Les Cahiers du GERAD ISSN: 0711–2440

Influence of the normalization constraint on the integral simplex using decomposition

S. Rosat, I. Elhallaoui,
F. Soumis, D. Chakour

G–2014–99

November 2014

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Influence of the normalization constraint on the integral simplex using decomposition

Samuel Rosat[a,c]
Issmail Elhallaoui[a,c]
François Soumis[a,c]
Driss Chakour[b,c]

[a] *Polytechnique Montréal, Montréal (Québec) Canada, H3C 3A7*

[b] *École Polytechnique, 91128 Palaiseau, France*

[c] *GERAD, HEC Montréal, Montréal (Québec) Canada, H3T 2A7*

samuel.rosat@gerad.ca
issmail.elhallaoui@gerad.ca
francois.soumis@gerad.ca
driss.chakour@polytechnique.edu

November 2014

Les Cahiers du GERAD
G–2014–99

Abstract: Since its introduction in 1969, the set partitioning problem has received much attention, and the structure of its feasible domain has been studied in detail. In particular, there exists a decreasing sequence of integer feasible points that leads to the optimum, such that each solution is a vertex of the polytope of the linear relaxation and adjacent to the previous one. Several algorithms are based on this observation and aim to determine that sequence; one example is the integral simplex using decomposition (ISUD) of Zaghrouti et al. (2014). In ISUD, the next solution is often obtained by solving a linear program without using any branching strategy. We study the influence of the normalization-weight vector of this linear program on the integrality of the next solution. We extend and strengthen the decomposition theory in ISUD, prove theoretical properties of the generic and specific convexity constraints, and propose new normalization constraints that encourage integral solutions. Numerical tests on scheduling instances (with up to 500,000 variables) demonstrate the potential of our approach.

Key Words: 0–1 programming, set partitioning problem, primal algorithms, convexity, augmenting algorithms.

Acknowledgments: This work was supported by a RD COOP research grant from CRSNG and Kronos Inc.

1 Introduction

Consider the set partitioning problem (SPP)

$$z^\star_{SPP} = \min_{\boldsymbol{x}} \left\{ \boldsymbol{c}^T \boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{e}\,,\, \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{e}\,,\, \boldsymbol{x} \text{ is integer} \right\} \tag{SPP}$$

where $\boldsymbol{A} \in \{0,1\}^{m\times n}$ is an $m \times n$ binary matrix, and $\boldsymbol{c} \in \mathbb{N}^n$ is the cost vector. The vector of all zeros (resp. ones) with dimension dictated by the context is denoted $\boldsymbol{0}$ (resp. $\boldsymbol{e}$). Without loss of generality, we assume that $\boldsymbol{A}$ is full rank, contains no zero rows or columns, and has no identical rows or columns. $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{e}$ are called the linear constraints and $\boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{e}$ the bound constraints. $\mathcal{F}_{\text{SPP}}$ denotes the set of all feasible solutions of SPP; $z^\star_{SPP}$ is called the optimal value (or optimum) of SPP; and any feasible solution $\boldsymbol{x}^\star \in \mathcal{F}_{\text{SPP}}$ such that $\boldsymbol{c}^T\boldsymbol{x}^\star = z^\star_{SPP}$ is called an optimal solution of SPP. Finally, given the bounds ($\boldsymbol{0}$ and $\boldsymbol{e}$) and the integrality constraints, $\mathcal{F}_{\text{SPP}}$ contains only 0–1 vectors, i.e., $\mathcal{F}_{\text{SPP}} \subseteq \{0,1\}^n$.

The linear relaxation of SPP, denoted SPP^{RL}, is the linear program obtained by removing the integrality constraints of SPP. Its feasible domain and optimal value are respectively denoted $\mathcal{F}_{\text{SPP}^{RL}}$ and $z^\star_{SPP^{RL}}$.

Lower-case bold symbols are used for column vectors, and upper-case bold symbols denote matrices. For subsets $\mathcal{I} \subseteq \{1,\dots m\}$ of the row indices and $\mathcal{J} \subseteq \{1,\dots n\}$ of the column indices, the submatrix of $\boldsymbol{A}$ with rows indexed by $\mathcal{I}$ and columns indexed by $\mathcal{J}$ is denoted $\boldsymbol{A}_{\mathcal{I}\mathcal{J}}$. Similarly, $\boldsymbol{A}_{\mathcal{I}\boldsymbol{\cdot}}$ is the set of rows of $\boldsymbol{A}$ indexed by $\mathcal{I}$, $\boldsymbol{A}_{\boldsymbol{\cdot}\mathcal{J}}$ is the set of columns of $\boldsymbol{A}$ indexed by $\mathcal{J}$, and for any vector $\boldsymbol{v} \in \mathbb{R}^n$, $\boldsymbol{v}_{\mathcal{J}}$ is the subvector of all v_j, $j \in \mathcal{J}$. Finally, given any matrix $\boldsymbol{M}$, $\boldsymbol{M}^T$ is its transpose, and $\mathit{Span}(\boldsymbol{M})$ is the linear span of all its columns, also called the image of $\boldsymbol{M}$.

Note that all optimization programs will be written in their minimization form, but the ideas and results also apply to maximization scenarios.

1.1 Integral simplex methods for the set partitioning problem

Since its introduction by Garfinkel and Nemhauser in 1969 [1], the SPP has received much attention because of its wide range of applications: vehicle and crew scheduling, clustering problems, etc. It often appears within column-generation frameworks for such problems.

Many algorithms have been developed. As is the case for generic integer linear programs, they can be classified into three main classes [2]: *dual fractional*, *dual integral*, and *primal* (or *augmentation*) methods. *Dual fractional* algorithms maintain optimality and linear-constraint feasibility at every iteration, and they stop when integrality is achieved. They are typically standard cutting plane procedures such as Gomory's algorithm [3]. The classical branch-and-bound scheme is also based on a dual-fractional approach, in particular for the determination of lower bounds. *Dual integral* methods maintain integrality and optimality, and they terminate once the primal linear constraints are satisfied. Letchford and Lodi give a single example: another algorithm of Gomory [4]. Finally, *primal algorithms* maintain feasibility (and integrality) throughout the process and stop when optimality is reached. These are in fact descent algorithms for which the improving sequence $(\boldsymbol{x}_k)_{k=1\dots K}$ satisfies the following conditions:

- `C1` $\boldsymbol{x}^k \in \mathcal{F}_{\text{ILP}}$;
- `C2` $\boldsymbol{x}^K$ is optimal;
- `C3` $\boldsymbol{c}^T\boldsymbol{x}^{k+1} < \boldsymbol{c}^T\boldsymbol{x}^k$.

A sequence satisfying the three conditions is called a sequence of *augmenting solutions* even in a minimization problem.

Given a current solution $\boldsymbol{x}^k \in \mathcal{F}_{\text{SPP}}$, primal algorithms are in fact based on the iterative solution of the *augmentation problem* defined as:

AUG	Find an augmenting vector $\boldsymbol{y} \in \mathbb{N}^n$ s.t. $\boldsymbol{x}^{k+1} = (\boldsymbol{x}^k + \boldsymbol{y}) \in \mathcal{F}_{\text{ILP}}$ and $\boldsymbol{c}^T\boldsymbol{y} < 0$ or assert that $\boldsymbol{x}^k$ is optimal for SPP.

Traditionally, papers on constraint aggregation and integral simplex algorithms deal with minimization problems, whereas most authors present generic primal algorithms for maximization problems. We therefore draw the reader's attention to the following: **to retain the usual classification, we call the improvement problem AUG, although it supplies a decreasing direction**. For further information on primal algorithms in a general context, see the review of Spille and Weismental [5].

Two special features of SPP make it a particularly promising candidate for specialized primal methods. First, by definition, SPP is a 0–1 program, so every (integer) point of $\mathcal{F}_{\text{SPP}}$ is an extreme point (or vertex) of $\mathcal{F}_{\text{SPP}^{RL}}$. Thus, there exists a decreasing sequence of basic solutions satisfying conditions `C1`–`C3` . Second, as shown by Trubin et al. [6], SPP is *quasi-integral*, i.e., every edge of $Conv(\mathcal{F}_{\text{SPP}})$ is an edge of $\mathcal{F}_{\text{SPP}^{RL}}$. Thus, there exists a sequence of augmenting solutions such that the following condition holds:

`C4` $\boldsymbol{x}^{k+1}$ is an adjacent vertex of $\boldsymbol{x}^k$ in $\mathcal{F}_{\text{SPP}^{RL}}$.

An augmentation that satisfies `C4` is called an *integral augmentation*. Any sequence satisfying `C1`–`C4` is a sequence of *augmenting adjacent solutions*, and an algorithm that yields such a sequence is an *integral simplex*. Every solution can be obtained from the previous one in the sequence by performing one or several simplex pivots in SPP^{RL}, hence the name. Such sequences were first introduced by Balas and Padberg [7]. Several other authors (Haus et al. [8], Thompson [9], Saxena [10]) have presented enumeration schemes that move from one integer solution to an adjacent one. An important recent contribution, in terms of both theory and applications, has been made by Rönnberg and Larsson [11, 12, 13]. However, none of these enumeration methods can guarantee a strict improvement (`C3`). They may perform degenerate pivots, in which there is no effective change in the solution (or the objective value) because the entering variable(s) takes the value zero. SPP tends to suffer severe degeneracy, so the computational time of these algorithms grows exponentially with the size of the instances.

We extend the theory and computations of Zaghrouti et al. [14] on the integral simplex using decomposition (ISUD). At each step of ISUD, we move from an integer vertex of $\mathcal{F}_{\text{SPP}^{RL}}$ to an adjacent vertex with a strictly better cost. In most cases, a better integer solution is obtained by solving a linear program, but in some cases we use a nonexhaustive branching method. Our method combines some results of Balas and Padberg [7] and Elhallaoui et al. [15]; it divides **AUG** into two subproblems and describes the necessary and sufficient conditions ensuring a transition from one integer solution to a strictly better one when a set of columns is pivoted into the working basis.

As will be explained in detail in Section 2, ISUD yields the *normalized direction* of the edge of $\mathcal{F}_{\text{SPP}^{RL}}$ that connects $\boldsymbol{x}^k$ to $\boldsymbol{x}^{k+1}$. It is therefore unable to predict the exact cost improvement $\boldsymbol{c}^T\boldsymbol{y}$ ($\boldsymbol{y} = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k$) and finds only a normalized reduced cost $(\boldsymbol{c}^T\boldsymbol{y})/(\boldsymbol{w}^T\boldsymbol{y})$, where $\boldsymbol{w}$ defines the weight of a normalization constraint $\boldsymbol{w}^T\boldsymbol{y} = 1$. The seminal paper of Zaghrouti et al. [14] did not conduct any analysis of these weights; they were fixed to 1 or 0 depending on the variable type (as discussed in Section 3.1). The aim of this paper is to show that the normalization weights influence both the theoretical and the practical behavior of the algorithm in terms of cost and integrality.

1.2 Contributions

In Section 2, we introduce a version of the ISUD algorithm with a generic convexity constraint in the augmentation subproblem. We give an original geometric interpretation of the row-reduced subproblem, and we present a detailed analysis of the linear transformation that binds the standard and the row-reduced augmentation problems. We then show that the extreme points of the domain of the linear relaxation are exactly those of the convex hull of integral directions. Moreover, we demonstrate the existence of a normalization constraint that gives a sequence of integer solutions satisfying C1–C4. We also prove an exact decomposition result that provides a stronger basis for the compatible-incompatible decomposition of ISUD. In Section 3, we present some normalization constraints. We prove a result concerning the weights used in ISUD that substantially modifies the algorithmic scheme, and we propose two new weight vectors. We give interpretations that better characterize the integral directions. We then use this characterization to penalize fractional solutions. Section 4 gives numerical results that support the theoretical work of the previous sections. Section 5 provides concluding remarks. This paper provides a proof of concept based on innovative theoretical results; a follow-up paper will focus on the numerical aspects.

2 Integral simplex using decomposition (ISUD): Generic framework

As mentioned in the Introduction, several augmentation algorithms have been proposed for the SPP. Some of them satisfy only conditions C1, C2, and C3, while others also guarantee C4. It is possible to find integral augmentations that satisfy C4 because the SPP is quasi-integral: there is an augmenting sequence of integral solutions $(x_k)_k$ such that for all k, $\boldsymbol{x}^{k+1}$ is a neighbor of $\boldsymbol{x}^k$ in $\mathcal{F}_{\mathrm{SPP}^{RL}}$ and can be obtained by performing a sequence of simplex pivots.

2.1 Maximum normalized augmentation: Specifics and linear formulation

We first introduce some useful notation. Hereinafter, $\boldsymbol{x}^0$ designates a known integer solution that we want to improve, and $\boldsymbol{x}^1$ is the next solution that we want to determine. $\mathcal{P}$ and $\mathcal{Z}$ respectively denote the set of indices of the positive and zero variables ($\boldsymbol{x}^0_{\mathcal{P}} = \boldsymbol{e}$, $\boldsymbol{x}^0_{\mathcal{Z}} = \boldsymbol{0}$). The set $(\mathcal{P}, \mathcal{Z})$ forms a partition of $\{1, \dots n\}$; $\mathcal{P}$ is called the *reduced basis* and has cardinality $p = |\mathcal{P}|$. Note that if $|\mathcal{P}| = m$, the submatrix $\boldsymbol{A}_{\cdot\mathcal{P}}$ is square and the partition $(\mathcal{P}, \mathcal{Z})$ corresponds to the simplex partition with basic and nonbasic variables (traditionally written $(\mathcal{B}, \mathcal{N})$).

To formulate **AUG** in a solvable way, we define $\boldsymbol{y} = \boldsymbol{x}^1 - \boldsymbol{x}^0$ and introduce the following definition:

Definition 1 A direction $\boldsymbol{d} \in \mathbb{R}^n$ is *feasible* at $\boldsymbol{x}^0$ if there exists a step $\rho > 0$ such that $\boldsymbol{x}^0 + \rho\boldsymbol{d} \in \mathcal{F}_{\mathrm{SPP}^{RL}}$, and it is *augmenting* if $\boldsymbol{c}^T\boldsymbol{d} < 0$. Moreover, given a feasible direction, we denote the largest feasible step by $r(\boldsymbol{d}) = \max\left\{\rho \in \mathbb{R} | \boldsymbol{x}^0 + \rho\boldsymbol{d} \in \mathcal{F}_{\mathrm{SPP}^{RL}}\right\} \geq 0$.

