European Journal of Operational Research 259 (2017) 801–817

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Invited Review

The Benders decomposition algorithm: A literature review

Ragheb Rahmaniani[a,c], Teodor Gabriel Crainic[a,b,*], Michel Gendreau[a,c], Walter Rei[a,b]

[a] CIRRELT - Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal H3C 3J7, Canada
[b] School of Management, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal H3C 3P8, Canada
[c] Department of Mathematics and Industrial Engineering, École Polytechnique de Montréal, P.O. Box 6079, Station Centre-ville, Montréal H3C 3A7, Canada

ARTICLE INFO

Article history:
Received 21 June 2016
Accepted 1 December 2016
Available online 9 December 2016

Keywords:
Combinatorial optimization
Benders decomposition
Acceleration techniques
Literature review

ABSTRACT

The Benders decomposition algorithm has been successfully applied to a wide range of difficult optimization problems. This paper presents a state-of-the-art survey of this algorithm, emphasizing its use in combinatorial optimization. We discuss the classical algorithm, the impact of the problem formulation on its convergence, and the relationship to other decomposition methods. We introduce a taxonomy of algorithmic enhancements and acceleration strategies based on the main components of the algorithm. The taxonomy provides the framework to synthesize the literature, and to identify shortcomings, trends and potential research directions. We also discuss the use of the Benders Decomposition to develop efficient (meta-)heuristics, describe the limitations of the classical algorithm, and present extensions enabling its application to a broader range of problems.

1. Introduction

It has been more than five decades since the *Benders Decomposition* (*BD*) algorithm was proposed by Benders (1962), with the main objective of tackling problems with *complicating variables,* which, when temporarily fixed, yield a problem significantly easier to handle. The BD method (also referred to as *variable partitioning*, Zaourar and Malick (2014), and *outer linearization*, Trukhanov, Ntaimo, and Schaefer (2010)) has become one of the most widely used exact algorithms, because it exploits the structure of the problem and decentralizes the overall computational burden. Successful applications are found in many divers fields, including planning and scheduling (Canto, 2008; Hooker, 2007), health care (Luong, 2015), transportation and telecommunications (Costa, 2005), energy and resource management (Cai, McKinney, Lasdon, & Watkins, 2001; Zhang & Ponnambalam, 2006), and chemical process design (Zhu & Kuno, 2003), as illustrated in Table 1.

The BD method is based on a sequence of projection, outer linearization, and relaxation (Geoffrion, 1970a, 1970b). The model is first projected onto the subspace defined by the set of complicating variables. The resulting formulation is then dualized, and the associated extreme rays and points respectively define the feasibility requirements (feasibility cuts) and the projected costs (optimality cuts) of the complicating variables. Thus, an equivalent formulation can be built by enumerating all the extreme points and rays. However, performing this enumeration and, then, solving the resulting formulation is generally computationally exhausting, if not impossible. Hence, one solves the equivalent model by applying a relaxation strategy to the feasibility and optimality cuts, yielding a *Master Problem* (*MP*) and a subproblem, which are iteratively solved to respectively guide the search process and generate the violated cuts.

The BD algorithm was initially proposed for a class of mixed-integer linear programming (*MILP*) problems. When the integer variables are fixed, the resulting problem is a continuous linear program (*LP*) for which we can use standard duality theory to develop cuts. Many extensions have since been developed to apply the algorithm to a broader range of problems (e.g., Geoffrion, 1972; Hooker & Ottosson, 2003). Other developments were proposed to increase the algorithm's efficiency on certain optimization classes (e.g., Costa, Cordeau, Gendron, & Laporte, 2012; Crainic, Hewitt, & Rei, 2014). In addition, BD often provides a basis for the design of effective heuristics for problems that would otherwise be intractable (Côté & Laughton, 1984; Raidl, 2015). The BD approach has thus become widely used for linear, nonlinear, integer, stochastic, multi-stage, bilevel, and other optimization problems, as illustrated in Table 2.

Fig. 1 depicts the increasing interest in the BD algorithm over the years. Despite this level of interest, there has been no comprehensive survey of the method in terms of its numerical and the-

* Corresponding author at: CIRRELT - Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal H3C 3J7, Canada.
E-mail address: TeodorGabriel.Crainic@cirrelt.net (T.G. Crainic).

http://dx.doi.org/10.1016/j.ejor.2016.12.005
0377-2217/

Table 1
Some applications of the Benders decomposition method.

	Reference	Application		Reference	Application
1	Behnamian (2014)	Production planning	17	Jiang et al. (2009)	Distribution planning
2	Adulyasak et al. (2015)	Production routing	18	Wheatley et al. (2015)	Inventory control
3	Boland et al. (2016)	Facility location	19	Laporte, Louveaux, and Mercure (1994)	Traveling salesman
4	Boschetti and Maniezzo (2009)	Project scheduling	20	Luong (2015)	Healthcare planning
5	Botton et al. (2013)	Survivable network design	21	Maravelias and Grossmann (2004)	Chemical process design
6	Cai et al. (2001)	Water resource management	22	Moreno-Centeno and Karp (2013)	Implicit hitting sets
7	Canto (2008)	Maintenance scheduling	23	Oliveira et al. (2014)	Investment planning
8	Codato and Fischetti (2006)	Map labeling	24	Osman and Baki (2014)	Transfer line balancing
9	Cordeau et al. (2006)	Logistics network design	25	Pérez-Galarce et al. (2014)	Spanning tree
10	Cordeau et al. (2001a)	Locomotive assignment	26	Pishvaee et al. (2014)	Supply chain network design
11	Cordeau et al. (2001b)	Airline scheduling	27	Rubiales et al. (2013)	Hydrothermal coordination
12	Corréa et al. (2007)	Vehicle routing	28	Saharidis et al. (2011)	Refinery system network planning
13	Côté et al. (2014)	Strip packing	29	Sen et al. (2015)	Segment allocation
14	Fortz and Poss (2009)	Network design	30	Bloom (1983)	Capacity expansion
15	Gelareh et al. (2015)	Transportation	31	Wang et al. (2016)	Optimal power flow
16	Jenabi et al. (2015)	Power management	32	Errico, Crainic, Malucelli, and Nonato (2016)	Public transit

Table 2
Examples of optimization problems handled via Benders method.

	Reference	Model		Reference	Model
1	Adulyasak et al. (2015)	Multi-period stochastic problem	16	Jenabi et al. (2015)	Piecewise linear mixed-integer problem
2	Behnamian (2014)	Multi-objective MILP	17	Wolf (2014)	Multi-stage stochastic program
3	Cai et al. (2001)	Multi-objective nonconvex nonlinear problem	18	Laporte et al. (1994)	Probabilistic integer formulation
5	Cordeau et al. (2001b)	Pure 0–1 formulation	19	Li (2013)	Large-scale nonconvex MINLP
6	Corréa et al. (2007)	Binary problem with logical expressions	20	Moreno-Centeno and Karp (2013)	Problem with constraints unknown in advance
7	Gabrel, Knippel, and Minoux (1999)	Step increasing cost	21	Bloom (1983)	Nonlinear multi-period problem with reliability constraint
8	Côté et al. (2014)	MILP with logical constraints	22	Osman and Baki (2014)	Nonlinear integer formulation
9	de Camargo et al. (2011)	Mixed-integer nonlinear program (MINLP)	23	Pérez-Galarce et al. (2014)	Minmax regret problem
10	Emami et al. (2016)	Robust optimization problem	24	Pishvaee et al. (2014)	Multi-objective possibilistic programming model
11	Fontaine and Minner (2014)	Bi-level problem with bilinear constraints	25	Raidl et al. (2014)	Integer, bilevel, capacitated problem
12	Fortz and Poss (2009)	Multi-layer capacitated network problem	27	Rubiales et al. (2013)	Quadratic MILP master problem and nonlinear subproblem
13	Gendron et al. (2014)	Binary problem with nonlinear constraints	28	Sahinidis and Grossmann (1991)	MINLP and nonconvex problems
14	Grothey et al. (1999)	Convex nonlinear problem	29	Harjunkoski and Grossmann (2001)	Multi-stage problem with logical and big-M constraints
15	O'Kelly et al. (2014)	MINLP with concave objective function and staircase constraint matrix structure			

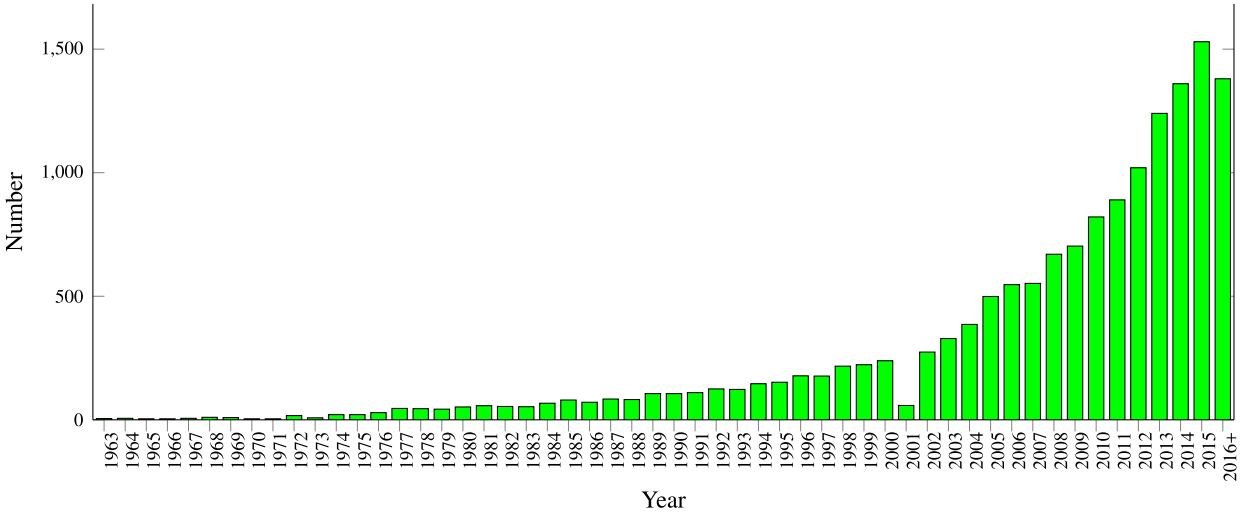

Fig. 1. Annual number of mentions of the Benders decomposition according to https://scholar.google.com/.

oretical challenges and opportunities. The now out-of-date survey by Costa (2005) reviews only applications to fixed-charge network design problems. The main goal of this paper therefore is to contribute to filling this gap by reviewing the current state-of-the-art, focusing on the main ideas for accelerating the method, discussing the main variants and extensions aimed to handle more general problems involving, e.g., nonlinear/integer/constraint programming subproblems, and identifying trends and promising research directions. Many different enhancement strategies were proposed to address the shortcomings of the BD method and accelerate it. This effort contributed significantly to the success of the method. We propose a taxonomy of the enhancement and acceleration strategies based on the main components of the algorithm: the decomposition strategy, the strategies to handle the MP and subproblem, and the strategies to generate solutions and cuts. The taxonomy provides the framework to classify and synthesize the literature and to identify relations among strategies and between these and the BD method.

The remainder of this paper is organized as follows. Section 2 presents the classical BD algorithm, the associated model selection criteria, and its relationship to other decomposition methods. Section 3 presents the proposed taxonomy, used to survey the acceleration strategies in Sections 4–7. Section 8 presents Benders-type heuristics, and Section 9 describes extensions of the classical algorithm. Finally, Section 10 provides concluding remarks and describes promising research directions.

2. The Benders decomposition method

We present in this section the classical version of the Benders algorithm (Benders, 1962). We review its extensions to a broader range of optimization problems in Section 9.

2.1. The classical version

We consider an MILP of the form

$$\text{Minimize} \quad f^T y + c^T x \tag{1}$$

$$\text{subject to} \quad Ay = b \tag{2}$$

$$By + Dx = d \tag{3}$$

$$x \geq 0 \tag{4}$$

$$y \geq 0 \quad \text{and integer}, \tag{5}$$

with complicating variables $y \in \mathbb{R}^{n_1}$, which must take positive integer values and satisfy the constraint set $Ay = b$, where $A \in \mathbb{R}^{m_1 \times n_1}$ is a known matrix and $b \in \mathbb{R}^{m_1}$ is a given vector. The continuous variables $x \in \mathbb{R}^{n_2}$, together with the y variables, must satisfy the linking constraint set $By + Dx = d$, with $B \in \mathbb{R}^{m_2 \times n_1}$, $D \in \mathbb{R}^{m_2 \times n_2}$, and $d \in \mathbb{R}^{m_2}$. The objective function minimizes the total cost with the cost vectors $f \in \mathbb{R}^{n_1}$ and $c \in \mathbb{R}^{n_2}$.

