

A Service of



Leibniz-Informationszentrum Wirtschaft Leibniz Information Centre for Economics

Glushko, Pavlo; Fábián, Csaba I.; Koberstein, Achim

Article — Published Version An L-shaped method with strengthened lift-and-project cuts

Computational Management Science

Provided in Cooperation with: Springer Nature

Suggested Citation: Glushko, Pavlo; Fábián, Csaba I.; Koberstein, Achim (2022) : An L-shaped method with strengthened lift-and-project cuts, Computational Management Science, ISSN 1619-6988, Springer, Berlin, Heidelberg, Vol. 19, Iss. 4, pp. 539-565, https://doi.org/10.1007/s10287-022-00426-y

This Version is available at: https://hdl.handle.net/10419/311075

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.



WWW.ECONSTOR.EU

https://creativecommons.org/licenses/by/4.0/

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



ORIGINAL PAPER



An L-shaped method with strengthened lift-and-project cuts

Pavlo Glushko¹ · Csaba I. Fábián² · Achim Koberstein¹

Received: 7 July 2021 / Accepted: 21 March 2022 / Published online: 5 May 2022 © The Author(s) 2022

Abstract

Lift-and-project (L&P) cuts are well-known general 0–1 programming cuts which are typically deployed in branch-and-cut methods to solve MILP problems. In this article, we discuss ways to use these cuts within the framework of Benders' decomposition algorithms for solving two-stage mixed-binary stochastic problems with binary first-stage variables and continuous recourse. In particular, we show how L&P cuts derived for the master problem can be strengthened with the second-stage information. An adapted L-shaped algorithm and its computational efficiency analysis is presented. We show that the strengthened L&P cuts can significantly reduce the number of iterations and the solution time.

Keywords Stochastic programming \cdot L-shaped method \cdot Lift-and-project cuts \cdot Benders' decomposition

1 Introduction

In this article, we address the class of two-stage mixed-binary stochastic problems with binary variables in the first stage (master problem) and continuous recourse (subproblems on the second stage). Solution algorithms based on Benders' decomposition (Benders 1962; Rahmaniani et al. 2017b), in this context commonly known as the *L-shaped* method (Van Slyke and Wets 1969), have been successfully applied for this class of models [e.g, in Rahmaniani et al. (2017a); Bihlmaier et al. (2009); Weskamp

 Pavlo Glushko glushko@europa-uni.de
 Csaba I. Fábián fabian.csaba@gamf.uni-neumann.hu
 Achim Koberstein koberstein@europa-uni.de

¹ European University Viadrina, Grosse Scharrnstrasse 59, 15230 Frankfurt (Oder), Germany

² Department of Informatics, John von Neumann University, 10 Izsáki út, 6000 Kecskemet, Hungary

et al. (2019)]. Binary variables in the master problem considerably raise the solution complexity. Often, more than 90% of the execution time is spent to solve the mixed-integer linear (MIP) master problem (Zarandi 2010).

McDaniel and Devine (1977) specifically targeted this class of problems with a *two-phase method*. According to their approach, Benders cuts are constructed for the LP relaxation of the master problem in the initial iterations the first phase. The algorithm continues with the MIP formulation of the master problem the second phase. This two-phase method stays popular in many contemporary implementations (Rahmaniani et al. 2017b; Angulo et al. 2016; Cordeau et al. 2006; Costa 2005) but can be inefficient if the master problem has a large integrality gap (Bodur and Luedtke 2016). We will refer to this issue as *the weak integer relaxation*. The natural solution to lower the integrality gap is to tighten the relaxation of the master problem with valid inequalities (Costa et al. 2012). Most implementations use problem-specific valid inequalities (Saharidis et al. 2011; Tang et al. 2013; Adulyasak et al. 2015) but there exist several general purpose extensions utilizing, e.g., mixed integer rounding inequalities (Vadlamani et al. 2018; Bodur and Luedtke 2016) or split cuts (Bodur et al. 2017).

Alternatively, cutting planes constructed for the second stage can potentially be modified accordingly and applied to the master problem to improve the weak integer relaxation. For example, (Gade et al. 2014) generate Gomori cuts to approximate the second-stage integer subproblems. Zhang and Küçükyavuz (2014) use Gomory cuts to convexify the first and the second stages. Qi and Sen (2017) developed two convexification schemes based on disjunctive cuts to obtain approximations of the second-stage mixed-integer problem. Fenchel cuts a class of disjunctive cuts are generated for the fractional second-stage variables in Ntaimo (2013).

In addition, decomposition possesses further known drawbacks, one of which is a lack of information between two subsequent stages in the initial iterations of the algorithm (Rahmaniani et al. 2017b), sometimes referred to as *the weak Benders relaxation*. Researchers suggested alternative decomposition strategies, in particular, partial decomposition (Crainic et al. 2016), dual decomposition (Carøe and Schultz 1999), and nonstandard decomposition (Gendron et al. 2016). In their essence, these strategies append constraints and variables from the subsequent stages into the current stage. Of course, the more information from the subsequent stage the current stage contains, the less decomposed (and hence, larger) every subproblem is.

In this work, we suggest to improve the weak integer and the weak Benders relaxation of the MIP master problem by deriving a special variant of *lift-and-project cuts*. Lift-and-project (L&P) cuts have been introduced in the seminal work of Balas (1979) as general binary programming cuts, i.e., not depending on a special problem structure. Balas et al. (1993) demonstrated the efficiency of these cuts in branch-and-cut frameworks for mixed binary programming models. L&P cuts have become a standard technique in commercial solution software for MIP and are as such also automatically deployed when the MIP master problem is solved within a decomposition approach. One of the first adaptions of these cuts for stochastic problems specifically was done by Carøe and Tind (1997) and is in the active research since then (see, for example, articles of Sen and Higle (2005) and Sen and Sherali (2006)). To the best of our knowledge, these and the recent works focus on stochastic problems with (mixed) integer recourse and revolve around convexification of the value functions of the second-stage IP.

Our approach is strategically different from existing techniques utilizing L&P cuts in that we consider problems with binary variables on the first stage and strengthen L&P cuts for master problem by incorporating second-stage information in their cut generating problem. This tactic counters both the weak integer and the weak Benders relaxations. Similar approach was applied in Kılınç et al. (2017) for convex mixed integer nonlinear programs. The stochastic master problem and the second stage approximation in our formulation corresponds to the mixed-integer nonlinear program and the linearizations of the nonlinear constraints, respectively, in their work. The authors strengthen the lift-and-project cuts by using an outer approximation within the lift-and-project cut generating problem. Key differences of our research are the application domain stochastic optimization, and the rank of the cuts applied. Kılınç et al. (2017) focus on the rank-one cuts, i.e., only linearization constraints without liftand-project cuts are included in the cut generating problem. While the work described above fits within the lift-and-project framework, it appears that this work is the first to consider using such strengthening for the stochastic problems.

The contribution of the article can be characterized as follows:

- We devise a reformulated cut generating problem for L&P cuts applied to a Benders master problem and strengthened by the addition of the second stage information.
- The integration of the strengthened L&P cuts into a Benders' decomposition scheme is showed, leading to a new variant of the L-shaped method.
- The new method is implemented within the framework of the PNB Solver, a solver system for two- and multi-stage stochastic programming problems (Wolf and Koberstein 2013).
- A numerical study is conducted on a benchmark set of suitable model instances. We investigate the effect of the strengthened L&P cuts on the number of iterations and the solution time. Our enhanced method decreases the required number of iterations and shortens the solution time for some groups of instances.

The rest of this article is organized as follows. To familiarize the reader with the concepts and the notation used, we provide a description of the two-stage stochastic optimization problem and the L-shaped method in Sect. 2 followed by the introduction to L&P cuts in Sect. 3. In Sect. 4, the formal description of domain and epigraph L&P cuts for the master problem of a stochastic optimization problem is provided. The main idea of L&P cut strengthening and a cut generation algorithm is developed in Sect. 5. In Sect. 6, we present and discuss the setup and the results of the computational study. Summary is offered in Sect. 7.

2 The L-shaped method

The two-stage stochastic mixed binary linear optimization problems are considered for a discrete finite scenario set. Let the first-stage decision variables be indexed by the set $\mathcal{N} = \{1, ..., n\}$ where all these variables are non-negative, and the binary variables are indexed by the set $\mathcal{B} \subseteq \mathcal{N}$. The first stage subproblem can be expressed

$$\min c^{T} \boldsymbol{x} + q(\boldsymbol{x})$$

such that
$$A\boldsymbol{x} \ge \boldsymbol{b},$$

$$\boldsymbol{x} \ge \boldsymbol{0},$$

$$x_{i} \in \{0, 1\} \quad (i \in \mathcal{B}),$$

(1)

where q(.) is the polyhedral convex expected recourse function, c is the vector of the objective function coefficients, A is the coefficient constraint matrix, and b is the vector of RHS coefficients. The function $q : I \mathbb{R}^n \to I \mathbb{R}$ is assumed to be finite valued and defined on the whole space. Let $\Gamma = \{ (\vartheta, \mathbf{x}) | \vartheta \ge q(\mathbf{x}) \}$ denote the epigraph of q(.), which is a convex polyhedron.

