



UNICA

UNIVERSITÀ
DEGLI STUDI
DI CAGLIARI



This is the Author's *accepted* manuscript version of the following contribution:

Antonio Frangioni, Bernard Gendron and Enrico Gorgone, *Separable Lagrangian Decomposition for Quasi-Separable Problems*, in *Combinatorial Optimization and Applications. A Tribute to Bernard Gendron*, ISORMS 358, Springer (2024), pp. 207 - 225.

The publisher's version is available at:

http://dx.doi.org/10.1007/978-3-031-57603-4_10

When citing, please refer to the published version.

This full text was downloaded from UNICA IRIS <https://iris.unica.it/>

Separable Lagrangian Decomposition for Quasi-Separable Problems

Antonio Frangioni and Bernard Gendron and Enrico Gorgone

Abstract Lagrangian relaxation is a powerful technique that applies when the removal of some appropriately chosen set of “complicating” constraints makes a(n hard) optimization problem “much easier” to solve. The most common reason for this is that the relaxed problem fully decomposes in (a large number of) independent subproblems. However, a different case happens when the removal of the constraints leaves a number of blocks of semi-continuous variables without constraints between them except those involving the single binary variable commanding them. In this case the relaxation can still be easily solvable, but this involves a two-stage approach whereby the separable blocks are solved first, possibly in parallel, and only then one single problem can be solved to find the optimal value of the design variables. We call this a quasi-separable setting. While the relaxation can be efficiently solved, the fact that it boils down to what formally amounts to a single problem prevents from using techniques—disaggregated master problems, possibly with “easy components”—that allow to solve the corresponding Lagrangian dual more efficiently. We develop an ad-hoc reformulation of the standard master problem of (stabilised) cutting-plane approaches that allow to define the Lagrangian function as the explicit sum of different components, thereby better exploiting the actual structure of the problem, at the cost of introducing a smaller number of extra Lagrangian multipliers w.r.t. what would be required by standard approaches. We also highlight the connection between this reformulation of the master problem and the Lagrangian Decomposition technique. We computationally test our approach on one relevant problem with the re-

[Redacted]

[Redacted]

[Redacted]

quired structure, i.e., hard Multicommodity Network Design with budget constraints on the design variables, showing that the approach can outperform state-of-the-art traditional ones.

1 Introduction

The Lagrangian relaxation approach in (Linear) Mixed-Integer Programming is described in general terms as being applied to Mixed-Integer (Linear) Programs of the form

$$(P) \quad \min\{ cx : Cx = b, x \in X \}$$

where X is an “easy” set, in the sense that (P) would be “much easier” to solve if the “complicating constraints” $Cx = b$ could be removed. In general, removing the constraints outright is not a feasible option; an alternative is to form the *Lagrangian relaxation* of (P) with respect to the complicating constraints, i.e., the (sub)problem

$$(P_\lambda) \quad \phi(\lambda) = \lambda b + \min\{ cx + \lambda(b - Cx) : x \in X \}$$

for a fixed vector λ of *Lagrangian multipliers*. (P_λ) can then be “easily” solved providing a valid lower bound $\phi(\lambda) \leq v(P)$, where $v(\cdot)$ indicating the optimal value of an optimization problem. One can then consider finding the *best possible* Lagrangian relaxation, i.e., the solution of the *Lagrangian dual*

$$(D) \quad \max\{ \phi(\lambda) = v(P_\lambda) : \lambda \in \mathbb{R}^n \}$$

of (P) with respect to the complicating constraints: clearly, $v(D) \leq v(P)$, i.e., (D) is still a relaxation of (P) , but possibly providing better bounds than the standard continuous one. Solving the Lagrangian dual of (P) , possibly approximately, yields information that can be used to drive solution methods to (P) , be them exact (Holmberg and Yuan, 2000; Gendron and Larose, 2014) or heuristic (Borghetti et al., February 2003; Crainic et al., 2004). Crucially, (D) can be “easily” solved since it is a concave maximization problem. On first reading, with $X = \{ \bar{x}_1, \bar{x}_2, \dots \}$ finite but possibly very large set, one sees that

$$(D) \quad \max\{ v : v \leq c\bar{x} + \lambda(b - C\bar{x}) \quad \bar{x} \in X \} \quad (1)$$

i.e., a possibly semi-infinite linear program. While typically (D) cannot be written down explicitly, it can be solved by *row generation*; that is, one takes a “small” $\mathcal{B} \subset X$ to define the (Dual) *Master Problem*

$$(D_{\mathcal{B}}) \quad \max\{ v : v \leq c\bar{x} + \lambda(b - C\bar{x}) \quad \bar{x} \in \mathcal{B} \}, \quad (2)$$

clearly a relaxation to (D) .

W.l.o.g. it is assumed that an optimal solution $(\tilde{v}, \tilde{\lambda})$ of $(D_{\mathcal{B}})$ exists. In fact, there are appropriate tricks available to ensure it happens (Frangioni, 2005). The solution

$(\tilde{v}, \tilde{\lambda})$ can be easily checked for optimality of the whole (D) by just solving $(P_{\tilde{\lambda}})$, i.e., computing $\phi(\tilde{\lambda})$. Indeed, $\tilde{v} \leq \phi(\tilde{\lambda})$ clearly implies $\tilde{v} = \phi(\tilde{\lambda}) = v(D)$, i.e., that the problem is solved. If the condition does not hold, any optimal \bar{x} to $(P_{\tilde{\lambda}})$ corresponds to a violated constraint of (1) that can be added to \mathcal{B} , and the process iterated. The corresponding *Cutting Plane algorithm* (CP) is globally convergent and, while not efficient in its by-the-book incarnations, it is the basis of the most efficient algorithms for solving (D) with high degree of accuracy. These algorithms are known as variants of the *bundle method* (Frangioni, 2020), although simpler alternatives exist if only coarse solutions are desired (Frangioni et al., 2017).

The primal viewpoint of Lagrangian duality is also well-known, and can be easily expressed by just noting that $(D_{\mathcal{B}})$ is a simple LP and has dual

$$(P_{\mathcal{B}}) \quad \min \left\{ \sum_{\bar{x} \in \mathcal{B}} c\bar{x}\theta_{\bar{x}} : \sum_{\bar{x} \in \mathcal{B}} (b - C\bar{x})\theta_{\bar{x}} = 0, \theta \in \Theta \right\}, \quad (3)$$

where $\Theta = \{ \theta \geq 0 : \sum_{\bar{x} \in \mathcal{B}} \theta_{\bar{x}} = 1 \}$ (the unitary simplex) indicates that a convex hull operation is taking place. Indeed, linearity of the constraints easily allows to recast (3) as

$$(P_{\mathcal{B}}) \quad \min \{ cx : Cx = b, x \in X_{\mathcal{B}} = \text{conv}(\mathcal{B}) \}, \quad (4)$$

where $\text{conv}(\cdot)$ denotes the convex hull. That is, the (primal) master problem is optimising the original objective on the intersection between the feasible region defined by the complicating constraints and the convex hull $X_{\mathcal{B}}$ generated by the known set of points in \mathcal{B} . With $\mathcal{B} = X$ (and ignoring formal complications for brevity), (4) is the *Dantzig-Wolfe reformulation* of the *convexified relaxation* of (P)

$$(\tilde{P}) \quad \min \{ cx : Cx = b, x \in \text{conv}(X) \} \quad (5)$$

to which then (D) is equivalent. This immediately implies that, under appropriate assumptions (D) can provide tighter bounds than the ordinary continuous relaxation. This happens when X is not “too easy”, i.e., its linear formulation, say $Ex \leq d$, does not have the integrality property (Frangioni, 2005). This fact justifies the interest in the technique.

