

# Solving the Multi-objective Integer Minimum Cost Flow Problem via Circulations

Antonio Sedeño-Noda<sup>a</sup>

<sup>a</sup>*Departamento de Matemáticas, Estadística e Investigación Operativa. Universidad de La Laguna, 38200 San Cristóbal de La Laguna, Santa Cruz de Tenerife, España.*

---

## Abstract

The Multi-objective Integer Minimum Cost Flow Problem (MOIMCF) asks for the efficient integer flows of a capacitated network whose arcs carry vectors of conflicting costs. We present an exact algorithm that computes a minimal complete set of efficient flows, based on a characterization of efficiency by circulations in the residual network. The feasible set is partitioned by integer bound decisions on single arcs, and each partition is represented by its lexicographically minimum flows for the  $q$  cyclic orders of the objectives, which give its ideal point and are obtained for each child by lexicographic shortest augmenting paths. The main result shows that a partition can have a new nondominated vector in a box of the search region only if the residual network of any of these representatives contains a lexicographically negative cycle for weights determined by the box and by the representative. One negative-cycle detection per box therefore either returns a circuit directed at the box, which is used to branch, or certifies that the partition and all its subpartitions have no vector in the box. Computational experiments with three to five objectives compare the method with two general multi-objective integer programming algorithms. The method is faster than both on most instances, with an advantage that is larger with four and five objectives; with three objectives, the defining point algorithm is faster on dense networks with large supply.

*Keywords:* Multi-objective Integer Minimum Cost Flow; Circulations; Network Optimization.

---

## 1. Introduction

The Minimum Cost Flow Problem (MCF) is a central model of network optimization; its constraint matrix is totally unimodular, and it is solved efficiently by network simplex, cost-scaling and successive shortest path methods [1]. In the Multi-objective Integer Minimum Cost Flow Problem (MOIMCF), every arc carries a cost vector of dimension  $q > 1$  and the goal is to identify

---

*Email address:* asedeno@ull.edu.es (Antonio Sedeño-Noda)

7 the efficient integer flows. The problem is harder: even the number of extreme  
8 supported nondominated points may be exponential in the biobjective case [30],  
9 and non-supported efficient flows must also be found.

10 The literature, reviewed by Hamacher et al. [12], is almost exclusively de-  
11 voted to the biobjective case. The continuous problem was solved by parametric  
12 network simplex [33], and exact methods for the integer problem include the  
13 two-phase method of Raith and Ehrgott [28] and the  $\varepsilon$ -constraint branch-and-  
14 bound method of Eusébio and Figueira [7]; earlier adjacency-based methods  
15 [25, 34] are incomplete, because the efficient set is not connected under basis  
16 adjacency. Supported extreme flows can be computed by enhanced parametric  
17 programming [29], and all supported efficient flows by the output-polynomial  
18 method of Könen and Stiglmayr [18, 19]. With two objectives the nondom-  
19 inated vectors are totally ordered by the first objective, and these methods  
20 exploit this order; our method applies to any  $q \geq 2$ , but we focus on  $q \geq 3$ ,  
21 where network-specific methods compute either the supported efficient flows [18]  
22 or representative subsets of efficient flows [36, 37], and, to the best of our knowl-  
23 edge, no network-specific method computing a minimal complete set, including  
24 the non-supported efficient flows, has been published; exact enumeration relies  
25 on general multi-objective integer programming methods [14, 27, 9, 5], which do  
26 not exploit the network structure. Könen, Löhken and Stiglmayr [21] used one  
27 of them [5] to compute the nondominated sets of a large collection of MOIMCF  
28 instances, and observed that this becomes computationally intensive even for  
29 moderately sized instances.

30 Our method combines two ingredients. The first is the partition-based rank-  
31 ing principle of Lawler [24], applied to  $K$ -best network flows by Hamacher [11],  
32 improved by Sedeño-Noda and Espino-Martín [32] and used in the second phase  
33 of the two-phase method [28]; Könen et al. [20] used it to enumerate all opti-  
34 mal integer flows. The second is the characterization of efficient solutions by  
35 residual circulations, introduced for the Multi-objective Assignment Problem  
36 (MAP) in [31] and extended here to capacitated networks. Starting from the  
37  $q$  lexicographic roots of the feasible set, one for each cyclic order of the ob-  
38 jectives, without a supported-solution phase, the feasible set is split by bound  
39 decisions on single arcs, and only residual circuits that may lead to new non-  
40 dominated vectors open branches. The basic step is local: given a partition  
41 and its lexicographic roots, decide whether the partition may still contain a  
42 nondominated vector that has not been found. Restricting the characteriza-  
43 tion to the boxes of the search region [15] turns this question into at most one  
44 negative-cycle detection per box: a detection that finds a cycle returns a circuit  
45 directed at the box, which is used to branch, and one that finds none certifies  
46 that the partition and all partitions derived from it have no vector in the box.  
47 As in multi-objective branch-and-bound [35, 9], exact scalar and weighted-sum  
48 bounds discard partitions and boxes, and partitions with a dominated represen-  
49 tative are still expanded, since they may contain efficient flows; efficient flows  
50 need not be connected [10].

51 The contributions of the paper are the following. First, efficient flows are  
52 characterized by residual circulations, written as a lexicographic  $\varepsilon$ -constraint

53 problem in the residual network (Theorems 2 and 3), which generalizes the Negative  
 54 Cycle Optimality Theorem. Second, the lexicographic root and valid vector  
 55 potentials of every child are obtained by one restricted lexicographic shortest-  
 56 path computation (Proposition 2); every partition keeps one lexicographic root  
 57 for each cyclic order of the objectives, and these  $q$  roots give its ideal point with-  
 58 out further computation. Third, the box-restricted relevance test (Theorem 4,  
 59 Proposition 5 and Corollary 2) either certifies, with one negative-cycle detection  
 60 from the root closest to the box, that a box of the search region contains no  
 61 vector of the partition, a certificate inherited by all subpartitions, or returns a  
 62 circuit directed at the box, which is used to branch; in addition, a bound on the  
 63 reduced costs of the representatives removes arcs from each detection and fixes  
 64 arcs for all subpartitions (Proposition 6). Finally, the method is implemented  
 65 and compared with two general exact methods, the defining point algorithm of  
 66 Dächert, Fleuren and Klamroth [5] and the method of Tamby and Vanderpooten  
 67 [38], on instances with three to five objectives. It is faster than both on most of  
 68 the instances; the exceptions are discussed in Section 10.3.

69 Sections 2–4 recall the residual network, state the characterization and in-  
 70 troduce the lexicographic root and the bound partition. Sections 5–7 develop  
 71 the circuit generator, the algorithm and the box-restricted test. Section 8 proves  
 72 correctness, Section 9 describes the implementation, Section 10 reports the ex-  
 73 periments and Section 11 concludes.

## 74 2. The single-objective Minimum Cost Flow Problem

75 Let  $G = (V, A)$  be a directed graph with  $n = |V|$  nodes and  $m = |A|$   
 76 arcs, without parallel arcs. Every node  $i \in V$  has an integer balance  $b_i$ , with  
 77  $\sum_{i \in V} b_i = 0$ , and every arc  $e = (i, j) \in A$  has integer lower and upper bounds  
 78  $l_e \leq u_e$  and an integer cost  $c_e$ ; rational costs can be scaled to integers. For a  
 79 node  $v \in V$ ,  $\delta^+(v)$  and  $\delta^-(v)$  denote the sets of arcs leaving and entering  $v$ .  
 80 The Integer Minimum Cost Flow Problem can be formulated as

$$\min \sum_{e \in A} c_e x_e, \quad (1)$$

$$\text{s.t.} \quad \sum_{e \in \delta^+(i)} x_e - \sum_{e \in \delta^-(i)} x_e = b_i, \quad i \in V, \quad (2)$$

$$l_e \leq x_e \leq u_e, \quad e \in A, \quad (3)$$

$$x_e \in \mathbb{Z}, \quad e \in A. \quad (4)$$

81 Let  $X$  denote the set of integer feasible flows. We allow the bounds to vary  
 82 during the algorithm; for integer vectors  $l \leq u$  we write  $X(l, u)$  for the set of  
 83 integer flows satisfying (2) and  $l \leq x \leq u$ .

84 For  $x \in X(l, u)$ , the residual network is  $G(x) = G_{l,u}(x) = (V, A(x))$ . Every  
 85 arc  $e = (i, j)$  induces at most two residual arcs: the forward arc  $e^+ = (i, j)$  with  
 86 residual capacity  $r_{e^+}(x) = u_e - x_e$  and cost  $\gamma_{e^+} = c_e$ , present when  $x_e < u_e$ ,  
 87 and the backward arc  $e^- = (j, i)$  with residual capacity  $r_{e^-}(x) = x_e - l_e$  and

cost  $\gamma_{e^-} = -c_e$ , present when  $x_e > l_e$ . For a residual arc  $a$  we denote by  $e(a)$  its original arc and by  $\bar{a}$  its twin, that is, the other residual arc of  $e(a)$ .

A circulation in  $G(x)$  is a nonnegative integer vector  $z$  on  $A(x)$  with zero net flow at every node and  $z_a \leq r_a(x)$  for every residual arc  $a$ .

**Definition 1** (Circulation augmentation). *Let  $z$  be a circulation in  $G(x)$ . The flow obtained by sending  $z$  through  $G(x)$  is denoted by  $y = x \oplus z$ , where*

$$y_e = x_e + z_{e^+} - z_{e^-}, \quad e \in A,$$

with  $z_{e^+} = 0$  or  $z_{e^-} = 0$  when the corresponding residual arc is absent. For a directed simple circuit  $w$  of  $G(x)$  we write  $x \oplus w = x \oplus \chi(w)$ , where  $\chi(w) \in \{0, 1\}^{A(x)}$  is the incidence vector of  $w$ , that is,  $\chi_a(w) = 1$  if  $a \in w$  and  $\chi_a(w) = 0$  otherwise.

**Lemma 1** (Residual feasibility). *If  $z$  is a circulation in  $G(x)$ , then  $x \oplus z \in X(l, u)$  and*

$$c(x \oplus z) = c(x) + \sum_{a \in A(x)} \gamma_a z_a. \quad (5)$$

*Proof.* Flow conservation of  $z$  preserves (2). For every arc  $e$ ,  $0 \leq z_{e^+} \leq u_e - x_e$  and  $0 \leq z_{e^-} \leq x_e - l_e$  give  $l_e \leq x_e + z_{e^+} - z_{e^-} \leq u_e$ . The cost identity follows from  $\gamma_{e^-} = -c_e$ .  $\square$

Lemma 1 does not require  $z$  to use at most one of the two residual arcs of an arc  $e$ . A circulation  $z$  in  $G(x)$  is called *conformal* if  $z_{e^+} z_{e^-} = 0$  for every  $e \in A$ .

**Theorem 1** (Conformal decomposition [1, Section 3.5]). *Let  $x, y \in X(l, u)$ . Then there exist directed simple circuits  $w_1, \dots, w_k$  of  $G(x)$ , with  $k \leq m$ , and positive integers  $\lambda_1, \dots, \lambda_k$  such that  $z = \sum_{t=1}^k \lambda_t \chi(w_t)$  is a conformal circulation in  $G(x)$ ,*

$$y = x \oplus z, \quad c(y) = c(x) + \sum_{t=1}^k \lambda_t \gamma(w_t).$$

**Lemma 2** (Box lemma). *In the setting of Theorem 1, for every integer vector  $0 \leq \mu \leq \lambda$ , the flow  $x \oplus \sum_t \mu_t \chi(w_t)$  belongs to  $X(l, u)$  and lies componentwise between  $x$  and  $y$ .*

*Proof.* By conformality, on every arc  $e$  all circuits move the flow in the same direction, and the total movement is at most  $|y_e - x_e|$ .  $\square$

The classical Negative Cycle Optimality Theorem [1] states that  $x^* \in X(l, u)$  is optimal if and only if  $G(x^*)$  contains no negative-cost directed cycle. Given node potentials  $d_i \in \mathbb{R}$ , the reduced cost of a residual arc  $a = (i, j)$  is

$$\bar{\gamma}_a = \gamma_a + d_i - d_j, \quad (6)$$

and directed-cycle costs are invariant under this transformation.

### 119 3. A circulation characterization of efficient flows

120 In the MOIMCF, every arc  $e$  has an integer cost vector  $c_e = (c_e^1, \dots, c_e^q) \in \mathbb{Z}^q$ ,  
 121 and the problem is

$$\min \quad c(x) = (c^1(x), \dots, c^q(x)), \quad (7)$$

$$\text{s.t.} \quad x \in X, \quad (8)$$

122 where  $c^r(x) = \sum_{e \in A} c_e^r x_e$  for  $r = 1, \dots, q$ . The residual arcs carry the vector  
 123 costs  $\gamma_{e^+} = c_e$  and  $\gamma_{e^-} = -c_e$ , and for a circulation  $z$  in  $G(x)$  we write  $\gamma(z) =$   
 124  $\sum_{a \in A(x)} \gamma_a z_a$ , so that  $c(x \oplus z) = c(x) + \gamma(z)$ ; in particular,  $\gamma(w)$  denotes the  
 125 cost of a residual circuit  $w$ . Throughout,  $c$  refers to arcs and flows of  $G$  and  $\gamma$   
 126 to arcs, circuits and circulations of a residual network.

127 For  $u, v \in \mathbb{R}^q$ ,  $u$  is said to dominate  $v$ , and we write  $u \prec_D v$ , if  $u_r \leq v_r$   
 128 for all  $r = 1, \dots, q$  and  $u \neq v$ . A feasible flow  $x$  is efficient if no  $y \in X$   
 129 satisfies  $c(y) \prec_D c(x)$ . Its objective vector  $c(x)$  is then called a nondominated  
 130 objective vector, and  $Y_N$  denotes the set of nondominated objective vectors.  
 131 Two flows are equivalent if they have the same objective vector. In this paper,  
 132 the MOIMCF consists of computing a *minimal complete set of efficient flows*  
 133 containing exactly one efficient flow for each nondominated objective vector.

134 The lexicographic relation  $\prec_{lex}$  on  $\mathbb{R}^q$  is defined as follows. We write  $u \prec_{lex} v$   
 135 if  $u_r < v_r$  for the first index  $r$  for which  $u_r \neq v_r$ . Throughout the paper, the  
 136 instance is assumed to be feasible, that is,  $X \neq \emptyset$ .

137 **Theorem 2** (Characterization of efficient flows). *A feasible flow  $x \in X$  is*  
 138 *efficient if and only if  $G(x)$  does not contain a circulation  $z$  such that*

$$\gamma(z) \prec_D 0. \quad (9)$$

139 *Equivalently,  $x$  is efficient if and only if there are no directed simple circuits*  
 140  *$w_1, \dots, w_k$  of  $G(x)$  and positive integers  $\lambda_1, \dots, \lambda_k$  with  $\sum_t \lambda_t \chi(w_t) \leq r(x)$  and*  
 141  *$\sum_t \lambda_t \gamma(w_t) \prec_D 0$ .*

142 *Proof.* If  $z$  satisfies (9), then  $x \oplus z$  is feasible by Lemma 1 and dominates  $x$ .  
 143 Conversely, if  $y \in X$  dominates  $x$ , Theorem 1 gives a conformal circulation  $z$  in  
 144  $G(x)$  with  $y = x \oplus z$  and cost  $c(y) - c(x) \prec_D 0$ . The circuit form follows from  
 145 the flow decomposition theorem and from Lemma 1.  $\square$

146 Theorem 2 is the multiobjective counterpart of the Negative Cycle Optimal-  
 147 ity Theorem, with a dominating circulation in the role of a negative cycle. For  
 148 the MAP, all residual capacities equal one and a bounded circulation is a family  
 149 of node-disjoint circuits, so Theorem 2 reduces to the characterization of [31].  
 150 In general, circuits may appear with multiplicities and may share nodes and  
 151 arcs, provided that the residual capacities are respected.

152 *3.1. A lexicographic circulation formulation*

153 For every residual arc  $a \in A(x)$ , let  $z_a$  be an integer variable, and define

$$\gamma(z) = (\gamma^1(z), \dots, \gamma^q(z)), \quad \gamma^r(z) := \sum_{a \in A(x)} \gamma_a^r z_a.$$

154 Consider

$$\text{lexmin } \gamma(z) = (\gamma^1(z), \dots, \gamma^q(z)), \quad (10)$$

$$\text{s.t. } \sum_{a \in \delta_x^+(i)} z_a - \sum_{a \in \delta_x^-(i)} z_a = 0, \quad i \in V, \quad (11)$$

$$\gamma^r(z) \leq 0, \quad r = 2, \dots, q, \quad (12)$$

$$0 \leq z_a \leq r_a(x), \quad z_a \in \mathbb{Z}, \quad a \in A(x), \quad (13)$$

155 where  $\delta_x^\pm(i)$  are taken in  $G(x)$ . The zero circulation is always feasible, so the lex-  
156 icographically optimal objective vector cannot be lexicographically larger than  
157 zero, and the constraint  $\gamma^1(z) \leq 0$  is unnecessary.

158 Formulation (10)–(13) is not solved as an integer program, and the side  
159 constraints (12) prevent treating it as an ordinary minimum cost flow problem;  
160 it is an optimization form of Theorem 2.

161 **Theorem 3** (Lexicographic circulation characterization of efficiency). *Let  $z^*$*   
162 *be a lexicographically optimal solution of (10)–(13). Then*

$$x \text{ is efficient} \iff \gamma(z^*) = 0. \quad (14)$$

163 *If  $\gamma(z^*) \neq 0$ , then  $\gamma(z^*) \prec_D 0$  and  $x \oplus z^*$  dominates  $x$ .*

164 *Proof.* Suppose first that  $x$  is inefficient. By Theorem 2, there exists a circu-  
165 lation  $z$  in  $G(x)$  with  $\gamma(z) \prec_D 0$ . It satisfies (11)–(13) and  $\gamma(z) \prec_{lex} 0$ , hence  
166  $\gamma(z^*) \prec_{lex} 0$  and  $\gamma(z^*) \neq 0$ . Conversely, assume  $\gamma(z^*) \neq 0$ . Since  $z = 0$  is  
167 feasible, lexicographic optimality implies  $\gamma^1(z^*) \leq 0$ , and constraints (12) give  
168  $\gamma^r(z^*) \leq 0$  for  $r \geq 2$ . Thus  $\gamma(z^*) \prec_D 0$ , and  $x \oplus z^*$  dominates  $x$  by Lemma 1.  $\square$

169 This formulation is the  $\varepsilon$ -constraint characterization of efficiency [3] with  
170  $\varepsilon = c(x)$ , written in the residual network: with  $y = x \oplus z$ , the constraints  
171  $\gamma^r(z) \leq 0$  are  $c^r(y) \leq c^r(x)$  and the objective is  $c(y) - c(x)$ . In this form  
172 the reference point disappears from the model, the zero circulation is always  
173 feasible, and efficiency becomes the absence of a dominating circulation. The  
174 objective-space methods used for comparison in Section 10 solve  $\varepsilon$ -constraint  
175 problems of this kind as integer programs; the method developed below works  
176 instead with the circuits of the residual network.