Ignoring the integrality constraints of SPP, we can therefore characterize an augmenting vector $\boldsymbol{y}$ in SPP^{RL} by a couple $(\boldsymbol{d}, \rho) \in \mathbb{R}^n \times \mathbb{R}$, where $\boldsymbol{d}$ can be normalized at will, as long as $\boldsymbol{x}^0 + \boldsymbol{y} = \boldsymbol{x}^0 + \rho\boldsymbol{d} \in \mathcal{F}_{\mathrm{SPP}^{RL}}$. The set of all normalized feasible directions can be described as the following cone:

$$\Gamma = \{\boldsymbol{d} \in \mathbb{R}^n \,|\, \boldsymbol{A}\boldsymbol{d} = 0\,,\, \boldsymbol{d}_{\mathcal{P}} \leq \boldsymbol{0}\,,\, \boldsymbol{d}_{\mathcal{Z}} \geq \boldsymbol{0}\} \tag{1}$$

from which we will only consider a section,

$$\Delta_{\boldsymbol{w}} = \left\{\boldsymbol{d} \in \mathbb{R}^n \,|\, \boldsymbol{A}\boldsymbol{d} = 0\,,\, \boldsymbol{w}^T\boldsymbol{d} = 1\,,\, \boldsymbol{d}_{\mathcal{P}} \leq \boldsymbol{0}\,,\, \boldsymbol{d}_{\mathcal{Z}} \geq \boldsymbol{0}\right\} \tag{2}$$

where $\boldsymbol{w}$ is the normalization vector. A geometric interpretation of Γ and $\Delta_{\boldsymbol{w}}$ is given in Figure 1.

In this paper, we will always assume that $\boldsymbol{w}_{\mathcal{P}} \leq \boldsymbol{0}$ and $\boldsymbol{w}_{\mathcal{Z}} > \boldsymbol{0}$, which are sufficient conditions for the cone section to be well defined. Indeed, in the general case, this ensures that whenever $\boldsymbol{d}$ is feasible, $\boldsymbol{w}^T\boldsymbol{d} > 0$ (because we have $\boldsymbol{A}\boldsymbol{d} = 0$, $\boldsymbol{d}_{\mathcal{Z}} = 0$ implies $\boldsymbol{d}_{\mathcal{P}} = 0$). Moreover, setting any of the $w_j = 0$ for $j \in \mathcal{Z}$ could make the problem unbounded. These conditions therefore seem sufficient without being restrictive.

The (fractional) augmentation problem can then be formulated as the following linear program, called the *maximum normalized augmentation*:

$$z^\star_{\mathrm{MNA}^{\boldsymbol{w}}} = \min\left\{\boldsymbol{c}^T\boldsymbol{d} \,|\, \boldsymbol{d} \in \Delta_{\boldsymbol{w}}\right\} \tag{$\mathrm{MNA}^{\boldsymbol{w}}$}$$

and the following proposition holds.

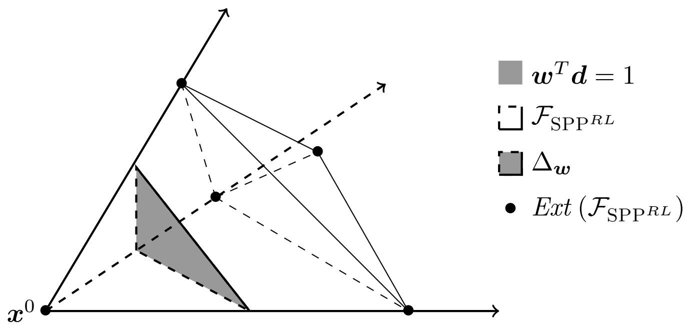

Figure 1: Geometric description of $\Delta_{\boldsymbol{w}}$. The cone of all feasible directions at $\boldsymbol{x}^0$ is defined by the three arrowed lines. The grey area represents $\Delta_{\boldsymbol{w}}$, i.e., the set of normalized feasible directions.

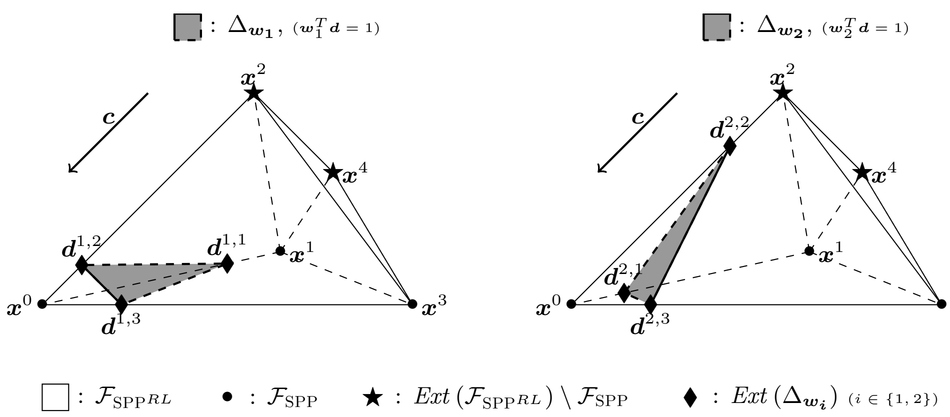

Figure 2: Example of two different convexity constraints for the same problem. The polyhedra represent $\mathcal{F}_{\mathrm{SPP}^{RL}}$, the feasible domain of SPP^{RL}. The dots ($\bullet$) are the integer extreme points (i.e., $\mathcal{F}_{\mathrm{SPP}}$), and the stars ($\bigstar$) are the fractional extreme points. The grey polyhedra are the cone sections $\Delta_{\boldsymbol{w_i}}$, $i \in \{1, 2\}$, and the diamonds are their extreme points.

Proposition 1 $\boldsymbol{x}^0$ *is optimal for* SPP^{RL} *if and only if* $z^{\star}_{\mathrm{MNA}\boldsymbol{w}} \geq 0$.

Proof. This proposition is a specialization of Proposition 1 of Omer et al. [16]. The proof is based on the following argument: on the one hand, if there exists an improving direction, taking a small step in this direction leads to a strictly better solution and the current one is nonoptimal. On the other hand, if the solution is nonoptimal, then any vector that leads to a strictly better solution can be normalized so that it becomes normalized, augmenting, and feasible. $\square$

The effect of a generalized convexity constraint is to maximize the marginal cost of a direction according to the weight vector $\boldsymbol{w}$. The geometric interpretation is shown in Figure 2: $\boldsymbol{d}^{1,1}$ and $\boldsymbol{d}^{2,2}$ are respectively optimal for the normalization weights $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. However, they respectively lead to $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$, which are different adjacent vertices of $\boldsymbol{x}^0$ in $\mathcal{F}_{\mathrm{SPP}^{RL}}$. Observe that in this example, $\boldsymbol{x}^1$ is integer, but $\boldsymbol{x}^2$ is fractional, showing that the normalization can influence the integrality of the next solution. Note also that $\boldsymbol{x}^4$ does not correspond to any extreme point of $\Delta_{\boldsymbol{w}}$, because it is not adjacent to $\boldsymbol{x}^0$.

Finally,

$$\textsc{Mna}^{\boldsymbol{w}} \Leftrightarrow \min \left\{ \frac{\boldsymbol{c}^T \boldsymbol{d}}{\boldsymbol{w}^T \boldsymbol{d}} \,|\, \boldsymbol{d} \in \Gamma \right\}. \tag{3}$$

In this equivalent nonlinear formulation, the domain is unbounded (cone Γ). However, the cost is constant for each direction of the cone, which ensures that formulation (3) is bounded.

2.2 Row-reduction of Mna$^{\boldsymbol{w}}$

In this section, we will show that p constraints and p variables can be eliminated from the linear program $\textsc{Mna}^{\boldsymbol{w}}$ if a suitable transformation on the constraint matrix is performed. This technique derives from the constraint aggregation methods first introduced by Elhallaoui et al. ([17] for SPP and [15] in a more general context). Only the nonbasic variables of $\mathcal{Z}$ are kept in the problem, the costs become the reduced costs computed as in the simplex algorithm, and the constraints are modified accordingly. First, observe that the constraint matrix can be partitioned as follows:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{I}_p & \boldsymbol{A}_{\mathcal{P}\mathcal{Z}} \\ \boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{P}} & \boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{Z}} \end{bmatrix} \tag{4}$$

where $\bar{P} = \{1, \dots m\} \setminus \mathcal{P}$ and $\boldsymbol{I}_p$ is the $p \times p$ identity matrix. The fact that the upper left-hand submatrix is an identity is a direct consequence of the linear constraints of SPP and the integrality of $\boldsymbol{x}^0 = (\boldsymbol{e}_{\mathcal{P}}, \boldsymbol{0}_{\mathcal{Z}})$.

Proposition 2 $\Delta_{\boldsymbol{w}} = \left\{\boldsymbol{d} = \boldsymbol{T}\boldsymbol{d}_{\mathcal{Z}} \,|\, \boldsymbol{d}_{\mathcal{Z}} \in \bar{\Delta}_{\boldsymbol{w}}\right\}$, *where*

$$\bar{\Delta}_{\boldsymbol{w}} = \left\{\boldsymbol{d}_{\mathcal{Z}} \in \mathbb{R}^{n-p} \,|\, \bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{Z}}\boldsymbol{d}_{\mathcal{Z}} = 0\,,\ \bar{\boldsymbol{w}}^T\boldsymbol{d}_{\mathcal{Z}} = 1\,,\ \boldsymbol{d}_{\mathcal{Z}} \geq \boldsymbol{0}\right\} \tag{5}$$

and

$$\begin{cases} \bar{\boldsymbol{w}}^T & = \boldsymbol{w}^T\boldsymbol{T} & = -\boldsymbol{w}_{\mathcal{P}}^T\boldsymbol{A}_{\mathcal{P}\mathcal{Z}} + \boldsymbol{w}_{\mathcal{Z}}^T \\ \bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{Z}} & = \boldsymbol{A}_{\bar{\mathcal{P}}.}\boldsymbol{T} & = -\boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{P}}\boldsymbol{A}_{\mathcal{P}\mathcal{Z}} + \boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{Z}} \end{cases} \tag{6}$$

and $\boldsymbol{T} = \left[\begin{smallmatrix} -\boldsymbol{A}_{\mathcal{P}\mathcal{Z}} \\ \boldsymbol{I}_{n-p} \end{smallmatrix}\right]$ *is the transformation matrix. Therefore, any feasible direction* $\boldsymbol{d}$ *is partitioned as* $\boldsymbol{d} = (\boldsymbol{d}_{\mathcal{P}}, \boldsymbol{d}_{\mathcal{Z}}) = (-\boldsymbol{A}_{\mathcal{P}\mathcal{Z}}\boldsymbol{d}_{\mathcal{Z}}, \boldsymbol{d}_{\mathcal{Z}}) = \boldsymbol{T}\boldsymbol{d}_{\mathcal{Z}}$.

Proof. Let $\boldsymbol{d} \in \mathbb{R}^n$. Given Equation (4) and since $\boldsymbol{A}$ is a binary matrix,

$$\begin{aligned} \boldsymbol{d} \in \Delta_{\boldsymbol{w}} \quad & \Leftrightarrow \quad \boldsymbol{A}\boldsymbol{d} = 0, \boldsymbol{w}^T\boldsymbol{d} = 1, \boldsymbol{d}_{\mathcal{P}} \leq \boldsymbol{0}, \boldsymbol{d}_{\mathcal{Z}} \geq 0 \\ & \Leftrightarrow \quad \begin{cases} \boldsymbol{d}_{\mathcal{P}} & +\boldsymbol{A}_{\mathcal{P}\mathcal{Z}}\boldsymbol{d}_{\mathcal{Z}} & = \boldsymbol{0} \\ \boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{P}}\boldsymbol{d}_{\mathcal{P}} & +\boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{Z}}\boldsymbol{d}_{\mathcal{Z}} & = \boldsymbol{0} \\ \boldsymbol{w}_{\mathcal{P}}^T\boldsymbol{d}_{\mathcal{P}} & +\boldsymbol{w}_{\mathcal{Z}}^T\boldsymbol{d}_{\mathcal{Z}} & = 1 \\ \boldsymbol{d}_{\mathcal{P}} \leq 0 & \boldsymbol{d}_{\mathcal{Z}} \geq 0 & \end{cases} \\ & \Leftrightarrow \quad \begin{cases} \boldsymbol{d}_{\mathcal{P}} & = -\boldsymbol{A}_{\mathcal{P}\mathcal{Z}}\boldsymbol{d}_{\mathcal{Z}} \\ \left(-\boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{P}}\boldsymbol{A}_{\mathcal{P}\mathcal{Z}} + \boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{Z}}\right)\boldsymbol{d}_{\mathcal{Z}} & = \boldsymbol{0} \\ \left(-\boldsymbol{w}_{\mathcal{P}}^T\boldsymbol{A}_{\mathcal{P}\mathcal{Z}} + \boldsymbol{w}_{\mathcal{Z}}^T\right)\boldsymbol{d}_{\mathcal{Z}} & = 1 \\ \boldsymbol{d}_{\mathcal{Z}} \geq 0 & \end{cases} \\ & \Leftrightarrow \quad \boldsymbol{d} = \boldsymbol{T}\boldsymbol{d}_{\mathcal{Z}} \text{ and } \boldsymbol{d}_{\mathcal{Z}} \in \bar{\Delta}_{\boldsymbol{w}} \end{aligned}$$

Hence, Proposition 2 holds. □

Regarding the cost of such a direction $\boldsymbol{d} = \boldsymbol{T}\boldsymbol{d}_{\mathcal{Z}}$, we have $\boldsymbol{c}^T\boldsymbol{d} = \boldsymbol{c}^T\boldsymbol{T}\boldsymbol{d}_{\mathcal{Z}}$. Write $\bar{\boldsymbol{c}} = \boldsymbol{T}^T\boldsymbol{c}$, i.e., $\bar{\boldsymbol{c}}^T = -\boldsymbol{c}_{\mathcal{P}}^T\boldsymbol{A}_{\mathcal{P}\mathcal{Z}} + \boldsymbol{c}_{\mathcal{Z}}^T$, then $\textsc{Mna}^{\boldsymbol{w}}$ is equivalent to the following program, called the *reduced* $\textsc{Mna}^{\boldsymbol{w}}$:

$$z^\star_{\text{R-MNA}^{\bar{\boldsymbol{w}}}} = \min \left\{\bar{\boldsymbol{c}}^T\boldsymbol{d}_{\mathcal{Z}} \,|\, \boldsymbol{d}_{\mathcal{Z}} \in \bar{\Delta}_{\boldsymbol{w}}\right\} \tag{R-Mna$^{\boldsymbol{w}}$}$$