This succinct description is however neglecting a number of details that are crucial computationally. Perhaps the most important one is that often (but not always) the relaxed constraints are “complicating” because they are “linking”; that is, $X = X^1 \times X^2 \times \dots \times X^k$ is a Cartesian product of sets that have no joint constraints other than these. This means that the problem can be written as

$$(P) \quad \min \left\{ \sum_{h \in K} c^h x^h : \sum_{h \in K} C^h x^h = b, x^h \in X^h, h \in K \right\} \quad (6)$$

with $K = \{ 1, \dots, k \}$. Hence, the Lagrangian relaxation is separable in k independent subproblems:

$$(P_{\lambda}) \quad \phi(\lambda) = \lambda b + \sum_{h \in K} \min \left\{ (c^h - \lambda A^h) x^h : x^h \in X^h \right\}.$$

One advantage is of potentially making it faster to solve (P_λ) by means of parallel processing techniques (Cappanera and Frangioni, 2003), but this is likely not the most important aspect. Rather, having independent subproblems allows to collect independent subsets $\mathcal{B}^h \subset X^h$ and use them to construct *disaggregated cuts* in the Dual Master Problem (Bacaud et al., 2001; Borghetti et al., February 2003). The disaggregation makes it roughly a factor of k larger in terms of constraints, and therefore more costly to solve, after the same number of iterations and save *selection* and *compression* techniques that can be used to reduce its size. Nonetheless, this typically results in a much faster convergence and therefore ultimately better performances (Jones et al., 1993).

The reason is easily understood by looking at the *disaggregated Primal Master Problem*

$$(P_{\mathcal{B}}) \quad \min \left\{ \sum_{h \in K} c^h \sum_{\bar{x}^h \in \mathcal{B}^h} \bar{x}^h \theta_{\bar{x}^h} : \sum_{h \in K} C^h \sum_{\bar{x}^h \in \mathcal{B}^h} \bar{x}^h \theta_{\bar{x}^h} = b, \right. \\ \left. \sum_{\bar{x}^h \in \mathcal{B}^h} \theta_{\bar{x}^h} = 1 \quad h \in K \right\} \quad (7)$$

It is plain to see that (3) is the restriction of (7) where all the components \bar{x}^h corresponding to the solution \bar{x} obtained at some iteration must have the same convex multiplier. Hence, for the same $\mathcal{B} \subseteq X$ (7) has more solutions than (3). Intuitively, forcing each component of \bar{x} to have the same multiplier does not make sense: in other words, the Cartesian product of the $\text{conv}(\mathcal{B}^h)$ is a better approximation of $\text{conv}(X)$ than $\text{conv}(\mathcal{B})$, usually so much so as to amply counterbalance the (significant) increase in the number of variables. This is largely because the process terminates after much fewer iterations, hence the average size of the solved master problems (often, but not always the computational bottleneck of the process) may still be significantly lower.

The underlying intuition is that the structure of the original problem (P) should be exploited as much as possible when constructing its Dantzig-Wolfe reformulation. Another instance of this concept is the “easy components” variant of the approach (Frangioni and Gorgone, 2014), which exploits the fact that the different Lagrangian subproblems need not be all of the same type. For illustration, consider an original problem (P) of the form

$$(P) \quad \min \{ c^1 x^1 + c^2 x^2 : C^1 x^1 + C^2 x^2 = b, \quad x^1 \in X, \quad E x^2 \leq d \} \quad (8)$$

i.e., where the feasible region X is the Cartesian product of a generic set X^1 (which we only know we can efficiently optimise upon) and of a set X^2 that is representable by a “simple, small” (linear) formulation. Clearly one could apply the disaggregated approach (7), as optimising over X^2 is “easy”. However, this is not the best approach: representing of X^2 via its vertices yields an exponential number of points, while a compact linear formulation is available. An arguably better solution is that only the first group of variables is Dantzig-Wolfe reformulated, while the second is just kept as it is, i.e.,

$$(P_B) \quad \min \left\{ c^1 \left(\sum_{\bar{x}^1 \in \mathcal{B}^1} \bar{x}^1 \theta_{\bar{x}^1} \right) + c^2 x^2 : C^1 \left(\sum_{\bar{x}^1 \in \mathcal{B}^1} \bar{x}^1 \theta_{\bar{x}^1} \right) + C^2 x^2 = b, \quad (9) \right. \\ \left. \sum_{\bar{x}^1 \in \mathcal{B}^1} \theta_{\bar{x}^1} = 1, \quad \theta \geq 0, \quad E x^2 \leq d \right\}$$

While such a master problem can initially be larger than the standard one, it contains “perfect” information on the feasible set of the x_2 variables, which therefore the solution process does not need to painstakingly accrue iteratively. This clearly holds for any number of “easy” and “difficult” components. This approach usually decreases the total number of iterations required to solve the problem and provides a positive trade-off on the total solution time (Scuzziato et al., 2018).

Hence, the Dantzig-Wolfe reformulation approach underlying Lagrangian duality is not necessarily a rigid scheme; rather, it possesses degrees of flexibility that can be exploited to make maximal use of the existing structure of the underlying problem. In this paper we consider one structure that, to the best of our knowledge, has not been analysed in this sense despite being relatively common. This is the case of a problem having several groups of semi-continuous variables x_i , each one “governed” by a single binary variables y_i , as in

$$A_i x_i \leq b_i y_i \quad i \in N \quad (10)$$

$$x_i \in X_i \cup \{0\} \quad i \in N \quad (11)$$

$$y \in Y \quad (12)$$

where $X_i \subseteq \{x_i \in \mathbb{R}^{n_i} : A_i x_i \leq b_i\}$. That is, the semi-continuous variables have a (potentially) disjoint feasible region, which we denote by $X'_i = X_i \cup \{0\}$. When $y_i = 0$ the condition $x_i = 0$ is enforced, while when $y_i = 1$ the condition $x_i \in X_i$ rather is. The textbook example is that of a single variable $x_i \in \{0\} \cup [l_i, u_i]$, with $0 < l_i < u_i < \infty$, describing a machinery or an activity that is either off, and therefore produces nothing, or on and therefore has to produce at least a minimum positive quantity (l_i). Although $0 \in X_i$ can also happen, a cost may be incurred to “activate the process”, i.e., to have $y_i = 1$ (that is, the objective contains a term $c_i y_i$ with $c_i \neq 0$). Thus, the case where $0 \in X_i$ is typically not easier than $0 \notin X_i$ and we consider the latter for full generality.