Let us reformulate (1) as

n

$$\min \vartheta + c^{T} x$$
such that
$$(\vartheta, x) \in \Gamma,$$

$$Ax \ge b,$$

$$x \ge 0,$$

$$x_{i} \in \{0, 1\} \ (i \in \mathcal{B})$$

$$(2)$$

and consider the convex hull of the integer feasible solutions:

$$\Gamma^{\mathcal{B}} = \operatorname{conv}\left(\{ (\vartheta, \boldsymbol{x}) \in \Gamma \mid A\boldsymbol{x} \ge \boldsymbol{b}, \ \boldsymbol{x} \ge \boldsymbol{0}, \ x_i \in \{0, 1\} \ (i \in \mathcal{B}) \}\right).$$
(3)

The traditional L-shaped method (Van Slyke and Wets 1969) finds the solution for (2) by iteratively approximating (3) using *optimality cuts* of the form $\vartheta \ge (a \text{ linear function of } x)$. Hence, the first line in (2) is substituted with the everextending set of these cuts in the form

$$\vartheta \mathbf{1} \ge V \boldsymbol{x} + \boldsymbol{w},\tag{4}$$

where **1** stands for $(1, ..., 1)^T \in I \mathbb{R}^k$, and $V \in I \mathbb{R}^{k \times n}$, $\boldsymbol{w} \in I \mathbb{R}^k$, with *k* increasing with every new cut appended to the model of the epigraph. We call problem (2) with cut set (4) the *current* (for some *k*) *master problem*.

Algorithm 1 outlines the L-shaped method. For the sake of simplicity, we assumed complete recourse, which implies that the second-stage subproblems defining the value of $q(\mathbf{x})$ are feasible for any value \mathbf{x} .

Algorithm 1 The L-shaped method

1.1 Initialization. Set $UB = +\infty$. Initialize optimality tolerance ϵ . 1.2 Solve current master problem. Solve the current master problem. If the master problem is infeasible or unbounded, stop. Let $(\overline{\vartheta}, \overline{x})$ denote the optimal solution of master problem. 1.3 Generate optimality cuts. Evaluate $q(\bar{x})$, i.e., solve the recourse problem at point \bar{x} . If $c^T \bar{x} + q(\bar{x}) < UB$,

set $UB = c^T \overline{x} + q(\overline{x})$ and $x^* = \overline{x}$.

Let $LB = c^T \overline{x} + \overline{\vartheta}$. If $UB - LB < \epsilon$, stop. x^* is a near-optimal solution. Generate an optimality cut and append it to the cut set (4). Go to step 1.2.

Multiple variants and extensions of the L-shaped method exist. Choosing the most suitable one for a specific problem instance can significantly decrease the number of iterations and the solution time. We discuss two such variants below while important implementation aspects are addressed later in Sect. 6.

Algorithm 1 represents the single-cut variant of the L-shaped method in which all the information from the second-stage subproblems is aggregated into one optimality cut. As a result, much of the information about the second stage is lost. The opposite approach is the multi-cut variant, in which one optimality cut is constructed for every second-stage subproblem defining the value of q(x). Thus, the number of aggregates is equal to the number of discrete scenarios. The master problem of this variant has multiple ϑ -variables (one per subproblem) in the objective function. The advantage of the multi-cut variant is that most information about the second stage is preserved. On the other hand, appending multiple cuts instead of one to the master problem in every iteration makes the problem larger and harder to solve.

The rule of thumb is to use single-cut when the number of scenarios is significantly larger than the number of first-stage constraints (Birge and Louveaux 2011). In principle, the number of aggregates can lie between one and the number of scenarios. This so-called hybrid-cut approach was introduced by Birge and Louveaux (1988).

A common computational technique for mixed-integer stochastic programming problems is the *two-phase method*. Its first phase is equivalent to the Algorithm 1 except that the LP relaxation of the master problems is solved in step 1.2. Normally, solving the relaxation is much faster. The *continuous optimality cuts* found in the first phase are then appended to the cut set (4) and the problem is solved again, this time with the integrality requirements. The first phase improves the approximation of $\Gamma^{\mathcal{B}}$ with continuous optimality cuts that are relatively cheap to construct.

3 Lift-and-project cuts for binary programming models

In this section we overview the derivation of lift and project cuts (L&P cuts) within a classic branch-and-cut framework for deterministic mixed binary programming problems. The discussion is based on Balas (1979) and Balas et al. (1993, 1996).

Decision variables will be indexed by the set $\mathcal{N} = \{1, ..., n\}$. Non-negative variables and binary variables will be indexed by the sets \mathcal{P} and \mathcal{B} , respectively. Here $\mathcal{B} \subseteq \mathcal{P} \subseteq \mathcal{N}$.

The mixed integer programming problem is formulated with some redundancy as

$$\min c^{T} x$$
such that
$$Mx \ge d,$$

$$x_{i} \ge 0 \quad (i \in \mathcal{P}),$$

$$x_{i} \in \{0, 1\} \quad (i \in \mathcal{B}),$$
(5)

where c is the vector of the objective function coefficients, M is the coefficient constraint matrix, and d is the vector of RHS coefficients.

To get the LP relaxation of (5) the bottom line is substituted with $x_i \le 1$ $(i \in B)$. In what follows, the inequality system $Mx \ge d$ is assumed to contain $-x_i \ge -1$ $(i \in B)$, hence the bottom line in the relaxation can be omitted. Let *m* denote the number of the rows of *M*.

We will consider cuts in the form $\boldsymbol{\alpha}^T \boldsymbol{x} \ge \beta$, where $\boldsymbol{\alpha} \in I \mathbb{R}^n$ and $\beta \in I \mathbb{R}$. The cut is called valid if $\boldsymbol{\alpha}^T \boldsymbol{x}^\circ \ge \beta$ holds for any integer feasible solution \boldsymbol{x}° .

Let *K* denote the feasible polyhedron of the LP relaxation of (5). The idea is to consider cuts satisfied by the points of the polyhedra $K \cap \{ x \in I \mathbb{R}^n \mid -x_i \ge 0 \}$ and $K \cap \{ x \in I \mathbb{R}^n \mid x_i \ge 1 \}$ for $i \in \mathcal{B}$. A dual-type characterization of such cuts is given in Balas et al. (1993). An overview with explanations and proofs is provided in the Online Supplement (available as supplemental material at https://doi.org/10.1007/s10287-022-00426-y).

The cut $\boldsymbol{\alpha}^T \boldsymbol{x} \geq \beta$ is satisfied by the above polyhedra iff there exist $\boldsymbol{u}, \boldsymbol{v} \in$ I R^{*m*}, $\boldsymbol{u}, \boldsymbol{v} \geq \mathbf{0}$ and $u_0, v_0 \in$ I R, $u_0, v_0 \geq 0$ such that

$$M^{T}\boldsymbol{u} - u_{0}\boldsymbol{e}_{i} \qquad [\leq / =] \boldsymbol{\alpha}, M^{T}\boldsymbol{v} + v_{0}\boldsymbol{e}_{i} \ [\leq / =] \boldsymbol{\alpha}, d^{T}\boldsymbol{u} \qquad \geq \boldsymbol{\beta}, d^{T}\boldsymbol{v} + v_{0} \geq \boldsymbol{\beta},$$
(6)

where e_i is the *i*th unit vector in I R^{*n*}, and in the top two lines, we have ' \leq ' in the individual constraints belonging to indices in \mathcal{P} , and '=' in the constraints belonging to indices not in \mathcal{P} . The system (6) is solvable in any non-trivial case.

Let $\overline{x} \in K$ be a feasible solution of the of the LP relaxation of (5). Assume that \overline{x} is not a feasible solution of the integer programming problem (5) because $0 < \overline{x}_i < 1$ for some $i \in \mathcal{B}$. The aim is to find a cut that separates \overline{x} from the integer feasible solutions. This will be obtained by solving an LP model, where decision variables are $\alpha \in I \mathbb{R}^n$, $\beta \in I \mathbb{R}$ and $u, v \in I \mathbb{R}^m$, $u_0, v_0 \in I \mathbb{R}$ with lower bounds $u, v \ge 0$ and $u_0, v_0 \ge 0$. The constraints are the inequality system (6) and the normalization constraints for (α, β) . Note 1: The latter are needed since, otherwise, the feasible domain would be a cone. Note 2: The goal is to find a cut as deep as possible, which is formulated by maximizing the objective $\beta - \overline{x}^T \alpha$.

Two types of normalization constraints were investigated in Balas et al. (1993). The β -normalization prescribes either $\beta = 1$ or $\beta = -1$. This is technically easy to handle, but may result an infeasible problem, or may not bound the objective. The general normalization prescribes $\sum_{i=1}^{n} |\alpha_i| = 1$; it is called α -normalization.

Another normalization type is the standard normalization condition of form $\sum_{i=1}^{m} u_i + \sum_{i=1}^{m} v_i + u_0 + v_0 = 1$. It is often considered to be more effective than others (Fischetti et al. 2011).