177 **4. Lexicographic roots and bound partitions**

178 For  $s \in \{1, \dots, q\}$  let

$$c_{\langle s \rangle}(x) = (c^s(x), c^{s+1}(x), \dots, c^q(x), c^1(x), \dots, c^{s-1}(x))$$

179 be the cost vector with its components rotated so that objective  $s$  comes first,  
 180 and, for vectors  $y, z \in \mathbb{Z}^q$ , write  $y \prec_s z$  when  $y_{\langle s \rangle} \prec_{lex} z_{\langle s \rangle}$ , where  $y_{\langle s \rangle}$  is the  
 181 same rotation of  $y$ ;  $\preceq_s$  and  $\succeq_s$  are defined accordingly. Thus  $\prec_1$  is  $\prec_{lex}$ , and  
 182  $\prec_s$  compares objective  $s$  first, then  $s+1, \dots, q$ , and then  $1, \dots, s-1$ . We fix  
 183 an index  $s$  and select one flow

$$x^s \in \arg \min_{\prec_s} \{c(x) : x \in X\}, \quad (15)$$

184 which we call the lexicographic root for  $\prec_s$ ; in particular  $c^s(x^s) = \min\{c^s(x) :$   
 185  $x \in X\}$ . The root can be computed by any minimum cost flow method that  
 186 compares the rotated costs lexicographically, or with the scalar weighting of  
 187 Isermann [13]; our implementation uses successive shortest paths with lexico-  
 188 graphic costs (Section 9). Until Section 7,  $s$  is fixed; Section 7 uses the roots of  
 189 all the  $q$  orders.

#### 190 4.1. Lexicographic potentials and reduced vector costs

191 A directed circuit  $w$  of  $G(x^s)$  with  $\gamma(w) \prec_s 0$  would yield, by Lemma 1,  
 192 a feasible flow with a cost vector smaller than that of  $x^s$  for  $\prec_s$ . Hence no  
 193 such residual circuit exists. Augment  $G(x^s)$  with a super-source  $s_0$  and zero  
 194 vector-cost arcs  $(s_0, v)$  for every  $v \in V$ . Since no directed cycle contains  $s_0$  and  
 195 no residual cycle is negative for  $\prec_s$ , shortest-path distances  $d_v = d_v^s \in \mathbb{Z}^q$  for  
 196  $\prec_s$  from  $s_0$  are well defined; they may be computed by a lexicographic label-  
 197 correcting method. Any other valid potentials, such as those returned by the  
 198 successive shortest path method together with  $x^s$ , may be used instead, since  
 199 only (17) below is needed. For every residual arc  $a = (i, j) \in A(x^s)$  define

$$\bar{\gamma}_a = \gamma_a + d_i - d_j = (\bar{\gamma}_a^1, \dots, \bar{\gamma}_a^q). \quad (16)$$

200 Shortest-path optimality gives

$$\bar{\gamma}_a \succeq_s 0, \quad a \in A(x^s). \quad (17)$$

201 Hence  $\bar{\gamma}_a^s \geq 0$  for every residual arc and, if  $\bar{\gamma}_a \neq 0$ , the first nonzero component  
 202 of  $\bar{\gamma}_a$  in the order  $s, s+1, \dots, q, 1, \dots, s-1$  is strictly positive.

#### 203 4.2. Branching states and bound partitions

204 The partition scheme follows the binary principle of Lawler [24], used for  
 205 integer network flows by Hamacher [11] and Sedeño-Noda and Espino-Martín  
 206 [32].

207 **Definition 2** (Branching state). *A branching state is denoted by*

$$P = (l^P, u^P, x, d),$$

208 where  $l \leq l^P \leq u^P \leq u$  are integer bound vectors,  $X(P) = X(l^P, u^P)$ ,  $x$  is a  
 209 minimum flow of  $X(P)$  for  $\prec_s$ , called the representative, or lexicographic root,  
 210 of  $P$ , and  $d$  are node potentials for which every residual arc of  $G_{l^P, u^P}(x)$  has a

211 reduced vector cost that is nonnegative for  $\prec_s$ . Branching states are stored in  
 212 a global array  $\mathcal{P}$  and are referred to by an integer index  $p$ . We write  $G_P(x) =$   
 213  $G_{l^P, u^P}(x)$  for the residual network of a flow  $x$  of  $X(P)$  with the bounds of  $P$ ;  
 214 unless stated otherwise, the residual arcs and circuits of a state are those of  
 215  $G_P(x)$ .

216 Let  $x$  be the representative of  $P$ , and let  $a$  be an arc of  $G_P(x)$  with original  
 217 arc  $e = e(a)$ , called the *opening arc*. We write  $P\langle a \rangle$  for the child restriction  
 218 induced by traversing  $a$  and  $P\langle \bar{a} \rangle$  for the complementary restriction:

$$\begin{aligned} X(P\langle a \rangle) &= \begin{cases} \{y \in X(P) : y_e \geq x_e + 1\}, & a = e^+, \\ \{y \in X(P) : y_e \leq x_e - 1\}, & a = e^-, \end{cases} \\ X(P\langle \bar{a} \rangle) &= \begin{cases} \{y \in X(P) : y_e \leq x_e\}, & a = e^+, \\ \{y \in X(P) : y_e \geq x_e\}, & a = e^-, \end{cases} \end{aligned}$$

219 where  $x_e$  is the value of the representative. Antiparallel original arcs are allowed;  
 220 residual arcs are identified by their original arc and direction. Both restrictions  
 221 are again of the form  $X(l', u')$ , obtained by changing one bound of  $e$ . Thus  
 222  $x \in X(P\langle \bar{a} \rangle)$ , and  $X(P)$  is the disjoint union of  $X(P\langle a \rangle)$  and  $X(P\langle \bar{a} \rangle)$ .

223 Suppose that a state successively returns opening arcs  $a_1, \dots, a_h$ , each with  
 224 respect to the representative of the current restriction. Child  $i$  is obtained after  
 225 imposing the complementary decisions for  $a_1, \dots, a_{i-1}$  and the child decision  
 226 for  $a_i$ ; the final residual partition imposes the complementary decision for every  
 227  $a_1, \dots, a_h$ . The same arc  $e$  may be selected twice, once in each direction, in  
 228 which case its flow becomes fixed.

229 **Proposition 1** (Bound partition). *The children generated in this way, together*  
 230 *with the final residual partition, are pairwise disjoint and their union is  $X(P)$ .*  
 231 *Every restriction strictly decreases  $\sum_{e \in A} (u_e^P - l_e^P)$ .*

232 *Proof.* A feasible flow  $y \in X(P)$  belongs to the child of the first index  $i$  for which  
 233  $y$  violates the complementary decision of  $a_i$ , and to the final residual partition  
 234 if no such index exists. The preceding decisions are satisfied by  $y$ , while the  
 235 child decision of  $a_i$  is exactly the negation of its complementary decision within  
 236  $X(P)$ . Hence every flow belongs to exactly one region. Each decision moves  
 237 one bound of  $e(a_i)$  strictly towards the other one, without crossing it, because  
 238  $a_i$  is an arc of the residual network of the current restriction.  $\square$

239 The following property restores the lexicographic invariant after a branch is  
 240 opened.

241 **Proposition 2** (Lexicographic root of a child). *Let  $x$  be the representative of*  
 242  *$P$ , with reduced costs  $\bar{\gamma} \succeq_s 0$  on  $G_P(x)$ , and let  $a = (g, h)$  be an arc of a directed*  
 243 *simple circuit  $w$  of  $G_P(x)$  with  $\bar{a} \notin w$ . Let  $G^a$  be  $G_P(x)$  without the arcs  $a$  and*  
 244  *$\bar{a}$ , let  $D_i$  be the shortest-path distance for  $\prec_s$  from  $h$  to  $i$  in  $G^a$  with respect to*  
 245  *$\bar{\gamma}$ , and let  $P_{hg}^*$  be a shortest simple  $h$ - $g$  path for  $\prec_s$  in  $G^a$ , which exists because*

246  $w \setminus \{a\}$  is a  $h$ - $g$  path of  $G^a$ . Then  $x \oplus w \in X(P\langle a \rangle)$ , so the child is nonempty,  
 247 and a minimum flow of  $P\langle a \rangle$  for  $\prec_s$  is

$$x^a = x \oplus w_a^*, \quad w_a^* = a \cup P_{hg}^*. \quad (18)$$

248 Moreover, with  $\Delta = D_g$  and

$$\delta_i = \min_{\prec_s} \{D_i, \Delta\}, \quad d_i^a = d_i + \delta_i, \quad (19)$$

249 where  $D_i = +\infty$  for an unreachable node and  $\min_{\prec_s} \{+\infty, \Delta\} = \Delta$ , every arc of  
 250 the residual network of  $x^a$  with the bounds of  $P\langle a \rangle$  has reduced cost nonnegative  
 251 for  $\prec_s$  under  $d^a$ .

252 *Proof.* Let  $y \in X(P\langle a \rangle)$ . The child decision implies  $y_e > x_e$  if  $a = e^+$  and  
 253  $y_e < x_e$  if  $a = e^-$ , so the conformal decomposition  $y = x \oplus \sum_t \lambda_t \chi(w_t)$  of  
 254 Theorem 1 in  $G_{IP, u^P}(x)$  contains a circuit  $w_1$  using  $a$ . By conformality, no  
 255 circuit uses  $\bar{a}$ , so  $w_1 \setminus \{a\}$  is a  $h$ - $g$  path of  $G^a$ .

256 By Lemma 2,  $x \oplus w_1$  lies between  $x$  and  $y$  and therefore in  $X(P)$ ; since it  
 257 moves the flow on  $e$  by one unit in the direction of  $a$ , it belongs to  $X(P\langle a \rangle)$ .  
 258 Every  $x \oplus w_t$  belongs to  $X(P)$  by Lemma 2, so the minimality of  $x$  for  $\prec_s$  gives  
 259  $\gamma(w_t) \succeq_s 0$  for all  $t$ . Consequently

$$c(y) = c(x) + \gamma(w_1) + (\lambda_1 - 1)\gamma(w_1) + \sum_{t \geq 2} \lambda_t \gamma(w_t) \succeq_s c(x) + \gamma(w_1).$$

260 Since costs and reduced costs coincide on circuits,  $\gamma(w_1) = \bar{\gamma}_a + \bar{\gamma}(w_1 \setminus \{a\}) \succeq_s$   
 261  $\bar{\gamma}_a + \bar{\gamma}(P_{hg}^*) = \gamma(w_a^*)$ . The path  $P_{hg}^*$  is simple, uses neither  $a$  nor  $\bar{a}$ , and every  
 262 arc of  $G^a$  has residual capacity at least one; therefore  $w_a^*$  is a residual circuit,  
 263  $x^a = x \oplus w_a^* \in X(P\langle a \rangle)$  by Lemma 1, and  $c(x^a) \preceq_s c(y)$ .

264 For an unchanged arc  $b = (i, j)$  of  $G^a$ , shortest-path optimality and the  
 265 truncation in (19) give  $\delta_j \preceq_s \delta_i + \bar{\gamma}_b$ , and therefore the reduced cost of  $b$  under  
 266  $d^a$ ,  $\bar{\gamma}_b + \delta_i - \delta_j$ , is nonnegative for  $\prec_s$ . Every arc of  $P_{hg}^*$  is tight, so its residual  
 267 copy (if capacity remains) and its reverse arc have zero reduced cost under  $d^a$ .  
 268 The arc  $a$  keeps residual capacity only if  $r_a(x) \geq 2$ ; since  $\delta_h = 0$  and  $\delta_g = \Delta$ , its  
 269 new reduced cost is  $\bar{\gamma}_a + \Delta = \gamma(w_a^*) \succeq_s 0$ . Finally, the twin  $\bar{a}$  is not a residual  
 270 arc of the child: if  $a = e^+$ , the child lower bound is  $x_e + 1 = x_e^a$ , so  $r_{e^-}(x^a) = 0$ ;  
 271 the case  $a = e^-$  is symmetric. Thus all residual arcs of the child have reduced  
 272 cost nonnegative for  $\prec_s$ .  $\square$

273 Proposition 2 generalizes to the lexicographic order two scalar tools: the  
 274 computation of the best flow of a restricted partition in  $K$ -best network flow  
 275 algorithms [11, 32], and the update of potentials in the successive shortest path  
 276 algorithm [1]. The argument for the potentials is the classical one, transferred  
 277 to the lexicographic order: the augmentation along  $w_a^*$  only creates the reverses  
 278 of the arcs of  $P_{hg}^*$ , and a lexicographically negative circuit through them would  
 279 give a  $h$ - $g$  path lexicographically shorter than  $P_{hg}^*$ .

280 The complementary partition  $P\langle \bar{a} \rangle$  keeps  $x$  as its lexicographic root for  $\prec_s$   
 281 and retains the same potentials, because it only removes residual capacity from  
 282  $a$ .

## 283 5. Detecting a relevant circuit

284 For a state  $P$  with representative  $x$ , the local step only has to decide whether  
 285 a *relevant* circuit of  $G_P(x)$  remains and, if so, return an arc of one such circuit.  
 286 A circuit  $w$  is relevant when

$$\exists r \in \{1, \dots, q\} \setminus \{s\} : \gamma^r(w) < 0. \quad (20)$$

287 Section 7 replaces this test by a stronger one. Since the representative  $x$  is  
 288 minimum in  $X(P)$  for  $\prec_s$ , every circuit  $w$  of  $G_P(x)$  satisfies  $\gamma(w) \succeq_s 0$ , and  
 289 in particular  $\gamma^s(w) \geq 0$ ; a non-relevant circuit is therefore componentwise non-  
 290 negative. All the objectives other than  $s$  must be tested, including those that  
 291 precede  $s$  in the cyclic order: a circuit with  $\gamma^s(w) > 0$  and  $\gamma^r(w) < 0$  for some  
 292  $r < s$  is positive for  $\prec_s$ , but it may lead to a flow that is not dominated by  $x$ .

### 293 5.1. The required generator property

294 The correctness proof of the global algorithm uses only the following property  
 295 of the local step.

296 **(N)** If  $G_P(x)$  contains a relevant circuit, the generator returns an  
 297 arc  $a$  of a relevant circuit of  $G_P(x)$ . If no relevant circuit exists, it  
 298 returns NULL.

299 Any arc of a relevant circuit yields a valid partition by Proposition 1; which one  
 300 is used is a heuristic choice.

### 301 5.2. Relevance as negative-cycle detection

302 **Proposition 3** (Detection of relevant circuits). *A relevant circuit exists in*  
 303  *$G_P(x)$  if and only if, for some  $r \neq s$ , the residual network  $G_P(x)$  with arc*  
 304 *costs  $\gamma^r$  contains a negative cycle. Running the Bellman–Ford–Moore algorithm*  
 305 *[2, 8, 26] once for every such  $r$  therefore satisfies property (N) in  $O((q-1)nm)$*   
 306 *time.*

307 *Proof.* A relevant circuit  $w$  with  $\gamma^r(w) < 0$  is a negative cycle for  $\gamma^r$ . Conversely,  
 308 the Bellman–Ford–Moore algorithm returns a negative cycle as a directed simple  
 309 circuit of  $G_P(x)$ ; every arc of  $G_P(x)$  has residual capacity at least one, so this  
 310 circuit is a residual circuit of  $P$ , and it is relevant because its cost in objective  
 311  $r$  is negative. Reduced and original costs coincide on circuits, so either may be  
 312 used.  $\square$

313 Algorithm 1 states the resulting generator. Any arc of the returned cycle  
 314 may be used as opening arc, since the correctness proof does not depend on this  
 315 choice; the rule used in the implementation is described in Section 9.

---

**Algorithm 1: RELEVANTCIRCUIT( $P$ )**

---

**Input:** A branching state  $P$  with representative  $x$ .

**Output:** An opening arc  $a$  of a relevant circuit of  $G_P(x)$ ; or NULL.

316 **1 for**  $r = s + 1, \dots, q, 1, \dots, s - 1$  **do**  
317     **2**     Run the Bellman–Ford–Moore algorithm on  $G_P(x)$  with arc costs  $\gamma^r$ ;  
318     **3**     **if** a negative cycle  $w$  is found **then**  
319         **4**         **return** an arc of  $w$ ;  
320     **5 return** NULL;

---

## 6. The circulation-ranking algorithm

For a state  $P$  and  $r = 1, \dots, q$ , let

$$m_r(P) = \min\{c^r(\hat{x}) : \hat{x} \in X(P)\}. \quad (21)$$

Each of these problems is a single-objective integer minimum cost flow problem with bounds  $l^P, u^P$  and is polynomially solvable. The vector  $m(P) = (m_1(P), \dots, m_q(P))$  need not be attainable, but  $m(P) \leq c(\hat{x})$  for every  $\hat{x} \in X(P)$ , and  $m_s(P) = c^s(x)$  for the representative  $x$ . Every state also stores, for every  $t \in \{1, \dots, q\}$ , a flow  $x^t$  that attains  $m_t(P)$ , with valid potentials; we choose  $x^t$  minimum in  $X(P)$  for  $\prec_t$ , which breaks the ties among the minimizers of  $c^t$  at no extra cost, and  $x^s = x$ . These flows are used for the bound  $m(P)$ ; Section 7 shows that they can also play the role of the representative.

The global heap  $H$  contains the active states, ordered lexicographically by  $m(P)$ . The relevant circuit of a state is searched only when it is extracted. This priority is heuristic: neither property (N) nor the correctness proof requires states or flows to be generated in any particular order.

### 6.1. Incumbent set

The incumbent set  $U$  stores one flow of each nondominated objective vector among the feasible flows known so far. A new flow is inserted when its objective vector is neither dominated by nor equal to a stored vector, and stored flows dominated by it are removed. The incumbent set may be warm-started with arbitrary feasible flows, which then act as dominators, and it is the output of the algorithm. The representative and the flows  $x^t$  of every state are offered to  $U$  when they are computed.

### 6.2. Processing an extracted branch

Let  $P = P_p$  be extracted from  $H$ , and let  $a$  be the arc returned by the generator, which lies on a relevant circuit  $w$  of  $G_P(x)$  for the representative  $x$  of  $P$ . Two disjoint restrictions are created with the value  $x_e$  of the representative on  $e = e(a)$ . The residual parent becomes  $P\langle\bar{a}\rangle$ , which keeps  $x$  and its potentials, and it is reinserted in the heap. The child restriction is  $P\langle a\rangle$ . Its representative for  $\prec_s$  is not assumed to be  $x \oplus w$ . Instead, Proposition 2 computes

$$x' = \arg \min_{\prec_s} \{c(y) : y \in X(P\langle a\rangle)\}$$

346 from the cheapest circuit of  $G_P(x)$  for  $\prec_s$  containing  $a$ , together with new valid  
347 potentials. The flows  $x^t$ ,  $t \neq s$ , of both restrictions are reoptimized after the  
348 change of the bound of  $e$  (Section 6.3). The new representative, and every flow  $x^t$   
349 that changes, is offered to  $U$ . A state is not discarded because its representative  
350 is dominated, since a dominated representative may have efficient descendants;  
351 it is discarded only by the bound test of Proposition 4 below.

### 352 6.3. Dominance pruning of a partition

353 Any state can be discarded when its whole active partition is provably dom-  
354 inated. This is the dominance test of multiobjective branch-and-bound, which  
355 compares a lower bound set of a node with the incumbent set [35, 9]; here the  
356 componentwise bound (21) is the simplest lower bound set.