The costs of R-$\textsc{Mna}^{\boldsymbol{w}}$ are in fact the reduced costs associated with the variables that are not in the working basis: their own marginal cost ($\boldsymbol{c}_{\mathcal{Z}}^T$), minus the marginal impact they have on the values of the variables

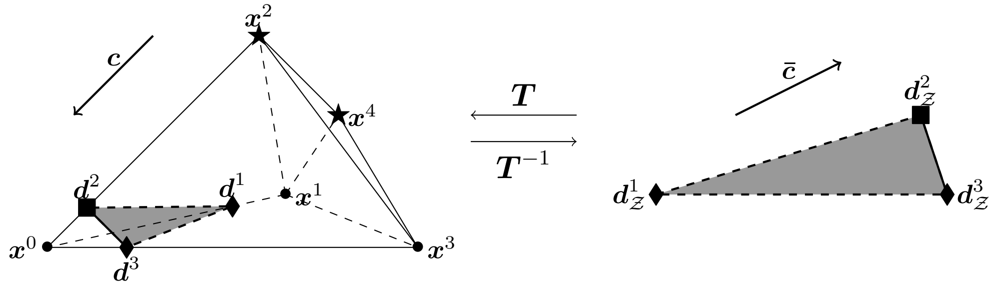

Figure 3: Geometric interpretation of transformation $\boldsymbol{T}$. The original space $\mathbb{R}^n$ with the feasible polyhedron $\mathcal{F}_{\text{SPP}^{RL}}$ is shown on the left. Bullets (•) indicate the integer extreme points ($\mathcal{F}_{\text{SPP}}$) and stars (★) indicate the fractional extreme points. The set of normalized feasible directions $\Delta_{\boldsymbol{w}}$ is shown in grey, and its extreme points are either integral (♦) or fractional (■) directions. The image of $\Delta_{\boldsymbol{w}}$ under $\boldsymbol{T}^{-1}$ in $\mathbb{R}^{(n-p)}$ is shown on the right. Diamonds (♦) indicate the integer reduced directions and squares (■) indicate the fractional reduced directions. The costs ($\boldsymbol{c} \in \mathbb{R}^n$) and reduced costs ($\bar{\boldsymbol{c}}$) are indicated. The solution of R-MNA$^{\boldsymbol{w}}$ is $\boldsymbol{d}^1_{\mathcal{Z}}$, corresponding to the integral direction $\boldsymbol{d}^1 = \boldsymbol{T}\boldsymbol{d}^1_{\mathcal{Z}}$.

from $\mathcal{P}$ if they enter the solution ($-\boldsymbol{w}_{\mathcal{P}}\boldsymbol{A}_{\mathcal{P}\mathcal{Z}}$). Interestingly, although not surprisingly, the same goes for the normalization constraint that undergoes the same linear transformation ($\bar{\boldsymbol{w}} = \boldsymbol{T}^T\boldsymbol{w}$). Proposition 2 describes only a global connection between $\Delta_{\boldsymbol{w}}$ and $\bar{\Delta}_{\boldsymbol{w}}$; this can be strengthened to a one-to-one correspondence:

Proposition 3 *Given normalization weights $\boldsymbol{w}$ in $\Delta_{\boldsymbol{w}}$ and the corresponding "reduced weights" $\bar{\boldsymbol{w}} = \boldsymbol{T}^T\boldsymbol{w}$ in $\bar{\Delta}_{\boldsymbol{w}}$, the function*

$$\begin{array}{rlcl} T: & \bar{\Delta}_{\boldsymbol{w}} \subseteq \mathbb{R}^n & \rightarrow & \Delta_{\boldsymbol{w}} \subseteq \mathbb{R}^{n-p} \\ & \boldsymbol{d}_{\mathcal{Z}} & \mapsto & \boldsymbol{d} = \boldsymbol{T}\boldsymbol{d}_{\mathcal{Z}} = (-\boldsymbol{A}_{\mathcal{P}\mathcal{Z}}\boldsymbol{d}_{\mathcal{Z}}, \boldsymbol{d}_{\mathcal{Z}}) \end{array} \tag{7}$$

is a bijective function. It maps every vertex of $\bar{\Delta}_{\boldsymbol{w}}$ to a vertex of $\Delta_{\boldsymbol{w}}$ with the same cost ($\bar{\boldsymbol{c}}^T\boldsymbol{d}_{\mathcal{Z}} = \boldsymbol{c}^T\boldsymbol{d}$), and vice versa.

Proof. This follows from the previous results and the fact that the image of a polyhedron after a linear transformation is a polyhedron whose vertices are the images of the vertices of the original polyhedron. □

Figure 3 gives a geometric interpretation of Proposition 3. As a consequence of the previous results, the following corollary holds.

Corollary 4 *$\boldsymbol{x}^0$ is optimal for SPP^{RL} if and only if $z^\star_{\text{R-MNA}^{\bar{\boldsymbol{w}}}} \geq 0$. Moreover, $\boldsymbol{d}^\star_{\mathcal{Z}}$ is an optimal solution of* R-MNA$^{\bar{\boldsymbol{w}}}$ *if and only if $\boldsymbol{d}^\star = \boldsymbol{T}\boldsymbol{d}^\star_{\mathcal{Z}}$ is an optimal solution of* MNA$^{\boldsymbol{w}}$.

From Proposition 3, we infer that if there exists an improving feasible direction, it can be expressed as the transformation $\boldsymbol{T}\boldsymbol{d}_{\mathcal{Z}}$ of any feasible solution $\boldsymbol{d}_{\mathcal{Z}} \in \bar{\Delta}_{\boldsymbol{w}}$ of negative reduced cost. Moreover, Corollary 4 shows that solving R-MNA$^{\bar{\boldsymbol{w}}}$ is sufficient to determine a feasible direction if one exists. The question of the fractional augmentation is thus equivalent to the question $z^\star_{\text{R-MNA}^{\bar{\boldsymbol{w}}}} < 0$?

Finally, note that MNA$^{\boldsymbol{w}}$ and MNA$^{\bar{\boldsymbol{w}}}$ are strictly equivalent, since they both yield the same R-MNA$^{\bar{\boldsymbol{w}}}$. This shows *a posteriori* that we could also have assumed $\boldsymbol{w}_{\mathcal{P}} = 0$ from the beginning, without any loss of generality. The normalization weights therefore need to apply only to the incoming variables and, when this is not the case, an equivalent weight vector $(\boldsymbol{0}, \bar{\boldsymbol{w}})$ can be easily determined. That condition still guarantees $\boldsymbol{w}_{\mathcal{Z}} > 0$ since $\boldsymbol{w}_{\mathcal{I}} = \bar{\boldsymbol{w}} = -\boldsymbol{A}_{\mathcal{P}\mathcal{Z}}\boldsymbol{w}_{\mathcal{P}} + \boldsymbol{w}_{\mathcal{Z}}$ and all coefficients of $\boldsymbol{A}$ are nonnegative, and $\boldsymbol{w}_{\mathcal{P}} \leq \boldsymbol{0}$.

Corollary 5 $\Delta_{\boldsymbol{w}} = \Delta_{\boldsymbol{w}'}$ *where $\boldsymbol{w}' = (\boldsymbol{0}, \bar{\boldsymbol{w}})$ with $\boldsymbol{w}'_{\mathcal{P}} = \boldsymbol{0}$.*

Hereinafter, unless stated otherwise, we will assume that $\boldsymbol{w}_{\mathcal{P}} = 0$.

Finally, in the same way that we proposed a nonlinear equivalent to the full augmentation problem, we can rewrite R-MNA$^{\bar{\boldsymbol{w}}}$ as

$$\text{R-MNA}^{\bar{\boldsymbol{w}}} \Leftrightarrow \min \left\{ \frac{\bar{\boldsymbol{c}}^T \boldsymbol{d}_{\mathcal{Z}}}{\bar{\boldsymbol{w}}^T \boldsymbol{d}_{\mathcal{Z}}} \mid \boldsymbol{d} \in \bar{\Gamma} \right\}, \tag{8}$$

where $\bar{\Gamma} = \left\{ \bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{Z}} \boldsymbol{d}_{\mathcal{Z}} = 0 \,,\, \boldsymbol{d}_{\mathcal{Z}} \geq \boldsymbol{0} \right\}$ is the cone of all normalized feasible directions. The domain of this nonlinear formulation is again unbounded, but formulation (8) is bounded because the reduced cost is constant for each direction of the cone.

2.3 Geometric structure of $\bar{\Delta}_{\boldsymbol{w}}$

The previous sections give a way to find a fractional augmentation. Given $\boldsymbol{d} \in \Delta_{\boldsymbol{w}}$, $\boldsymbol{x}^1 = \boldsymbol{x}^0 + r(\boldsymbol{d})\boldsymbol{d} \in \mathcal{F}_{\text{SPP}^{RL}}$ may indeed be fractional and not in $\mathcal{F}_{\text{SPP}}$. If it is integer, $\boldsymbol{d}$ (or $\boldsymbol{d}_{\mathcal{Z}}$) is called an *integral direction*; otherwise it is *fractional*. In Sections 2.3 and 2.4, we characterize these integer directions and show that $\boldsymbol{d}$ is integral provided $\boldsymbol{w}$ is chosen well.

For any polyhedron P, let $Ext\,(P)$ be the set of its extreme points. Moreover, a solution $\boldsymbol{d}_{\mathcal{Z}}$ of R-MNA$^{\bar{\boldsymbol{w}}}$ is *minimal* if there exists no other solution of R-MNA$^{\bar{\boldsymbol{w}}}$ whose support is strictly contained in that of $\boldsymbol{d}$. Terms such as *nondecomposable* [7] or *irreducible* [8] are also used to describe such directions. Lemma 6 below will be used in the next section and shows that, provided we use the simplex algorithm, the solution of R-MNA$^{\bar{\boldsymbol{w}}}$ is indeed minimal.

Lemma 6 *$Ext\left(\bar{\Delta}_{\boldsymbol{w}}\right)$ is the set of minimal directions of $\bar{\Delta}_{\boldsymbol{w}}$.*

Proof. First, let $\boldsymbol{d}^\star_{\mathcal{Z}} \in Ext\left(\bar{\Delta}_{\boldsymbol{w}}\right)$; we will prove that it is minimal. Let $\boldsymbol{u}_{\mathcal{Z}} \in \bar{\Delta}_{\boldsymbol{w}}$ such that $Supp(\boldsymbol{u}_{\mathcal{Z}}) \subseteq Supp(\boldsymbol{d}^\star_{\mathcal{Z}})$. Moreover, both $\boldsymbol{u}_{\mathcal{Z}} \geq 0$ and $\boldsymbol{d}^\star_{\mathcal{Z}} \geq 0$, so there exists $\epsilon \in (0,1)$ such that $\boldsymbol{v}_{\mathcal{Z}} = \boldsymbol{d}^\star_{\mathcal{Z}} - \epsilon \boldsymbol{u}_{\mathcal{Z}} \geq 0$. Let $\boldsymbol{v}'_{\mathcal{Z}} = \boldsymbol{v}_{\mathcal{Z}} / (1-\epsilon)$. $\boldsymbol{v}'_{\mathcal{Z}}$ satisfies $\bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{Z}} \boldsymbol{v}'_{\mathcal{Z}} = \frac{1}{1-\epsilon}\left(\bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{Z}} \boldsymbol{d}^\star_{\mathcal{Z}} - \epsilon \bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{Z}} \boldsymbol{u}_{\mathcal{Z}}\right) = \boldsymbol{0}$, $\bar{\boldsymbol{w}}^T \boldsymbol{v}'_{\mathcal{Z}} = \frac{\bar{w}^T \boldsymbol{d}^\star_{\mathcal{Z}} - \epsilon \bar{w}^T \boldsymbol{u}_{\mathcal{Z}}}{1-\epsilon} = \frac{1-\epsilon}{1-\epsilon} = 1$, and $\boldsymbol{v}'_{\mathcal{Z}} \geq 0$. Hence, $\boldsymbol{v}'_{\mathcal{Z}} \in \bar{\Delta}_{\boldsymbol{w}}$ and $\boldsymbol{d}^\star_{\mathcal{Z}} = \epsilon \boldsymbol{u}_{\mathcal{Z}} + (1-\epsilon)\,\boldsymbol{v}'_{\mathcal{Z}}$ is a convex combination of two elements of $\bar{\Delta}_{\boldsymbol{w}}$. Since $\boldsymbol{d}^\star_{\mathcal{Z}} \in Ext\left(\bar{\Delta}_{\boldsymbol{w}}\right)$, we have $\boldsymbol{u}_{\mathcal{Z}} = \boldsymbol{v}'_{\mathcal{Z}} = \boldsymbol{d}_{\mathcal{Z}}$.