Removing a set of complicating constraints, say $Dy + \sum_{i \in N} C_i x_i = b$, the problem does not decompose as a sum of independent subproblems, but can still be solved by applying two separate solution approaches “in cascade”—one of which exploiting a degree of decomposability—because the variables can be partitioned into k groups of semi-continuous ones. This is due to the fact that there are no other constraints—apart for the complicating ones—linking variables in different blocks, save for those linking each group to the binary variable “governing” it, and (possibly) the ones linking the binary variables among themselves. In such a case, the Lagrangian subproblem is formally undivided, and therefore the use of disaggregated master problems a-la (7) is not possible, which may have a significant detrimental effect on the ability to efficiently solve the Lagrangian dual. This is counter-intuitive, in that there clearly is a significant element of separability in these *quasi-separable problems*. We extend and computationally validate the results of (Frangioni et al., 2020) by showing that a relatively minor modification of the master problem formulation allows to efficiently

inject a full separable structure into the approach, at the cost of a (likely) relatively minor increase of the number of Lagrangian multipliers. We contrast our result with the “brute force” approach in which all the constraints linking the semi-continuous variables and the corresponding integer ones are relaxed, which yields an analogous final effect with a possibly weaker bound, but at the cost of a (likely) much greater number of Lagrangian multipliers. After having introduced the idea in the master problem space we link it to an application of the *Lagrangian decomposition* approach which gives it a different perspective.

The structure of the Chapter is the following. In Section 2 we formally introduce the family of quasi-separable problems and discuss different possible master problem (Dantzig-Wolfe) formulations for them. In Section 3 we describe our target application, the fixed-charge multicommodity capacitated network design problem, and how the results of the previous sections apply to it. Then, in Section 4 we describe our computational experience aimed at verifying the practical efficiency of the newly proposed approach, and Section 5 provides some closing discussion.

2 Master Problem Reformulations For Quasi-Separable Problems

We now formally present the relevant formulations. For a finite index set $N = \{1, \dots, n\}$, a set $Y \subseteq \{0, 1\}^n$ is given, represented by a vector of binary variables $y = [y_i]_{i \in N}$, together with n sets $X_i \subseteq \mathbb{R}^{n_i}$ and the corresponding partitioned vector(s) of variables $x = [x_i]_{i \in N}$ as in (10)–(12). That is, each x_i is a semi-continuous variable “governed” by the corresponding y_i , in the sense that $y_i = 0 \implies x_i = 0$, while $y_i = 1 \implies x_i \in X_i$. We assume that (10) define a compact polyhedron when $y_i = 1$, i.e., such that $\{x_i \in \mathbb{R}^{n_i} : A_i x_i \leq 0\} = \{0\}$. Clearly, these constraints also influence the shape of the feasible set for x_i ; that is, without loss of generality we can assume that $X_i \subseteq \{x_i \in \mathbb{R}^{n_i} : A_i x_i \leq b_i\}$ (which immediately implies that each X_i is compact). The sets X_i may entail integrality restrictions, but since we are concerned with relaxations (of the Lagrangian type) we will actually only work with the convex hull of X_i ; hence, to ease on notation we will consider X_i to be convex, and compact. The relevant object is then the set \bar{X}_i of the extreme points of X_i , so that $X_i = \text{conv}(\bar{X}_i)$. It is intuitive, and it will be confirmed later on, that a slight difference exists between the case where $0 \in X_i$ and that where $0 \notin X_i$. It will therefore be expedient to introduce $X'_i = X_i \cup \{0\}$, which may or may not coincide with X_i , and its set of extreme points \bar{X}'_i . The set Y is decidedly integer, but as Lagrangian relaxations cannot distinguish a set from its convex hull, we will basically only work with the latter; yet, because of its binary nature, $Y = \text{ext}(Y)$ and therefore we don’t need to explicitly define the set of extreme points. An “abstract” formulation of our problem (neglecting the details of how X'_i and Y are formulated) is

$$\max dy + \sum_{i \in N} c_i x_i \quad (13)$$

$$Dy + \sum_{i \in N} C_i x_i = b \quad (14)$$

$$A_i x_i \leq b_i y_i \quad i \in N \quad (15)$$

$$x_i \in X'_i \quad i \in N \quad (16)$$

$$y \in Y \quad (17)$$

Note that in the problem we have to impose $x_i \in X'_i$, rather than $x_i \in X_i$ (although the two may indeed be equal), to make it possible to set $x_i = 0$ when $y_i = 1$. The m constraints (14) are the “linking” ones, whose removal makes the problem (much) easier to solve. Hence, it may make sense to form the standard Lagrangian relaxation w.r.t. (14)

$$\phi(\lambda) = \lambda b + \max \left\{ (d - \lambda D)y + \sum_{i \in N} (c_i - \lambda C_i)x_i : (15), (16), (17) \right\} \quad (18)$$

and aim at (efficiently) solving the corresponding Lagrangian dual. This requires being able to compute $\phi(\lambda)$ efficiently. Contrarily to what often happens with Lagrangian relaxation, however, $\phi(\lambda)$ is not easy to solve because it decomposes in the sum of independent problems. Rather, due to the semi-continuous nature of the x_i one can first solve independent subproblems in those variables, and then a unique problem in the y ones using the results. More formally, one can define

$$\phi_i(\lambda) = \max \left\{ (c_i - \lambda C_i)x_i : x_i \in X_i \right\} \quad (19)$$

(where note that, unlike in (16), the original X_i is used instead of the “extended” X'_i , although of course the two may coincide). Clearly, $\phi_i(\lambda)$ is the best value one can ever hope for, for the fixed λ , to be provided by the x_i variables; we will denote by x_i^* any one of the corresponding optimal solutions, eschewing the dependency on λ for notational simplicity. It is then clear that

$$\phi(\lambda) = \lambda b + \max \left\{ \sum_{i \in N} (d_i - \lambda D^i + \phi_i(\lambda))y_i : y \in Y \right\} . \quad (20)$$

Provided that (19) (n independent problems) and (20) (one single problem concerning the y variables only) can be efficiently solved, the Lagrangian function can be easily computed. For any y^* optimal for (20), clearly (y^*, \bar{x}) is optimal for (18) with $\bar{x}_i = 0$ if $y_i^* = 0$, and $\bar{x}_i = x_i^*$ if $y_i^* = 1$. Hence, $(b - Dy^* + \sum_{i \in N} C_i \bar{x}_i)$ is a valid subgradient of ϕ in λ —providing the linear inequality used in the *aggregated* dual master problem (1)—which can be used in any of the known approaches to seek for the optimal λ^* .

