.15 *If two functions $\psi_A : \mathbb{R}^m \to \mathbb{R}$ and $\psi_B : \mathbb{R}^m \to \mathbb{R}$ are positive homogeneous, continuous, and strictly positive for all $u \neq 0$ then there exists a finite $\gamma > 0$ such that*

$$\psi_A(u) \geqslant \gamma\psi_B(u) \quad \textit{for all } u \in \mathbb{R}^m.$$

Proof. Since they are positive homogeneous, ψ_A and ψ_B vanish at zero and so the required property trivially holds with equality at $u = 0$. To obtain the required inequality for nonzero u, set $V = \{u : ||u|| = 1\}$ (where $||\cdot||$ is any norm) and take $\alpha = \min_{u\in V} \psi_A(u)$ and $\beta = \max_{u\in V} \psi_B(u)$. Since ψ_A and ψ_B are continuous and defined on the closed and bounded set V, by the Extreme Value Theorem these extrema exist and are attained by their respective functions. Since these functions are strictly positive and finite valued on V, and they attain their extrema, α and β both strictly positive and finite.

For any point $u \in \mathbb{R}^m\backslash\{0\}$, $||u||$ is strictly positive and the point $\frac{u}{||u||}$ is in V. Therefore, by the positive homogeneity of $\psi_A(u)$ and $\psi_B(u)$ we have

$$\psi_A(u) = ||u||\psi_A\left(\frac{u}{||u||}\right) \geqslant \alpha||u||$$

and

$$\beta||u|| \geqslant ||u||\psi_B\left(\frac{u}{||u||}\right) = \psi_B(u).$$

Let $\gamma = \alpha/\beta$. Since α and β are strictly positive and finite, γ is also strictly positive and finite. The required inequality follows:

$$\psi_A(u) \geqslant \alpha||u|| = \frac{\alpha}{\beta}\beta||u|| = \gamma\beta||u|| \geqslant \gamma\psi_B(u).$$

■

Proposition 6.16 *For any positive basis N, the augmenting functions ψ_∞^N and ψ_1^N given in* (6.11) *and* (6.12) *respectively satisfy the conditions given in Assumption 6.3.*

Proof. Let $V = B_\varepsilon^\infty(0)$ be an open ball in the infinity norm with radius $\varepsilon > 0$ centred at the origin. This is an appropriate open neighbourhood of 0 for the purposes of Conditions 2 and 3 of Assumption 6.3.

Condition 1: $\psi(0) = 0$.

If $u = 0$ then $\mathbf{n}_i^\top u = 0$ and therefore $\psi_\infty^N(u) = 0$ and $\psi_1^N(u) = 0$, as required.

Condition 2: $\psi(u) \geqslant \delta > 0, \forall u \notin V$ for some positive scalar δ.

Using Theorem 6.8, for any $u \neq 0$ we have some i such that $\mathbf{n}_i^\top u > 0$ and hence $\psi_\infty^N(u) > 0$. Now define

$$\delta \;:=\; \min_u \{ \max_{i=1,\ldots,l} \{\mathbf{n}_i^\top u\} \mid \|u\|_\infty = \varepsilon \} > 0, \tag{6.13}$$

where $\delta > 0$ follows from the compactness of the ε- ball, the continuity of $u \mapsto \max_{i=1,\ldots,l}\{\mathbf{n}_i^\top u\}$, and Theorem 6.8. For any $u \notin V$, the point $v := \varepsilon \frac{u}{\|u\|_\infty}$ is in V and hence $\psi_\infty^N(v) \geqslant \delta > 0$. Using the positive homogeneity property we have

$$\frac{\varepsilon}{\|u\|_\infty} \psi_\infty^N(u) \geqslant \delta > 0$$

$$\text{and so} \qquad \psi_\infty^N(u) \geqslant \delta \frac{\|u\|_\infty}{\varepsilon} \geqslant \delta > 0,$$

using the fact that $u \notin V$ means $\|u\|_\infty \geqslant \varepsilon$. This is the required inequality for ψ_∞^N.

Apply Lemma 6.15 to deduce that there exists a $\eta > 0$ such that:

$$\psi_1^N(u) \geqslant \eta \psi_\infty^N(u) \geqslant \eta\delta > 0 \text{ for all } u \notin V.$$

$\eta\delta$ is also a positive scalar and so this is the required inequality for ψ_1^N.

Condition 3: $\psi(u) \geqslant \gamma ||u||_\infty, \forall u \in V$ for some positive scalar γ.

The property holds trivially for $u = 0$. For any $u \in V \backslash \{0\}$, the point $v := \varepsilon \frac{u}{\|u\|_\infty}$ is in V and using the same δ as defined in (6.13) we have

$$\frac{\varepsilon}{\|u\|_\infty} \psi_\infty^N(u) \geqslant \delta > 0$$

$$\text{and so} \qquad \psi_\infty^N(u) \geqslant \delta \frac{\|u\|_\infty}{\varepsilon} \geqslant \frac{\delta}{\varepsilon} \|u\|_\infty > 0,$$

and so we may place $\gamma := \frac{\delta}{\varepsilon} > 0$. This is the required inequality for ψ_∞^N.

As above, apply Lemma 6.15 to deduce that there exists a $\eta > 0$ such that:

$$\psi_1^N(u) \geqslant \eta \psi_\infty^N(u) \geqslant \eta\gamma \|u\|_\infty > 0.$$

$\eta\gamma$ is also a positive scalar and so this is the required inequality for ψ_1^N. ■

Corollary 6.17 *Suppose that ζ^{SIP} (as defined in* (6.1)*) satisfies Assumption 6.1, and that the penalty function ψ appearing in $z_\rho^{LR+}(\boldsymbol{\omega})$ and ζ_ρ^{LD+} (as defined in* (6.6) *and* (6.7) *respectively) has the form of* (6.11) *or* (6.12)*. Then, for each $\omega \in \Omega$, there is a finite $\rho > 0$ for which we have*

$$\zeta_\rho^{LD+} = z_\rho^{LR+}(\boldsymbol{\omega}) = \zeta^{SIP} \tag{6.14}$$

Proof. Equalities (6.14) follow directly from Theorem 6.2, Remark 6.3, and Proposition 6.16. ■

This result may be generalised to an even wider class of penalty functions.

Remark 6.18 *Consider a positive basis $N = \{n_1, \ldots, n_l\}$ and the functions*

$$g_i(u) := \max\{n_i^\top u, 0\}.$$

Each function $g_i(u)$ is non-negative, positive homogeneous and finite valued, and these properties are preserved if multiple g_is are summed, or their maximum is taken. By Theorem 6.8, for any non-zero u there exists an index $i \in \{1, \ldots, l\}$ such that $g_i(u)$ is strictly positive. Therefore, if every one of the g_i functions is combined using a combination of summation and/or maximisation, the resulting function $g(u)$ will be strictly positive for all non-zero u. Applying Lemma 6.15 to bound g below by a positive multiple of ψ_∞^N (as ψ_1^N was treated in Proposition 6.16) shows that this function $g(u)$ satisfies the conditions of Assumption 6.3, and as such will close the duality gap if used as an augmenting function (as per Corollary 6.17).

Remark 6.19 *By using the positive basis $\{+e_i\}_{i=1}^m \cup \{\sum_{i=1}^m -e_i\}$ or similar to define an augmenting function, we can obtain penalty terms analogous to the Lagrangian terms obtained through surrogate semi-Lagrangian relaxation (see Section 2.2.2).*

6.2.4 Defining an Appropriate Penalty Function for SIP

Section 6.2.3 provides us with a large class of viable penalty functions for achieving strong duality using the augmented Lagrangian dual (6.7). The task remains to actually choose a penalty function (or clearly defined subclass of penalty functions) which is specifically appropriate for two-stage SIP problems.

The choice of penalty function is motivated by the following criteria:

- The penalty function should have the flexibility to penalise deviation of each variable from consensus to varying degrees, in reaction to the impositions placed upon each variable by the outcome scenarios. A single variable's deviation from consensus in the positive and negative directions should also be separately variable.

- As we do not assume any information about the relationship between the first-stage variables, it is more appropriate to penalise each variable for deviation from consensus individually, rather than link multiple variables together in a single penalty term. This implies that we should choose a positive basis composed of vectors with one non-zero component each, so that each basis vector (and thus each penalty term) relates to only one variable. This criteria for the penalty function should be revisited when designing a penalty function and associated algorithm to solve a specific SIP instance or instance class.

- In particular, the x_s variables which correspond to different scenarios should be penalised separately, to minimise computational difficulty.

These criteria naturally lead to the definition of a penalty function which penalises each variable linearly for deviation from consensus in the positive or negative direction, with the degree of penalisation for each variable in each direction controlled by its own penalty parameter. This penalty function is formalised as follows.

Given a vector $u := (u_s)_{s\in S} \in \prod_{s\in S} \mathbb{R}^{n_x}$ representing the deviation of each first-stage variable from the consensus, we define for each scenario s the penalty function

$$\psi_s^\rho(u_s) := \underline{\rho}_s^\top [u_s]^- + \overline{\rho}_s^\top [-u_s]^-,$$

where $\rho = (\underline{\rho}_s, \overline{\rho}_s)_{s\in S} \in \mathbb{R}_{>0}^{2n_x|S|}$ and $[v]^- := -\min\{0, v\}$ (performed component wise), where in this case $v \in \mathbb{R}^{n_x}$. The definition of the overall penalty function ψ^ρ is the sum of the scenario-specific penalty functions over all scenarios, as follows:

$$\psi^\rho(u) := \sum_{s\in S} \psi_\rho^s(u_s) = \sum_{s\in S} \underline{\rho}_s^\top [u_s]^- + \sum_{s\in S} \overline{\rho}_s^\top [-u_s]^-. \tag{6.15}$$

This penalty function has the same form as $\psi_1^N(u)$ (as defined in (6.12)) with the positive basis

$$N_\rho = \{\overline{\rho}_{s,i} e_{i+(s-1)n_x} \mid s \in S, i \in \{1, \ldots, n_x\}\} \cup \{-\underline{\rho}_{s,i} e_{i+(s-1)n_x} \mid s \in S, i \in \{1, \ldots, n_x\}\}.$$

Substituting this penalty function into the augmented Lagrangian corresponding to the problem of interest (6.5) we obtain

$$L_\rho(\boldsymbol{x}, \boldsymbol{y}, z, \boldsymbol{\omega}) = \sum_{s \in S} \left[p_s((c^T + \omega_s^T)x_s + d_s^T y_s) + \underline{\rho}_s^\top [x_s - z]^- + \overline{\rho}_s^\top [z - x_s]^- \right]. \tag{6.16}$$

As stated in Remark 6.3, for any given $\boldsymbol{\omega}$ there exists a penalty parameter $\rho > 0$ such that $\zeta^{SIP} = \zeta_\rho^{LR+}(\boldsymbol{\omega})$. In particular, for purposes of simplicity we can set $\boldsymbol{\omega} = 0$ and therefore obtain the following representation of ζ^{SIP} for some ρ:

$$\begin{aligned} \zeta^{SIP} = & \min_{\boldsymbol{x}, \boldsymbol{y}, z} \sum_{s \in S} \left[p_s(c^T x_s + d_s^T y_s) + \sum_{s \in S} \underline{\rho}_s^\top [x_s - z]^- + \sum_{s \in S} \overline{\rho}_s^\top [z - x_s]^- \right] \\ & \text{s.t. } x_s \in X \quad \forall s \in S \\ & \quad y_s \in Y_s(x_s) \quad \forall s \in S \end{aligned} \tag{6.17}$$

The only terms which are non-separable in this formulation are the z terms in the objective. In the next section we will develop a Gauss-Seidel based approach which solves this optimisation problem for $\boldsymbol{x}$ and $\boldsymbol{y}$ with z held constant (and vice versa) which circumvents this difficulty.

6.3 Algorithm Design and Theory

6.3.1 Block Gauss-Seidel Method

The block Gauss-Seidel optimisation approach applies to optimisation problems of the form

$$\min \{ f(x, y) \mid x \in X, y \in Y \} \tag{6.18}$$

where the objective function f is convex and the feasible sets X and Y are closed. The important feature of this formulation is that the variables are split into two distinct groups. Our problem of interest (6.17) can be placed in this form by separating the (x, y) and the z variables.

The basic principle of block Gauss-Seidel methods is to separate this optimisation problem into two simpler subproblems, optimising over x and y separately while holding the other set of variables constant:

$$\min \{ f(x, \hat{y}) \mid x \in X \} \tag{6.19}$$

for some $\hat{y} \in Y$ and

$$\min \{f(\hat{x}, y) \mid y \in Y\} \tag{6.20}$$

for some $\hat{x} \in X$. There is obviously no guarantee that for any given $\hat{y}$ the optimal decision x of (6.19) will be an optimal, or even good, decision with respect to (6.18). The same applies to (6.20).

To achieve a useful result, Gauss-Seidel methods alternate between optimising with respect to x and y. Starting at an initial point (x^0, y^0), at each iteration k the method minimises over x while holding $y = y^{k-1}$ constant, finding an optimal point x^k. The method then minimises over y while holding $x = x^k$ constant, finding an optimal point y^k. Since we are carrying out a minimisation at each step, the inequality

$$f(x^{k-1}, y^{k-1}) \geqslant f(x^k, y^{k-1}) \geqslant f(x^k, y^k)$$

holds for all k, meaning the decision (x^k, y^k) found in each iteration is guaranteed to be no worse than the previous decision (x^{k-1}, y^{k-1}). This basic framework is formalised in Algorithm 6.1.

Algorithm 6.1 A block GS method

1: **initialise** $(x^0, y^0) \in X \times Y$
2: **for** $k = 1, \ldots, k_{\max}$ **do**
3: $\quad x^k \leftarrow \operatorname{argmin}_x \left\{f(x, y^{k-1}) \mid x \in X\right\}$
4: $\quad y^k \leftarrow \operatorname{argmin}_y \left\{f(x^k, y) \mid y \in Y\right\}$
5: $\quad k \leftarrow k + 1$
6: **end for**
7: **return** $(x^{k_{\max}}, \zeta^{k_{\max}})$

The sequence of decisions $\left\{(x^k, y^k)\right\}$ generated by Algorithm 6.1 has limit points if X and Y are compact. Furthermore, if f is continuous and bounded from below the following proposition may be applied.

Proposition 6.20 *For problem* (6.18)*, let f be continuous and bounded from below, and let X and Z be compact. Then the limit points (x^*, y^*) of the sequence $\left\{(x^k, y^k)\right\}$ generated by iterations of Algorithm 6.1 are partial minima, i.e. they satisfy the following conditions:*

$$f(x^*, y^*) \leqslant f(x, y^*) \quad \forall x \in X \tag{6.21}$$

$$f(x^*, y^*) \leqslant f(x^*, y) \quad \forall y \in Y \tag{6.22}$$

Proof. We have by construction that $f(x^k, y^k) \leqslant f(x^k, y)$ for all $y \in Y$, and (6.22) follows directly from the continuity of f. To establish (6.21), assume for sake of contradiction that there is an $\bar{x} \in X$ for which $f(x^*, y^*) > f(\bar{x}, y^*)$. Due to the continuity of f, we have, for some infinite subsequence index set $\mathcal{K}$ such that $\lim_{k\to\infty, k\in\mathcal{K}}(x^k, y^k) = (x^*, y^*)$, the existence of $\gamma > 0$ such that $f(x^k, y^k) - f(\bar{x}, y^k) > \gamma > 0$. Thus, $f(x^k, y^k) > f(\bar{x}, y^k) + \gamma \geqslant f(x^{k+1}, y^k) + \gamma \geqslant f(x^{k+1}, y^{k+1}) + \gamma$, which would imply that $\lim_{k\to\infty} f(x^k, y^k) = -\infty$ since $\mathcal{K}$ is an infinite index set and $f(x^k, y^k)$ is monotonically non-increasing in the original sequence, so that f is unbounded from below, a contradiction. Therefore, (x^*, y^*) must be a partial minimum for problem (6.18). ■

If f is convex, differentiable and inf-compact, and X and Y are non-empty, closed and convex, the limit points of Algorithm 6.1 are optimal for (6.18) (see, for example, [12, 49, 108]). However, these conditions do not hold for our reformulation of ζ^{SIP}, since the $[\cdot]^-$ terms in the objective of (6.17) are non-differentiable and the integrality constraints on x and y mean that their feasible regions are non-convex. When these conditions do not hold, the partial minima found as limit points of Algorithm 6.1 may be neither global nor even local minima of the problem.

The following examples illustrate the details of this problem and yield some insight into what measures may be taken (particularly with relation to the penalty parameter ρ) to reduce its impact on solution quality.

Examples:

1. Let problem (6.18) be specified so that $f(x, y) : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ is defined to be $f(x, y) = 7x^2 + 10xy + 7y^2$, and let $X = Y = \{-2, -1, 0, 1, 2\}$. For $(x^0, y^0) = (2, -2)$, the application of Algorithm 6.1 leads immediately to the one limit point $(\hat{x}, \hat{y}) = (1, -1)$. We have $f(1, -1) = 4$, but $f(0, 0) = 0$, so $(\bar{x}, \bar{y}) = (1, -1)$ is not optimal. Note here that f is convex and continuously differentiable, but the constraint set $X \times Y$ is nonconvex due to the integer restriction, and this is the reason that the limit point was not guaranteed to be optimal.

2. Let problem (6.18) be specified so that $f(x, y) : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ is defined to be $f(x, y) = -2x - y + \rho\,|x - y|$, $X = [-2, 3]$, and $Y = [0, 5]$. Observe that $\rho\,|x - y|$ is in effect a penalty term which penalises infeasibility with respect to the constraint $x = y$. For $\rho \in [0, 1)$, the optimal solution is $(x^*, y^*) = (3, 5)$. For $\rho = 1$, the optimal solutions are taken from $(x^*, y^*) \in \{3\} \times [3, 5]$, and for $\rho > 1$, the optimal solution is $(x^*, y^*) = (3, 3)$.

(a) When applying the GS approach of Algorithm 6.1 with $\rho \in (0,1)$, the resulting sequence stabilises after one iteration at the optimum $(x^*, y^*) = (3,5)$ for any feasible starting point.

(b) For $\rho = 1$ with $zy^0 \geqslant 3$, we have after half an iteration $(x^1, y^0) = (3, y^0)$ which is an optimum solution, and the remaining updates stay at some optimal solution $(x^*, y^*) \in \{3\} \times [3,5]$. For $\rho = 1$ with starting point $y^0 < 3$, we have $x^1 = 3$ and $y^1 \in [3,5]$ and so stabilisation at an optimal solution also occurs.

(c) For $\rho > 1$ with $y^0 \geqslant 3$, we have $(x^1, y^1) = (3,3)$, which is optimal. However, for $\rho > 1$ with $y^0 < 3$, we have $x^1 = y^0$ and $y^1 = y^0$, so that stabilisation occurs at $(\bar{x}, \bar{y}) = (y^0, y^0)$, which is not optimal.

3. Let problem (6.18) be specified so that $f(x,y) : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \mapsto \mathbb{R}$ is defined to be

$$f(x,y) = 2x_{1,1} - 1x_{1,2} - 2x_{1,3} - 2x_{2,1} - 1x_{2,2} + 2x_{2,3} + \rho \sum_{i=1,2} \sum_{j=1,2,3} |x_{i,j} - y_j| \,,$$

and let X and Y be defined so that

$$Y = \left\{ (x)_{i,j} : \sum_{j=1}^{3} x_{i,j} \leqslant 1 \text{ for } i = 1,2;\ x_{i,j} \in \{0,1\} \text{ for } i = 1,2,\ j = 1,2,3 \right\},$$

and $Z = \{0,1\}^3$. For $\rho \to \infty$ (simulating the enforcement of constraints $x_{i,j} = y_j$ for $i = 1,2$ and $j = 1,2,3$) we have the optimal solution

$$(x^*, y^*) = ([(0,1,0),(0,1,0)], 0,1,0)\,.$$

If such constraints are altogether ignored ($\rho = 0$), then the optimal x-component is $x^* = ((0,0,1),(1,0,0))$. This behaviour would only change at the threshold $\rho = 1$. For $\rho > 1$, the optimal solution would be $(x^*, y^*) = ([(0,1,0),(0,1,0)],(0,1,0))$.

(a) Now we consider what happens when the GS approach of Algorithm 6.1 is applied. Let $y^0 = (0,0,0)$. Starting with a small penalty such as $\rho = 0.5$, we have

$$x^1 = ((0,0,1),(1,0,0)) \quad \text{and} \quad y^1 \in \{(0,0,0),(0,0,1),(1,0,0),(1,0,1)\}\,,$$

where there is more than one way to choose y^1. If, for example, we make it a policy to choose y by some bitwise lexicographical rule, then we choose $y^1 = (0,0,0)$. Keeping

this same penalty $\rho = 0.5$, we find that stabilisation has occurred, where $x^k = x^1$ and $y^k = y^1$ for $k \geqslant 1$. If we increase the penalty value to $\rho = 2$ for iteration $k = 2$, then we have the stabilisation $x^1 = ((0,0,0),(0,0,0))$ and $y^1 = (0,0,0)$, which is suboptimal (and $\rho = 2$ is the threshold for this change in stabilisation to occur).

If, instead, the y update is chosen by a reverse-lexicographic rule, so that $y^1 = (1,0,1)$, then we have immediate stabilisation with

$$(x^k, y^k) = ((0,0,1),(1,0,0),(1,0,1))$$

for all $k \geqslant 1$ for all $\rho > 0$. (Notice that no matter how large the penalty is, consensus is not achieved in the GS setting. That is, without additional restriction on how y is updated, the optimal y update may be chosen to always correspond to a consensus solution that is infeasible for both scenarios. In practice, we would need a rule to insure that the y update is chosen to satisfy $\sum_{j=1}^{3} y_j \leqslant 1$ to match with the constraints in the x update subproblems.)

(b) The shortcomings of the above GS approach motivate the introduction of more precision in how the consensus discrepancies are penalised, where f is redefined to be

$$f(x,y) = 2x_{1,1} - 1x_{1,2} - 2x_{1,3} - 2x_{2,1} - 1x_{2,2} + 2x_{2,3} + \sum_{i=1,2}\sum_{j=1,2,3} \rho_{i,j}\left|x_{i,j} - y_j\right|.$$

That is, instead of one scalar ρ, we have term-specific $\rho_{i,j} > 0$ for each $i = 1,2$ and $j = 1,2,3$. We start as before with $y^0 = (0,0,0)$, and let $\rho_{i,j} = 0.5$ for each $i = 1,2$ and $j = 1,2,3$. Assuming lexicographic rule in choosing y, we have as before

$$x^1 = ((0,0,1),(1,0,0)) \quad \text{and} \quad y^1 = (0,0,0),$$

and this is stable if the penalty does not change. Now increase $\rho_{1,3} = \rho_{2,1} = 1$, and we have

$$x^2 = ((0,0,1),(1,0,0)) \quad \text{and} \quad y^2 = (1,0,1),$$

and this is stable if the penalty does not change. Increasing $\rho_{1,1} = \rho_{2,3} = 1$, we have again

$$x^3 = ((0,0,1),(1,0,0)) \quad \text{and} \quad y^3 = (0,0,0),$$

and this is stable. But once we again increase $\rho_{1,3} = \rho_{2,1} = 2$, we have

$$x^4 = ((0,1,0),(0,1,0)) \quad \text{and} \quad y^4 = (0,1,0),$$

which is optimal for the original problem.

Example 2 illustrates the trade-off inherent in choosing the penalty parameter. If a large ρ is chosen, the non-differentiable terms in the objective function become dominant and the block Gauss-Seidel method does not converge to the optimal decision for (6.18). This is dependent on the initial point chosen; even if ρ is large we may still obtain convergence to the optimal point if our initial point was chosen well. This problem is averted for a smaller choice of ρ, but the optimal decision for the resulting problem may be infeasible with respect to the constraints relaxed by the penalty term (in the context of SIP, these are the non-anticipativity constraints).

Example 3 further demonstrates that there may in fact be no viable choice of a single ρ which avoids both of these problems. Part (b) of Example 3 suggests a possible solution; if the ρ parameter (which multiplies each penalty term) is divided into a separate penalty parameter for each constraint, then each component of ρ may be set only as large as is necessary to ensure that the final optimal solution respects the relaxed constraint, which gives us the best possible chance to avoid sub-optimal convergence of the block Gauss-Seidel method.