Now, let $\boldsymbol{d}^\star_{\mathcal{Z}} \in \bar{\Delta}_{\boldsymbol{w}}$ be a minimal direction; we will prove that $\boldsymbol{d}^\star_{\mathcal{Z}} \in Ext\left(\Delta_{\boldsymbol{w}}\right)$. Let $\epsilon > 0$, $\boldsymbol{u}_{\mathcal{Z}}$, and $\boldsymbol{v}_{\mathcal{Z}}$ be such that

$$\begin{cases} \boldsymbol{d}^\star_{\mathcal{Z}} = \epsilon \boldsymbol{u}_{\mathcal{Z}} + (1-\epsilon)\,\boldsymbol{v}_{\mathcal{Z}} \\ \boldsymbol{u}_{\mathcal{Z}}, \boldsymbol{v}_{\mathcal{Z}} \in \bar{\Delta}_{\boldsymbol{w}} \end{cases}$$

and suppose $\boldsymbol{u}_{\mathcal{Z}} \neq \boldsymbol{d}^\star_{\mathcal{Z}}$, $\boldsymbol{d}^\star_{\mathcal{Z}} \neq \boldsymbol{d}^\star_{\mathcal{Z}}$. Since $\boldsymbol{d}^\star_{\mathcal{Z}}$ is minimal, $Supp(\boldsymbol{u}_{\mathcal{Z}})$ and $Supp(\boldsymbol{v}_{\mathcal{Z}})$ cannot be contained in $Supp(\boldsymbol{d}^\star_{\mathcal{Z}})$, so there exists $j \notin Supp(\boldsymbol{d}^\star_{\mathcal{Z}})$ such that $\epsilon u_j + (1-\epsilon)\,v_j = 0$ and $u_j \geq 0$ and $v_j \neq 0$. Hence, either u_j or v_j is negative, which contradicts $\boldsymbol{u}_{\mathcal{Z}}, \boldsymbol{v}_{\mathcal{Z}} \in \bar{\Delta}_{\boldsymbol{w}}$. Therefore, $\boldsymbol{d}^\star_{\mathcal{Z}} = \boldsymbol{u}_{\mathcal{Z}} = \boldsymbol{v}_{\mathcal{Z}}$ and $\boldsymbol{d}^\star_{\mathcal{Z}} \in Ext\left(\bar{\Delta}_{\boldsymbol{w}}\right)$. □

2.4 Normalization and integrality

Let us now consider the integrality of the directions. Let $\bar{\Delta}^{int}_{\boldsymbol{w}}$ (or $\Delta^{int}_{\boldsymbol{w}}$) be the set of integral directions at $\boldsymbol{x}^0$. The quasi-integral property of SPP yields

$$\bar{\Delta}^{int}_{\boldsymbol{w}} \subseteq Conv(\bar{\Delta}^{int}_{\boldsymbol{w}} \cap Ext\left(\bar{\Delta}_{\boldsymbol{w}}\right)), \tag{9}$$

i.e., the integral directions are contained in the convex hull of those that are also extreme points of $\bar{\Delta}_{\boldsymbol{w}}$. Therefore, we can restrict the search for an augmenting integral direction to the extreme points of $\bar{\Delta}_{\boldsymbol{w}}$.

Proposition 7 gives a simple way to test whether or not $\boldsymbol{d}^\star_{\mathcal{Z}} \in Ext\left(\bar{\Delta}_{\boldsymbol{w}}\right)$ is also in $\bar{\Delta}^{int}_{\boldsymbol{w}}$.

Definition 2 Given a set of indices of the variables $\mathcal{S} \subseteq \{1, \ldots n\}$, it is *column-disjoint* if no pair of columns of $\boldsymbol{A}_{\cdot\mathcal{S}}$ has a common nonzero entry, i.e., $\forall i, j \in \mathcal{S}, i \neq j, \forall k \in \{1, \ldots m\}, A_{ki} A_{kj} = 0$. This definition is extended to $\boldsymbol{A}_{\cdot\mathcal{S}}$ and $\boldsymbol{x}_{\mathcal{S}}$. Since $\boldsymbol{A}$ is binary, $\mathcal{S}$ is column-disjoint iff $\boldsymbol{A}_{\cdot\mathcal{S}}$ is a set of orthogonal columns.

Proposition 7 *Given $\boldsymbol{d}^\star_{\mathcal{Z}} \in Ext\left(\bar{\Delta}_{\boldsymbol{w}}\right)$ and $\mathcal{S} = Supp(\boldsymbol{d}^\star_{\mathcal{Z}})$,*

$$\boldsymbol{d}^\star_{\mathcal{Z}} \textit{ is an integral direction} \Leftrightarrow \mathcal{S} \textit{ is column-disjoint.} \tag{10}$$

For such a direction, $d^\star_j = 1/\bar{W}_{\mathcal{S}}$ if $j \in \mathcal{S}$ and 0 otherwise, where $\bar{W}_{\mathcal{S}} = \sum_{j\in\mathcal{S}} \bar{w}j$. Also, $r(\boldsymbol{d}_{\mathcal{Z}}) = \bar{W}_{\mathcal{S}}$.

Proof. Zaghrouti et al. [14] proved this proposition in the particular case of $\bar{w} = e$. We will show that, whatever the convexity constraint, we can reduce the problem to this case. Let $\boldsymbol{d}^\star_{\mathcal{Z}} \in Ext\left(\bar{\Delta}_{\boldsymbol{w}}\right)$, $\mathcal{S} = Supp(\boldsymbol{d}^\star_{\mathcal{Z}})$, and $\boldsymbol{\delta}^\star_{\mathcal{Z}} = \boldsymbol{d}^\star_{\mathcal{Z}}/(\boldsymbol{e}^T\boldsymbol{d}^\star_{\mathcal{Z}})$. $\boldsymbol{\delta}^\star_{\mathcal{Z}}$ is the extreme point of $\bar{\Delta}_{\boldsymbol{e}} = \left\{\boldsymbol{d}_{\mathcal{Z}} \mid \bar{\boldsymbol{A}}_{\mathcal{P}\mathcal{Z}}\boldsymbol{d}_{\mathcal{Z}} = 0\,,\, \boldsymbol{e}^T\boldsymbol{d}_{\mathcal{Z}} = 1\,,\, \boldsymbol{d}_{\mathcal{Z}} \geq \boldsymbol{0}\right\}$ that represents the same direction normalized with a different weight vector. Hence, $Supp(\boldsymbol{\delta}^\star_{\mathcal{Z}}) = \mathcal{S}$ and using the result of Zaghrouti et al.,

$$\begin{aligned}
\boldsymbol{d}^\star_{\mathcal{Z}} \text{ is an integral direction} \quad &\Leftrightarrow \quad \boldsymbol{\delta}^\star_{\mathcal{Z}} \text{ is an integral direction}\\
&\Leftrightarrow \quad Supp(\boldsymbol{\delta}^\star_{\mathcal{Z}}) \text{ is column-disjoint}\\
&\Leftrightarrow \quad \mathcal{S} \text{ is column-disjoint}
\end{aligned}$$

The rest of the proposition follows from the convexity constraint $\bar{\boldsymbol{w}}^T\boldsymbol{d}_{\mathcal{Z}} = 1$. □

Set $\mathcal{S} = Supp(\boldsymbol{d}_{\mathcal{Z}})$ is the set of columns to be pivoted into the reduced basis to improve the value of the solution. By Lemma 6, all the extreme integral solutions are minimal. As shown in [14], the support corresponds to a nondecomposable set of Balas on the global direction $\boldsymbol{d} = \boldsymbol{T}\boldsymbol{d}_{\mathcal{Z}}$, which yields Corollary 8.

Corollary 8 *Given $\boldsymbol{d}^\star_{\mathcal{Z}} \in \bar{\Delta}^{int}_{\boldsymbol{w}} \cap Ext\left(\bar{\Delta}_{\boldsymbol{w}}\right)$, the corresponding extreme direction $\boldsymbol{d}^\star = \boldsymbol{T}\boldsymbol{d}^\star_{\mathcal{Z}}$ is*

$$d_j = \begin{cases} -\alpha & \textit{if } j \in \mathcal{R}\\ \alpha & \textit{if } j \in \mathcal{S}\\ 0 & \textit{otherwise}\end{cases}$$

where $\mathcal{S} = Supp(\boldsymbol{d}^\star_{\mathcal{Z}})$, $\mathcal{R} = Supp(\boldsymbol{d}^\star_{\mathcal{P}})$, and $\alpha = 1/\bar{W}_{\mathcal{S}}$. Moreover, $r(\boldsymbol{d}^\star) = \bar{W}_{\mathcal{S}}$.

The next integer solution is then $\boldsymbol{x}^1 = \boldsymbol{x}^0 + r(\boldsymbol{d}^\star)\boldsymbol{d}^\star$ defined as

$$x^1_j = \begin{cases} 1 & \textit{if } j \in \mathcal{S} \cup (\mathcal{P} \setminus \mathcal{R})\\ 0 & \textit{if } j \in \mathcal{R} \cup (\mathcal{Z} \setminus \mathcal{S})\end{cases}$$

and the partition of the variables becomes (at $\boldsymbol{x}^1$): $\mathcal{P}^1 = \mathcal{S} \cup (\mathcal{P} \setminus \mathcal{R})$ and $\mathcal{Z}^1 = \mathcal{R} \cup (\mathcal{Z} \setminus \mathcal{S})$.

Classical linear programming theory ensures that at least one of the optimal solutions of R-MNA$^{\bar{\boldsymbol{w}}}$ is an extreme point of $\bar{\boldsymbol{\Delta}}_{\boldsymbol{w}}$, i.e., the direction of an edge at $\boldsymbol{x}^0$. Solving the problem with the simplex algorithm naturally yields an extreme solution, and the previous results play a role in that decision. Moreover, as seen in Equation (8), the normalization constraint influences the marginal cost of the vertices of $\bar{\Delta}_{\boldsymbol{w}}$. For instance, with the classic normalization constraint, $\bar{\boldsymbol{w}} = \boldsymbol{e}$, this impact is measured by the size of the entering set $|\mathcal{S}|$. With the generalized version, we are free to choose this measure. We can thus hope that a convenient modification of $\bar{\boldsymbol{w}}$ will increase the likelihood that the optimal solution $\boldsymbol{d}^\star_{\mathcal{Z}}$ of R-MNA$^{\bar{\boldsymbol{w}}}$ found by the simplex algorithm leads to an integer $\boldsymbol{x}^1 = \boldsymbol{x}^0 + r(\boldsymbol{d}^\star_{\mathcal{Z}})\boldsymbol{d}^\star_{\mathcal{Z}}$. The following existence results prove the theoretical validity of this approach.

Lemma 9 *Given any minimal feasible integral direction $\boldsymbol{d}$ of negative cost $\boldsymbol{c}^T\boldsymbol{d} < 0$, there exists $\boldsymbol{w} \in \mathbb{R}^n$ such that $\boldsymbol{d}^{\boldsymbol{w}} = \boldsymbol{d}/(\boldsymbol{w}^T\boldsymbol{d})$ is an optimal solution of MNA$^{\boldsymbol{w}}$ that represents the same geometric direction.*

Proof. Let $\boldsymbol{d}$ be any minimal feasible integral direction of negative cost, and $\mathcal{S} = Supp(\boldsymbol{d}_{\mathcal{Z}})$. Let $M > 0$ and define the weight vector $\boldsymbol{w}$ to be $w_j = 1$ if $j \in \mathcal{S}$, and M otherwise. Let $\boldsymbol{d}^{\boldsymbol{e}} = \boldsymbol{d}/(\boldsymbol{e}^T\boldsymbol{d}) \in \Delta_{\boldsymbol{e}}$ and $\boldsymbol{d}^{\boldsymbol{w}} = \boldsymbol{d}/(\boldsymbol{w}^T\boldsymbol{d}) \in \Delta_{\boldsymbol{w}}$ be the normalized directions corresponding to $\boldsymbol{d}$. We will prove that if

$$M > \max\left\{\frac{\boldsymbol{c}^T\boldsymbol{u}^{\boldsymbol{e}}}{\left(\sum_{j\notin S} u^{\boldsymbol{e}}_j\right)\boldsymbol{c}^T\boldsymbol{d}^{\boldsymbol{e}}} \mid \boldsymbol{u}^{\boldsymbol{e}} \in Ext\left(\Delta_{\boldsymbol{e}}\right) \setminus \{\boldsymbol{d}^{\boldsymbol{e}}\}\right\}, \tag{11}$$

then $\boldsymbol{d^w}$ is optimal for $\textsc{Mna}^{\boldsymbol{w}}$.

Let $\boldsymbol{u^w} \in \mathit{Ext}\,(\Delta_{\boldsymbol{w}})$ and $\boldsymbol{u^e} = \boldsymbol{u^w} / \left(\boldsymbol{e}^T \boldsymbol{u^w}\right)$ be the corresponding direction in $\Delta_{\boldsymbol{e}}$. If $\boldsymbol{c}^T \boldsymbol{u^w} \geq 0$, then $\boldsymbol{u^w}$ is clearly not better than the negative-cost $\boldsymbol{d^w}$. If $\boldsymbol{c}^T \boldsymbol{u^w} < 0$ then, given the condition on M and the positivity of $\boldsymbol{u^e}$,

$$\begin{aligned}
\boldsymbol{c}^T \boldsymbol{u^w} = \frac{\boldsymbol{c}^T \boldsymbol{u^e}}{\boldsymbol{w}^T \boldsymbol{u^e}} \quad &= \quad \frac{\boldsymbol{c}^T \boldsymbol{u^e}}{\sum\limits_{i \in \mathcal{S}} u_i^{\boldsymbol{e}} + M \sum\limits_{j \notin \mathcal{S}} u^{\boldsymbol{e}}} \\
&\geq \quad \frac{\boldsymbol{c}^T \boldsymbol{u^e}}{M \sum\limits_{j \notin \mathcal{S}} u^{\boldsymbol{e}}} \\
&> \quad \frac{\boldsymbol{c}^T \boldsymbol{u^e}}{\frac{\boldsymbol{c}^T \boldsymbol{u^e}}{\left(\sum_{j \notin S} u_j^{\boldsymbol{e}}\right) \boldsymbol{c}^T \boldsymbol{d^e}} \sum\limits_{j \notin \mathcal{S}} u^{\boldsymbol{e}}} = \boldsymbol{c}^T \boldsymbol{d^w}
\end{aligned}$$

□

Theorem 10 *There exists a vector* $\boldsymbol{w} = (w_1, \ldots, w_n) \in \mathbb{R}_+^n$ *such that the repeated solution of* $\textsc{Mna}^{\boldsymbol{w}}$ *(or* $\textsc{r-Mna}^{\bar{\boldsymbol{w}}}$ *with* $\bar{w} = \boldsymbol{T_k w}$*) yields an augmenting sequence of (integer) solutions of SPP leading to an optimal solution, i.e., satisfying conditions* `C1`*–*`C4`*. Note that the normalization weights do not change during the process.*

Proof. Let $\boldsymbol{x}^0 \in \mathcal{F}_{\text{SPP}}$ be the starting point of the sequence. Since SPP is quasi-integral, there exist $K \in \mathbb{N}$ and an augmenting sequence $\left(\boldsymbol{x}^0, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^K\right)$ of solutions of SPP satisfying `C1`–`C4`. For $k = 0$ to $K-1$, denote $\mathcal{S}_k = \left\{j | x_j^{k+1} - x_j^k > 0\right\}$. $\mathcal{S}_k$ is the nondecomposable set associated with direction $\boldsymbol{d}^k = \frac{\boldsymbol{x}^1 - \boldsymbol{x}^0}{\boldsymbol{w}^T(\boldsymbol{x}^1 - \boldsymbol{x}^0)}$, i.e., the support of $\boldsymbol{d}_{\mathcal{Z}}^k$ ($\mathcal{Z}$ depends on each new solution). Also, each $\mathcal{S}_k$ is a minimal set.