4 Lift-and-project cuts for master problem

As the master problem in the L-shaped method is a mixed integer problem, L&P cuts approximating $\Gamma^{\mathcal{B}}$ of the stochastic problem (2) can be constructed. Using the objects from Sect. 2, the current master problem LP relaxation can be expressed as

$$\begin{array}{rcl} \min & \vartheta & + c^T \mathbf{x} \\ \text{such that } \vartheta \mathbf{1} - V \mathbf{x} \geq \mathbf{w}, \\ & A \mathbf{x} \geq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{array}$$
(7)

Note that $-x_i \ge -1$ $(i \in \mathcal{B})$ is included in $Ax \ge b$.

Let L denote the feasible polyhedron of this LP problem, and let

$$K = \{ \boldsymbol{x} \mid A\boldsymbol{x} \ge \boldsymbol{b}, \ \boldsymbol{x} \ge \boldsymbol{0} \}.$$
(8)

The L&P cut will be constructed in the form $(\zeta, \alpha)^T (\vartheta, x) \ge \beta$, where $\zeta \in I \mathbb{R}$, $\alpha \in I \mathbb{R}^n$ and $\beta \in I \mathbb{R}$. Such cut is called valid if $\zeta \vartheta^\circ + \alpha^T x^\circ \ge \beta$ holds for any feasible solution $(\vartheta^\circ, x^\circ)$ where x° is integer.

Let $(\overline{\vartheta}, \overline{x})$ be a feasible solution of the problem (7) in which \overline{x} is not integer. Therefore, this solution is not feasible for problem (2). The aim is to find cuts that separate $(\overline{\vartheta}, \overline{x})$ from the integer feasible solutions. We distinguish two types of L&P cuts for master problem: domain and epigraph cuts.

4.1 Domain lift-and-project cuts

The first type *domain cuts* do not include the ϑ in their formulation, i.e., it has coefficient 0. These cuts approximate the projection of $\Gamma^{\mathcal{B}}$ to the *x*-space. The projection is, of course, the convex hull of the integer feasible solutions of (2). The ever-extending set of domain cuts will be formulated as

$$A\boldsymbol{x} \ge \boldsymbol{b}.\tag{9}$$

Let *m* denote the number of the rows of *A*, increasing with every new domain cut that is generated. The initial set of domain cuts is inherited from (7).

If $\overline{x} \notin \text{conv}(\{x \in K | -x_i \ge 0\} \cup \{x \in K | x_i \ge 1\})$ for some $i \in \mathcal{B}$, then a domain cut can be constructed. To do this, the construction of Sect. 3 is applied with the following objects. Let $\mathcal{N} := \mathcal{P}$ denote the set of the indices of the *x*-components. Let $\mathcal{M} := A$ and d := b.

4.2 Epigraph lift-and-project cuts

The second type of L&P cuts for master problem are *epigraph cuts*. They have the form $\vartheta \ge (a \text{ linear function of } x)$ and are approximating $\Gamma^{\mathcal{B}}$. Since epigraph cuts have the same form as optimality cuts, they are appended to the optimality cut set (4).

Let $i \in \mathcal{B}$ such that $0 < \overline{x}_i < 1$. The idea is to consider cuts satisfied by the points of the polyhedra $L_0 = L \cap \{ (\vartheta, \mathbf{x}) \in I \mathbb{R}^{1+n} \mid -x_i \ge 0 \}$ and $L_1 = L \cap \{ (\vartheta, \mathbf{x}) \in I \mathbb{R}^{1+n} \mid x_i \ge 1 \}$. The construction of Sect. 3 is applied with the following objects. Let \mathcal{N} denote the whole index set, belonging to the variables (ϑ, \mathbf{x}) , and let \mathcal{P} denote the set of the indices of the \mathbf{x} -components. Moreover, let

$$M := \begin{pmatrix} \mathbf{1} - V \\ A \end{pmatrix}, \qquad \mathbf{d} := \begin{pmatrix} \mathbf{w} \\ \mathbf{b} \end{pmatrix}. \tag{10}$$

The characterization of the resulting cuts is discussed in 1. The plain epigraph cut generating LP problem (plain epigraph CGP) is described in 1. A natural means of normalization here is prescribing $\zeta = 1$. Let us call it ζ -normalization. Having found an optimal solution to the plain epigraph CGP a ζ -normalized epigraph cut for (7) can be constructed. The proof of the following observation can be found in 1.

Observation 1 Suppose that the objective of the plain epigraph CGP is still not bounded. Then a domain cut can be constructed, separating \overline{x} from the polyhedra $\{x \in K | x_i \ge 0\}$ and $\{x \in K | x_i \ge 1\}$.

5 Strengthened epigraph cuts

The procedure described so far can only use the information already included in the LP relaxation of the current master problem (7), which is limited to first-stage constraints, epigraph cuts and the current approximation of the recourse function through optimality cuts. Especially in the initial iterations of the algorithm, the latter is non-existent or very coarse. Therefore, in this section we suggest a strengthening procedure for epigraph L&P cuts to overcome the weak Benders relaxation. Strengthening is based on the inclusion of the second-stage information into the plain epigraph CGP.

The dual of the plain epigraph CGP is the following problem. Decision variables are τ , $\upsilon \in I R$, z, $y \in I R^n$ and z_0 , $y_0 \in I R$ with the bounds z, $y \ge 0$, z_0 , $y_0 \ge 0$. The constraint set is

$$\begin{aligned}
\mathbf{1}\tau & -Vz & -\boldsymbol{w} z_0 \geq \mathbf{0}, \\
& Az & -\boldsymbol{b} z_0 \geq \mathbf{0}, \\
& -z_i & \geq 0, \\
& \mathbf{1}\upsilon & -Vy & -\boldsymbol{w} y_0 \geq \mathbf{0}, \\
& & Ay & -\boldsymbol{b} y_0 \geq \mathbf{0}, \\
& & y_i & -y_0 \geq \mathbf{0}, \\
& & z & + y & = \overline{x}, \\
& & & z_0 & +y_0 = 1,
\end{aligned}$$
(11)

where $i \in \mathcal{B}$ has been selected such that $0 < \overline{x}_i < 1$. The objective is min $\tau + \upsilon$. Let us bring (11) to a simpler form. To alleviate notation, let $\lambda = \overline{x}_i$.

- 1. From $-z_i \ge 0$ in line 3 of (11), and the non-negativity bound, it follows that $z_i = 0$.
- 2. Consider the equality system $z + y = \overline{x}$ in line 7 of (11). The equation belonging to the index *i* is $z_i + y_i = \overline{x}_i$. Hence from item 1, above, we get $y_i = \overline{x}_i = \lambda$.
- 3. We have $Ay by_0 \ge 0$ in line 5 of (11). Since $i \in \mathcal{B}$, this inequality system contains $-y_i + y_0 \ge 0$ (see the dual problem depicted in the 'graph cut & dual' sheet). Moreover, we have $y_i y_0 \ge 0$ in line 6 of (11). Hence $y_0 = y_i$, and form item 2, above, we get $y_0 = \lambda$.
- 4. Notice that $z_0 + y_0 = 1$ in the bottom line of (11). Hence from item 3, above, it results $z_0 = 1 \lambda$.

Summing up: $y_0 = \lambda$ and $z_0 = 1 - \lambda$, removing the variables in z_0 and y_0 in (11), by shifting the right-hand-side vector with the corresponding columns, it results in the constraint system:

$$\begin{aligned}
\mathbf{1}\tau & -Vz &\geq (1-\lambda)w, \\
Az &\geq (1-\lambda)b, \\
z_i &= 0, \\
\mathbf{1}\upsilon & -Vy \geq \lambda w, \\
Ay \geq \lambda b, \\
y_i &= \lambda, \\
z &+ y &= \overline{x}.
\end{aligned}$$
(12)

Let us divide lines 1, 2, 3 of (12) by $(1 - \lambda)$, and lines 4, 5, 6 by λ . (Note that it is assumed that $0 < \lambda < 1$.)

By substituting the objects

$$\vartheta' = \frac{1}{1-\lambda}\tau, \quad \mathbf{x}' = \frac{1}{1-\lambda}z, \\ \vartheta'' = \frac{1}{\lambda}\upsilon, \quad \mathbf{x}'' = \frac{1}{\lambda}\mathbf{y},$$

the system (12) transforms to (13).

Decision variables of the CGP (13) are $\vartheta', \vartheta'' \in I \mathbb{R}$ and $x', x'' \in I \mathbb{R}^n$. Having found its optimal solution with a simplex-type method, a ζ -normalized epigraph cut for (7) can be constructed using LP duality.

Instead of the above CGP (13), we are going to solve a stronger problem that additionally includes second-stage information in the form of the convex constraints

$$\vartheta' \ge q(\mathbf{x}') \text{ and}$$

 $\vartheta'' \ge q(\mathbf{x}'').$
(14)

This strengthened problem can be effectively solved by decomposition again (we present a modified variant of the L-shaped algorithm below), meanwhile generating new optimality cuts for the expected recourse function q(.). Let $\vartheta \mathbf{1} \ge \widetilde{V}\mathbf{x} + \widetilde{\mathbf{w}}$ denote the system of the optimality cuts generated until a near-optimal solution has been found. Let us append the following constraints to the system (13):

$$\vartheta' \mathbf{1} - \widetilde{V} \mathbf{x}' \qquad \geq \quad \widetilde{\mathbf{w}}, \\ \vartheta'' \mathbf{1} - \widetilde{V} \mathbf{x}'' \geq \quad \widetilde{\mathbf{w}}.$$
 (15)

Having found an optimal solution to the CGP (13) plus (15) with a simplex-type method, a ζ -normalized *strengthened epigraph cut* can be constructed using LP duality. This strengthened cut is then appended to the model as an epigraph cut.