While all this is perfectly sensible, we can argue that the process somewhat “hides” some of the precious structure of the problem. In particular, computing the Lagrangian function actually boils down to solving $n + 1$ separate subproblems, but *not independent* ones. Hence the Lagrangian function is not a simple sum-function, and therefore *disaggregated master problems* a-la (7) cannot be used, losing one major opportunity at improving convergence speed. There is of course a simple way to make the function separable: also relax (in Lagrangian fashion) the constraint (15)

with multipliers $\mu = [\mu_i]_{i \in N} \geq 0$, which yields the Lagrangian function

$$\phi(\lambda, \mu) = \lambda b + \psi(\lambda, \mu) + \sum_{i \in N} \psi_i(\lambda, \mu_i) \quad (21)$$

with

$$\psi_i(\lambda, \mu_i) = \max \{ (c_i - \lambda C_i - \mu_i A_i)x_i : x_i \in X_i \} , \quad (22)$$

$$\psi(\lambda, \mu) = \max \{ \sum_{i \in N} (d_i - \lambda D^i - \mu_i b_i)y_i : y \in Y \} . \quad (23)$$

Clearly, (22) and (23) can be expected to be efficiently solvable if (19) and (20) are; note that, actually, in (19) one may use X_i *without* the constraints $A_i x_i \leq b_i$, should this help in any way. Then, $\phi(\lambda, \mu)$ is a classical sum-function for which the corresponding approaches can be used. The dual space is extended when relaxing more constraints, which is necessary to obtain a better decomposition; hence, the Lagrangian dual corresponding to maximising (18) can provide a strictly better bound than that corresponding to maximising (21). However, the results of (Frangioni et al., 2020, §3) show that this is not happening if Y has the integrality property, as in our application (cf. §3); we will return to this equivalence later on. Thus, in this case the two Lagrangian approaches provide alternative ways to compute the same (strong) bound. However, the drawback of the disaggregated approach is that (15) can be “many”, which means that the (λ, μ) space might be much larger than the λ space; this increase in size may well negate the advantage of having a disaggregated function, as our computational results will confirm. One would then therefore have a way to reap the algorithmic benefits of disaggregation without the significant extra cost incurred by relaxing all the constraints (15).

2.1 The Master Problem Reformulation

Our proposal hinges on a reformulation of the two classical Dantzig-Wolfe reformulations of the problem corresponding to the two Lagrangian approaches described above. For the first, we define by YX the set of extreme points $(\bar{y}, \bar{x} = [\bar{x}_i]_{i \in N})$ of (18), that can be obtained (for any given value of λ) with the procedure illustrated above; then, the standard the Dantzig-Wolfe relaxation of (13)–(17) is

$$\max \sum_{(\bar{y}, \bar{x}) \in YX} (d\bar{y} + \sum_{i \in N} c_i \bar{x}_i) \theta_{(\bar{y}, \bar{x})} \quad (24)$$

$$\sum_{(\bar{y}, \bar{x}) \in YX} (D\bar{y} + \sum_{i \in N} C_i \bar{x}_i) \theta_{(\bar{y}, \bar{x})} = b \quad (25)$$

$$\sum_{(\bar{y}, \bar{x}) \in YX} \theta_{(\bar{y}, \bar{x})} = 1 \quad (26)$$

$$\theta_{(\bar{y}, \bar{x})} \geq 0 \quad (\bar{y}, \bar{x}) \in YX \quad (27)$$

For the decomposable version (21), the Dantzig-Wolfe relaxation rather is

$$\max \sum_{\bar{y} \in Y} (d\bar{y})\theta_{\bar{y}} + \sum_{i \in N} \sum_{\bar{x}_i \in \bar{X}_i} (c_i \bar{x}_i)\theta_{\bar{x}_i} \quad (28)$$

$$\sum_{\bar{y} \in Y} (D\bar{y})\theta_{\bar{y}} + \sum_{i \in N} \sum_{\bar{x}_i \in \bar{X}_i} (C_i \bar{x}_i)\theta_{\bar{x}_i} = b \quad (29)$$

$$\sum_{\bar{x}_i \in \bar{X}_i} (A_i \bar{x}_i)\theta_{\bar{x}_i} \leq \sum_{\bar{y} \in Y} (b_i \bar{y}_i)\theta_{\bar{y}} \quad i \in N \quad (30)$$

$$\sum_{\bar{y} \in Y} \theta_{\bar{y}} = 1 \quad (31)$$

$$\sum_{\bar{x}_i \in \bar{X}_i} \theta_{\bar{x}_i} \leq 1 \quad i \in N \quad (32)$$

$$\theta_{\bar{y}} \geq 0 \quad \bar{y} \in Y \quad (33)$$

$$\theta_{\bar{x}_i} \geq 0 \quad \bar{x}_i \in \bar{X}_i \quad (34)$$

We remark that the constraints (32) would ordinarily be equalities. In fact, they are so if the sum is extended to \bar{X}'_i , i.e., if 0 is added to the extreme points: then, the variable θ_0 acts as a slack—it has zero coefficients everywhere else except (32)—so we can rather use \bar{X}_i as the index set and consider the constraint an inequality (note, again, that $0 \in X_i$ might happen anyway). What we propose here is a modification of (28)–(34) that directly exploits the specific quasi-separable structure, i.e., the constraints (15) defining the semi-continuous variables:

$$\max \sum_{\bar{y} \in Y} (d\bar{y})\theta_{\bar{y}} + \sum_{i \in N} \sum_{\bar{x}_i \in \bar{X}_i} (c_i \bar{x}_i)\theta_{\bar{x}_i} \quad (28)$$

$$\sum_{\bar{y} \in Y} (D\bar{y})\theta_{\bar{y}} + \sum_{i \in N} \sum_{\bar{x}_i \in \bar{X}_i} (C_i \bar{x}_i)\theta_{\bar{x}_i} = b \quad (29)$$

$$\sum_{\bar{x}_i \in \bar{X}_i} \theta_{\bar{x}_i} = \sum_{\bar{y} \in Y} \theta_{\bar{y}} \quad i \in N \quad (35)$$

$$(31), (32), (33), (34) \quad (36)$$

Basically, the idea is just to replace (30) with the simpler, “logically equivalent” constraint (35). If Y is an “easy set” (which implies the integrality property and therefore the result of (Frangioni et al., 2020, §3)) then (28)–(34) can be given the “easy component” treatment (9), which just corresponds to the *partial* DW reformulation whereby the X_i are expressed as convex combination of their extreme points, but Y is not:

$$\max dy + \sum_{i \in N} \sum_{\bar{x}_i \in \bar{X}_i} (c_i \bar{x}_i)\theta_{\bar{x}_i} \quad (37)$$

$$Dy + \sum_{i \in N} \sum_{\bar{x}_i \in \bar{X}_i} (C_i \bar{x}_i)\theta_{\bar{x}_i} = b \quad (38)$$

$$\sum_{\bar{x}_i \in \bar{X}_i} (A_i \bar{x}_i)\theta_{\bar{x}_i} \leq b_i y_i \quad i \in N \quad (39)$$

$$(17), (32), (34) \quad (40)$$

Our proposal may actually look more striking if applied to this “easy component” version, where the constraint to be replaced is rather (39):

$$\max dy + \sum_{i \in N} \sum_{\bar{x}_i \in \bar{X}_i} (c_i \bar{x}_i)\theta_{\bar{x}_i} \quad (37)$$

$$Dy + \sum_{i \in N} \sum_{\bar{x}_i \in \bar{X}_i} (C_i \bar{x}_i)\theta_{\bar{x}_i} = b \quad (38)$$

$$\sum_{\bar{x}_i \in \bar{X}_i} \theta_{\bar{x}_i} = y_i \quad i \in N \quad (41)$$