357 **Proposition 4** (Dominance bound pruning). *If a flow  $z$  in the incumbent set*  
358  *$U$  satisfies*

$$c(z) \prec_D m(P), \quad (22)$$

359 *then every flow in  $X(P)$  is dominated by  $z$  and  $P$  can be discarded.*

360 *Proof.* For every  $\hat{x} \in X(P)$ ,  $c(z) \leq m(P) \leq c(\hat{x})$  componentwise, and the  
361 strict component in  $c(z) \prec_D m(P)$  remains strict. If  $z$  is later removed from  
362  $U$  because another feasible flow dominates it, transitivity preserves the pruning  
363 certificate.  $\square$

364 Since only a minimal complete set is computed, the non-strict test  $c(z) \leq$   
365  $m(P)$  also discards  $P$ : every flow of  $X(P)$  then has a vector dominated by  
366 or equal to  $c(z)$ , so  $P$  contains no new vector. The implementation uses this  
367 non-strict form, which is also the one given by the search-region test of Section 7.

368 The exact vector  $m(P)$  is maintained rather than recomputed from scratch,  
369 through the flows  $x^t$ . At the root state  $P^0$ , whose bounds are  $l$  and  $u$ , the  $q$   
370 lexicographic problems are solved; their optimal flows and potentials are the  
371 flows  $x^t$  of  $P^0$ , and  $x^s$  is its representative. After one bound change on an arc  
372  $e$ , a stored flow  $x^t$  either remains feasible, and hence optimal, or violates the  
373 new bound on  $e$  by some amount  $k$ . In the latter case the flow on  $e$  is moved to  
374 the new bound, which creates an excess and a deficit of  $k$  units at the endpoints  
375 of  $e$  while keeping all residual reduced costs lexicographically nonnegative, and  
376 optimality is restored by at most  $k$  lexicographic shortest augmenting paths  
377 [1]. Hence the exact ideal point is propagated with at most  $k$  shortest-path  
378 computations per affected flow; for the representative,  $k = 1$  in the child, and  
379 the reoptimization is the computation of Proposition 2. The weighted bounds  
380 of Section 7 are single-objective problems with cost  $\mu \cdot c$  and are maintained in  
381 the same way.

382 6.4. General algorithm

---

**Algorithm 2:** Circulation-Ranking Algorithm for the MOIMCF

---

**Input:** Network  $G = (V, A)$ , balances  $b$ , bounds  $l, u$  and vector arc costs  $c \in \mathbb{Z}^q$ .

**Output:** A minimal complete set  $U$  of efficient flows.

- 1 Fix an index  $s \in \{1, \dots, q\}$ ;
  - 2 Compute the lexicographically optimal flows  $x^t$  for  $\prec_t$  and valid potentials  $d^t$ ,  
 $t = 1, \dots, q$ ;
  - 3  $P^0 \leftarrow (l, u, x^s, d^s)$ , which also stores  $x^1, \dots, x^q$  for  $m(P^0)$ ;  $H \leftarrow \emptyset$ ; initialize  
the incumbent set  $U$  and update it with  $x^1, \dots, x^q$ ;
  - 4 Insert  $P^0$  into  $H$ ;
  - 5 **while**  $H \neq \emptyset$  **do**
  - 6     Extract a state  $P = P_p$  from  $H$ ;
  - 7     **if** an exact bound certifies  $P$  dominated by  $U$  **then** discard  $P$ ; **continue**;
  - 383 8      $a \leftarrow \text{RELEVANTCIRCUIT}(P)$ , an arc of a relevant circuit of  $G_P(x)$  for the  
representative  $x$  of  $P$ ;
  - 9     **if**  $a = \text{NULL}$  **then** discard  $P$ , which contains no new vector; **continue**;
  - 10    Construct the child restriction  $P' \leftarrow P\langle a \rangle$ ; compute its representative by  
(18)–(19) and reoptimize its flows  $x^t$ ,  $t \neq s$ ;
  - 11    Update  $U$  with the representative of  $P'$  and the flows  $x^t$  that changed;
  - 12    **if** the bound of  $P'$  does not certify it dominated by  $U$  **then** insert  $P'$  into  
 $H$ ;
  - 13    Replace  $P_p$  by the complementary restriction  $P\langle \bar{a} \rangle$ , which keeps its  
representative, reoptimize its flows  $x^t$ ,  $t \neq s$ , and update  $U$  with those  
that changed;
  - 14    **if** the complementary partition is not certified dominated by  $U$  **then**  
insert  $P_p$  into  $H$ ;
  - 15 **return**  $U$ ;
- 

384     The test of Proposition 4 is applied to every restriction before it is inserted  
385 in  $H$ , and again when it is extracted, since  $U$  may have grown in between.

386 6.5. Search strategy and warm start

387     Three further choices of the implementation affect only the running time  
388 and the memory.

389     *Ideal-point priority and bounded memory.* States are inserted in the heap  
390 without searching their circuit, with key  $m(P)$ , and are extracted in lexico-  
391 graphic order of this key. Since this best-first order may keep many states open,  
392 new states are explored depth first once more than  $10^5$  states are open; this  
393 limits the growth of the heap without affecting the instances that do not reach  
394 the limit.

395     *Warm-started incumbent set.* As in the multi-objective successive shortest  
396 path algorithm of [31], where  $2q + 1$  vectors (the  $q$  single-objective optima and  
397  $q + 1$  weighted-sum optima) initialize the search, the incumbent set  $U$  is warm-  
398 started with weighted-sum optima: besides the  $q$  lexicographic roots of  $P^0$ , it  
399 receives the optima of the all-ones weighted sum and of the  $q$  weighted sums  
400 with one weight doubled. The warm-start flows act only as dominators and do  
401 not constitute a first phase.

402 *Probing along the returned circuit.* When the generator returns an arc of a  
 403 relevant circuit  $w$  of  $G_P(x)$ , where  $x$  is the representative used by the generator,  
 404 the flows  $x \oplus k\chi(w)$ ,  $k = 1, 2, \dots$ , up to the smallest residual capacity of the arcs  
 405 of  $w$  and at most 64, are offered to  $U$  as long as their vectors are new. They  
 406 are feasible by Lemma 1 and, like the warm-start flows, act only as dominators:  
 407 the partition, and hence the correctness argument, is unchanged.

## 408 7. The search region and box-restricted relevance

409 The relevance test (20) used by the local generator asks only for a circuit  
 410 of  $G_P(x^s)$  that improves some objective other than  $s$ . This section replaces it  
 411 by a test that depends on the part of the objective space that still has to be  
 412 explored. It is stated for an arbitrary number of objectives.

413 From now on all the flows  $x^t$  of Section 6 are used. Since  $x^t$  is minimum  
 414 in  $X(P)$  for  $\prec_t$ , it is, with its potentials, a representative of  $P$  in the sense  
 415 of Definition 2 for the order  $\prec_t$ , and Proposition 2 and the relevance test of  
 416 Section 5 hold for each of them. We therefore call  $x^1, \dots, x^q$  the *representatives*  
 417 of  $P$ . The results of this section hold for any of them, and the generator of  
 418 Section 7.3 examines each box from the representative closest to it.

### 419 7.1. Local upper bounds

420 We recall the representation of the search region by local upper bounds of  
 421 Klamroth et al. [15]; the definitions and the update rule below are theirs, and  
 422 the results of Sections 7.2 and 7.3 are new.

423 Let  $y^I \in \mathbb{Z}^q$  be the ideal point,  $y_r^I = \min\{c^r(x) : x \text{ feasible}\}$ , that is,  $y^I =$   
 424  $m(P^0)$  for the root state, and let  $M \in \mathbb{Z}^q$  be any vector with  $M > c(x)$  compo-  
 425 nentwise for every feasible flow  $x$ ; for instance  $M_r = 1 + \sum_{e \in A} \max\{c_e^r l_e, c_e^r u_e\}$ .  
 426 Every feasible vector lies in the box  $[y^I, M]$ . Let  $U \subseteq \mathbb{Z}^q$  be a finite set of  
 427 objective vectors of feasible flows known to the algorithm; for the incumbent  
 428 set of Section 6.1 we identify  $U$  with the set of its objective vectors. A vector  
 429  $y$  is *new* if it is neither dominated by nor equal to a member of  $U$ , that is, if  
 430 no  $z \in U$  satisfies  $z \leq y$ . Klamroth et al. [15] define the *search region* of  $U$   
 431 as the set of vectors of  $[y^I, M]$  that are new, and its *local upper bounds* as the  
 432 maximal points  $u$  of  $\{u \leq M : \text{no } z \in U \text{ satisfies } z < u\}$ , that is, the maximal  
 433 corners of strict boxes  $\{y : y < u\}$  that contain no vector of  $U$ ; the search region  
 434 is the union of the sets  $\{y \in [y^I, M] \cap \mathbb{Z}^q : y < u\}$  over the local upper bounds  
 435  $u$ . For integer vectors,  $y < u$  is equivalent to  $y \leq u - \mathbf{1}$ . We denote by  $\mathcal{U}(U)$   
 436 the set of local upper bounds; for  $U = \emptyset$  it is  $\{M\}$ .

437 The incremental maintenance of  $\mathcal{U}(U)$  is also due to Klamroth et al. [15]:  
 438 when a new vector  $y$  is inserted, every bound  $u$  with  $y < u$  is replaced by  
 439 its  $q$  children  $u^k$ , obtained from  $u$  by setting  $u_k^k = y_k$ , and only the maximal  
 440 bounds are kept, since a box contained in another adds nothing; Klamroth et al.  
 441 decide maximality with the defining points of each bound, and the family is then  
 442 exactly  $\mathcal{U}(U)$ . The tests below use the boxes of  $\mathcal{U}(U)$ ; they remain valid for any  
 443 family of boxes that covers the new vectors and contains no vector dominated  
 444 by or equal to a vector of  $U$ .

445 7.2. Box-restricted relevance

446 **Theorem 4** (Box-restricted relevance). *Let  $P$  be a state, let  $s \in \{1, \dots, q\}$  and*  
 447 *let  $x$  minimize  $c^s$  over  $X(P)$ , for instance its representative  $x^s$ , let  $u$  be an upper*  
 448 *bound, and put*

$$B_s = u_s - 1 - c^s(x), \quad B_r = c^r(x) - u_r + 1 \quad (r \neq s).$$

449 *Let  $y \in X(P)$  with  $c(y) \leq u - \mathbf{1}$ , and let  $y = x \oplus \sum_t \lambda_t \chi(w_t)$  be a conformal*  
 450 *decomposition as in Theorem 1. Then*

- 451 (i)  $0 \leq \gamma^s(w_t) \leq B_s$  for every  $t$ ;  
 452 (ii) if  $c(x) \not\leq u - \mathbf{1}$ , then  $B_r \geq 1$  for some  $r \neq s$ , and for every such  $r$  at least  
 453 one circuit  $w$  of the decomposition satisfies

$$B_r \gamma^s(w) + B_s \gamma^r(w) \leq 0. \quad (23)$$

454 *Proof.* Since  $x$  minimizes  $c^s$  over  $X(P)$ , no circuit of  $G_P(x)$  has negative cost  
 455  $\gamma^s$ , so  $\gamma^s(w_t) \geq 0$  for every  $t$ . As  $\lambda_t \geq 1$  and  $\sum_t \lambda_t \gamma^s(w_t) = c^s(y) - c^s(x) \leq B_s$ ,  
 456 claim (i) follows.

457 If  $c(x) \not\leq u - \mathbf{1}$ , some component satisfies  $c^r(x) \geq u_r$ ; since  $c^s(x) \leq c^s(y) \leq$   
 458  $u_s - 1$ , this component is  $r \neq s$ , and  $B_r \geq 1$ . Fix  $r$  with  $B_r \geq 1$  and suppose  
 459 that  $B_r \gamma^s(w_t) + B_s \gamma^r(w_t) > 0$  for every  $t$ . Multiplying by  $\lambda_t$  and summing,

$$B_r(c^s(y) - c^s(x)) + B_s(c^r(y) - c^r(x)) > 0.$$

460 Now  $c^r(y) \leq u_r - 1 = c^r(x) - B_r$  gives  $c^r(y) - c^r(x) \leq -B_r$ , hence  $B_r(c^s(y) -$   
 461  $c^s(x)) > B_s B_r$  and, since  $B_r \geq 1$ ,  $c^s(y) > c^s(x) + B_s = u_s - 1$ , contradicting  
 462  $c(y) \leq u - \mathbf{1}$ .  $\square$

463 **Corollary 1** (Box relevance test). *Let  $\mathcal{N}$  be a family of boxes that contain only*  
 464 *new vectors and cover the search region, let  $P$  be a state whose representatives*  
 465 *are not new, and let  $\mathcal{L} \subset \mathbb{Z}_{\geq 0}^q$  be a finite, possibly empty, family of weight vectors*  
 466 *with the weighted bounds*

$$m_\mu(P) = \min\{\mu \cdot c(\hat{x}) : \hat{x} \in X(P)\}, \quad \mu \in \mathcal{L}, \quad (24)$$

467 and let

$$\mathcal{N}(P) = \{u \in \mathcal{N} : m(P) \leq u - \mathbf{1} \text{ and } m_\mu(P) \leq \mu \cdot (u - \mathbf{1}) \text{ for every } \mu \in \mathcal{L}\},$$

468 where  $m(P)$  is the componentwise bound (21). For  $u \in \mathcal{N}(P)$  and  $s \in \{1, \dots, q\}$ ,  
 469 with  $B_s(u)$  and  $B_r(u)$  computed from  $x^s$  as in Theorem 4, declare a circuit  $w$   
 470 of  $G_P(x^s)$  relevant for  $(u, s)$  when

$$\exists r \neq s : \quad \gamma^s(w) \leq B_s(u), \quad B_r(u) \geq 1, \quad B_r(u) \gamma^s(w) + B_s(u) \gamma^r(w) \leq 0. \quad (25)$$

471 If, for every  $u \in \mathcal{N}(P)$ , there is some  $s$  such that no circuit of  $G_P(x^s)$  is relevant  
 472 for  $(u, s)$ , then  $X(P)$  contains no new vector.

473 *Proof.* Let  $c(y)$  be a new vector of  $X(P)$ . It lies in a box  $u \in \mathcal{N}$ , and  $m(P) \leq$   
474  $c(y) \leq u - \mathbf{1}$  and  $m_\mu(P) \leq \mu \cdot c(y) \leq \mu \cdot (u - \mathbf{1})$  for every  $\mu \in \mathcal{L}$  show  $u \in \mathcal{N}(P)$ .  
475 For every  $s$ , the box  $u$  does not contain  $c(x^s)$ , which is not new, so Theorem 4  
476 provides a circuit relevant for  $(u, s)$  in the conformal decomposition of  $y$  with  
477 respect to  $x^s$ . Hence the hypothesis of the corollary fails.  $\square$

478 The hypothesis that the representatives are not new holds in every state: a  
479 representative is compared with  $U$  when it is computed, and either it is inserted  
480 or a vector of  $U$  dominates it or equals it; a vector leaves  $U$  only when another  
481 one dominates it, so the relation persists by transitivity. The hypothesis is  
482 needed, since a box containing  $c(x^s)$  has  $B_r(u) \leq 0$  for every  $r \neq s$ ; if  $c(x^s)$  is  
483 new, the representative itself lies in the box and no circuit is needed.

484 The weighted bounds (24) are single-objective integer minimum cost flow  
485 problems, exactly like the components of  $m(P)$ , and they are maintained in the  
486 same way (Section 6.3). Together with  $m(P)$  they define the lower bound set  
487  $\{y : y \geq m(P), \mu \cdot y \geq m_\mu(P), \mu \in \mathcal{L}\}$  of the partition in the sense of Ehrgott  
488 and Gandibleux [6], and the definition of  $\mathcal{N}(P)$  is the test of a lower bound set  
489 against the local upper bounds used in multiobjective branch-and-bound [35, 9].  
490 They discard boxes of  $\mathcal{N}$  before any circuit is searched; the family used in the  
491 implementation is described in Section 9.

### 492 7.3. Detection restricted to boxes

493 A relaxation of the box test of Corollary 1, without the condition  $\gamma^s(w) \leq$   
494  $B_s(u)$ , can be decided in the same way as the basic test of Proposition 3, one  
495 box and one objective at a time. A detection that finds no cycle certifies that  
496 the box is empty for the partition; a cycle found gives a branching direction,  
497 but does not prove that the box contains a vector of the partition.

498 **Proposition 5** (Box-restricted negative-cycle detection). *Let  $P$  be a state, let*  
499  *$x = x^s$  be one of its representatives, let  $u \in \mathcal{N}(P)$  and  $r \neq s$  with  $B_s(u) \geq 1$*   
500 *and  $B_r(u) \geq 1$ , and give every arc  $a$  of  $G_P(x)$  the pair*

$$\theta_a = (B_r(u) \gamma_a^s + B_s(u) \gamma_a^r, \gamma_a^r),$$

501 *ordered lexicographically. A circuit  $w$  of  $G_P(x)$  satisfies  $B_r \gamma^s(w) + B_s \gamma^r(w) \leq 0$*   
502 *and  $\gamma^r(w) < 0$  if and only if  $\theta(w) \prec_{lex} 0$ . If  $G_P(x)$  has no lexicographically*  
503 *negative cycle with respect to  $\theta$ , then no  $y \in X(P)$  satisfies  $c(y) \leq u - \mathbf{1}$ .*

504 *Proof.* If  $\theta(w) \prec_{lex} 0$ , either  $B_r \gamma^s(w) + B_s \gamma^r(w) < 0$ , which with  $\gamma^s(w) \geq 0$  gives  
505  $\gamma^r(w) < 0$ , or the first component is zero and  $\gamma^r(w) < 0$ . Conversely, a circuit  
506 with both properties has  $\theta(w) \prec_{lex} 0$ . For the second statement, let  $y \in X(P)$   
507 with  $c(y) \leq u - \mathbf{1}$  and let  $y = x \oplus \sum_t \lambda_t \chi(w_t)$  be a conformal decomposition  
508 as in Theorem 1. Since  $\sum_t \lambda_t \gamma(w_t) = c(y) - c(x)$ ,  $c^s(y) - c^s(x) \leq B_s$  and  
509  $c^r(y) - c^r(x) \leq -B_r$ ,

$$\sum_t \lambda_t \theta(w_t) = \left( B_r(c^s(y) - c^s(x)) + B_s(c^r(y) - c^r(x)), c^r(y) - c^r(x) \right) \leq (0, -B_r)$$

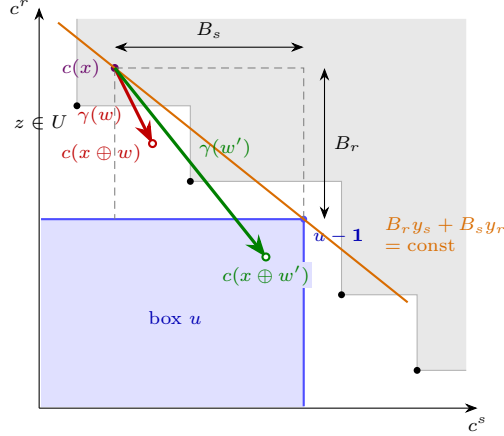


Figure 1: One detection of Proposition 5, in the plane of the objectives  $s$  and  $r$ . The point  $c(x)$  of the representative  $x = x^s$  lies above the box  $u$  in objective  $r$  ( $B_r \geq 1$ ), and its component  $s$  does not exceed that of  $u - 1$  ( $B_s \geq 0$ ). The line through  $c(x)$  and  $u - 1$  has normal  $(B_r, B_s)$ . Both circuits  $w$  and  $w'$  have costs below it, so both are lexicographically negative for  $\theta$  and either may be returned to branch;  $c(x \oplus w')$  lies in the box, whereas  $c(x \oplus w)$  does not. If no such circuit exists, the box is certified empty for  $X(P)$ .