In practice it is unlikely to be clear *a priori* which of the relaxed constraints should be penalised to a greater or lesser degree, in which case it will be necessary to modify the penalty parameters in the course of the algorithm. The algorithm presented in the following sections takes the natural approach of starting with small penalty parameters, then dynamically increasing them in an attempt to enforce feasibility with respect to the relaxed constraints. This has the additional benefit of providing the block Gauss-Seidel method with a reasonable starting point (based on optimising with a small penalty parameter) with which to approach the more difficult problem (with a large penalty parameter, and hence a sharper non-differentiable point).

6.3.2 Formalising a block Gauss-Seidel method for SIP

In this section we will examine how a block Gauss-Seidel type method can be used to obtain solutions for SIP problems, in the form of (6.17). The central motivation is to separate the minimisation over z from the minimisation over x and y so that the latter minimisation problem is separable.

For purposes of simplicity let

$$\phi^{\rho}(x, y, z, \rho) \quad := \quad \sum_{s \in S} \phi_s^{\rho}(x_s, y_s, z, \rho_s),$$

where

$$\phi_s^{\rho}(x_s, y_s, z, \rho_s) := p_s(c^T x_s + d_s^T y_s) + \sum_{s \in S} \underline{\rho}_s^{\top} [x_s - z]^- + \sum_{s \in S} \overline{\rho}_s^{\top} [z - x_s]^-.$$

ζ^{SIP} may now be written in the form

$$\begin{aligned} \zeta^{SIP} = & \min_{\boldsymbol{x}, \boldsymbol{y}, z} \sum_{s \in S} \phi_s^{\rho}(x_s, y_s, z, \rho_s) \\ & \text{s.t. } x_s \in X \quad \forall s \in S \\ & y_s \in Y_s(x_s) \quad \forall s \in S \end{aligned} \tag{6.23}$$

For a given $\rho^k = (\underline{\rho}_s^k, \overline{\rho}_s^k)_{s \in S}$ and an initial $z^{0,0}$, the proposed method will iterate between the solution of the following $l = 0, 1, \ldots, l_{\max}$ subproblems:

$$\begin{aligned} (x^{k,l+1}, y^{k,l+1})_{s \in S} \leftarrow & \operatorname*{argmin}_{x,y} \ \phi^{\rho}(x, y, z^{k,l}, \rho^k) \\ & \text{s.t.: } x_s \in X, \ \forall s \in S \\ & \quad y_s \in Y_s(x_s), \ \forall s \in S, \end{aligned} \tag{6.24}$$

and

$$z^{k,l+1} \leftarrow \operatorname*{argmin}_{z} \ \phi^{\rho}(x^{k,l+1}, y^{k,l+1}, z, \rho^k), \tag{6.25}$$

followed by $l \leftarrow l + 1$ and successive repetition until *partial convergence* is approximately achieved, i.e.

$$\phi^{\rho}(x^{k,l}, y^{k,l}, z^{k,l}, \rho^k) - \phi^{\rho}(x^{k,l+1}, y^{k,l+1}, z^{k,l+1}, \rho^k) \leqslant \epsilon$$

given a threshold $\epsilon \geqslant 0$.

If the current primal infeasibility level, given by a residual measure such as $||x^{k,l} - z^{k,l}||_2^2$, is not acceptable for a ϵ threshold, the set of penalties ρ^k are then updated to ρ^{k+1} and the process is repeated for iteration $k + 1$.

Note that the (x, y)-update step depicted in (6.24) is fully separable by scenario, since the z term is held constant.

The remaining problems are to recast the objective function $\phi^\rho(x, y, z, \rho)$ (particularly the $[\cdot]^-$ terms) in a form amenable to conventional MILP solvers, and to clarify the form of the z update step. To accomplish the first, we consider an equivalent reformulation of the problem given by

$$\begin{aligned} \min_{x,y,\underline{w},\overline{w}} \; & p_s(c^\top x_s + q_s^\top y_s) + \underline{\rho}_s^{k\top} \underline{w}_s + \overline{\rho}_s^{k\top} \overline{w}_s \\ \text{s.t.: } & \underline{w}_s \geqslant 0, \quad \underline{w}_s \geqslant z^{k,l} - x_s \\ & \overline{w}_s \geqslant 0, \quad \overline{w}_s \geqslant x_s - z^{k,l} \\ & x_s \in X, \quad y_s \in Y_s(x_s). \end{aligned}$$

The calculation of

$$z^{k,l+1} \in \operatorname*{argmin}_z \; \phi^\rho(x^{k,l+1}, y^{k,l+1}, z, \rho^k) \tag{6.26}$$

may be performed by computing

$$z^{k,l+1} \in \operatorname*{argmin}_z \; \Phi^\rho(x^{k,l+1}, z, \rho^k),$$

where the penalty function $(x, z) \mapsto \Phi^\rho(x, z, \rho)$ is defined by

$$\Phi^\rho(x, z, \rho) := \psi^\rho((x_s - z)_{s\in S}) = \sum_{s\in S} \left(\underline{\rho}_s^\top [x_s - z]^- + \overline{\rho}_s^\top [z - x_s]^- \right).$$

The last displayed problem can be solved using the following equivalent mathematical programming formulation:

$$\begin{aligned} \Phi^\rho(x^{k,l+1}, z^{k,l+1}, \rho^k) = \; & \min_{z,\underline{w},\overline{w}} \sum_{s\in S} (\underline{\rho}_s^k)^\top \underline{w}_s + (\overline{\rho}_s^k)^\top \overline{w}_s \\ & \text{s.t.: } \underline{w}_s \geqslant 0, \; \forall s \in S, \quad \underline{w}_s \geqslant z - x_s^{k,l+1}, \; \forall s \in S \\ & \qquad \overline{w}_s \geqslant 0, \; \forall s \in S, \quad \overline{w}_s \geqslant x_s^{k,l+1} - z, \; \forall s \in S. \end{aligned} \tag{6.27}$$

When the x components are all restricted to take binary values, it is possible to show that the calculation of $z^{k,l+1}$ can be performed in the following closed form where each component of $z^{k,l+1}$ always takes binary value. In that case, its optimal solution is given by

$$z_i^{k,l+1} = \begin{cases} 1, \text{ if } \sum_{s\in S}(1 - x_{s,i}^{k,l+1})\underline{\rho}_s^k < \sum_{s\in S} x_{s,i}^{k,l+1}\overline{\rho}_s^k \\ 0, \text{ if } \sum_{s\in S}(1 - x_{s,i}^{k,l+1})\underline{\rho}_s^k > \sum_{s\in S} x_{s,i}^{k,l+1}\overline{\rho}_s^k \\ \text{either 0 or 1,} \qquad \text{otherwise} \end{cases}, \; i = 1, \ldots, n_x. \tag{6.28}$$

The cases in which we have a tie might require "flipping a coin" for deciding on the value for $z^{k,l+1}$, as it becomes a case of multiple minima. The existence of multiple minima can be better understood

from the following explicit form of the solution for the general case given in Proposition 6.21, from which (6.28) is a special case. In the following proposition, we assume Z is a closed convex set, so that no explicit integrality constraints are enforced.

Proposition 6.21 *Suppose a set of scenario dependent solutions $(x_s)_{s\in S}$, where $x_s = (x_{s,i})_{i=1,\ldots,n_x}$, are given and $z := (z_i)_{i=1,\ldots,n_x}$. For each $i \in \{1,\ldots,n_x\}$ define*

$$\begin{aligned} I^+(z_i) &:= \{s \in S \mid x_{s,i} > z_i\} \\ I^-(z_i) &:= \{s \in S \mid x_{s,i} < z_i\} \\ I^0(z_i) &:= \{s \in S \mid x_{s,i} = z_i\} \end{aligned}$$

Then z_i solves problem (6.26) given fixed $(x_s)_{s\in S}$ if and only if

$$\sum_{s\in I^+(z_i)} \overline{\rho}_{s,i} - \sum_{s\in I^-(z_i)} \underline{\rho}_{s,i} \in \left[-\sum_{s\in I^0(z_i)} \overline{\rho}_{s,i}, \sum_{s\in I^0(z_i)} \underline{\rho}_{s,i} \right], \tag{6.29}$$

defining the right-hand side to be $\{0\}$ if I^0 is empty; this situation will only arise for small ρ.

Proof. The index s term of the penalty function Φ^ρ may be written as

$$\Phi^\rho_s((x_s)_{s\in S}, z, \rho) := \sum_{i=1}^{n_x} \left[\sum_{s\in I^+(z_i)} \overline{\rho}_{s,i} \max\{0, x_{s,i} - z_i\} + \sum_{s\in I^-(z_i)} \underline{\rho}_{s,i} \max\{0, z_i - x_{s,i}\} \right].$$

As this is separable in the variables $(z_1,\ldots,z_{n_x})$, its subdifferential is defined as the cross product of intervals, one for each component i. Thus, the necessary and sufficient condition

$$0 \in \partial\Phi^\rho_s((x_s)_{s\in S}, z, \rho),$$

can be equivalently stated as

$$0 \in \partial_{z_i}\Phi^\rho_s((x_s)_{s\in S}, z_i, \rho),$$

for each $i = 1,\ldots,n_x$, which is given by:

$$\begin{aligned} 0 &\in \sum_{s\in I^-(z_i)} \underline{\rho}_{s,i} - \sum_{s\in I^+(z_i)} \overline{\rho}_{s,i} + \sum_{s\in I^0(z_i)} \left[-\overline{\rho}_{s,i}, \underline{\rho}_{s,i} \right] \\ &= \sum_{s\in I^-(z_i)} \underline{\rho}_{s,i} - \sum_{s\in I^+(z_i)} \overline{\rho}_{s,i} + \left[-\sum_{s\in I^0(z_i)} \overline{\rho}_{s,i}, \sum_{s\in I^0(z_i)} \underline{\rho}_{s_i,i} \right]. \end{aligned}$$

which in turn is equivalent to (6.29). ■

Remark 6.22 *In practice I^0 is always non-empty; if we consider the formulation in 6.27 it is clear that any solution where $z_i \neq x_s, i$ for all i cannot be optimal since*

We now consider how to update the penalty parameters ρ^k. A simple strategy is

$$\underline{\rho}_s^{k+1} = \underline{\rho}_s^k + \gamma[x_s^{k,l} - z^{k,l}]^-$$
$$\overline{\rho}_s^{k+1} = \overline{\rho}_s^k + \gamma[z^{k,l} - x_s^{k,l}]^-$$

for some positive γ. By doing so, we are reinforcing the penalties associated with the respective discrepancies. Recalling that $[v]^- := -\min\{0, v\}$, we have namely that for each $i = 1, \ldots, n_x$:

$$\underline{\rho}_{s,i}^{k+1} = \begin{cases} \underline{\rho}_{s,i}^k + \gamma(z_i^{k,l} - x_{s,i}^{k,l}), & \text{if } x_{s,i}^{k,l} < z_i^{k,l} \\ \underline{\rho}_{s,i}^k, & \text{if } x_{s,i}^{k,l} \geqslant z_i^{k,l} \end{cases}$$

$$\overline{\rho}_{s,i}^{k+1} = \begin{cases} \overline{\rho}_{s,i}^k + \gamma(x_{s,i}^{k,l} - z_i^{k,l}), & \text{if } z_i^{k,l} < x_{s,i}^{k,l} \\ \overline{\rho}_{s,i}^k, & \text{if } z_i^{k,l} \geqslant x_{s,i}^{k,l} \end{cases}$$

Remark 6.23 *The update in ρ^{k+1} has the effect of changing the left hand side of (6.29) at the next iteration by the amount:*

$$\Delta_i^{k+1} := \gamma \left[\sum_{s \in I^+(z_i^k)} [z_i^k - x_{s,i}^k]^- - \sum_{s \in I^-(z_i^k)} [x_{s,i}^k - z_i^k]^- \right], \tag{6.30}$$

for each $i = 1, \ldots, n_x$. If the addition of this factor ensures the sum in left hand side of (6.29) at iteration $k+1$ exits the interval $\left[-\sum_{s \in I^0(z_i)} \overline{\rho}_{s,i}, \sum_{s \in I^0(z_i)} \underline{\rho}_{s,i}\right]$ associated with the prior choice of $z_i^k = x_{s,i}^k$ then we would be forced to choose new consensus values z_i^k in order to re-establish the satisfaction of the optimality condition (6.29). In doing so, a reassignment of the index sets $I^+(z_i^k)$, $I^-(z_i^k)$, and $I^0(z_i^k)$ is effected. As intuition would suggest, the optimality condition (6.29) is more easily satisfied when $s \in I^0(z_i^k)$ for large $\underline{\rho}_{s,i}$ and $\overline{\rho}_{s,i}$, as this makes the target interval $\left[-\sum_{s \in I^0(z_i)} \overline{\rho}_{s,i}, \sum_{s \in I^0(z_i)} \underline{\rho}_{s,i}\right]$ larger.

To effect a gradual increase in the terms Δ^k in an attempt to improve convergence with the satisfaction of the non-anticipativity condition, we considered an increasing multiplier factor to ψ^ρ given by $\beta^{(k-1)} - 1$ (where $\beta \in (1, 2]$ to keep penalty growth at a reasonable rate and $(k-1)$

represents an exponent and not an iteration index). In other words, we consider the objective at a given iteration k as being

$$\phi^{\rho,k}(x_s, y_s, z, \omega) := \sum_{s\in S} p_s(c^\top x_s + q_s^\top y_s) + (\beta^{(k-1)} - 1)\left[\sum_{s\in S} \underline{\rho}_s^\top [x_s - z]^- + \sum_{s\in S} \overline{\rho}_s^\top [z - x_s]^-\right].$$

These observations culminate in the Penalty-based block Gauss-Seidel (PBGS) method presented in Algorithm 6.2.

Algorithm 6.2 Penalty-based block Gauss-Seidel (PGBS) method for SIP

initialise $\rho^0 = (\underline{\rho}^0, \overline{\rho}^0), \hat{z}^0, \epsilon, \gamma, \beta, l_{\max}, k_{\max}$
for $s \in S$ **do**
  $\hat{x}_s^0 \leftarrow \operatorname{argmin}_{x,y} \ \{\phi^{\rho,1}(x_s, y_s, \hat{z}^0, \rho^0) : x_s \in X, y_s \in Y_s(x_s)\}$
end for
for $k = 1, \ldots, k_{\max}$ **do**
  $x^{k,0} \leftarrow \hat{x}^{k-1}$
  $z^{k,0} \leftarrow \hat{z}^{k-1}$
  for $l = 1, \ldots, l_{\max}$ **do**
    for $s \in S$ **do**
      $(x_s^{k,l}, y_s^{k,l}) \leftarrow \operatorname{argmin}_{x,y} \ \left\{\phi^{\rho,k}(x_s, y_s, z^{k,l-1}, \rho^k) : x_s \in X, y_s \in Y_s(x_s)\right\}$
    end for
    $z^{k,l} \leftarrow \operatorname{argmin}_z \ \phi^{\rho,k}(x^{k,l}, y^{k,l}, z, \rho^k)$
    $\Gamma \leftarrow \phi^{\rho,k}(x^{k,l-1}, y^{k,l-1}, z^{k,l-1}, \rho^k) - \phi^{\rho,k}(x^{k,l}, y^{k,l}, z^{k,l}, \rho^k)$
    if $\Gamma \leqslant \epsilon$ **or** $l = l_{\max}$ **then**
      $(\hat{x}_s^k, \hat{y}_s^k) \leftarrow (x_s^{k,l}, y_s^{k,l})$ for all $s \in S$
      $\hat{z}^k \leftarrow z^{k,l}$
      break
    end if
    $l \leftarrow l + 1$
  end for
  if $||\hat{x}^k - \hat{z}^k||_2^2 \leqslant \epsilon$ **or** $k = k_{\max}$ **then**
    return $((\hat{x}_s^k, \hat{y}_s^k)_{s\in S}, \hat{z}^k)$
  else
    $\underline{\rho}_s^k = \underline{\rho}_s^{k-1} + \gamma[\hat{x}_s^k - \hat{z}^k]^-$ for all $s \in S$
    $\overline{\rho}_s^k = \overline{\rho}_s^{k-1} + \gamma[\hat{z}^k - \hat{x}_s^k]^-$ for all $s \in S$
  end if
  $k \leftarrow k + 1$
end for

6.4 Computational Results

6.4.1 Preliminary Information

The SIP instance classes selected for testing the capabilities of Algorithm 6.2 are the CAP [15], DCAP [4] and SSLP [92] problem classes (see Section 2.4.2 for more details). Since Algorithm 6.2 is a heuristic with potentially high sensitivity to changes in the problem data, sets of 50 instances were taken from two problems from each class so as to more reliably evaluate its potential. The specific test problems from each class are:

- CAP101 and CAP111, with 100 scenarios sampled from the 5000 scenarios available.
- DCAP233 and DCAP342, with 100 scenarios sampled from the 500 scenarios available.
- SSLP5-50 and SSLP10-50, with 100 scenarios generated as per [92].

The Progressive Hedging algorithm (see Section 2.5.1) was also used to solve these problems as a point of comparison with an existing heuristic for calculating primal solutions of SIPs. An important point to note is that the Progressive Hedging algorithm uses the squared 2-norm as a penalty function, and hence its subproblems are quadratic integer programs. Since the block Gauss-Seidel based method presented here only requires linear integer programs to be solved, its subproblems could be expected to be computationally easier. However, due to the other differing characteristics of the two algorithms this does not suffice as a comparison of their merits.

Another advantage of PGBS in the context of SIP problems is that $z^{k,l+1}$ (computed in Line 12) frequently satisfies the integrality constraints of the problem. By comparison, the z consensus values computed in PH by averaging the first-stage variables tend to be non-integer. When the first-stage variables are binary, the PH averaging computation of the consensus $z^{k,l+1}$ is especially prone to producing many non-integer components which can lead to episodic cycling in binary values set in the assignment of scenario specific variables.

One important difference between PGBS and PH is that in PGBS we do not update the dual variables ω. In principle it would be possible to incorporate a dual variable update in PGBS. However, preliminary experiments with the subgradient method to update the dual solutions ω did not provide significant improvement in the PGBS performance. It seems likely that this observation

is related to the asymmetrical way we update the penalty terms which, in effect, subsumes the role usually taken by the linear multiplier term. Consequently, in Algorithm 6.2 we use $\omega = 0$ and do not update the dual solutions.

In the PGBS experiments, the parameters were chosen from $\beta \in \{1.11, 1.25\}$, and $\gamma \in \{0.5\rho^0, \rho^0\}$. Three different initial values for ρ^0 were used in both the PGBS and PH experiments; these were chosen individually to suit each problem class. In the Progressive Hedging algorithm, dual multipliers were initialised as 0.

As the CAP and SSLP problems have pure binary first-stage variables, we used (6.28) to perform the z-update in PGBS (Line 12 of Algorithm 6.2). For DCAP, we relied on solving (6.27) explicitly.

A time limit of 1000 seconds and termination condition of $\epsilon = 10^{-3}$ was used for both methods. A total of 300 ($3 \times 2 \times 50$) instances were solved with three parameters choices for PH (different choices of ρ^0) and 12 combinations of parameter choices for PBGS (different choices of ρ^0, β, and γ). Parameters ρ^0 were set based on early experiments (in case of CAP) and considering values known to be (near) optimal from other studies that have applied PH to instances of these problems (in case of DCAP and SSLP, e.g., [50] and [45]).

All methods have been implemented using AIMMS 4.14 and subproblems have been solved using CPLEX 12.6.2 with its standard configurations.

6.4.2 Numerical Results

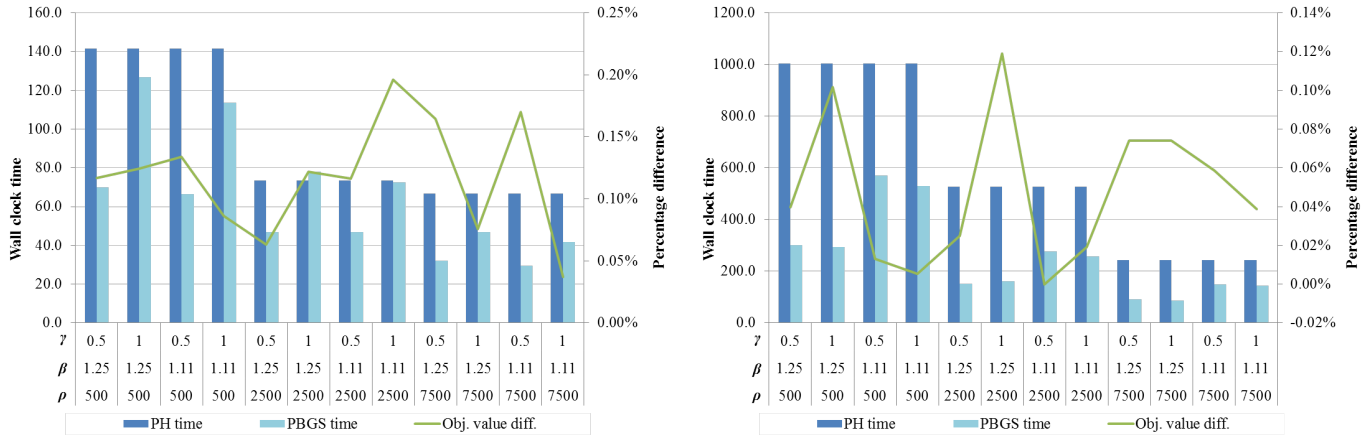
A summary of the computational results is presented in Figures 6.1 to 6.3, which depicts the average computational time and objective value difference for the 50 instances considered for both PH and PBGS in all parameter settings that have been tested.

The blue bars indicate the average wall clock times for both methods. The instances in which PH terminated due to the time limit of 1000s have been removed from the average calculations, these being treated as outliers. The green line shows the average objective value relative difference, which is calculated as

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\zeta^i_{PBGS} - \zeta^i_{PH}}{\zeta^i_{PH}},$$

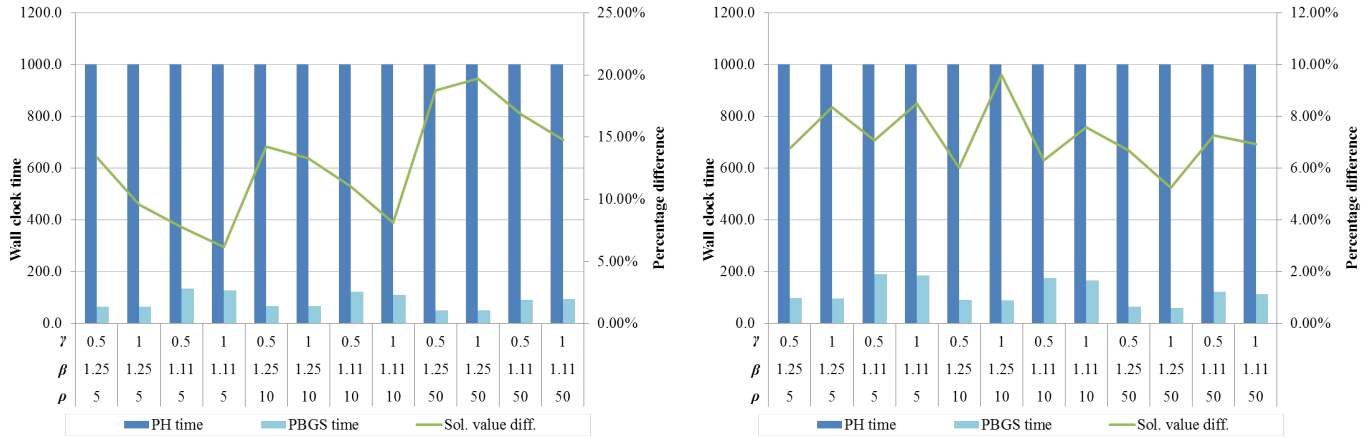
where ζ^i_{PBGS} and ζ^i_{PH} are the objective function values obtained for the solutions returned by PBGS and PH for instance i, respectively, and N is the total number of instances considered for average

value calculations. To obtain ζ^i_{PBGS} and ζ^i_{PH}, we used the last solution returned by both methods and evaluated it a posteriori. For the cases in which PH returned solutions that were infeasible in regard to integrality restrictions (typically those obtained when the algorithm stopped due to the time criterion), rounding has been performed to recover a feasible solution to be evaluated when applicable.