$$(17), (32), (34) \quad (42)$$

That is, the only required modification is that the “many” extra linking constraints (15)/(30)/(39), that one has to relax if separability has to be attained, are replaced with “much fewer” linking constraints (35)/(41): exactly one for each $i \in N$. According to (Frangioni et al., 2020, Corollary 1), (28)–(36) provides the same bound as (24)–(27). If Y has the integrality property then (37)–(40) yields the same bound and therefore (37)–(42) also does; hence, due to the cited result, every version is equivalent to (28)–(34). We will comment further on these equivalences in the next subsection. Here we focus on the fact that we can efficiently solve (37)–(42) by pricing procedures that are obvious modifications to (22)/(23). We start by introducing the extended set W_i of vectors of the form $[x_i, v]$, with $v \in \{0, 1\}$, where $x_i \in X_i$ and $v = 1$ if $x_i \neq 0$, while $v = 0$ if $x_i = 0$. Then, one is solving

$$\psi_i(\lambda, \mu_i) = \max \{ (c_i - \lambda C_i)x_i + \mu_i v : [x_i, v] \in W_i \} \quad (43)$$

which is clearly not significantly more expensive than (22). In fact, if $\psi_i(\lambda) + \mu_i > 0$, which clearly implies $x_i^* \neq 0$, then this is kept as the solution, and therefore $v^* = 1$; otherwise, the solution $[x_i^*, v^*] = [0, 0]$ is preferable (it has 0 value, which is better than negative in a maximization problem) and it is taken instead. The required variant of (23) is rather just

$$\psi(\lambda, \mu) = \max \{ \sum_{i \in N} (d_i - \lambda D^i - \mu_i) y_i : y \in Y \} . \quad (44)$$

Given the optimal solutions $[x_i^*, v^*]$ and y^* of (43) and (44), respectively,

$$\begin{bmatrix} -C_i x_i^* \\ e_i v^* \end{bmatrix} \in \partial \psi_i(\lambda, \mu_i) \quad \text{and} \quad \begin{bmatrix} -D_i y_i^* \\ -y^* \end{bmatrix} \in \partial \psi(\lambda, \mu)$$

(where e_i is the i -th vector of the canonical base of \mathbb{R}^n) are clearly the right new columns to be added to (28)–(36). In the “easy component” version, only (43) need be solved. Note that that (35)/(41) together with with the “true” form of (32), i.e.,

$$\sum_{\bar{x}_i \in \bar{X}_i} \theta_{\bar{x}_i} = 1$$

would imply $y_i = 1$ in each feasible solution to the master problem. Yet, our solution avoids the issue since the multiplier corresponding to the all-0 x_i solution do not participate to (35)/(41), as the column inserted in the master problem is $[0, 0]$ when $x_i^* = 0$.

The master problems (28)–(36) and (37)–(42) have typically far fewer “extra” multipliers μ than their original counterparts (28)–(34) and (37)–(40); the reduced dual space should make the solution of the corresponding Lagrangian duals easier, while retaining the separability advantage over the aggregated (24)–(27). This will be confirmed in the computational section.

2.2 Relationship With Lagrangian Decomposition

We now discuss an alternative derivation of the results of the previous paragraph in terms of the well-known *Lagrangian decomposition* approach (Guignard and Kim, 1987). The idea is to reformulate (13)–(17) as

$$\max dy + \sum_{i \in N} c_i x_i \quad (13)$$

$$Dy + \sum_{i \in N} C_i x_i = b \quad (14)$$

$$A_i x_i \leq b_i v_i \quad i \in N \quad (45)$$

$$v_i = y_i \quad i \in N \quad (46)$$

$$v_i \in \{0, 1\} \quad i \in N \quad (47)$$

$$x_i \in X'_i \quad i \in N \quad (16)$$

$$y \in Y \quad (17)$$

That is, we create an explicit copy v of the design variables y and we use it in the semi-continuous constraints (45), while keeping the original y in whatever constraints are needed to express (17): of course we need also add the corresponding “copy constraints” (46), that are then treated as “complicating” ones. In so doing the Lagrangian relaxation decomposes in the problems (43) and (44) as in the previous section, and one readily recovers the corresponding master problems (28)–(36) and (37)–(42). Looked at in reverse, this provides an interesting interpretation of the master problem of the Lagrangian Decomposition in our model, justifying why the approach should be expected to provide better results than the direct relaxation of the semi-continuous constraints: basically, one is using a more compact form *in the master problem* of the logical constraints “ $y_i = 0 \implies x_i = 0$ ”.

In turn, this interpretation also provides a different way to prove the equivalence results of (Frangioni et al., 2020, §2). In fact, once the complicating constraints (14) are relaxed, the formulation (13)–(17) is a *star-shaped MI(N)LP* (Bacci et al., 2023): its variables can be partitioned into $n + 1$ subsets, the “core” one that contain all linking variables and n “ray” ones, each of which only shares one single binary variable with the “core”. (Bacci et al., 2023, Lemma 1) (applied repeatedly as in (Bacci et al., 2023, Theorem 1)) then proves that a formulation of the convex hull of the whole problem can be simply obtained by concatenating the formulations of the convex hulls of the $n + 1$ subsets. Applied in our DW reformulation setting, this easily proves that the relaxation of (45) and (46) provides a bound that is as strong as that of the aggregated relaxation (19)–(20). The bound can actually be stronger than the one obtained by the natural disaggregated relaxation in case Y does not have the integrality property; however, we are more interested in the impact on the computational performances of the solution to the Lagrangian dual. For this reason in the next Section we will concentrate on an application where Y does have the integrality property, so that the result of (Frangioni et al., 2020, §3) applies negating any bound advantage to the new approach and thereby being a “worst case scenario” for the technique.

3 Application To Multicommodity Network Design

The Fixed-Charge Multicommodity Capacitated Network Design problem (FC-MCND) is a general network design problem with many applications (see, e.g., (Crainic et al., 1999; Chouman et al., 2016; Frangioni and Gorgone, 2014; Frangioni et al., 2017) and the references therein). Efficiently computing tight lower bounds on its optimal value is crucial for solution approaches, and Lagrangian techniques have been repeatedly shown to be competitive (Balakrishnan and Magnanti, 1989; Crainic et al., 2004, 2001; Frangioni and Gorgone, 2014; Frangioni et al., 2017, 2018; Gendron, 2019; Holmberg and Yuan, 2000; Crainic and Gendron, 2021).