Let $M \in \mathbb{R}$ be a sufficiently large but finite number (as suggested in the proof of Lemma 9), and let $\boldsymbol{w}_j$ be such that

$$w_j = \begin{cases} M^i & \text{if } j \in \mathcal{S}_i \\ M^K & \text{if } j \notin \bigcup\limits_{k=0}^{K-1} \mathcal{S}_k \end{cases},$$

where M^i is M to the power of i. If M is sufficiently large, similarly to the proof of Lemma 9, this convexity constraint yields a sequence of minimal integer solutions $\boldsymbol{d}_k$ of support $\mathit{Supp}(\boldsymbol{d}_k) = \mathcal{S}_k$ and the result holds. To ensure that M is finite, one can choose the largest number given by Equation (11) for each of the iterations (a finite number). □

Theorem 10 does not show how to find the right convexity constraint, but it provides a strong theoretical base for the method we present here. It also confirms the potential effects of the normalization on the outcome of the algorithm. Note also that the constant weight vector $\boldsymbol{w}$ applies to $\textsc{Mna}$ and not to $\textsc{r-Mna}$. The reduced-weight vector $\bar{w}$ depends on the current partition $(\mathcal{P}, \mathcal{Z})$, in the same way that the cost vector $\boldsymbol{c}$ is constant, but the reduced-cost vector $\bar{\boldsymbol{c}}$ changes with the solution. Before studying some specific weight vectors, we will discuss another feature of our algorithm: the decomposition of $\textsc{r-Mna}^{\boldsymbol{w}}$ into two smaller problems.

2.5 Decomposition of the augmentation problem

Another efficient feature of ISUD is the decomposition of the search space into two smaller spaces. We now introduce a definition and recall previous results for the ISUD.

Definition 3 Given $j \in \mathcal{Z}$, column $\boldsymbol{A}_{\cdot j}$ is said to be *compatible* with the working basis $\mathcal{P}$ if $\boldsymbol{A}_{\cdot j} \in \mathit{Span}(\boldsymbol{A}_{\cdot j} | j \in \mathcal{P})$. The same definition applies to index j and variable x_j. The other columns are said to be *incompatible*. $\mathcal{C}$ and $\mathcal{I}$ respectively denote the sets of all compatible and incompatible indices.

Lemma 11 (Lemma 1 of Rosat et al. [18]) *Column $\boldsymbol{A}_{\boldsymbol{\cdot} j}$ is compatible iff it is the sum of a subset of the basic columns. This set is unique and denoted $\mathcal{R}_j$, and $\boldsymbol{A}_{\boldsymbol{\cdot} j} = \sum_{i\in\mathcal{R}_j} \boldsymbol{A}_{\boldsymbol{\cdot} i}$.*

Observation 1 *Column $\boldsymbol{A}_{\boldsymbol{\cdot} j}$ is compatible iff $\bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{Z}}\boldsymbol{A}_{\boldsymbol{\cdot} j} = 0$.*

Given $j \in \mathcal{C}$, its associated direction $\boldsymbol{\delta}^j$ is defined to be

$$\forall i \in \{1,\dots n\}\,, \delta_i^j = \begin{cases} -1/W^j & \text{if } i \in \mathcal{R}_j \\ 1/W^j & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where $W^j = w_j - \sum_{i\in\mathcal{R}_j} w_j$ is feasible and integer. Also, $r(\boldsymbol{\delta}^j) = W^j$.

Lemma 12 *$\boldsymbol{\delta}^j \in Ext\,(\Delta_{\boldsymbol{w}})$ and is minimal for $\Delta_{\boldsymbol{w}}$.*

Proof. By the definition of $\boldsymbol{\delta}^j$ and Observation 1, $\boldsymbol{\delta}^j \in \bar{\Delta}_{\boldsymbol{w}}$. Since $\boldsymbol{\delta}^j_{\mathcal{Z}}$ has only one nonzero entry, it is necessarily minimal in $\bar{\Delta}_{\boldsymbol{w}}$. By Lemma 6, $\boldsymbol{\delta}^j_{\mathcal{Z}} \in Ext\left(\bar{\Delta}_{\boldsymbol{w}}\right)$, so by Proposition 2, the lemma holds. □

If its reduced cost $\bar{c}_j = c_j - \sum_{i\in\mathcal{R}_j} c_j < 0$, then $\boldsymbol{\delta}^j$ is augmenting (and so is the corresponding normalized direction). Following $\boldsymbol{\delta}^j$ is equivalent to pivoting j into the working basis and making $\mathcal{R}_j$ exit. Incompatible variables yield degenerate pivots. The following decomposition theorem justifies the use of the decomposition.

Theorem 13

$$z^\star_{\text{R-MNA}^{\boldsymbol{w}}} = \min\left\{z^\star_{\text{RP}}, z^\star_{\text{R-MNA}^{\bar{\boldsymbol{w}}}_{\mathcal{I}}}\right\} \tag{13}$$

where R-MNA$^{\boldsymbol{w}}_{\mathcal{I}}$ is the restriction of R-MNA$^{\boldsymbol{w}}$ to the incompatible variables, and the reduced problem RP is

$$z^\star_{\text{RP}} = \min\left\{\frac{\bar{c}_j}{W^j} | j \in \mathcal{C}\right\}, \tag{RP}$$

which is the normalized reduced-cost problem.

Proof. Let $\boldsymbol{d} \in Ext\,(\Delta_{\boldsymbol{w}})$, so $\boldsymbol{d}_{\mathcal{Z}} \in Ext\left(\bar{\Delta}_{\boldsymbol{w}}\right)$. Let $S = Supp(\boldsymbol{d}_{\mathcal{Z}})$. We will show that either $S \subseteq \mathcal{I}$ or there exists $j \in \mathcal{C}$ such that $\boldsymbol{d} = \boldsymbol{\delta}^j$. If this assertion is true, the result will hold.

Suppose that $S \nsubseteq \mathcal{I}$ and let $j \in \mathcal{C}$ be such that $d_j > 0$. By Lemma 6, $\boldsymbol{d}_{\mathcal{Z}}$ is minimal in $\bar{\Delta}_{\boldsymbol{w}}$. Moreover, $Supp(\boldsymbol{\delta}^j_{\mathcal{Z}}) \subseteq S$. Therefore, $\boldsymbol{\delta}^j_{\mathcal{Z}}$ is a feasible solution whose support is contained in the support of the minimal solution $\boldsymbol{d}_{\mathcal{Z}}$. Hence, $\boldsymbol{d}_{\mathcal{Z}} = \boldsymbol{\delta}^j_{\mathcal{Z}}$ and either $S \subseteq \mathcal{I}$ or there exists $j \in \mathcal{C}$ such that $\boldsymbol{d} = \boldsymbol{\delta}^j_{\mathcal{Z}}$. □

Theorem 13 strengthens the result of Zaghrouti et al. [14] that states that, for $\bar{\boldsymbol{w}} = \boldsymbol{e}$, $z^\star_{\text{R-MNA}^{\boldsymbol{w}}} < 0 \Leftrightarrow (z^\star_{\text{RP}} < 0$ or $z^\star_{\text{R-MNA}^{\bar{\boldsymbol{w}}}_{\mathcal{I}}} < 0)$. Our theorem strengthens this in two ways: any normalization constraint can be used, and we have shown the equality of the left-hand and right-hand sides. This result justifies the algorithmic scheme given in the next section: first look for an improving compatible column; if none is found, look for a linear combination of incompatible columns that is compatible overall.

2.6 Algorithmic scheme

Algorithm 1 summarizes the previous ideas. Various branching and cutting-plane techniques can be used to encourage integral solutions; they mostly derive from primal methods. A study of the effects of cutting planes in ISUD can be found in [18] and [19]. Examples of branching techniques can be found in [14].

Algorithm 1: Integral Simplex Using Decomposition with Normalization Weight Vector $\boldsymbol{w}$

Input: $\boldsymbol{x}^0$, a solution of SPP.
Output: $\boldsymbol{x}^k$, a better solution of SPP.

Compute partition $\left(\mathcal{P}^k, \mathcal{C}^k, \mathcal{I}^k\right)$ associated with $\boldsymbol{x}^k$; $k \leftarrow 0$;
while *true* **do**
  if $z^\star_{\mathrm{RP}} < 0$ **then**
    $\boldsymbol{d}^i \leftarrow$ an optimal solution of RP ($\bar{c}_i < 0$, $i \in \mathcal{C}^k$);
    $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + r(\boldsymbol{d}^i)\boldsymbol{d}^i$; $k \leftarrow k+1$; Update $\left(\mathcal{P}^k, \mathcal{C}^k, \mathcal{I}^k\right)$;
  else if $z^\star_{\text{R-MNA}^{\bar{\boldsymbol{w}}}} < 0$ **then**
    $\boldsymbol{d}^\star_{\mathcal{Z}} \leftarrow$ an optimal solution of R-MNA$^{\boldsymbol{w}}$; $\boldsymbol{d}^\star \leftarrow \boldsymbol{T}\boldsymbol{d}_{\mathcal{Z}}$;
    if $\boldsymbol{d}^\star_{\mathcal{Z}}$ *is column-disjoint* **then**
      $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + r(\boldsymbol{d}^\star)\boldsymbol{d}^\star$; $k \leftarrow k+1$; Update $\left(\mathcal{P}^k, \mathcal{C}^k, \mathcal{I}^k\right)$;
    else
      if *stopping criterion is reached* **then**
        return $\boldsymbol{x}^k$ (potentially nonoptimal);
      else
        Use branching or cutting-plane techniques or modify $\boldsymbol{w}$ to encourage column-disjoint solutions in R-MNA$^{\bar{\boldsymbol{w}}}$;
  else
    return $\boldsymbol{x}^k$ (optimal)

3 Convexity constraint in ISUD

We now study some specific normalization constraints. First, we will present the constraint currently used in ISUD ($\bar{\boldsymbol{w}} = \boldsymbol{e}$) and give a new theoretical result that modifies the algorithm when this constraint is used. Second, we will present the standard MMA constraint ($\boldsymbol{w} = (-\boldsymbol{e}_{\mathcal{P}}, \boldsymbol{e}_{\mathcal{Z}})$). Third, we will introduce two new constraints and explain their advantages. The normalization constraint is effectively a pricing rule: Equation (3) shows that this constraint directly influences the cost function.

3.1 Maximum incoming mean augmentation (Mima)

To date, the ISUD algorithm has always been run with a convexity constraint on the incoming variables of $\boldsymbol{Z}$ (or $\boldsymbol{I}$ in the decomposition). That is, the algorithm solved the following augmentation, called the *maximum incoming mean augmentation* (MIMA):

$$z^\star_{\text{MIMA}} = \min\left\{\bar{\boldsymbol{c}}^T\boldsymbol{d}_{\mathcal{Z}} \mid \boldsymbol{d}_{\mathcal{Z}} \in \bar{\Delta}_{\boldsymbol{e}}\right\} \tag{MIMA}$$

i.e., the normalization constraint was $\sum_{j\in\mathcal{Z}} d_j = 1$.

Recall that, from Equation (3), this is equivalent to optimizing over the cost function $\frac{\boldsymbol{c}^T\boldsymbol{d}}{\boldsymbol{e}^T\boldsymbol{d}_{\mathcal{Z}}}$. Hence, the convexity constraint tends to encourage solutions whose supports $\mathcal{S} = Supp(\boldsymbol{d}_{\mathcal{Z}})$ have fewer variables. When there are fewer variables with nonnegative values there is a greater likelihood that the direction will be integral. Example 1 gives an illustration of this convexity constraint.

Example 1 Let us give a simple example of what an augmentation problem can be. Let

$$
\begin{array}{rc}
j = & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\boldsymbol{A} = & \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \\
\boldsymbol{c} = & \begin{bmatrix} 3 & 4 & 2 & 2 & 1 & 1 & 1 \end{bmatrix} \\
\boldsymbol{d}^1 = & \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \end{bmatrix} \\
\boldsymbol{d}^2 = & \begin{bmatrix} -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}
\end{array}
$$

In this case, $\mathcal{S}_1 = Supp(\boldsymbol{d}^1_{\mathcal{Z}}) = \{3,4\}$, $\mathcal{S}_2 = Supp(\boldsymbol{d}^2_{\mathcal{Z}}) = \{5,6,7\}$, and the corresponding exiting set for both directions is $\mathcal{R} = \{1,2\}$. Here, with the normalization constraint $\boldsymbol{e}^T\boldsymbol{d}_{\mathcal{Z}} = 1$, direction $\boldsymbol{d}^1$ is better than $\boldsymbol{d}^2$ since $\boldsymbol{c}^T\boldsymbol{d}^1 = -\frac{3}{2} < -\frac{4}{3} = \boldsymbol{c}^T\boldsymbol{d}^2$.

Note also that following $\boldsymbol{d}^i$, $i \in \{1,2\}$, leads to $\boldsymbol{x}^i = \boldsymbol{x}^0 + r(\boldsymbol{d}^i)\boldsymbol{d}^i$. The corresponding change in the objective function is thus $\boldsymbol{c}^T(r(\boldsymbol{d}^i)\boldsymbol{d}^i) = r(\boldsymbol{d}^i)\boldsymbol{c}^T\boldsymbol{d}^i$. Since $r(\boldsymbol{d}^1) = 2$ and $r(\boldsymbol{d}^2) = 3$, the modifications are -3 for $\boldsymbol{d}^1$ and -4 for $\boldsymbol{d}^2$. The best normalized solution is not necessarily the one yielding the largest improvement after the step has been computed. Here, $\bar{\boldsymbol{w}} = \boldsymbol{e}$ tends to minimize $|Supp(\boldsymbol{d}_{\mathcal{Z}})|$ ($|\mathcal{S}_1| < |\mathcal{S}_2|$). However, as in any gradient-descent algorithm, only the reduced cost is available prior to making the decision, so these steps are not known in advance.