510 componentwise, and this vector is lexicographically negative. The lexicographic  
 511 order is compatible with addition and with multiplication by positive integers,  
 512 so some circuit  $w_t$  has  $\theta(w_t) \prec_{lex} 0$ , and it is a lexicographically negative cycle  
 513 of  $G_P(x)$  for  $\theta$ .  $\square$

514 The absence of a  $\theta$ -negative cycle is the optimality condition of the minimum  
 515 cost flow problem with cost  $B_r c^s + B_s c^r$ , with  $c^r$  as tie-breaker:  $x$  minimizes this  
 516 weighted sum over  $X(P)$ , and the hyperplane  $B_r y_s + B_s y_r = B_r c^s(x) + B_s c^r(x)$   
 517 passes through  $c(x)$  and through the corner  $u - 1$  of the box, since  $B_r(u_s - 1) +$   
 518  $B_s(u_r - 1) = B_r c^s(x) + B_s c^r(x)$  (Figure 1). Proposition 5 is thus a weighted-sum  
 519 bound with weights adapted to the box, but no optimization problem is solved:  
 520 only the optimality of the representative is checked, and when it fails the cycle  
 521 found points to the box and opens the next branch.

522 Boxes with  $B_s(u) = 0$  are handled with  $\theta_a = (\gamma_a^r)$ : for a flow  $y \in X(P)$   
 523 with  $c(y) \leq u - 1$ , the same computation gives  $\sum_t \lambda_t \gamma^r(w_t) \leq -B_r < 0$ , so  
 524 some circuit of the decomposition is a negative cycle for  $\gamma^r$ . The conclusion of  
 525 Proposition 5 therefore holds for  $B_s(u) = 0$  as well.

526 The conclusion of Proposition 5 refers to the feasible set  $X(P)$  and to the  
 527 box  $u$ , and neither to the representative  $x$  nor to the objectives  $s$  and  $r$  used in  
 528 the detection. This gives certificates that can be reused.

529 **Corollary 2** (Box certificates). *Say that a pair  $(S, u)$ , where  $S$  is a state and  $u$*   
 530 *an upper bound, is a certificate if no  $y \in X(S)$  satisfies  $c(y) \leq u - 1$ . If  $(S, u)$*   
 531 *is a certificate, so is  $(P, u')$  for every state  $P$  with  $X(P) \subseteq X(S)$  and every*  
 532  *$u' \leq u$ . If  $(P, u)$  is a certificate for every  $u \in \mathcal{N}(P)$ , then  $X(P)$  contains no*

533 *new vector.*

534 *Proof.* If  $y \in X(P)$  satisfied  $c(y) \leq u' - \mathbf{1}$ , then  $y \in X(S)$  and  $c(y) \leq u - \mathbf{1}$ .  
 535 For the last claim, a new vector  $c(y)$  with  $y \in X(P)$  lies in a box  $u \in \mathcal{N}$ , and  
 536  $m(P) \leq c(y) \leq u - \mathbf{1}$  and  $m_\mu(P) \leq \mu \cdot c(y) \leq \mu \cdot (u - \mathbf{1})$  for  $\mu \in \mathcal{L}$  show  
 537  $u \in \mathcal{N}(P)$ , which is impossible if  $(P, u)$  is a certificate.  $\square$

538 By Proposition 5, a detection that finds no cycle for some objective  $r$  at  
 539  $S$  produces a certificate  $(S, u)$ , which, like a lower bound set in branch-and-  
 540 bound, remains valid for the child  $S\langle a \rangle$ , the complementary restriction  $S\langle \bar{a} \rangle$   
 541 and all their descendants, whatever representatives they use. For a state  $P$  we  
 542 denote by  $\mathcal{C}(P)$  the set of upper bounds  $u$  for which a certificate  $(S, u)$  with  
 543  $X(P) \subseteq X(S)$  is known; every box  $u' \leq u$  with  $u \in \mathcal{C}(P)$  can be skipped at  $P$ .

544 For every box  $u \in \mathcal{N}(P)$  and every  $s$ ,  $B_s(u) \geq 0$  because  $m_s(P) = c^s(x^s)$ ,  
 545 and some  $r \neq s$  has  $B_r(u) \geq 1$ , since  $c(x^s)$  is not new, so that  $c(x^s) \not\leq u - \mathbf{1}$ ,  
 546 and  $c^s(x^s) = m_s(P) \leq u_s - 1$ . The generator examines the boxes of  $\mathcal{N}(P)$   
 547 not covered by  $\mathcal{C}(P)$  and runs one detection for each of them, from the rep-  
 548 resentative closest to the box, that is, the one that minimizes the total excess  
 549  $\sum_{k=1}^q \max\{0, c^k(x^s) - u_k + 1\}$ , and with the first  $r$  after  $s$ , in cyclic order, with  
 550  $B_r(u) \geq 1$ . By Theorem 4, any other choice of  $s$  and  $r$  would also be valid.  
 551 The Bellman–Ford–Moore algorithm either returns a lexicographically negative  
 552 cycle for  $\theta$  in  $O(nm)$  time, and the generator stops, or certifies the box; the  
 553 detections therefore take  $O(|\mathcal{N}(P)|nm)$  time, and the choice of the opening arc  
 554 (Section 9) adds at most  $|w|$  lexicographic shortest-path computations. The  
 555 generator satisfies the following property, which replaces **(N)** for the box test:

556 **(N<sub>box</sub>)** If the generator returns NULL, then  $X(P)$  contains no new  
 557 vector; otherwise it returns an arc  $a$  of a circuit  $w$  of  $G_P(x^s)$  that  
 558 is lexicographically negative for the weights  $\theta$  of Proposition 5, for  
 559 some  $u \in \mathcal{N}(P)$ , some  $s$  and some  $r \neq s$ .

560 The first part follows from Corollary 2. Such a circuit is simple and does not  
 561 contain  $\bar{a}$ , since an arc and its twin have costs that cancel, so Proposition 2  
 562 applies to the returned arc.

#### 563 7.4. The complete method

564 Algorithms 3 and 4 state the method used in the implementation. With  
 565 respect to Algorithms 1 and 2, the generator is replaced by Algorithm 3, the  
 566 pruning test becomes  $\mathcal{N}(P) = \emptyset$ , which contains the dominance test of Propo-  
 567 sition 4 and adds the weighted bounds (24), the child and the complementary  
 568 restriction inherit the certificates of the parent (Corollary 2), and the local up-  
 569 per bounds are updated whenever a vector enters the incumbent set, including  
 570 the representatives, the warm-start and probing flows of Section 6.5. A worked  
 571 example of the method is given in Section S1 of the Supplementary Material  
 572 (SuppMat).

---

**Algorithm 3:** BOXRELEVANTCIRCUIT( $P$ )

---

**Input:** A state  $P$  with its representatives  $x^1, \dots, x^q$  and its certificates  $\mathcal{C}(P)$ ; the local upper bounds  $\mathcal{U}(U)$  of the incumbent set.

**Output:** An opening arc  $a$  of a relevant circuit of  $G_P(x^s)$  for some  $s$ ; or NULL.

```

1 foreach  $u \in \mathcal{U}(U)$  with  $m(P) \leq u - 1$  and  $m_\mu(P) \leq \mu \cdot (u - 1)$  for every
   $\mu \in \mathcal{L}$  do
2   if  $u \leq u'$  for some  $u' \in \mathcal{C}(P)$  (Corollary 2) then continue;
3    $s \leftarrow$  the representative closest to  $u$ , minimizing
      $\sum_k \max\{0, c^k(x^s) - u_k + 1\}$ ;  $x \leftarrow x^s$ ;
4    $B_s \leftarrow u_s - 1 - c^s(x)$ ;
5    $r \leftarrow$  the first  $r \neq s$  after  $s$ , in cyclic order, with  $B_r = c^r(x) - u_r + 1 \geq 1$ ;
573   // exists, since  $c(x)$  is not new
6   Give every arc  $a$  of  $G_P(x)$  the weight  $\theta_a = (B_r \gamma_a^s + B_s \gamma_a^r, \gamma_a^r)$  if  $B_s \geq 1$ ,
     and  $\theta_a = (\gamma_a^r)$  if  $B_s = 0$ ; // Proposition 5
7   Run the Bellman–Ford–Moore algorithm on the arcs of  $G_P(x)$  compatible
     with  $u$  (Section 7.5) with the weights  $\theta$ ;
8   if a lexicographically negative cycle  $w$  is found then
9     Offer  $x \oplus k\chi(w)$ ,  $1 \leq k \leq \min\{64, \min_{a \in w} r_a(x)\}$ , to  $U$ ; // probing,
     Section 6.5
10    return an arc of  $w$ ; // rule of Section 9
11  else
12     $\mathcal{C}(P) \leftarrow \mathcal{C}(P) \cup \{u\}$ ; // no vector of  $X(P)$  in the box  $u$ 
13 return NULL;

```

---

### 7.5. Arcs compatible with a box

The detection of Proposition 5 can be run on a smaller network. For a representative  $x^k$  of  $P$ , with the potentials  $d^k$  of Definition 2, and a residual arc  $a = (i, j)$  of  $G_P(x^k)$ , write  $\bar{\gamma}_a^k = \gamma_a^k + d_{i,k}^k - d_{j,k}^k$  for the component  $k$  of its reduced vector cost (16); by (17),  $\bar{\gamma}_a^k \geq 0$ . Recall that  $m_k(P) = c^k(x^k)$ .

**Proposition 6** (Arcs compatible with a box). *Let  $P$  be a state, let  $u$  be an upper bound, let  $y \in X(P)$  with  $c(y) \leq u - 1$ , and let  $k \in \{1, \dots, q\}$ . If  $a$  is a residual arc of  $G_P(x^k)$  with original arc  $e$  and  $y$  moves  $e$  away from  $x_e^k$  in the direction of  $a$ , that is,  $y_e > x_e^k$  if  $a = e^+$  and  $y_e < x_e^k$  if  $a = e^-$ , then*

$$\bar{\gamma}_a^k \leq u_k - 1 - m_k(P).$$

*Proof.* Let  $y = x^k \oplus \sum_t \lambda_t \chi(w_t)$  be a conformal decomposition as in Theorem 1. By conformality,  $a$  belongs to some circuit  $w_t$ . Directed-cycle costs do not change with the potentials, and every reduced cost  $\bar{\gamma}^k$  is nonnegative, so

$$c^k(y) - m_k(P) = \sum_t \lambda_t \gamma^k(w_t) = \sum_t \lambda_t \sum_{a' \in w_t} \bar{\gamma}_{a'}^k \geq \bar{\gamma}_a^k.$$

Since  $c^k(y) \leq u_k - 1$ , the claim follows.  $\square$

We use Proposition 6 in two ways.

---

**Algorithm 4:** Circulation-Ranking Algorithm for the MOIMCF with the search region

---

**Input:** Network  $G = (V, A)$ , balances  $b$ , bounds  $l, u$ , vector arc costs  $c \in \mathbb{Z}^q$  and a family  $\mathcal{L}$  of weight vectors (Section 9).  
**Output:** A minimal complete set  $U$  of efficient flows.

- 1 Compute the lexicographically optimal flows  $x^s$  for  $\prec_s$  with their potentials,  $s = 1, \dots, q$ , and the weighted bounds  $m_\mu(P^0)$ ,  $\mu \in \mathcal{L}$ ;
- 2  $P^0 \leftarrow (l, u, (x^s, d^s)_s)$  with  $\mathcal{C}(P^0) = \emptyset$ ;  $H \leftarrow \{P^0\}$ ;
- 3 Initialize  $U$  with  $x^1, \dots, x^q$  and the warm-start flows of Section 6.5, and  $\mathcal{U}(U)$  as in Section 7.1;
- 4 **while**  $H \neq \emptyset$  **do**
- 5     Extract from  $H$  the state  $P = P_p$  with lexicographically smallest  $m(P)$  (depth first above  $10^5$  open states, Section 6.5);
- 6     **if** every box  $u \in \mathcal{N}(P)$  is covered by  $\mathcal{C}(P)$  **then** discard  $P$ , which contains no new vector; **continue**;
- 7     Fix arcs of  $P$  (Section 7.5), reoptimize the representatives and weighted bounds that violate the new bounds, and update  $U$  and  $\mathcal{U}(U)$ ;
- 8     **if**  $X(P) = \emptyset$  after the fixing **then** discard  $P$ , which contains no new vector; **continue**;
- 9      $a \leftarrow \text{BOXRELEVANTCIRCUIT}(P)$ , an arc of a circuit of  $G_P(x^s)$ ;  
       // Algorithm 3
- 10    **if**  $a = \text{NULL}$  **then** discard  $P$ , which contains no new vector; **continue**;
- 11    Construct the child restriction  $P' \leftarrow P\langle a \rangle$ , with the value of  $x^s$  on  $e(a)$ , and  $\mathcal{C}(P') \leftarrow \mathcal{C}(P)$ ;
- 12    Compute the representative  $x^s$  of  $P'$  by (18)–(19), reoptimize its other representatives and its weighted bounds, and update  $U$  and  $\mathcal{U}(U)$  with the representatives that changed;
- 13    **if**  $\mathcal{N}(P') \neq \emptyset$  **then** insert  $P'$  into  $H$ ;
- 14    Replace  $P_p$  by the complementary restriction  $P\langle \bar{a} \rangle$ , which keeps the representative  $x^s$  of  $P$ , its potentials and  $\mathcal{C}(P)$ ; reoptimize its other representatives and its weighted bounds, and update  $U$  and  $\mathcal{U}(U)$  with the representatives that changed;
- 15    **if**  $\mathcal{N}(P_p) \neq \emptyset$  **then** insert  $P_p$  into  $H$ ;
- 16 **return**  $U$ ;

---

588     *Detection on the compatible arcs of a box.* When the box  $u$  is examined  
 589 from the representative  $x^s$ , a residual arc  $a$  of  $G_P(x^s)$  with original arc  $e$  is  
 590 removed if, for some  $k$  with  $x_e^k = x_e^s$  for which  $a$  is also an arc of  $G_P(x^k)$ ,  
 591  $\bar{\gamma}_a^k > u_k - 1 - m_k(P)$ ; for  $k = s$  the condition is  $\bar{\gamma}_a^s > B_s(u)$ . Since  $x_e^k = x_e^s$ ,  
 592 Proposition 6 excludes  $a$  from the conformal decomposition of every flow of  
 593  $X(P)$  in the box, so Proposition 5 holds on the reduced network.

594     *Inherited fixing of arcs.* Let

$$S_k(P) = \max\{u_k - 1 : u \in \mathcal{N}(P), u \text{ not covered by } \mathcal{C}(P)\} - m_k(P).$$

595     If a residual arc  $a$  of  $G_P(x^k)$  with original arc  $e$  satisfies  $\bar{\gamma}_a^k > S_k(P)$ , then,  
 596 by Proposition 6, no flow of  $X(P)$  whose vector is new moves  $e$  away from

597  $x_e^k$  in the direction of  $a$ , and the bound of  $e$  in  $P$  can be set to  $u_e^P = x_e^k$  if  
 598  $a = e^+$ , or  $l_e^P = x_e^k$  if  $a = e^-$ . The new bounds are bounds of the state, so  
 599 the child and the complementary restriction, and all their descendants, inherit  
 600 them. All the bounds found for  $P$  are imposed at once when  $P$  is extracted,  
 601 and each representative and each weighted bound whose flow violates them  
 602 is reoptimized once. Bounds obtained from different representatives may cross,  
 603  $l_e^P > u_e^P$ , or leave no feasible flow; then  $X(P)$  is empty after the fixing and, since  
 604 the removed flows have vectors that are not new,  $P$  contains no new vector and  
 605 is discarded.

## 606 8. Correctness

607 The proof uses the lexicographic-root invariant of Proposition 2, for each of  
 608 the  $q$  cyclic orders, together with the reoptimization of Section 6.3, the bound  
 609 partition of Proposition 1, and the properties of the local generator: **(N)** for  
 610 the basic test (Proposition 3) and **(N<sub>box</sub>)** for the box test (Section 7.3). No  
 611 ordering property of the global heap is required.

612 **Proposition 7** (Exhaustion of a partition). *Let  $P$  be a state with representatives*  
 613  *$x^1, \dots, x^q$  for which the generator returns NULL.*

- 614 (i) *With the basic test (20) for the representative  $x = x^s$  of Algorithm 2,*  
 615  *$c(x) \leq c(y)$  componentwise for every  $y \in X(P)$ .*
- 616 (ii) *With the box test, no  $y \in X(P)$  has a vector that is neither dominated by*  
 617 *nor equal to a vector of the current incumbent set  $U$ .*

618 *Proof.* (i) By property **(N)**, no relevant circuit remains in  $G_P(x^s)$ . Let  $y \in$   
 619  $X(P)$  and let  $y = x^s \oplus \sum_t \lambda_t \chi(w_t)$  be the conformal decomposition of Theorem 1  
 620 in  $G_P(x^s)$ . Every circuit  $w_t$  is a circuit of  $G_P(x^s)$ , hence nonnegative for  $\prec_s$  and  
 621 not relevant; hence it is componentwise nonnegative, and so is  $c(y) - c(x^s)$ . (ii)  
 622 This is property **(N<sub>box</sub>)**: when the generator returns NULL, every box of  $\mathcal{N}(P)$   
 623 is covered by a certificate, inherited from a state whose feasible set contains  
 624  $X(P)$  or obtained by a detection during the call, and Corollary 2 applies.  $\square$

625 **Theorem 5** (Correctness). *Algorithm 2, with the basic generator of Algo-*  
 626 *rithm 1, and Algorithm 4, with the box generator of Algorithm 3, terminate*  
 627 *and return a minimal complete set of efficient flows.*

628 *Proof.* By Proposition 1, every processed state is replaced by restrictions with  
 629 a strictly smaller total bound width  $\sum_e (u_e^P - l_e^P)$ , a nonnegative integer, so the  
 630 partition tree is finite; by Proposition 2 and Section 6.3, every state has its  $q$   
 631 lexicographic roots, so the generator is well defined.