Model (1–5) can be re-expressed as

$$\min_{\bar{y} \in Y} \left\{ f^T \bar{y} + \min_{x \geq 0} \{c^T x : Dx = d - B\bar{y}\}, \right\} \tag{6}$$

where $\bar{y}$ is a given value for the complicating variables, which belongs to the set $Y = \{y | Ay = b, y \geq 0 \text{ and integer}\}$. The inner minimization is a continuous linear problem that can be dualized by means of dual variables π associated with the constraint set $Dx = d - B\bar{y}$:

$$\max_{\pi \in \Re^{m_2}} \{\pi^T (d - B\bar{y}) : \pi^T D \leq c\} \tag{7}$$

Based on duality theory, the primal and dual formulations can be interchanged to extract the following equivalent formulation:

$$\min_{\bar{y} \in Y} \left\{ f^T \bar{y} + \max_{\pi \in \Re^{m_2}} \{\pi^T (d - B\bar{y}) : \pi^T D \leq c\} \right\} \tag{8}$$

The feasible space of the inner maximization, i.e., $F = \{\pi | \pi^T D \leq c\}$, is independent of the choice of $\bar{y}$. Thus, if F is not empty, the inner problem can be either unbounded or feasible for any arbitrary choice of $\bar{y}$. In the former case, given the set of extreme rays Q of F, there is a direction of unboundedness r_q, $q \in Q$ for which $r_q^T (d - B\bar{y}) > 0$; this must be avoided because it indicates the infeasibility of the $\bar{y}$ solution. We add a cut

$$r_q^T (d - B\bar{y}) \leq 0 \qquad \forall q \in Q \tag{9}$$

to the problem to restrict movement in this direction. In the latter case, the solution of the inner maximization is one of the extreme points π_e, $e \in E$, where E is the set of extreme points of F. If we add all the cuts of the form (9) to the outer minimization problem, the value of the inner problem will be one of its extreme points. Consequently, problem (8) can be reformulated as:

$$\min_{\bar{y} \in Y} \quad f^T \bar{y} + \max_{e \in E} \{\pi_e^T (d - B\bar{y})\} \tag{10}$$

$$\text{subject to} \quad r_q^T (d - B\bar{y}) \leq 0 \qquad \forall q \in Q \tag{11}$$

This problem can easily be linearized via a continuous variable $\eta \in \Re^1$ to give the following equivalent formulation to problem (1–5), which we refer to as the Benders *Master Problem* (*MP*):

$$\min_{y, \eta} \quad f^T y + \eta \tag{12}$$

$$\text{subject to} \quad Ay = b \tag{13}$$

$$\eta \geq \pi_e^T (d - By) \qquad \forall e \in E \tag{14}$$

$$0 \geq r_q^T (d - By) \qquad \forall q \in Q \tag{15}$$

$$y \geq 0 \quad \text{and integer} \tag{16}$$

Constraints (14) and (15) are referred to as *optimality* and *feasibility* cuts, respectively. The complete enumeration of these cuts is generally not practical. Therefore, Benders (1962) proposed a relaxation of the feasibility and optimality cuts and an iterative approach. Thus, the BD algorithm repeatedly solves the MP, which includes only a subset of constraints (14) and (15), to obtain a trial value for the y variables, i.e., $\bar{y}$. It then solves subproblem (7) with $\bar{y}$. If the subproblem is feasible and bounded, a cut of type (14) is produced. If the subproblem is unbounded, a cut of type (15) is produced. If the cuts are violated by the current solution, they are inserted into the current MP and the process repeats.

Fig. 2 illustrates the BD algorithm. After deriving the initial MP and subproblem, the algorithm alternates between them (starting with the MP) until an optimal solution is found. To confirm the convergence, the optimality gap can be calculated at each iteration. The objective function of the MP gives a valid lower bound on the optimal cost because it is a relaxation of the equivalent Benders reformulation. On the other hand, if solution $\bar{y}$ yields a feasible subproblem, then the sum of both $f^T \bar{y}$ and the objective value associated to the subproblem provides a valid upper bound for the original problem (1)–(5).

2.2. Model selection for Benders decomposition

A given problem can usually be modeled with different but equivalent formulations. However, from a computational point of view, the various formulations may not be equivalent. Geoffrion and Graves (1974) observed that the formulation has a direct

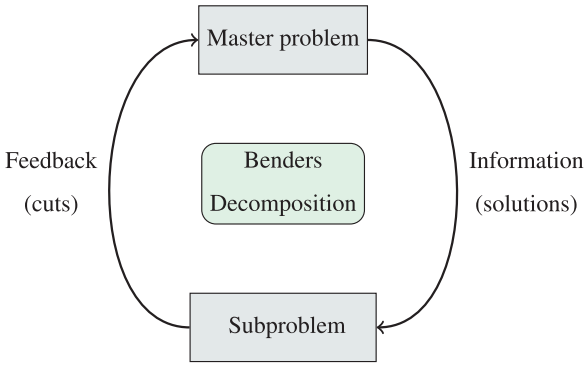

Fig. 2. Schematic representation of the Benders decomposition method.

impact on the performance of the BD. Magnanti and Wong (1981) demonstrated that a formulation with a stronger LP relaxation will have, in general, a better performance. This is because of the tighter root node and the smaller number of fractional variables and also because the generated cuts are provably stronger. Sahinidis and Grossmann (1991) proved that the BD method applied to a mixed integer nonlinear programming (NLP) formulation with a zero NLP relaxation gap requires only the cut corresponding to the optimal solution to converge. Cordeau, Pasin, and Solomon (2006) studied a stochastic logistic network design problem. They found that when the original formulation was strengthened with a set of valid inequalities (VIs), the performance of the BD method was considerably improved.

These observations confirm the importance of tight formulations in the context of the BD method. However, tighter formulations are often obtained by adding additional, problem-dependent, constraints. This may result in a more time-consuming subproblem, which may also exhibit a higher degree of degeneracy. Therefore, there must be a trade-off between the reduction in the number of iterations and the additional difficulty of the subproblem.

2.3. Relationship to other decomposition methods

The BD method is closely related to other decomposition methods for LP, such as Dantzig–Wolfe and Lagrangian optimization (see Lim (2010) for details). In particular, solving an LP by Dantzig–Wolfe decomposition is equivalent to applying the BD approach to its dual. The relationship is clear since Dantzig–Wolfe is particularly suitable for problems with complicating constraints, and the dual of these constraints will be the complicating variables in the BD method. Note that the subproblems are also equivalent in the two methods. The BD method is also equivalent to a cutting-plane method applied to the Lagrangian dual.

In integer programming, the situation is more complex, and there is no simple relationship among the decomposition methods. In contrast to Lagrangian relaxation and Dantzig–Wolfe decomposition, the BD method directly converges to an optimal solution to the MILP rather than to a relaxation of the problem; therefore, there is no need to embed it in a branch-and-bound framework. However, the classical BD approach cannot handle integrality requirements in the subproblems; variants have been proposed (Section 9).

Finally, there are close relationships between Benders cuts and various classical VIs (Magnanti, Mireault, & Wong, 1986). For instance, Costa, Cordeau, and Gendron (2009) demonstrated that cutset inequalities are essentially Benders feasibility cuts, while Benders feasibility cuts are not, in general, metric inequalities and require additional lifting procedures for conversion into metric inequalities. Therefore, the classical BD method has several numerical and theoretical limitations, for which various enhancement and acceleration strategies have been proposed.

3. Taxonomy of the enhancement strategies

A straightforward application of the classical BD algorithm may require excessive computing time and memory (Magnanti & Wong, 1981; Naoum-Sawaya & Elhedhli, 2013). Its main drawbacks include: time-consuming iterations; poor feasibility and optimality cuts; ineffective initial iterations; zigzagging behavior of the primal solutions and slow convergence at the end of the algorithm (i.e., a tailing-off effect); and upper bounds that remain unchanged in successive iterations because equivalent solutions exist.

Much research was dedicated to exploring ways to improve the convergence of the algorithm by reducing the number of iterations and the time required for each iteration. The former goal is aimed for by improving the quality of both the generated solutions and the cuts, and the latter by improving the solution procedure used to optimize the MP and subproblem in each iteration. The decomposition strategy that defines the initial MP and subproblem is another fundamental building block of the algorithm with significant consequences for its efficiency, as it determines both the difficulty of the problems and the quality of the solutions. We define therefore a four-dimension taxonomy, illustrated in Fig. 3, that captures all these factors.

The *decomposition strategy* specifies how the problem is partitioned to obtain the initial MP and subproblem. In a *classical* decomposition all the linking constraints and noncomplicating variables are projected out. In a *modified* decomposition these constraints and variables are partially projected to maintain an approximation of the projected terms in the MP.

The *solution procedure* concerns the algorithms used for the MP and subproblem. The *standard* techniques are the simplex method and branch-and-bound. They are treated as black-box solvers, and no attempt is made to adapt them to the characteristics of the problem or the convergence requirements of the BD algorithm. *Advanced* strategies exploit the structure of the MP and subproblem or the search requirements of the BD algorithm. For example, these strategies may aim to control the size of the problems or relax the requirement that they are solved to optimality at every iteration. We write “S/A” to indicate that standard techniques are used for the MP and advanced techniques are used for the subproblem. Similarly, we define A/A, A/S, and S/S.

The *solution generation* concerns the method used to set trial values for the complicating variables. The classical strategy is to solve the MP without modification (referred to as *regular MP*). *Heuristics*, an *alternative MP*, or an *improved MP* can be used to generate solutions more quickly or to find better solutions. *Hybrid* strategies can also be defined, e.g., one can use the regular MP to get an initial value for the master variables and then use heuristics to improve it.

The *cut generation* concerns the strategy used to generate optimality and feasibility cuts. Classically, this is done by solving the regular subproblem obtained from the decomposition. Other strategies reformulate the subproblem or solve auxiliary subproblems. The “C” and “I” symbols represent the *classical* and the *improved* strategies, respectively. For example, “C/I” indicates that the classical strategy is used to generate optimality cuts and the improved strategies are used to generate feasibility cuts. Sections 4–7 survey the strategies of each component of the taxonomy.

4. Decomposition strategies

Recent studies have presented various modified decomposition strategies. Crainic et al. (2014, 2016) emphasized that the BD method causes the MP to lose all the information associated with

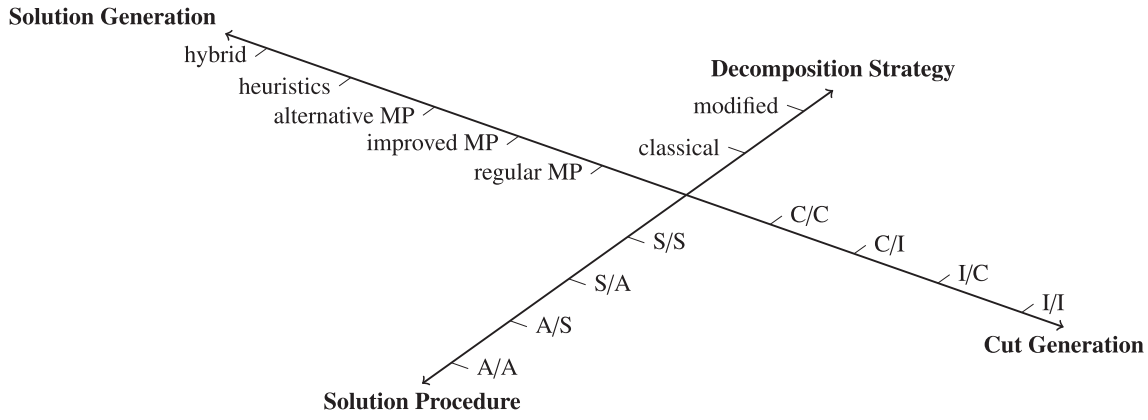

Fig. 3. Components of taxonomy.

the noncomplicating variables. This results in instability, erratic progression of the bounds, and a large number of iterations. Moreover, the problem structure associated with the linking constraints (3) is lost, and thus many classical VIs are not applicable. The authors proposed *Partial Benders Decomposition* strategies that add to the master explicit information from the scenario subproblems, by retaining or creating scenarios, or both. They obtained significant improvements in terms of number of generated cuts and computational time.

The *nonstandard decomposition* strategy of Gendron, Scutellà, Garroppo, Nencioni, and Tavanti (2014) is another interesting example of a modified decomposition. After decomposing the problem, the authors obtained a subproblem with integer variables and nonlinear constraints. To improve the convergence of the algorithm, the authors retained the projected variables in the MP but relaxed the integrality requirement. They also included in the MP a linear approximation of the nonlinear constraints. They observed a significant improvement in performance, although the difficulty of the MP was noticeably increased.

As indicated by the results mentioned above and the limited studies conducted in this dimension (4.17% of the cited articles in this review paper), further research into decomposition strategies is worthwhile, as modified decomposition strategies may significantly strengthen the relaxed MP. Such approaches are not only complementary to adding VIs, discussed in Section 6.2, but may also provide the opportunity to derive a wider range of, possibly stronger, VIs.

5. Solution procedure

The iterative solution of the MP and subproblem is a major computational bottleneck. In particular, the MP, an MILP formulation, is often lacking special structure, and is continually growing in size becoming more and more difficult to solve. Classically, the MP is solved to optimality via branch-and-bound, while the subproblem is handled with the simplex method. In this section, we survey the various alternatives that have been proposed. These strategies exploit the structure of the MP and subproblem or are designed to improve the convergence speed. Fig. 4 lists the strategies that we discuss.

5.1. MP level

It has often been reported that more than 90% of the total execution BD time is spent on solving the MP (Magnanti & Wong, 1981; Zarandi, 2010). The strategies proposed to partially alleviate this computational bottleneck, discussed in the next two subsections, either a) manage the size of the problem or b) solve it more efficiently.

5.1.1. Size management

In any optimal solution to the MP, the number of active constraints never exceeds the number of decision variables (Minoux, 1986). Thus, many of the generated cuts do not contribute to the convergence of the algorithm and merely slow it down (extra handling effort and memory limitations). Therefore, cut deletion or *clean-up strategies* are important, especially when multiple cuts are inserted into the MP at each iteration.

There is no reliable way to identify the useless cuts, so the clean-up strategies are heuristic (Ruszczyński, 2003). They usually inspect the slack values associated with each cut, a cut with a relatively high slack value over some predetermined number of iterations being a good candidate for removal. One can avoid the regeneration of eliminated cuts (and prevent the possible cycling of the algorithm) by keeping them in a pool and reinserting them into the MP whenever they are violated by the current solution. Cuts should be removed infrequently because constraint removal has a disturbing impact on off-the-shelf optimization solvers, especially when re-optimization techniques are used for faster solution of the MP; see Geoffrion (1972) and Pacqueau, Francois, and Le Nguyen (2012) for implementation details.

Sometimes, multiple cuts are available for insertion but not all of them are worth adding. It is actually important to add cuts cautiously to avoid an explosion in the size of the MP. Holmberg (1990) defined *cut improvement* as follows: an optimality cut provides cut improvement if it is new and may be active in an optimal solution. Rei, Cordeau, Gendreau, and Soriano (2009) used this definition when selecting solutions in order to generate cuts. They stated that a solution yields cut improvement if its value in the current MP is strictly smaller than the best known upper bound. Yang and Lee (2012) selected tighter (feasibility) constraints from a set of available cuts by measuring their distance from different interior points. After choosing a first cut from the pool, they added other cuts for which the “difference measure” from the first cut was greater than a user-defined threshold. This yielded considerable savings in computational time and number of iterations.