(a) CAP101 (b) CAP111

Figure 6.1: PBGS results for CAP instances

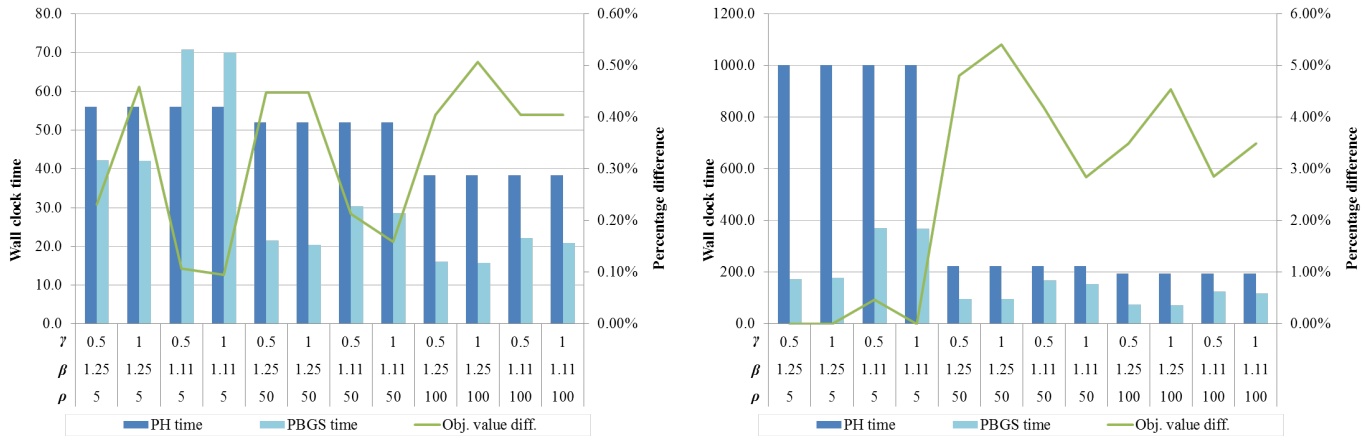


(a) DCAP233 (b) DCAP342

Figure 6.2: PBGS results for DCAP instances

For the CAP instances, all configurations tested with PBGS and PH presented similar values for the objective function, and in most configurations PBGS presented better performance in terms of computational time. For the DCAP instances, in all cases PH terminated due to the time

limit of 1000 seconds. For these problems, a comparison in terms of objective function shows that the differences between the objective function value of the solutions found by PGBS and PH are more pronounced. A similar behaviour can be observed in the SSLP instances, in which PBGS outperforms PH in terms of solution times in most cases while providing solutions that are, in the worst case, 0.5% worse for SSLP5-50 and 5% worse for SSLP10-50. Overall, PBGS seems to be able to obtain comparably good solutions however presenting more reliable convergence behaviour.



(a) SSLP5-10 (b) SSLP10-50

Figure 6.3: PBGS results for SSLP instances

6.5 Conclusions

This chapter has explored an alternative approach for solving stochastic mixed-integer problems based on the combination of penalty-based and block Gauss-Seidel methods. The motivation for this method arises from recently developed theoretical results which encourage the use of non-smooth penalties within Lagrangian-based methods.

The computational experiments detailed in Section 6.4 suggest that there is potential for exploiting this framework as the resulting algorithm is competitive with contemporary methods in terms of computational efficiency. Similarly to Progressive Hedging, the methodology developed here is easily amenable to both parallel computation and multi-stage SIP problems. It can also potentially benefit from many modern enhancements for PH available in literature, such as improved techniques to set values for ρ [112] and scenario aggregation techniques [27, 45].

Further developments of this research can be separated into two classes. From a theoretical

perspective, suitable alternative extensions of the block Gauss-Seidel approach into non-smooth non-separable problems are worth investigation. A better understanding of how to fine-tune the updates of the penalty coefficients would improve the likelihood (or perhaps even guarantee!) that the block Gauss-Seidel iterations do not display suboptimal stationarity. This would improve the trend of the objective values computed by the main algorithm. In terms of practical considerations, it would be of interest to evaluate the performance of the proposed approach with alternative versions of PH that rely on heuristically improved ways of setting the penalty parameter ρ to provide a more broad comparison of performance between the two algorithms. It would be also of interest to evaluate the performance of the proposed approach in contexts other than SIPs, its extension to the multi-stage case, and compare its performance with other algorithms that are tailored to problems of similar structure, such as presented in [2], for example.

It is noteworthy that the PBGS algorithm is not theoretically guaranteed to converge to even a feasible point (although when applied to our test problems it appears to do so invariably). A modified version of the PBGS algorithm which is guaranteed to produce a feasible point for a subclass of SIP is presented in the next chapter.

Chapter 7

Theoretical Extension for the Penalty-based Gauss-Seidel Method

7.1 Introduction

7.1.1 Problem Formulation

In this chapter we will propose a modified version of the penalty-based Gauss-Seidel method of Chapter 6 with stronger theoretical properties for certain sub-classes of Stochastic Integer Programs. In particular we will show that the modified algorithm always converges to a feasible point. To simplify proof construction later in this section we will use a slightly modified non-anticipativity constraint formulation:

$$\begin{aligned}
\zeta^{SIP} = \min_{\boldsymbol{x},\boldsymbol{y},\bar{x},\bar{\boldsymbol{y}}} & \sum_{s\in S} p_s(c^T x_s + d_s^T y_s) \\
\text{s.t. } & x_s \in X \quad \forall s \in S \\
& y_s \in Y_s(x_s) \quad \forall s \in S \\
& z - x_s = 0 \quad \forall s \in S \\
& w_s - y_s = 0 \quad \forall s \in S
\end{aligned} \tag{7.1}$$

As in Chapter 6 we have

$$X := \left\{x \in \mathbb{R}^{n-q} \times \mathbb{Z}^q \mid Ax \leqslant b\right\}$$

$$Y_s(x) := \left\{y \in \mathbb{R}^{m-r} \times \mathbb{Z}^r \mid T_s x + W_s y_s \leqslant h_s\right\}$$

and we make the following assumption on ζ^{SIP} throughout:

Assumption 7.1 *ζ^{SIP} is feasible, the optimal value of ζ^{SIP} is bounded, and p_s, c, d_s, A, b, T_s, W_s, and h_s are vectors or matrices (as appropriate) of rational numbers.*

This formulation has one important distinction from those used in previous chapters; non-anticipativity variables $\bar{\boldsymbol{y}} := \bar{y}_1, \ldots, \bar{y}_s$ and constraints $\bar{y}_s - y_s = 0$ are added for the second stage. Since the second-stage non-anticipativity variables are independent for each outcome scenario and otherwise unconstrained, the non-anticipativity constraints $\bar{y}_s - y_s = 0$ have no practical effect on the feasibility of the second-stage decisions y_s. The advantage of this formulation is that the first- and second-stage variables are represented in a consistent manner. Note that in a multi-stage SIP formulation all stages except the last have active non-anticipativity constraints, so this change in notation is particularly well suited to generalising results for two-stage SIPs to the multi-stage case.

Define $\mathcal{I}_x$ and $\mathcal{R}_x$ as the index sets of first-stage integer and continuous variables respectively. Similarly, define $\mathcal{I}_y$ and $\mathcal{R}_y$ as the index sets of second-stage integer and continuous variables respectively. We can then write $(x, y) \equiv \left(\left(x_{\mathcal{I}_x}, x_{\mathcal{R}_x}\right), \left(y_{\mathcal{I}_y}, y_{\mathcal{R}_y}\right)\right)$ without loss of generality by reordering the components of x and y.

Many of the results in this chapter will require the first-stage variables to be pure integer i.e. $|\mathcal{I}_x| = r = m$ and $\mathcal{R}_x = \varnothing$. This is a significant restriction on the subclasses of SIP which those results can be applied to.

Define

$$K_s = \left\{(x, y) \in \mathbb{R}^{n-q} \times \mathbb{Z}^q \times \mathbb{R}^{m-r} \times \mathbb{Z}^r \mid x \in X, y \in Y_s(x)\right\}.$$

Condition 7.1 *K_s is a non-empty set of feasible decisions constructed with linear constraints and integrality constraints on the x_s and y_s variables. This also implies that it is closed.*

Condition 7.2 *The optimal value of the SIP minimisation problem is bounded from below.*

Condition 7.3 *The set of constraints in K_s which act on only the first-stage decision variables is identical for all $s \in S$.*

Condition 7.4 *The SIP model has relatively complete recourse i.e. for all scenarios $s \in S$, if a first-stage decision x satisfies all constraints in K_s which act on only the first-stage decision variables (i.e. $x \in X$), a second-stage decision y_s exists which satisfies*

$$(x, y_s) \in K_s.$$

The augmented Lagrangian dual problem corresponding to ζ^{SIP} with penalty function $\psi_\rho(u) := \sum_{s\in S}\psi_\rho^s(u)$ and penalty parameter ρ is:

$$\zeta_\rho^{LD+} := \sup_\lambda \zeta_\rho^{LR+}(\lambda) \tag{7.2}$$

where

$$\zeta_\rho^{LR+}(\lambda) := \min_{x,y,z,w}\{\sum_{s\in S} p_s\left[c^\top x_s + d_s^\top y_s + \lambda_s^\top((z,w_s)-(x_s,y_s)) + \psi_s^{\rho_s}((z,w_s)-(x_s,y_s))\right] \mid (x_s,y_s)\in K_s \quad \forall s\in S\}.$$

The theoretical results in subsequent sections will require a set of conditions on the penalty function ψ_ρ to be satisfied. In the remainder of this section we will discuss the consequences and limitations of a sequence of condition sets, each stronger than the last, before reaching the full set of conditions required for our final results.

The first and weakest set of assumptions on ψ_ρ we will consider were introduced in [18] and later used in [37] for the study of augmented Lagrangian duality in mixed integer programming.

Condition 7.5 *$\psi_\rho : \prod_{s\in S}\mathbb{R}^{|S|n} \mapsto \mathbb{R}$ is a separable augmenting function $\psi_\rho(u) := \sum_{s\in S}\psi_s^{\rho_s}(u_s)$ which satisfies the conditions*

1. $\psi_\rho(0) = 0$
2. $\psi_\rho(u) \geqslant \delta > 0, \forall u \notin V$
3. $\psi_\rho(u) \geqslant \gamma||u||_\infty, \forall u \in V$

for some open neighbourhood V of 0 and positive scalars $\delta,\gamma > 0$, $\rho := (\rho_1,\rho_2,\ldots,\rho_{|S|})$ with the later a vector of positive scalars, which acts as a scaling parameter for ψ_ρ.

Theorem 7.6 *[37, Thm. 5] Consider a two-stage SMIP problem ζ^{SIP} as defined in (7.1) which satisfies Conditions 7.1, 7.2, 7.3 and 7.4. Further, consider the augmented Lagrangian dual problem ζ_ρ^{LD+} as defined in (7.2) which corresponds to ζ^{SIP}.*

If the penalty function $\psi_\rho(u_s) := \sum_{s\in S}\psi_s^{\rho_s}(u_s)$ used in ζ_ρ^{LD+} satisfies Condition 7.5, then there exists a finite ρ such that

$$\zeta_\rho^{LD+} = \zeta_\rho^{LR+}(\lambda_{LP}) = \zeta^{SIP}$$

where λ_{LP} may be taken as an optimal multiplier of the linear programming relaxation of the non-anticipativity conditions $x_s = z$.

Proof. Apply the general theorem [37, Thm. 5] to our problem (7.1). For the case of multidimensional ρ observe that $\psi_\rho(u)$ is bounded below by $\psi_{(v,\ldots,v)}(u)$ where $v = \min\{\rho_1, \ldots, \rho_I\}$. ■

As in Chapter 6 we may set the dual variables in ζ_ψ^{LD+} to zero to obtain

$$\zeta_\rho^{LR+}(0) = \min_{x,y,z,w} \{\sum_{s\in S} p_s \left[c^\top x_s + d_s^\top y_s + \psi_s^\rho((z, w_s) - (x_s, y_s))\right] \mid (x_s, y_s) \in K_s \quad \forall s \in S\}.$$

If ψ_ρ satisfies Condition 7.5 then by Theorem 7.6 for a sufficiently large finite ρ we have

$$\zeta^{SIP} = \zeta_\rho^{LR+}(0).$$

Since $\zeta_\rho^{LR+}(0)$ and ζ^{SIP} are respectively lower and upper bounds on ζ^{LD+} we also have $\zeta^{SIP} = \zeta^{LD+}$.

The minimum over x_s and y_s of the augmented Lagrangian corresponding to (7.1) with dual multipliers set to zero is

$$\varphi_\rho(z, w) := \min_{\substack{(x_s,y_s)\in K_s \\ s\in S}} \left\{\sum_{s\in S} p_s \left[c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}((z, w_s) - (x_s, y_s))\right]\right\} \tag{7.3}$$

where $z \in \mathbb{R}^n$, $w \in \mathbb{R}^{n\times|S|}$ and we write K_s in the following form:

$$K_s = \{(x_s, y_s) \mid x_s \in X \text{ and } y_s \in Y_s\}.$$

We consider the augmented Lagrangian dual function with dual multipliers set to zero:

$$\zeta_\rho^{LR+}(0) = \min_{(z,w)} \varphi_\rho(z, w). \tag{7.4}$$

This is a continuous, unconstrained (global) optimisation problem. As in the previous Theorem 7.6, under Assumption 7.1 on ζ^{SIP} and suitable conditions on ψ_s^ρ we have $\zeta^{SIP} = \zeta_\rho^{LR+}(0)$ for sufficiently large but finite ρ. Any z and w which are optimal decisions for ζ^{SIP} clearly must also be optimal decisions for $\zeta_\rho^{LR+}(0)$. However the converse does not necessarily hold; in general the z and w decisions which are optimal for $\zeta_\rho^{LR+}(0)$ need not even be feasible with respect to ζ^{SIP}. We are interested in what we can discover about the local solutions to the minimisation of φ_ρ.

Assumption 7.2 *$\psi_s^\rho : \mathbb{R}^{n+m} \to \mathbb{R}$ is a continuous* ***integer compatible regularization function*** *(ICRF), meaning that it satisfies the following conditions:*

1. *$\psi_s^\rho(u,v) \geqslant 0$ for all (u,v) and $(u,v) = (0,0)$ if and only if $\psi_s^\rho(u,v) = 0$.*

2. *If $\gamma \in (0,1)$ then $\psi_s^\rho(\gamma(u,v)) < \psi_s^\rho(u,v)$ for all $(u,v) \neq 0$.*

3. *There exists a continuous, strictly increasing, function $\gamma(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ and a $\overline{M}$ such that for all $M < \overline{M}$ we have $\psi_s^\rho(u,v) \leqslant M$ implies $\|(u,v)\|_1 \leqslant \gamma^{-1}(\frac{M}{\min_s\{\rho_s\}})$ for the l_1 norm $\|\cdot\|_1$ on $\mathbb{R}^{n+m}$.*

Remark 7.7 *In practice it is generally the case that*

$$\psi_s^\rho(u,v) = \rho\psi_s(u,v)$$

for some coercive, differentiable function ψ and so $\frac{\psi_s^\rho(u,v)}{\rho_s} = \psi_s(u,v)$.

As separation between the first and second stage variables is a basic tool in stochastic optimisation we need this property to be reflected in the penalty functions as well. This is achieved by allowing component wise radial strict monotonicity. We require the following further strengthened assumptions:

Assumption 7.3 *We say that $\psi_s^\rho = \rho_s\psi_s : \mathbb{R}^{n+m} \to \mathbb{R}$ is an **integer compatible regularization function plus** (ICRF$^+$), if we assume assumption 7.2 as well as the following conditions holding for all $s \in S$:*

1. *The penalty $(u,v) \mapsto \psi_s(u,v)$ is uniformly continuous over a bounded set.*

2. *If $\gamma \in [0,1)$ then $\psi_s(\gamma u, v) < \psi_s(u,v)$ and $\psi_s(u, \gamma v) < \psi_s(u,v)$ for all $(u,v) \neq 0$.*

3. *There exists a function $h_3 : [0,1] \times [0,+\infty) \to \mathbb{R}_+$ which is uniformly continuous on bounded sets, satisfies the condition*

$$\psi_s(t(u,v)) - \psi_s(u,v) \leqslant -h_1(t, \|(u,v)\|) < 0$$

for all $t \in (0,1)$, and satisfies the condition that $h_1(t, \|(u,v)\|) = 0$ if and only if either $t = 1$ or $\|(u,v)\| = 0$.

4. *There exists a function* $h_2 : \mathbb{R}^{n+m} \to \mathbb{R}_+$ *which is uniformly continuous on bounded sets, vanishes at the origin and at no other point, and satisfies the condition*

$$|\psi_s(u,u) - \psi_s(u',v')| \leqslant h_2((u,v)-(u',v')) \quad \textit{for all } (u,v),(u',v')$$

Remark 7.8 *The penalty function used in the Progressive Hedging algorithm is* $\psi(u,v) = \|(u,v)\|^2$. *This function is an ICRF*$^+$*; in particular,*

$$h_1(t, \|(u,v)\|) = \left(t^2 - 1\right) \|(u,v)\|^2$$

and

$$h_2(u,v) = 2\max\{\|(u,v)\|, \|(u',v')\|\}(\|(u,v)-(u',v')\|)$$

satisfy the conditions of parts 3 and 4 of Assumption 7.3.

7.1.2 Applying Gauss-Seidel to Modified SIP Formulation

For a general treatment of the block Gauss-Seidel method see Section 6.3.1. In this section we will consider the application of a Gauss-Seidel type method to find a local solution for

$$\min_{(z,w)} \varphi_\rho(z,w).$$

Algorithm 7.1 Block GS method for SIP

1: **initialise** $(z^0, w^0) \in X \times Y(z^0)$
2: **for** $k = 1, \ldots, k_{\max}$ **do**
3: $\quad w^{k+1} \leftarrow \operatorname{argmin}_w \ \varphi_\rho\left(z^k, w\right)$
4: $\quad z^{k+1} \leftarrow \operatorname{argmin}_z \ \varphi_\rho\left(z, w^{k+1}\right)$
5: $\quad k \leftarrow k+1$
6: **end for**
7: **return** $(z^{k_{\max}}, w^{k_{\max}})$

Algorithm 7.1 apparently alternates between optimisation over the proximal first-stage and proximal second-stage non-anticipativity variables z and w, respectively. In this section we will see that (after some rearrangement) this algorithm can be implemented by alternating between optimisation over the decision variables (x, y) and the proximal non-anticipativity variables z.

Suppose that (x^*, y^*) is an optimal solution of the proximal point problem (7.3) for $(z, w) = \left(z^k, w^{k+1}\right)$ i.e. for all scenarios $s \in S$ we have

$$(x_s^*, y_s^*) \in \operatorname{argmin}_{(x_s,y_s)\in K_s} c^\top x_s + d_s^\top y_s + \psi_\rho^s\left(\left(z^k, w_s^{k+1}\right) - (x_s, y_s)\right).$$

If $w_s^{k+1} \neq y_s^*$ for some $s \in S$ we have

$$\begin{aligned}
\varphi_\rho\left(z^k, w^{k+1}\right) &= \sum_{s\in S} p_s \min_{(x_s,y_s)\in K_s}\left[c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}\left(\left(z^k, w_s^{k+1}\right) - (x_s, y_s)\right)\right] \\
&= \sum_{s\in S} p_s\left[c^\top x_s^* + d_s^\top y_s^* + \psi_s^{\rho_s}\left(\left(z^k, w_s^{k+1}\right) - (x_s^*, y_s^*)\right)\right] \\
&> \sum_{s\in S} p_s\left[c^\top x_s^* + d_s^\top y_s^* + \psi_s^{\rho_s}\left(\left(z^k, y_s^*\right) - (x_s^*, y_s^*)\right)\right] \quad \text{due to condition 2 of Assumptions 7.3} \\
&\geqslant \sum_{s\in S} p_s \min_{(x_s,y_s)\in K_s}\left[c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}\left(\left(z^k, y_s^*\right) - (x_s, y_s)\right)\right] = \varphi_\rho\left(z^k, y^*\right)
\end{aligned}$$

which implies that $w^{k+1} \notin \operatorname{argmin}_w \left\{\varphi_\rho\left(z^k, w\right)\right\}$.

Therefore, $w^{k+1} \in \operatorname{argmin}_w \left\{\varphi_\rho\left(z^k, w\right)\right\}$ implies that $w_s^{k+1} = y_s^*$ for all $s \in S$ so that

$$\begin{aligned}
&\min_{(x_s,y_s)\in K_s} c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}\left(\left(z^k, w_s^{k+1}\right) - (x_s, y_s)\right) \\
= \;& c^\top x_s^* + d_s^\top y_s^* + \psi_s^{\rho_s}\left(\left(z^k, w_s^{k+1}\right) - (x_s^*, y_s^*)\right) = \min_{(x_s,y_s)\in K_s} c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}\left(z^k - x_s, 0\right).
\end{aligned}$$

Hence w^{k+1} takes the component value y_s^* of the optimal decision (x^*, y^*) for the minimisation problems

$$\begin{aligned}
&\min_{(x_s,y_s)\in K_s} c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}\left(z^k - x_s, 0\right) \text{ and} \\
\varphi_\rho\left(z^k, w^{k+1}\right) = &\min_{(x,y)\in \Pi_s K_s} \sum_{s\in S} p_s\left[c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}\left(z^k - x_s, 0\right)\right]. \qquad (7.5)
\end{aligned}$$

Define $f(x, y) := \sum_{s\in S} p_s\left[c^\top x_s + d_s^\top y_s\right] + \delta_{\Pi_s K_s}(x, y)$ and place $\psi_\rho(u, v) := \sum_{s\in S} p_s \psi_s^{\rho_s}(u_s, v_s)$. Also, define $z_{|S|}$ as a concatenation of the same vector $|S|$ times to produce a vector in $\mathbb{R}^{n\times|S|}$ i.e. $z_{|S|} = (z, z, \ldots, z)$. Observe that

$$\begin{aligned}
z^{k+1} &\in \arg\min_z \varphi_\rho\left(z, w^{k+1}\right) = \\
&= \arg\min_z \min_{(x,y)}\Big\{\sum_{s\in S} p_s\left[c^\top x_s + d_s^\top y_s\right] + \delta_{\Pi_s K_s}(x, y) \\
&\qquad + \sum_{s\in S} p_s \psi_s^{\rho_s}\left(\left(z, w_s^{k+1}\right) - (x_s, y_s)\right)\Big\} \\
&= \arg\min_z \min_{(x,y)}\left\{f(x, y) + \psi_{\rho_s}\left(\left(z_{|S|}, w^{k+1}\right) - (x, y)\right)\right\} \\
&:= \arg\min_z \left(f \,\square\, \psi_\rho\right)\left(z_{|S|}, w^{k+1}\right) \qquad (7.6)
\end{aligned}$$

where $\square$ denotes the infimal convolution. We can now rewrite Algorithm 7.1 in a form which separates the minimisation over the decision and non-anticipativity variables, as shown in Algorithm 7.2.

Algorithm 7.2 Restatement of the block GS method for SIP

1: **initialise** $(z^0, w^0) \in X \times Y(z^0)$
2: **for** $k = 1, \ldots, k_{\max}$ **do**
3: $\quad \left(x_s^{k+1}, y_s^{k+1}\right) \in \arg\min_{(x_s,y_s)\in K_s} \left[c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}\left(z^k - x_s, 0\right)\right] \quad$ for all $s \in S$
4: $\quad w^{k+1} \leftarrow y^{k+1}$
5: $\quad z^{k+1} \leftarrow \arg\min_z \left(f \square \psi_\rho\right)\left(z_{|S|}, w^{k+1}\right)$
6: $\quad k \leftarrow k+1$
7: **end for**
8: **return** $(z^{k_{\max}}, w^{k_{\max}})$

The z update step (as in 7.6) is too complicated to solve exactly as the evaluation of the objective requires the solution of a MIP of equal difficulty as that provided by the original SMIP. Still this problem is useful from a *theoretical* standpoint as it links the consensus problem to the Gauss-Seidel step of the continuous regularisation. Later we will see that in practise we can still produce a descent using the usual consensus update. Note that when $z^{k+1} \in \arg\min_z \sum_{s\in S} p_s \psi_s^{\rho_s}\left(z - x_s^{k+1}\right)$, the (x^{k+1}, w^{k+1}) update step guarantees that

$$\begin{aligned}
\varphi_\rho\left(z^k, w^{k+1}\right) &= \sum_{s\in S} p_s \left[c^\top x_s^{k+1} + d_s^\top y_s^{k+1} + \psi_s^{\rho_s}\left(\left(z^k, w_s^{k+1}\right) - \left(x_s^{k+1}, y_s^{k+1}\right)\right)\right] \\
&\geqslant \sum_{s\in S} p_s \left[c^\top x_s^{k+1} + d_s^\top y_s^{k+1}\right] + \min_z \sum_{s\in S} p_s \psi_s^{\rho_s}\left(z - x_s^{k+1}, 0\right) \qquad (7.7) \\
&= \sum_{s\in S} p_s \left[c^\top x_s^{k+1} + d_s^\top y_s^{k+1}\right] + \sum_{s\in S} p_s \psi_s^{\rho_s}\left(\left(z^{k+1}, w_s^{k+1}\right) - \left(x_s^{k+1}, y_s^{k+1}\right)\right) \\
&\geqslant \min_{\substack{(x_s,y_s)\in K_s \\ s\in S}} \left\{\sum_{s\in S} p_s \left[c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}\left(\left(z^{k+1}, w_s^{k+1}\right) - (x_s, y_s)\right)\right]\right\} \\
&= \varphi_\rho\left(z^{k+1}, w^{k+1}\right) \geqslant \min_z \varphi_\rho\left(z, w^{k+1}\right)
\end{aligned}$$

Thus the solution of $z^{k+1} \in \arg\min_z \sum_{s\in S} p_s \psi_s^{\rho_s}\left(z - x_s^{k+1}\right)$ produces a non-ascent step in $\varphi_{\rho_s}\left(\cdot, w^{k+1}\right)$ in that $\varphi_\rho\left(z^k, w^{k+1}\right) \geqslant \varphi_\rho\left(z^{k+1}, w^{k+1}\right) \geqslant \min_z \varphi_\rho\left(z, w^{k+1}\right)$. Using this observation we can write a Gauss-Seidel algorithm that is guaranteed to produce non-ascent steps, which is given in Algorithm 7.3.