The use of this incoming mean normalization was initially intended to encourage integrality (see [14]): the fewer the variables entering the basis, the greater the likelihood that they are column-disjoint, and hence that the direction is integral. Proposition 14 below gives a theoretical result that simplifies the algorithm when $\bar{\boldsymbol{w}} = \boldsymbol{e}$. For this proposition, we will index sets $\mathcal{P}$, $\mathcal{C}$, and $\mathcal{I}$ on k (the index of the solution in the augmenting sequence) since they formally depend on the current solution.

Proposition 14 *Let $\boldsymbol{x}^k$ be an integer solution of SPP and $(\mathcal{P}^k, \mathcal{C}^k, \mathcal{I}^k)$ the corresponding partition of the variables. Suppose that $z^\star_{\mathrm{RP}^k} \geq 0$ and let $\boldsymbol{d}^k_{\mathcal{Z}}$ be an extreme solution of* $\textsc{Mima}^k$ *that is column-disjoint and has the lowest cost among the column-disjoint solutions. Let $\boldsymbol{d}^k = \boldsymbol{T}\boldsymbol{d}^k_{\mathcal{Z}}$. One of the following is true:*

$$\text{(i) } \boldsymbol{x}^k \textit{ is optimal for SPP, or } \text{(ii) } z^\star_{\mathrm{RP}^{k+1}} \geq 0,$$

where $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + r(\boldsymbol{d}^k)\boldsymbol{d}^k$ and RP^{k+1} is the corresponding reduced problem.

This proposition can be rephrased as: if, at step k, no compatible column has a negative reduced cost, then this is also true at step $k+1$.

Proof. Define $\boldsymbol{x}^k$, $\mathcal{P}^k$, $\mathcal{C}^k$, $\mathcal{I}^k$, $\boldsymbol{d}^k_{\mathcal{Z}}$, $\boldsymbol{d}^k$, and $\boldsymbol{x}^{k+1}$ as in the proposition and assume that $\boldsymbol{x}^k$ is not optimal for SPP. Note that $Supp(\boldsymbol{d}^k) = \mathcal{S}^k \cup \mathcal{R}^k$. We will show that $z^\star_{\mathrm{RP}^{k+1}} \geq 0$ by contradiction.

Let $j_0 \in \mathcal{P}^{k+1}$ be the index of a negative-reduced-cost compatible column at $\boldsymbol{x}^{k+1}$: $\bar{c}^{k+1}_{j_0} < 0$. As defined in Equation (12), $\boldsymbol{d}^{k+1} = \boldsymbol{\delta}^{j_0}$ is thus an augmenting integral direction at $\boldsymbol{x}^{k+1}$. Let $\mathcal{R}_{j_0} \subseteq \mathcal{P}^{k+1}$ be such that $\boldsymbol{A}_{\cdot j_0} = \sum_{i\in\mathcal{R}_{j_0}} \boldsymbol{A}_{\cdot i}$ (as defined in Lemma 11). We will distinguish three cases that correspond to the part of the partition $(\mathcal{P}^k, \mathcal{C}^k, \mathcal{I}^k)$ to which j_0 belonged.

(i) $\underline{j_0 \in \mathcal{P}^k}$. Then j_0 left the working basis between k and $k+1$, i.e., $j_0 \in \mathcal{R}^k$. Therefore, $\mathcal{R}_{j_0}$ entered the working basis at step k and $\mathcal{R}_{j_0} \subseteq \mathcal{S}^k$. Hence, $Supp(\boldsymbol{\delta}^{j_0}) \subseteq Supp(\boldsymbol{d}^k)$. Since $\boldsymbol{d}^k \in Ext\left(\Delta^k_{\boldsymbol{e}}\right)$, $\boldsymbol{d}^k = -\boldsymbol{\delta}^{j_0}$, and $\boldsymbol{x}^{k+1}$ is a worse solution than $\boldsymbol{x}^k$, which is impossible.

(ii) $j_0 \in \mathcal{C}^k$. If j_0 was already compatible and yielded no improvement at $\boldsymbol{x}^k$ ($z^\star_{\mathrm{RP}^k} \geq 0$) it obviously also yields no improvement at $k+1$. This contradicts $\bar{c}^{k+1}_{j_0} < 0$.

(iii) $j_0 \in \mathcal{I}^k$. j_0 switched from incompatible to compatible. Here, we will conduct a more detailed analysis of the different sets. Without loss of generality, suppose that there exists no variable apart from those of the supports of directions $\boldsymbol{d}^k$ and $\boldsymbol{d}^{k+1} = \boldsymbol{\delta}^{j_0}$.

Recall that $\boldsymbol{x}^k$, $\boldsymbol{x}^{k+1}$, and $\boldsymbol{x}^{k+2}$ are integer, so going from one to the other is equivalent to exchanging columns between $\mathcal{P}$ and $\mathcal{Z} = \mathcal{C} \cup \mathcal{I}$. We summarize the different steps in the following chart:

	$\mathcal{P}$	$\mathcal{C}$	$\mathcal{I}$	Entering $\mathcal{P}$	Exiting $\mathcal{P}$
$\boldsymbol{x}^k$:	$\mathcal{R}^k, \mathcal{R}_2^{k+1}$		$\mathcal{S}_1^k, \mathcal{S}_2^k, j_0$		
				$\mathcal{S}^k = \mathcal{S}_1^k \cup \mathcal{S}_2^k$	$\mathcal{R}^k$
$\boldsymbol{x}^{k+1}$:	$\mathcal{S}_1^k, \mathcal{S}_2^k, \mathcal{R}_2^{k+1}$	j_0	$\mathcal{R}^k$		
				$\mathcal{S}^{k+1} = \{j_0\}$	$\mathcal{R}^{k+1} = \mathcal{S}_2^k \cup \mathcal{R}_2^{k+1}$
$\boldsymbol{x}^{k+2}$:	$\mathcal{S}_1^k, j_0$		$\mathcal{R}^k, \mathcal{S}_2^k, \mathcal{R}_2^{k+1}$		

At $\boldsymbol{x}^k$, the augmenting direction chosen is $\boldsymbol{d}^k = \left(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\right)/|\mathcal{S}^k|$. Consider now the feasible normalized direction that *could have been* followed and that leads directly from $\boldsymbol{x}^k$ to $\boldsymbol{x}^{k+2}$: $\boldsymbol{d}^\star = \left(\boldsymbol{x}^{k+2} - \boldsymbol{x}^k\right)/|\mathcal{S}^\star|$, where $\mathcal{S}^\star = Supp(\boldsymbol{x}^{k+2} - \boldsymbol{x}^k)$. Then,

$$\mathcal{S}^\star = \mathcal{S}_1^k \cup \{j_0\} \text{ and } \forall i \in \{1, \ldots n\}, d_i^\star = \begin{cases} -1/|\mathcal{S}^\star| & \text{if } i \in \mathcal{R}^\star \\ 1/|\mathcal{S}^\star| & \text{if } i \in \mathcal{S}^\star \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathcal{R}^\star = \mathcal{R}^k \cup \mathcal{R}_2^{k+1}$ is the set of columns that leaves the working basis when we go from $\boldsymbol{x}^k$ to $\boldsymbol{x}^{k+2}$. Let us compute the cost of this direction:

$$\begin{aligned} \bar{\boldsymbol{c}}^T \boldsymbol{d}^\star_{\mathcal{Z}} = \boldsymbol{c}^T \boldsymbol{d}^\star = \frac{\boldsymbol{c}^T(\boldsymbol{x}^{k+2} - \boldsymbol{x}^0)}{|\mathcal{S}^\star|} &= \frac{\boldsymbol{c}^T(\boldsymbol{x}^{k+2} - \boldsymbol{x}^{k+1})}{|\mathcal{S}^\star|} + \frac{\boldsymbol{c}^T(\boldsymbol{x}^{k+1} - \boldsymbol{x}^0)}{|\mathcal{S}^\star|} \\ &= \frac{|\mathcal{S}^k|}{|\mathcal{S}^\star|}\boldsymbol{c}^T\boldsymbol{d}^k + \frac{|\mathcal{S}^{k+1}|}{|\mathcal{S}^\star|}\boldsymbol{c}^T\boldsymbol{d}^{k+1} \\ &= \frac{|\mathcal{S}_1^k| + |\mathcal{S}_2^k|}{|\mathcal{S}_1^k| + 1}\boldsymbol{c}^T\boldsymbol{d}^k + \frac{1}{|\mathcal{S}_1^k| + 1}\boldsymbol{c}^T\boldsymbol{d}^{k+1} \end{aligned}$$

Note here that $\mathcal{S}_2^k \neq \emptyset$, otherwise $\mathcal{R}^k \cap \mathcal{R}^{k+1} = \emptyset$ and $j_0 \in \mathcal{C}^k$ was compatible at $\boldsymbol{x}^k$. Since $\boldsymbol{c}^T\boldsymbol{d}^k < 0$ and $\boldsymbol{c}^T\boldsymbol{d}^{k+1} < 0$, we have $\bar{\boldsymbol{c}}^T\boldsymbol{d}^\star_{\mathcal{Z}} < \boldsymbol{c}^T\boldsymbol{d}^k = \bar{\boldsymbol{c}}^T\boldsymbol{d}^k_{\mathcal{Z}}$. Hence, $\boldsymbol{d}^\star$ is a feasible normalized direction at $\boldsymbol{x}^k$ that leads to an integer solution, and which has a lower reduced cost than $\boldsymbol{d}^k$. This contradicts the definition of $\boldsymbol{d}^k$ (the most augmenting integer direction at $\boldsymbol{x}^k$), which completes the proof.

□

Therefore, in the special case of Mima, when no augmenting compatible pivot can be found, we no longer need to consider the compatible variables. In this case, if, at some point in the execution of the algorithm, $z^\star_{\mathrm{RP}} \geq 0$ (line 3 of Algorithm 1), RP no longer needs to be solved and lines 3–5 are never read again.

3.2 Maximum mean augmentation (Mma)

Another classical normalization constraint corresponds to the *maximum mean augmentation* (Mma) problem: $\boldsymbol{w} = \boldsymbol{\epsilon} = (-\boldsymbol{e}_{\mathcal{P}}, \boldsymbol{e}_{\mathcal{Z}})$. We will not present a full analysis of this constraint since it has already been studied (see [5] or [20] for instance). The constraint is $-\sum_{j\in\mathcal{P}} d_j + \sum_{j\in\mathcal{Z}} d_j = \|\boldsymbol{d}\|_1 = 1$, and the objective function corresponds to $\left(\boldsymbol{\epsilon}^T\boldsymbol{d}\right)/\|\boldsymbol{d}\|_1$. The Mma program is

$$z^\star_{\text{Mima}} = \min\left\{\bar{\boldsymbol{c}}^T\boldsymbol{d}_{\mathcal{Z}} \mid \boldsymbol{d}_{\mathcal{Z}} \in \bar{\Delta}_{\bar{\boldsymbol{\epsilon}}}\right\} \qquad \text{(Mma)}$$

where $\bar{\boldsymbol{\epsilon}} = \boldsymbol{T}^T\boldsymbol{\epsilon}$.

This problem originally comes from the maximum mean cycle cancellation algorithm proposed by Goldberg and Tarjan for the minimum-cost maxium-flow problem in a directed network (see [21] for more details). Its canonical extension to generic 0–1 programs is a classical example of augmentation subproblems. For more details on its properties and its influence on the behavior of a primal algorithm see the review of Spille et al. [5]. They show that this convexity constraint guarantees a pseudo-polynomial bound on the number of directions that must be found to reach optimality. Note also that in the particular case of the SPP (and of 0–1 programs in general), MMA is equivalent to the *maximum ratio augmentation* for which each term w_j of the normalization constraint is computed as the distance from its current value x_j to its furthest bound (see Schulz and Weismantel [20] for more details).

Example 2 Let us continue with Example 1, but with the convexity constraint $\sum_j \epsilon_j d_j$, $\epsilon_j = -1$ if $j \in \{1,2\}$ and $+1$ if $j \in \{3,4,5,6,7\}$. We have

$$\boldsymbol{d}^1 = \begin{bmatrix} -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{d}^2 = \begin{bmatrix} -\frac{1}{5} & -\frac{1}{5} & 0 & 0 & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}$$

and $\boldsymbol{c}^T\boldsymbol{d}^1 = -3/4 > -4/5 = \boldsymbol{c}^T\boldsymbol{d}^2$. Hence, this normalization constraint would select $\boldsymbol{d}^2$ rather than $\boldsymbol{d}^1$, unlike MIMA.

The weight vector $\boldsymbol{\epsilon}$ measures the number of columns in $Supp(\boldsymbol{d})$, whereas $\bar{\boldsymbol{w}} = \boldsymbol{e}$ in MIMA measured the size of $Supp(\boldsymbol{d}_{\mathcal{Z}})$.

3.3 Encouraging integrality via the normalization constraint

This section is a first attempt to propose new ways of encouraging integrality through specific normalization weights. We will not give an exhaustive description of the possibilities of this method. We will indicate the kind of constraints that can be used to encourage integral solutions and give a foretaste of their computational influence on ISUD (Section 4). As mentioned in the Introduction, a follow-up paper will investigate in more detail the ways to influence integrality and will include more computational results.

We will consider the situation where a set of leaving variables $\mathcal{R}$ can be exchanged with two sets of entering variables $\mathcal{S}_1$ and $\mathcal{S}_2$. Furthermore, we will assume that $\mathcal{S}_1$ is column-disjoint and $\mathcal{S}_2$ is not. The corresponding directions are denoted $\boldsymbol{d}^1$ and $\boldsymbol{d}^2$. We want to define a constraint $\boldsymbol{w}^T\boldsymbol{d} = 1$ such that R-MNA$^{\bar{\boldsymbol{w}}}$ tends to select $\mathcal{S}_1$ rather than $\mathcal{S}_2$. To do this, we will use parameters that will allow us to encourage column-disjoint solutions, namely the norm and the incompatibility degree of a column.