In summary, one can control the size of the MP by removing unnecessary cuts or avoiding the insertion of an available cut. However, these strategies are not often used. Considering the articles cited in the present paper, only 3.13% of them used these strategies. One reason is that simultaneously inserting many cuts into the MP reduces the number of iterations, and this often compensates for the increase in the difficulty of addressing the problem. Another reason is that these techniques are heuristic, so they

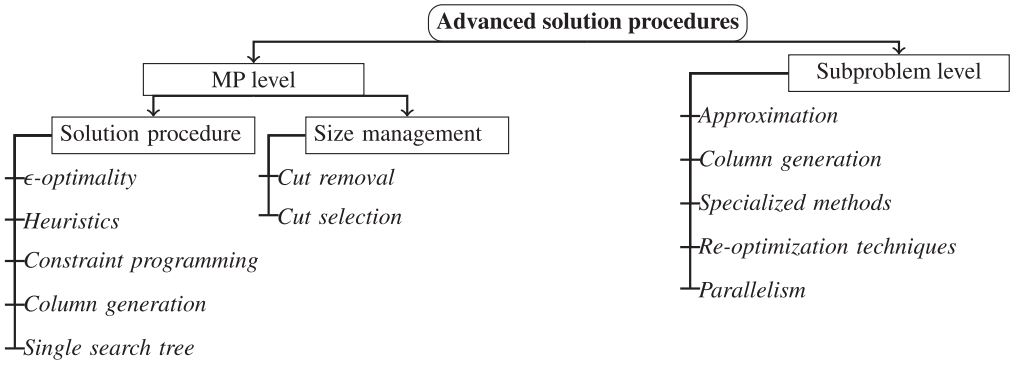

Fig. 4. Advanced solution procedures.

may remove necessary cuts or add cuts that prove unhelpful. There is a need for further research in this area.

5.1.2. Algorithm

It is not necessary to solve the MP to optimality at every iteration in order to achieve global convergence. Some researchers have focused on quickly finding suboptimal solutions. Others have taken advantage of the structure of the MP to solve it more efficiently.

Geoffrion and Graves (1974) were the first to address the MP's computational difficulties. They observed that it is not necessary to solve the MP to optimality at every iteration to produce valid cuts. In fact, there is no incentive to do so at the beginning of the algorithm because the relaxation is weak. They solved the MP to *ϵ-optimality* at each iteration, with ϵ decreasing as the algorithm proceeds to ensure global convergence. Lin and Üster (2014) terminated the algorithm whenever the MP could not produce feasible solutions in the presence of an ϵ-optimality constraint, indicating that the upper bound lies within ϵ% of optimality. Kewcharoenwong and Üster (2014) obtained an encouraging speedup with this approach in the context of fixed-charge relay network design in telecommunications.

An alternative is to solve the MP via *(meta-)heuristics*, which not only reduces the time but also allows the generation of multiple cuts per iteration, yielding faster improvement of the lower bound (Raidl, 2015). This strategy may lead, however, to worse bounds and a lack of control, which could prevent the generation of necessary cuts. For an MILP, the bounds may be worse than those of the LP relaxation of the original problem (Holmberg, 1994). Thus, the MP must be solved, at certain iterations, to optimality in order to ensure global convergence. However, fewer of these iterations are usually needed (Poojari & Beasley, 2009).

Constraint Programming (*CP*) is another possible approach. In a workforce scheduling application, Benoist, Gaudin, and Rottembourg (2002) showed that CP can be a better choice than mixed integer programming (MIP) solvers, because of its greater ability to handle special constraints. Similarly, Corréa, Langevin, and Rousseau (2007) considered the simultaneous scheduling and routing of automated guided vehicles, using CP to solve the scheduling MP and an MIP solver to handle the routing subproblem. They observed improvements of several orders of magnitude when the MP was solved by CP.

A few researchers have applied *Column Generation* (*CG*) to the MP to handle certain structures more effectively, aiming for tighter bounds at the root node of the branch-and-bound tree. Cordeau, Stojković, Soumis, and Desrosiers (2001b) proposed using BD to handle linking constraints in a simultaneous aircraft routing and crew scheduling problem. They formulated an aircraft routing MP and a crew pairing subproblem. Because of their special structure, the linear relaxation of both problems were handled by CG (see Mercier, Cordeau, and Soumis, 2005, for a similar application). Restrepo, Gendron, and Rousseau (2015) applied BD to solve a multi-tour activity scheduling problem; the MP was handled by CG embedded in a branch-and-price framework because of its large number of variables. The integration of CG into the BD framework appears theoretically challenging because of the simultaneous addition of rows and columns that are often interdependent. The discussion of this issue is beyond the scope of this paper; see Muter, Birbil, and Bülbül (2015) for further information.

In the classical BD, one MP (an MILP) is solved to optimality at each iteration. Each time, a new branch-and-bound tree is built and considerable time is likely spent revisiting candidate solutions that have been eliminated earlier. One can instead build a *single search tree* and generate valid cuts for the integer (and fractional) solutions encountered inside the tree, attaining the same optimal solution. This strategy, often referred to as *Branch-and-Benders-cut* (*B&BC*), yielded promising results (e.g., Fortz & Poss, 2009; Fischetti, Salvagnin, & Zanette, 2010; de Camargo, de Miranda, & Ferreira, 2011; Taşkın & Cevik, 2013; de Sá, de Camargo, & de Miranda, 2013; Gendron et al., 2014; Crainic et al., 2014; Pérez-Galarce, Álvarez-Miranda, Candia-Véjar, & Toth, 2014). In addition to the numerical superiority of a modern implementation in comparison with the classical one, Naoum-Sawaya and Elhedhli (2013) showed that B&BC can make better use of the re-optimization tools of MILP solvers. Various strategies can be used to produce the cuts. For example, one can generate cuts in all feasible nodes or only when new incumbent solutions are found. It is necessary to establish a trade-off between cut generation and branching effort. For instance, Botton, Fortz, Gouveia, and Poss (2013) studied when to generate cuts, their results indicating that generating cuts at every node of the search tree is inefficient because too many cuts are added and too much time is spent solving the subproblems. Better performance was achieved by finding as many violated cuts as possible at the root node and subsequently checking for violated Benders inequalities only at integer nodes.

We complete this section with two remarks. First, solving the MP via approximations, heuristics, or a single search tree may not be superior to the classical cutting-plane implementation, especially in applications where solving the MP takes significantly less time than the dual component of the algorithm. The modified approaches may then enumerate many solutions that are usually ignored by classical implementations. This is not the case, however, for most combinatorial optimization problems, for which the acceleration strategies we discussed are the most popular to handle the MP. Considering the cited researches throughout this article, 33.33% of them implemented one of these strategies. In particular, the single search tree has recently received considerable attention. This strategy leads to interesting research perspectives regarding the cut-generation strategies, the branching rules, the

node selection, and the pruning strategies that have not been fully explored.

Second, CP has been shown to be better than MIP techniques for the MP if there are special constraints such as "all-different" constraints, logical relations, arithmetic expressions, integer division, and expressions that index an array of values by a decision variable. Finally, one can use alternative formulations for the MP to generate solutions more quickly, reducing the number of iterations. This topic is addressed in Section 6.1.

5.2. Subproblem level

The subproblem can be large, inherit complex features, or decompose into an excessive number of smaller subproblems. Various strategies have been proposed to solve the subproblem more effectively.

The solution of the subproblem may be extremely complex because it is a large-scale LP. Zakeri, Philpott, and Ryan (2000) showed that suboptimal solutions of the dual subproblem can be used to generate useful valid cuts. The authors observe that these inexact cuts are computationally less expensive and produce good results. In the same situation, commercial solvers (e.g., CPLEX) prefer the interior point approach to the simplex method. Thus, the BD method may converge to an incorrect solution, since the cuts are not necessarily associated with extreme points of the dual polyhedron (Yang & Lee, 2012). However, given the condition introduced by Zakeri et al. (2000) for inexact Benders cuts, convergence can still be guaranteed.

CG is another effective approach for large-scale linear subproblems with special structure (Cordeau et al., 2001b). Mercier et al. (2005) showed that for large subproblems complete enumeration is impossible, but that variables can be iteratively generated via CG. Similar applications of CG within a BD framework can be found in Cordeau, Soumis, and Desrosiers (2001a), Mercier and Soumis (2007), and Papadakos (2009).

Subproblems with special structure can often be solved efficiently. One can derive a closed-form solution or apply specialized algorithms. Fischetti, Ljubic, and Sinnl (2016) observed that the subproblem for the uncapacitated facility location problem can reduce to a knapsack problem, which has a closed-form solution. Similarly, Randazzo, Luna, and Mahey (2001) obtained a series of trivial network flow subproblems with a closed-form solution. Contreras, Cordeau, and Laporte (2011) obtained a semi-assignment problem for each commodity in their hub location application; these problems could be solved more efficiently by a specialized method compared to an LP solver. Kouvelis and Yu (1997) derived shortest-path subproblems that were solved with Dijkstra's algorithm.

The decomposition sometimes yields several independent subproblems. One may consider solving only a subset of the subproblems at each iteration, especially at the beginning of the algorithm. To the best of our knowledge, there is no such strategy for combinatorial optimization problems, but Fábián (2000) has studied a similar idea for convex programming problems. The algorithm initially calculates rough estimates of the function values and gradients. As it proceeds, the calculations become more accurate.

In some applications, many subproblems are similar. Considering that the algorithm updates only the objective function of the dual subproblem between two consecutive iterations, one can exploit these similarities using re-optimization techniques: given the solution to one subproblem, the next can be optimized in just a few iterations (see, e.g., Birge & Louveaux, 1997; Vladimirou, 1998). However, since the algorithm updates only the objective function of the dual subproblem between one iteration and the next, further investigation of re-optimization techniques is required.

When there are many subproblems, parallel computing techniques are often used: the subproblems are solved in parallel on different processors. One processor, the *master*, usually solves the MP and coordinates the other processors, the *slaves*, which solve the subproblems. The primal solution of the MP is passed to the slave processors, and the objective function and the dual information obtained from solving the subproblems is returned to the master processor. Experiments have shown that this strategy is effective (e.g., Ariyawansa & Hudson, 1991; Nielsen & Zenios, 1997; Linderoth & Wright, 2003; Wolf & Koberstein, 2013; Pacqueau et al., 2012). The literature discusses some of the algorithmic challenges of this approach. For example, Linderoth and Wright (2003) implemented asynchronous communications in which the MP is re-optimized as soon as λ% of the cuts are received. Dantzig, Ho, and Infanger (1991) used dynamic work-allocation: the next idle processor gets the next subproblem based on a first-in-first-out strategy until all the subproblems are solved and the MP can be recomputed with the added cuts. Nielsen and Zenios (1997) exploited the structural similarities of the subproblems, applying an interior point algorithm on a fine-grained parallel machine. Vladimirou (1998) implemented a partial-cut aggregation approach to reduce the communication overheads. Chermakani (2015) observed that when the number of subproblems is considerably larger than the number of available processors, so that some subproblems must be solved sequentially, it may be better to aggregate some of the subproblems.

In summary, the simplex method is the most widely used algorithm for the subproblem. However, when the subproblem has special structure, specialized algorithms are a better option. In either case, when the decomposition yields several independent subproblems, parallelization is the method of choice. There has been limited investigation of approximation and heuristic methods. Yet, when the subproblem is a large-scale LP that cannot be further decomposed and has no special structure, heuristics may yield considerable speed-up. Constraint programming has also been a prominent technique to solve subproblems with particular structures. This is discussed in Sections 9.1 and 9.2. In terms of statistics, 31.25% of the cited works in this article implemented one of the outlined acceleration strategies, among which parallelization and specialized algorithms are the most popular techniques.

6. Solution generation

The quality of the solutions for the set of complicating variables directly determines the number of iterations, as they are used to generate cuts and bounds. These solutions are traditionally found by exactly or approximately solving the regular MP. Three approaches have been proposed to improve the quality of the solutions or generate them more quickly: (1) using alternative formulations, (2) improving the MP formulation, and (3) using heuristics to independently generate solutions or to improve those already found. Fig. 5 lists the techniques that we discuss. Note that, the strategies are not mutually exclusive, and hybrid approaches may work well. A complete presentation of the hybrid strategies is beyond the scope of this article, however, since the appropriate combination of strategies is problem-dependent.

6.1. Alternative formulation

Altering the MP, albeit temporarily, provides the means to address two main drawbacks in addressing it: (1) slow generation of solutions that may be poor-quality and (2) instability. The *two-phase* approach of McDaniel and Devine (1977) and the *cross decomposition* of Van Roy (1983) address the first drawback. The former uses an approximation of the MP to generate solutions more

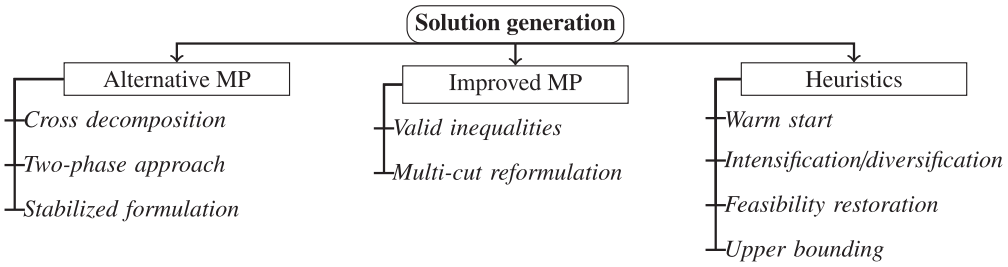

Fig. 5. Strategies to generate solutions for the set of complicating variables.

quickly, while the latter uses an alternative MP to generate potentially better solutions.

McDaniel and Devine (1977) showed that valid Benders cuts can be obtained from the solutions to the LP relaxation of the MP. They applied the BD algorithm in two phases. In the first phase, the linear relaxation of the MILP MP is used to quickly generate solutions and tighten the relaxed MP, and in the second phase the integrality requirements are reintroduced and the solution process continues. The cross-decomposition algorithm (Van Roy, 1983) exploits the structure of both the primal and dual problems, combining the advantages of Lagrangian relaxation and BD. It alternates between two steps: (i) for a given y, the Benders subproblem (7) is solved to get the dual multipliers π, and (ii) for a given π, the Lagrangian subproblem is solved to produce a new y. After this alternation is terminated, the solutions of both the Benders and Lagrangian subproblems are used to generate new cuts for the Benders or Lagrangian MP. Again, convergence to optimality can only be ensured by periodically solving the Benders MP. However, doing this less often accelerates the solution process.