Algorithm 7.3 Modified block GS method for SIP

initialise $(z^0, w^0) \in X \times Y(z^0)$
for $k = 1, \ldots, k_{\max}$ **do**
  $\left(x_s^{k+1}, y_s^{k+1}\right) \in \arg\min_{(x_s,y_s)\in K_s} \left[c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}\left(z^k - x_s, 0\right)\right]$ for all $s \in S$
  $w^{k+1} \leftarrow y^{k+1}$
  $z^{k+1} \leftarrow \arg\min_z \sum_{s\in S} p_s \psi_s^{\rho_s}\left(z - x_s^{k+1}\right)$
  $k \leftarrow k+1$
end for
return $(z^{k_{\max}}, w^{k_{\max}})$

Note that for now the (z^{k+1}, w^{k+1}) update steps do not guarantee *strict* descent in ϕ_ρ. We will address this in the next section, particularly in Lemma 7.23.

Remark 7.9 *If the penalty function used in Algorithm 7.3 is the squared 2-norm, this algorithm is equivalent to Progressive Hedging with no dual multiplier update.*

7.2 Theoretical Results

7.2.1 Properties of the infimal regularisation for a SIP

In this section we develop the properties of the continuous regularisation φ_ρ of our SMIP. Critically we are able to associate the local minima of this function with the feasible points of the associated SMIP. Many of the following results are adapted from similar findings in [19] which apply to the Feasibility Pump algorithm for mixed-integer programming. Recall that by $z_{|S|}$ we denote a concatenation of the same vector $|S|$ times to produce a vector in $\mathbb{R}^{n\times|S|}$ i.e. $z_{|S|} = (z, z, \ldots, z) \in \operatorname{diag}\mathbb{R}^{n\times|S|}$.

Lemma 7.10 *Assume $\psi_s^{\rho_s}$ are ICRFs for each $s \in S$. Then for all $(z, w) \in \mathbb{R}^n \times \mathbb{R}^{m\times|S|}$ the function $\rho \mapsto \varphi_\rho(z, w)$ is non-decreasing.*

Proof. This follows from the non-decreasing function dependence on ρ of the penalty terms $\rho_s \mapsto \varphi_s^{\rho_s}$. ■

Lemma 7.11 *Assume $\psi_s^{\rho_s}$ are ICRFs for each $s \in S$. Then*

$$\varphi_\rho(z, w) \leqslant f(z, w) := \sum_{s\in S} p_s \left[c^\top z + d_s^\top w_s + \delta_{K_s}(z, w_s)\right]$$

for all $(z, w) \in \mathbb{R}^n \times \mathbb{R}^{m\times|S|}$ for all $\rho \geqslant 0$.

Proof. Observe that for $(z, w_s) \in K_s$ we have

$$\begin{aligned}\varphi_\rho(z,w) &= \min_{\substack{(x_s,y_s)\in K_s\\ s\in S}} \left\{ \sum_{s\in S} p_s \left[c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}\left((z,w_s)-(x_s,y_s)\right)\right]\right\} \\ &\leqslant \sum_{s\in S} p_s \left[c^\top z + d_s^\top w_s + \delta_{K_s}(z,w_s) + \psi_s^{\rho_s}\left((z,w_s)-(z,w_s)\right)\right] = f(z,w).\end{aligned}$$

■

Note that since $\rho \mapsto \varphi_\rho(z,w)$ is non-decreasing as per Lemma 7.10, the assumption that $\{\varphi_\rho(z^0,w^0) \mid \rho > 0\}$ is bounded in the following lemma implies that there exists a sequence $\{\rho_i\}$ and a constant C such that $\varphi_{\rho_i}(z^0,w^0) \to C$, where C is the least upper bound on $\varphi_\rho(z^0,w^0)$ for all $\rho > 0$. Now a local minimum of an increasing family of functions might indeed increase in function value at the minimiser and decrease in the domain on which it is a local minimiser. It is those local minimisers that do not suffer from these issues that we wish to isolate.

Lemma 7.12 *Assume that $\psi_s^{\rho_s}$, for all $s \in S$, satisfies the assumption to be an ICRF$^+$ (see Assumption 7.3). Assume that $\min_{(x,y)\in\Pi_{s\in S}K_s} f(x,y) < +\infty$. Let*

$$\varphi_\rho\left(z^0,w^0\right) = \inf\left\{ f_\rho\left((x,y),\left(z^0_{|S|},w^0\right)\right) \mid (x,y)\in\mathbb{R}^{n\times|S|}\times\mathbb{R}^{m\times|S|}\right\} \tag{7.8}$$

where

$$(x,y)\mapsto f_\rho\left((x,y),\left(z^0_{|S|},w^0\right)\right) := \sum_{s\in S} p_s\left[c^\top x_s + d_s^\top y_s + \delta_{K_s}(x_s,y_s) + \psi_s^{\rho_s}\left(\left(z^0_{|S|},w^0_s\right)-(x_s,y_s)\right)\right].$$

Then f_ρ attains a local minimum for all $\rho = \left(\rho_1,\rho_2,\ldots,\rho_{|S|}\right) > 0$.

Proof. Let $M > \min_{(x,y)\in\Pi_{s\in S}K_s} f(x,y)$ and let $\rho > 0$ be sufficiently large that the condition

$$\left(\frac{M - \min_{(x,y)\in\Pi_{s\in S}K_s} f(x,y)}{\min_s\{\rho_s\}}\right) < \overline{M}.$$

is satisfied. Consider

$$\begin{aligned}F_K(z,w) &:= \left\{(x,y)\in\Pi_{s\in S}K_s \mid f(x,y) + \psi_\rho\left(\left(z_{|S|},w\right)-(x,y)\right)\leqslant M\right\} \\ &\subseteq \left\{(x,y)\in\Pi_{s\in S}K_s \mid \left[\min_{(x,y)\in\Pi_{s\in S}K_s} f(x,y)\right] + \psi_\rho\left(\left(z_{|S|},w\right)-(x,y)\right)\leqslant M\right\} \\ &\subseteq \left\{(x,y)\in\Pi_{s\in S}K_s \mid \left\|\left(z_{|S|},w\right)-(x,y)\right\|_1 \leqslant |S|\,\gamma^{-1}\left(\frac{M-\min_{(x,y)\in\Pi_{s\in S}K_s} f(x,y)}{\min_s\{\rho_s\}}\right)\right\}.\end{aligned} \tag{7.9}$$

This set F_K is bounded and non-empty, and so the minimum of f_ρ is attained. ■

Lemma 7.13 *Assume that $\psi_s^{\rho_s}$ are convex functions that for all $s \in S$, satisfies the assumption to be an ICRF$^+$ (see Assumption 7.3). Assume that the SIP has no continuous variables in the first stage and that K_s for all $s \in S$ are bounded. Furthermore assume that we have a point (z^0, w^0) which is a local minimum of φ_ρ for all sufficiently large $\rho = \left(\rho_1, \rho_2, \ldots, \rho_{|S|}\right)$, and that $\{\varphi_\rho(z^0, w^0)\}_{\rho>0}$ is bounded above.*

Then there exists a $\bar{\rho} = \left(\bar{\rho}_1, \bar{\rho}_2, \ldots, \bar{\rho}_{|S|}\right) > 0$ such that for $\rho \geqslant \bar{\rho}$ the minimum in (7.8) is attained by some $(x_s, y_s) \in K_s$ for $s \in S$ and there is a fixed neighbourhood $B_\delta(z^0, w^0)$ (independent of ρ) *which verifies the local optimality of (z^0, w^0) for all $\rho > \overline{\rho}$.*

Proof. We now need to show that (z^0, w^0) is a local minimum of φ_ρ with respect to a <u>fixed</u> neighbourhood $B_{\bar{\delta}}(z^0, w^0)$ for all ρ greater than a sufficiently large $\bar{\rho}$. To show this, we use the fact that $\rho \to \varphi_\rho(\cdot)$ is increasing with $\{\varphi_\rho(z^0, w^0)\}_{\rho>0}$ bounded above by a constant C. We take C to be the smallest such constant.

For sufficiently large $\bar{\rho} > 0$, define U_ρ to be the largest (convex) connected component of

$$\left\{(z, w) \in \mathbb{R}^n \times \mathbb{R}^{m \times |\mathcal{S}|} \mid \varphi_\rho(z, w) < \varphi_\rho\left(z^0, w^0\right) (\leqslant C)\right\}$$

containing (z^0, w^0) which contains all the points (z, w) which demonstrate that φ_ρ fails to be a strict local minimum at (z^0, w^0). Indeed if (z^0, w^0) fails to be local minimum within a *fixed* convex neighbourhood there must exists $(z^\rho, w^\rho) \in U_\rho$ with $(z^\rho, w^\rho) \to (z^0, w^0)$ as $\rho \to \infty$. This situation can be eliminated by showing $U_\rho = \varnothing$ for $\rho > \overline{\rho}$.

Consider $\rho \to U_\rho$, noting that as U_ρ is convex and let $\liminf_{\rho \uparrow \infty} U_\rho := U_\infty$, a convex set. Note that

$$U_\rho \subseteq \left\{(z, w) \in \mathbb{R}^n \times \mathbb{R}^{m \times |\mathcal{S}|} \mid \varphi_\rho(z, w) \leqslant C\right\} := W_\rho,$$

where $\{W_\rho\}$ is a monotonically decrease family of sets, and hence convergent, to the right hand side of

$$\left(z^0, w^0\right) \in \left\{(x, y) \mid \sum_{s \in S} p_s \left[c^\top x + d_s^\top y_s\right] \leqslant C,\ (x, y_s) \in K_s \text{ for all } s\right\}. \tag{7.10}$$

Thus we have U_∞ contained in the largest connected component containing (z^0, w^0) of the set in (7.10).

We make the following observations:

- $\sum_{s\in S} p_s \left[c^\top z^0 + d_s^\top w_s^0\right] \uparrow C$ and $(z^0, w_s^0) \in K_s$ for all s by the initial assumptions of the lemma.
- For all $(z,w) \in U_\rho$ and ρ is sufficiently large, every (x^*, y^*) which satisfies

$$(x^*, y^*) \in \arg \min_{\substack{(x_s,y_s)\in K_s \\ s\in S}} \left\{ \sum_{s\in S} p_s \left[c^\top x_s + d_s^\top y_s + \psi_s^\rho \left((z, w_s) - (x_s, y_s)\right)\right] \right\}$$

  must also satisfy $(x_{\mathcal{I}}^*, (y_s)_{\mathcal{I}}^*) = (z_{\mathcal{I}}^0, (w_s^0))_{\mathcal{I}})$. If this were not the case then $(x_{\mathcal{I}}^*, (y_s^*))_{\mathcal{I}}) \neq (z_{\mathcal{I}}^0, (w_s^0))_{\mathcal{I}})$ for arbitrarily large ρ, and then $\{\varphi_\rho(z,w)\}_{\rho>0}$ would not be bounded above for all $\rho > 0$, contrary to assumption.

We next show that for ρ sufficiently large $U_\rho = \varnothing$. To this end assume that there exists $(z,w) \in U_\rho$ and so

$$\begin{aligned} \varphi_\rho(z,w) &= \min_{\substack{(x_s,y_s)\in K_s \\ s\in S}} \left\{ \sum_{s\in S} p_s \left[c^\top x_s + d_s^\top y_s + \psi_s^\rho \left((z, w_s) - (x_s, y_s)\right)\right] \right\} \\ &< \varphi_\rho\left(z^0, w^0\right) \leqslant C. \end{aligned} \tag{7.11}$$

Thus for $(z,w) \in U_\rho$ and ρ sufficiently large ($\rho \geqslant \bar{\rho}$) we have

$$\varphi_\rho(z,w) = \min \left\{ \sum_{s\in S} p_s \left[c^\top x_s + d_s^\top y_s + \psi_s^\rho \left((z, w_s) - (x_s, y_s)\right)\right] \mid (x,y) = \left(\left(x_{\mathcal{R}}, z_{\mathcal{I}}^0\right), \left(y_{\mathcal{R}}, w_{\mathcal{I}}^0\right)\right) \in \Pi_{s\in S} K_s \right\}.$$

When we locally minimise over a compact, convex set

$$K_{\mathcal{R}}(z,w) := \left\{(x,y) \in \Pi_{s\in S} K_s \mid x_{\mathcal{I}} = z_{\mathcal{I}}^0 \text{ and } y_{\mathcal{I}} = w_{\mathcal{I}}^0\right\},$$

$(z,w) \mapsto \varphi_\rho(z,w)$ can be locally represented as a closed convex function on U_ρ; it is the infimal convolution of two closed, convex functions

$$(x,y) \mapsto \sum_{s\in S} p_s \left[c^\top x_s + d_s^\top y_s\right] + \delta_{K_{\mathcal{R}}}(x,y),$$

and

$$(u,v) \mapsto \sum_{s\in S} p_s \psi_s^\rho (u_s, v_s)$$

with $(z,w) = (x,y) + (u,v)$.

Hence taking $(z', w') = t\left(z^0, w^0\right) + (1-t)(z,w) \in U_\infty$ (recall that $(z_{\mathcal{I}}, (w_s))_{\mathcal{I}}) = (z_{\mathcal{I}}^0, (w_s^0))_{\mathcal{I}})$ for all $(z,w) \in U_\rho$) then by the local convexity of φ_ρ on the set U_ρ and (7.11) we have for all $t \in (0,1)$ that

$$\varphi_\rho(z', w') \leqslant t\varphi_\rho\left(z^0, w^0\right) + (1-t)\varphi_\rho(z,w) < \varphi_\rho\left(z^0, w^0\right),$$

which contradicts the initial assumption of the lemma that (z^0, w^0) is a local minimum of φ_ρ. Hence $U_\rho = \varnothing$. ■

The following is another way of framing the observation made in the last result that the minimisers in (7.12), the definition of the function $\varphi_{\bar{\rho}}$, are attained exactly at the local minimiser of the same function $\varphi_{\bar{\rho}}(\cdot)$. This observation is critical to subsequent analysis so further justification is provided.

Lemma 7.14 *Assume that $\psi_s^{\rho_s}$, for all $s \in S$, satisfies the assumption to be an ICRF$^+$ (see Assumption 7.3). Assume that the SIP has no continuous variables in the first stage and that K_s for all $s \in S$ are bounded. Furthermore assume that we have a point (z^0, w^0) which is a local minimum of φ_ρ with respect to a fixed neighbourhood $B_{\bar{\delta}}(z^0, w^0)$ for all $\rho = \left(\rho_1, \rho_2, \ldots, \rho_{|S|}\right)$ sufficiently large. Then there exists $\bar{\rho} = \left(\bar{\rho}_1, \bar{\rho}_2, \ldots, \bar{\rho}_{|S|}\right) > 0$ such that for $\rho \geqslant \bar{\rho}$, $\left(z^0_{|S|}, w^0\right) \in \mathbb{R}^{n\times|S|} \times \mathbb{R}^{m\times|S|}$ is the unique minimum point of $(x, y) \mapsto f_\rho((x, y), \left(z^0_{|S|}, w^0\right))$ in*

$$\varphi_\rho\left(z^0, w^0\right) = \inf\left\{f_\rho\left((x, y), \left(z^0_{|S|}, w^0\right)\right) \mid (x, y) \in \mathbb{R}^{n\times|S|} \times \mathbb{R}^{m\times|S|}\right\} \tag{7.12}$$

and (z^0, w^0) also satisfies the conditions $\varphi_\rho(z^0, w^0) = \sum_{s\in S} p_s\left[c^\top z^0 + d_s^\top w_s^0\right]$ for all $\rho \geqslant \bar{\rho}$ with $(z^0, w_s^0) \in K_s$ for all $s \in S$.

Moreover $\bar{\rho}$ can be obtained by the strategy of of selectively increasing $\bar{\rho}_s$ for all the $s \in S$ whenever the minimiser $(x_s^*, y_s^*)_{s\in S}$ in $(x, y) \mapsto f_\rho((x, y), \left(z^0_{|S|}, w^0\right))$ has $x_s^* \neq z^0$.

Proof. For all sufficiently large ρ there exists $(\bar{x}, \bar{y}) \in K$ that attains the infimum in the definition of $\varphi_\rho(z^0, w^0)$ i.e. we have

$$\varphi_\rho\left(z^0, w^0\right) = \sum_{s\in S} p_s\left[c^\top \bar{x}_s + d_s^\top \bar{y}_s + \psi_s^{\rho_s}\left(\left(z^0, w_s^0\right) - (\bar{x}_s, \bar{y}_s)\right)\right] \leqslant \ \varphi_\rho(z', w')$$

for all $(z', w') \in B_{\bar{\delta}}\left(z^0_{|S|}, w^0\right)$. Thus

$$\begin{aligned}\sum_{s\in S} \ p_s \ & \left[c^\top \bar{x}_s + d_s^\top \bar{y}_s + \psi_s^{\rho_s}\left(\left(z^0, w_s^0\right) - (\bar{x}_s, \bar{y}_s)\right)\right] \leqslant \varphi_\rho(z', w') \\ \leqslant \ & \sum_{s\in S} p_s\left[c^\top \bar{x}_s + d_s^\top \bar{y}_s + \psi_s^{\rho_s}\left((z', w_s') - (\bar{x}_s, \bar{y}_s)\right)\right]\end{aligned}$$

or

$$\begin{aligned}\Psi_\rho\left(\left(z^0_{|S|}, w^0\right) - (\bar{x}, \bar{y})\right) \ &= \ \sum_{s\in S} p_s\psi_s^{\rho_s}\left(\left(z^0, w_s^0\right) - (\bar{x}_s, \bar{y}_s)\right) \\ &\leqslant \ \sum_{s\in S} p_s\psi_s^{\rho_s}\left((z', w_s') - (\bar{x}_s, \bar{y}_s)\right) = \Psi_\rho\left(\left(z'_{|S|}, w'\right) - (\bar{x}, \bar{y})\right)\end{aligned}$$

Hence for any given $(z', w') \in B_{\bar{\delta}}\left(z^0_{|\mathcal{S}|}, w^0\right)$

$$0 \leqslant \sum_{s \in S} p_s \left[\psi_s^{\rho_s}\left((z', w'_s) - (\bar{x}_s, \bar{y}_s)\right) - \psi_s^{\rho_s}\left(\left(z^0, w^0_s\right) - (\bar{x}_s, \bar{y}_s)\right)\right]. \tag{7.13}$$

We will show that for all $s \in S_\rho := \{s \in S \mid (z^0, w^0_s) \neq (\bar{x}_s, \bar{y}_s)\}$ we must have

$$\begin{aligned} &\psi_s^{\rho_s}\left(\left(z^0, w^0_s\right) - (\bar{x}_s, \bar{y}_s)\right) > \psi_s^{\rho_s}\left((z', w'_s) - (\bar{x}_s, \bar{y}_s)\right) \\ \text{for all} \quad &(z', w'_s) = \left(z^0, w^0_s\right) + t\left((\bar{x}_s, \bar{y}_s) - \left(z^0, w^0_s\right)\right) \in B_{\bar{\delta}}\left(z^0, w^0_s\right) \subseteq \mathbb{R}^n \times \mathbb{R}^m. \end{aligned} \tag{7.14}$$

Assume that $(z^0, w^0_{s'}) \neq (\bar{x}_{s'}, \bar{y}_{s'})$ for some $s' \in S$ and $t \in (0, 1)$ is taken sufficiently small so that

$$\begin{aligned} \left(z^0, w^0_{s'}\right) &\neq (z'_{s'}, w'_{s'}) := \left(z^0, w^0_{s'}\right) + t\left((\bar{x}_{s'}, \bar{y}_{s'}) - \left(z^0, w^0_{s'}\right)\right) \\ &\in B_{\bar{\delta}}\left(z^0, w^0_{s'}\right) \subseteq \mathbb{R}^n \times \mathbb{R}^m. \end{aligned}$$

is satisfied. Indeed we may take t to be a fixed number

$$t = \frac{\bar{\delta}}{B + \left\|\left(z^0_{|\mathcal{S}|}, w^0\right)\right\|}$$

for a fixed $B \geqslant \sup\{\|(z, w)\| \mid (z, w) \in F_K(z^0, w^0)\}$, where F_K as defined in (7.9) is a bounded set (note that t is independent of $\rho \geqslant \bar{\rho}$). Given this, we have

$$\begin{aligned} t\left\|(\bar{x}_{s'}, \bar{y}_{s'}) - \left(z^0, w^0_{s'}\right)\right\| \leqslant t\left\|(\bar{x}, \bar{y}) - \left(z^0_{|\mathcal{S}|}, w^0\right)\right\| &\leqslant t\left[\left\|\left(z^0_{|\mathcal{S}|}, w^0\right)\right\| + \|(\bar{x}, \bar{y})\|\right] \\ &\leqslant t\left[B + \left\|\left(z^0_{|\mathcal{S}|}, w^0\right)\right\|\right] = \bar{\delta} \end{aligned}$$

so $(z'_{s'}, w'_{s'}) \in B_{\bar{\delta}}(z^0, w^0_{s'})$ for this given $t > 0$ and all $s' \in S$. Now if we presume that the opposite inequality to (7.14) holds for $s = s'$ and use the second part of Assumption 7.3 we have

$$\begin{aligned} \psi_{s'}^{\rho_{s'}}\left(\left(z^0, w^0_{s'}\right) - (\bar{x}_{s'}, \bar{y}_{s'})\right) &\leqslant \psi_{s'}^{\rho_{s'}}\left((1-t)\left[\left(z^0, w^0_{s'}\right) - (\bar{x}_{s'}, \bar{y}_{s'})\right]\right) \\ &< \psi_{s'}^{\rho_{s'}}\left(\left(z^0, w^0_{s'}\right) - (\bar{x}_{s'}, \bar{y}_{s'})\right) \end{aligned}$$

which is evidently a contradiction. Thus we have (7.14) holding for all $s \in S_\rho$.

Consider (7.13) for a variable ρ and assume that for all $\rho > 0$ we have $s \in S_\rho \neq \varnothing$. We now consider two cases.

Case 1: $\bar{x}_s \neq z^0$ for some $s \in S$.

As we have a pure SIP in the x variables, and $\psi_s^{\rho_s}$ is an ICRF$^+$ for all $s \in S$ (see Assumption 7.3), we deduce form (7.13) that there exists a fixed $\varepsilon > 0$ such that

$$\begin{aligned} & \frac{1}{\rho_s}\left[\psi_s^{\rho_s}\left((z', w_s') - (\bar{x}_s, \bar{y}_s)\right) - \psi_{\rho_s}^{s}\left(\left(z^0, w_s^0\right) - (\bar{x}_s, \bar{y}_s)\right)\right] \\ = \; & \psi_s\left((1-t)\left[\left(z^0, w_s^0\right) - (\bar{x}_s, \bar{y}_s)\right]\right) - \psi_s\left(\left(z^0, w_s^0\right) - (\bar{x}_s, \bar{y}_s)\right) \\ \leqslant \; & -h_3\left((1-t), \left\|\left(z^0, w_s^0\right) - (\bar{x}_s, \bar{y}_s)\right\|\right) \leqslant -\varepsilon < 0. \end{aligned}$$

Then for $\rho_s > 0$ sufficiently large for $s \in S_\rho \neq \varnothing$ we have the following contradiction arising from (7.13):

$$\begin{aligned} 0 \;\leqslant\; & \sum_{s \in S_\rho} \rho_s p_s \frac{1}{\rho_s}\left[\psi_s^{\rho_s}\left((z', w_s') - (\bar{x}_s, \bar{y}_s)\right) - \psi_s^{\rho_s}\left(\left(z^0, w_s^0\right) - (\bar{x}_s, \bar{y}_s)\right)\right] \\ & + \sum_{s \notin S_\rho} p_s \psi_s^{\rho_s}\left((z', w_s') - (\bar{x}_s, \bar{y}_s)\right) < 0. \end{aligned}$$

Thus there exists $\bar{\rho} > 0$ such that $S_\rho = \varnothing$ for all $\rho \geqslant \bar{\rho}$ and $\bar{x} = z_s^0$ (fixed) for all $s \in S$.