The data of FC-MCND is a directed graph $G = (N, A)$, where F_i and B_i respectively denote the set of outbound and inbound arcs of node $i \in N$, and a set of commodities K . Each $k \in K$ has a *deficit vector* $b^k = [b_i^k]_{i \in N}$ that denotes the net amount of flow asked at each node. Each arc $(a_+, a_-) = a \in A$ can only be used if the corresponding *fixed cost* $f_a > 0$ is paid, in which case the *mutual capacity* $u_a > 0$ bounds the total amount of flow on a , while *individual capacities* u_a^k bound the flow of commodity k . The *routing cost* c_a^k has to be paid for each unit of commodity k moving through a . The flow variables are denoted by x_a^k , for each $a \in A$ and $k \in K$, while the design variables are denoted by y_a , $a \in A$. While the most prevalent form of the problem in the literature does not have constraints directly linking design variables among themselves, making the design set Y just a binary hypercube, we assume the presence of a simple *budget constraint* that limits the maximum number of arcs to be opened, irrespectively of their cost. The formulation is then

$$\min \sum_{k \in K} \sum_{a \in A} c_a^k x_a^k + \sum_{a \in A} f_a y_a \quad (48)$$

$$\sum_{a \in F_i} x_a^k - \sum_{a \in B_i} x_a^k = b_i^k \quad i \in N, k \in K \quad (49)$$

$$\sum_{k \in K} x_a^k \leq u_a \quad a \in A \quad (50)$$

$$x_a^k \leq u_a^k y_a \quad a \in A, k \in K \quad (51)$$

$$0 \leq x_a^k \leq u_a^k \quad a \in A, k \in K \quad (52)$$

$$\sum_{a \in A} y_a \leq K_A \quad (53)$$

$$y_a \in \{0, 1\} \quad a \in A \quad (54)$$

A classical approach to FC-MCND for deriving lower bounds on its optimal value is the *knapsack relaxation* (KR) (Crainic et al., 2001; Frangioni and Gorgone, 2014; Frangioni et al., 2018), whereby one dualizes the flow conservation constraints (49) with multipliers $\lambda = [\lambda_i^k]_{i \in N, k \in K}$, yielding the objective function

$$\min \sum_{a \in A} \left[\sum_{k \in K} (c_a^k + \lambda_{a_+}^k - \lambda_{a_-}^k) x_a^k + f_a y_a \right] + \sum_{i \in N} \sum_{k \in K} \lambda_i^k b_i^k$$

If the budget constraint (53) were not present, the Lagrangian relaxation would reduce to $|A|$ continuous knapsack problems (KP) of the type

$$g^a(\lambda) = \min \left\{ \sum_{k \in K} [\bar{c}_a^k x_a^k] : \sum_{k \in K} x_a^k \leq u_a, (52) \right\} \quad (55)$$

being $\bar{c}_a^k = c_a^k + \lambda_{a_+}^k - \lambda_{a_-}^k$; once $g^a(\lambda)$ is computed, the trivial one-dimensional problem

$$\min \{ (g^a(\lambda) + f_a)y_a : (54) \}$$

can be solved independently for all arcs, retaining full separability. The presence of (53) immediately changes this to a quasi-separable setting, in which the (simple) knapsack problem on the y variables

$$\min \{ \sum_{a \in A} (g^a(\lambda) + f_a)y_a : (53) , (54) \}$$

has to be solved, linking the $|A|$ KP together and killing separability in the process. The “obvious” way to retain separability is dualizing the forcing capacity constraints (51) too, i.e., working in the (much) extended multipliers space $\lambda = [\gamma, \beta] = [\gamma_i^k, \beta_a^k \geq 0]_{i \in N, a \in A, k \in K}$. This yields the objective function

$$\min \sum_{k \in K} \sum_{a \in A} (c_a^k + \gamma_{a_+}^k - \gamma_{a_-}^k + \beta_a^k)x_a^k + \sum_{a \in A} (f_a - \sum_{k \in K} \beta_a^k u_a^k)y_a$$

(besides the constant term $f^0(\lambda) = \sum_{i \in N} \sum_{k \in K} \gamma_i^k b_i^k$), whose minimization subject to the remaining (50), (52)–(54) reduce to $|A|$ KP of the form (55) except with $\bar{c}_a^k = c_a^k + \gamma_{a_+}^k - \gamma_{a_-}^k + \beta_a^k$. Independently from these one solves

$$f^1(\lambda) = \min \{ \sum_{a \in A} (f_a - \sum_{k \in K} \beta_a^k u_a^k)y_a : (53) \} .$$

This leads to a decomposable objective function of the form $f^0(\lambda) + f^1(\lambda) + \sum_{a \in A} g^a(\lambda)$, which we call the Fully decomposed Relaxation (FR).

According to what has been discussed in §2.2 the new form of the master problem can be obtained, following (45)–(47), by the simple approach of reformulating FC-MCND as

$$\min \sum_{k \in K} \sum_{a \in A} c_a^k x_a^k + \sum_{a \in A} f_a y_a \quad (48)$$

$$\sum_{a \in F_i} x_a^k - \sum_{a \in B_i} x_a^k = b_i^k \quad i \in N, k \in K \quad (49)$$

$$\sum_{k \in K} x_a^k \leq u_a v_a \quad a \in A \quad (56)$$

$$x_a^k \leq u_a^k v_a \quad a \in A, k \in K \quad (57)$$

$$v_a = y_a \quad a \in A \quad (58)$$

$$0 \leq x_a^k \leq u_a^k \quad a \in A, k \in K \quad (52)$$

$$\sum_{a \in A} y_a \leq K_A \quad (53)$$

$$y_a \in \{0, 1\} \quad a \in A \quad (54)$$

$$v_a \in \{0, 1\} \quad a \in A \quad (59)$$

Then, one relaxes (58) in addition to (49) with multipliers $\lambda = [\mu_a, \lambda_i^k]_{a \in A, i \in N, k \in K}$; this yields the objective function:

$$\min \sum_{a \in A} [\sum_{k \in K} (c_a^k + \lambda_{a_+}^k - \lambda_{a_-}^k)x_a^k + f_a y_a + \mu_a(v_a - y_a)]$$

(besides the usual constant term), whose minimization subject to (56)–(59) reduces to $|A|$ independent KP and a simple problem defined on the y variables. Indeed, for each arc $a \in A$, one has to solve the usual (55) with $\bar{c}_a^k = c_a^k + \lambda_{a_+}^k - \lambda_{a_-}^k$, and then the trivial

$$f^a(\lambda) = \min \{ (g^a(\lambda) + \mu_a) v_a : (59) \} .$$

Independently from the evaluation of $f^a(\lambda)$, one solves

$$f^1(\lambda) = \min \{ \sum_{a \in A} (f_a - \mu_a) y_a : (53) \} ,$$

overall leading again to a decomposable objective function of the form $f^0(\lambda) + f^1(\lambda) + \sum_{a \in A} f^a(\lambda)$, except with the terms defined differently from the (FR) case. We call this model the Quasi-separable Relaxation (QR).

4 Numerical Experiments

We have solved all the Lagrangian duals by means of a *bundle method*, i.e., a stabilised version of the Cutting Plane algorithm that corresponds to the naïve version of the Dantzig-Wolfe decomposition. Proper stabilisation is well-known to be crucial for performances and typically corresponds to modifications to the master problem: luckily, most general forms of stabilisation change the master problem in a simple way and are therefore independent from other modifications aimed at ensuring better performances, see, e.g., (Frangioni and Gendron, 2013; Frangioni and Gorgone, 2014). Thus we have so far avoided to explicitly describe stabilisation and we will continue to do so; the interested reader can consult, e.g., (Frangioni, 2020) for details.