632 Let  $v \in Y_N$ . Several efficient flows may have the vector  $v$ , and the algorithm  
 633 need not compute a given one; we show that exactly one flow with vector  $v$  ends  
 634 in  $U$ . Suppose that no flow with vector  $v$  ever enters  $U$ . Since  $v$  is nondominated,  
 635  $v$  is then new at every moment. Let  $y$  be an efficient flow with  $c(y) = v$ . At  
 636 every moment  $y$  belongs to  $X(P)$  for exactly one state  $P$  of the partition, since

637 the fixing of Section 7.5 removes only flows whose vectors are not new, and,  
 638 since the tree is finite, this state is eventually discarded, either by a bound or  
 639 because the generator returns NULL. In both cases  $X(P)$  contains no new vector  
 640 at that moment, which contradicts  $y \in X(P)$ . Indeed, a state discarded by a  
 641 bound has  $c(z) \leq m(P)$  for some  $z \in U$  (Proposition 4, in its non-strict form) or  
 642  $\mathcal{N}(P) = \emptyset$  (Corollary 1). If the generator returns NULL, this is Proposition 7(ii)  
 643 for the box test; for the basic test, Proposition 7(i) gives  $c(x^s) \leq c(\hat{x})$  for every  
 644  $\hat{x} \in X(P)$ , and  $c(x^s)$  is not new, since  $x^s$  was compared with  $U$  when it was  
 645 computed. Hence some flow with vector  $v$ ,  $y$  or an equivalent one, enters  $U$ . It is  
 646 never removed, since a flow leaves  $U$  only when a flow that dominates it enters,  
 647 and no other flow with vector  $v$  enters  $U$ , since equal vectors are rejected.  
 648 Conversely, a flow  $z$  in  $U$  with  $c(z) \notin Y_N$  is dominated by some  $v \in Y_N$ ; a  
 649 flow with vector  $v$  eventually enters  $U$  and removes  $z$ , and later insertions of  $z$   
 650 or of flows equivalent to it are rejected.  $\square$

## 651 9. Implementation

652 The algorithm is implemented in C, without external libraries, and runs on  
 653 a single thread. Instances are read in the DIMACS minimum cost flow format  
 654 with  $q$  cost columns per arc, which allows lower bounds and negative costs.

655 The network is stored in forward and reverse star form, and the residual  
 656 network is never built explicitly: the residual capacity of an arc is read from  
 657 the flow and the bounds of the current state. Each state keeps its bounds, its  
 658 lexicographic roots with their node potentials, which give  $m(P)$ , the bounds  
 659  $m_\mu(P)$  with the flows that attain them, and its box certificates. A child copies  
 660 these data and reoptimizes only the problems affected by the changed arc. All  
 661 minimum cost flow problems, whether lexicographic, in any cyclic order, or  
 662 weighted, are solved by the same successive shortest path routine. All quantities  
 663 are 64-bit integers.

664 The local upper bounds are updated as in Klamroth et al. [15]. The generator  
 665 does not list  $\mathcal{N}(P)$ : it scans the bounds and examines each live box as soon as  
 666 it is found, so that the scan stops at the first box with a cycle. For the weighted  
 667 bounds we use, for each pair of objectives  $i < j$ , the vectors  $e_i + e_j$ ,  $2e_i + e_j$  and  
 668  $e_i + 2e_j$ , together with the all-ones vector. Each detection is a Bellman–Ford–  
 669 Moore run in  $O(nm)$  time [2, 8, 26]; when it finds no cycle, the box is stored as  
 670 a certificate of the state.

671 The generator returns a cycle  $w$  of  $G_P(x^s)$ , but the child is represented by the  
 672 lexicographically cheapest circuit for  $\prec_s$  through the chosen arc (Proposition 2),  
 673 which need not point towards the box. We therefore compute, for each arc of  $w$ ,  
 674 the representative of the corresponding child, and choose the arc whose circuit is  
 675 most negative for the first component of the weight  $\theta$  of the box that produced  
 676  $w$ . Further details (computation from scratch, bounds on the labels, storage of  
 677 the local upper bounds and of the certificates) are given in Section S2 of the  
 678 SuppMat.

## 679 10. Computational experiments

### 680 10.1. Reference algorithms

681 For  $q \geq 3$  we compare the circulation-ranking algorithm with two exact  
682 objective-space methods for multi-objective integer programming, applied to  
683 the integer program with objectives (7) and constraints (2)–(4) for the same  
684 instances.

- 685 • The search-zone method of Tamby and Vanderpooten [38], which extends  
686 the  $\varepsilon$ -constraint method to any number of objectives and improves in their  
687 experiments on earlier objective-space algorithms, including that of Kirlik  
688 and Sayın [14]. We implemented it in C for this study; we refer to it as  
689 TV.
- 690 • The defining point algorithm of Dächert, Fleuren and Klamroth [5], in its  
691 augmented variant, with the code published by the authors [4]; we only  
692 added a limit on the CPU time and a summary of the output. We refer  
693 to it as DPA.

694 Both solve their integer subproblems with CPLEX 22.1.2 on a single thread. The  
695 three methods are exact: they are designed to compute a minimal complete set.

### 696 10.2. Instances

697 Our benchmark has a full factorial design. The networks are generated with  
698 NETGEN [16], with  $n \in \{20, 30, 40, 50, 75\}$  nodes,  $m \in \{3n, 4n, 5n\}$  arcs,  $\lceil \sqrt{n} \rceil$   
699 sources and  $\lceil \sqrt{n} \rceil$  sinks, half of them transshipment nodes, as in the instances  
700 of Kovács [23], capacities in  $[0, 50]$  and total supply  $S \in \{10, 20, 30, 40, 50\}$ ; the  
701 first cost is generated by NETGEN in  $[0, 100]$  and the remaining  $q - 1$  costs are  
702 uniform in  $[0, 100]$ . For each  $q \in \{3, 4, 5\}$  and each combination of  $n$ ,  $m$  and  $S$ ,  
703 three instances are generated with different seeds, 675 instances in all (Table S3  
704 of the SuppMat gives the networks and the median of  $|Y_N|$  for each class). A  
705 further block with  $q = 3$  and  $S = 100$  (45 instances, with the same networks  
706 and seeds), where the nondominated sets have tens of thousands of vectors, is  
707 reported in Section S4 of the SuppMat.

708 We have also considered 66 instances with  $q \geq 3$  of Könen, Löhken and  
709 Stiglmayr [21], which are publicly available [22]: three instances in each of the  
710 22 classes with at most 200 nodes whose results are published in [21] or in the  
711 thesis of Könen [17], chosen among those whose nondominated set is known  
712 with certainty. Since they are chosen among instances that DPA and TV solve,  
713 they serve as an external validation set rather than as a neutral comparison of  
714 the number of solved instances. The selection, the parameters of the classes and  
715 the results are given in Section S5 of the SuppMat.

716 The number of nondominated vectors depends less on the size of the network  
717 than on the range of values of each objective, which grows with the costs, the  
718 supply and the capacities: with  $q = 3$  and  $n = 20$ , for instance, the median of  
719  $|Y_N|$  over the 9 instances grows from 140 with supply 10 to 4007 with supply  
720 50 (see Table S3 of the SuppMat).

### 721 10.3. Results

722 All experiments are performed on an Intel Xeon 3.70 GHz processor with 4  
723 cores and 8 hardware threads and 64 GB of RAM running Ubuntu 16.04. We  
724 refer to the circulation-ranking algorithm as CR. All runs are performed on this  
725 machine, each on a single thread fixed to its own hardware thread, so that two  
726 runs may share a physical core, with a limit of 3600 seconds of CPU time (900  
727 seconds in the experiment of Table 2). We record the CPU time, excluding the  
728 reading of the instance, the number  $|Y_N|$  of nondominated vectors, the number  
729 of processed partitions of CR, the number of integer subproblems solved by TV  
730 and DPA, and the peak memory of each run. The outputs of TV and DPA  
731 may contain dominated or repeated vectors, due to the numerical tolerances of  
732 CPLEX; they are removed in a post-processing step whose time is not included,  
733 which can only favour TV and DPA. After this step, the methods that terminate  
734 on an instance return the same nondominated sets, with one exception, an  
735 instance of Könen et al. on which the output of TV misses one nondominated  
736 vector (Section S4 of the SuppMat). CR uses exact integer arithmetic and is  
737 not affected by tolerances; its fronts coincide with those of DPA on the 519  
738 instances solved by both.

739 Table 1 summarizes the comparison. For each group of instances it gives  
740 the number of instances solved within the time limit, the shifted geometric  
741 mean of the CPU time with a shift of 1 second, counting unsolved instances at  
742 3600 seconds, the median of the known values of  $|Y_N|$ , and, over the instances  
743 solved by CR, the geometric mean of the number of processed partitions per  
744 nondominated vector. The instances are grouped by  $q$  and by total supply. The  
745 results for every class are given in Section S4 of the SuppMat, and Figure 2  
746 compares the times of CR and DPA instance by instance.

747 CR is the fastest method for every number of objectives: its mean time is  
748 2.7, 4.7 and 3.6 times smaller than that of DPA for  $q = 3, 4$  and 5. Figure 2  
749 confirms this instance by instance: on the instances solved by both methods,  
750 CR is faster than DPA on 175 of 203 with  $q = 3$ , 142 of 145 with  $q = 4$  and all  
751 75 with  $q = 5$ , and the geometric mean of the ratio between their times grows  
752 from 7.6 for  $q = 3$  to 18.8 for  $q = 4$  and 48.5 for  $q = 5$ . With  $q = 5$ , CR  
753 also solves more instances, 102 against 75 for DPA and 38 for TV, and with  
754  $q = 4$  CR and DPA solve almost the same number, 149 and 150, against 94 for  
755 TV. With  $q = 3$ , however, DPA solves all the instances with supply at most  
756 50, whereas CR leaves 22 of them unsolved: one with supply 30, 6 with supply  
757 40 and 15 with supply 50. Their nondominated sets have a median of 11 500  
758 vectors, and DPA solves each of them in less than 24 minutes. In the additional  
759 block with  $q = 3$  and supply 100 (Section S4 of the SuppMat), DPA solves 38  
760 of the 45 instances, TV 18 and CR 14, and DPA is also faster.

761 The last column of Table 1 explains the effect of the supply. Each processed  
762 partition with a live box not covered by a certificate costs at least one negative-  
763 cycle detection, and the number of live boxes, which bounds the number of  
764 detections per partition, grows with  $|Y_N|$ . With  $q = 3$ ,  $n = 20$  and  $m = 60$ ,  
765 the number of partitions per nondominated vector grows from 2 with supply

Table 1: Comparison of CR, DPA and TV on our benchmark with a limit of 3600 seconds, by number of objectives  $q$  and total supply. Solved: number of solved instances. Time: shifted geometric mean of the CPU time in seconds, with unsolved instances counted at the limit; the best value is in bold.  $|Y_N|$ : median of the known values (number of instances with known  $|Y_N|$  in parentheses when not all). Part./vec.: geometric mean of the number of partitions processed by CR per nondominated vector.

| $q$ | Supply | #   | Solved |     |     | Time (s)      |              |        | Part./     |      |
|-----|--------|-----|--------|-----|-----|---------------|--------------|--------|------------|------|
|     |        |     | CR     | DPA | TV  | CR            | DPA          | TV     | $ Y_N $    | vec. |
| 3   | 10     | 45  | 45     | 45  | 45  | <b>0.3</b>    | 6.2          | 9.3    | 140        | 5    |
|     | 20     | 45  | 45     | 45  | 45  | <b>3.7</b>    | 27.4         | 43.4   | 638        | 17   |
|     | 30     | 45  | 44     | 45  | 45  | <b>18.0</b>   | 61.9         | 94.7   | 1534       | 36   |
|     | 40     | 45  | 39     | 45  | 45  | <b>91.4</b>   | 122.0        | 215.5  | 3110       | 66   |
|     | 50     | 45  | 30     | 45  | 43  | 237.8         | <b>203.6</b> | 394.1  | 5342       | 65   |
|     | All    | 225 | 203    | 225 | 223 | <b>18.1</b>   | 49.4         | 81.2   | 1346       | 24   |
| 4   | 10     | 45  | 45     | 45  | 44  | <b>1.1</b>    | 49.3         | 78.9   | 345        | 5    |
|     | 20     | 45  | 44     | 44  | 29  | <b>41.7</b>   | 445.0        | 1137.9 | 3284 (44)  | 17   |
|     | 30     | 45  | 34     | 33  | 12  | <b>290.7</b>  | 1108.6       | 2057.9 | 7640 (36)  | 24   |
|     | 40     | 45  | 15     | 17  | 6   | <b>1092.2</b> | 1850.0       | 2372.8 | 10849 (17) | 22   |
|     | 50     | 45  | 11     | 11  | 3   | <b>1792.8</b> | 2487.8       | 3068.3 | 10251 (12) | 24   |
|     | All    | 225 | 149    | 150 | 94  | <b>137.4</b>  | 647.4        | 1063.1 | 2659 (154) | 14   |
| 5   | 10     | 45  | 43     | 40  | 29  | <b>6.3</b>    | 277.1        | 554.4  | 527 (43)   | 6    |
|     | 20     | 45  | 30     | 19  | 6   | <b>358.6</b>  | 1862.7       | 2952.8 | 4732 (30)  | 11   |
|     | 30     | 45  | 16     | 10  | 1   | <b>1473.8</b> | 2822.9       | 3568.3 | 6454 (16)  | 11   |
|     | 40     | 45  | 8      | 4   | 1   | <b>2303.1</b> | 3116.9       | 3314.1 | 8260 (8)   | 12   |
|     | 50     | 45  | 5      | 2   | 1   | <b>2863.0</b> | 3370.9       | 3599.7 | 14210 (5)  | 11   |
|     | All    | 225 | 102    | 75  | 38  | <b>479.1</b>  | 1726.5       | 2337.2 | 3048 (102) | 8    |

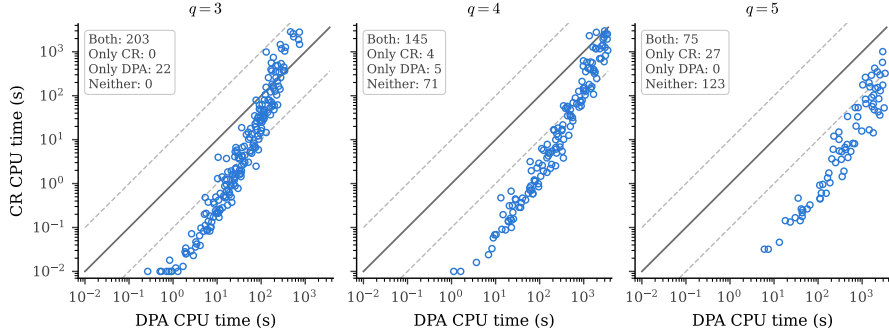


Figure 2: CPU time of CR against that of DPA for every instance of our benchmark, for  $q = 3$ , 4 and 5, in logarithmic scale. Only the instances solved by both methods are drawn, and each panel gives the number of instances solved by both, by only one of them or by neither. Points below the diagonal are instances on which CR is faster; the dashed lines mark a factor of 10, and times below 0.01 seconds are drawn at 0.01.

10 to 36 with supply 50 and 104 with supply 100 (Section S4 of the SuppMat). A diagnosis on six instances of the block with  $q = 3$  and supply 100 shows where this work goes. Between 94% and 99.5% of the processed partitions are sterile: neither they nor the partitions derived from them yield a vector of the final nondominated set. Most of the effort is therefore spent certifying that the boxes of the search region contain no new vector, not in finding the vectors, and this verification grows with the number of live boxes.

To identify the factors behind this behaviour, we generated a small one-factor-at-a-time experiment with  $q = 3$  and 5 instances per configuration, in which one parameter of a base configuration is changed at a time (Table 2). The base configuration is that of our benchmark, with  $n = 30$ ,  $m = 150$ , costs in  $[0, 100]$ , 7 sources and 7 sinks, and total supply 20; the variants use costs in  $[1, 10]$  (C10), 2 sources and 2 sinks (S2),  $m = 300$  (D10), total supply 50 (O50) or 100 (O100), and the four changes C10, S2, D10 and O50 together (ALL); details are given in Section S3 of the SuppMat. Smaller costs and fewer sources and sinks do not harm CR, which remains between 6 and 9 times faster than DPA, whereas a higher density or a larger supply multiply the number of partitions per nondominated vector by 9 (D10) and 5 (O50); with  $m = 300$  DPA becomes faster, and with supply 50 or more, or with the four changes together, CR leaves instances unsolved within 900 seconds that DPA solves.

On the 66 instances of Könen, Löhken and Stiglmayr, DPA and TV terminate on all of them and CR solves 59, and it is faster than DPA on 48 of these. The seven instances that CR does not solve belong to the classes of the thesis of Könen with  $n = 200$ ,  $m = 800$  and with  $n = 50$ ,  $m = 400$  and 600, whose nondominated sets have up to 17 782 vectors (Section S5 of the SuppMat).

The partition has a cost in memory: every open state stores its flows, potentials and bounds. On our benchmark, the median peak memory is 266 MB

Table 2: One-factor-at-a-time experiment with  $q = 3, 5$  instances per configuration and a limit of 900 seconds. Columns as in Table 1.

| Configuration | # | Solved |     |    | Time (s)   |              |       | $ Y_N $ med | Part./vec. |
|---------------|---|--------|-----|----|------------|--------------|-------|-------------|------------|
|               |   | CR     | DPA | TV | CR         | DPA          | TV    |             |            |
| BASE          | 5 | 5      | 5   | 5  | <b>6.2</b> | 54.8         | 92.2  | 1660        | 24         |
| C10           | 5 | 5      | 5   | 5  | <b>1.6</b> | 13.0         | 19.2  | 544         | 17         |
| S2            | 5 | 5      | 5   | 5  | <b>3.7</b> | 21.2         | 34.8  | 554         | 15         |
| D10           | 5 | 5      | 5   | 5  | 178.2      | <b>132.9</b> | 242.7 | 2642        | 215        |
| O50           | 5 | 3      | 5   | 4  | 324.6      | <b>228.9</b> | 394.4 | 4027        | 128        |
| O100          | 5 | 0      | 2   | 0  | 900.0      | <b>809.5</b> | 900.0 | 22494 (2)   | –          |
| ALL           | 5 | 0      | 5   | 4  | 900.0      | <b>277.7</b> | 534.0 | 14452       | –          |

for CR, 129 MB for DPA and 33 MB for TV. CR exceeds 1 GB on 251 runs, 212 of them stopped by the time limit with many open states, and reaches 8.9 GB, whereas DPA reaches 1.8 GB and TV 122 MB; on the instances that CR solves, its median peak memory is 53 MB.

Finally, an ablation study on 24 instances, reported in Section S6 of the SuppMat, measures the contribution of each component through the number of processed partitions: the search region and the weighted bounds are the decisive components, and the arcs compatible with a box and their fixing reduce the number of partitions by a factor of 1.4 with  $q = 3$ , 1.7 with  $q = 4$  and 2.1 with  $q = 5$ .

## 11. Conclusions

We have developed an exact algorithm for the MOIMCF that exploits a characterization of efficient flows by circulations in the residual network and computes a minimal complete set without a solver and without a supported-solution phase. Every partition is represented by its lexicographic roots for the  $q$  cyclic orders of the objectives, which give its ideal point, integer bound decisions on single arcs partition the feasible set, and the root and valid potentials of each child are recovered by one restricted lexicographic shortest-path computation.

The central result relates the residual circuits to the boxes of the search region: if a partition has a new nondominated vector in a box, the residual network of each of its representatives contains a lexicographically negative cycle for weights determined by the box and by the representative, so that the detection can start from the representative closest to the box. One negative-cycle detection per box therefore either returns a circuit directed at the box, which is used to branch, or certifies that the box contains no vector of the partition, a certificate inherited by all its subpartitions.