Case 2: $\bar{x}_s = z^0$ for all $s \in S$

If $w^0 \neq \bar{y}$ then by (7.13) we have some $t \in (0,1)$ (independent of ρ) which satisfies

$$\begin{aligned} \left(z_{|S|}^0, w^0\right) \;\neq\; & \left(z_{|S|}^0, w_s'\right) := \left(z_{|S|}^0, w_s^0\right) + t\left(\left(z_{|S|}^0, \bar{y}_s\right) - \left(z_{|S|}^0, w_s^0\right)\right) \\ \in\; & B_\delta\left(z_{|S|}^0, w^0\right) \subseteq \mathbb{R}^{n \times |S|} \times \mathbb{R}^{m \times |S|}. \end{aligned}$$

and then

$$\begin{aligned} \Psi_\rho\left(\left(z_{|S|}^0, w^0\right) - \left(z_{|S|}^0, \bar{y}\right)\right) \;\leqslant\; & \Psi_\rho\left(\left(z_{|S|}^0, w'\right) - \left(z_{|S|}^0, \bar{y}\right)\right) \\ = \; & \Psi_\rho\left((1-t)\left(\left(z_{|S|}^0, \bar{y}\right) - \left(z_{|S|}^0, w^0\right)\right)\right) < \Psi_\rho\left(\left(z_{|S|}^0, \bar{y}\right) - \left(z_{|S|}^0, w^0\right)\right) \end{aligned}$$

which is a contradiction. Therefore in this case we have $(\bar{x}_s, \bar{y}_s) = (z^0, w_s^0) \in K_s$ for all $s \in S$. Hence if $\rho \geqslant \bar{\rho}$ we have the equality

$$\left(\min_{(x,y) \in K}\left\{f_\rho\left((x,y), \left(z_{|S|}^0, w^0\right)\right)\right\} =\right) \varphi_\rho\left(z^0, w^0\right) = \sum_{s \in S} p_s\left[c^\top z^0 + d_s^\top w_s^0\right]. \tag{7.15}$$

holding. Since the RHS is not dependent on ρ the minimum over f_ρ is constant for all $\rho \geqslant \bar{\rho}$.

It remains to show that if $\bar{\rho}$ is defined such that (z^0, w^0) is a local minimum of φ_ρ with respect to a fixed neighbourhood $B_{\bar{\delta}}(z^0, w^0)$ for all $\rho \geqslant \bar{\rho}$, then the point (z^0, w^0) uniquely obtains the minimum of f_ρ for all $\rho \geqslant \bar{\rho}$.

Let $(\hat{x}, \hat{y})$ be the minimiser of $f_\rho(\cdot, (z^0, w^0))$ for some sufficiently large $\rho \geqslant \bar{\rho}$. Then $(\hat{x}, \hat{y})$ cannot be anticipative since the penalty term and hence the minimum of f_ρ would increase as ρ increases (indeed the integral first stage assumption implies $\hat{x}_s = z^0$ for all $s \in S$ for sufficiently large ρ). If $(\hat{x}, \hat{y})$ is non-anticipative but not equal to $\left(z^0_{|S|}, w^0\right)$ then for sufficiently small $t \in (0, 1)$, $(1-t)\left(z^0_{|S|}, w^0\right) + t(\hat{x}, \hat{y})$ remains within $B_{\bar{\delta}}(z^0, w^0)$ (for all $\rho \geqslant \bar{\rho}$). Since the second part of Assumption 7.3 implies

$$\sum_{s \in S} p_s \psi_s^{\rho_s}\left(\left((1-t)\left(z^0, w^0_s\right) + t(\hat{x}_s, \hat{y}_s)\right) - (\hat{x}_s, \hat{y}_s)\right) < \sum_{s \in S} p_s \psi_s^{\rho_s}\left(\left(z^0, w^0_s\right) - (\hat{x}_s, \hat{y}_s)\right)$$

we have

$$f_\rho\left((\hat{x}, \hat{y}), (1-t)\left(z^0, w^0\right) + t(\hat{x}, \hat{y})\right) < f_\rho\left((\hat{x}, \hat{y}), \left(z^0, w^0\right)\right) = \varphi_\rho\left(z^0, w^0\right)$$

which contradicts our assumption that (z^0, w^0) is a local minimum of φ_ρ with respect to $B_{\bar{\delta}}(z^0, w^0)$. Therefore, $\left(z^0_{|S|}, w^0\right)$ uniquely obtains the minimum of f_ρ for all $\rho \geqslant \bar{\rho}$, as required.

Finally, observe that by (7.15) and

$$f_\rho(\left(z^0_{|S|}, w^0\right), \left(z^0_{|S|}, w^0\right)) = \sum_{s \in S} p_s \left[c^\top z^0 + d_s^\top w^0_s + \delta_{K_s}\left(z^0, w^0_s\right) + \psi_s^{\rho_s}\left(\left(z^0, w^0_s\right) - \left(z^0, w^0_s\right)\right)\right]$$

we see that $\left(z^0_{|S|}, w^0\right)$ can only attain the minimum $\sum_{s \in S} p_s \left[c^\top z^0 + d_s^\top w^0_s\right]$ if $(z^0, w^0_s) \in K_s$ for all $s \in S$.

■

Lemma 7.15 *Assume that $\psi_s^{\rho_s}$, for $s \in S$, satisfy the assumption to be an ICRF$^+$ (see Assumption 7.3). Then local minima of $\varphi_{\bar{\rho}}$ are also local minima of $\varphi_{\rho'}$ for $\rho' \geqslant \bar{\rho}$. Furthermore, if the SIP has no continuous variables in the first stage, local minima of $\varphi_{\bar{\rho}}$ are strict local minima of $\varphi_{\rho'}$ for $\rho' \geqslant \bar{\rho}$.*

Proof. Follows immediately from the proof of Lemmas 7.13 and 7.14. ■

7.2.2 Characterising Solutions of the SIP

We now characterise the solutions of the SIP (7.1) in terms of the minimisers of φ_ρ. Let $g(x, y) := \sum_{s \in S} p_s \left[c^\top x_s + d_s^\top y_s\right]$ and $F := \left\{(x, y) \in \mathbb{R}^n \times \mathbb{R}^{m \times |S|} \mid (x, y_s) \in K_s \text{ for all } s \in S\right\}$; note that this implicitly guarantees consensus for all elements of F.

Theorem 7.16 *Assume that $\psi_s^{\rho_s}$, for $s \in S$, satisfy the assumption to be an ICRF$^+$ (see Assumption 7.3). Suppose in addition that either:*

1. *We have a pure SIP that contains no real variables in both stages i.e. x and y_s for all $s \in S$ or*

2. *We have a pure SIP that contains no first stage real variables and given any $(x, y) \in F$ then $\left(x_{\mathcal{I}_x}, \left(y_{\mathcal{T}_y}, w^0_{\mathcal{R}_y}\right)\right) \in F$ implies $y_{\mathcal{R}_y} = w^0_{\mathcal{R}_y}$.*

Then there exists a $\bar{\rho} > 0$ such that for $\rho > \bar{\rho}$ any local minimum of φ_ρ lies in the feasible region F. Moreover for $\rho > \bar{\rho}$ large enough, the local minima of φ_ρ are exactly the points of F.

Proof. Lemma 7.14 demonstrates that any local minimum of φ_ρ lies in F; therefore we need only show that if $(z^0, w^0) \in F$ then (z^0, w^0) is a local minimum of φ_ρ for ρ sufficiently large. Let

$$\begin{aligned} g(x, y) &:= \sum_{s \in S} p_s \left[c^\top x_s + d_s^\top y_s\right] \\ f(x, y) &:= g(x, y) + \delta_{\Pi_s K_s}(x, y) \quad \text{and} \\ \psi_\rho(u, v) &:= \sum_{s \in S} p_s \psi_s^{\rho_s}(u_s, v_s). \end{aligned}$$

Consider $K := \varphi_\rho(z^0, w^0) + 2$, $\rho > \bar{\rho}$ sufficiently large so via (7.9) we know that the set

$$F_K\left(z^0, w^0\right) := \left\{(x, y) \in \Pi_{s \in S} K_s \mid f(x, y) + \psi_\rho\left(\left(z^0_{|S|}, w^0\right) - (x, y)\right) \leqslant K \right\} \tag{7.16}$$

is bounded. As f is lower semi-continuous and $\psi_\rho\left(\left(z^0_{|S|}, w^0\right) - (\cdot, \cdot)\right)$ is continuous their sum is also lower semi-continuous, implying that $F_K(z^0, w^0)$ is a closed set. Next note that since $\varphi_\rho(z^0, w^0)$ is defined as an infimum of continuous functions $(x, y) \mapsto g(x, y) + \psi_\rho\left(\left(z^0_{|S|}, w^0\right) - (x, y)\right)$, it is upper semi-continuous. Thus we may take (z', w') sufficiently close to (z^0, w^0) so that $\varphi_\rho(z', w') + 1 < \varphi_\rho(z^0, w^0) + 2 := K$; since $\psi_s^{\rho_s}$, is an ICRF$^+$ for $s \in S$ (see Assumption 7.3), for all $(x, y) \in F_K(z^0, w^0)$

with $\left(x_{\mathcal{I}_x}, y_{\mathcal{I}_y}\right) \neq \left(\left(z^0_{\mathcal{I}_x}\right)_{|S|}, w^0_{\mathcal{I}_y}\right)$ we have by for $\hat{\rho} := \min_s\{\rho_s\}$ that

$$\begin{aligned}
&\frac{1}{\hat{\rho}}\psi_\rho\left((x,y) - \left(z'_{|S|}, w'\right)\right) = \sum_{s\in S} p_s \frac{\rho_s}{\hat{\rho}}\psi_s\left(x_s - z', y_s - w'_s\right) \\
&\geqslant \sum_{s\in S} p_s\psi_s\left(x_s - z', y_s - w'_s\right) := \psi\left((x,y) - \left(z'_{|S|}, w'\right)\right) \\
&= \psi\left((x,y) - \left(z^0_{|S|}, w^0\right)\right) + \left[\psi\left((x,y) - \left(z'_{|S|}, w'\right)\right) - \psi\left((x,y) - \left(z^0_{|S|}, w^0\right)\right)\right] \\
&\geqslant \inf\left\{\psi\left((x,y) - \left(z^0_{|S|}, w^0\right)\right) \mid (x,y) \in F_K\left(z^0, w^0\right) \text{ with } \left(x_{\mathcal{I}_x}, y_{\mathcal{T}_y}\right) \neq \left(\left(z^0_{\mathcal{I}_x}\right)_{|S|}, w^0_{\mathcal{I}_y}\right)\right\} \\
&\qquad - \left[\psi\left((x,y) - \left(z^0_{|S|}, w^0\right)\right) - \psi^\rho\left((x,y) - \left(z'_{|S|}, w'\right)\right)\right] \\
&= \epsilon - \left[\psi\left((x,y) - \left(z^0_{|S|}, w^0\right)\right) - \psi\left((x,y) - \left(z'_{|S|}, w'\right)\right)\right] > 0
\end{aligned}$$

where

$$\begin{aligned}
&\inf\left\{\psi\left((x,y) - \left(z^0_{|S|}, w^0\right)\right) \mid (x,y) \in F_K\left(z^0, w^0\right) \text{ with } \left(x_{\mathcal{I}_x}, y_{\mathcal{T}_y}\right) \neq \left(\left(z^0_{\mathcal{I}_x}\right)_{|S|}, w^0_{\mathcal{I}_y}\right)\right\} \\
\geqslant\ &\inf\left\{h_1\left((x,y) - \left(z^0_{|S|}, w^0\right)\right) \mid (x,y) \in F_K\left(z^0, w^0\right) \text{ with } \left(x_{\mathcal{I}_x}, y_{\mathcal{T}_y}\right) \neq \left(\left(z^0_{\mathcal{I}_x}\right)_{|S|}, w^0_{\mathcal{I}_y}\right)\right\} := \epsilon > 0.
\end{aligned}$$

Take $\rho > 0$ sufficiently large so that $\frac{\epsilon}{2} > \frac{\Gamma}{\hat{\rho}} = \frac{\Gamma}{\min_s\{\rho_s\}} > 0$ where

$$\begin{aligned}
\Gamma &:= \max\left\{f(x,y) - f(z',w') \mid (x,y), (z',w') \in F_K\left(z^0, w^0\right)\right\} \\
&= \max\left\{g(x,y) - g(z',w') \mid (x,y), (z',w') \in F \cap F_K\left(z^0, w^0\right)\right\}
\end{aligned}$$

which is finite since g is continuous, $F_K\left(z^0, w^0\right)$ is compact and F is closed. Refer to the definition of an ICRF$^+$ given in Assumption 7.3 and assume we take (z', w') sufficiently close to (z^0, w^0) so that at least

$$\left[\psi\left((x,y) - \left(z^0_{|S|}, w^0\right)\right) - \psi\left((x,y) - \left(z'_{|S|}, w'\right)\right)\right] < \frac{\epsilon}{2}$$

and hence $\psi_\rho\left((x,y) - \left(z'_{|S|}, w'\right)\right) \geqslant \frac{\hat{\rho}\epsilon}{2} > \hat{\rho}\frac{\Gamma}{\min_s\{\rho_s\}} = \Gamma$.

Then for $(x,y) \in F_K\left(z^0, w^0\right)$ with $\left(x_{\mathcal{I}_x}, y_{\mathcal{I}_y}\right) \neq \left(\left(z^0_{\mathcal{I}_x}\right)_{|S|}, w^0_{\mathcal{I}_y}\right)$ we have

$$\begin{aligned}
&f(x,y) + \psi_\rho\left(\left(z'_{|S|}, w'\right) - (x,y)\right) = \left\{f(x,y) - f\left(z^0_{|S|}, w^0\right) + \psi_\rho\left(\left(z'_{|S|}, w'\right) - (x,y)\right)\right\} + f\left(z^0_{|S|}, w^0\right) \\
&\geqslant -\Gamma + \rho\frac{\epsilon}{2} + f\left(z^0_{|S|}, w^0\right) \geqslant -\Gamma + \Gamma + f\left(z^0_{|S|}, w^0\right) = f\left(z^0_{|S|}, w^0\right) \geqslant \varphi_\rho\left(z^0, w^0\right).
\end{aligned}$$

Next note that if $(x,y) \in F_{K-1}(z', w')$ and we have (z', w') is sufficiently close to (z^0, w^0) so that

$$\left|\psi_\rho\left((x,y) - \left(z'_{|S|}, w'\right)\right) - \psi_\rho\left((x,y) - \left(z^0_{|S|}, w^0\right)\right)\right| \leqslant 1$$

then

$$\begin{aligned}
K - 1 &\geqslant f(x,y) + \psi_\rho\left((x,y) - \left(z'_{|S|}, w'\right)\right) \\
&= f(x,y) + \psi_\rho\left((x,y) - \left(z^0_{|S|}, w^0\right)\right) + \left[\psi_\rho\left((x,y) - \left(z'_{|S|}, w'\right)\right) - \psi_\rho\left((x,y) - \left(z^0_{|S|}, w^0\right)\right)\right] \\
&\geqslant f(x,y) + \psi_\rho\left((x,y) - \left(z^0_{|S|}, w^0\right)\right) - 1
\end{aligned}$$

and so $F_{K-1}(z',w') \subseteq F_K(z^0,w^0)$ giving

$$\inf\{f(x,y)+\psi_\rho\left(\left(z'_{|S|},w'\right)-(x,y)\right) \quad | \quad (x,y)\in F_{K-1}(z',w') \text{ with } \left(x_{\mathcal{I}_x},y_{\mathcal{T}_y}\right)\neq\left(\left(z^0_{\mathcal{I}_x}\right)_{|S|},w^0_{\mathcal{I}_y}\right)\}$$
$$\geqslant \quad \varphi_\rho\left(z^0,w^0\right).$$

On the other hand for $(x,y)\in F_K(z^0,w^0)$ with $\left(x_{\mathcal{I}_x},y_{\mathcal{T}_y}\right)=\left(\left(z^0_{\mathcal{I}_x}\right)_{|S|},w^0_{\mathcal{I}_y}\right)$ we may take (z',w') sufficiently close to (z^0,w^0) so $\left(x_{\mathcal{I}_x},y_{\mathcal{T}_y}\right)=\left(\left(z^0_{\mathcal{I}_x}\right)_{|S|},w^0_{\mathcal{I}_y}\right)=\left(z'_{\mathcal{I}_x},w'_{\mathcal{T}_y}\right)$.

If the second stage variables are pure integer we then have $(z',w')=(z^0,w^0)=(x,y)$ and hence

$$\varphi_\rho(z',w')\geqslant\varphi_\rho\left(z^0,w^0\right)$$

for (z',w') sufficiently close to (z^0,w^0), which completes the proof for the first condition.

If the second stage variables are mixed integer then we know that φ_ρ is locally minimised in the integer variables at $\left(z^0_{\mathcal{I}_x},w^0_{\mathcal{I}_y}\right)$. Suppose the local minimum of φ_ρ lies at $\left(z^0_{\mathcal{I}_x},w^0_{\mathcal{I}_y},w'_{\mathcal{R}_y}\right)$; then by Lemma 7.14 we have $\left(z^0_{\mathcal{I}_x},w^0_{\mathcal{I}_y},w'_{\mathcal{R}_y}\right)\in F$ and the second condition gives $w'_{\mathcal{R}_y}=w^0_{\mathcal{R}_y}$, which implies that φ_ρ is locally minimised at (z^0,w^0). ■

We finish this section by making a comparison with the results of [18] and [37]. The following version of the strong augmented duality is a version of these results applied to SIP reproduced from Chapter 6. The results of [37] can be used to show that this theorem may extend to mixed integer SIP. In effect this is augmented Lagrangian duality for a SIP where we have put the dual multipliers to zero.

Theorem 7.17 *Suppose that $\psi_s^{\rho_s}$, for all $s\in S$, satisfies the assumption to be an ICRF$^+$ (see Assumption 7.3), and in addition we have a pure SIP that contains no first stage real variables. Then the there exists $\bar\rho\geqslant 0$ such that for all $\rho\geqslant\bar\rho$ we have the global minimisers of φ_ρ are exactly the optimal solutions of the SIP. Indeed*

$$\min_{(z,w)\in\mathbb{R}^n\times\mathbb{R}^{m\times|S|}}\varphi_\rho(z,w)=\zeta^{SIP}:=\min_{(x,y)}\left\{\sum_{s\in S}p_s\left[c^\top x+d_s^\top y_s\right]\mid(x,y_s)\in K_s \text{ for all } s\in S\right\},$$

the optimal value of the SIP.

Proof. We note that the previous proof shows that even when the second condition does not hold, the only way that a given (z^0,w^0) is not a local minimum of φ_ρ is for there to exist some other $\left(z^0,\left(w^0_{\mathcal{I}_y},w'_{\mathcal{R}_y}\right)\right)\in F$ (by Lemma 7.14 for which $\varphi_\rho(z^0,w^0)>\varphi_\rho\left(z^0,\left(w^0_{\mathcal{I}_y},w'_{\mathcal{R}_y}\right)\right)$. Thus the minimum of all such feasible solutions must result in the smallest and hence the optimal one. ■

7.2.3 Analysis of the Gauss-Seidel Step

Guass-Seidel is usually only studied for differentiable optimisation problems. We need to perform Gauss-Seidel for a nonsmooth functions of special structure and hence develop the necessary theory here to facilitate this analysis.

Definition 7.18 *A function $f : X \mapsto \mathbb{R} \cup +\infty$ is **Fréchet subdifferentiable** at x if there exists a **Fréchet subderivative** x^* such that*

$$\liminf_{h \to 0} \frac{f(x+h) - f(x) - \langle x^*, h \rangle}{\|h\|} \geqslant 0.$$

*We denote the collection of all such subderivatives by $\partial f(x)$, the **Fréchet subdifferential** of f at x.*

Lemmas 7.19 and 7.20 allow us to characterise the stationary points of φ_ρ as its partial minima, which are points satisfying the stopping criteria of the Gauss-Seidel algorithm. Denote the unit sphere by $S_1(0) := \{x \in X \mid \|x\| = 1\}$.

Lemma 7.19 *Suppose that $G(x,y) : X \times Y \to \mathbb{R}$ is such that $\nabla_x G(\cdot, y^*) \in C^1(X)$ exists and the Fréchet partial subdifferential $\partial_y G(x^*, y^*)$ exists. Suppose in addition that (x^*, y^*) is a partial minimum of G in that*

$$\begin{aligned} G(x^*, y) &\geqslant G(x^*, y^*) \quad \text{for all } y \quad \text{and} \\ G(x, y^*) &\geqslant G(x^*, y^*) \quad \text{for all } x. \end{aligned} \tag{7.17}$$

Then $(0,0) \in \partial G(x^*, y^*)$ (the Fréchet subdifferential) implying (x^*, y^*) is a stationary point. When $G(x,y)$ is jointly convex we have (x^*, y^*) a global minimum of G.

Proof. Clearly $\nabla_x G(x^*, y^*) = 0$ and $0 \in \partial_y G(x^*, y^*)$. We now show that

$$\partial G(x^*, y^*) = \{\nabla_x G(x^*, y^*)\} \times \partial_y G(x^*, y^*).$$

Consider $(z_x, z_y) \in \partial G(x^*, y^*)$ then by the definition of the Fréchet subdifferential

$$G(x,y) - G(x^*, y^*) - \langle z_x, x - x^* \rangle - \langle z_y, y - y^* \rangle \geqslant o(\|(x,y) - (x^*, y^*)\|) \tag{7.18}$$

within some neighbourhood of (x^*, y^*). Now fix $y = y^*$ and consider the restricted version of (7.18) which says that

$$G(x, y^*) - G(x^*, y^*) - \langle z_x, x - x^* \rangle \geqslant o(\|x - x^*\|)$$

in a neighbourhood of x^* (in X). But as $\nabla_x G(\cdot, y^*) \in C^1(X)$ we must have $\frac{G(x,y^*)-G(x^*,y^*)}{\|x-x^*\|} - \langle \nabla_x G(x^*, y^*), \frac{x-x^*}{\|x-x^*\|} \rangle \to 0$ as $\|x - x^*\| \to 0$ and so taking $\frac{x-x^*}{\|x-x^*\|} \to d \in S_1(0)$ we have

$$\left[\frac{G(x, y^*) - G(x^*, y^*)}{\|x - x^*\|} - \langle \nabla_x G(x^*, y^*), \frac{x - x^*}{\|x - x^*\|} \rangle \right] + \langle \nabla_x G(x^*, y^*) - z_x, \frac{x - x^*}{\|x - x^*\|} \rangle \geqslant \frac{o(\|x - x^*\|)}{\|x - x^*\|} \to 0$$

implying $\langle \nabla_x G(x^*, y^*) - z_x, d \rangle \geqslant 0$ for all $d \in S_1(0)$.