Remark 1 Optionally, the optimality cuts $\vartheta \mathbf{1} \ge \widetilde{V}\mathbf{x} + \widetilde{\mathbf{w}}$ may also be appended as epigraph cuts, where V and \mathbf{w} are enhanced as follows,

$$V := \begin{pmatrix} V \\ \widetilde{V} \end{pmatrix}, \qquad \boldsymbol{w} := \begin{pmatrix} \boldsymbol{w} \\ \widetilde{\boldsymbol{w}} \end{pmatrix}. \tag{16}$$

Let us name the process of L&P cut generation for all or some $\{\overline{x}_i | i \in \mathcal{B}, 0 < \overline{x}_i < 1\}$ in the current master problem (7) an *L&P round*. Strengthened epigraph cuts are integrated into the traditional L-shaped method by starting it with several L&P rounds. In every round, the LP relaxation of the current master problem (7) is solved for deriving L&P cuts as discussed above. In principle, the strengthened L&P cuts suffice to reach an integer optimal solution of problem (2) by a pure cutting plane procedure. It is however well-known, that such a procedure is not an efficient approach to solve problem (2). Therefore, after a certain number of L&P rounds, it is recommended to switch to the traditional L-shaped type iterations, involving the solution of the MIP master problem and the derivation of optimality cuts. The resulting *L-shaped method with L&P cuts* is stated in Algorithm 2.

Algorithm 2 The L-shaped method with L&P cuts

2.1 Initialization. Set $UB = +\infty$. Initialize optimality tolerance ϵ . Set L&P rounds limit. Set L&P rounds counter = 0. 2.2 Solve current master problem. If L&P rounds counter $\geq L\&P$ rounds limit, solve the master problem, else solve its LP relaxation (7). If the master problem is infeasible or unbounded, stop.

🖄 Springer

Otherwise, let $(\overline{\vartheta}, \overline{x})$ denote its optimal solution. Let $LB = c^T \overline{x} + \overline{\vartheta}$. Set $\mathcal{F} = \{i \in \mathcal{B} | \ 0 < \overline{x}_i < 1\}$. If $\mathcal{F} = \emptyset$, then set L&P rounds counter = L&P rounds limit and go to step 1.4. 2.3 L&P round. For all or some $i \in \mathcal{F}$ do:

2.3.1 Solve the CGP (13) plus (15) using Algorithm 3. 2.3.2 If CGP (13) plus (15) is infeasible, construct a domain cut, else construct an epigraph cut using LP duality.

Append domain cuts to the constraint set (9) and epigraph cuts to (4) in the manner of (16).

Set L&P rounds counter = L&P rounds counter + 1. Go to step 2.2. 2.4 Generate optimality cuts. Evaluate $q(\bar{x})$, i.e., solve the recourse problem at point \bar{x} .

Evaluate $q(\mathbf{x})$, i.e., solve the recourse problem at point \mathbf{x} If $c^T \overline{\mathbf{x}} + q(\overline{\mathbf{x}}) < UB$,

set
$$UB = c^T \overline{x} + q(\overline{x})$$
 and $x^* = \overline{x}$.

If $UB - LB < \epsilon$, stop. x^* is a near-optimal solution.

Generate an optimality cut and append it to the master problem. Go to step 2.2.

In Algorithm 2, the parameter L&P rounds limit controls the number of rounds. As soon as the L&P rounds counter reaches the limit or when there are no fractional variables in the solution of the current master problem LP relaxation, the algorithm switches to the L-shaped method (see step 2.2). When L&P rounds limit = 0, Algorithm 2 is equivalent to the traditional L-shaped method (Algorithm 1). Alternatively, if L&P rounds limit = ∞ , Algorithm 2 is equivalent to a pure cutting plane approach with L&P-cuts. In all experiments the CGP (13) plus (15) never was unbounded, thus, domain cuts were never constructed in step 2.3.2.

It is assumed that a comparatively small number of L&P rounds significantly reduces the number of iterations afterwards or makes the solution of the MIP master problem easier in these iterations.

As suggested in Balas et al. (1996), L&P cuts can be generated for a subset of the fractional variables. The parameter % *of fract with L&P cuts* defines the proportion of the fractional variables in \mathcal{F} for which cuts are generated in step 2.3 in Algorithm 2. For instance, if $|\mathcal{F}|$ is 10 and % *of fract with LP cuts* = 50, then the binary variables in \mathcal{F} are sorted according to their fractionality (closeness to 0.5) and for the first half (in this case, 5) of them cuts are generated.

Note, that the strengthened CGP is characterised by a double block-angular structure and is, therefore, itself amenable to decomposition. Algorithm 3 describes an adapted L-shaped method to solve this problem. The main difference in comparison with the traditional L-shaped method is that the recourse function needs to be evaluated twice in step 3.3 of Algorithm 3, each evaluation entailing the solution of all second-stage subproblems. Since CGP (13) plus (15) is a pure LP, Algorithm 3 can be sped up by all the extensions known for the classical L-shaped method, including regularization, on-demand accuracy and adapted multi-cut aggregation (see e.g., Wolf and Koberstein 2013 and Wolf et al. 2014 and the references therein). Note, that a distinct optimality tolerance ϵ^{CGP} is used as a stopping criterion in Algorithm 3. We assume that the higher the value of ϵ^{CGP} is the weaker epigraph cuts are generated. Epigraph cuts from the previous rounds are included in the constraints of (13), therefore, epigraph cuts found with Algorithm 2 are higher rank cuts derived by using previously-generated cuts.

To construct plain epigraph cuts instead of strengthened, CGP (13) should not be strengthened with (15) on step 2.3.1. CGP (13) is then solved as standard LP with a simplex-type method.

Algorithm 3 Decomposition method for solving the CGP for strengthened epigraph cuts

3.1 Initialization. Set $UB^{CGP} = +\infty$. Initialize optimality tolerance ϵ^{CGP} . Initialize \widetilde{V} and \widetilde{w} as empty. 3.2 Solve CG current master problem. Solve the CG master problem (13) plus (15). Constraint set (15) in this problem is empty in the first iteration. Let $(\overline{\vartheta}', \overline{\vartheta}'', \overline{x}', \overline{x}'')$ denote the optimal solution. If the CG master problem is infeasible, stop. 3.3 Solve recourse problems. Evaluate $q(\overline{x}')$ and $q(\overline{x}'')$, i.e., solve the recourse problem at the points \overline{x}' and \overline{x}'' . If $(1 - \lambda) \max(\overline{\vartheta}', q(\overline{x}')) + \lambda \max(\overline{\vartheta}'', q(\overline{x}'')) < UB$, set $UB^{CGP} = (1 - \lambda) \max(\overline{\vartheta}', q(\overline{x}')) + \lambda \max(\overline{\vartheta}'', q(\overline{x}''))$ and $x'^* = \overline{x}'$, $x''^* = \overline{x}''$. Let $LB^{CGP} = (1 - \lambda)\overline{\vartheta}' + \lambda\overline{\vartheta}''$.

If $UB^{CGP} - LB^{CGP'} < \epsilon^{CGP}$, stop. $(\mathbf{x}'^*, \mathbf{x}''^*)$ is a near-optimal solution. 3.4 Generate optimality cuts.

Generate optimality cuts from the dual information of the recourse problems $q(\overline{x}')$ and $q(\overline{x}'')$ and append them to the constraint set (15). Go to step 3.2.

6 Computational study

In this section we evaluate the efficiency of epigraph cut strengthening. We start by identifying the best set of parameters for Algorithm 2.

Let us call a particular set of parameters a *test case*. Then, a *test* is a solution of a specific problem instance using the test case parameters.

The following measures of efficiency are collected for every test:

• *Number of iterations*—the number of times Algorithm 2 executed step 2.4, i.e., second-stage subproblems were solved, until the near optimal solution for problem (1) is found.

• *Solution time*—the wall-clock time of the algorithm run until the near optimal solution is found.

We aggregate these measures using the arithmetic mean (AM) and the geometric mean (GM). The GM is sensitive to changes in computing times for small instances. Bigger instances on the contrary dominate the results aggregated with the AM (Achterberg 2007). The latter is more suitable for this study as we aim to improve the solution time of difficult instances.