Instability is another widely recognized drawback of the regular MP (Birge & Louveaux, 1997; Zaourar & Malick, 2014), as it can yield excessively slow convergence. Two reasons are evoked for this phenomenon, the large initial steps and the excessive oscillation that occurs as the algorithm approaches the optimal solution (Rubiales, Lotito, & Parente, 2013). Several studies aimed to mitigate this undesirable behavior by adding constraints to the MP or changing the objective function. Regularized decomposition, trust-region, and level decomposition strategies have been used to obtain a stabilized MP formulation.

Regularized decomposition was introduced by Ruszczyński (1986) and extended by Ruszczyński and Świętanowski (1997). A quadratic term is added to the MP objective function to keep the solutions close to the current reference point, which is updated whenever certain conditions are satisfied. Computational results indicate that the method is efficient although it solves a quadratic problem rather than a linear one, since it decreases the number of expensive iterations (Ruszczyński & Świętanowski, 1997). The trust-region method can be thought of as a hypercube around a reference solution at iteration k, denoted y^k. The next solution must stay within this "trusted region." This is usually achieved by adding the constraint $||y - y^k||_\infty \leq \Delta$ to the MP; the reference point y^k and the trust-region bound Δ will be updated during the algorithm (see Linderoth and Wright, 2003, for an implementation, computational results, and a proof of convergence). Level decomposition was developed by Lemaréchal, Nemirovskii, and Nesterov (1995) for nonsmooth convex optimization and adapted to stochastic programming by Fábián and Szőke (2007). Its goal is to dampen the zigzagging behavior of the BD method with respect to the master solutions. This is achieved by solving a modified MP that minimizes the Euclidean distance of its solution y from the previous solution y^k (initialized to y^0), while ensuring that the approximate value of the next solution is not greater than a convex combination of the current best lower and upper bounds. A favorable comparison of this method with the standard BD method, regularized decomposition, and the trust-region approach was carried out by Zverovich, Fábián, Ellison, and Mitra (2012).

The above stabilization techniques may not be directly applicable to combinatorial optimization contexts. Santoso, Ahmed, Goetschalckx, and Shapiro (2005) demonstrated that a trust region with the ℓ_2- or ℓ_∞-distance either yields a quadratic MILP or is not meaningful for a 0–1 MILP MP. Thus, they used a trust region that bounds the Hamming distance between the current MP solution and the previous one. However, since convergence cannot be ensured in the presence of this constraint, they used it only in the early iterations. Oliveira, Grossmann, and Hamacher (2014) observed an acceleration of up to 46% with the approach of Santoso et al. (2005). van Ackooij, Frangioni, and de Oliveira (2015) instead stabilized the BD method via a proximal term (a trust region in the ℓ_1 norm) or a level constraint rather than adding quadratic terms to the objective function. In continuous problems the quadratic proximal term provides second-order information to the MP, but its impact in the combinatorial case tends to be insignificant. The authors showed the potential of stabilization techniques, particularly for large and difficult instances. Zaourar and Malick (2014) studied quadratic stabilization. This approach, common in the context of constraint decomposition methods, could drastically reduce the number of cuts, especially feasibility cuts. The computational time may not decrease, however, due to complexity of the method.

In summary, an alternative master formulation is used in more than 34% of the works cited in this paper. The two-phase approach has become one of the most popular acceleration strategies for problems with time-consuming integer MPs (see e.g., Angulo, Ahmed, & Dey, 2014; Rei et al., 2009; Papadakos, 2008; Cordeau et al., 2006; Costa, 2005; de Sá et al., 2013; Sen, Krishnamoorthy, Rangaraj, & Narayanan, 2015). Algorithms based on the single-search-tree strategy often use this approach to tighten the root node and reduce the size of the tree (see e.g., Botton et al., 2013). Cross-decomposition has received much less attention, although a recent study by Mitra, Garcia-Herreros, and Grossmann (2016) has demonstrated that it can be superior to the BD method when the underlying LP relaxation is weak. Stabilization techniques significantly reduce the number of iterations, but the cost of each iteration may increase because of the additional complexities that they introduce in the MP. This has usually prevented the use of stabilization techniques in combinatorial contexts.

Inexact methods offer a simple and efficient stabilization tool that may lead to a considerable time reduction. The success of these methods highlights the potential of heuristics, local branching in particular, for solving the MP or more thoroughly exploring the neighborhood of the current solution to partially dampen the oscillations of the primal solutions.

Finally, as the instability of the classical BD method, especially in the early iterations, is mainly the result of a weak MP, strategies that strengthen the relaxation can mitigate the chaotic behavior. These strategies are the subject of the following subsection.

6.2. Improving the master formulation

After the projection and relaxation steps, the MP has no information on the subproblems and provides a weak approximation of the original feasible region. Therefore, we can expect erratic progression of the bounds and ineffective initial iterations. We can partially overcome these drawbacks by modifying the decomposition (see Section 4). The strategies discussed in this section will further improve the convergence rate (see e.g., Crainic, Hewitt, & Rei, 2016; 2014).

One way to strengthen the MP is to add VIs. Naoum-Sawaya and Elhedhli (2010) observed that this can significantly reduce the solution time and the number of generated cuts (feasibility cuts in particular). As a result, a wider range of instances can be solved. Saharidis, Boile, and Theofanis (2011) applied the BD method to a refinery system, adding two groups of VIs to the initial MP. The total time reduction ranged from 26% to 76%. Tang, Jiang, and Saharidis (2013) used VIs to improve the initial lower bound by 21.39% to 208.95%. In addition, the number of instances solved increased by 30%. Adulyasak, Cordeau, and Jans (2015) studied a production routing problem and added lower-bound lifting inequalities to improve the initial lower bounds and solutions. These cuts provide information about the part of the original objective function that has been removed. The authors observed a significant reduction in the time, the optimality gap and the number of explored nodes. For other uses of VI see Taşkın and Cevik (2013), Kewcharoenwong and Üster (2014), Pishvaee, Razmi, and Torabi (2014), Jenabi, Ghomi, Torabi, and Hosseinian (2015), Emami, Moslehi, and Sabbagh (2016), and Jeihoonian, Zanjani, and Gendreau (2016).

Feasibility cuts are undesirable because they do not improve the lower bound. In some applications, one can avoid unboundedness of the dual subproblem by including a set of VIs that exclude the infeasible solutions (see, e.g., Geoffrion & Graves, 1974; Birge & Louveaux, 1997; Contreras et al., 2011; de Sá et al., 2013). This avoids the burden of deriving the feasibility cuts. It can also reduce the cost of solving the MP, since the addition of both optimality and feasibility cuts makes it more difficult to solve (Wu, Hartman, & Wilson, 2003).

Once we fix the complicating variables, it may be possible to make the decisions in the remaining problem independently. Thus, multiple smaller subproblems can be solved separately. The classic BD method inserts a single cut into the MP per iteration by aggregating the dual information gathered from all the subproblems (Van Slyke & Wets, 1969). An alternative approach is to add a cut for each subproblem. This strategy, often referred to as *multi-cut reformulation*, generally outperforms the single-cut approach: it strengthens the MP more quickly and prevents the loss of information in the aggregation step (see, e.g., Contreras et al., 2011; Tang et al., 2013; Pishvaee et al., 2014; Sen et al., 2015; Jenabi et al., 2015). The size of the MP grows more rapidly, however, and the trade-off between the number of iterations and the computational time is problem-dependent. Birge and Louveaux (1997) gave the rule of thumb that the multi-cut is generally preferable when the number of subproblems is not much larger than the size of the dimension space of the master variables. Trukhanov et al. (2010) explored *partial cut aggregation*, in which the subproblems are divided into $|D|$ clusters and a cut is added for each cluster. They concluded that the best performance is attained for $1 < |D| < |S|$, where $|S|$ is the number of subproblems. They did not offer a specific strategy for the clustering. Brandes (2011) showed that clustering methods, in particular k-means and hierarchical clustering, can reduce the number of major iterations.

In summary, it is essential to strengthen the relaxed MP, and VIs are a powerful tool for this. Multi-cut reformulation is useful when the subproblem can be further decomposed into smaller problems. Using both strategies will often be more efficient. It should be noted that the reviewed strategies in this section have been used in 37.50% of the cited articles in this review paper. Finally, heuristics can be used to quickly tighten the MP by generating a set of initial cuts or multiple cuts per iteration. We discuss these strategies in the following subsection.

6.3. Heuristics

Numerous modifications may be needed to develop an efficient and competitive BD method (O'Kelly, Luna, Camargo, & Miranda, 2014). Many researchers therefore apply heuristic procedures to generate solutions or, as a subordinate method, improve previously generated ones (Botton et al., 2013; Gelareh, Monemi, & Nickel, 2015; Kewcharoenwong & Üster, 2014; Oliveira et al., 2014; Taşkın & Cevik, 2013).

Heuristics are widely used as a warm-start strategy to generate an initial set of tight cuts to strengthen the relaxed MP. Obviously, the selected heuristic should be appropriate for the problem at hand (Contreras et al., 2011). Lin and Üster (2014) observed that the initial selection of cuts is important in the context of wireless network design. They proposed a simple heuristic to generate feasible solutions and a set of good initial cuts. Easwaran and Üster (2009) used a tabu search as a warm-start meta-heuristic for a supply-chain network design problem; the convergence rate and the size of the instances solved were considerably increased. A different warm-start approach is to generate particular solutions. Randazzo et al. (2001) applied a shortest-path algorithm to obtain a special feasible solution to their local-access uncapacitated network design problem. Papadakos (2008) showed that convergence can be significantly improved when the algorithm starts from an initial point that lies in the interior of the MP solution domain rather than the initial MP solution itself.

Heuristics are also used to explore the neighborhood of the current MP solution as an intensification/diversification strategy. Rei et al. (2009) use a local branching heuristic to simultaneously improve the lower and upper bounds. They apply a limited number of local branching steps to either determine that the neighborhood contains no feasible solutions to the original problem or provide a pool of high-quality and diverse solutions. When the neighborhood is infeasible, they exclude the infeasible region by adding combinatorial cuts, which are a better alternative to classical feasibility cuts. When a pool of solutions is found, they are used to generate multiple optimality cuts. These cuts reduce the number of major iterations and cause the lower bound to increase more quickly. This strategy has also been applied to the closed-loop supply chain problem (Jeihoonian et al., 2016) and the sustainable supply chain network design problem (Pishvaee et al., 2014). Costa et al. (2012) gave general guidelines for the application of heuristics within the two-phase approach of McDaniel and Devine (1977). The goal is to quickly generate extra cuts associated with the heuristic feasible or infeasible solutions to reduce the need for integer iterations. The authors suggested using simple heuristics after each regular iteration of the BD method or whenever the incumbent solution is updated. The gains should compensate for the additional time spent on the heuristics.

Heuristics can be used to alleviate undesirable properties of the MP solutions. Wu et al. (2003) avoided the generation of feasibility cuts because they made the solution of the MP more expensive. It is possible in their application to acquire additional supply capacity at arbitrarily high prices, but convergence may be slow because of the side effects of big-M coefficients. The authors applied a heuristic to restore the feasibility of the solutions. The heuristic shifts excess demand from an infeasible subproblem to another source (i.e., to a different subproblem) so that a feasible solution can be found quickly. Emami et al. (2016) used a heuristic to extract

feasible solutions from infeasible MP solutions, considerably improving convergence.

The upper bounds obtained from heuristics are often better than those obtained by the BD method itself, especially in the early stages of the algorithm. Roussel, Ferland, and Pradenas (2004) successfully accelerated the BD method by applying tabu search to the original formulation to improve the upper bound. Santoso et al. (2005) stated that when the algorithm approaches an optimal solution, the various incumbent solutions differ in variables that have a small impact on the objective function, and so the upper bound changes little. They found that a simple fix-and-optimize heuristic can yield a considerable acceleration. Improving the upper bound will also impact other parts of the algorithm. For instance, Contreras et al. (2011) observed that their heuristic not only improved overall convergence but also found better upper bounds that help with the reduction testing procedures. Pérez-Galarce et al. (2014) examined the interaction between the incumbent solution and the branching efforts. To improve the performance of their B&BC algorithm, the authors used a heuristic to enhance the incumbent solution.

In summary, heuristics are an important component of acceleration strategies. They are used in more than 25% of the cited articles in this review paper. Heuristics, even simple ones, can generate high-quality initial solutions and cuts, repair infeasible solutions, improve the quality of the MP solutions, reduce the computational cost of the MP and subproblem, and generate multiple cuts per iteration. Moreover, the BD method can be used to design effective heuristics; see Section 8.

7. Cut generation

The number of iterations is closely related to the strength of the cuts, i.e., the values selected for the dual variables. Researchers have explored ways to select or strengthen the traditional feasibility and optimality cuts or to generate additional valid cuts.

Magnanti and Simpson (1978) and Magnanti and Wong (1981) were the first to consider the degeneracy of the subproblems, when the dual subproblem has multiple optimal solutions that do not yield cuts of equal strength. Hence, to find the strongest cuts, the solution of the dual subproblem must be judiciously chosen at each iteration. Magnanti and Wong (1981) selected a dual solution that dominates other possible cuts in terms of Pareto-optimality. A Pareto-optimal solution produces the maximum value at a core point $\hat{y}$, which is required to be in the relative interior of the convex hull of the subregion defined by the MP variables. After solving the regular dual subproblem (7), the authors solve an auxiliary subproblem of the form (17) to find the Pareto-optimal optimality cut.

$$\max_{\pi \in \Re^{m_2}} \{\pi^T(d - B\hat{y}) : \pi^T D \leq c, \quad \pi^T(d - B\bar{y}) = Q(\bar{y})\}, \tag{17}$$

where $Q(\bar{y})$ indicates the optimal cost of the regular subproblem for the current MP solution $\bar{y}$. Although this approach has proven effective in practice (e.g., Mercier et al., 2005), it must solve the secondary problem (17), which may be numerically instable and time-consuming. Additionally, it may be difficult to find a core point. This difficulty can be overcome by using approximate core points (Santoso et al., 2005), arbitrarily fixing components of the core-point vector (Mercier et al., 2005), or finding alternative points for a given problem structure (Papadakos, 2008), although these methods do not guarantee the generation of Pareto-optimal cuts.