Hence $\nabla_x G(x^*, y^*) = z_x$ and $\partial G(x^*, y^*) = \{\nabla_x G(x^*, y^*)\} \times \partial_y G(x^*, y^*)$. Thus $\nabla_x G(x^*, y^*) = 0$ and $0 \in \partial_y G(x^*, y^*)$ implies $(0, 0) \in \partial G(x^*, y^*)$ with the rest following immediately from known results. ■

Lemma 7.20 *Suppose $G(x, y) = \sum_s g_s(y_s) + \sum_s h(x, y_s)$ where $g_s(\cdot)$ and $h(x, \cdot)$ are proper, convex functions and $h(\cdot, y) \in C^1(\mathbb{R}^n)$ for each y. Suppose in addition that (7.17) holds. Then (x^*, y^*) is a Fréchet stationary point of G and when h is jointly convex actually a global minimum.*

Proof. Since the subdifferential of the sum of convex functions is the sum of their subdifferentials, and the non-smooth component of G is separable in y_s for each $s \in S$, we have

$$\begin{aligned} 0 &\in \partial_y G(x^*, y^*) = \prod_s (\partial_{y_s} g_s(y_s^*) + \partial_{y_s} h(x^*, y_s^*)) \quad \text{and} \\ 0 &= \nabla_x G(x^*, y^*) = \sum_s \{\nabla_x h(x^*, y_s^*)\} \quad \text{exists.} \end{aligned}$$

Hence it follows from Lemma 7.19 that

$$(0, 0) \in \partial G(x^*, y^*) = \{\nabla_x G(x^*, y^*)\} \times \partial_y G(x^*, y^*).$$

This in turn implies (x^*, y^*) is a minimum for G when h is jointly convex (as the convex subdifferential equals the Fréchet subdifferential). ■

Remark 7.21 *The proof of Lemma 7.20 holds under the less restrictive assumption that $h(\cdot, y^*) \in C^1(\mathbb{R}^n)$ for the y^* given in (7.17).*

7.2.4 Properties of the Consensus Variable Update Step

In this section we address the existence of descent for φ_ρ when the Guass-Seidel step in x in φ_ρ is replaced by the consensus step. By $\operatorname{conv} K$ we denote the smallest convex set containing K.

Lemma 7.22 *Suppose $\psi \in C^1(\mathbb{R}^n)$ is a finite, proper convex function and K a set defined by linear inequality and the integrality constraints. Let $\hat{\psi}$ be the largest convex function on $\mathbb{R}^n \times \operatorname{conv} K$ majorised by:*

$$(\bar{x}, x) \mapsto \psi(\bar{x} - x) + \delta_{\mathbb{R}^n \times K}(\bar{x}, x).$$

Then $\hat{\psi}(\bar{x}, u) = \psi(\bar{x} - u)$ for all $(\bar{x}, u) \in \mathbb{R}^n \times K$ and so $\nabla_x \hat{\psi}(\cdot, u)(\bar{x})$ exists for each $u \in K$. Furthermore, if

$$\psi(\bar{x} - x) \geqslant \psi\left(\bar{x}^0 - x^0\right) \quad \forall (\bar{x}, x) \in \mathbb{R}^n \times K \tag{7.19}$$

holds then we have

$$\hat{\psi}(\bar{x} - x) \geqslant \psi\left(\bar{x}^0 - x^0\right) \quad \forall (\bar{x}, x) \in \mathbb{R}^n \times \operatorname{conv} K.$$

Proof. First we note that as $\psi \in C^1(\mathbb{R}^n)$ is a finite, proper convex function by definition

$$\psi \in U := \{g : \mathbb{R}^n \times \operatorname{conv} K \to \mathbb{R} \mid g \text{ convex and } g(\bar{x}, u) \leqslant \psi(\bar{x} - u), \forall (\bar{x}, u) \in \mathbb{R}^n \times K\}.$$

Now $\hat{\psi}(\bar{x}, x) = \sup\{g(\bar{x}, x) \mid g \in U\}$ and as $(\bar{x}, x) \mapsto \psi(\bar{x} - x)$ is jointly convex on $\mathbb{R}^n \times \operatorname{conv} K$ we have

$$\psi(\bar{x} - x) \leqslant \hat{\psi}(\bar{x}, x) \quad \forall (\bar{x}, x) \in \mathbb{R}^n \times \operatorname{conv} K$$

and so

$$\psi(\bar{x} - u) \leqslant \hat{\psi}(\bar{x}, u) \leqslant \psi(\bar{x} - u) \quad \forall (\bar{x}, x) \in \mathbb{R}^n \times K.$$

Hence $\hat{\psi}(\bar{x}, u) = \psi(\bar{x} - u)$ for all $(\bar{x}, u) \in \mathbb{R}^n \times K$ and so $\nabla_{\bar{x}} \hat{\psi}(\cdot, u)(\bar{x})$ exists for each $u \in K$.

Now assume that (7.19) holds. Let $(\bar{x}, x) = \sum_i \lambda_i (\bar{x}^i, x^i) \in \mathbb{R}^n \times \operatorname{conv} K$ where $\lambda_i \geqslant 0$, $\sum_i \lambda_i = 1$ and $(\bar{x}^i, x^i) \in \mathbb{R}^n \times K$ then

$$\sum_i \lambda_i \psi\left(\bar{x}^i - x^i\right) \geqslant \psi\left(\bar{x}^0 - x^0\right)$$

and so by [97, Theorem 5.6] we have

$$\begin{aligned} \hat{\psi}(\bar{x} - x) &= \inf_{\substack{(\bar{x}^i, x^i) \in \mathbb{R}^n \times K \\ \lambda_i \geqslant 0,\ \sum_i \lambda_i = 1}} \left\{ \sum_i \lambda_i \psi\left(\bar{x}^i - x^i\right) \mid (\bar{x}, x) = \sum_i \lambda_i \left(\bar{x}^i, x^i\right) \right\} \\ &\geqslant \psi\left(\bar{x}^0 - x^0\right) \quad \forall (\bar{x}, x) \in \mathbb{R}^n \times \operatorname{conv} K, \end{aligned}$$

as required for the second part of the lemma. ■

Note that in Lemma 7.23 we assume relatively complete recourse. This implies that $\operatorname{proj}_x K_s = X$ and as a consequence of the linearity of this projection, $\operatorname{proj}_x(\operatorname{conv} K_s) = \operatorname{conv} X$.

Lemma 7.23 *Suppose that $\psi_s^{\rho_s} \in C^1(\mathbb{R}^n)$ is a finite, proper convex function for all $s \in S$ and K a set defined by linear inequality and integrality constraints. Further, suppose that φ_ρ is as defined in* (7.3) *and that it is derived from a SIP* (7.1) *with relatively complete recourse.*

If x^k is not a minimum of $\varphi_\rho\left(\cdot, y^{k+1}\right)$ then for

$$\bar{x}^{k+1} \in \arg\min_{\bar{x}} \sum_{s\in S} p_s \psi_s^{\rho_s}\left(\bar{x} - u_s^{k+1}\right)$$

we have

$$\varphi_\rho\left(x^k, y^{k+1}\right) > \varphi_\rho\left(\bar{x}^{k+1}, y^{k+1}\right) \geqslant \min_{\bar{x}} \varphi_\rho\left(\bar{x}, y^{k+1}\right). \tag{7.20}$$

Proof. Suppose there is no strict decrease in step (7.7) above. Then we have

$$\min_{\bar{x}} \sum_{s\in S} p_s \psi_s^{\rho_s}\left(\bar{x} - u_s^{k+1}\right) = \sum_{s\in S} p_s \psi_s^{\rho_s}\left(x^k - u_s^{k+1}\right).$$

Denote

$$G(x, y, \bar{x}) = \sum_{s\in S} p_s\left[c^\top x_s + d_s^\top y_s\right] + \sum_{s\in S} p_s \psi_s^{\rho_s}\left(\bar{x} - x_s\right),$$

a proper convex function jointly in $(x, y, \bar{x})$. Then by construction we have

$$G\left(x, y, x^k\right) \geqslant G\left(u^{k+1}, v^{k+1}, x^k\right) \quad \forall\, (x, y) \in \Pi_{s\in S} K_s \tag{7.21}$$

and

$$G\left(u^{k+1}, v^{k+1}, \bar{x}\right) \geqslant G\left(u^{k+1}, v^{k+1}, x^k\right) \quad \forall \bar{x}. \tag{7.22}$$

Note that when $(\bar{x}, x, y) = \sum_i \lambda_i\left(\bar{x}^i, x^i, y^i\right) \in \mathbb{R}^n \times \operatorname{conv} \Pi_{s\in S} K_s = \mathbb{R}^n \times \Pi_{s\in S} \operatorname{conv} K_s$ (where $\sum_i \lambda_i = 1$ with $\lambda_i \geqslant 0$) we have

$$\begin{aligned}
\sum_i \lambda_i G\left(x^i, y^i, \bar{x}^i\right) &= \sum_i \lambda_i \left[\sum_{s\in S} p_s\left[c^\top x_s^i + d_s^\top y_s^i\right] + \sum_{s\in S} p_s \psi_s^{\rho_s}\left(\bar{x}^i - x_s^i\right)\right] \\
&= \sum_{s\in S} p_s\left[c^\top x_s + d_s^\top y_s\right] + \sum_{s\in S} p_s \sum_i \lambda_i \psi_s^{\rho_s}\left(\bar{x}^i - x_s^i\right).
\end{aligned}$$

We now form the largest convex function H (defined on $(\mathbb{R}^n \times \Pi_{s\in S} \operatorname{conv} K)$) which is majorised by $G|_{\Pi_{s\in S}\mathbb{R}^n \times K_s}$. Denote

$$\hat{\psi}^s_{\rho_s}(\bar{x}, x_s) := \inf\left\{\sum_i \lambda_i \psi^{\rho_s}_s\left(\bar{x}^i - x^i_s\right) \mid \bar{x}^i \in \mathbb{R}^n,\ x^i_s \in X,\ \sum_i \lambda_i = 1,\ \lambda_i \geqslant 0,\ x_s = \sum_i \lambda_i x^i_s\right\} \quad (7.23)$$

where $\hat{\psi}^{\rho_s}_s$ is the largest convex function majorised by $\psi^{\rho_s}_s(\bar{x} - x)$ on $\mathbb{R}^n \times X$ (noting that by relatively complete recourse then $P_x K_s = X$) and noting that also $(\bar{x}, x) \mapsto \hat{\psi}^{\rho_s}_s(\bar{x}, x_s)$ is jointly convex on $\mathbb{R}^n \times \operatorname{conv} K_s$. Next note that as $\psi^{\rho_s}_s \in C^1(\mathbb{R}^n)$ and $\hat{\psi}^{\rho_s}_s\left(\bar{x}, u^{k+1}\right) = \psi^{\rho_s}_s\left(\bar{x} - u^{k+1}\right)$ for a given $\left(u^{k+1}, v^{k+1}\right) \in K_s$ (recall that relatively complete recourse gives $u^{k+1} \in X$) and for all $\bar{x} \in \mathbb{R}^n$ we have $\nabla_{\bar{x}} \hat{\psi}^{\rho_s}_s\left(\cdot, u^{k+1}\right)$ existing. As $\left(u^{k+1}, v^{k+1}\right) \in \Pi_s K_s$ then

$$H\left(u^{k+1}, v^{k+1}, \bar{x}\right) = G\left(u^{k+1}, v^{k+1}, \bar{x}\right).$$

Thus by (7.21), (7.22) and Lemma 7.22 it follows that

$$\begin{aligned} H\left(x, y, x^k\right) &\geqslant H\left(u^{k+1}, v^{k+1}, x^k\right) \quad \text{for all } (x, y) \in \Pi_{s\in S} \operatorname{conv} K_s \quad \text{and} \\ H\left(u^{k+1}, v^{k+1}, \bar{x}\right) &\geqslant H\left(u^{k+1}, v^{k+1}, x^k\right) \quad \text{for all } \bar{x}. \end{aligned}$$

Hence by Lemma 7.20 we have $\left(u^{k+1}, v^{k+1}, x^k\right)$ a minimum for H. That is

$$\begin{aligned} H(x, y, \bar{x}) &\geqslant H\left(u^{k+1}, v^{k+1}, x^k\right) \\ &= G\left(u^{k+1}, v^{k+1}, x^k\right) = \varphi_\rho\left(x^k, y^{k+1}\right) \quad \text{for all } (x, y, \bar{x}). \end{aligned}$$

Restricting $(x, y, \bar{x}) \in \Pi_{s\in S} K_s \times \mathbb{R}^n$ this implies for all $(x, y, \bar{x}) \in \Pi_{s\in S} K_s \times \mathbb{R}^n$ that

$$\begin{aligned} G(x, y, \bar{x}) &\geqslant G\left(u^{k+1}, v^{k+1}, x^k\right) \\ &= G\left(u^{k+1}, v^{k+1}, x^k\right) = \varphi_\rho\left(x^k, y^{k+1}\right) \\ \text{so } \varphi_\rho\left(\bar{x}, y^{k+1}\right) &= \min_{(x,y)\in\Pi_{s\in S} K_s} \sum_{s\in S} p_s\left[c^\top x_s + d_s^\top y_s\right] + \sum_{s\in S} p_s \psi^{\rho_s}_s\left(\left(\bar{x}, y^{k+1}\right) - (x_s, y_s)\right) \\ &\geqslant \min_{(x,y)\in\Pi_{s\in S} K_s} \sum_{s\in S} p_s\left[c^\top x_s + d_s^\top y_s\right] + \sum_{s\in S} p_s \psi^{\rho_s}_s(\bar{x} - x_s) \\ &= \min_{(x,y)\in\Pi_{s\in S} K_s} G(x, y, \bar{x}) \geqslant \varphi_\rho\left(x^k, y^{k+1}\right) \quad \text{for all } \bar{x} \in \mathbb{R}^n \end{aligned}$$

a contradiction to assumption that $x^k \notin \arg\min \varphi_\rho\left(\cdot, y^{k+1}\right)$. ■

Define

$$\operatorname{diag} \mathbb{R}^{n\times|S|} := \left\{x \in \mathbb{R}^{n\times|S|} \mid x_s = x \in \mathbb{R}^n\right\}.$$

So $x_{|S|} \in \operatorname{diag} \mathbb{R}^{n\times|S|}$ for any $x \in \mathbb{R}^n$. Denote $\hat{\varphi}_\rho\left(x_{|S|}, y\right) := \varphi_\rho(x, y)$. Note that $\left(x_{|S|}, y, r\right) \in \operatorname{epi} \hat{\varphi}_\rho$ if and only if $r \geqslant \hat{\varphi}_\rho\left(x_{|S|}, y\right)$. Since $\operatorname{epi} \hat{\varphi}_\rho$ is a superset of $\operatorname{epi} f + \operatorname{epi} \psi_\rho$ restricted to $\left(\operatorname{diag} \mathbb{R}^{n\times|S|}\right) \times \mathbb{R}^{m\times|S|} \times \mathbb{R}$, for all $(x', y', r) \in \operatorname{epi} f$ and $\left(x_{|S|} - x', y - y', r'\right) \in \operatorname{epi} \psi^\rho$ we have $\left(x_{|S|}, y, r + r'\right) \in \operatorname{epi} \hat{\varphi}_\rho$ or $\hat{\varphi}_\rho\left(x_{|S|}, y\right) \leqslant r + r'$ implying $\varphi_\rho(x, y) \leqslant f(x', y') + \psi_\rho\left(x_{|S|} - x', y - y'\right)$ for all $(x', y') \in \Pi_{s\in S} K_s$ or

$$\varphi_\rho(x, y) \leqslant \min_{(x', y') \in \Pi_{s\in S} K_s} \left[f(x', y') + \psi_\rho\left(x_{|S|} - x', y - y'\right)\right].$$

Thus geometrically $\operatorname{epi} \hat{\varphi}_\rho$ is the largest extended real valued function defined on $\left(\operatorname{diag} \mathbb{R}^{n\times|S|}\right) \times \mathbb{R}^{m\times|S|}$ whose epigraph contains the sum $\operatorname{epi} f + \operatorname{epi} \psi_\rho$ restricted to $\left(\operatorname{diag} \mathbb{R}^{n\times|S|}\right) \times \mathbb{R}^{m\times|S|} \times \mathbb{R}$. The following lemma is derived from [19, Theorem 3.3].

Lemma 7.24 *Suppose that the regularized function $\varphi_{\bar{\rho}}(x_0, y_0)$ is greater than $-\infty$ for some (x_0, y_0) and that ψ_s are ICRF+ with ψ_s are Lipschitz continuous (with a global Lipschitz constant). Then φ_ρ is finitely valued for $\rho \geqslant \bar{\rho}$ and globally Lipschitz continuous.*

Proof. Clearly $\varphi_\rho(x, y) \geqslant \varphi_{\bar{\rho}}(x, y)$ for all y and $\rho \geqslant \bar{\rho}$. Thus when $\varphi_{\bar{\rho}}(x, y) > -\infty$ for all y then $\varphi_\rho(x, y) > -\infty$ for all $\rho \geqslant \bar{\rho}$. Moreover as $\varphi_\rho(\bar{x}, \bar{y}) \leqslant f(x, y) + \psi_\rho\left(\left(\bar{x}_{|S|}, \bar{y}\right) - (x, y)\right) < +\infty$ for any $(x, y) \in \Pi_{s\in S} K_s$ finiteness of φ_ρ follows. Thus if we can show $\varphi_\rho(x, y)$ is bounded away from negative infinity for (x, y) then we would have shown it to be finite valued.

Let C_μ denote the cone $\operatorname{epi} \mu\|\cdot\|$ in $\mathbb{R}^{(n+m)\times|S|+1}$. Note next that the Lipschitz continuity property of $\frac{1}{\rho}\psi_\rho$ corresponds to the existence of a Lipschitz constant $\mu > 0$ such that $\operatorname{epi} \frac{1}{\rho}\psi_\rho + C_\mu \subseteq \operatorname{epi} \frac{1}{\rho}\psi_\rho$. Now $\operatorname{epi} \frac{1}{\rho}\psi_\rho + C_\mu \subseteq \operatorname{epi} \frac{1}{\rho}\psi_\rho$ in turn implies $\operatorname{epi} \psi_\rho + C_{\rho\mu} \subseteq \operatorname{epi} \psi_\rho$ with $\operatorname{int} C_{\rho\mu} \neq \varnothing$ for all ρ. Suppose $\varphi_\rho(x, y) = -\infty$ for some (x, y) then

$$(x_{|S|}, y, -n) \in \operatorname{epi} f + \operatorname{epi} \psi_\rho$$

for all $n \in \mathbf{Z}^+$ and so (using $\operatorname{int} C_{\rho\mu} \neq \varnothing$)

$$\begin{aligned}
\mathbb{R}^{|S|\times(n+m)+1} \subseteq \bigcup_{n=1}^{\infty} \left((x_{|S|}, y, -n) + C_{\rho\mu}\right) &\subseteq \left[(\operatorname{epi} f + \operatorname{epi} \psi_\rho) + C_{\rho\mu}\right] \cap \left[\left(\operatorname{diag} \mathbb{R}^{n\times|S|}\right) \times \mathbb{R}^{m\times|S|} \times \mathbb{R}\right] \\
&\subseteq \left[\operatorname{epi} f + (\operatorname{epi} \psi_\rho + C_{\rho\mu})\right] \cap \left[\left(\operatorname{diag} \mathbb{R}^{n\times|S|}\right) \times \mathbb{R}^{m\times|S|} \times \mathbb{R}\right] \\
&\subseteq \left[\operatorname{epi} f + \operatorname{epi} \psi_\rho\right] \cap \left[\left(\operatorname{diag} \mathbb{R}^{n\times|S|}\right) \times \mathbb{R}^{m\times|S|} \times \mathbb{R}\right] \subseteq \operatorname{epi} \hat{\varphi}_\rho
\end{aligned}$$

implying $\varphi_\rho \equiv -\infty$, contradicting $\varphi_\rho(x_0, y_0)$ finite. Let $\varphi_\rho(x,y) < \alpha$ and take (x', y') such that $f(x', y') + \psi_\rho(\big(x_{|S|}, y\big) - (x', y')) < \alpha$. Let $(u, v) \in \mathbb{R}^{n+m\times|S|}$ then

$$\begin{aligned}
\varphi_\rho(u,v) &\leqslant f(x',y') + \psi_\rho(\big(u_{|S|}, v\big) - (x',y')) \\
&\leqslant \big(f(x',y') + \psi_\rho(\big(x_{|S|}, y\big) - (x',y'))\big) + \rho\left(\frac{1}{\rho}\psi_\rho(\big(u_{|S|}, v\big) - (x',y')) - \frac{1}{\rho}\psi_\rho(\big(x_{|S|}, y\big) - (x',y'))\right) \\
&< \alpha + \rho\left(\frac{1}{\rho}\psi_\rho(\big(u_{|S|}, v\big) - (x',y')) - \frac{1}{\rho}\psi_\rho(\big(x_{|S|}, y\big) - (x',y'))\right) \leqslant \alpha + \mu\| \left(u,v\right) - (x,y) \|,
\end{aligned}$$

where we have used the Lipschitz continuity of $\frac{1}{\rho}\psi^\rho$ again. As this holds for all $\alpha > \varphi_\rho(x,y)$ we have $\varphi_\rho((u,v) \leqslant \varphi_\rho(x,y) + \mu\|z - y\|$. As $(u,v), (x,y) \in \mathbb{R}^{n+m\times|S|}$ are arbitrary this completes the proof. ■

7.2.5 Final Results

In the next lemma we will require $\nabla_x \varphi_\rho\left(\cdot, y^*\right)$ to exist; this follows from the non-emptiness of the partial Fréchet subdifferential since $\varphi_\rho\left(x, y^*\right) \geqslant \varphi_\rho\left(x^*, y^*\right)$ for all x would imply $0 \in \partial_x \varphi_\rho\left(\cdot, y^*\right)\left(x^*\right) \neq \varnothing$ and

$$\begin{aligned}
(x,y) \mapsto f_\rho((x,y), (\bar{x}, \bar{y})) &:= \sum_{s\in S} p_s\left[c^\top x_s + d_s^\top y_s + \psi_s^{\rho_s}\left((\bar{x}, \bar{y}_s) - (x_s, y_s)\right)\right] \\
\text{gives} \quad \varphi_\rho\left(\bar{x}, \bar{y}\right) &= \min_{(x,y)\in\Pi_s K_s} f_\rho((x,y), (\bar{x}, \bar{y})). \qquad (7.24)
\end{aligned}$$

We associate $z \equiv (x,y) \in \Pi_s K_s \equiv K$ with $z \in K$, $f_\rho((x,y), (\bar{x}, \bar{y}))$ with $f((x,y), z)$ and $\varphi_\rho\left(\bar{x}, \bar{y}\right)$ with $\varphi\left(\bar{x}, \bar{y}\right)$ in the following results.

Proposition 7.25 *Suppose $(x,y) \to f((x,y), z) \in C^2\left(\mathbb{R}^{n+m}\right)$ and K is a compact set. Define*

$$\begin{aligned}
\mathcal{K}\left(x,y\right) &:= \{z \in K \mid \varphi\left(x,y\right) = f((x,y), z)\} \qquad \textit{and} \\
\varphi\left(x,y\right) &= \min_{z\in K} f((x,y), z).
\end{aligned}$$

Then $(x,y) \to \mathcal{K}\left(x,y\right)$ has a closed graph and the non-emptiness of the partial Fréchet subdifferential $\partial_x \varphi\left(\bar{x}, \bar{y}\right) \neq \varnothing$ implies $\nabla_x \varphi\left(\bar{x}, \bar{y}\right)$ exists.

If in addition we assume:

1. *The mapping $(x,y) \to \varphi\left(x,y\right)$ is Lipschitz continuous.*

2. *The functions* $(x,y) \to f((x,y),z)$ *are strictly convex at* $(\bar{x},\bar{y})$ *for every* $z \in K$.

3. *The functions* $(x,y,z) \mapsto \langle \nabla_{(x,y)} f((x,y),z), h\rangle$ *and* $(x,y,z) \mapsto \langle \nabla^2_{(x,y)} f((x,y),z)h, h\rangle$ *is jointly lower semi-continuous for any fixed* h.

4. *We have* $(0,0) \in \partial_x \varphi(\bar{x},\bar{y})$.

   Then the one side directional derivatives

$$\begin{aligned} \varphi'((\bar{x},\bar{y}),h) &= \min_{\bar{z}\in K(\bar{x},\bar{y})} \langle \nabla_{(x,y)} f((\bar{x},\bar{y}),\bar{z}), h\rangle = 0 \quad \textit{and} \\ \varphi''((\bar{x},\bar{y}),h) &= \min_{\bar{z}\in K(\bar{x},\bar{y})} \langle \nabla^2_{(x,y)} f((\bar{x},\bar{y}),\bar{z}), h\rangle > 0 \quad \textit{for all } h \end{aligned}$$

   and so $(\bar{x},\bar{y})$ *is a strict local minimum.*

Proof. We first show that $(x,y) \mapsto \mathcal{K}(x,y)$ has a closed graph. Let $(x_k,y_k) \to (\bar{x},\bar{y})$ and $z_k \in K(x_k,y_k)$ then there exists a subsequence $z_{k_m} \to \bar{z} \in K$ with

$$\varphi(x_{k_m},y_{k_m}) = f((x_{k_m},y_{k_m}),z_{k_m}) \to f((\bar{x},\bar{y}),\bar{z}).$$

By the Berge maximum principle [8, Theorem 6.3.8] we have $(x,y) \to \varphi(x,y)$ continuous and so

$$\varphi(\bar{x},\bar{y}) = f((\bar{x},\bar{y}),\bar{z})$$

and so $\bar{z} \in \mathcal{K}(\bar{x},\bar{y})$.