The implementation of the bundle method we have been using is in C++, within the NDOSolver/FiOracle open-source project available at

https://gitlab.com/frangio68/ndosolver_fioracle_project

that have already been successfully used a number of times in this context (Frangioni and Gorgone, 2014; Frangioni et al., 2017, 2018). The code was compiled with GNU g++ 4.4.5 (with `-O3` optimization option) and ran on an Opteron 6174 processor (2.2 GHz) with 32 GB of RAM, under a i686 GNU/Linux operating system.

The numerical experiments have been performed on 80 randomly generated instances already used in several papers (Frangioni and Gorgone, 2014; Frangioni et al., 2017), and available at

<https://commalab.di.unipi.it/datasets/mmf/#Canada>

These instances are of problems without constraints on the design variables, so we had to figure out a significant value for the constant K_A of the arc budget constraint (53). This has been chosen in the following way: first, we solved the continuous relaxation of the model excluding (53) by a general-purpose solver, then we computed the number aa of active arcs (with $y_a > 0$) in the optimal continuous solution and we set $K_A = 0.8 * aa$. The factor 0.8 was the smallest one that consistently produced

feasible solutions: any K_A chosen with a smaller factor produces some unfeasible instances.

Since KR results in a single Lagrangian relaxation and therefore a unique function, we have used an aggregated bundle method (BKR) to solve it. The advantage is that we have therefore been able to solve the (aggregated) master problem with (quadratic) stabilization by using an efficient specialized quadratic solver (Frangioni, 1996). Despite this the master problem time is still by far the bottleneck of the approach, and therefore we have experienced with using different maximum sizes of the bundle (set of collected subgradients/columns): this is possible with quadratic stabilisation due to *selection* and *aggregation* techniques (Frangioni, 2020). In particular we report the results with a “small” maximum bundle size and a “large” one, set to 10 and 4000, respectively; in Table 1 we these are the columns labeled *BKR-10* and *BKR-4k*, respectively. We remark with bundle methods with “small” (“minimal”) maximum bundle size behave in practice somehow like subgradient-type methods (Bahense et al., 2002), and therefore BKR-10 can be considered a proxy of what a subgradient approach may provide, except subgradient methods hardly ever provide solutions with accuracy greater than $1e-4$ (Frangioni et al., 2017), while bundle methods can, as our results show.

To solve FR and QR we have rather used a disaggregated bundle method where each subproblem produces separate subgradients/columns in the master problem, a-la (7). Since a specialized solver for the disaggregated master problem is not available yet, the master problem has been solved by IBM-ILOG CPLEX 20.1. We have therefore performed a significant tuning phase to find the stabilising term and the solution algorithm that work best in this case. Unlike in other cases (Frangioni and Gendron, 2013; Frangioni and Gorgone, 2014) quadratic stabilisation performed better than the linear one and it has been selected; among the applicable CPLEX algorithms, the *sifting* one turned out to be the better choice. For QR we have also adopted the “easy components” approach discussed above, while for FR we did not as it did not bring any significant performance improvement. The results of the bundle method applied to the Lagrangian dual of FR and QR are reported in Table 1 under the columns *BFR* and *BQR*, respectively. All codes have been ran with a time limit equal to 18000 seconds, no limit on the number of iterations and a required relative gap tolerance of $1e-6$.

Table 1 reports the comparison among all the three approaches. The first column indicates the name of the instance, after which we reported the dimensions in terms of number of nodes, arcs and commodities, respectively. For each method we have reported the total execution time at termination and the relative gap between the optimal bound attainable (the same for all approaches, as discussed in the previous sections) and that provided by the Lagrangian approach in the columns called *time* and *gap*, respectively. If the time column is not present, this means that the time limit has been reached for all instances; this happens for BKR-10 and BFR. Otherwise, time limit being reached is indicated by “—”.

The results show that BQR significantly outperforms all the other approaches for all the instances. BKR-4k at least manages to terminate in the smallest instances but not for medium and large ones; on the others provide between about four or five

Table 1 Comparison of the different approaches

#	dimension			BKR-10	BKR-4k		BFR	BQR	
	N	A	K	gap	time	gap	gap	time	gap
p33	20	226	40	5.7e-6	227.68	6.6e-7	4.6e-2	5.27	1.3e-07
p34	20	226	40	8.2e-6	233.14	3.5e-7	5.4e-2	5.36	3.3e-07
p35	20	226	40	7.3e-6	260.01	8.6e-7	8.9e-2	5.83	3.3e-09
p36	20	226	40	9.6e-6	57.02	8.5e-7	9.3e-2	4.59	3.8e-07
p37	20	230	200	5.1e-4	—	3.2e-4	9.2e-2	3954.59	1.4e-07
p38	20	230	200	4.8e-4	—	3.2e-4	5.7e-2	3724.92	2.6e-07
p39	20	230	200	4.5e-6	—	2.5e-5	4.5e-2	964.00	1.3e-09
p40	20	230	200	5.0e-6	—	1.5e-5	5.1e-2	838.73	4.7e-09
p41	20	292	40	3.2e-6	212.67	3.1e-8	4.9e-2	6.75	2.5e-08
p42	20	292	40	3.3e-6	130.07	2.6e-8	7.3e-2	6.66	2.8e-10
p43	20	292	40	9.9e-6	193.61	3.0e-8	9.0e-2	5.25	5.9e-10
p44	20	292	40	5.2e-6	134.04	1.3e-6	1.3e-1	6.56	2.3e-07
p45	20	292	200	6.4e-5	—	3.4e-5	2.7e-1	1375.14	7.7e-09
p46	20	292	200	5.5e-6	—	9.5e-5	9.9e-2	775.76	9.8e-07
p47	20	292	200	6.4e-6	—	7.0e-6	2.0e-2	552.91	7.8e-07
p48	20	292	200	6.2e-6	—	5.2e-5	8.4e-2	897.78	1.1e-08
p49	30	519	100	6.9e-6	—	6.4e-5	5.5e-2	261.61	2.8e-09
p50	30	519	100	1.1e-5	—	1.7e-4	7.0e-2	379.68	2.2e-09
p51	30	519	100	8.4e-6	—	2.5e-5	6.0e-2	122.46	3.2e-10
p52	30	519	100	9.8e-6	—	2.1e-5	9.4e-2	163.40	8.4e-09
p53	30	519	400	7.9e-6	—	7.1e-6	7.6e-2	11624.50	1.5e-10
p54	30	519	400	3.1e-5	—	5.3e-5	1.8e-1	14904.24	2.9e-09
p55	30	519	400	1.4e-5	—	5.6e-5	1.0e-1	7890.78	2.9e-10
p56	30	519	400	1.2e-4	—	1.8e-4	1.7e-1	15731.90	3.5e-07
p57	30	684	100	7.0e-6	—	2.9e-5	7.4e-2	155.03	1.5e-09
p58	30	684	100	6.1e-6	—	3.7e-5	7.6e-2	169.55	2.3e-08
p59	30	684	100	1.9e-5	—	8.2e-5	5.3e-2	164.25	3.2e-09
p60	30	684	100	7.2e-6	—	3.9e-5	9.7e-2	124.95	5.3e-07
p61	30	692	400	1.5e-5	—	1.2e-4	7.0e-2	11960.57	3.8e-09
p62	30	692	400	1.1e-5	—	1.4e-4	1.2e-1	16747.33	7.4e-07
p63	30	692	400	6.6e-6	—	9.5e-5	1.0e-1	9327.00	2.6e-07
p64	30	692	400	1.3e-6	—	2.6e-4	1.1e-1	8720.21	7.0e-08