3.3.1 Column norm

Definition 4 The *norm* of a column $\boldsymbol{A}_{\cdot j}$ is $n_j = \|\boldsymbol{A}_{\cdot j}\|_1 = \sum_{i\in\{1,\dots m\}} |A_{ij}|$. Since $\boldsymbol{A}$ is binary, the norm of a column is just the number of its nonzero components. Moreover, for any set $\mathcal{X}$ of columns, $N_{\mathcal{X}} = \sum_{j\in\mathcal{X}} n_j$.

As the following proposition shows, the sum of the norms of the columns of $Supp(\boldsymbol{d})$ tends to be smaller for disjoint combinations.

Proposition 15 *Let $\boldsymbol{d}$ be a feasible direction and $\mathcal{S} = Supp(\boldsymbol{d}_{\mathcal{Z}})$ the set of entering variables. Then $N_{\mathcal{S}} \geq N_{\mathcal{R}}$, and $N_{\mathcal{S}} = N_{\mathcal{R}}$ if and only if $\mathcal{S}$ is column-disjoint.*

Proof. Given a column $\boldsymbol{A}_{\cdot j}$, let $\Omega_j = \{i | A_{ij} = 1\}$ be the set of rows covered by $\boldsymbol{A}_{\cdot j}$. Then $n_j = Card(\Omega_j)$. $\mathcal{R}$ is column-disjoint, i.e., $\cup_{j\in\mathcal{R}}\Omega_j$ is a disjoint union ($\forall i,j \in \mathcal{R}, i \neq j, \Omega_i \cup \Omega_j = \emptyset$), so $N_{\mathcal{R}} = \sum_{j\in\mathcal{R}} Card(\Omega_j) = Card(\cup_{j\in\mathcal{R}}\Omega_j)$. Hence, $\sum_{j\in\mathcal{S}} Card(\Omega_j) \geq Card(\cup_{j\in\mathcal{S}}\Omega_j) = N_{\mathcal{R}}$ and this inequality is an equality if and only if $\cup_{j\in\mathcal{S}}\Omega_j$ is a disjoint union, i.e., if $\mathcal{S}$ is column-disjoint. □

This result suggests the idea of using the norms of the columns as normalization weights: $w_j = n_j$, $j \in \mathcal{Z}$ tends to encourage combinations of disjoint columns. Consider the aforementioned example of two possible directions $\boldsymbol{d}^1$ and $\boldsymbol{d}^2$ such that $\mathcal{R} = Supp(\boldsymbol{d}^1_{\mathcal{P}}) = Supp(\boldsymbol{d}^2_{\mathcal{P}})$. Suppose that these directions respectively lead to two adjacent vertices $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ with the same cost $\boldsymbol{c}^T\boldsymbol{x}^1 = \boldsymbol{c}^T\boldsymbol{x}^2$. Since $\mathcal{S}^1$ is column-disjoint, but $\mathcal{S}^2$ is not, by Proposition 15, the corresponding reduced costs are

$$\boldsymbol{c}^T\boldsymbol{d}^1 = \frac{\boldsymbol{c}^T\left(\boldsymbol{x}^1 - \boldsymbol{x}^0\right)}{N_{\mathcal{S}^1}} < \frac{\boldsymbol{c}^T\left(\boldsymbol{x}^2 - \boldsymbol{x}^0\right)}{N_{\mathcal{S}^2}} = \boldsymbol{c}^T\boldsymbol{d}^2,$$

and the algorithm will select the column-disjoint $\boldsymbol{d}^1_{\mathcal{Z}}$ rather than the non-column-disjoint $\boldsymbol{d}^2_{\mathcal{Z}}$.

3.3.2 Incompatibility degree

Definition 5 The *incompatibility degree* ι_j of a column $\boldsymbol{A}_{\cdot j}$ is the number of nonzero components of $\bar{\boldsymbol{A}}_{\cdot j}$.

Note that compatible columns as defined in Definition 3 are exactly the columns with incompatibility degree $\iota_j = 0$. In the same way as the norm, the incompatibility degree measures the integrality of a solution.

Proposition 16 *If the entering set $\mathcal{S}$ is column-disjoint, then each nonzero row of $\bar{\boldsymbol{A}}_{\mathcal{N}\mathcal{S}}$ has exactly one component equal to 1 and one component equal to -1, all others being zero.*

To prove this proposition, we need the following lemma.

Lemma 17 *Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two binary matrices. Then:*

(i) $\boldsymbol{X}$ is column-disjoint $\Leftrightarrow$ $\boldsymbol{X}^T\boldsymbol{X}$ is diagonal;

(ii) $\boldsymbol{X}$ and $\boldsymbol{Y}$ are binary and column-disjoint $\Rightarrow$ $\boldsymbol{X}\boldsymbol{Y}$ is column-disjoint.

Proof. We will prove these two points separately.

(i) is a consequence of the definition of column-disjoint and the nonnegativity of the entries of $\boldsymbol{A}$.

(ii) Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two column-disjoint binary matrices. By (i), $\boldsymbol{D} = \boldsymbol{X}^T\boldsymbol{X}$ is diagonal, so $(\boldsymbol{X}\boldsymbol{Y})^T(\boldsymbol{X}\boldsymbol{Y}) = \boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y}$. Therefore, $\left[\boldsymbol{Y}^T\boldsymbol{D}\boldsymbol{Y}\right]_{ij} = \sum_k \left(D_{kk}Y_{ki}Y_{kj}\right)$. According to the definition of column-disjoint, if $i \neq j$, for all k, $Y_{ki}Y_{kj} = 0$, so $(\boldsymbol{X}\boldsymbol{Y})^T(\boldsymbol{X}\boldsymbol{Y})$ is diagonal. Since $\boldsymbol{X}$ and $\boldsymbol{Y}$ are binary, by (i), $\boldsymbol{X}\boldsymbol{Y}$ is column-disjoint.

□

Proof. *(Proof of Proposition 16)* Let $\boldsymbol{d} \in Ext\left(\Delta_{\boldsymbol{w}}\right)$ be an extreme feasible direction at $\boldsymbol{x}^0$ and $\mathcal{S} = Supp(\boldsymbol{d}_{\mathcal{Z}})$ the set of entering columns. From Equation (6), we have $\bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{S}} = -\boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{P}}\boldsymbol{A}_{\mathcal{P}\mathcal{S}} + \boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{S}}$. Since by assumption $\mathcal{S}$ is column-disjoint, $\boldsymbol{A}_{\mathcal{P}\mathcal{S}}$ and $\boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{S}}$ are column-disjoint. Note that the condition $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{e}$ implies that every row of $\boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{P}}$ contains one and only one positive component equal to 1, all other components being zero. Hence, $\boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{P}}$ is also column-disjoint.

By Lemma 17, we have that $\boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{P}}\boldsymbol{A}_{\mathcal{P}\mathcal{S}}$ is column-disjoint. Consequently, since $\bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{S}} = -\boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{P}}\boldsymbol{A}_{\mathcal{P}\mathcal{S}} + \boldsymbol{A}_{\bar{\mathcal{P}}\mathcal{S}}$, a row of $\bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{S}}$ has at most two nonzero components. Furthermore, by Corollary 8, the linear constraints $\bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{S}}\boldsymbol{d}_{\mathcal{N}} = \boldsymbol{0}$ yield $\sum_{j\in\mathcal{S}} \bar{\boldsymbol{A}}_{\bar{\mathcal{P}}j} = 0$. Hence, a nonzero row of $\bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{S}}$ must have exactly two nonzero components: 1 and -1. □

Proposition 16 shows that fractional solutions may have several nonzero components in every row of the matrix $\bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{S}}$, while integer solutions have only two. Note that $\sum_{j\in\mathcal{S}} \iota_j$ is the number of nonzero components in the matrix $\bar{\boldsymbol{A}}_{\bar{\mathcal{P}}\mathcal{S}}$. Hence, like $\sum_{j\in\mathcal{S}} n_j$, the quantity $\sum_{j\in\mathcal{S}} \iota_j$ tends to be smaller for integer solutions. These properties naturally lead to the idea of using $\bar{w}_j = \iota_j$ (or $\bar{w}_j = \iota_j|\bar{c}_j|$) as normalization weights.

4 Numerical results

The results presented in this section establish a proof of concept for our theoretical results: we show that a judicious choice of the convexity constraint plays an important role in encouraging integral solutions in ISUD.

4.1 Methodology

Instances The instances presented here are those used by Zaghrouti et al. [14] in their seminal paper on ISUD: SPPAA01, VCS1200, and VCS1600. SPPAA01 is a small flight assignment problem (823 constraints, 8,904 variables) with an average of 9 nonzero entries per column. VCS1200 is a medium-size bus driver scheduling problem (1200 constraints, 133,000 variables), and VCS1600 is a large vehicle crew scheduling problem (1,600 constraints, 571,000 variables); both have an average of 40 nonzeros per column. These numbers of nonzeros are typical of aircrew and bus driver scheduling problems and do not vary with the number of constraints. The nonzeros correspond to the number of tasks per duty. The optimal solutions of the problems are known: CPLEX has solved SPPAA01, and GENCOL[1] has solved VCS1200 and VCS1600. Note that within a time limit of 10 hours, CPLEX cannot find a feasible solution for VCS1200 or VCS1600.

Initial solutions In scheduling applications, a column generally represents a sequence of tasks performed by an employee. Define the *primal information* of a solution as the percentage of consecutive tasks in that solution that are also consecutive in the optimal schedule. In practice, this quantity is not known precisely a priori, but the following two observations hold. First, crew do not often change vehicles during their duties, and their schedules therefore have many consecutive tasks in common with those of the vehicle routes. Second, when the schedule is updated, e.g., because of unforeseen events, the reoptimized schedule usually has many pieces in common with the original schedule (companies generally add penalties in the objective function to discourage changes). Thus, many consecutive tasks from the initial *paths* (vehicle routes or the original schedule) remain consecutive in the optimal schedule. In bus driver scheduling problems, the initial solution that follows the bus routes typically contains 90% of primal information (VCS1200, VCS1600). In aircrew scheduling, the figure is generally around 75% [14] (SPPAA01).

Therefore, we perturb the (known) optimal solutions to generate initial solutions that contain a similar level of primal information to that seen in practice. These solutions are generally infeasible, so we add artificial columns with a high cost, as is done for the schedules that follow the bus/airplane routes. In our perturbation method, the input parameter π is the percentage of columns of the optimal solution that will appear in the new initial solution. For SPPAA01, we generated initial solutions for $\pi = 10\%$, 15%, 20%, and 35%; for VCS1200 and VCS1600, we used $\pi = 20\%$, 35%, and 50%. These parameters were chosen so that the resulting primal information (given in Table 1) is consistent with the typical values. The initial gaps range from 50% to 80%, depending on the instance.

Table 1: Percentage of primal information in the initial solutions of the benchmark.

			Primal Information				
Instance	m	n	$\pi = 10\%$	$\pi = 15\%$	$\pi = 20\%$	$\pi = 35\%$	$\pi = 50\%$
SPPAA01	803	8,904	71.5	75.6	80.0	86.2	-
VCS1200	1,200	133,000	-	-	87.6	91.1	93.9
VCS1600	1,600	570,000	-	-	86.8	90.9	93.9

[1]GENCOL is commercial software developed at the GERAD research center and now owned by the AD OPT company, a division of KRONOS.

Normalization weights The tests were conducted for the following normalization weights:

- MIMA, where $\bar{w}_j = 1$ for all $j \in \mathcal{Z}$;
- MMA, where $w_j = -1$ if $j \in \mathcal{P}$, 1 if $j \in \mathcal{Z}$;
- NORM, where $\bar{w}_j = n_j$ for all $j \in \mathcal{Z}$;
- DEG, where $\bar{w}_j = \iota_j$ for all $j \in \mathcal{Z}$.

Combinations or modifications of these weights could of course be tested; however, for this first study, we present only these four possibilities.

Branching technique In the example shown here, a nonexhaustive branching technique is used (line 14 of Algorithm 1): whenever a solution $\boldsymbol{d}^\star$ is not column-disjoint, all the variables of $\mathcal{S}^\star = Supp(\boldsymbol{d}^\star_{\mathcal{Z}})$ are set to zero in R-MNA$^{\boldsymbol{w}}$ until an augmentation is performed.

4.2 Results

All the tests were performed on a Linux PC with 8 processors of 3.4 GHz. Table 2 gives the results for SPPAA01. For each perturbation parameter $\pi \in \{10, 15, 20, 25\}$, 10 different instances (initial solutions and corresponding artificial columns) are generated. Tables 3 and 4 give the results for VCS1200 and VCS1600. For each of these instances, 10 different initial solutions were generated for each $\pi \in \{20, 35, 50\}$. Column $\boldsymbol{w}$ indicates the normalization weights used; BEST is the best of the four on each instance, i.e., the one with the smallest gap, and in the event of a tie, the shortest computational time to reach the best solution. The first columns give the quality of the solution, showing how many instances of the 10 were solved to optimality (0%), with a gap $\leq 2\%$, and with a gap $> 2\%$; the mean gap is given in column MEAN. The next columns give the total computational time (ISUD), the time to reach the best solution found (BEST), and the average time per augmentation, i.e., from $\boldsymbol{x}^k$ to $\boldsymbol{x}^{k+1}$ (**AUG**). The final columns give the mean number of augmentations K performed to reach the best solution, the percentage of directions that were disjoint even before any variable was fixed (DISJ), and the mean size of $|S| = Supp(d)$ for disjoint ($|\mathcal{S}|^D$) and nondisjoint ($|\mathcal{S}|^{ND}$) directions. Note that, while the other mean values are computed over all executions of the algorithm, the mean number of augmentations K is based on the instances for which the final gap is $\leq 2\%$. Large gaps often mean a much smaller number of iterations, and taking these instances into account would distort the mean.

Almost all the instances were solved to optimality by at least one of the algorithms (94/100), but no algorithm is perfect.