Papadakos (2008) proposed an algorithmic modification to circumvent the computational difficulties of Magnanti and Wong (1981)'s approach. The author showed that if a new core point is utilized at each iteration, Pareto-optimal cuts can be obtained from an independent formulation of the Magnanti–Wong cut generation procedure. This formulation removes the equality constraint in (17) that implies the dependency on the solution of the regular subproblem (7). The author showed that any convex combination of a valid initial core point and the MP solution gives an alternative Magnanti–Wong core point. de Sá et al. (2013) then showed that when the MP solution is rendered infeasible by the primal subproblem, a clever choice of the convex-combination weights can yield significant speed-ups. They chose the weights in such a way that the convex combination of the current MP solution with the previous Magnanti–Wong point results in a feasible subproblem.

Fortz and Poss (2009) and Naoum-Sawaya and Elhedhli (2013) generated Pareto-optimal cuts from the points obtained by an analytic-center cutting plane method. In other words, the extracted analytic centers are used as core points. Note that the dual subproblem is equivalent to that of Papadakos (2008), although this is not directly mentioned. The procedure has great potential for accelerating the classical BD algorithm in the context of capacitated facility location and multi-commodity capacitated fixed charge network design problems. However, its effectiveness depends on the quality of the core points and the efficiency of the re-optimization techniques. Moreover, similarly to the methodology of Magnanti and Wong (1981) and Papadakos (2008), they produce classical feasibility cuts based on the random selection of an extreme ray to cut-off the infeasibility.

Gelareh et al. (2015) generated analytic center points at every integer node of their B&BC algorithm to deal with degeneracy. They used analytic centers and their convex combination with the current MP solution to generate multiple cuts. The authors presented a box method to deal with degeneracy. A constraint is included in the auxiliary subproblem to bound the dual objective function, while the dual solution must be at least κ units distant from the actual optimal dual. The cuts produced are inexact, but the method performed well.

Sherali and Lunday (2013) developed the maximal nondominated cut generation scheme by formulating the cut selection as a multi-objective optimization problem. They showed that a small perturbation in the right-hand side of the primal subproblem is enough to give a maximal nondominated optimality cut, avoiding the need to solve the secondary subproblem as in Magnanti and Wong (1981) and Papadakos (2008). Given a goal-programming weight $\mu > 0$, the dual subproblem is

$$\max_{\pi \in \Re^{m_2}} \{\pi^T(d - B\bar{y}) + \mu\pi^T(d - B\hat{y}) : \pi^T D \leq c\}. \tag{18}$$

Oliveira et al. (2014) considered the definition of the weight μ. The authors observed that the solutions obtained in the early iterations gave poor descriptions of the project cost, which is what the cuts attempt to approximate. They iteratively adjusted μ to favor solutions that focus on improving the original objective value $\pi^T(d - B\bar{y})$ rather than (18). To ensure convergence, the sequence $\{\mu^{(k)}\}_{k=1,\ldots,\infty}$ must satisfy $\sum_{k=1,\ldots,\infty} \mu^{(k)} \to \infty$ and $\mu^{(k)} \to 0$ as $k \to \infty$. The authors obtained results that compared favorably with those of Sherali and Lunday (2013) and Magnanti and Wong (1981).

Better feasibility cuts have also been investigated. Codato and Fischetti (2006) considered a binary problem where the BD method generates feasibility cuts exclusively. The authors observed that these cuts are weak because of the big-M constraints, and showed that stronger cuts, referred to as *combinatorial Benders cuts*, can be obtained by searching for minimal infeasible subsystems for the MP solutions. Experiments on two classes of mixed-integer problems indicated significant improvements in the bounds for the LP relaxation of the MP. Note that combinatorial cuts are not in general stronger than the classical feasibility cuts. Yang and Lee (2012) observed that the slow convergence of the BD algorithm is due to the selection of weak feasibility cuts. They extended the

dominance rule of Magnanti and Wong (1981) in order to extract tighter feasibility cuts. However, this involves solving an auxiliary bilinear problem, which can be computationally expensive.

Fischetti et al. (2010) used an idea from Fukuda, Liebling, and Margot (1997): finding the most-violated optimality cut is equivalent to finding an optimal vertex of a polyhedron with unbounded rays. Fischetti et al. (2010) thus reformulated the subproblem as a feasibility problem where the optimality and feasibility cuts are derived by searching for minimal infeasible subsystems:

$$\max_{\pi\in\Re^{m_2},\pi_0\in\Re^1}\{\pi^T(d-B\bar{y})-\pi_0\bar{\eta}:\pi^TD\le\pi_0c,\quad w^T\pi+w_0\pi_0=1\}, \tag{19}$$

where $\bar{\eta}$ is the current value of η, and w is a vector of normalization coefficients. The generated cut takes the form $\bar{\pi}^T(d-By)\le\bar{\pi}_0\eta$. This approach simultaneously generates optimality and feasibility cuts without solving an auxiliary subproblem. It compared favorably with the classical cut selection scheme.

Some algorithms iteratively generate multiple cuts to obtain specific desirable characteristics. Saharidis, Minoux, and Ierapetritou (2010) considered low-density cuts, i.e., cuts that include only a few MP variables. The ability of such cuts to strengthen the relaxed MP tends to be limited. To improve these cuts, the authors developed a *covering cut bundle* cut-generation procedure. At each iteration, it produces a set of low-density BD cuts that cover α% of the MP variables. The authors observed that adding several low-density cuts is better than adding a single high-density cut corresponding to the sum of the low-density cuts because it ensures a level of diversification in the cuts. Saharidis and Ierapetritou (2013) observed that it can be computationally less expensive to cover all the MP variables. Their strategy, referred to as *maximum density cut generation*, generates a cut that involves all the MP decision variables that are not covered in the BD cut. The authors observed that this significantly decreases the number of iterations and the time requirement for two different scheduling problems. Saharidis and Ierapetritou (2010) considered a case where obtaining optimality cuts using the classical BD method is hard. The bounds progress slowly because numerous feasibility cuts are generated before an initial feasible solution yielding an optimality cut is found. Whenever a feasibility cut is generated, they apply a *maximal feasible subsystem* to produce an optimality cut. This cut is produced by relaxing a minimum number of constraints in order to obtain a feasible subproblem. The authors observed significant improvements in convergence. The weakness of this strategy is that the reduction in the number of iterations may not always compensate for the additional time required to solve the auxiliary MILP subproblem.

In summary, the classical cut-generation scheme can be inefficient, particularly when the subproblems are degenerate or infeasible. Thus, 31.25% and 17.71% of the cited articles in this review paper have used one of the mentioned strategies to generate optimality and feasibility cuts, respectively. Almost every application of the BD that yields degenerate subproblems uses one of the strategies we have discussed to generate Pareto-optimal cuts. However, generating Pareto-optimal cuts may not yield a net computational advantage, since the reduction in the number of iterations might not compensate for the increase in the number of subproblems at each iteration (Mercier & Soumis, 2007). Strategies such as maximal nondominated cut generation may be more efficient since they eliminate the need to solve the auxiliary subproblem. Regarding the feasibility cuts, the strategies based on nondominated cuts have focused on optimality cuts; feasibility cuts are found based on a random selection of the extreme rays. Only the strategy that generates combinatorial cuts for subproblems with big-M constraints has proven its worth in practice. Moreover, feasibility and optimality cuts are usually treated separately. To the best of our knowledge, only Fischetti et al. (2010) have developed a unified framework for both types of cuts. Clearly, further research is necessary.

8. Benders-type heuristics

Because of time and memory limitations, the execution of the BD method might be stopped before its convergence is established. Moreover, in many practical applications, decision-makers do not need a provably optimal solution, a good feasible solution being deemed sufficient. Such a solution is often obtained somewhat early in the solution process.

From a heuristic point of view, the BD method is an attractive methodology because it can take advantage of special structures and provides a rich framework for the design of efficient search mechanisms (Côté & Laughton, 1984; Raidl, 2015). The method also overcomes many drawbacks of heuristics such as the inability to verify the solution quality and the difficulty to reduce the search space by using dual information (Boschetti & Maniezzo, 2009; Easwaran & Üster, 2009). These factors have promoted the development of algorithms that we refer to as *Benders-type heuristics*. We now discuss some of these.

Applying Lagrangian relaxation to the Benders cuts has been a popular approach, especially when the MP without cuts has a special structure (Minoux, 1984; Paula & Maculan, 1988). Côté and Laughton (1984) applied Lagrangian relaxation to the feasibility and optimality cuts so that the remaining constraint sets have a special structure and specialized algorithms can be applied. Aardal and Larsson (1990) proposed a heuristic for a multi-item dynamic production planning problem. They created structured subproblems and MPs, priced out the BD cuts using Lagrangian multipliers in order to maintain the problem structure, and used a subgradient procedure to update the Lagrangian multipliers. The algorithm attained an average deviation of 2.34% from the optimum. Holmberg (1994) studied different approximations for the Benders MP, concluding that its Lagrangian dual cannot yield better bounds than the Lagrangian dual of the original problem even if all the feasibility and optimality cuts are present in the MP.

An alternative to the above approaches is to first apply Lagrangian relaxation and then use the BD method to optimize the Lagrangian dual subproblem. Pinheiro and Oliveira (2013) tackled problems with complicating constraints that are challenging for the BD method. They first applied Lagrangian relaxation to these constraints and then used the BD method to optimize the problem at each iteration of the dual Lagrangian algorithm. Wang, McCalley, Zheng, and Litvinov (2016) applied a similar methodology to solve a corrective risk-based security-constrained optimal power flow problem. The results of both studies point to the ability of the approach to handle large-scale, complex problems. Further research in this area would thus be worthwhile.

One challenge in large-scale problems is the need to solve a sequence of difficult integer MPs. Many researchers have explored the use of meta-heuristics for the MPs. Poojari and Beasley (2009) used a genetic algorithm combined with a feasibility pump. This enabled the authors to add multiple cuts per iteration, which yielded larger increases in the lower bounds. Although the MP was never solved to optimality, good results were obtained. Jiang, Tang, and Xue (2009) used a similar hybridization, based on tabu search, for multi-product distribution network design. A genetic-BD hybrid algorithm for vehicle routing and scheduling (Lai, Sohn, Tseng, & Bricker, 2012) and the capacitance plant location problem (Lai, Sohn, Tseng, & Chiang, 2010) has greatly reduced the computational time in comparison with the classical BD method. Boschetti and Maniezzo (2009) solved both the MP and the subproblem heuristically; their algorithm was competitive with state-of-the-art meta-heuristics. Note that these strategies do not

provide a valid lower bound, and thus, to assess the solution quality, the MP must be solved to optimality or approximated from below.

Proximity Benders is a decomposition heuristic proposed by Boland, Fischetti, Monaci, and Savelsbergh (2016). The authors observed that the BD method rarely improves the incumbent solution, and finding good solutions may require considerable computing time. The authors used a *proximity heuristic* to more frequently improve the upper bound obtained from the sequence of MP solutions. Computational experiments demonstrated the potential of the method. Kudela and Popela (2015) proposed a genetic algorithm where the BD method is used to take advantage of the block structure. The authors reported favorable results in comparison with the genetic algorithm without the decomposition. Behnamian (2014) proposed a Benders-based variable neighborhood search algorithm for a multi-objective scheduling problem. The goal was to accelerate the assessment of the estimated improvement of each neighborhood. The new heuristic outperformed a variable neighborhood search, a tabu search, and a hybrid of these two methods, particularly on larger instances.

Another approach solves the LP relaxation of the MP and uses round-off heuristics to find an integer solution. Pacqueau et al. (2012) use the BD method to solve the linear relaxation and then fix some of the variables to their upper/lower bounds. Their algorithm iterates until an integer solution is obtained. They solved problems with up to 10 million integer variables in less than 27 minutes with an average accuracy of 0.2%, while CPLEX could handle only instances with fewer than 500,000 integer variables. This highlights the potential of efficient Benders-type heuristics for problems with computationally intractable MPs, particularly those with tight linear relaxations.

In summary, the BD method enables heuristics to take advantage of special structures and use dual information. Many Benders-type heuristics either solve the MP heuristically or use approximate MP formulations that do not provide global convergence. Benders-type heuristics can handle a wider range of structures than the BD method. However, these algorithms cannot find a provably optimal solution. We discuss extensions of the BD algorithm that can exactly solve a wider range of problems in Section 9.

9. Extensions of the classical Benders decomposition method

The classical BD algorithm was proposed for certain classes of MILPs for which the integer variables were considered to be complicating, and standard duality theory could be applied to the subproblem to develop cuts. Extensions of the method have allowed it to address a broader range of optimization problems, including integer subproblems (e.g., Carøe & Tind, 1998), nonlinear functions (e.g., Cai et al., 2001; Geoffrion, 1972), logical expressions (e.g., Eremin & Wallace, 2001), multi-stage programming (e.g., Lorenz & Wolf, 2015), and stochastic optimization (e.g., Van Slyke & Wets, 1969). When applied to stochastic problems the BD method is commonly referred to as *L-shaped decomposition*. It enables such problems to be decomposed by the realization of the uncertain parameters. Many algorithms for these important and challenging problems rely heavily on its premises (Ruszczyński, 2003). We have already discussed the literature on this variant, which is equivalent to the classical BD method. In this section, we discuss the extensions to problems with discrete subproblems, logical expressions, and nonlinear terms as well as multi-stage programming.

9.1. Discrete subproblems

When some of the projected variables are required to be integer, standard duality theory cannot be applied to derive the classical Benders cuts. A different theoretical framework or modifications to the generation scheme are needed to handle integer subproblems effectively.