6.1 Implementation

We integrated the generation of strengthened epigraph cuts as an additional routine in the solver system PNB Solver (Wolf and Koberstein 2013). The solver itself is written in C++ and uses components from the COIN-OR project, such as COIN SMI for reading the SMPS files and COIN OSI for communicating with the external solvers. The core component is an implementation of a nested L-shaped algorithm for two and multi-stage stochastic LP problems, allowing for integer variables on the first stage. The solver possesses advanced techniques including level-decomposition and regularization, warm start, parallelization, cut aggregation and consolidation, and ondemand accuracy (Wolf et al. 2014). To track the effectiveness of strengthened L&P cuts we apply the single- and the multi-cut L-shaped method depending on what better suits the specific test set. The master problem and every subproblem are solved with the external solver IBM CPLEX version 12.9 (https://www.ibm.com/products/ilogcplex-optimization-studio). Unless particularly specified, the optimality gap tolerance is set to $1e^{-6}$.

The PNB solver code is compiled in Visual Studio 19 in release mode. All test cases are solved on a workstation with AMD Ryzen 9 3950X 16-Core (32 Threads) 3.5 GHz processor working on a 64-bit Windows 10 operating system. The time limit is set to three hours per test.

Warm start is used for all experiments. The warm start technique uses the optimal basis information of previously solved subproblems. Every time a subproblem needs to be solved the simplex algorithm is initialized with the last optimal basis if one is available. Since every subproblem has only minor differences from iteration to iteration, warm start has a significant positive effect on the second-stage solution time.

Parallelization is applied to multiple mutually independent steps in the algorithm. For instance, the second-stage subproblems evaluating q(.) are solved in all versions of the L-shaped algorithm in parallel. Solution of L&P cut generating problems for variables in \mathcal{F} in Algorithm 2 is also parallelized. Considering that the strengthened CGP for epigraph cut itself is solved using the L-shaped method, multiple instances of this algorithm can run simultaneously.

6.2 Test instances

Appropriate instances for our experiments are two-stage stochastic programming problems with complete continuous recourse and binary variables on the first stage. Table 1 provides the key dimensions of the instances we use, including the number of binary variables on the first stage *BV1*, the number of continuous variables on the second stage per scenario, *BV2*, the number of constraints (rows) on the first and the second stage per scenario, *R1* and *R2*, respectively, and the number of scenarios *S*. Every row in the table corresponds to the instance group that consists of multiple instances. None of the instances has first-stage continuous variables.

The main class of test instances is the multi-commodity flow network design problems with demand uncertainty (*MCFNDSD*) based on the **R** instances from Rahmaniani et al. (2017a). We generated a complete recourse version to avoid the necessity of feasibility cuts by adding slack variables to the right-hand side of the flow conservation constraints at the origin and the destination nodes of each commodity (see constraint set (4) in Rahmaniani et al. (2017a)). Therefore, the dimensions of these instances slightly differ from the dimensions of the original ones. The names of the MCFNDSD instances start with 'r', followed by a class number, cost/capacity ratio, demand correlation (always 0), and the number of scenarios. The **R** instances are gathered according to their classes in r01 to r04 groups. Every group is a combination of instances that have 1, 3, 5, 7, and 9 cost/capacity ratios and 16, 32, and 64 scenarios fifteen instances per group. The modified instances are available for download in smps format at https://github.com/pavloglushko/mcfndsd_complete_recourse/releases/tag/ 0.0.1.

Additionally, we use model instances arising in stochastic server location problems (*SSLP*) taken from Ntaimo (2004). The second-stage integrality constraints were relaxed to comply with the problem requirements. These instances are named 'sslp_m_n_s_i', where m is the number of potential server locations, n is the number of potential clients, s is the number of scenarios, and i is the version of the scenario set (same size but different stochastic parameter values). The SSLP instances are grouped by m_n. We generated instances with 1000 and 2000 scenarios for the groups sslp_5_25, sslp_5_50, and sslp_15_45 using the authors' methodology. The sslp_5_25 and sslp_5_50 groups consist of 20 instances with a various number of scenarios and a - e scenario sets each; sslp_15_45 and sslp_10_50 include 35 and 25 instances, respectively.

6.3 Parameter tuning

Before we evaluate the impact of epigraph cuts strengthening on the L-shaped method, it is necessary to identify the set of parameters that works best for these cuts. Different usage options of optimality cuts generated in strengthened CGP are examined below. Furthermore, the influence of the parameters ϵ^{CGP} , % of fract with L&P cuts, and L&P rounds limit is investigated.

The parameters in the *default test case* are set to the following values: $\epsilon^{CGP} = 0.01$, $L\&P \ rounds \ limit = 20$, and % of fract with $L\&P \ cuts = 0.1$. The two-phase method described in Sect. 2 is applied as it showed to be beneficial. The MCFNDSD instances converge much faster when the multi-cut variant is used, while the SSLP problems are solved faster with the single-cut L-shaped method as they have a relatively large number of scenarios. Therefore, we use the multi-cut variant for the MCFNDSD

Table 1 Prop(erties of the test instances. Se	cond-stage integrality cons	traints were relaxed in sslp instance.	s. The dimensions of the secc	nd stage are provided per scenario
Group	BV1	CV2	Rl	R2	S
r01	35	360	0	135	16, 32, 64
r02	25, 35	650, 900	0	275, 285	16, 32, 64
r03	35	1800	0	535	16, 32, 64
r04	60	610	0	160	16, 32, 64
sslp_5_25	5	130	1	30	50, 100, 1000, 2000
sslp_5_50	5	255	1	55	50, 100, 1000, 2000
sslp_15_45	15	069	1	60	5, 10, 15, 20, 25, 1000, 2000
sslp_10_50	1	60	50, 100, 1000, 2000	10	510

instances and the single-cut variant for the SSLP instances. Any other test case is generated by changing only one parameter value of the default test case. This approach helps us precisely track the influence of every parameter on the solution process.

6.3.1 Usage of CGP optimality cuts

As mentioned in Remark 1, optimality cuts from every run of Algorithm 3 for the the CGP (13) plus (15) can be stored and appended to the constraint set (13) in the manner of (16). Their inclusion into the master problem might have positive effects regarding the number of iterations while simultaneously increasing the size of the master problem. Two approaches were tested to avoid cut proliferation of the master problem:

- Distance violation filtering Only efficient CGP cuts are appended to the master problem. Efficient cuts are those that have a distance violation of at least $1e^{-4}$, which is defined as the absolute violation of the cut divided by the 2-norm of its coefficients (Wesselman 2010). The absolute violation is calculated for the solution of the current master problem. About 90% of the cuts are not efficient. The threshold value of distance violation itself turned out to be not overly important because most of the cuts have a negative absolute violation, i.e., they are not active for the current master problem.
- *Cut consolidation* A technique used to reduce the number of optimality cuts appended to the master problem (Wolf and Koberstein 2013). According to this technique, CGP cuts are generated in the same iteration of Algorithm 3 but for different aggregates are consolidated by summing up their coefficients. Naturally, this technique is only applicable to the multi-cut variant of the L-shaped method. As a result, *S* (number scenarios) optimality cuts are transformed into one, which leads to a smaller master problem but also to a loss of the information. Consolidated cuts have the same form as the traditional optimality cuts in the multi-cut schema except that coefficients for all ϑ s are 1. After the consolidation the system of the CGP cuts has a form

$$\vartheta_1 \mathbf{1} + \dots + \vartheta_S \mathbf{1} \ge V \mathbf{x} + \mathbf{w},\tag{17}$$

where 1 stands for $(1, ..., 1)^T \in I \mathbb{R}^k$, *S* is the number of scenarios, and $V \in I \mathbb{R}^{k \times n}$, $\boldsymbol{w} \in I \mathbb{R}^k$, with *k* increasing with every new consolidated cut appended to the model. Cut set (4) can simultaneously include consolidated and not consolidated optimality cuts, and epigraph cuts. When this approach is utilized, we consolidated CGP cuts on step 2.3.2 of Algorithm 2 before appending them to the current master problem.

We provide the results of our experiments regarding these above approaches in Table 2. For each group of instances, we present the arithmetic means of the solution time and the number of iterations for the case when no CGP cuts are appended at all (*no cuts*), only efficient cuts are appended (*eff*), all cuts are appended in a consolidated fashion (*cons*), or simply all cuts are appended (*all*). Besides, the combined approach

Group	AM Time (A	M Iterations)			
	Use CGP opt	imality cuts			
	No cuts	Eff + Cons	Eff	Cons	All
r01	37 (11)	37 (10)	37(9)	50(10)	57(5)
r02	277(44)	278(45)	269(42)	303(43)	227(24)
r03	534 (42)	539(43)	548(40)	551(40)	577(28)
r04	2404(76)	2305 (75)	2483(74)	2781(76)	2323(44)
sslp_5_25	62(10)	NA	62(10)	NA	52 (6)
sslp_5_50	80(12)	NA	76(11)	NA	65 (6)
sslp_15_45	642(195)	NA	647(199)	NA	532(163)
sslp_10_50	187(138)	NA	243(177)	NA	140(103)
AM (all inst.)	580(72)	NA	510(89)	NA	372(71)
GM (all inst.)	171(141)	NA	180(145)	NA	101(122)

 Table 2
 Effect of adding CGP cuts to the master problem. The solution time (in seconds) and the number of iterations (in parenthesis) are reported in arithmetic (AM) and geometric (GM) means

Best results are emphasized with bold font. Consolidation of CGP cuts is not applicable to single-cut, therefore, sslp instances have NA values

was tested, when only efficient CGP cuts were consolidated (*eff+cons*). Since the SSLP instances are solved with the single-cut variant, the consolidation of CGP cuts is not applicable for them.