Now consider $z^* \in \partial_x \varphi(\bar{x},\bar{y})$ and so locally around $\bar{x}$ we have

$$\varphi(x,\bar{y}) - \varphi(\bar{x},\bar{y}) - \langle x^*, x-\bar{x}\rangle \geqslant o(\|x-\bar{x}\|)$$

and for $\bar{z} \in K(\bar{x},\bar{y})$ we have $f((x,\bar{y}),\bar{z}) \geqslant \varphi(x,\bar{y})$ and $f((\bar{x},\bar{y}),\bar{z}) = \varphi(\bar{x},\bar{y})$ so

$$f((x,\bar{y}),\bar{z}) - f((\bar{x},\bar{y}),\bar{z}) - \langle x^*, x-\bar{x}\rangle \geqslant o(\|x-\bar{x}\|).$$

That is, for $\frac{x-\bar{x}}{\|x-\bar{x}\|} \to d \in S_1(0)$ as $x \to \bar{x}$, we have

$$\frac{f((x,\bar{y}),\bar{z}) - f((\bar{x},\bar{y}),\bar{z})}{\|x-\bar{x}\|} - \left\langle x^*, \frac{x-\bar{x}}{\|x-\bar{x}\|}\right\rangle \geqslant \frac{o(\|x-\bar{x}\|)}{\|x-\bar{x}\|}$$

$$\text{implying} \qquad \langle \nabla_x f((\bar{x},\bar{y}),\bar{z}) - x^*, d\rangle \geqslant 0 \quad \text{for all } d \in S_1(0).$$

Hence

$$\nabla_x f((\bar{x},\bar{y}),\bar{z}) = x^*$$

and so $\partial_x \varphi(\bar{x},\bar{y}) = \{\nabla_x f((\bar{x},\bar{y}),\bar{z})\}$ implying $\nabla_x \varphi(\bar{x},\bar{y})$ exists.

Finally we assume in addition that $(x,y) \to f((x,y),z)$ are strictly convex for every $z \in K$ then $(0,0) \in \partial\varphi(\bar{x},\bar{y})$. Then the previous argument can be adapted to show that $(0,0) = \nabla_{(x,y)}\varphi(\bar{x},\bar{y})$ exists. We next characterise that the one side directional derivative in the direction $h := (h_1,h_2)$

$$\begin{aligned}\varphi'_x((\bar{x},\bar{y}),h) := & \liminf_{t\downarrow 0, h'\to h} \frac{1}{t}\left(\varphi(\bar{x}+th'_1,\bar{y}+th'_2) - \varphi(\bar{x},\bar{y})\right)\\ = & \min_{\bar{z}\in K(\bar{x},\bar{y})} f'((\bar{x},\bar{y}),h) = \min_{\bar{z}\in K(\bar{x},\bar{y})} \langle \nabla_{(x,y)} f((\bar{x},\bar{y}),\bar{z}), h\rangle.\end{aligned}$$

First note that as $(x,y) \mapsto \varphi(x,y)$ is Lipschitz continuous we have the first equality below (the second inequality follows from definitions) and so for $\bar{z} \in K(\bar{x},\bar{y})$

$$\begin{aligned}\varphi'((\bar{x},\bar{y}),h) &= \liminf_{t\downarrow 0} \frac{1}{t}\left(\varphi(\bar{x}+th_1,\bar{y}+th_2) - \varphi(\bar{x},\bar{y})\right)\\ &\leqslant \liminf_{t\downarrow 0} \frac{1}{t}\left(f((\bar{x}+th_1,\bar{y}+th_2),\bar{z}) - f((\bar{x},\bar{y}),\bar{z})\right) = \langle \nabla_{(x,y)} f((\bar{x},\bar{y}),\bar{z}), h\rangle.\end{aligned}$$

Hence $\min_{\bar{z}\in K(\bar{x},\bar{y})}\langle \nabla_x f((\bar{x},\bar{y}),\bar{z}), h\rangle \geqslant \varphi'((\bar{x},\bar{y}),h)$. Now let $t^k \downarrow 0$ be such that

$$\varphi'((\bar{x},\bar{y}),h) = \liminf_k \frac{1}{t^k}\left(\varphi\left(\bar{x}+t^k h_1,\bar{y}+t^k h_2\right) - \varphi(\bar{x},\bar{y})\right).$$

Now take $z^k \in K\left(\bar{x}+t^k h_1,\bar{y}+t^k h_2\right)$ such that $\varphi\left(\bar{x}+t^k h_1,\bar{y}+t^k h_2\right) = f\left(\left(\bar{x}+t^k h_1,\bar{y}+t^k h_2\right),z^k\right)$ for all k. Using the closed graph property of $K(\cdot)$ and the compactness of K we may assume (by taking subsequences and re-numbering accordingly) that $z^k \to \bar{z} \in K(\bar{x},\bar{y})$. Then by the mean value theorem

$$\begin{aligned}\varphi'((\bar{x},\bar{y}),h) &= \lim_k \frac{1}{t^k}\left(f\left(\left(\bar{x}+t^k h_1,\bar{y}+t^k h_2\right),z^k\right) - f((\bar{x},\bar{y}),\bar{z})\right)\\ &\geqslant \inf_{\mu\in(0,1)} \liminf_k \langle \nabla f\left(\left(\bar{x}+\mu t^k h_1,\bar{y}+\mu t^k h_2\right),z^k\right), h\rangle\\ &= \langle \nabla_{(x,y)} f((\bar{x},\bar{y}),\bar{z}), h\rangle \geqslant \min_{\bar{z}\in K(\bar{x},\bar{y})} \langle \nabla_{(x,y)} f((\bar{x},\bar{y}),\bar{z}), h\rangle,\end{aligned}$$

showing equality along with

$$(0,0) = \langle \nabla_{(x,y)}\varphi(\bar{x},\bar{y}), h\rangle = \langle \nabla_{(x,y)} f((\bar{x},\bar{y}),\bar{z}), h\rangle \quad \text{for all } \bar{z} \in K(\bar{x},\bar{y}).$$

We now use the fact that each $(x,y) \mapsto f((x,y),\bar{z})$ are strictly convex, from which it follows that for any h we have $\langle \nabla^2_{(x,y)} f((\bar{x},\bar{y}),\bar{z})h, h\rangle > 0$. We now show that

$$\begin{aligned}f''(\bar{x},\bar{y},h) &:= \liminf_{t\downarrow 0, h'\to h} \tfrac{2}{t^2}\left(f(\bar{x}+th'_1,\bar{y}+th'_2) - f(\bar{x},\bar{y}) - \langle (0,0),(h'_1,h'_2)\rangle\right)\\ &\geqslant \min_{\bar{z}\in K(\bar{x},\bar{y})} \langle \nabla^2_{(x,y)} f((\bar{x},\bar{y}),\bar{z})h, h\rangle > 0\end{aligned}$$

which implies the existence of a strict local minimum (see [106]). Take $t_k \downarrow 0$ and $h^k \to h$ be such that

$$f''(\bar{x}, \bar{y}, h) = \liminf_k \frac{2}{t_k^2} \left(f\left(\bar{x} + t_k h_1^k, \bar{y} + t_k h_2^k\right) - f(\bar{x}, \bar{y}) \right).$$

Let $z^k \in K\left(\bar{x} + t_k h_1^k, \bar{y} + t_k h_2^k\right)$ and by taking subsequences and re-numbering we may assume $z^k \to \bar{z} \in K(\bar{x}, \bar{y})$. Applying the second order Taylor approximation we have

$$f\left(\left(\bar{x} + t_k h_1^k, \bar{y} + t_k h_2^k\right), z^k\right) - f\left((\bar{x}, \bar{y}), z^k\right) = \frac{t_k^2}{2} \langle \nabla^2_{(x,y)} f\left(\left(\bar{x} + \mu^k t_k h_1^k, \bar{y} + \mu^k t_k h_2^k\right), z^k\right) h^k, h^k \rangle$$

for some $\mu^k \in (0, 1)$. Hence

$$\begin{aligned} f''(\bar{x}, \bar{y}, h) &= \liminf_k \langle \nabla^2_{(x,y)} f\left(\left(\bar{x} + \mu^k t_k h_1^k, \bar{y} + \mu^k t_k h_2^k\right), z^k\right) h^k, h^k \rangle \\ &\geqslant \langle \nabla^2_{(x,y)} f\left((\bar{x}, \bar{y}), \bar{z}\right) h, h \rangle \geqslant \min_{z \in K(\bar{x}, \bar{y})} \langle \nabla^2_{(x,y)} f\left((\bar{x}, \bar{y}), z\right) h, h \rangle. \end{aligned}$$

Now suppose $\min_{z \in K(\bar{x}, \bar{y})} \langle \nabla^2_{(x,y)} f\left((\bar{x}, \bar{y}), z\right) h, h \rangle = 0$ for any h then by the compactness of $K(\bar{x}, \bar{y})$ and the lower semi-continuity of $z \mapsto \langle \nabla^2_{(x,y)} f((x, y), z) h, h \rangle$ the minimum is attained and we have the contradiction to the strict convexity of $\langle \nabla^2_{(x,y)} f\left((\bar{x}, \bar{y}), z\right) h, h \rangle = 0$ for some $z \in K(\bar{x}, \bar{y})$. ■

We will need the following result from [8, Propositions 1.3.5 and 1.3.6] or [99, Proposition 7.30 and Theorem 7.31].

Proposition 7.26 *[8, Propositions 1.3.5 and 1.3.6] Let $\{f_\lambda\}_{\lambda \in \Lambda}$ be a net of lower semi-continuous functions to the extended real line defined on a fixed Hausdorf space X. Suppose $f : X \to \mathbb{R}_{+\infty} := \mathbb{R} \cup \{+\infty\}$ is lower semi-continuous.*

1. *Suppose $\lambda \mapsto \operatorname{epi} f_\lambda$ is a lower semi-continuous with respect to f as a multi-function i.e. for any open set V in $X \times \mathbb{R}$ such that $\operatorname{epi} f \cap V \neq \varnothing$ we have $\operatorname{epi} f_\lambda \cap V \neq \varnothing$ eventually.*

2. *The set $\operatorname{epi} f$ contains all cluster points of nets $\{(x_\lambda, \alpha_\lambda)\}_{\lambda \in \Lambda}$ where $(x_\lambda, \alpha_\lambda) \in \operatorname{epi} f_\lambda$.*

3. *There exists a net $\{x_\lambda\}_{\lambda \in \Lambda}$ such that $\lim_\lambda (f(x_\lambda) - \inf_X f) = 0$.*

If $\{x_\lambda\}$ converges to x then $\inf_X f = \lim_\lambda \inf_X f_\lambda$ and $x \in \arg\min f$.

Proposition 7.27 demonstrates that the mapping defined by one full Gauss-Seidel iterate of Algorithm 7.3, forms a multi-value mapping which has a closed graph. This allows us to analyse the descent properties of Algorithm 7.3. To do so we will need to exploit the previous result regarding the convergence of the objective values of parametrised problems.

Proposition 7.27 *Suppose that $\psi_s^{\rho_s} \in C^1(\mathbb{R}^n)$ is a finite, coercive, proper convex function for all $s \in S$ and the associated SIP has pure integer first stage variables. Consider the following iterated optimisation process: Given (x, y) for all $s \in S$ to find $(u_s, v_s) \in K_s$ (a closed compact set) that is a solution to:*

$$\min_{(u_s,v_s)\in K_s} c^\top u_s + d_s^\top v_s + \psi_s^{\rho_s}(x - u_s), \tag{7.25}$$

place $y^{+1} = v_s$ and then place

$$x^{+1} \in \arg\min_x \sum_{s\in S} p_s \psi_s^{\rho_s}(x - u_s) \tag{7.26}$$

returning (x^{+1}, y^{+1}). Then the mapping $(x, y) \mapsto (x^{+1}, y^{+1})$ has a closed graph.

Proof. Suppose $\left(x^k, y^k\right) \to (x, y)$ and for each k we have $(y^k)^{+1} = v_s^{k+1}$ where $\left(u_s^{k+1}, v_s^{k+1}\right)$ solves

$$\min_{(u_s,v_s)\in K_s} c^\top u_s + d_s^\top v_s + \psi_s^{\rho_s}\left(x^k - u_s\right),$$

and

$$(x^k)^{+1} \in \arg\min_x \sum_{s\in S} p_s \psi_s^{\rho_s}\left(x - u_s^{k+1}\right). \tag{7.27}$$

We show that if $\left((x^k)^{+1}, (y^k)^{+1}\right) \to (x^{+1}, y^{+1})$ then (x^{+1}, y^{+1}) solve (7.25) and (7.26). Both problems are parametrised optimisation problems. Denote $f_k(u_s, v_s) := c^\top u_s + d_s^\top v_s + \psi_s^{\rho_s}\left(x^k - u_s\right) + \delta_{K_s}(u_s, v_s)$, $f(u_s, v_s) := c^\top u_s + d_s^\top v_s + \psi_s^{\rho_s}(x - u_s) + \delta_{K_s}(u_s, v_s)$ and $g(x, u) := \sum_{s\in S} p_s \psi_s^{\rho_s}(x - u_s)$. First note that without loss of generality we may take a member of the neighbourhood basis $V = B_\delta(\bar{u}_s, \bar{v}_s) \times (\alpha, \beta)$ and suppose

$$\operatorname{epi} f \cap [B_\delta(\bar{u}_s, \bar{v}_s) \times (\alpha, \beta)] \neq \varnothing.$$

Then there exists $(u_s', v_s') \in B_\delta(\bar{u}_s, \bar{v}_s) \cap K_s$ for which $f(u_s', v_s') \in (\alpha, \beta)$. Now consider $x^k \to x$ and note that by continuity of $\psi_s^{\rho_s}$ we have $\left|\psi_s^{\rho_s}\left(x^k - u_s'\right) - \psi_s^{\rho_s}(x - u_s')\right| \to 0$ and hence $f_k(u_s', v_s') \in (\alpha, \beta)$ eventually. Now take a cluster point (u_s, v_s, α) of a sequence $\left(u_s^k, v_s^k, \alpha_k\right) \in \operatorname{epi} f_k$ (with necessarily $\left(u_s^k, v_s^k\right) \in K_s$). Note that we do not need to consider nets in finite dimensions. Again by continuity we get

$$\begin{aligned} f_k\left(u_s^k, v_s^k\right) &= c^\top u_s^k + d_s^\top v_s^k + \psi_s^{\rho_s}\left(x^k - u_s^k\right) \leqslant \alpha_k \\ \text{implying} \quad f(u_s, v_s) &= c^\top u_s + d_s^\top v_s + \psi_s^{\rho_s}(x - u_s) \leqslant \alpha \end{aligned}$$

and so $(u_s, v_s, \alpha) \in \operatorname{epi} f$. Finally we note that [8, Proposition 1.3.5] we have

$$\inf_{(u_s,v_s)\in K_s} f(u_s, v_s) \geqslant \limsup_k \inf_{(u_s,v_s)\in K_s} f_k(u_s, v_s)$$

and so once can certainly obtain a minimising sequence $\left\{\left(u_s^k, v_s^k\right)\right\}$ with

$$\left| f_k\left(u_s^k, v_s^k\right) - \inf_{(u_s,v_s)\in K_s} f(u_s, v_s) \right| \to 0.$$

Convergence of a suitable such sequence follows from the compactness of the set K_s. Thus we can use Proposition 7.26 to claim that when $\left(u_s^{k+1}, v_s^{k+1}\right) \to (u_s^{+1}, v_s^{+1})$ we have $(u_s^{+1}, v_s^{+1}) \in \arg\min_{(u_s,v_s)\in K_s} f(u_s, v_s)$ establishing the first part of our claim as we take $(y_s)^{+1} := v_s^{+1}$. Similarly we can consider the associated optimisation problem where $u_s^{k+1} \to (u_s)^{+1}$ for all $s \in S$ and the associated solutions to (7.27). We apply the same arguments to the associated coercive objective $g\left(\cdot, u^{k+1}\right) \to g(\cdot, u^{+1})$ where convergence is uniform on bounded sets. This convergence is well known to imply epi-convergence which gives us 1 and 2 of the Proposition 7.26 assumptions (see [99, Theorem 7.11]). The only deviation from the previous arguments is the existence of a convergent minimising sequence follows from the equi-coersivity of $x \mapsto g\left(x, u^{k+1}\right)$. Indeed as $g\left(\cdot, u^{k+1}\right) \to g(\cdot, u^{+1})$ in epi-convergent sense its level sets converge (see [99, Proposition 7.7]). As the level set of $x \mapsto g(x, u^{+1})$ is bounded we may contain any minimising sequence within a fixed bounded set and hence a convergent subsequences can again be extracted. ■

Proposition 7.28 demonstrates that Algorithm 7.3 converges to a partial minimum of φ_ρ; this is not immediate from the general properties of Gauss-Seidel methods since the x update is approximated with a non-ascent step based on the consensus problem.

Proposition 7.28 *Suppose that φ_ρ is as defined in* (7.3) *and that it is derived from a SIP* (7.1) *with relatively complete recourse. Suppose that ψ_s^ρ satisfies the assumptions of Lemma 7.23 and that the sequence $\left(x^k, y^k\right)$ generated by Algorithm 7.3 converges to (x^*, y^*). Then we have for $\rho > 0$ sufficiently large that*

$$\begin{aligned} \varphi_\rho(x^*, y) &\geqslant \varphi_\rho(x^*, y^*) \quad \textit{for all } y \textit{ and} \\ \varphi_\rho(x, y^*) &\geqslant \varphi_\rho(x^*, y^*) \quad \textit{for all } x. \end{aligned}$$

Proof. Clearly for every k we have

$$\varphi_\rho\left(x^k, y^k\right) \geqslant \varphi_\rho\left(x^k, y^{k+1}\right) \geqslant \varphi_\rho\left(x^{k+1}, y^{k+1}\right)$$

and as φ_ρ is Lipschitz continuous and bounded below when $\left(x^k, y^k\right) \to (x^*, y^*)$ we have $\varphi_\rho\left(x^k, y^k\right) \downarrow \varphi_\rho(x^*, y^*)$ so that $\varphi_\rho\left(x^k, y^k\right) \geqslant \varphi_\rho(x^*, y^*)$ for all k. Moreover for each k we also have

$$\varphi_\rho\left(x^k, y\right) \geqslant \varphi_\rho\left(x^k, y^k\right) \quad \text{for all } y$$

and so letting $k \to \infty$ we obtain $\varphi_\rho(x^*, y) \geqslant \varphi_\rho(x^*, y^*)$ for all y. It remains to be shown that $\varphi_\rho(x, y^*) \geqslant \varphi_\rho(x^*, y^*)$ for all x. Suppose to the contrary that x^* is not the minimiser of $x \mapsto \varphi_\rho(x, y^*)$.

By our earlier observations, made just prior to identity (7.5), we find that as $y^* \in \arg\min \varphi_\rho(x^*, \cdot)$ implies $y_s^* \equiv (y_s^*)^{+1} = v_s^*$ for all $s \in S$ and (u_s^*, v_s^*) which solves

$$\min_{(u_s, v_s) \in K_s} c^\top u_s + d_s^\top v_s + \psi_s^{\rho_s}(x^* - u_s).$$

This is the first iteration of our GS as applied to the initial point (x^*, y^*) to obtain $\left(x^*, (y^*)^{+1}\right) = (x^*, y^*)$. We may now apply the second part of the GS iteration under the assumption that x^* is not the minimiser of $x \mapsto \varphi_\rho(x, y^*)$. Lemma 7.23 implies for any

$$(x^*)^{+1} \in \arg\min_x \sum_{s \in S} p_s \psi_s^{\rho_s}(x - u_s^*)$$

that

$$\varphi_\rho(x^*, y^*) = \varphi_\rho\left(x^*, (y^*)^{+1}\right) > \varphi_\rho\left((x^*)^{+1}, (y^*)^{+1}\right).$$

Let

$$2\gamma = \varphi_\rho(x^*, y^*) - \varphi_\rho\left((x^*)^{+1}, (y^*)^{+1}\right)$$

and hence for k sufficiently large (i.e. $k \geqslant \bar{k}$) we must have by continuity of φ_ρ and of closed graph of $(x, y) \to \left((x)^{+1}, (y)^{+1}\right)$ proved in Proposition 7.27 and the fact that we take $y^{k+1} = (y^k)^{+1}$ and $x^{k+1} = (x^k)^{+1}$ we have

$$-\gamma \geqslant \varphi_\rho\left(x^{k+1}, y^{k+1}\right) - \varphi_\rho\left(x^k, y^k\right)$$

implying for any K that

$$-\gamma K \geqslant \varphi_\rho\left(x^{K+\bar{k}}, y^{K+\bar{k}}\right) - \varphi_\rho\left(x^{\bar{k}}, y^{\bar{k}}\right).$$

Letting $K \to \infty$ we get $\varphi_\rho(x^*, y^*) = -\infty$ a contradiction. Hence x^* minimises $x \mapsto \varphi_\rho(x, y^*)$. ■

Theorem 7.29 is the culmination of the results in this section, and demonstrates that Algorithm 7.3 always converges to a feasible solution when the first stage of the SIP consists of purely integer variables.

Theorem 7.29 *Assume that $\psi_s^{\rho_s}$, for $s \in S$, satisfy the assumption to be an ICRF$^+$ (see Assumption 7.3). Suppose that φ_ρ is as defined in* (7.3) *and that it is derived from a SIP* (7.1) *with relatively complete recourse. Suppose that $\psi_s^{\rho_s} \in C^2(\mathbb{R}^{n+m})$ is a finite, coercive, proper convex function for all $s \in S$ with ψ_s for $s \in S$ are globally Lipschitz and that the sequence $\left(x^k, y^k\right)$ generated by Algorithm 7.3 converges to (x^*, y^*). Then (x^*, y^*) is a stationary point of φ_ρ. When we have a SIP with pure integer first stage variables then for ρ sufficiently large this implies $(x^*, y^*) \in F$ i.e. is a feasible solution.*

Proof. Note that by Proposition 7.28 we have $\varphi_\rho(x, y^*) \geqslant \varphi_\rho(x^*, y^*)$ for all x and so $0 \in \partial_x \varphi_\rho(\cdot, y^*)(x^*)$. Thus by (7.24) and Proposition 7.25 we have $\nabla_x \varphi_\rho(\cdot, y^*)(x^*) = 0$ existing. Furthermore by Proposition 7.28 we have $\varphi_\rho(x^*, y) \geqslant \varphi_\rho(x^*, y^*)$ for all y and so $0 \in \partial_y \varphi_\rho(x^*, \cdot)(y^*)$. Hence by Lemma 7.19 we have $(0, 0) \in \partial \varphi_\rho(x^*, y^*)$ and (x^*, y^*) is a stationary point and by Proposition 7.25 we have a local minimum. It follows from Lemma 7.14 that $(x^*, y^*) \in F$ is a feasible solution. ■

7.3 Conclusions

Theorem 7.29 demonstrates that the modified Gauss-Seidel algorithm (Algorithm 7.3) is guaranteed to converge to a feasible solution of the original SIP, as long as the first-stage decision variables are purely integer. The results in this chapter can be viewed as providing partial theoretical support for both Progressive Hedging ([112] and Chapter 4) and PBGS (Chapter 6) as heuristics for solving SIPs. It does differ from the implementations used in the literature in that no multipliers are used and there is consequently no multiplier update. Indeed multiplier updates can be viewed as a kind of penalty strategy, which this analysis provides no insight into.

The most obvious direction for future investigation based on the results in this chapter is to implement the Gauss-Seidel based algorithm developed in this chapter and compare its performance with algorithms present in the literature, as well as other SIP algorithms such as Progressive Hedging.

The algorithm and results of this chapter could also be extended to multi-stage SIP problems in which all decision variables, except potentially the final stage variables, are pure integer.

Chapter 8

Conclusions and Future Work

This thesis has investigated algorithms and theory relating to Stochastic Mixed-Integer Programming, in particular approaches based on scenario-wise separation. Each of these algorithms focuses on calculating either high-quality dual bounds or feasible solutions for SIPs, rather than finding the optimal solution directly. The algorithms presented herein for calculating dual bounds significantly improve on all previously known approaches, while the heuristics for finding feasible solutions are at worst competitive, depending on the choice of algorithm parameters. A large proportion of the computational work required for each of these algorithms can be performed in parallel, which means that they can be reasonably applied to large-scale SIP instances.