significant digits of accuracy. By dint of being able to perform more iterations, due to the much leaner master problem, BKR-10 sometimes achieves better accuracy than BKR-4k in the same time, at times close to the required 6 significant digits; however, it never really terminates within the time limit. The naïve fully disaggregated approach, BFR, basically does not work at all: the much larger master problem size allows for performing far fewer iterations in the allotted time, and the much larger dual space makes it way more difficult to find a good dual solution with that effort. On the contrary, the new BQR approach always manages to find solutions with the required accuracy, albeit possibly at a significant computational cost for the largest instances (that are, however, notoriously hard to solve). Note that, by the results of (Frangioni et al., 2017), the alternative provided by subgradient-type methods would ever be able to achieve such accuracies, except possibly using smoothing techniques

(Frangioni et al., 2018) but at the cost of a disproportionate number of iterations. All in all the results confirm that the proposed modification of master problem can significantly improve performances of a Lagrangian-type approach.

5 Conclusions

While the Dantzig-Wolfe / Lagrangian approach to compute tight bounds for large structured combinatorial optimization problems is well-known, its by-the-book application hardly directly provides compelling results. One of the main ways to improve its performances is to tinker with the master problem to better reflect the structure of the underlying original model. This has been already done a number of times, with Multicommodity flow and Network Design problems being a prime application, with good success (Frangioni and Gendron, 2013; Frangioni and Gorgone, 2014); usually general stabilisation approaches (Frangioni, 2020) have also to be used at the same time, but this is in general possible and relatively easy to do (with the appropriate software). This work provides yet another step in this direction by developing a special modification for models where the crucial separability structure is not naturally present, but can be revealed by appropriate (master problem) reformulation techniques. Future work in this direction comprise the development of full-fledged approaches to the underlying problem, be them exact or heuristic, that exploit the Lagrangian bound and the corresponding primal information.

Acknowledgements

The first and third authors want to express their gratitude to the second author for the hugely positive role that he has had in their scientific and human lives, only a tiny bit of which has been initiating the research leading to this Chapter.

References

- Bacaud L, Lemaréchal C, Renaud A, Sagastizábal C (2001) Bundle Methods in Stochastic Optimal Power Management: A Disaggregated Approach Using Preconditioners. *Computational Optimization and Applications* 20:227–244
- Bacci T, Frangioni A, Gentile C, Tavlaridis-Gyparakis K (2023) New minlp formulations for the unit commitment problem with ramping constraints. *Operations Research* to appear
- Bahiense L, Maculan N, Sagastizábal C (2002) The volume algorithm revisited: relation with bundle methods. *Mathematical Programming* 94(1):41–70
- Balakrishnan A, Magnanti T (1989) A Dual-Ascent Procedure for Large-Scale Uncapacitated Network Design. *Operations Research* 37(5):716–740

- Borghetti A, Frangioni A, Lacalandra F, Nucci C (February 2003) Lagrangian Heuristics Based on Disaggregated Bundle Methods for Hydrothermal Unit Commitment. *IEEE Transactions on Power Systems* 18(1):313–323
- Cappanera P, Frangioni A (2003) Symmetric and asymmetric parallelization of a cost-decomposition algorithm for multi-commodity flow problems. *INFORMS Journal on Computing* 15(4):369–384
- Chouman M, Crainic T, Gendron B (2016) Commodity Representations and Cut-Set-Based Inequalities for Multicommodity Capacitated Fixed-Charge Network Design. *Transportation Science* [Online; posted 27-July-2016]
- Crainic T, Gendron B (2021) Network Design with Applications to Transportation and Logistics, Springer Cham, chap Exact Methods for Fixed-Charge Network Design, pp 29–88
- Crainic T, Frangioni A, Gendron B (1999) Multicommodity Capacitated Network Design. In: Soriano, P, Sanso, B (eds) *Telecommunications Network Planning*, Kluwer Academics Publisher, pp 1–19
- Crainic T, Frangioni A, Gendron B (2001) Bundle-based Relaxation Methods for Multicommodity Capacitated Fixed Charge Network Design Problems. *Discrete Applied Mathematics* 112:73–99
- Crainic T, Gendron B, Hernu G (2004) A Slope Scaling/Lagrangian Perturbation Heuristic with Long-Term Memory for Multicommodity Capacitated Fixed-Charge Network Design. *Journal of Heuristics* 10(5):525–545
- Frangioni A (1996) Solving semidefinite quadratic problems within nonsmooth optimization algorithms. *Computers & Operations Research* 21:1099–1118
- Frangioni A (2005) About Lagrangian methods in integer optimization. *Annals of Operations Research* 139:163–193
- Frangioni A (2020) *Numerical Nonsmooth Optimization: State of the Art Algorithms*, Springer, chap Standard Bundle Methods: Untrusted Models and Duality, pp 61–116
- Frangioni A, Gendron B (2013) A Stabilized Structured Dantzig-Wolfe Decomposition Method. *Mathematical Programming* 140:45–76
- Frangioni A, Gorgone E (2014) Generalized bundle methods for sum-functions with “easy” components: Applications to multicommodity network design. *Mathematical Programming* 145(1):133–161
- Frangioni A, Gorgone E, Gendron B (2017) On the computational efficiency of subgradient methods: A case study in combinatorial optimization. *Mathematical Programming Computation* 9:573–604
- Frangioni A, Gendron B, Gorgone E (2018) Dynamic smoothness parameter for fast gradient methods. *Optimization Letters* 12(1):43–53
- Frangioni A, Gendron B, Gorgone E (2020) Quasi-separable dantzig-wolfe reformulations for network design. In: Baiou M, Gendron B, Gunluk O, Mahjoub A (eds) *Proceedings of the International Symposium on Combinatorial Optimization ISCO 2020*, Lecture Notes in Computer Science, vol 12176, Springer-Verlag, pp 227—236
- Gendron B (2019) Revisiting Lagrangian relaxation for network design. *Discrete Applied Mathematics* 261:203–218
- Gendron B, Larose M (2014) Branch-and-price-and-cut for large-scale multicommodity capacitated fixed-charge network design. *EURO Journal on Computational Optimization* 2:55–75
- Guignard M, Kim S (1987) Lagrangian Decomposition: a Model Yelding Stronger Lagrangian Bounds. *Mathematical Programming* 39:215–228
- Holmberg K, Yuan D (2000) A Lagrangian Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem. *Operations Research* 48(3):461–481
- Jones K, Lustig I, Farwolden J, Powell W (1993) Multicommodity Network Flows: The Impact of Formulation on Decomposition. *Mathematical Programming* 62:95–117
- Scuzziato M, Finardi E, Frangioni A (2018) Comparing spatial and scenario decomposition for stochastic hydrothermal Unit Commitment problems. *IEEE Transactions on Sustainable Energy* 9(3):1307–1317