As expected, the application of all CGP cuts considerably lowers the number of iterations (by around 40%). For the tests with the SSLP instances, smaller numbers of iterations result in shorter solution times. On the other hand, the effectiveness of the CGP cuts in the tests with the MCFNDSD instances is arguable and often increases solution time. This observation can be explained by the fact that these instances are solved with the multi-cut variant of the L-shaped method. Therefore, Algorithm 3 also applies the multi-cut approach, which leads to the creation of a relatively large number of CGP cuts. If appended to the master problem, these cuts make it considerably larger and thereby more difficult to solve. Limiting the number of the cuts appended or even not using them at all is preferable for comparatively large MCFNDSD instances. We consider not to use CGP cuts in tests with MCFNDSD instances and apply all of them in tests with SSLP instances.

6.3.2 CGP optimality tolerance

The impact of the tolerance ϵ^{CGP} , which is the optimality tolerance for Algorithm 3, has been tested by considering the values 0.5, 0.2, 0.01, $1e^{-4}$, and $1e^{-6}$. Taking into account results in Sect. 6.3.1, CGP cuts are appended to the master problem in the tests with SSLP instances.

The average values in Table 3 indicate that, for larger instances, the lower ϵ^{CGP} is, the fewer iterations Alogrithm 2 needs to converge. However, the higher precision of the algorithm comes at the cost of an increased solution time as the algorithm requires more iterations. Taking into consideration the two observations above, we conclude

Group	AM time (AM	I iterations)			
	ϵ^{CGP}				
	1E-06	1E-04	0.01	0.1	0.5
r01	37(10)	38(11)	37(11)	36(11)	34 (11)
r02	263(43)	272(44)	277(44)	258(44)	238(43)
r03	548(41)	519(40)	534(42)	499(41)	493 (41)
r04	2603(77)	2566(76)	2404 (76)	2529(77)	2563(76)
sslp_5_25	67(5)	56(6)	52 (6)	57(9)	57(8)
sslp_5_50	82(6)	67(5)	65 (6)	71(9)	70(9)
sslp_15_45	736(111)	524(112)	532(163)	514(160)	505 (158)
sslp_10_50	172(75)	126 (77)	140(103)	159(127)	159(125)
AM (all inst.)	530(53)	468(54)	458 (69)	465(73)	463(72)
GM (all inst.)	155(155)	124(124)	123(110)	125(125)	122 (122)

Table 3 Effect of ϵ^{CGP} . The solution time (in seconds) and the number of iterations (in parenthesis) are reported in arithmetic (AM) and geometric (GM) means

Best results emphasized with bold font

that it is recommendable to solve the strengthened CGP only approximately with ϵ^{CGP} set to 0.01.

6.3.3 Selection of fractional variables

Next, we investigate the impact of the parameter % of fract with L&P cuts that controls the number of fractional variables, for which L&P cuts are generated. As mentioned above, only in the tests with the SSLP instances CGP cuts are appended to the master problem. Increasing the percentage of considered fractional variables does not show a profound impact on the number of iterations in the tests with the MCFNDSD instances (see Table 4). On the other hand, the solution time of these instances increases as more cut generating problems need to be solved. The number of iterations for the tests with the SSLP instances drops greatly with the increase of the number of considered fractional variables. More importantly, this effect leads to a decrease in solution time. We explain this observation by the fact that these instances have a small number of binary variables, so that even a large %-value of the parameter corresponds to at most 15 binaries. We conclude that a reasonable setting for this parameter is to generate cuts for the 10% most fractional variables. A lower value does not seem recommendable since all test instances have less than 100 first-stage binary variables.

6.3.4 Number of cut generating rounds

For every selected instance we run tests with three values of the parameter L&P rounds limit: 20, 40, and 100. According to the results in Table 5, increasing the number of L&P rounds tends to lower the total number of iterations. As with % of fract with L&P cuts, the number of cut generating problems solved has

Group	AM Time (AM	iterations)		
	% of fract with	L&P cuts		
	5	10	20	100
r01	29 (11)	37(11)	44(11)	227(12)
r02	275(45)	299(44)	271 (43)	726(56)
r03	512 (42)	587(41)	517(40)	1178(44)
r04	2974(78)	2457(75)	2546(78)	4376(76)
sslp_5_25	52 (6)	53(6)	53(6)	80(2)
sslp_5_50	73(6)	73(6)	65 (6)	103(3)
sslp_15_45	550(171)	452(131)	478(146)	227(48)
sslp_10_50	192(141)	190(140)	124(83)	86(32)
AM (all inst.)	520(77)	448 (69)	455(62)	712(34)
GM (all inst.)	119 (119)	123(110)	125(125)	182(182)

Table 4 Effect of % of fract with L&P cuts. The solution time (in seconds) and the number of iterations (in parenthesis) are reported in arithmetic (AM) and geometric (GM) means

Best results emphasized with bold font

Table 5 Effect of L&P roundslimit. The solution time (inseconds) and the number ofiterations (in parenthesis) arereported in arithmetic (AM) andgeometric (GM) means

Group	AM Time (A	M iterations)	
	L&P rounds	limit	
	20	40	100
r01	37 (11)	51(10)	85(10)
r02	277 (44)	317(44)	324(42)
r03	534 (42)	599(42)	680(40)
r04	2404 (76)	2489(77)	2529(73)
sslp_5_25	52 (6)	59(3)	72(2)
sslp_5_50	65 (6)	72(4)	89(3)
sslp_15_45	532(163)	445(125)	298 (71)
sslp_10_50	140 (103)	193(122)	165(101)
AM (all instances)	458 (69)	478(63)	470(47)
GM (all instances)	123(110)	138(138)	147(147)

Best results emphasized with bold font

a negative impact on the solution time. These results support the assumption that strengthened epigraph cuts decrease the number of iterations and demonstrate a tradeoff between solving a mixed integer master problem versus solving linear strengthened cut generating problems. It can be concluded that the generation of strengthened epigraph cuts should be carefully controlled for every instance individually. We set the parameter L&P rounds limit to 20 in the following experiments.

Group	AM Time (AM	I Iterations)		
	Algorithm			
	DETEQ	BENDERS	EPIG	EPIG-STR
r01	3	25 (11)	34(11)	37(11)
r02	11	369(45)	416(45)	277 (44)
r03	38	716(40)	671(42)	534 (42)
r04	45	3852(72)	4489(69)	2404 (76)
sslp_5_25	29	39 (14)	40(14)	52(6)
sslp_5_50	52	49 (16)	50(16)	65(6)
sslp_15_45	313	683(191)	786(202)	532(163)
sslp_10_50	205	174(140)	189(141)	140(103)
AM (all inst.)	120	653(83)	739(86)	458 (69)
GM (all inst.)	15	92(92)	102(102)	123(110)

 Table 6
 Comparison of the algorithms. The solution time (in seconds) and the number of iterations (in parenthesis) are reported in arithmetic (AM) and geometric (GM) means

Best results among L-shaped methods emphasized with bold font. Number of iteration is not applicable for DETEQ, therefore, not provided

6.4 Efficiency of epigraph cut strengthening

Finally, the efficiency of the epigraph cut strengthening is evaluated by comparing the performance of the L-shaped algorithm without additional cuts (*BENDERS*), with plain epigraph cuts (*EPIG*), and with strengthened epigraph cuts (*EPIG-STR*). The two-phase method is applied for all tests. The MCFNDSD instances are solved with the multi-cut and the SSLP instances with the single-cut L-shaped method. In EPIG and EPIG-STR runs, the parameter L&P rounds limit is set to 20. We set the best parameters found in Sect. 6.3 for the EPIGR-STR runs: $\epsilon^{CGP} = 0.01$, % of fract with L&P cuts = 0.1, and CGP optimality cuts are appended to master problem in the tests with the SSLP instances. Results of solving the deterministic equivalent with CPLEX (*DETEQ*), are also provided for reference.

As can be observed in Table 6, the approach with strengthened epigraph cuts outperforms the other L-shaped variants when applied to the relatively complicated instances that need more iterations to be solved. The application of epigraph cuts and their strengthening does not seem to pay off for the smaller instances. Two and four r04 instances were not solved within the time limit with BENDERS and EPIG algorithms, respectively. Their solution time was set to the time limit. Non-aggregated detailed results for every test can be found in the Online Supplement.