The computational study, with three to five objectives, compares the method with two objective-space algorithms whose integer subproblems are solved with CPLEX. The method is faster than DPA on most of the instances solved by both, on all of them with five objectives, and the ratio between their times is larger with four and five objectives; with five objectives it also solves more instances

than DPA and TV. With three objectives and a large supply, however, it leaves unsolved instances that DPA solves, and the same happens on the densest classes of Könen, Löhken and Stiglmayr. The number of partitions per nondominated vector and a one-factor-at-a-time experiment identify the density of the network and the supply as the factors behind this behaviour, and a diagnosis shows that most of the processed partitions only verify that the boxes of the search region contain no new vector. Reducing this verification work on dense networks with large supplies is the main open direction.

## Acknowledgements

This research has been partially supported by the Spanish Ministry of Science, Innovation and Universities under grant PID2023-148599NB-I00.

## Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The code of CR and of TV, the instances of our benchmark and of the factorial experiment, the generator scripts and the results are available at <https://doi.org/10.5281/zenodo.23209259>. The instances of Könen, Löhken and Stiglmayr are available at their repository [22].

## Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used Claude (Anthropic) to improve the language and readability of the manuscript, to check proofs and cross-references, and to assist in the implementation and testing of the computer code. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

## References

- [1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993.
- [2] R. Bellman, On a routing problem, *Quarterly of Applied Mathematics* 16(1) (1958) 87–90.
- [3] V. Chankong, Y.Y. Haimes, *Multiobjective Decision Making: Theory and Methodology*, North-Holland, New York, 1983.

- 858 [4] K. Dächert, T. Fleuren, K. Klamroth, DefiningPointAlgorithm,  
859 source code, MIT License, [https://github.com/kerstindaechert/](https://github.com/kerstindaechert/DefiningPointAlgorithm)  
860 [DefiningPointAlgorithm](https://github.com/kerstindaechert/DefiningPointAlgorithm), accessed 25 September 2026.
- 861 [5] K. Dächert, T. Fleuren, K. Klamroth, A simple, efficient and versatile  
862 objective space algorithm for multiobjective integer programming, *Mathe-*  
863 *matical Methods of Operations Research* 100 (2024) 351–384.
- 864 [6] M. Ehrgott, X. Gandibleux, Bound sets for biobjective combinatorial opti-  
865 mization problems, *Computers & Operations Research* 34(9) (2007) 2674–  
866 2694.
- 867 [7] A. Eusébio, J.R. Figueira, Finding non-dominated solutions in bi-objective  
868 integer network flow problems, *Computers & Operations Research* 36 (2009)  
869 2554–2564. doi:10.1016/j.cor.2008.11.001.
- 870 [8] L.R. Ford Jr., Network flow theory, Paper P-923, The RAND Corporation,  
871 Santa Monica, CA (1956).
- 872 [9] N. Forget, S.L. Gadegaard, K. Klamroth, L.R. Nielsen, A. Przybylski,  
873 Branch-and-bound and objective branching with three or more objectives,  
874 *Computers & Operations Research* 148 (2022) 106012.
- 875 [10] J. Gorski, K. Klamroth, S. Ruzika, Connectedness of efficient solutions  
876 in multiple objective combinatorial optimization, *Journal of Optimization*  
877 *Theory and Applications* 150(3) (2011) 475–497. doi:10.1007/s10957-011-  
878 9849-8.
- 879 [11] H.W. Hamacher, A note on  $K$  best network flows, *Annals of Operations*  
880 *Research* 57(1) (1995) 65–72. doi:10.1007/BF02099691.
- 881 [12] H.W. Hamacher, C.R. Pedersen, S. Ruzika, Multiple objective minimum  
882 cost flow problems: A review, *European Journal of Operational Research*  
883 176(3) (2007) 1404–1422.
- 884 [13] H. Isermann, Proper efficiency and the linear vector maximum problem,  
885 *Operations Research* 22 (1974) 189–191.
- 886 [14] G. Kirlik, S. Sayın, A new algorithm for generating all nondominated solu-  
887 tions of multiobjective discrete optimization problems, *European Journal*  
888 *of Operational Research* 232 (2014) 479–488.
- 889 [15] K. Klamroth, R. Lacour, D. Vanderpooten, On the representation of the  
890 search region in multi-objective optimization, *European Journal of Opera-*  
891 *tional Research* 245(3) (2015) 767–778.
- 892 [16] D. Klingman, A. Napier, J. Stutz, NETGEN: A program for generating  
893 large scale capacitated assignment, transportation, and minimum cost flow  
894 network problems, *Management Science* 20(5) (1974) 814–821.
- 895 [17] D. Könen, *Multi-Objective Combinatorial Optimization: Supportedness,*  
896 *Representations, and Integer Network Flows*, PhD thesis, University of  
897 Wuppertal, 2025.
- 898 [18] D. Könen, M. Stiglmayr, An output-polynomial time algorithm to de-  
899 termine all supported efficient solutions for multi-objective integer net-  
900 work flow problems, *Discrete Applied Mathematics* 376 (2025) 1–14.  
901 doi:10.1016/j.dam.2025.06.001.
- 902 [19] D. Könen, M. Stiglmayr, Output-sensitive complexity of multi-objective  
903 integer network flow problems, *Journal of Combinatorial Optimization* 51

- (2026), Article 14. doi:10.1007/s10878-025-01376-2.
- [20] D. Könen, D.R. Schmidt, C. Spisla, Finding all minimum cost flows and a faster algorithm for the  $K$  best flow problem, *Discrete Applied Mathematics* 321 (2022) 333–349. doi:10.1016/j.dam.2022.07.007.
- [21] D. Könen, L. Löhken, M. Stiglmayr, Representing the non-dominated set of multi-objective network problems by supported non-dominated points, arXiv:2607.11821 (2026).
- [22] D. Könen, L. Löhken, M. Stiglmayr, Supp\_Representations: test instances, [https://git.uni-wuppertal.de/loehken/supp\\_representations](https://git.uni-wuppertal.de/loehken/supp_representations), commit 166a4527. Accessed: 2026-10-04.
- [23] P. Kovács, Minimum-cost flow algorithms: an experimental evaluation, *Optimization Methods and Software* 30(1) (2015) 94–127.
- [24] E.L. Lawler, A procedure for computing the  $K$  best solutions to discrete optimization problems and its application to the shortest path problem, *Management Science* 18 (1972) 401–405.
- [25] H. Lee, P.S. Pulat, Bicriteria network flow problems: Integer case, *European Journal of Operational Research* 66(1) (1993) 148–157.
- [26] E.F. Moore, The shortest path through a maze, in: *Proceedings of the International Symposium on the Theory of Switching*, Harvard University Press, Cambridge, MA (1959) 285–292.
- [27] M. Ozlen, B.A. Burton, C.A.G. MacRae, An improved recursive algorithm for multi-objective integer programming, *Journal of Optimization Theory and Applications* 160 (2014) 470–482.
- [28] A. Raith, M. Ehrgott, A two-phase algorithm for the biobjective integer minimum cost flow problem, *Computers & Operations Research* 36(6) (2009) 1945–1954. doi:10.1016/j.cor.2008.06.008.
- [29] A. Raith, A. Sedeño-Noda, Finding extreme supported solutions of biobjective network flow problems: An enhanced parametric programming approach, *Computers & Operations Research* 82 (2017) 153–166.
- [30] G. Ruhe, Complexity results for multicriteria and parametric network flows using a pathological graph of Zadeh, *Zeitschrift für Operations Research* 32 (1988) 9–27.
- [31] A. Sedeño-Noda, Generalizing the Successive Shortest Path Algorithm to solve the Multi-objective Assignment Problem, Optimization Online, 2025. Also available as SSRN 5699606, doi:10.2139/ssrn.5699606.
- [32] A. Sedeño-Noda, J.J. Espino-Martín, On the  $K$  best integer network flows, *Computers & Operations Research* 40(2) (2013) 616–626.
- [33] A. Sedeño-Noda, C. González-Martín, The biobjective minimum cost flow problem, *European Journal of Operational Research* 124(3) (2000) 591–600.
- [34] A. Sedeño-Noda, C. González-Martín, An algorithm for the biobjective integer minimum cost flow problem, *Computers & Operations Research* 28(2) (2001) 139–156.
- [35] F. Sourd, O. Spanjaard, A multiobjective branch-and-bound framework: application to the biobjective spanning tree problem, *INFORMS Journal on Computing* 20(3) (2008) 472–484.
- [36] M. Sun, Finding integer efficient solutions for multiple objective network

- 950 programming problems, *Networks* 57 (2011) 362–375.
- 951 [37] M. Sun, A branch-and-bound algorithm for representative integer efficient  
952 solutions in multiple objective network programming problems, *Networks*  
953 62 (2013) 56–71.
- 954 [38] S. Tamby, D. Vanderpooten, Enumeration of the nondominated set of mul-  
955 tiobjective discrete optimization problems, *INFORMS Journal on Com-*  
956 *puting* 33(1) (2021) 72–85. doi:10.1287/ijoc.2020.0953.

# Supplementary Material for “Solving the Multi-objective Integer Minimum Cost Flow Problem via Circulations”

Antonio Sedeño-Noda

*Departamento de Matemáticas, Estadística e Investigación Operativa. Universidad de La  
Laguna, 38200 San Cristóbal de La Laguna, Santa Cruz de Tenerife, España.*

---

This document complements the computational study of the paper. Section S1 follows the algorithm on a small instance, Section S2 gives details of the implementation, Section S3 gives details on the instances, Section S4 reports the results for every class of instances, Section S5 reports the results on the instances of Könen, Löhken and Stiglmayr, and Section S6 the ablation study. Sections, tables and equations of the paper are referred to by their numbers in the paper.

## S1. A worked example

We follow CR on an instance with  $n = 6$  nodes,  $m = 12$  arcs and  $q = 3$  objectives, with a supply of 3 units at node 1, a demand of 3 units at node 6, zero lower bounds, and the capacities and costs of Table S1. Its feasible flows have 56 distinct objective vectors, of which five are nondominated (Table S2); we checked this by enumerating all the integer flows. The steps below are taken from the trace of the implementation. The residual arc that increases the flow on the arc  $(i, j)$  is written  $(i, j)^+$ , and the one that decreases it, from  $j$  to  $i$ , is written  $(i, j)^-$ ; partitions are numbered in order of creation, and a restriction of the form  $x_{ij} \leq k$  or  $x_{ij} \geq k$  is the bound decision of Section 4.2; the representatives of a partition are its lexicographic roots  $x^1$ ,  $x^2$  and  $x^3$  for the cyclic orders  $(1, 2, 3)$ ,  $(2, 3, 1)$  and  $(3, 1, 2)$ .

Table S1: Arcs of the example: arc, capacity and cost vector.

| $(i, j)$ | $u_{ij}$ | $c_{ij}$  | $(i, j)$ | $u_{ij}$ | $c_{ij}$  | $(i, j)$ | $u_{ij}$ | $c_{ij}$  |
|----------|----------|-----------|----------|----------|-----------|----------|----------|-----------|
| (1, 2)   | 2        | (7, 0, 8) | (2, 3)   | 1        | (9, 6, 8) | (5, 3)   | 2        | (5, 4, 3) |
| (1, 3)   | 1        | (6, 7, 2) | (3, 4)   | 2        | (5, 6, 0) | (5, 6)   | 2        | (7, 6, 2) |
| (1, 4)   | 2        | (2, 7, 8) | (4, 5)   | 2        | (0, 6, 3) | (6, 2)   | 2        | (0, 6, 6) |
| (1, 6)   | 1        | (8, 7, 9) | (4, 6)   | 2        | (9, 1, 0) | (6, 5)   | 1        | (0, 4, 0) |

Table S2: Minimal complete set of the example: nondominated vectors and the positive flows of the efficient flows.

| $c(x)$       | $x$                                                                                  |
|--------------|--------------------------------------------------------------------------------------|
| (26, 45, 35) | $x_{14} = 2, x_{16} = 1, x_{45} = 2, x_{56} = 2$                                     |
| (28, 34, 30) | $x_{14} = 2, x_{16} = 1, x_{45} = 1, x_{46} = 1, x_{56} = 1$                         |
| (30, 23, 25) | $x_{14} = 2, x_{16} = 1, x_{46} = 2$                                                 |
| (37, 40, 24) | $x_{13} = 1, x_{14} = 1, x_{16} = 1, x_{34} = 1, x_{45} = 1, x_{46} = 1, x_{56} = 1$ |
| (39, 29, 19) | $x_{13} = 1, x_{14} = 1, x_{16} = 1, x_{34} = 1, x_{46} = 2$                         |

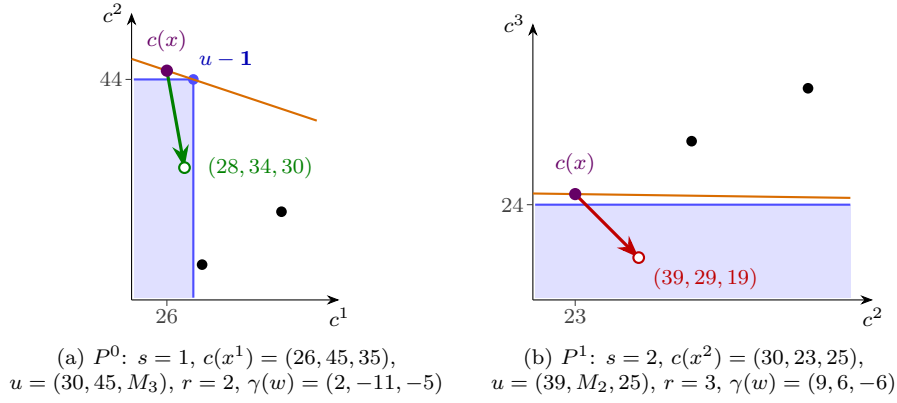


Figure S1: The two detections of the example, in the plane of the objectives  $s$  and  $r$  of each detection. Black dots: projections of the vectors of  $U$  at that moment; violet: the representative  $c(x^s)$  used in the detection; blue: the projection of the box  $u$ , whose side  $M_k - 1$  in a component  $k$  with  $u_k = M_k$  lies outside the window and is not drawn; orange: the line  $B_r y_s + B_s y_r = B_r c^s(x) + B_s c^r(x)$ . The arrow is  $\gamma(w)$  for the returned cycle  $w$ , and its end is  $c(x \oplus w)$ , shown with all its components: green if it lies in the box  $u$ , red otherwise. In (b) the end lies in the projection of the box but not in the box, since its first component exceeds  $u_1 - 1 = 38$ .

*Initialization.* The representatives of the root, its lexicographic roots for the three cyclic orders, have the vectors  $c(x^1) = (26, 45, 35)$ ,  $c(x^2) = (30, 23, 25)$  and  $c(x^3) = (39, 29, 19)$ , so that  $m(P^0) = (26, 23, 19)$ . The three vectors enter  $U$ , and they are nondominated, so that three of the five nondominated vectors are known before any detection; the warm-start flows add nothing new. The local upper bounds of these three vectors define the live boxes of  $P^0$ , with  $M = (94, 97, 80)$  as the upper corner of the search region. At every extraction, the arcs are first fixed as in Section 7.5 of the paper, with the slacks  $S(P) = (S_1(P), S_2(P), S_3(P))$  computed over the live boxes; at the root,  $S(P^0) = (67, 73, 60)$ , no reduced cost exceeds them, and no arc is fixed.

*Partition  $P^0$  (Figure S1(a)).* The first live box examined is  $u = (30, 45, M_3)$ . The closest representative is  $x^1$ , which exceeds  $u - 1$  only in the second component, so that  $s = 1$ , with  $B_1 = 30 - 1 - 26 = 3$ , and  $r = 2$ , with  $B_2 = 45 - 45 + 1 = 1$ . The detection with  $\theta = (c^1 + 3c^2, c^2)$  finds the cycle  $w = (5, 6)^-, (4, 6)^+, (4, 5)^-$ , with  $\gamma(w) = (2, -11, -5)$ : indeed  $B_2\gamma^1(w) + B_1\gamma^2(w) = 2 - 33 < 0$ . Probing adds  $c(x^1 \oplus w) = (28, 34, 30)$  to  $U$ . The three arcs of  $w$  lead to the same lexicographic root of the child, and the opening arc is  $(5, 6)^-$ , with  $x_{56}^1 = 2$ . The child  $P^1$ , with  $x_{56} \leq 1$ , has the representatives  $(28, 34, 30)$ ,  $(30, 23, 25)$  and  $(39, 29, 19)$ . The complementary restriction  $x_{56} \geq 2$  keeps  $x^1$ , and its other two representatives change to the dominated vectors  $(35, 44, 32)$  and  $(47, 51, 25)$ , so that its ideal point rises to  $(26, 44, 25)$ .

*Complement of  $P^0$  ( $x_{56} \geq 2$ ).* Only the box  $u = (28, M_2, 35)$  is live, so that  $S = (28 - 1 - 26, 97 - 1 - 44, 35 - 1 - 25) = (1, 52, 9)$ . Five residual arcs of  $G_P(x^1)$  have a reduced cost  $\bar{\gamma}^1$  greater than  $S_1 = 1$ :  $(2, 3)^+, (3, 4)^+, (4, 6)^+, (6, 2)^+$  and  $(6, 5)^+$ , with 10, 9, 2, 2 and 7. The flow of these five arcs, zero in  $x^1$ , is fixed to zero. After the reoptimization, the three representatives coincide with  $x^1$ , so that  $m(P) = c(x^1) = (26, 45, 35)$ , which is in  $U$ : no box is live, and the partition is discarded without any detection.

*Partition  $P^1$  ( $x_{56} \leq 1$ , Figure S1(b)).* Here  $S(P^1) = (65, 73, 60)$ , and no arc is fixed. For the box  $u = (39, M_2, 25)$ , the closest representative is  $x^2$ , with  $c(x^2) = (30, 23, 25)$ , which exceeds  $u - 1$  only in the third component:  $s = 2$ , with  $B_2 = 97 - 1 - 23 = 73$ , and  $r = 3$ , with  $B_3 = 1$ . The detection returns the cycle  $w = (1, 3)^+, (1, 4)^-, (3, 4)^+$ , with  $\gamma(w) = (9, 6, -6)$ , whose end  $(39, 29, 19)$  is already in  $U$ , so that probing adds nothing. The three candidate arcs lead to two different roots of the child for the order  $(2, 3, 1)$ , and the rule of Section 9 chooses  $(1, 3)^+$ , whose child root  $(39, 29, 19)$  has the most negative weight  $B_3\gamma^2 + B_2\gamma^3 = 6 - 438$ . The child  $P^2$ , with  $x_{13} \geq 1$ , has the representatives  $(37, 40, 24)$ , which is new and enters  $U$ , and  $(39, 29, 19)$  for the orders  $(2, 3, 1)$  and  $(3, 1, 2)$ ; the complement fixes  $x_{13} = 0$ .

*Complement of  $P^1$ .* This partition has  $x_{56} \leq 1$  and  $x_{13} = 0$ , and  $S(P) = (1, 73, 54)$ . Six residual arcs of  $G_P(x^1)$ , with  $c(x^1) = (28, 34, 30)$ , have  $\bar{\gamma}^1 > 1$ :  $(1, 6)^-$  and  $(5, 6)^-$ , with 3 and 2, give  $x_{16} \geq 1$  and  $x_{56} \geq 1$ , and  $(2, 3)^+, (3, 4)^+$ ,

$(6, 2)^+$  and  $(6, 5)^+$ , with 10, 9, 4 and 9, fix their flows to zero. Together with  $x_{56} \leq 1$ , the flow of  $(5, 6)$  is fixed to 1. After the reoptimization, the three representatives coincide with  $x^1$ , whose vector is already in  $U$ , and the partition is discarded.