The FW-PH and SDM-GS-ALM algorithms discussed in Chapters 4 and 5 obtain dual bounds for a wide variety of SIPs more quickly and effectively than previously known methods. The FW-PH algorithm obtains dual bounds by applying Progressive Hedging to the convex-hull relaxation of SIPs, using a simplified form of the simplicial decomposition method to solve the primal update subproblems. This approach is validated by the theoretical results in Section 3.2, which guarantee that SDM will eventually generate a sufficiently accurate approximation of the convex hull. By comparison, the SDM-GS-ALM algorithm utilises a more 'feature-complete' version of SDM, and generalises the overall algorithm beyond SIP problems to MIPs of a particular structure (which includes SIPs).

As compared to FW-PH, the SDM-GS-ALM algorithm is considerably more sophisticated, particularly with respect to the implementation of the simplicial decomposition component. Despite this distinction, the computational results which we have obtained with SDM-GS-ALM are not substantially better in terms of bound quality when compared with the FW-PH results. Since

SDM-GS-ALM has more 'moving parts' and hence more parameters which must be set correctly to obtain ideal performance (a difficult task), this outcome is not entirely unexpected. In any case, the theoretical results given in the development of SDM-GS-ALM yield valuable insights into the behaviour of this entire class of algorithms.

In terms of future research, possible improvements to the FW-PH and SDM-GS-ALM algorithms are discussed in the conclusions of their respective chapters. Further tuning of the heuristics for selecting algorithm parameters (particularly the initial value and update scheme of the penalty parameters) would lead to better practical results. A greater understanding of the behaviour of FW-PH and SDM-GS-ALM when applied to infeasible problems would strengthen their respective theoretical bases.

The most obvious application of these algorithms is to calculate bounds within a branch-and-bound type method to find primal SIP solutions. The Dual Decomposition method (discussed in Section 2.5.2) is a branch-and-bound type method which has previously been modified to calculate dual bounds using Progressive Hedging. Since FW-PH and SDM-GS-ALM calculate high-quality dual bounds more reliably and (generally) quickly compared to PH, it seems reasonably likely that a Dual Decomposition-type method which incorporates FW-PH or SDM-GS-ALM in the place of PH will outperform previous versions of the Dual Decomposition algorithm.

A separate direction for future investigation would be to formally explore the use of FW-PH or SDM-GS-ALM as a primal solution heuristic, since both of these algorithms generate primal feasible solutions as vertices in their respective SDM components.

The PBGS algorithm discussed in Chapter 6 uses augmented Lagrangian duality theory based on a sharp penalty function to eliminate the need for dual multipliers. A block Gauss-Seidel method is then applied as a heuristic to find (hopefully high-quality) feasible solutions. In practice PBGS invariably converged to a feasible solution, although this is not guaranteed by currently known theory. The variation of PBGS developed in Chapter 7 has stronger theoretical properties in that it is guaranteed to converge to a feasible solution if the first-stage variables of the SIP are pure integer.

In computational experiments PBGS tended to generate good feasible solutions more quickly than Progressive Hedging when applied to SIPs. However, the quality of the solutions generated by PBGS compared to PH was frequently worse. Since the performance of PBGS is very strongly

dependent on the choice of parameters, further investigation into the ideal parameter configuration for various problem classes is a potentially fruitful area for future research. An alternative potential direction for improvement is to explore the use of scenario clustering to escape the suboptimal local minimum. Verifying the speed-up benefits of solving the PBGS subproblems in parallel would also be valuable.

All of the above mentioned algorithms have the potential to be applied to multi-stage SIPs. The underlying theory of each algorithm is easily applicable to multi-stage problems due to their scenario-splitting structure. If and when this is pursued in the future, the most difficult challenges are likely to be achieving clear and concise notation and implementing the algorithms in an efficient and effective manner, since multi-stage problems have a much more complicated structure.

A general trend throughout the theoretical and experimental results presented herein is that two-stage SIPs with pure-integer first stage variables tend to be 'easier' than those with mixed-integer first stage. The FW-PH and SDM-GS-ALM algorithms of Chapters 4 and 5, when applied in practice, tended to discover the exact optimal dual bound of the CAP and SSLP problems (which have integer first-stage) relatively quickly, whereas in the case of the DCAP problems (which have mixed-integer first stage) asymptotic convergence was typically observed. Similarly, the feasible solutions found by PBGS for the DCAP problems were generally of lower quality than those found for the CAP and SSLP problems. The theoretical convergence proof for the modified version of PBGS given in Chapter 7 only applies at all when the first stage is pure integer. A potential cause for this distinction between problem classes is that penalty function based approaches which utilise smooth penalty functions are more effective when the variables they apply to are discrete, since non-feasible solutions are always meaningfully penalised.

The use of non-smooth penalty functions offers some hope of tackling mixed-integer first stage SIPs with more success, since a 'sharp' penalty function has more latitude to impose a larger penalty on a 'nearly feasible' solution while maintaining a reasonably-sized penalty parameter. Fortunately the Frank-Wolfe-based component of FWPH and SDM-GS-ALM is amenable to non-smooth optimisation (as demonstrated in Section 3.3). However, these non-smooth functions are still far more difficult to deal with computationally; the alternating-update approaches like ADMM and Gauss-Seidel, which are otherwise ideal for Stochastic Integer Programming, generally do not handle non-smooth objective functions gracefully (recall that the PBGS method of Chapter 6 is

merely a heuristic even though the theory guarantees strong duality for a sufficiently large penalty parameter). The discovery of an algorithm or heuristic capable of dealing more effectively with these non-smooth penalty functions, and hence (potentially) general mixed-integer SIPs as well, would have a substantial impact not only on the field of Stochastic Integer Programming, but also Mixed-Integer Programming in general.

Bibliography

[1] *COmputational INfrastructure for Operations Research*. Last accessed 28 January, 2016.

[2] S. Ahmed, *A scenario decomposition algorithm for 0–1 stochastic programs*, Operations Research Letters, 41 (2013), pp. 565–569.

[3] S. Ahmed, R. Garcia, N. Kong, L. Ntaimo, G. Parija, F. Qiu, and S. Sen, *SIPLIB: A stochastic integer programming test problem library*, (2015).

[4] S. Ahmed, M. Tawarmalani, and N. V. Sahinidis, *A finite branch-and-bound algorithm for two-stage stochastic integer programs*, Math. Program., 100 (2004), pp. 355–377.

[5] H. B. Amor, J. Desrosiers, and A. Frangioni, *On the choice of explicit stabilizing terms in column generation*, Discrete Applied Mathematics, 157 (2009), pp. 1167 – 1184.

[6] L. Armijo, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific J. Math., 16 (1966), pp. 1–3.

[7] H. Attouch, *Variational convergence for functions and operators*, Applicable Mathematics Series, Pitman (Advanced Publishing Program), Boston, MA, 1984.

[8] G. Beer, *Topologies on closed and closed convex sets*, vol. 268 of Mathematics and its Applications, Kluwer Academic Publishers Group, Dordrecht, 1993.

[9] C. Beltran, C. Tadonki, and J.-P. Vial, *Solving the p-median problem with a semi-Lagrangian relaxation*, Comput. Optim. Appl., 35 (2006), pp. 239–260.

[10] C. Beltran-Royo, J.-P. Vial, and A. Alonso-Ayuso, *Semi-Lagrangian relaxation applied to the uncapacitated facility location problem*, Comput. Optim. Appl., 51 (2012), pp. 387–409.

[11] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 1982.

[12] ——, *Nonlinear Programming*, Athena Scientific, 1999.

[13] ——, *Convex Optimization Algorithms*, Athena Scientific, 2015.

[14] ——, *Incremental aggregated proximal and augmented Lagrangian algorithms.* arXiv preprint arXiv:1509.09257, 2015.

[15] M. Bodur, S. Dash, O. Günlük, and J. Luedtke, *Strengthened Benders cuts for stochastic integer programs with continuous recourse*, (2014). Last accessed on 13 January 2015.

[16] N. Boland, J. Christiansen, B. Dandurand, A. Eberhard, J. Linderoth, J. Luedtke, and F. Oliveira, *Combining Progressive Hedging with a Frank–Wolfe Method to Compute Lagrangian Dual Bounds in Stochastic Mixed-Integer Programming*, SIAM J. Optim., 28 (2018), pp. 1312–1336.

[17] N. Boland, J. Christiansen, B. Dandurand, A. Eberhard, and F. Oliveira, *A parallelizable augmented lagrangian method applied to large-scale non-convex-constrained optimization problems*, Mathematical Programming, (2018), pp. 1–34. Article in Press.

[18] N. L. Boland and A. C. Eberhard, *On the augmented Lagrangian dual for integer programming*, Math. Program., 150 (2015), pp. 491–509.

[19] N. L. Boland, A. C. Eberhard, F. Engineer, and A. Tsoukalas, *A new approach to the feasibility pump in mixed integer programming*, SIAM J. Optim., 22 (2012), pp. 831–861.

[20] S. Bonettini, *Inexact block coordinate descent methods with application to non-negative matrix factorization*, IMA Journal of Numerical Analysis, 31 (2011), pp. 1431–1452.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundation and Trends in Machine Learning, 3 (2011), pp. 1–122.

[22] C. C. Carøe and R. Schultz, *Dual decomposition in stochastic integer programming*, Oper. Res. Lett., 24 (1999), pp. 37–45.

[23] N. Chatzipanagiotis, D. Dentcheva, and M. Zavlanos, *An augmented Lagrangian method for distributed optimization*, Mathematical Programming, 152 (2014), pp. 405–434.

[24] G. Chen and M. Teboulle, *A proximal-based decomposition method for convex minimization problems*, Mathematical Programming, 64 (1994), pp. 81–101.

[25] E. W. Cheney and A. A. Goldstein, *Newton's method for convex programming and Tchebycheff approximation.*, Numer. Math., 1 (1959), pp. 253–268.

[26] F. Clarke, *Optimization and Nonsmooth Analysis*, Society for Industrial and Applied Mathematics, 1990.

[27] T. G. Crainic, M. Hewitt, and W. Rei, *Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design*, Computers & Operations Research, 43 (2014), pp. 90–99.

[28] C. Davis, *Theory of positive linear dependence*, American Journal of Mathematics, 76 (1954), pp. 733–746.

[29] W. de Oliveira and C. Sagastizábal, *Bundle Methods in the XXIst Century: A Bird's-eye view*, Pesquisa Operacional, 34 (2014), pp. 647–670.

[30] W. de Oliveira, C. Sagastizábal, and C. Lemaréchal, *Convex proximal bundle methods in depth: a unified analysis for inexact oracles*, Mathematical Programming, 148 (2014), pp. 241–277.

[31] J. Eckstein, *A practical general approximation criterion for methods of multipliers based on Bregman distances*, Mathematical Programming, 96, pp. 61–86.

[32] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Programming, 55 (1992), pp. 293–318.

[33] J. Eckstein and P. Silva, *A practical relative error criterion for augmented lagrangians*, Mathematical Programming, 141 (2013), pp. 319–348.

[34] J. Eckstein and W. Yao, *Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives*, tech. rep., Rutgers University, 2014.

[35] L. F. Escudero, M. Araceli Garín, G. Pérez, and A. Unzueta, *Scenario cluster decomposition of the Lagrangian dual in two-stage stochastic mixed 0-1 optimization*, Comput. Oper. Res., 40 (2013), pp. 362–377.

[36] L. F. Escudero, M. A. Garín, M. Merino, and G. Pérez, *An algorithmic framework for solving large-scale multistage stochastic mixed 0-1 problems with nonsymmetric scenario trees*, Comput. Oper. Res., 39 (2012), pp. 1133–1144.

[37] M. J. Feizollahi, S. Ahmed, and A. Sun, *Exact augmented lagrangian duality for mixed integer linear programming*, Mathematical Programming, 161 (2017), pp. 365–387.

[38] M. J. Feizollahi, M. Costley, S. Ahmed, and S. Grijalva, *Large-scale decentralized unit commitment*, International Journal of Electrical Power & Energy Systems, 73 (2015), pp. 97–106.

[39] F. Fischer and C. Helmberg, *A parallel bundle framework for asynchronous subspace optimization of nonsmooth convex functions*, SIAM Journal on Optimization, 24 (2014), pp. 795–822.

[40] A. Frangioni, *Generalized bundle methods*, SIAM Journal on Optimization, 13 (2003), pp. 117–156.

[41] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval research logistics quarterly, 3 (1956), pp. 95–110.

[42] R. M. Freund and P. Grigas, *New analysis and results for the Frank-Wolfe method*, Math. Program., 155 (2016), pp. 199–230.

[43] D. Gabay, *Chapter IX Applications of the Method of Multipliers to Variational Inequalities*, vol. 15 of Studies in Mathematics and its Applications, 1983.

[44] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Computers and Mathematics with Applications, 2 (1976), pp. 17–40.

[45] D. Gade, G. Hackebeil, S. M. Ryan, J. . Watson, R. J. . Wets, and D. L. Woodruff, *Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs*, Mathematical Programming, (2016), pp. 1–21. Article in Press.

[46] A. M. Geoffrion, *Lagrangean relaxation for integer programming*, Math. Programming Stud., (1974), pp. 82–114. Approaches to integer programming.

[47] E. Gertz and S. Wright, *Object-oriented software for quadratic programming*, ACM Transactions on Mathematical Software, 29 (2003), pp. 58–81.

[48] R. Glowinski and A. Marrocco, *Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualité, d'une classe de problems de dirichlet non lineares*, Revue Française d'Automatique, Informatique, et Recherche Opérationelle, 9 (1975), pp. 41–76.

[49] L. Grippo and M. Sciandrone, *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints*, Operations Research Letters, 26 (2000), pp. 127–136.

[50] G. Guo, G. Hackebeil, S. M. Ryan, J.-P. Watson, and D. L. Woodruff, *Integration of progressive hedging and dual decomposition in stochastic integer programs*, Operations Research Letters, 43 (2015), pp. 311–316.

[51] A. Hamdi, P. Mahey, and J. P. Dussault, *Recent Advances in Optimization: Proceedings of the 8th French-German Conference on Optimization Trier, July 21–26, 1996*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1997, ch. A New Decomposition Method in Nonconvex Programming via a Separable Augmented Lagrangian, pp. 90–104.

[52] S. C. S. M. Hare, W., *A proximal bundle method for nonsmooth nonconvex functions with inexact information*, Computational Optimization and Applications, 63 (2016), pp. 1–28.

[53] R. J. Hathaway and J. C. Bezdek, *Grouped coordinate minimization using Newton's method for inexact minimization in one vector coordinate*, Journal of Optimization Theory and Applications, 71 (1991), pp. 503–516.

[54] K. K. Haugen, A. Løkketangen, and D. L. Woodruff, *Progressive hedging as a meta-heuristic applied to stochastic lot-sizing*, European J. Oper. Res., 132 (2001), pp. 116–122.

[55] B. He, L.-Z. Liao, D. Han, and H. Yang, *A new inexact alternating directions method for monotone variational inequalities*, Mathematical Programming, 92 (2002), pp. 103–118.

[56] D. W. Hearn, S. Lawphongpanich, and J. A. Ventura, *Restricted simplicial decomposition: computation and extensions*, Math. Programming Stud., (1987), pp. 99–118.

[57] T. Helgason and S. W. Wallace, *Approximate scenario solutions in the progressive hedging algorithm. A numerical study with an application to fisheries management*, Ann. Oper. Res., 31 (1991), pp. 425–444. Stochastic programming, Part II (Ann Arbor, MI, 1989).

[58] M. R. Hestenes, *Multiplier and gradient methods*, Journal of Optimization Theory and Applications, (1969), pp. 303–320.

[59] C. Hildreth, *A quadratic programming procedure*, Naval Research Logistics Quarterly, 4 (1957), pp. 79–85, 361.

[60] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms. II. , Advanced theory and bundle methods*, Grundlehren der mathematischen Wissenschaften, Springer-Verlag, Berlin, New-york, 1996.

[61] C. A. Holloway, *An extension of the Frank and Wolfe method of feasible directions*, Math. Programming, 6 (1974), pp. 14–27.

[62] IBM Corporation, *IBM ILOG CPLEX Optimization Studio CPLEX User's Manual*. Last accessed 22 August 2016.

[63] ——, *IBM ILOG CPLEX V12.5*. Last accessed 28 Jan 2016.

[64] K. Jörnsten and A. Klose, *An improved lagrangian relaxation and dual ascent approach to facility location problems*, Computational Management Science, (2015). Article in Press.

[65] P. Kall and S. W. Wallace, *Stochastic Programming*, John Wiley & Sons, 1994.

[66] J. E. Kelley, Jr., *The cutting-plane method for solving convex programs*, J. Soc. Indust. Appl. Math., 8 (1960), pp. 703–712.

[67] K. Kiwiel, C. Rosa, and A. Ruszczyński, *Proximal decomposition via alternating linearization*, SIAM Journal on Optimization, 9 (1999), pp. 668–689.

[68] K. C. Kiwiel, *Proximity control in bundle methods for convex nondifferentiable minimization*, Math. Programming, 46 (1990), pp. 105–122.

[69] K. C. Kiwiel, *Approximations in proximal bundle methods and decomposition of convex programs*, Journal of Optimization Theory and Applications, 84 (1995), pp. 529–548.

[70] S. Kontogiorgis and R. R. Meyer, *A variable-penalty alternating directions method for convex optimization*, Math. Programming, 83 (1998), pp. 29–53.

[71] G. Laporte and F. V. Louveaux, *The integer L-shaped method for stochastic integer programs with complete recourse*, Oper. Res. Lett., 13 (1993), pp. 133–142.

[72] T. Larsson, M. Patriksson, and A. Strömberg, *Conditional subgradient optimization - theory and applications*, European Journal of Operational Research, 88 (1996), pp. 382–403. Cited By :52.

[73] C. Lemaréchal and C. Sagastizábal, *Variable metric bundle methods: From conceptual to implementable forms*, Mathematical Programming, Series B, 76 (1997), pp. 393–410.

[74] C. Lemaréchal, *An extension of Davidon methods to non differentiable problems*, Springer Berlin Heidelberg, 1975, pp. 95–109.

[75] A. Lenoir and P. Mahey, *Global and adaptive scaling in a separable augmented lagrangian algorithm*, Optimization Online, (2007).

[76] X. Lin, M. Pham, and A. Ruszczyński, *Alternating linearization for structured regularization problems*, Journal of Machine Learning Research, 15 (2014), pp. 3447–3481.

[77] O. Listes and R. Dekker, *A scenario aggregation-based approach for determining a robust airline fleet composition for dynamic capacity allocation*, Transportation Science, 39 (2005), pp. 367–382.

[78] A. Løkketangen and D. L. Woodruff, *Progressive hedging and tabu search applied to mixed integer (0,1) multistage stochastic programming*, Journal of Heuristics, 2 (1996), pp. 111–128.

[79] M. Lubin, K. Martin, C. Petra, and B. Sandikçi, *On parallelizing dual decomposition in stochastic integer programming*, Operations Research Letters, 41 (2013), pp. 252–258.

[80] M. Lubin, C. Petra, M. Anitescu, and V. Zavala, *Scalable stochastic optimization of complex energy systems*, in Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, ACM, Seattle, WA, 2011, p. 64:1–64:10.

[81] G. Lulli and S. Sen, *A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems*, Management Science, 50 (2004), pp. 786–796.

[82] P. Mahey, S. Oualibouch, and P. D. Tao, *Proximal decomposition on the graph of a maximal monotone operator*, SIAM Journal on Optimization, 5 (1995), pp. 454–466.

[83] The MathWorks, Natick, *MATLAB 2012b*, 2014.

[84] J. Máyer, *Stochastic linear programming algorithms: A comparison based on a model management system*, Gordon and Breach Science Publishers, 1998.

[85] E. Monabbati, *An application of a Lagrangian-type relaxation for the uncapacitated facility location problem*, Jpn. J. Ind. Appl. Math., 31 (2014), pp. 483–499.

[86] J. M. Mulvey and A. Ruszczyński, *A diagonal quadratic approximation method for large-scale linear programs*, Oper. Res. Lett., 12 (1992), pp. 205–215.

[87] ——, *A diagonal quadratic approximation method for linear multistage stochastic programming problems*, in System modelling and optimization (Zurich, 1991), vol. 180 of Lect. Notes Control Inf. Sci., Springer, Berlin, 1992, pp. 588–597.

[88] J. M. Mulvey and H. Vladimirou, *Applying the progressive hedging algorithm to stochastic generalized networks*, Ann. Oper. Res., 31 (1991), pp. 399–424. Stochastic programming, Part II (Ann Arbor, MI, 1989).

[89] National Computing Infrastructure (NCI), *NCI Website*. Last accessed 19 November 2016.

[90] G. Nemhauser and L. Wolsey, *Integer and combinatorial optimization*, Wiley-Interscience series in discrete mathematics and optimization, Wiley, 1988.

[91] L. Ntaimo, *Decomposition Algorithms for Stochastic Combinatorial Optimization: Computational Experiments and Extensions*, PhD thesis, 2004.

[92] L. Ntaimo and S. Sen, *The million-variable "march" for stochastic combinatorial optimization*, J. Global Optim., 32 (2005), pp. 385–400.

[93] F. Oliveira, J. Christiansen, B. Dandurand, and A. Eberhard, *Combining penalty-based and gauss-seidel methods for solving stochastic mixed-integer problems*, International Transactions in Operational Research, (2018). Article in Press.

[94] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., New York: Academic Press, 1969.

[95] R. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877–898.

[96] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Mathematics of Operations Research, 1 (1976), pp. 97–116.

[97] R. T. Rockafellar, *Convex analysis*, Princeton Landmarks in Mathematics, Princeton University Press, Princeton, NJ, 1997. Reprint of the 1970 original, Princeton Paperbacks.

[98] R. T. Rockafellar and R. J.-B. Wets, *Scenarios and policy aggregation in optimization under uncertainty*, Math. Oper. Res., 16 (1991), pp. 119–147.

[99] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, vol. 317 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer-Verlag, Berlin, 1998.

[100] A. Ruszczyński, *Nonlinear Optimization*, Princeton University Press, 2006.

[101] A. Ruszczyński, *On convergence of an augmented Lagrangian decomposition method for sparse convex optimization*, Mathematics of Operations Research, 20 (1995), pp. 634–656.

[102] A. P. Ruszczyński, *Nonlinear optimization*, Princeton, N.J. Princeton University Press, 2006.

[103] N. Shor, *Minimization Methods for Non-differentiable Functions*, Springer-Verlag, New York, 1985.

[104] M. Sion, *On general minimax theorems*, Pacific J. Math., 8 (1958), pp. 171–176.

[105] J. E. Spingarn, *Submonotone subdifferentials of Lipschitz functions*, Trans. Amer. Math. Soc., 264 (1981), pp. 77–89.

[106] M. Studniarski, *Necessary and sufficient conditions for isolated local minima of nonsmooth functions*, SIAM J. Control Optim., 24 (1986), pp. 1044–1049.

[107] R. Tappenden, P. Richtárik, and B. Büke, *Separable approximations and decomposition methods for the augmented Lagrangian*, Optimization Methods & Software, 30 (2015), pp. 643–668.

[108] P. Tseng, *Convergence of a block coordinate descent method for nondifferentiable minimization*, Journal of Optimization Theory and Applications, 109 (2001), pp. 475–494.

[109] J. A. Ventura and D. W. Hearn, *Restricted simplicial decomposition for convex constrained problems*, Math. Programming, 59 (1993), pp. 71–85.

[110] B. von Hohenbalken, *Simplicial decomposition in nonlinear programming algorithms*, Math. Programming, 13 (1977), pp. 49–68.

[111] J. Warga, *Minimizing certain convex functions*, SIAM Journal on Applied Mathematics, 11 (1963), pp. 588–593.

[112] J.-P. Watson and D. L. Woodruff, *Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems*, Computational Management Science, 8 (2011), pp. 355–370.

[113] R. Wenczel and A. Eberhard, *Slice convergence of parametrised sums of convex functions in non-reflexive spaces*, Bull. Austral. Math. Soc., 60 (1999), pp. 429–458.

[114] G. L. Zenarosa, O. A. Prokopyev, and A. J. Schaefer, *M-smiplib: A multi-stage stochastic mixed-integer programming test set library.* http://www.cs.cmu.edu/ gzen/m-smiplib/, 2014.

[115] ——, *Scenario-tree decomposition: Bounds for multistage stochastic mixed-integer programs*, Optimization Online, (2014).