To get better insights, the detailed profiling results for selected instances are provided in Table 7. Other tests in the test set show similar behavior. In this table, we report the number of iterations (*Iter.*), total wall-clock time (*Time*), including the wall-clock time for solving the MIP master problem (*T. Master*), and for the L&P cut generation (*T. L&P Cuts*). In addition, we provide the average wall-clock time per iteration (ϕ *T. Master*) that is calculated as *T.Master/Iter*. From the result on solution time we can conclude that the application of strengthened L&P cuts can provide two ben-

Ξ.	
le a	
san	
s the	
ng i	
eani	
ir m	
The	
abel.	
nce l	
instar	
/ery	
er ev	
pun	
ded	
irovi	
are p	
ons	
ensi	
Dim	
lds.	
ecor	
in s	
iven	
re g	
les a	
Tin	
ests.	
ted t	
elec	
the s	
for 1	
stics	
statis	
ner s	
d tin	
taile	
De	
ole 7	ble 1

Table 7Detailed timer statistics for the selecTable 1	sted tests. Times are give	en in seconds. D	imensions are provi	ded under every insta	nce label. Their meanir	g is the same as in
Instance	Algorithm	Iter.	Time	T. Master	øT. Master	T. L&P Cuts
r01_1_0_32	BENDERS	2	1.7	0.3	0.1	0.0
BV1:35 CV2:360 R1:0 R2:135 S:32	EPIGR	2	4.0	0.4	0.2	2.2
	EPIGR-STR	2	14.5	0.4	0.2	12.4
r02_1_0_32	BENDERS	139	1200.2	1191.3	8.6	0.0
BV1:35 CV2:900 R1:0 R2:285 S:32	EPIGR	138	1367.5	1354.6	9.8	3.6
	EPIGR-STR	138	667.0	614.7	4.5	45.5
r04_1_0_32	BENDERS	7	70.7	56.3	8.0	0.0
BV1:60 CV2:610 R1:0 R2:160 S:32	EPIGR	7	120.4	99.3	14.2	7.0
	EPIGR-STR	7	214.1	56.3	8.0	142.5
sslp_10_50_1000a	BENDERS	209	47.6	38.7	0.2	0.0
BV1:10 CV2:510 R1:1 R2:60 S:1000	EPIGR	194	47.8	38.7	0.2	0.1
	EPIGR-STR	173	72.5	58.0	0.3	5.7
sslp_10_50_1000b	BENDERS	181	78.4	6.69	0.4	0.0
BV1:10 CV2:510 R1:1 R2:60 S:1000	EPIGR	185	91.3	82.3	0.4	0.1
	EPIGR-STR	119	60.6	45.5	0.4	7.0
sslp_10_50_2000a	BENDERS	208	250.4	228.9	1.1	0.0
BV1:10 CV2:510 R1:1 R2:60 S:2000	EPIGR	191	227.0	207.2	1.1	0.3
	EPIGR-STR	142	182.8	146.0	1.0	17.8

continued	
Ð	
q	
Ta	

 $\underline{\textcircled{O}}$ Springer

Instance	Algorithm	Iter.	Time	T. Master	øT. Master	T. L&P Cuts
sslp_10_50_2000b	BENDERS	210	261.1	241.8	1.2	0.0
BV1:10 CV2:510 R1:1 R2:60 S:2000	EPIGR	210	285.9	266.0	1.3	0.3
	EPIGR-STR	140	193.3	155.8	1.1	18.1
sslp_15_45_15c	BENDERS	498	2079.6	2064.7	4.1	0.0
BV1:15 CV2:675 R1:1 R2:60 S:15	EPIGR	497	2189.2	2173.0	4.4	0.8
	EPIGR-STR	415	1527.1	1468.9	3.5	45.1
sslp_5_25_100b	BENDERS	13	59.3	50.1	3.9	0.0
BV1:5 CV2:130 R1:1 R2:30 S:100	EPIGR	14	62.5	53.0	3.8	0.1
	EPIGR-STR	5	80.6	19.0	3.8	52.4
sslp_5_50_100a	BENDERS	16	80.5	68.5	4.3	0.0
BV1:5 CV2:255 R1:1 R2:55 S:100	EPIGR	16	86.9	74.7	4.7	0.1
	EPIGR-STR	6	112.7	39.1	4.3	62.1
sslp_5_50_2000a	BENDERS	21	33.9	26.6	1.3	0.0
BV1:5 CV2:255 R1:1 R2:55 S:2000	EPIGR	24	36.6	29.7	1.2	0.1
	EPIGR-STR	6	37.3	10.1	1.1	20.9
sslp_5_50_2000b	BENDERS	16	25.1	18.3	1.1	0.0
BV1:5 CV2:255 R1:1 R2:55 S:2000	EPIGR	15	23.8	17.1	1.1	0.1
	EPIGR-STR	9	34.1	6.4	1.2	21.3

efits: a decrease of the number of iterations and/or a shorter average solution time per MIP master problem in every iteration (in comparison to the L-shaped method with plain epigraph cuts). No decrease of iterations is observable in the tests with the MCFNDSD instances. However, the master problem is often solved faster (see, for example, instance $r02_1_0_32$). When strengthening is applied to the SSLP instances, the algorithm converges after a smaller number of iterations and the master problem is sometimes solved in a shorter time compared to the standard L-shaped method. Of course, cut generation takes a considerable amount of time and the net change in time should be considered. For example, the decrease of iterations achieved in the tests with the instances sslp_5_25_100b is insufficient to result in a reduction of solution time.

Our implementation of the L-shaped method performs better than the commercial CPLEX solver only for tests on sslp_5_50 and sslp_10_50 groups. See, for example, instances sslp_10_50_1000a-b, sslp_10_50_2000a-b, sslp_5_50_2000a-b in Table 7. We explain this behavior by the fact that solely in these groups most of the instances have over 1000 scenarios. In addition, results in Table 1 in the Online Supplement indicate that the L-shaped method is almost always faster for instances with over 1000 scenarios as it was designed to overcome the curse of dimensionality (Powell and Topaloglu 2003). Additionally, CPLEX, as one of the leading commercial solvers, has advanced preprocessing and cutting plane techniques that are not implemented in our algorithm. We do not aim to compete with the commercial solver but to demonstrate that L&P cut strengthening has a potential upon a reasonable implementation.

7 Summary and outlook

The computational results demonstrate that the application of strengthened epigraph L&P cuts in the preprocessing phase of the L-shaped method can decrease the number of iterations and the solution time. The effectiveness varies for different instances and depends on the number and the accuracy of the generated cuts, which can be controlled by limiting the number of cut generation rounds, the number of considered fractional variables and the optimality tolerance of the cut generating problem. The main trade-off comes from the fact that in every iteration, a mixed-binary (master) problem is solved, while all problems that are solved during the cut generation round are linear though quite large.

It is efficient to append the optimality cuts constructed for solving the strengthened cut generating problem to the master problem in the single-cut L-shaped method. Too many of these cuts are generated during the multi-cut L-shaped method and, therefore, their usage in the master problem should be avoided.

Strengthened L&P cuts can also be integrated into the single-search-tree version of the L-shaped method for integer problems (see a Branch-and-Benders-cut method mentioned in Bodur et al. (2017); Gendron et al. (2016)). According to this approach, the branch-and-bound tree is initialized for the integer master problem once and domain or (strengthened) epigraph cuts are appended to the nodes during the solution

process. Such an implementation would require both theoretical and computational developments that we leave for future research.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10287-022-00426-y.

Acknowledgements The research of Csaba Fábián is supported by EFOP-3.6.1-16-2016-00006 "The development and enhancement of the research potential at John von Neumann University" project. The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability Raw data were generated at European University Viadrina. Derived data supporting the findings of this study are available from the corresponding author Pavlo Glushko on request.

Declarations

Conflict of interest The authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

A cuts for the stochastic programming master problem

A.1 Characterization of the plain epigraph cuts

The notation of Sect. 4 is to be used. The supplemental material referenced in the Appendix is available at https://doi.org/10.1007/s10287-022-00426-y.

The construction of lift-and-project cuts for the generic inequality system $Mx \ge d$ is described in Section 1.1 of the supplement. For the present master problem, the procedure with M and d is applied as defined in (10). Its dual-type characterization is the conjunctive inequality system below.

The cut $\zeta \vartheta + \boldsymbol{\alpha}^T \boldsymbol{x} \ge \beta$ is satisfied by the facets L_0, L_1 iff there exist $\boldsymbol{s}, \boldsymbol{t} \in$ I R^k, $\boldsymbol{s}, \boldsymbol{t} \ge \boldsymbol{0}, \quad \boldsymbol{u}, \boldsymbol{v} \in$ I R^m, $\boldsymbol{u}, \boldsymbol{v} \ge \boldsymbol{0}$ and $u_0, v_0 \in$ I R, $u_0, v_0 \ge 0$ such that

$$\begin{aligned}
 1^{T}s &= \zeta, \\
 -V^{T}s + A^{T}u - u_{0}e_{i} &\leq \alpha, \\
 1^{T}t &= \zeta, \\
 -V^{T}t + A^{T}v + v_{0}e_{i} \leq \alpha, \\
 w^{T}s + b^{T}u &\geq \beta, \\
 w^{T}t + b^{T}v + v_{0} \geq \beta,
 \end{aligned}$$
(18)

where e_i is the *i*th unit vector in I R^{*n*}. The above system is solvable in any non-trivial case. The feasible set is a non-empty cone.