*Partition  $P^2$ .* This partition has  $x_{56} \leq 1$  and  $x_{13} \geq 1$ , and  $S(P^2) = (1, 67, 5)$ . In  $G_P(x^1)$ , with  $c(x^1) = (37, 40, 24)$ , the arcs  $(1, 6)^-$ ,  $(5, 6)^-$ ,  $(2, 3)^+$ ,  $(5, 3)^+$ ,  $(6, 2)^+$  and  $(6, 5)^+$  have  $\bar{\gamma}^1$  equal to 3, 2, 19, 10, 4 and 9, all above  $S_1 = 1$ , which gives  $x_{16} = 1$ ,  $x_{56} = 1$  and zero flow on  $(2, 3)$ ,  $(5, 3)$ ,  $(6, 2)$  and  $(6, 5)$ ; in  $G_P(x^3)$ , the arcs  $(2, 3)^+$  and  $(6, 2)^+$ , with  $\bar{\gamma}^3$  equal to 8 and 7, above  $S_3 = 5$ , give two of these bounds again. After the reoptimization, the three representatives coincide with  $x^1$ , and the partition is discarded without any detection.

*End.* The heap is then empty, and  $U$  is the minimal complete set of Table S2. In total, 5 partitions are extracted and only 2 detections are run, both returning a cycle; the fixing discards the other three partitions, in each of which it makes the three representatives coincide, and no box needs a certificate. Three non-dominated vectors are representatives of the root,  $(37, 40, 24)$  is a representative of  $P^2$ , and  $(28, 34, 30)$  is found by probing. Figure S1 shows the two detections, each in the plane of the objectives  $s$  and  $r$  that it uses, as in Figure 1 of the paper.

## S2. Implementation details

This section complements Section 9 of the paper. A computation from scratch starts with every arc at the bound that minimizes its cost, lexicographically for vector costs, so that zero potentials are valid, and then removes the imbalances by shortest augmenting paths. All quantities are 64-bit integers. In a detection,  $B_s$  and  $B_r$  are at most  $\Delta = \max_r (M_r - y_r^I)$ , and every label is the weight of a walk with at most  $2nm$  arcs, so the labels are bounded in absolute value by  $4nm \Delta \max_{e,r} |c_e^r|$ , far below  $2^{63}$  in our instances.

The local upper bounds are kept sorted by decreasing first component, so that the pruning test can stop early; the sorted copy is rebuilt only when it is read, so that the insertions of a probing or of the representatives of a state are processed together. A new certificate replaces those whose boxes it contains, and a state keeps at most 64, discarding the oldest; a discarded certificate remains valid, and its boxes are examined again.

## S3. Instances

### S3.1. Own benchmark

The networks are generated with NETGEN [1], in the C translation of the original code included in GLPK 4.57 [2] (routine `glp_netgen`), modified only to write the arcs in the order in which NETGEN generates them. For an instance with  $q$  objectives,  $n$  nodes,  $m$  arcs, total supply  $S$  and index  $k \in \{0, 1, 2\}$ , the

parameters of NETGEN are: seed  $36581249 + 7919k + m + q$ ;  $n$  nodes;  $\lceil\sqrt{n}\rceil$  sources and  $\lceil\sqrt{n}\rceil$  sinks, of which  $\lfloor\lceil\sqrt{n}\rceil/2\rfloor$  are transshipment sources and as many transshipment sinks;  $m$  arcs; costs in  $[0, 100]$ ; total supply  $S$ ; no high-cost arcs; and all arcs capacitated, with capacities in  $[0, 50]$ . NETGEN gives each arc of the skeleton that guarantees feasibility a capacity equal to the supply of the source of its chain, even if it exceeds the maximum; this happens only in instance `q3_n20_m100_s100_0`, where three arcs have capacity 51. The first cost of each arc is the one generated by NETGEN; the remaining  $q - 1$  costs are added by our generator script, uniformly in  $[0, 100]$ , arc by arc in the order of the file, with Python's `random.Random` (Mersenne Twister) initialized with the same seed. NETGEN requires the total supply to exceed the number of sources; with  $S = 10$  this excludes  $n = 100$ , which would have 10 sources, and this is why the largest networks have 75 nodes. The block with  $q = 3$  and  $S = 100$  uses the same networks and seeds. The instances, with a manifest that records the parameters and the seed of every file, the generator script and the modified NETGEN code are available in the repository of the paper (<https://doi.org/10.5281/zenodo.23209259>).

Table S3 gives, for every combination of  $S$  and  $n$ , the networks and the median of  $|Y_N|$ .

### *S3.2. One-factor-at-a-time experiment*

The instances of the one-factor-at-a-time experiment of Section 10.3 are generated with PyNETGEN [3], a Python implementation of NETGEN, with 5 seeds per configuration (seeds  $12345671 + 7919k$ ,  $k = 0, \dots, 4$ ), no high-cost arcs and all arcs capacitated, with capacities in  $[0, 50]$ ; as in our benchmark, the costs  $2, \dots, q$  are uniform in the range of the first one. NETGEN may generate fewer arcs than requested: with 7 sources and 7 sinks, the instances with  $m = 150$  have between 138 and 148 arcs, and those with  $m = 300$  between 282 and 296. The configurations are described in the paper; the configuration ALL is labelled K in the file names, and the instances and the generator script (PyNETGEN 1.0.0) are available in the repository of the paper (<https://doi.org/10.5281/zenodo.23209259>).

## **S4. Results per class**

Table S4 gives the results of Table 1 of the paper, and of the additional block with  $q = 3$  and  $S = 100$ , for every class of instances, defined by  $q$ ,  $n$ ,  $m$  and the total supply  $S$ , with the same columns. Because of the numerical tolerances of CPLEX, the output of TV contains dominated or repeated vectors on 12 runs, 4 of them complete; they are removed before comparing fronts. After this filtering, the front of TV coincides with that of DPA on 466 of the 467 instances solved by both methods; the exception is instance `N4/instance40_120_14` of Könen et al. (Section S5): TV returns  $(8884, 10194, 9281)$ , which is dominated by  $(8883, 10194, 9281)$ , also in its output, and misses the nondominated vector  $(9858, 10080, 9058)$ , which DPA and CR find. Its count of 4920 vectors therefore agrees with the published one, but its front is incomplete.

Table S3: Own benchmark. For each total supply  $S$  and number of nodes  $n$ , the three numbers of arcs  $m$ , the number of sources, equal to the number of sinks, with the number of transshipment nodes among them in parentheses, and, for each  $q$ , the median of  $|Y_N|$  over the 9 instances of the cell (3 values of  $m$  and 3 seeds), with the number of instances with known  $|Y_N|$  in parentheses when not all. Costs are in  $[0, 100]$  and capacities in  $[0, 50]$ . The block with  $S = 100$  is generated only for  $q = 3$ .

| $S$ | $n$ | $m$           | Sources<br>(transship.) | Median $ Y_N $ |           |           |
|-----|-----|---------------|-------------------------|----------------|-----------|-----------|
|     |     |               |                         | $q = 3$        | $q = 4$   | $q = 5$   |
| 10  | 20  | 60, 80, 100   | 5 (2)                   | 140            | 356       | 247       |
|     | 30  | 90, 120, 150  | 6 (3)                   | 118            | 281       | 355 (8)   |
|     | 40  | 120, 160, 200 | 7 (3)                   | 84             | 420       | 434       |
|     | 50  | 150, 200, 250 | 8 (4)                   | 189            | 345       | 1389      |
|     | 75  | 225, 300, 375 | 9 (4)                   | 135            | 251       | 921 (8)   |
| 20  | 20  | 60, 80, 100   | 5 (2)                   | 638            | 3677      | 3624 (7)  |
|     | 30  | 90, 120, 150  | 6 (3)                   | 513            | 3417 (8)  | 2347 (5)  |
|     | 40  | 120, 160, 200 | 7 (3)                   | 501            | 3065      | 4894 (7)  |
|     | 50  | 150, 200, 250 | 8 (4)                   | 736            | 5724      | 7432 (4)  |
|     | 75  | 225, 300, 375 | 9 (4)                   | 787            | 1636      | 6396 (7)  |
| 30  | 20  | 60, 80, 100   | 5 (2)                   | 1130           | 5770 (8)  | 6736 (7)  |
|     | 30  | 90, 120, 150  | 6 (3)                   | 898            | 4469 (6)  | 6001 (3)  |
|     | 40  | 120, 160, 200 | 7 (3)                   | 2073           | 12166 (8) | 6172 (3)  |
|     | 50  | 150, 200, 250 | 8 (4)                   | 2152           | 7359 (7)  | 15364 (2) |
|     | 75  | 225, 300, 375 | 9 (4)                   | 1570           | 6270 (7)  | 5352 (1)  |
| 40  | 20  | 60, 80, 100   | 5 (2)                   | 1765           | 16390 (6) | 6911 (5)  |
|     | 30  | 90, 120, 150  | 6 (3)                   | 1739           | 6089 (4)  | —         |
|     | 40  | 120, 160, 200 | 7 (3)                   | 5618           | 11933 (2) | 10758 (1) |
|     | 50  | 150, 200, 250 | 8 (4)                   | 2595           | 16712 (2) | 8423 (1)  |
|     | 75  | 225, 300, 375 | 9 (4)                   | 5200           | 7094 (3)  | 8097 (1)  |
| 50  | 20  | 60, 80, 100   | 5 (2)                   | 4007           | 16571 (4) | 11342 (2) |
|     | 30  | 90, 120, 150  | 6 (3)                   | 3676           | 11286 (4) | 14210 (1) |
|     | 40  | 120, 160, 200 | 7 (3)                   | 6031           | 1317 (1)  | 19262 (1) |
|     | 50  | 150, 200, 250 | 8 (4)                   | 6926           | 18618 (2) | —         |
|     | 75  | 225, 300, 375 | 9 (4)                   | 9479           | 5940 (1)  | 4732 (1)  |
| 100 | 20  | 60, 80, 100   | 5 (2)                   | 22555          |           |           |
|     | 30  | 90, 120, 150  | 6 (3)                   | 18382          |           |           |
|     | 40  | 120, 160, 200 | 7 (3)                   | 13545 (7)      |           |           |
|     | 50  | 150, 200, 250 | 8 (4)                   | 31362 (8)      |           |           |
|     | 75  | 225, 300, 375 | 9 (4)                   | 40153 (5)      |           |           |

Table S4: Results per class with a limit of 3600 seconds: class ( $q$ ,  $n$ ,  $m$ , total supply  $S$ ), number of instances, solved instances, shifted geometric mean of the CPU time in seconds (best in bold), median of the known values of  $|Y_N|$  and geometric mean of the number of partitions processed by CR per nondominated vector.

| $q$ | $n$ | $m$ | $S$ | # | Solved |     |    | Time (s)    |              |        | $ Y_N $   | Part./<br>vec. |
|-----|-----|-----|-----|---|--------|-----|----|-------------|--------------|--------|-----------|----------------|
|     |     |     |     |   | CR     | DPA | TV | CR          | DPA          | TV     |           |                |
| 3   | 20  | 60  | 10  | 3 | 3      | 3   | 3  | <b>0.0</b>  | 0.8          | 1.3    | 55        | 2              |
| 3   | 20  | 60  | 20  | 3 | 3      | 3   | 3  | <b>0.2</b>  | 6.3          | 10.0   | 371       | 7              |
| 3   | 20  | 60  | 30  | 3 | 3      | 3   | 3  | <b>0.5</b>  | 11.5         | 21.2   | 501       | 8              |
| 3   | 20  | 60  | 40  | 3 | 3      | 3   | 3  | <b>3.3</b>  | 23.1         | 37.1   | 1573      | 21             |
| 3   | 20  | 60  | 50  | 3 | 3      | 3   | 3  | <b>15.2</b> | 50.1         | 92.8   | 4301      | 36             |
| 3   | 20  | 60  | 100 | 3 | 3      | 3   | 3  | 600.1       | <b>245.8</b> | 863.0  | 13495     | 104            |
| 3   | 20  | 80  | 10  | 3 | 3      | 3   | 3  | <b>0.0</b>  | 3.3          | 5.1    | 140       | 4              |
| 3   | 20  | 80  | 20  | 3 | 3      | 3   | 3  | <b>0.6</b>  | 12.3         | 20.7   | 638       | 9              |
| 3   | 20  | 80  | 30  | 3 | 3      | 3   | 3  | <b>1.6</b>  | 27.0         | 40.6   | 1079      | 13             |
| 3   | 20  | 80  | 40  | 3 | 3      | 3   | 3  | <b>6.5</b>  | 34.9         | 58.5   | 1765      | 21             |
| 3   | 20  | 80  | 50  | 3 | 3      | 3   | 3  | <b>10.5</b> | 51.3         | 82.8   | 2371      | 27             |
| 3   | 20  | 80  | 100 | 3 | 2      | 3   | 3  | 1457.2      | <b>323.1</b> | 1361.5 | 20662     | 194            |
| 3   | 20  | 100 | 10  | 3 | 3      | 3   | 3  | <b>0.1</b>  | 5.3          | 9.1    | 186       | 6              |
| 3   | 20  | 100 | 20  | 3 | 3      | 3   | 3  | <b>2.6</b>  | 33.3         | 63.0   | 1102      | 20             |
| 3   | 20  | 100 | 30  | 3 | 3      | 3   | 3  | <b>11.1</b> | 72.5         | 112.5  | 1663      | 34             |
| 3   | 20  | 100 | 40  | 3 | 3      | 3   | 3  | 193.9       | <b>137.7</b> | 268.1  | 5209      | 159            |
| 3   | 20  | 100 | 50  | 3 | 3      | 3   | 3  | 380.1       | <b>190.5</b> | 361.6  | 6766      | 211            |
| 3   | 20  | 100 | 100 | 3 | 0      | 3   | 1  | 3600.0      | <b>975.1</b> | 3406.5 | 27782     | —              |
| 3   | 30  | 90  | 10  | 3 | 3      | 3   | 3  | <b>0.0</b>  | 1.6          | 2.4    | 28        | 2              |
| 3   | 30  | 90  | 20  | 3 | 3      | 3   | 3  | <b>0.4</b>  | 10.6         | 16.8   | 460       | 6              |
| 3   | 30  | 90  | 30  | 3 | 3      | 3   | 3  | <b>0.8</b>  | 12.7         | 17.6   | 645       | 8              |
| 3   | 30  | 90  | 40  | 3 | 3      | 3   | 3  | <b>3.4</b>  | 19.1         | 30.6   | 1221      | 21             |
| 3   | 30  | 90  | 50  | 3 | 3      | 3   | 3  | <b>18.7</b> | 59.9         | 97.3   | 3215      | 35             |
| 3   | 30  | 90  | 100 | 3 | 3      | 3   | 1  | 727.3       | <b>294.2</b> | 1259.8 | 18382     | 142            |
| 3   | 30  | 120 | 10  | 3 | 3      | 3   | 3  | <b>0.1</b>  | 2.6          | 3.9    | 54        | 5              |
| 3   | 30  | 120 | 20  | 3 | 3      | 3   | 3  | <b>1.7</b>  | 19.0         | 28.9   | 543       | 18             |
| 3   | 30  | 120 | 30  | 3 | 3      | 3   | 3  | <b>9.3</b>  | 42.2         | 60.5   | 898       | 40             |
| 3   | 30  | 120 | 40  | 3 | 3      | 3   | 3  | <b>42.7</b> | 87.4         | 158.8  | 2178      | 62             |
| 3   | 30  | 120 | 50  | 3 | 3      | 3   | 3  | 242.9       | <b>208.4</b> | 401.1  | 3676      | 133            |
| 3   | 30  | 120 | 100 | 3 | 1      | 3   | 1  | 3275.9      | <b>642.9</b> | 2705.2 | 16474     | 431            |
| 3   | 30  | 150 | 10  | 3 | 3      | 3   | 3  | <b>0.7</b>  | 11.6         | 17.4   | 190       | 9              |
| 3   | 30  | 150 | 20  | 3 | 3      | 3   | 3  | <b>7.2</b>  | 29.2         | 47.9   | 960       | 35             |
| 3   | 30  | 150 | 30  | 3 | 3      | 3   | 3  | <b>33.1</b> | 71.0         | 107.2  | 2272      | 71             |
| 3   | 30  | 150 | 40  | 3 | 2      | 3   | 3  | 388.1       | <b>194.9</b> | 397.7  | 7022      | 153            |
| 3   | 30  | 150 | 50  | 3 | 1      | 3   | 3  | 443.4       | <b>191.5</b> | 368.4  | 8312      | 29             |
| 3   | 30  | 150 | 100 | 3 | 0      | 3   | 1  | 3600.0      | <b>916.8</b> | 2471.3 | 41207     | —              |
| 3   | 40  | 120 | 10  | 3 | 3      | 3   | 3  | <b>0.0</b>  | 2.7          | 4.0    | 61        | 3              |
| 3   | 40  | 120 | 20  | 3 | 3      | 3   | 3  | <b>0.4</b>  | 7.0          | 11.0   | 188       | 6              |
| 3   | 40  | 120 | 30  | 3 | 3      | 3   | 3  | <b>4.0</b>  | 22.6         | 32.4   | 261       | 19             |
| 3   | 40  | 120 | 40  | 3 | 3      | 3   | 3  | <b>13.6</b> | 62.3         | 114.1  | 1762      | 26             |
| 3   | 40  | 120 | 50  | 3 | 3      | 3   | 3  | <b>24.7</b> | 71.9         | 135.4  | 1498      | 41             |
| 3   | 40  | 120 | 100 | 3 | 2      | 3   | 2  | 572.6       | <b>321.5</b> | 634.3  | 6861      | 140            |
| 3   | 40  | 160 | 10  | 3 | 3      | 3   | 3  | <b>0.0</b>  | 2.4          | 3.3    | 65        | 4              |
| 3   | 40  | 160 | 20  | 3 | 3      | 3   | 3  | <b>0.9</b>  | 9.4          | 14.4   | 365       | 11             |
| 3   | 40  | 160 | 30  | 3 | 3      | 3   | 3  | <b>13.0</b> | 46.0         | 67.0   | 1479      | 39             |
| 3   | 40  | 160 | 40  | 3 | 3      | 3   | 3  | <b>37.1</b> | 73.9         | 108.1  | 1904      | 75             |
| 3   | 40  | 160 | 50  | 3 | 2      | 3   | 3  | 165.2       | <b>133.0</b> | 245.2  | 3544      | 69             |
| 3   | 40  | 160 | 100 | 3 | 1      | 2   | 2  | 1253.2      | <b>746.0</b> | 1353.5 | 10972 (2) | 31             |
| 3   | 40  | 200 | 10  | 3 | 3      | 3   | 3  | <b>1.0</b>  | 28.7         | 45.4   | 257       | 12             |