When the complicating variables are required to take 0–1 values, one can use *lower-bounding functions* (LBF) instead of the regular optimality cuts (Laporte & Louveaux, 1993). These constraints enforce a change to the current solution or the acceptance of its associated cost. They usually take the form

$$\eta \geq (Q(\bar{y}) - L)\left(\sum_{a\in A_1} y_a - \sum_{a\in A_0} y_a - |A_1|\right) + Q(\bar{y}), \tag{20}$$

where $Q(\bar{y})$ is the cost of the subproblem for the given solution $\bar{y}$, A_1 and A_0 are respectively the variables with values of 1 and 0 in $\bar{y}$, and L is a lower bound on $Q(y)$ over y. The BD method with LBF cuts (20) is also applicable to problems where the subproblem can be evaluated with a closed-form analytical formula. Given the enumerative nature of (20), it is usually complemented with other VIs to improve the lower bound. A common strategy is based on solving the linear relaxation of the subproblem to generate regular optimality cuts (Cordeau et al., 2001b; Mercier & Soumis, 2007; Papadakos, 2008). Moreover, the optimality cut (20) depends on the exact solution of $Q(\bar{y})$ and gives no useful information on the other y solutions. These issues are partly addressed by Angulo et al. (2014).

A similar variant of the classical BD method, often referred to as *combinatorial Benders decomposition*, likewise does not use the dual information to generate cuts. This variant can handle problems where the MP is a 0–1 integer program and the subproblem is a feasibility problem (i.e., a problem with no objective function). It excludes the current MP solution from further consideration via *combinatorial cuts*, which usually take the form

$$\sum_{a\in A:\bar{y}_a=1} (1 - y_a) + \sum_{a\in A:\bar{y}_a=0} y_a \geq 1, \tag{21}$$

Constraints of the form (21) are frequently used in the BD method as feasibility cuts. They are often strengthened according to the structure of the application, e.g., nonlinear power design in green wireless local networks (Gendron et al., 2014), lock scheduling (Verstichel, Kinable, De Causmaecker, & Berghe, 2015), strip packing (Côté, Dell'Amico, & Iori, 2014), and radiation therapy (Taşkın & Cevik, 2013).

Carøe and Tind (1998) used general duality theory to reformulate the subproblems, using VIs based on dual price functions to produce the BD cuts. They showed how such functions can be obtained when the subproblem is solved via standard techniques such as branch-and-bound or cutting planes. Sherali and Fraticelli (2002) considered applications with 0–1 mixed integer subproblems. They showed that the classical BD method is applicable if a convex hull representation of the constrained region is available. They employed the reformulation–linearization technique and lift-and-project cuts as a sequential convexification procedure for the subproblems. The cuts generated by these two methods were functions of the MP variables and were globally valid, which could lead to a finite convergent BD algorithm. Sen and Higle (2005) applied disjunctive programming to produce a convex characterization for the discrete subproblems. They showed that VIs generated for a given MP solution and a particular subproblem can be used to obtain VIs for any other solution or subproblem. This result can be used to define the cut-generation procedure in an overall BD approach. This approach was extended by Sen and Sherali (2006), who showed how branch-and-cut algorithms can be used for the subproblems.

We conclude with a remark on solving the integer subproblems. Heuristics have proven their worth in accelerating the BD method, particularly when the subproblem reduces to a feasibility-checking

program or generates more feasibility cuts than optimality cuts (Osman & Baki, 2014). Heuristics can rapidly detect infeasibility and avoid the exact solution of difficult subproblems (e.g., Luong, 2015) or find approximate optimality cuts quickly (e.g., Raidl, Baumhauer, & Hu, 2014). In the latter case additional refinement is required, since the cuts may eliminate optimal solutions. On the other hand, CP is widely used to handle feasibility subproblems with special constraints because of its ability to handle those constraints and because it identifies infeasibility more quickly than traditional MIP-based approaches can (see e.g., Jain & Grossmann, 2001; Maravelias & Grossmann, 2004).

9.2. Logic-based Benders decomposition

There is a growing interest in optimization models that include logic relations. These models can usually be transformed into regular optimization models, but the extra variables and big-M constraints often yield a weak formulation. Furthermore, one cannot always obtain a continuous linear subproblem; it may contain some integer variables and nonlinear functions. In these cases, standard linear duality cannot be used to develop classical BD cuts.

Hooker and Ottosson (2003) and Hooker (2011) introduced an extension known as *logic-based Benders decomposition* (*LBBD*). The LBBD method is similar to the classical BD method. It decomposes a given problem into an MP and one or many subproblems, and it uses constraint-generation techniques to gradually reduce the solution space of the relaxed MP. However, each subproblem is an "inference dual" problem that finds the tightest bound on the MP's cost function implied by its current solution. This bound is then used to generate cuts that are passed back to the MP. If the MP solution satisfies all the bounds produced by the subproblems, convergence has been achieved; otherwise, the process continues.

A major advantage of the LBBD method is that the subproblem needs not take a specific form: it can be an MILP (Roshanaei, Luong, Aleman, & Urbach, 2017), a CP (Hooker, 2005), an NLP (Wheatley, Gzara, & Jewkes, 2015), or a feasibility-checking problem (Harjunkoski & Grossmann, 2002). However, the LBBD method does not have a standard template for the production of valid cuts. Instead, they must be tailored to the problem at hand, typically based on knowledge of its structure. For some problems simple cuts exist (Hooker, 2007), but one must balance their effectiveness with the ease of extraction (Zarandi, 2010). It has been shown that The LBBD method can outperform state-of-the-art MIP and CP solvers in various applications, often by several orders of magnitude (e.g., Jain & Grossmann, 2001). It has been applied to a range of problems, including planning and scheduling (e.g., Hooker, 2007; Benoist et al., 2002), facility location/fleet management (Zarandi, 2010), radiation therapy (Luong, 2015), transportation network design (Peterson & Trick, 2009), and the minimal dispatching problem of automated guided vehicles (Corréa et al., 2007). It is worth mentioning that the B&BC framework discussed in Section 5.1.2 is often referred to as the branch-and-check method in context of LBBD (Beck, 2010; Thorsteinsson, 2001).

9.3. Generalized Benders decomposition

Many optimization problems involve nonlinear functions and constraints. If the problem is easily linearized or the nonlinearity occurs only in the domain of the complicating variables, it can be solved via the classical BD method (Cai et al., 2001; Fontaine & Minner, 2014; Osman & Baki, 2014). Specifically, whenever the subproblem takes the form of a continuous linear formulation. Otherwise, an extended BD method is necessary.

Geoffrion (1972) proposed *Generalized Benders Decomposition* (*GBD*). It can solve nonlinear problems for which the subproblem is a convex program, because dual multipliers satisfying strong duality conditions can be calculated for such problems (Bazaraa, Sherali, & Shetty, 2013). It is also particularly appealing for nonconvex nonlinear problems that can be convexified after fixing a subset of variables (Costa, 2005).

Sahinidis and Grossmann (1991) showed that the GBD method may not lead to a global or even local optimum for MINLP problems. Specifically, when the objective function and some of the constraints are nonconvex or when nonlinear equations are present, the subproblem may not have a unique local optimum and the MP may cut off the global optimum. Rigorous global optimization approaches can be used if the continuous terms have a special structure (e.g., bilinear, linear fractional, concave separable). The basic idea is to use convex envelopes (or underestimators) to formulate lower-bounding convex MINLPs. These are then integrated with global optimization techniques for continuous variables, which usually take the form of spatial branch-and-bound methods (see Grossmann, 2002, for further details). Similarly, Grothey, Leyffer, and McKinnon (1999) observed that a simplistic application of the GBD algorithm to a convex nonlinear problem may converge to a nonstationary point. They showed that the convergence failure results from the way in which the infeasible subproblems are handled, and they proposed a feasibility restoration procedure.

9.4. Nested Benders decomposition

The *Nested Benders Decomposition* (*NBD*) method is based on the idea of applying the BD method to a problem more than once. It is particularly appropriate for multi-stage (stochastic) problems (Birge, 1985) in which each pair of adjacent stages can be considered "separately". The NBD views the scenario tree as a set of nested two-stage problems and applies the BD method recursively. Every problem associated with an inner node in the tree is both MP to its children and a subproblem of its parent. It is necessary to choose the sequencing protocols: after solving the problems at a given stage, one can either push primal information down toward the leaf nodes or pass dual information up toward the root node. This issue and some acceleration strategies are addressed by Wolf (2014).

The NBD method can also be applied to deterministic single-stage problems, particularly when one wishes to simplify the MP by reducing the number of integer variables. For example, Naoum-Sawaya and Elhedhli (2010) applied the BD method to obtain a binary MP and a mixed integer subproblem. They then applied the BD method to the subproblem to obtain an integer MP and a linear subproblem.

10. Conclusions and future research

We have presented a state-of-the-art survey of the BD method. We have discussed the classical algorithm, the impact of the problem formulation on its convergence, and the relationship to other decomposition methods. We have developed a taxonomy to classify the literature on acceleration strategies, based on the main components of the algorithm, which provided rich guidelines to analyze various enhancements, identifying shortcomings, trends and potential research directions. We have also discussed the use of the BD to develop efficient (meta-)heuristics, described the limitations of the classical algorithm, and presented extensions enabling its application to a broader range of problems.

The BD method was originally proposed for MILPs with continuous subproblems, and it has since been extended to handle a wider range of problems such as nonlinear, integer, multi-stage, and constraint programming problems. Four main classes of acceleration strategies have been developed to enhance the classical al-

gorithm: modifying the decomposition, solving the MP and subproblem more effectively, generating stronger cuts, and extracting better solutions. The effectiveness of these strategies is problem-dependent, and a combination of them usually gives the best results. The BD method has also been used to develop efficient heuristics for complex problems, particularly those that numerically or structurally are out of reach of the method. This is not however to say that research into the BD is over. There are still many challenges and open questions.

Generally speaking, the BD method has been suitable for problems in which temporarily fixing the complicating variables makes the remaining problem significantly easier to handle by, e.g., becoming suitable for specialized algorithms or offering the opportunity to transform a nonconvex problem into a convex one. The BD method appeared particularly appropriate for problems with a "few" complicating (normally 0–1) variables and so many continuous variables that solving the problem as a whole is inefficient. There are many examples of such problems in stochastic programming. The range of problem settings addressed is also expanding, however. Moreover, many problems suffer from having weak linear relaxations and numerical instability as a result of big–M constraints and the binary variables used to turn them on and off. The BD method can handle such problems by moving the big–M constraints to the subproblems and using specialized cuts to represent them. The BD algorithm has also been applied to bilevel optimization problems that cannot be transformed via the Karush–Kuhn–Tucker optimality conditions into single-level problems (Saharidis & Ierapetritou, 2009). Moreover, there are interesting optimization problems for which some of the constraints are not known in advance and must be generated iteratively. In other cases the subproblem does not have an amenable formulation but can be obtained via a closed-form analytical formula. We are aware of only one survey article focusing on the applications of the BD method, and it restricted its scope to fixed-charged network design problems (Costa, 2005). There is certainly a need for a comprehensive synthesis of the various applications of the BD algorithm.

The acceleration strategies are all problem-dependent, so they are all part of the BD toolbox and their interconnections are important. A better understanding of these interconnections could have a considerable impact on the convergence rate. There is also a need for comprehensive research into the acceleration methodologies to better understand their limitations and implications. This is certainty true for more recent strategies, e.g., decomposition and cut generation schemes.

Strategies that tighten the MP usually add inequalities only once, before the initial iterations. Given the encouraging results obtained, it would be interesting to explore the use of more advanced cutting-plane methods to further tighten the MP at each iteration. The proper sequencing of the VIs and the classical BD cuts would be of great importance.

Tightening the subproblem is another effective acceleration strategy, since stronger cuts will be generated. We are aware of only one relevant study, Bodur, Dash, and Luedtke (2014) that iteratively generate Gomory mixed-integer cuts to tighten the subproblem. We encourage further research in this area. We note that one can solve the linear relaxation of the original problem with a cutting-plane method, adding VIs that involve the continuous variables. After the decomposition, these VIs will be moved to the subproblem, and this can yield a pronounced improvement in the quality of the Benders cuts. Moreover, one can use the partial decomposition to iteratively generate VIs for the subproblems, provided the subproblem retained in the MP gives VIs for the projected subproblems as well.

In terms of generating solutions for the set of complicating variables, we are not aware of any study showing how to obtain better cuts via a careful selection from the multiple optimal solutions of the MP or showing how to modify the MP to generate solutions with specific characteristics, e.g., geometrically centered solutions. These two ideas have been successfully applied in the context of Dantzig–Wolfe decomposition (e.g., Holloway, 1973; Nemhauser & Widhelm, 1971).

Sometimes the subproblem further divides into several independent subproblems that can be optimized concurrently. The literature on parallel algorithms for combinatorial optimization problems indicates that the parallel variants of the BD algorithm are still in their infancy. The current model is the master-slave paradigm, which is not the most efficient strategy (Crainic, 2015). Executing heuristics in the inner loop of the BD method or hybridizing the algorithm with other methods can yield tremendous improvements in the convergence rate. These approaches have been developed in a sequential framework, although they can be run almost independently of the main BD algorithm. Therefore, the development of new parallel algorithms, particularly in cooperative frameworks, would be worthwhile.

The BD decomposition often yields subproblems with identical dual polyhedra. In this situation, the solution of any subproblem gives valid cuts for the other subproblems. To the best of our knowledge, this information has not yet been used in an acceleration strategy. It should be interesting to develop a strategy in which only some of the subproblems are solved at each iteration and their solutions are used to generate cuts for other subproblems. Clearly, the main challenges are the selection of the set of representative subproblems and the demonstration of the convergence of the algorithm.

There has been limited research into stabilization techniques for the BD method in the context of combinatorial optimization. All the approaches surveyed were based on binary complicating variables, and state-of-the-art strategies for combinatorial problems, attempt to stabilize the MP only at the beginning of the algorithm. There is thus a need for more effective techniques in more general settings. One can take advantage of stabilization strategies developed for continuous problems by first solving the linear relaxation of the MP. Although this idea has not yet been explored, it may prove effective.

There are various BD extensions for which researchers have explored ways to enhance them. We plan to survey these enhancements in a future article.

Acknowledgments

Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Discovery Grant program and by the Fonds de recherche du Québec through its Team Grant program. We also gratefully acknowledge the support of Fonds de recherche du Québec through their strategic infrastructure grants.