A.2 The plain epigraph cut generating problem

Let $(\overline{\vartheta}, \overline{x}) \in L$ be a feasible solution of the master problem LP relaxation. Assume that $(\overline{\vartheta}, \overline{x})$ is not an integer feasible solution because we have $0 < \overline{x}_i < 1$ for some $i \in \mathcal{B}$. The aim is to find a cut that separates $(\overline{\vartheta}, \overline{x})$ from the integer feasible solutions. Deepest cuts are obtained by maximizing $\beta - (\overline{\vartheta}, \overline{x})^T(\zeta, \alpha)$ over the above defined cone, having added some normalization constraints. A natural means of normalization here is prescribing $\zeta = 1$.

The cut generating LP problem (plain epigraph CGP) is as follows. Decision variables of this problem will be: $\zeta \in I \mathbb{R}$, $\boldsymbol{\alpha} \in I \mathbb{R}^n$, $\beta \in I \mathbb{R}$ and $s, t \in I \mathbb{R}^k$, $\boldsymbol{u}, \boldsymbol{v} \in I \mathbb{R}^m$, $u_0, v_0 \in I \mathbb{R}$ with lower bounds $s, t \ge 0$ and $\boldsymbol{u}, \boldsymbol{v} \ge 0$, $u_0, v_0 \ge 0$. The constraints are: the inequality system (18), and normalization constraint $\zeta = 1$. The objective to maximized is $\beta - \overline{\vartheta} \zeta - \overline{\boldsymbol{x}}^T \boldsymbol{\alpha}$.

This ζ -normalized problem is still solvable provided k > 0, i.e., the epigraph model is non-trivial, and hence the matrix V^T has at least one column. So, it is assumed hereafter.

Since ζ is constant in this problem, the corresponding column vector can be eliminated from the matrix by shifting the right-hand-side vector.

A.3 Proof of Observation 1 in Sect. 4

Suppose that the objective of the plain epigraph CGP is still not bounded. Hence there exists a feasible ray $(s, u, u_0, t, v, v_0, \alpha, \beta)$ with positive objective value. Any feasible ray must satisfy $\mathbf{1}^T s = 0$, and together with the non-negativity of *s*, this implies $s = \mathbf{0}$. Similarly, we must have $t = \mathbf{0}$.

Figure 1 depicts the inequality system characterizing such a ray. The columns corresponding to *s* or *t* have been eliminated. The bottom constraint $-\overline{x}^T \alpha + \beta > 0$ represents non-negativity of the ray. Let's apply the characterization (6) with M = A, d = b and $\mathcal{P} = \mathcal{N}$. It states that the cut $\alpha^T x \ge \beta$ is valid for the facets $K \cap \{x \in I \mathbb{R}^n \mid -x_i \ge 0\}$ and $K \cap \{x \in I \mathbb{R}^n \mid x_i \ge 1\}$. The bottom constraint $-\overline{x}^T \alpha + \beta > 0$ states that \overline{x} is cut away by this cut. This verifies Observation 1.



Fig. 1 Inequality system of the feasible ray for unbounded epigraph CGP

References

Achterberg T (2007) Constraint integer programming. PhD thesis, Technische Universität Berlin

- Adulyasak Y, Cordeau JF, Jans R (2015) Benders decomposition for production routing under demand uncertainty. Oper Res 63(4):851–867
- Angulo G, Ahmed S, Dey SS (2016) Improving the integer l-shaped method. INFORMS J Comput 28(3):483–499
- Balas E (1979) Disjunctive programming. Ann Discrete Math 5:3-51
- Balas E, Ceria S, Cornuéjols G (1993) A lift-and-project cutting plane algorithm for mixed 0–1 programs. Math Progr 58(1):295–324
- Balas E, Ceria S, Cornuéjols G (1996) Mixed 0–1 programming by lift-and-project in a branch-and-cut framework. Manag Sci 42(9):1229–1246
- Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. Numer Math 4(1):238–252
- Bihlmaier R, Koberstein A, Obst R (2009) Modeling and optimizing of strategic and tactical production planning in the automotive industry under uncertainty. OR Spectr 31(2):311
- Birge JR, Louveaux F (2011) Introduction to stochastic programming, 2nd edn. Springer, Berlin
- Birge JR, Louveaux FV (1988) A multicut algorithm for two-stage stochastic linear programs. Eur J Oper Res 34(3):384–392
- Bodur M, Luedtke JR (2016) Mixed-integer rounding enhanced benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty. Manag Sci 63(7):2073–2091
- Bodur M, Dash S, Günlük O, Luedtke J (2017) Strengthened benders cuts for stochastic integer programs with continuous recourse. INFORMS J Comput 29(1):77–91
- Carøe CC, Schultz R (1999) Dual decomposition in stochastic integer programming. Oper Res Lett 24(1):37–45
- Carøe CC, Tind J (1997) A cutting-plane approach to mixed 0–1 stochastic integer programs. Eur J Oper Res 101(2):306–316
- Cordeau JF, Pasin F, Solomon MM (2006) An integrated model for logistics network design. Ann Oper Res 144(1):59–82
- Costa AM (2005) A survey on benders decomposition applied to fixed-charge network design problems. Comput Oper Res 32(6):1429–1450
- Costa AM, Cordeau JF, Gendron B, Laporte G (2012) Accelerating benders decomposition with heuristicmaster problem solutions. Pesquisa Operacional 32(1):03–20
- Crainic TG, Hewitt M, Maggioni F, Rei W (2016) Partial Benders decomposition strategies for two-stage stochastic integer programs. Université du Québec à Montréal, Iteruniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

- Fischetti M, Lodi A, Tramontani A (2011) On the separation of disjunctive cuts. Math Prog 128(1):205–230 Gade D, Küçükyavuz S, Sen S (2014) Decomposition algorithms with parametric Gomory cuts for two-stage
- stochastic integer programs. Math Prog 144(1):39-64
- Gendron B, Scutellà MG, Garroppo RG, Nencioni G, Tavanti L (2016) A branch-and-benders-cut method for nonlinear power design in green wireless local area networks. Eur J Oper Res 255(1):151–162
- Kılınç MR, Linderoth J, Luedtke J (2017) Lift-and-project cuts for convex mixed integer nonlinear programs. Math Programm Comput 9(4):499–526. https://doi.org/10.1007/s12532-017-0118-1
- McDaniel D, Devine M (1977) A modified benders' partitioning algorithm for mixed integer programming. Manag Sci 24(3):312–319
- Ntaimo L (2004) Decomposition algorithms for stochastic combinatorial optimization: Computational experiments and extensions. PhD thesis, The University of Arizona
- Ntaimo L (2013) Fenchel decomposition for stochastic mixed-integer programming. J Global Optim 55(1):141–163
- Powell WB, Topaloglu H (2003) Stochastic programming in transportation and logistics. Handbooks in operations research and management science stochastic programming. Elsevier, Amsterdam, pp 555– 635
- Qi Y, Sen S (2017) The Ancestral Benders' cutting plane algorithm with multi-term disjunctions for mixedinteger recourse decisions in stochastic programming. Math Programm Ser A and B 161(1-2)
- Rahmaniani R, Crainic T, Gendreau M, Rei W (2017a) A Benders decomposition method for two-stage stochastic network design problems. CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise
- Rahmaniani R, Crainic TG, Gendreau M, Rei W (2017) The benders decomposition algorithm: a literature review. Eur J Oper Res 259(3):801–817
- Saharidis GK, Boile M, Theofanis S (2011) Initialization of the benders master problem using valid inequalities applied to fixed-charge network problems. Expert Syst Appl 38(6):6627–6636
- Sen S, Higle JL (2005) The c3 theorem and a d2 algorithm for large scale stochastic mixed-integer programming: Set convexification. Math Program 104(1):1–20
- Sen S, Sherali HD (2006) Decomposition with branch-and-cut approaches for two-stage stochastic mixedinteger programming. Math Program 106(2):203–223
- Tang L, Jiang W, Saharidis GKD (2013) An improved benders decomposition algorithm for the logistics facility location problem with capacity expansions. Ann Oper Res 210(1):165–190
- Vadlamani S, Schweitzer D, Medal H, Nandi A, Eksioglu B (2018) A mixed-integer programming approach for locating jamming devices in a flow-jamming attack. Comput Oper Res 95:83–96
- Van Slyke R, Wets R (1969) L-shaped linear programs with applications to optimal control and stochastic programming. SIAM J Appl Math 17(4):638–663
- Weskamp C, Koberstein A, Schwartz F, Suhl L, Voß S (2019) A two-stage stochastic programming approach for identifying optimal postponement strategies in supply chains with uncertain demand. Omega 83:123–138
- Wesselman F (2010) Generating general-purpose cutting planes for mixed-integer programs. phdthesis, Universität Padeborn
- Wolf C, Koberstein A (2013) Dynamic sequencing and cut consolidation for the parallel hybrid-cut nested l-shaped method. Eur J Oper Res 230(1):143–156
- Wolf C, Fábián CI, Koberstein A, Suhl L (2014) Applying oracles of on-demand accuracy in two-stage stochastic programming: a computational study. Eur J Oper Res 239:437–448
- Zarandi MMF (2010) Using decomposition to solve facility location/fleet management problems. PhD thesis, University of Toronto
- Zhang M, Küçükyavuz S (2014) Finitely convergent decomposition algorithms for two-stage stochastic pure integer programs. SIAM J Optim 24(4):1933–1951

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.