On SPPAA01 (Table 2), the performances of the various normalization weights vary, and some get much worse when the initial primal information decreases ($\pi = 10\%$). However, the best solution over the different normalization constraints is always within 2% of the optimal solution, and it is nearly always optimal. The computational times are similar for the different normalization weights, as are the time to reach the best solution (BEST) and the mean time per augmentation (**AUG**). The same holds for the overall number of iterations K and the size of the disjoint combinations $|\mathcal{S}|^D$. However, when the algorithm performs well, the percentage of disjoint solutions (DISJ) is higher and the size of the nondisjoint supports $|\mathcal{S}|^{ND}$ tends to be significantly smaller than in the other cases. For instance, when $\pi = 20\%$, the best weights are NORM and DEG, for which the mean sizes of nondisjoint combinations are respectively 50 and 53, against 67 and 77 for the other constraints. This is a consequence of the branching technique used: fixing more variables to zero increases the risk of finding no improving feasible solutions (nonexhaustive branching).

The results for VCS1200 and VCS1600 are given in Tables 3 and 4; the conclusions that can be drawn from these tables are quite similar. First, ISUD performs well on the instances that CPLEX cannot solve. Nearly all the instances are solved to optimality, and when ISUD fails, it generally fails regardless of the normalization weights (the best-of-four also fails). Using the column norm (NORM) as the normalization vector is significantly faster but yields slightly worse results: only 53 instances were solved to optimality, against at least 56 for the other weights. Note that the mean computational time is not biased by the lower performance since the mean time is computed only for the instances for which the best solution is within

Table 2: Evaluation of ISUD on the aircrew scheduling instance SPPAA01.

π	$\boldsymbol{w}$	GAP 0%	GAP $\leq 2\%$	GAP $> 2\%$	GAP MEAN	Time (s) ISUD	Time (s) Best	Time (s) **AUG**	**AUG** K	**AUG** DISJ	**AUG** $\lvert\mathcal{S}\rvert^D$	**AUG** $\lvert\mathcal{S}\rvert^{ND}$
10%	MIMA	3	3	4	141.3%	11.4	8.7	0.27	38	41%	4.4	112
	MMA	4	3	3	73.8%	11.8	9.4	0.26	38	46%	4.5	95
	NORM	2	3	5	87.3%	13.8	10.7	0.29	41	34%	4.5	78
	DEG	7	2	1	31.1%	12.5	10.1	0.26	39	47%	4.6	78
	BEST	**8**	**2**	**0**	**0.1%**	**12.3**	**10.3**	-	-	-	-	-
15%	MIMA	2	4	4	30.9%	15	12.2	0.30	44	36%	4	87
	MMA	8	1	1	4.4%	12.6	10	0.24	42	44%	4	69
	NORM	7	0	3	72.1%	11.5	8.5	0.23	38	42%	3.8	97
	DEG	8	1	1	29.2%	11.5	9.1	0.21	43	53%	3.9	74
	BEST	**10**	**0**	**0**	**0%**	**10.6**	**8**	-	-	-	-	-
20%	MIMA	7	2	1	20.1%	9.9	7	0.17	42	51%	3.3	67
	MMA	6	2	2	20.9%	9.6	6.4	0.16	40	52%	3.4	77
	NORM	7	3	0	0.1%	10.3	7.8	0.18	42	49%	3.5	50
	DEG	7	2	1	3.5%	10.8	7.9	0.17	45	54%	3.4	53
	BEST	**9**	**1**	**0**	**0%**	**9.4**	**7**	-	-	-	-	-
35%	MIMA	9	1	0	0%	7	4.4	0.13	34	56%	2.9	60
	MMA	9	1	0	0%	7.1	4.5	0.13	35	56%	2.9	57
	NORM	10	0	0	0%	7.4	4.5	0.13	35	55%	2.9	58
	DEG	8	2	0	0.1%	7.4	4.6	0.13	35	55%	2.9	46
	BEST	**10**	**0**	**0**	**0%**	**7.1**	**4.4**	-	-	-	-	-

Table 3: Evaluation of ISUD on the bus driver scheduling instance VCS1200.

π	$\boldsymbol{w}$	GAP 0%	GAP $\leq 2\%$	GAP $> 2\%$	GAP MEAN	Time (s) ISUD	Time (s) Best	Time (s) **AUG**	**AUG** K	**AUG** DISJ	**AUG** $\lvert\mathcal{S}\rvert^D$	**AUG** $\lvert\mathcal{S}\rvert^{ND}$
20%	MIMA	8	0	2	9.7%	93	58	3	21	56%	3	296
	MMA	9	0	1	5.5%	99	65	3.3	21	57%	3	281
	NORM	8	0	2	9.7%	87	55	2.9	21	56%	3	300
	DEG	8	0	2	9.7%	96	61	3.1	22	57%	3	289
	BEST	**9**	**0**	**1**	**5.5%**	**93**	**62**	-	-	-	-	-
35%	MIMA	9	0	1	1.2%	86	46	2.5	19	52%	2	248
	MMA	9	0	1	1.2%	87	47	2.6	18	52%	2	250
	NORM	8	0	2	3.5%	79	40	2.3	18	50%	2	280
	DEG	9	0	1	1.2%	87	47	2.6	19	52%	2	250
	BEST	**9**	**0**	**1**	**1.2%**	**83**	**43**	-	-	-	-	-
50%	MIMA	10	0	0	0%	71	28	1.9	15	48%	2	233
	MMA	10	0	0	0%	72	28	1.9	15	48%	2	232
	NORM	10	0	0	0%	69	27	1.8	15	49%	2	226
	DEG	10	0	0	0%	72	28	1.9	15	49%	2	233
	BEST	**10**	**0**	**0**	**0%**	**70**	**27**	-	-	-	-	-

2% of the optimum. Similarly to the case of SPPAA01, the algorithm performs better but more slowly when the size of the nondisjoint combinations is smaller. All the other parameters do not vary significantly for the different normalizations.

Since the performances of the four normalizations were similar for VCS1200 and VCS1600, we drew the performance diagram of the five weights (including BEST) only for instance SPPAA01; see Figure 4. The instances considered are the 40 instances generated from SPPAA01 (10 for each $\pi \in \{10, 15, 20, 35\}$). The figure shows the percentage of instances solved to within 2% of the optimal solution against the computational time. This diagram confirms that no weight vector is significantly better than any other. Interestingly, the

Table 4: Evaluation of ISUD on the bus driver scheduling instance VCS1600.

π	$\boldsymbol{w}$	GAP 0%	GAP $\leq 2\%$	GAP $> 2\%$	GAP MEAN	Time (s) ISUD	Time (s) Best	Time (s) **AUG**	**AUG** K	**AUG** DISJ	**AUG** $\lvert\mathcal{S}\rvert^D$	**AUG** $\lvert\mathcal{S}\rvert^{ND}$
20%	MIMA	10	0	0	0%	1278	450	17.1	26	69%	3	463
	MMA	10	0	0	0%	1238	451	17	27	69%	3	483
	NORM	9	0	1	3.7%	892	395	15.5	26	66%	3	546
	DEG	10	0	0	0%	1130	469	17.3	27	70%	3	467
	BEST	**10**	**0**	**0**	**0%**	**1045**	**414**	-	-	-	-	-
35%	MIMA	10	0	0	0%	1232	312	12.9	24	68%	2	441
	MMA	10	0	0	0%	1159	304	12.6	24	68%	2	446
	NORM	9	0	1	1.1%	866	261	10.9	24	68%	2	468
	DEG	10	0	0	0%	1009	311	12.7	24	69%	2	430
	BEST	**10**	**0**	**0**	**0%**	**924**	**277**	-	-	-	-	-
50%	MIMA	9	0	1	1.1%	1084	189	11	17	60%	2	485
	MMA	9	0	1	1.1%	1048	189	11	17	60%	2	488
	NORM	9	0	1	1.4%	756	159	9.2	17	61%	2	507
	DEG	9	0	1	1.1%	868	182	10.6	17	60%	2	477
	BEST	**9**	**0**	**1**	**1.1%**	**767**	**168**	-	-	-	-	-

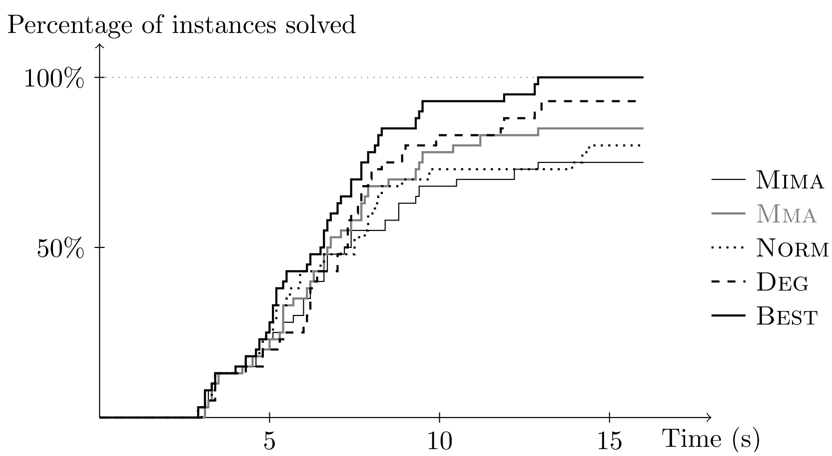

Figure 4: Performance diagram of ISUD on SPPAA01. An instance is considered solved if the gap is $\leq 2\%$.

new weights introduced in this paper outperform those used until now (MIMA). This shows that the choice of the normalization vector has the potential to encourage integral solutions in ISUD.

5 Conclusions

We have studied the potential of the normalization-weight vector to encourage integral directions in ISUD. We have strengthened and extended the theoretical base laid in [14], and we have underlined the bijection between the original and the reduced search spaces. We have proved an algorithmic improvement for the MIMA constraint currently used in ISUD. We compared it to the classical MMA and to two other normalization constraints. We have presented the theoretical background for these new constraints, and we ran numerical tests showing that the choice of the normalization weights is crucial in the performance of the algorithm. Our new constraints outperform the existing one, and nearly all the instances were solved to optimality or quasi-optimality (GAP $\leq 2\%$).

A future paper will present more normalization weights and apply them to a larger benchmark that will include *nonscheduling* problems. We also intend to develop an algorithm that dynamically updates the normalization weights. The updates may be based on the current solution. Alternatively, we may perform updates when the augmentation problem fails to find an integral direction, to encourage integral solutions and to replace or support the current nonexhaustive branching.

References

[1] R.S. Garfinkel, G.L. Nemhauser, The set-partitioning problem: Set covering with equality constraints, Operations Research 17(5) (1969) 848–856. doi:10.1287/opre.17.5.848.

[2] A.N. Letchford, A. Lodi, Primal cutting plane algorithms revisited, Mathematical Methods of Operations Research 56(1) (2002) 67–81. doi:10.1007/s001860200200.

[3] R.E. Gomory, Outline of an algorithm for integer solutions to linear programs, Bulletin of the American Mathematical Society 64(5) (1958) 275–278.

[4] R.E. Gomory, All-integer integer programming algorithm, Industrial Scheduling (1963) 193–206.

[5] B. Spille, R. Weismantel, Primal integer programming, in: K. Aardal, G. Nemhauser, R. Weismantel (Eds.), Discrete Optimization, Vol. 12 of Handbooks in Operations Research and Management Science, Elsevier, Amsterdam, 2005, 245–276.

[6] V. Trubin, On a method of solution of integer linear programming problems of a special kind, Soviet Mathematics Doklady 10 (1969) 1544–1546.

[7] E. Balas, M. Padberg, On the set-covering problem: 2 - An algorithm for set partitioning, Operations Research 23(1) (1975) 74–90. doi:10.1287/opre.23.1.74.

[8] U.-U. Haus, M. Köppe, R. Weismantel, A primal all-integer algorithm based on irreducible solutions, Mathematical Programming 96(2) (2003) 205–246. doi:10.1007/s10107-003-0384-8.

[9] G.L. Thompson, An integral simplex algorithm for solving combinatorial optimization problems, Computational Optimization and Applications 22(3) (2002) 351–367. doi:10.1023/A:1019758821507.

[10] A. Saxena, Set-partitioning via integral simplex method, Unpublished manuscript, OR Group, Carnegie-Mellon University, Pittsburgh (2003).

[11] E. Rönnberg, Contributions within two topics in integer programming: Nurse scheduling and column generation, Ph.D. thesis, Linköping University (2012).

[12] E. Rönnberg, T. Larsson, Column generation in the integral simplex method, European Journal of Operations Research 192(1) (2009) 333–342. doi:http://dx.doi.org/10.1016/j.ejor.2007.09.037.

[13] E. Rönnberg, T. Larsson, All-integer column generation for set partitioning: Basic principles and extensions, European Journal of Operational Research 233(3) (2014) 529–538. doi:http://dx.doi.org/10.1016/j.ejor.2013.08.036.

[14] A. Zaghrouti, F. Soumis, I. Elhallaoui, Integral simplex using decomposition for the set partitioning problem, Operations Research 62 (2) (2014) 435–449. doi:10.1287/opre.2013.1247.

[15] I. Elhallaoui, A. Metrane, G. Desaulniers, F. Soumis, An improved primal simplex algorithm for degenerate linear programs, INFORMS J. Comput. 23(4) (2011) 569–577. doi:10.1287/ijoc.1100.0425.

[16] J. Omer, S. Rosat, V. Raymond, F. Soumis, Improved primal simplex: A more general theoretical framework and an extended experimental analysis, To appear in INFORMS Journal on Computing.

[17] I. Elhallaoui, D. Villeneuve, F. Soumis, G. Desaulniers, Dynamic aggregation of set-partitioning constraints in column generation, Operations Research 53(4) (2005) 632–645. doi:10.1287/opre.1050.0222.

[18] S. Rosat, I. Elhallaoui, F. Soumis, A. Lodi, Integral simplex using decomposition with primal cuts, in: J. Gudmundsson, J. Katajainen (Eds.), Experimental Algorithms, Vol. 8504 of Lecture Notes in Computer Science, Springer International Publishing, 2014, 22–33. doi:10.1007/978-3-319-07959-2_3.

[19] S. Rosat, I. Elhallaoui, F. Soumis, A. Lodi, Primal cuts in the integral simplex using decomposition (to be submitted).

[20] A.S. Schulz, R. Weismantel, The complexity of generic primal algorithms for solving general integer programs, Mathematics of Operations Research 27(4) (2002) 681–692. doi:10.1287/moor.27.4.681.305.

[21] A.V. Goldberg, R.E. Tarjan, Finding minimum-cost circulations by canceling negative cycles, Journal of the Association for Computing Machinery 36(4) (1989) 873–886. doi:10.1145/76359.76368.