References

Aardal, K., & Larsson, T. (1990). A Benders decomposition based heuristic for the hierarchical production planning problem. *European Journal of Operational Research, 45*(1), 4–14.

Adulyasak, Y., Cordeau, J.-F., & Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research, 63*(4), 851–867.

Angulo, G., Ahmed, S., & Dey, S. S. (2014). Improving the integer L-shaped method. Optimization Online, Available at http://www.optimization-online.org/DB_FILE/2014/04/4332.pdf.

Ariyawansa, K. A., & Hudson, D. D. (1991). Performance of a benchmark parallel implementation of the Van Slyke and Wets algorithm for two-stage stochastic programs on the sequent/balance. *Concurrency: Practice and Experience, 3*(2), 109–128.

Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2013). *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons.

Beck, J. C. (2010). Checking-up on branch-and-check. In D. Cohen (Ed.), *Principles and practice of constraint programming – CP 2010: 16th international conference: vol. 6308* (pp. 84–98). Berlin, Heidelberg: Springer.

Behnamian, J. (2014). Decomposition based hybrid VNS-TS algorithm for distributed parallel factories scheduling with virtual corporation. *Computers & Operations Research, 52*, 181–191.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik, 4*(1), 238–252.

Benoist, T., Gaudin, E., & Rottembourg, B. (2002). Constraint programming contribution to Benders decomposition: a case study. In *Proceedings of the 8th international conference on principles and practice of constraint programming. CP '02.* (pp. 603–617). London, UK: Springer-Verlag.

Birge, J. R. (1985). Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research, 33*(5), 989–1007.

Birge, J. R., & Louveaux, F. (1997). *Introduction to stochastic programming*. Springer, New York.

Bloom, J. A. (1983). Solving an electricity generating capacity expansion planning problem by generalized Benders' decomposition. *Operations Research, 31*(1), 84–100.

Bodur, M., Dash, S., & Luedtke, J. (2014). Strengthened Benders cuts for stochastic integer programs with continuous recourse. Optimization Online, Available at http://www.optimization-online.org/DB_FILE/2014/03/4263.pdf.

Boland, N., Fischetti, M., Monaci, M., & Savelsbergh, M. (2016). Proximity Benders: a decomposition heuristic for stochastic programs. *Journal of Heuristics, 22*(2), 181–198.

Boschetti, M., & Maniezzo, V. (2009). Benders decomposition, lagrangean relaxation and metaheuristic design. *Journal of Heuristics, 15*(3), 283–312.

Botton, Q., Fortz, B., Gouveia, L., & Poss, M. (2013). Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal on Computing, 25*(1), 13–26.

Brandes, K. T. (2011). *Implementierung und analyse verschiedener strategien zur aggregation und disaggregation von multi-cuts im benders dekompositionsverfahren*. Universität Paderborn, North Rhine-Westphalia, Germany Master's thesis.

Cai, X., McKinney, D. C., Lasdon, L. S., & Watkins, D. W. (2001). Solving large nonconvex water resources management models using generalized Benders decomposition. *Operations Research, 49*(2), 235–245.

Canto, S. P. (2008). Application of Benders decomposition to power plant preventive maintenance scheduling. *European Journal of Operational Research, 184*(2), 759–777.

Carøe, C. C., & Tind, J. (1998). L-Shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming, 83*(1–3), 451–464.

Chermakani, D. P. (2015). Optimal aggregation of blocks into subproblems in linear programs with block-diagonal-structure. ArXiv, Available at https://arxiv.org/ftp/arxiv/papers/1507/1507.05753.pdf.

Codato, G., & Fischetti, M. (2006). Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research, 54*(4), 756–766.

Contreras, I., Cordeau, J.-F., & Laporte, G. (2011). Benders decomposition for large-scale uncapacitated hub location. *Operations Research, 59*(6), 1477–1490.

Cordeau, J.-F., Pasin, F., & Solomon, M. M. (2006). An integrated model for logistics network design. *Annals of Operations Research, 144*(1), 59–82.

Cordeau, J.-F., Soumis, F., & Desrosiers, J. (2001a). Simultaneous assignment of locomotives and cars to passenger trains. *Operations Research, 49*(4), 531–548.

Cordeau, J.-F., Stojković, G., Soumis, F., & Desrosiers, J. (2001b). Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science, 35*(4), 375–388.

Corréa, A. I., Langevin, A., & Rousseau, L.-M. (2007). Scheduling and routing of automated guided vehicles: A hybrid approach. *Computers & Operations Research, 34*(6), 1688–1707.

Costa, A. M. (2005). A survey on Benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research, 32*(6), 1429–1450.

Costa, A. M., Cordeau, J.-F., & Gendron, B. (2009). Benders, metric and cutset inequalities for multicommodity capacitated network design. *Computational Optimization and Applications, 42*(3), 371–392.

Costa, A. M., Cordeau, J.-F., Gendron, B., & Laporte, G. (2012). Accelerating Benders decomposition with heuristic master problem solutions. *Pesquisa Operacional, 32*(1), 03–20.

Côté, G., & Laughton, M. A. (1984). Large-scale mixed integer programming: Benders-type heuristics. *European Journal of Operational Research, 16*(3), 327–333.

Côté, J.-F., Dell'Amico, M., & Iori, M. (2014). Combinatorial Benders' cuts for the strip packing problem. *Operations Research, 62*(3), 643–661.

Crainic, T. G. (2015). Parallel meta-heuristic search. In *Publication CIRRELT-2015-42*. Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Université de Montréal, Montréal, QC, Canada.

Crainic, T. G., Hewitt, M., & Rei, W. (2014). Partial decomposition strategies for two-stage stochastic integer programs. In *Publication CIRRELT-2014-13, centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport*. Université de montréal, Montréal, QC, Canada.

Crainic, T. G., Hewitt, M., & Rei, W. (2016). Partial Benders decomposition strategies for two-stage stochastic integer programs. In *Publication CIRRELT-2016-37, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport*. Université de Montréal, Montréal, QC, Canada.

Dantzig, G. B., Ho, J. K., & Infanger, G. (1991). *Solving stochastic linear programs on a hypercube multicomputer*. DTIC Document Tech. rep. ada240443.

de Camargo, R. S., de Miranda, G., Jr., & Ferreira, R. P. (2011). A hybrid outer-approximation/Benders decomposition algorithm for the single allocation hub location problem under congestion. *Operations Research Letters, 39*(5), 329–337.

de Sá, E. M., de Camargo, R. S., & de Miranda, G. (2013). An improved Benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research, 226*(2), 185–202.

Easwaran, G., & Üster, H. (2009). Tabu search and Benders decomposition approaches for a capacitated closed-loop supply chain network design problem. *Transportation Science, 43*(3), 301–320.

Emami, S., Moslehi, G., & Sabbagh, M. (2016). A Benders decomposition approach for order acceptance and scheduling problem: a robust optimization approach. *Computational and Applied Mathematics*, 1–45.

Eremin, A., & Wallace, M. (2001). Hybrid Benders decomposition algorithms in constraint logic programming. In *In proceding of the 7th international conference on principles and practice of constraint programming. CP '01. springer-verlag, paphos, cyprus* (pp. 1–15).

Errico, F., Crainic, T. G., Malucelli, F., & Nonato, M. (2016). A Benders decomposition approach for the symmetric TSP with generalized latency arising in the design of semiflexible transit systems. In *Transportation Science* (pp. 1–17).

Fábián, C. I. (2000). Bundle-type methods for inexact data. *Central European Journal of Operations Research, 8*(1), 35–55.

Fábián, C. I., & Szőke, Z. (2007). Solving two-stage stochastic programming problems with level decomposition. *Computational Management Science, 4*(4), 313–353.

Fischetti, M., Ljubic, I., & Sinnl, M. (2016). Redesigning Benders decomposition for large-scale facility location. *Management Science*. doi:10.1287/mnsc.2016.2461.

Fischetti, M., Salvagnin, D., & Zanette, A. (2010). A note on the selection of Benders' cuts. *Mathematical Programming, 124*(1–2), 175–182.

Fontaine, P., & Minner, S. (2014). Benders decomposition for discrete–continuous linear bilevel problems with application to traffic network design. *Transportation Research Part B: Methodological, 70*, 163–172.

Fortz, B., & Poss, M. (2009). An improved Benders decomposition applied to a multi-layer network design problem. *Operations Research Letters, 37*(5), 359–364.

Fukuda, K., Liebling, T. M., & Margot, F. (1997). Analysis of backtrack algorithms for listing all vertices and all faces of a convex polyhedron. *Computational Geometry, 8*(1), 1–12.

Gabrel, V., Knippel, A., & Minoux, M. (1999). Exact solution of multicommodity network optimization problems with general step cost functions. *Operations Research Letters, 25*(1), 15–23.

Gelareh, S., Monemi, R. N., & Nickel, S. (2015). Multi-period hub location problems in transportation. *Transportation Research Part E: Logistics and Transportation Review, 75*, 67–94.

Gendron, B., Scutellà, M. G., Garroppo, R. G., Nencioni, G., & Tavanti, L. (2014). A branch-and-Benders-cut method for nonlinear power design in green wireless local area networks. In *Publication CIRRELT-2014-42, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport*. Université de montréal, Montréal, QC, Canada.

Geoffrion, A. M. (1970a). Elements of large-scale mathematical programming: Part I: Concepts. *Management Science, 16*(11), 652–675.

Geoffrion, A. M. (1970b). Elements of large scale mathematical programming: Part II: synthesis of algorithms and bibliography. *Management Science, 16*(11), 676–691.

Geoffrion, A. M. (1972). Generalized Benders decomposition. *Journal of Optimization Theory and Applications, 10*(4), 237–260.

Geoffrion, A. M., & Graves, G. W. (1974). Multicommodity distribution system design by Benders decomposition. *Management Science, 20*(5), 822–844.

Grossmann, I. E. (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering, 3*(3), 227–252.

Grothey, A., Leyffer, S., & McKinnon, K. (1999). A note on feasibility in Benders decomposition. In *Numerical analysis report NA/188, university of dundee, nethergate, dundee, scotland, UK*.

Harjunkoski, I., & Grossmann, I. E. (2001). A decomposition approach for the scheduling of a steel plant production. *Computers & Chemical Engineering, 25*(11), 1647–1660.

Harjunkoski, I., & Grossmann, I. E. (2002). Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. *Computers & Chemical Engineering, 26*(11), 1533–1552.

Holloway, C. A. (1973). A generalized approach to Dantzig-Wolfe decomposition for concave programs. *Operations Research, 21*(1), 210–220.

Holmberg, K. (1990). On the convergence of cross decomposition. *Mathematical Programming, 47*(1–3), 269–296.

Holmberg, K. (1994). On using approximations of the Benders master problem. *European Journal of Operational Research, 77*(1), 111–125.

Hooker, J. (2011). *Logic-based methods for optimization: combining optimization and constraint satisfaction*. John Wiley & Sons.

Hooker, J. N. (2005). A hybrid method for the planning and scheduling. *Constraints, 10*(4), 385–401.

Hooker, J. N. (2007). Planning and scheduling by logic-based Benders decomposition. *Operations Research, 55*(3), 588–602.

Hooker, J. N., & Ottosson, G. (2003). Logic-based Benders decomposition. *Mathematical Programming, 96*(1), 33–60.

Jain, V., & Grossmann, I. E. (2001). Algorithms for hybrid MILP/CP models for a class of optimization problems. *INFORMS Journal on computing, 13*(4), 258–276.

Jeihoonian, M., Zanjani, M. K., & Gendreau, M. (2016). Accelerating Benders decomposition for closed-loop supply chain network design: Case of used durable products with different quality levels. *European Journal of Operational Research, 251*(3), 830–845.

Jenabi, M., Ghomi, S. F., Torabi, S., & Hosseinian, S. (2015). Acceleration strategies of Benders decomposition for the security constraints power system expansion planning. *Annals of Operations Research, 235*(1), 337–369.

Jiang, W., Tang, L., & Xue, S. (2009). A hybrid algorithm of tabu search and Benders decomposition for multi-product production distribution network design. In *Proceedings of the IEEE international conference on automation and logistics. ICAL '09* (pp. 79–84). China: Shenyang.

Kewcharoenwong, P., & Üster, H. (2014). Benders decomposition algorithms for the fixed-charge relay network design in telecommunications. *Telecommunication Systems, 56*(4), 441–453.

Kouvelis, P., & Yu, G. (1997). Robust discrete optimization and its applications. In *Springer US, boston, MA, ch. robust uncapacitated network design and international sourcing problems* (pp. 290–332).

J. Kudela, P.Popela, Two-stage stochastic facility location problem: GA with Benders decomposition 2015, 53 58, Mendel 2015-January.

Lai, M.-C., Sohn, H.-S., Tseng, T.-L., & Bricker, D. L. (2012). A hybrid Benders/genetic algorithm for vehicle routing and scheduling problem. *International Journal of Industrial Engineering, 19*(1), 33–46.

Lai, M.-C., Sohn, H.-s., Tseng, T.-L. B., & Chiang, C. (2010). A hybrid algorithm for capacitated plant location problem. *Expert Systems with Applications, 37*(12), 8599–8605.

Laporte, G., & Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters, 13*(3), 133–142.

Laporte, G., Louveaux, F. V., & Mercure, H. (1994). A priori optimization of the probabilistic traveling salesman problem. *Operations Research, 42*(3), 543–549.

Lemaréchal, C., Nemirovskii, A., & Nesterov, Y. (1995). New variants of bundle methods. *Mathematical Programming, 69*(1–3), 111–147.

Li, X. (2013). Parallel nonconvex generalized Benders decomposition for natural gas production network planning under uncertainty. *Computers & Chemical Engineering, 55*, 97–108.

Lim, C. (2010). Relationship among Benders, Dantzig-Wolfe, and Lagrangian optimization. In J.J. Cochran, L.A. Cox, P. Keskinocak, J.P. Kharoufeh, & J.C. Smith (Eds.), Wiley encyclopedia of operations research and management science. John Wiley & Sons

Lin, H., & Üster, H. (2014). Exact and heuristic algorithms for data-gathering cluster-based wireless sensor network design problem. *IEEE/ACM Transactions on Networking, 22*(3), 903–916.

Linderoth, J., & Wright, S. (2003). Decomposition algorithms for stochastic programming on a computational grid. *Computational Optimization and Applications, 24*(2-3), 207–250.

Lorenz, U., & Wolf, J. (2015). Solving multistage quantified linear optimization problems with the alpha–beta nested Benders decomposition. *EURO Journal on Computational Optimization, 3*(4), 349–370.

Luong, C. (2015). *An examination of Benders decomposition approaches in large-scale healthcare optimization problems*. University of Toronto Master's thesis.

Magnanti, T. L., Mireault, P., & Wong, R. T. (1986). Tailoring Benders decomposition for uncapacitated network design. *Mathematical Programming Study, 26*, 112–154.

Magnanti, T. L., & Simpson, R. W. (1978). *Transportation network analysis and decomposition methods*. U.S. Department of Transportation Report no. dot-tsc-r-spd-78-6.

Magnanti, T. L., & Wong, R. T. (1981). Accelerating Benders decomposition: algorithmic enhancement and model selection criteria. *Operations Research, 29*(3), 464–484.

Maravelias, C. T., & Grossmann, I. E. (2004). Using MILP and CP for the scheduling of batch chemical processes. In J.-C. Régin, & M. Rueher (Eds.), *First international conference on integration of AI and OR techniques in constraint programming for combinatorial optimization problems. CPAIOR '04. springer, nice, france* (pp. 1–20).

McDaniel, D., & Devine, M. (1977). A modified Benders' partitioning algorithm for mixed integer programming. *Management Science, 24*(3), 312–319.

Mercier, A., Cordeau, J.-F., & Soumis, F. (2005). A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research, 32*(6), 1451–1476.

Mercier, A., & Soumis, F. (2007). An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research, 34*(8), 2251–2265.

Minoux, M. (1984). Subgradient optimization and Benders decomposition for large scale programming. In R. W. Cottle, M. L. Kelmanson, & B. Korte (Eds.), *Mathematical Programming* (pp. 271–288). Elsevier Science, Amsterdam, The Netherlands.

Minoux, M. (1986). Mathematical programming: theory and algorithms. *Wiley-Interscience series in discrete mathematics and optimization*. Wiley.

Mitra, S., Garcia-Herreros, P., & Grossmann, I. E. (2016). A cross-decomposition scheme with integrated primal–dual multi-cuts for two-stage stochastic programming investment planning problems. *Mathematical Programming, 157*(1), 95–119.

Moreno-Centeno, E., & Karp, R. M. (2013). The implicit hitting set approach to solve combinatorial optimization problems with an application to multigenome alignment. *Operations Research, 61*(2), 453–468.

Muter, I., Birbil, c., & Bülbül, K. (2015). Benders decomposition and column-and-row generation for solving large-scale linear programs with column-dependent-rows. Optimization Online, Available at http://www.optimization-online.org/DB_FILE/2015/11/5184.pdf.

Naoum-Sawaya, J., & Elhedhli, S. (2010). A nested Benders decomposition approach for telecommunication network planning. *Naval Research Logistics (NRL), 57*(6), 519–539.

Naoum-Sawaya, J., & Elhedhli, S. (2013). An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research, 210*(1), 33–55.

Nemhauser, G. L., & Widhelm, W. B. (1971). A modified linear program for columnar methods in mathematical programming. *Operations Research, 19*(4), 1051–1060.

Nielsen, S. S., & Zenios, S. A. (1997). Scalable parallel Benders decomposition for stochastic linear programming. *Parallel Computing, 23*(8), 1069–1088.

O'Kelly, M. E., Luna, H. P. L., Camargo, R. S., & Miranda, G. (2014). Hub location problems with price sensitive demands. *Networks and Spatial Economics, 15*(4), 917–945.

Oliveira, F., Grossmann, I. E., & Hamacher, S. (2014). Accelerating Benders stochastic decomposition for the optimization under uncertainty of the petroleum product supply chain. *Computers & Operations Research, 49*, 47–58.

Osman, H., & Baki, M. (2014). Balancing transfer lines using Benders decomposition and ant colony optimisation techniques. *International Journal of Production Research, 52*(5), 1334–1350.

Pacqueau, R., Francois, S., & Le Nguyen, H. (2012). A fast and accurate algorithm for stochastic integer programming, appllied to stochastic shift scheduling. In *Publication g-2012-29, groupe d'études et de recherche en analyse des décisions (GERAD)*. Université de Montréal, Montréal, QC, Canada.

Papadakos, N. (2008). Practical enhancements to the magnanti–wong method. *Operations Research Letters, 36*(4), 444–449.

Papadakos, N. (2009). Integrated airline scheduling. *Computers & Operations Research, 36*(1), 176–195.

Paula, J., & Maculan, N. (1988). A p-median location algorithm based on the convex lagrangean relaxation of the Benders master problem. In *Presented at 13th international symposium on mathematical programming. Tokyo, Japan*.

Pérez-Galarce, F., Álvarez-Miranda, E., Candia-Véjar, A., & Toth, P. (2014). On exact solutions for the minmax regret spanning tree problem. *Computers & Operations Research, 47*, 114–122.

Peterson, B., & Trick, M. A. (2009). A Benders' approach to a transportation network design problem. In *Proceedings of the 6th international conference on integration of AI and OR techniques in constraint programming for combinatorial optimization problems. CPAIOR '09* (pp. 326–327). Berlin, Heidelberg: Springer-Verlag.

Pinheiro, P. R., & Oliveira, P. R. (2013). A hybrid approach of bundle and Benders applied large mixed linear integer problem. *Journal of Applied Mathematics*, 11. Article ID 678783

Pishvaee, M., Razmi, J., & Torabi, S. (2014). An accelerated Benders decomposition algorithm for sustainable supply chain network design under uncertainty: a case study of medical needle and syringe supply chain. *Transportation Research Part E: Logistics and Transportation Review, 67*, 14–38.

Poojari, C. A., & Beasley, J. E. (2009). Improving Benders decomposition using a genetic algorithm. *European Journal of Operational Research, 199*(1), 89–97.

Raidl, G. R. (2015). Decomposition based hybrid metaheuristics. *European Journal of Operational Research, 244*(1), 66–76.

Raidl, G. R., Baumhauer, T., & Hu, B. (2014). Speeding up logic-based Benders' decomposition by a metaheuristic for a bi-level capacitated vehicle routing problem. In M. J. Blesa, C. Blum, & S. Voß (Eds.), *9th International Workshop on Hybrid Metaheuristics. HM 2014* (pp. 183–197). Springer International Publishing, Hamburg, Germany.

Randazzo, C. D., Luna, H. P. L., & Mahey, P. (2001). Benders decomposition for local access network design with two technologies. *Discrete Mathematics and Theoretical Computer Science, 4*(2), 235–246.

Rei, W., Cordeau, J.-F., Gendreau, M., & Soriano, P. (2009). Accelerating Benders decomposition by local branching. *INFORMS Journal on Computing, 21*(2), 333–345.

Restrepo, M. I., Gendron, B., & Rousseau, L.-M. (2015). Combining Benders decomposition and column generation for multi-activity tour scheduling. In *Publication CIRRELT-2015-57, centre de recherche sur les transports*. Université de Montréal, Montréal, QC, Canada.

Roshanaei, V., Luong, C., Aleman, D. M., & Urbach, D. (2017). Propagating logic-based Benders decomposition approaches for distributed operating room scheduling. *European Journal of Operational Research, 257*(2), 439–455.

Roussel, S., Ferland, J. A., & Pradenas, L. (2004). Improving Benders decomposition to solve the tree-bucking problem, working paper. Available at http://www.iro.umontreal.ca/~ferland/Tutorial/Forestry/Tree_Bucking.pdf.

Rubiales, A., Lotito, P., & Parente, L. (2013). Stabilization of the generalized Benders decomposition applied to short-term hydrothermal coordination problem. *IEEE Latin America Transactions, 11*(5), 1212–1224.

Ruszczyński, A. (1986). A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming, 35*(3), 309–333.

Ruszczyński, A. (2003). Decomposition methods. In A. Ruszczyński, & A. Shapiro (Eds.), *Stochastic Programming. vol. 10 of Handbooks in Operations Research and Management Science* (pp. 141–211). Elsevier.

Ruszczyński, A., & Świętanowski, A. (1997). Accelerating the regularized decomposition method for two stage stochastic linear problems. *European Journal of Operational Research, 101*(2), 328–342.

Saharidis, G. K., Boile, M., & Theofanis, S. (2011). Initialization of the Benders master problem using valid inequalities applied to fixed-charge network problems. *Expert Systems with Applications, 38*(6), 6627–6636.

Saharidis, G. K., & Ierapetritou, M. G. (2009). Resolution method for mixed integer bi-level linear problems based on decomposition technique. *Journal of Global Optimization, 44*(1), 29–51.

Saharidis, G. K., & Ierapetritou, M. G. (2010). Improving Benders decomposition using maximum feasible subsystem (MFS) cut generation strategy. *Computers & Chemical Engineering, 34*(8), 1237–1245.

Saharidis, G. K., & Ierapetritou, M. G. (2013). Speed-up Benders decomposition using maximum density cut (MDC) generation. *Annals of Operations Research, 210*(1), 101–123.

Saharidis, G. K., Minoux, M., & Ierapetritou, M. G. (2010). Accelerating Benders method using covering cut bundle generation. *International Transactions in Operational Research, 17*(2), 221–237.

Sahinidis, N., & Grossmann, I. E. (1991). Convergence properties of generalized Benders decomposition. *Computers & Chemical Engineering, 15*(7), 481–491.

Santoso, T., Ahmed, S., Goetschalckx, M., & Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research, 167*(1), 96–115.

Sen, G., Krishnamoorthy, M., Rangaraj, N., & Narayanan, V. (2015). Exact approaches for static data segment allocation problem in an information network. *Computers & Operations Research, 62*, 282–295.

Sen, S., & Higle, J. L. (2005). The C3 theorem and a D2 algorithm for large scale stochastic mixed-integer programming: set convexification. *Mathematical Programming, 104*(1), 1–20.

Sen, S., & Sherali, H. D. (2006). Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming, 106*(2), 203–223.

Sherali, H. D., & Fraticelli, B. M. (2002). A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization, 22*(1–4), 319–342.

Sherali, H. D., & Lunday, B. J. (2013). On generating maximal nondominated Benders cuts. *Annals of Operations Research, 210*(1), 57–72.

Taşkın, Z. C., & Cevik, M. (2013). Combinatorial Benders cuts for decomposing IMRT fluence maps using rectangular apertures. *Computers & Operations Research, 40*(9), 2178–2186.

Tang, L., Jiang, W., & Saharidis, G. K. (2013). An improved Benders decomposition algorithm for the logistics facility location problem with capacity expansions. *Annals of Operations Research, 210*(1), 165–190.

Thorsteinsson, E. S. (2001). Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. In *Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming. CP '01* (pp. 16–30). London, UK, UK: Springer-Verlag. http://dl.acm.org/citation.cfm?id=647488.726832

Trukhanov, S., Ntaimo, L., & Schaefer, A. (2010). Adaptive multicut aggregation for two-stage stochastic linear programs with recourse. *European Journal of Operational Research, 206*(2), 395–406.

van Ackooij, W., Frangioni, A., & de Oliveira, W. (2015). Inexact stabilized Benders' decomposition approaches to chance-constrained problems with finite support. *Applied Mathematics and Computation, 270*, 193–215.

Van Roy, T. J. (1983). Cross decomposition for mixed integer programming. *Mathematical Programming, 25*(1), 46–63.

Van Slyke, R. M., & Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics, 17*(4), 638–663.

Verstichel, J., Kinable, J., De Causmaecker, P., & Berghe, G. V. (2015). A combinatorial Benders decomposition for the lock scheduling problem. *Computers & Operations Research, 54*, 117–128.

Vladimirou, H. (1998). Computational assessment of distributed decomposition methods for stochastic linear programs. *European Journal of Operational Research, 108*(3), 653–670.

Wang, Q., McCalley, J. D., Zheng, T., & Litvinov, E. (2016). Solving corrective risk-based security-constrained optimal power flow with Lagrangian relaxation and Benders decomposition. *International Journal of Electrical Power & Energy Systems, 75*, 255–264.

Wheatley, D., Gzara, F., & Jewkes, E. (2015). Logic-based Benders decomposition for an inventory-location problem with service constraints. *Omega, 55*, 10–23.

Wolf, C. (2014). *Advanced acceleration techniques for nested Benders decomposition in stochastic programming*. Universität Paderborn Master's thesis.

Wolf, C., & Koberstein, A. (2013). Dynamic sequencing and cut consolidation for the parallel hybrid-cut nested L-shaped method. *European Journal of Operational Research, 230*(1), 143–156.

Wu, P., Hartman, J. C., & Wilson, G. R. (2003). A demand-shifting feasibility algorithm for Benders decomposition. *European Journal of Operational Research, 148*(3), 570–583.

Yang, Y., & Lee, J. M. (2012). A tighter cut generation strategy for acceleration of Benders decomposition. *Computers & Chemical Engineering, 44*, 84–93.

Zakeri, G., Philpott, A. B., & Ryan, D. M. (2000). Inexact cuts in Benders decomposition. *SIAM Journal on Optimization, 10*(3), 643–657.

S. Zaourar, J.Malick, Quadratic stabilization of Benders decomposition, 2014, working paper.

Zarandi, M. M. F. (2010). *Using decomposition to solve facility location/fleet management problems*. University of Toronto Master's thesis.

Zhang, J. L., & Ponnambalam, K. (2006). Hydro energy management optimization in a deregulated electricity market. *Optimization and Engineering, 7*(1), 47–61.

Zhu, Y., & Kuno, T. (2003). Global optimization of nonconvex MINLP by a hybrid branch-and-bound and revised general Benders decomposition approach. *Industrial & Engineering Chemistry Research, 42*(3), 528–539.

Zverovich, V., Fábián, C. I., Ellison, E. F., & Mitra, G. (2012). A computational study of a solver system for processing two-stage stochastic LPs with enhanced Benders decomposition. *Mathematical Programming Computation, 4*(3), 211–238.