| $q$ | $n$ | $m$ | $S$ | # | CR | DPA | TV | CR            | DPA           | TV            | $ Y_N $   | vec. |
|-----|-----|-----|-----|---|----|-----|----|---------------|---------------|---------------|-----------|------|
| 3   | 40  | 200 | 20  | 3 | 3  | 3   | 3  | <b>39.6</b>   | 153.3         | 237.7         | 2283      | 63   |
| 3   | 40  | 200 | 30  | 3 | 3  | 3   | 3  | <b>172.2</b>  | 242.6         | 397.3         | 2440      | 130  |
| 3   | 40  | 200 | 40  | 3 | 2  | 3   | 3  | 1765.7        | <b>560.3</b>  | 1451.3        | 11880     | 285  |
| 3   | 40  | 200 | 50  | 3 | 1  | 3   | 2  | 2540.0        | <b>617.3</b>  | 1411.0        | 9076      | 510  |
| 3   | 40  | 200 | 100 | 3 | 0  | 2   | 0  | 3600.0        | <b>2702.0</b> | 3600.0        | 66290 (2) | –    |
| 3   | 50  | 150 | 10  | 3 | 3  | 3   | 3  | <b>0.1</b>    | 5.0           | 7.8           | 58        | 4    |
| 3   | 50  | 150 | 20  | 3 | 3  | 3   | 3  | <b>1.0</b>    | 17.4          | 28.6          | 586       | 9    |
| 3   | 50  | 150 | 30  | 3 | 3  | 3   | 3  | <b>2.4</b>    | 33.2          | 44.6          | 692       | 12   |
| 3   | 50  | 150 | 40  | 3 | 3  | 3   | 3  | <b>18.6</b>   | 98.8          | 153.2         | 2088      | 31   |
| 3   | 50  | 150 | 50  | 3 | 3  | 3   | 3  | <b>74.9</b>   | 160.1         | 320.0         | 4644      | 43   |
| 3   | 50  | 150 | 100 | 3 | 1  | 3   | 1  | 2005.7        | <b>441.2</b>  | 2303.7        | 12798     | 123  |
| 3   | 50  | 200 | 10  | 3 | 3  | 3   | 3  | <b>0.4</b>    | 17.1          | 28.5          | 283       | 7    |
| 3   | 50  | 200 | 20  | 3 | 3  | 3   | 3  | <b>3.5</b>    | 46.3          | 77.6          | 736       | 16   |
| 3   | 50  | 200 | 30  | 3 | 3  | 3   | 3  | <b>54.3</b>   | 157.9         | 246.5         | 2493      | 59   |
| 3   | 50  | 200 | 40  | 3 | 3  | 3   | 3  | <b>41.4</b>   | 124.9         | 188.1         | 2018      | 58   |
| 3   | 50  | 200 | 50  | 3 | 2  | 3   | 3  | 593.6         | <b>369.1</b>  | 748.5         | 5994      | 101  |
| 3   | 50  | 200 | 100 | 3 | 0  | 3   | 1  | 3600.0        | <b>1790.4</b> | 3502.5        | 43592     | –    |
| 3   | 50  | 250 | 10  | 3 | 3  | 3   | 3  | <b>0.5</b>    | 17.9          | 29.1          | 253       | 8    |
| 3   | 50  | 250 | 20  | 3 | 3  | 3   | 3  | <b>20.4</b>   | 97.9          | 161.4         | 1375      | 41   |
| 3   | 50  | 250 | 30  | 3 | 3  | 3   | 3  | <b>179.1</b>  | 206.4         | 344.5         | 3803      | 111  |
| 3   | 50  | 250 | 40  | 3 | 2  | 3   | 3  | 1640.4        | <b>392.6</b>  | 682.2         | 8386      | 456  |
| 3   | 50  | 250 | 50  | 3 | 0  | 3   | 3  | 3600.0        | <b>760.9</b>  | 2048.6        | 11064     | –    |
| 3   | 50  | 250 | 100 | 3 | 0  | 2   | 0  | 3600.0        | <b>2536.3</b> | 3600.0        | 42300 (2) | –    |
| 3   | 75  | 225 | 10  | 3 | 3  | 3   | 3  | <b>0.1</b>    | 3.1           | 4.7           | 35        | 2    |
| 3   | 75  | 225 | 20  | 3 | 3  | 3   | 3  | <b>3.3</b>    | 32.4          | 50.7          | 584       | 16   |
| 3   | 75  | 225 | 30  | 3 | 3  | 3   | 3  | <b>11.7</b>   | 84.7          | 121.7         | 1262      | 22   |
| 3   | 75  | 225 | 40  | 3 | 3  | 3   | 3  | <b>106.1</b>  | 217.8         | 405.8         | 3314      | 52   |
| 3   | 75  | 225 | 50  | 3 | 2  | 3   | 3  | 383.1         | <b>294.0</b>  | 553.2         | 3672      | 70   |
| 3   | 75  | 225 | 100 | 3 | 1  | 2   | 1  | <b>710.1</b>  | 821.4         | 1348.2        | 20932 (2) | 34   |
| 3   | 75  | 300 | 10  | 3 | 3  | 3   | 3  | <b>0.3</b>    | 11.0          | 15.6          | 112       | 8    |
| 3   | 75  | 300 | 20  | 3 | 3  | 3   | 3  | <b>6.3</b>    | 45.7          | 61.1          | 736       | 25   |
| 3   | 75  | 300 | 30  | 3 | 3  | 3   | 3  | <b>107.0</b>  | 117.7         | 175.0         | 1570      | 126  |
| 3   | 75  | 300 | 40  | 3 | 1  | 3   | 3  | 1499.8        | <b>319.1</b>  | 526.0         | 3695      | 158  |
| 3   | 75  | 300 | 50  | 3 | 1  | 3   | 3  | 1265.3        | <b>431.5</b>  | 735.3         | 9479      | 138  |
| 3   | 75  | 300 | 100 | 3 | 0  | 2   | 1  | 3600.0        | <b>1911.4</b> | 2882.2        | 32336 (2) | –    |
| 3   | 75  | 375 | 10  | 3 | 3  | 3   | 3  | <b>1.4</b>    | 30.8          | 48.5          | 249       | 11   |
| 3   | 75  | 375 | 20  | 3 | 3  | 3   | 3  | <b>87.6</b>   | 153.5         | 240.8         | 1581      | 89   |
| 3   | 75  | 375 | 30  | 3 | 2  | 3   | 3  | 921.4         | <b>283.2</b>  | 521.1         | 2816      | 293  |
| 3   | 75  | 375 | 40  | 3 | 2  | 3   | 3  | 2984.9        | <b>580.4</b>  | 1123.3        | 7371      | 544  |
| 3   | 75  | 375 | 50  | 3 | 0  | 3   | 2  | 3600.0        | <b>1045.0</b> | 2252.0        | 11938     | –    |
| 3   | 75  | 375 | 100 | 3 | 0  | 1   | 0  | 3600.0        | <b>3284.2</b> | 3600.0        | 48541 (1) | –    |
| 4   | 20  | 60  | 10  | 3 | 3  | 3   | 3  | <b>0.1</b>    | 9.8           | 16.9          | 299       | 3    |
| 4   | 20  | 60  | 20  | 3 | 3  | 3   | 3  | <b>1.0</b>    | 34.1          | 75.2          | 472       | 4    |
| 4   | 20  | 60  | 30  | 3 | 3  | 3   | 3  | <b>3.2</b>    | 89.8          | 306.0         | 1178      | 6    |
| 4   | 20  | 60  | 40  | 3 | 3  | 3   | 2  | <b>12.7</b>   | 203.3         | 806.1         | 2623      | 9    |
| 4   | 20  | 60  | 50  | 3 | 3  | 3   | 0  | <b>193.1</b>  | 940.0         | 3600.0        | 7452      | 13   |
| 4   | 20  | 80  | 10  | 3 | 3  | 3   | 3  | <b>0.4</b>    | 31.8          | 54.2          | 553       | 4    |
| 4   | 20  | 80  | 20  | 3 | 3  | 3   | 1  | <b>29.8</b>   | 430.0         | 3392.8        | 4142      | 11   |
| 4   | 20  | 80  | 30  | 3 | 3  | 3   | 0  | <b>306.6</b>  | 1278.6        | 3600.0        | 12631     | 22   |
| 4   | 20  | 80  | 40  | 3 | 1  | 2   | 0  | <b>3017.1</b> | 3166.2        | 3600.0        | 29988 (2) | 34   |
| 4   | 20  | 80  | 50  | 3 | 0  | 1   | 0  | 3600.0        | <b>2839.0</b> | 3600.0        | 25690 (1) | –    |
| 4   | 20  | 100 | 10  | 3 | 3  | 3   | 3  | <b>1.2</b>    | 60.6          | 116.9         | 583       | 6    |
| 4   | 20  | 100 | 20  | 3 | 3  | 3   | 2  | <b>107.5</b>  | 595.1         | 2288.3        | 3799      | 28   |
| 4   | 20  | 100 | 30  | 3 | 1  | 2   | 1  | <b>617.1</b>  | 1438.6        | 2629.8        | 12390 (2) | 19   |
| 4   | 20  | 100 | 40  | 3 | 0  | 1   | 0  | 3600.0        | <b>3467.5</b> | 3600.0        | 26501 (1) | –    |
| 4   | 20  | 100 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | –         | –    |

| $q$ | $n$ | $m$ | $S$ | # | CR | DPA | TV | CR            | DPA           | TV            | $ Y_N $   | vec. |
|-----|-----|-----|-----|---|----|-----|----|---------------|---------------|---------------|-----------|------|
| 4   | 30  | 90  | 10  | 3 | 3  | 3   | 3  | <b>0.0</b>    | 3.8           | 5.5           | 53        | 2    |
| 4   | 30  | 90  | 20  | 3 | 3  | 3   | 3  | <b>3.1</b>    | 47.0          | 117.3         | 368       | 8    |
| 4   | 30  | 90  | 30  | 3 | 3  | 3   | 2  | <b>8.4</b>    | 143.7         | 376.9         | 1327      | 9    |
| 4   | 30  | 90  | 40  | 3 | 3  | 3   | 2  | <b>16.3</b>   | 148.1         | 347.6         | 1329      | 17   |
| 4   | 30  | 90  | 50  | 3 | 3  | 3   | 2  | <b>52.2</b>   | 365.6         | 712.9         | 2485      | 18   |
| 4   | 30  | 120 | 10  | 3 | 3  | 3   | 3  | <b>0.7</b>    | 54.6          | 93.9          | 281       | 5    |
| 4   | 30  | 120 | 20  | 3 | 3  | 3   | 2  | <b>40.0</b>   | 462.0         | 1779.5        | 3544      | 15   |
| 4   | 30  | 120 | 30  | 3 | 3  | 3   | 1  | <b>77.0</b>   | 671.3         | 1781.4        | 7416      | 21   |
| 4   | 30  | 120 | 40  | 3 | 1  | 1   | 0  | 3110.0        | <b>2593.4</b> | 3600.0        | 12701 (1) | 151  |
| 4   | 30  | 120 | 50  | 3 | 1  | 0   | 0  | <b>2327.8</b> | 3600.0        | 3600.0        | 20087 (1) | 15   |
| 4   | 30  | 150 | 10  | 3 | 3  | 3   | 2  | <b>9.0</b>    | 218.2         | 463.8         | 921       | 12   |
| 4   | 30  | 150 | 20  | 3 | 2  | 2   | 0  | <b>1381.3</b> | 2016.7        | 3600.0        | 11356 (2) | 63   |
| 4   | 30  | 150 | 30  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 30  | 150 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 30  | 150 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 40  | 120 | 10  | 3 | 3  | 3   | 3  | <b>0.1</b>    | 13.2          | 17.4          | 137       | 3    |
| 4   | 40  | 120 | 20  | 3 | 3  | 3   | 2  | <b>8.7</b>    | 191.9         | 377.1         | 1263      | 9    |
| 4   | 40  | 120 | 30  | 3 | 3  | 3   | 1  | <b>202.7</b>  | 938.5         | 1475.7        | 17161     | 22   |
| 4   | 40  | 120 | 40  | 3 | 2  | 2   | 1  | <b>208.2</b>  | 596.3         | 757.8         | 11933 (2) | 16   |
| 4   | 40  | 120 | 50  | 3 | 1  | 1   | 1  | <b>429.9</b>  | 1277.1        | 1652.8        | 1317 (1)  | 12   |
| 4   | 40  | 160 | 10  | 3 | 3  | 3   | 3  | <b>0.7</b>    | 73.8          | 124.1         | 420       | 6    |
| 4   | 40  | 160 | 20  | 3 | 3  | 3   | 3  | <b>34.5</b>   | 487.0         | 1558.5        | 3065      | 22   |
| 4   | 40  | 160 | 30  | 3 | 3  | 3   | 0  | <b>741.7</b>  | 1723.8        | 3600.0        | 9712      | 50   |
| 4   | 40  | 160 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 40  | 160 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 40  | 200 | 10  | 3 | 3  | 3   | 3  | <b>4.4</b>    | 236.2         | 488.0         | 1288      | 10   |
| 4   | 40  | 200 | 20  | 3 | 3  | 3   | 1  | <b>218.7</b>  | 1107.8        | 2907.3        | 8251      | 34   |
| 4   | 40  | 200 | 30  | 3 | 1  | 2   | 0  | <b>976.6</b>  | 1889.4        | 3600.0        | 11112 (2) | 21   |
| 4   | 40  | 200 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 40  | 200 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 50  | 150 | 10  | 3 | 3  | 3   | 3  | <b>0.6</b>    | 42.5          | 64.7          | 151       | 4    |
| 4   | 50  | 150 | 20  | 3 | 3  | 3   | 1  | <b>52.9</b>   | 905.5         | 1851.6        | 6371      | 13   |
| 4   | 50  | 150 | 30  | 3 | 3  | 2   | 2  | <b>83.8</b>   | 711.5         | 1357.8        | 3156      | 18   |
| 4   | 50  | 150 | 40  | 3 | 2  | 2   | 0  | <b>1829.8</b> | 2640.0        | 3600.0        | 16712 (2) | 43   |
| 4   | 50  | 150 | 50  | 3 | 1  | 1   | 0  | 3384.2        | <b>3381.1</b> | 3600.0        | 24185 (1) | 44   |
| 4   | 50  | 200 | 10  | 3 | 3  | 3   | 3  | <b>0.4</b>    | 35.8          | 47.1          | 315       | 5    |
| 4   | 50  | 200 | 20  | 3 | 3  | 3   | 2  | <b>73.0</b>   | 850.7         | 2503.8        | 3677      | 25   |
| 4   | 50  | 200 | 30  | 3 | 3  | 2   | 0  | <b>1518.4</b> | 2599.2        | 3600.0        | 15352     | 84   |
| 4   | 50  | 200 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 50  | 200 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 50  | 250 | 10  | 3 | 3  | 3   | 3  | <b>2.5</b>    | 141.0         | 211.1         | 583       | 10   |
| 4   | 50  | 250 | 20  | 3 | 3  | 3   | 1  | <b>203.5</b>  | 1321.1        | 2522.3        | 6259      | 35   |
| 4   | 50  | 250 | 30  | 3 | 1  | 1   | 0  | <b>1659.6</b> | 2430.9        | 3600.0        | 7359 (1)  | 35   |
| 4   | 50  | 250 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 50  | 250 | 50  | 3 | 1  | 1   | 0  | 3300.8        | <b>2928.3</b> | 3600.0        | 13050 (1) | 109  |
| 4   | 75  | 225 | 10  | 3 | 3  | 3   | 3  | <b>1.2</b>    | 86.7          | 125.8         | 407       | 5    |
| 4   | 75  | 225 | 20  | 3 | 3  | 3   | 3  | <b>10.6</b>   | 266.9         | 590.2         | 1636      | 10   |
| 4   | 75  | 225 | 30  | 3 | 3  | 3   | 2  | <b>87.3</b>   | 873.3         | 1639.2        | 3899      | 21   |
| 4   | 75  | 225 | 40  | 3 | 2  | 2   | 1  | <b>427.0</b>  | 1329.2        | 1518.0        | 8336 (2)  | 30   |
| 4   | 75  | 225 | 50  | 3 | 1  | 1   | 0  | <b>1955.7</b> | 2481.3        | 3600.0        | 5940 (1)  | 108  |
| 4   | 75  | 300 | 10  | 3 | 3  | 3   | 3  | <b>0.4</b>    | 41.8          | 54.9          | 157       | 5    |
| 4   | 75  | 300 | 20  | 3 | 3  | 3   | 3  | <b>15.5</b>   | 443.2         | 744.6         | 1472      | 13   |
| 4   | 75  | 300 | 30  | 3 | 3  | 2   | 0  | <b>816.9</b>  | 2537.4        | 3600.0        | 10277     | 48   |
| 4   | 75  | 300 | 40  | 3 | 1  | 1   | 0  | <b>1328.6</b> | 2581.8        | 3600.0        | 7094 (1)  | 17   |
| 4   | 75  | 300 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 75  | 375 | 10  | 3 | 3  | 3   | 3  | <b>2.7</b>    | 117.5         | 190.6         | 321       | 12   |

| $q$ | $n$ | $m$ | $S$ | # | CR | DPA | TV | CR            | DPA           | TV            | $ Y_N $   | vec. |
|-----|-----|-----|-----|---|----|-----|----|---------------|---------------|---------------|-----------|------|
| 4   | 75  | 375 | 20  | 3 | 3  | 3   | 2  | <b>155.3</b>  | 1068.8        | 1768.9        | 2838      | 46   |
| 4   | 75  | 375 | 30  | 3 | 1  | 1   | 0  | <b>1966.6</b> | 2650.8        | 3600.0        | 5054 (1)  | 96   |
| 4   | 75  | 375 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 4   | 75  | 375 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 20  | 60  | 10  | 3 | 3  | 3   | 3  | <b>0.2</b>    | 23.4          | 44.0          | 247       | 3    |
| 5   | 20  | 60  | 20  | 3 | 3  | 3   | 1  | <b>11.9</b>   | 229.7         | 765.6         | 2989      | 5    |
| 5   | 20  | 60  | 30  | 3 | 3  | 2   | 1  | <b>150.8</b>  | 1295.6        | 3153.2        | 11922     | 7    |
| 5   | 20  | 60  | 40  | 3 | 2  | 1   | 1  | <b>212.7</b>  | 776.2         | 1040.1        | 9700 (2)  | 6    |
| 5   | 20  | 60  | 50  | 3 | 2  | 1   | 1  | <b>496.6</b>  | 1436.1        | 3594.9        | 11342 (2) | 9    |
| 5   | 20  | 80  | 10  | 3 | 3  | 3   | 2  | <b>3.2</b>    | 85.7          | 122.7         | 269       | 4    |
| 5   | 20  | 80  | 20  | 3 | 2  | 2   | 0  | <b>271.4</b>  | 1734.9        | 3600.0        | 3742 (2)  | 8    |
| 5   | 20  | 80  | 30  | 3 | 2  | 1   | 0  | <b>520.6</b>  | 1822.2        | 3600.0        | 7667 (2)  | 10   |
| 5   | 20  | 80  | 40  | 3 | 1  | 1   | 0  | <b>1560.1</b> | 2762.9        | 3600.0        | 6911 (1)  | 11   |
| 5   | 20  | 80  | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 20  | 100 | 10  | 3 | 3  | 3   | 2  | <b>1.2</b>    | 73.9          | 213.4         | 224       | 5    |
| 5   | 20  | 100 | 20  | 3 | 2  | 1   | 1  | <b>418.2</b>  | 1571.4        | 3031.5        | 7772 (2)  | 11   |
| 5   | 20  | 100 | 30  | 3 | 2  | 2   | 0  | <b>409.6</b>  | 1771.3        | 3600.0        | 5160 (2)  | 10   |
| 5   | 20  | 100 | 40  | 3 | 2  | 1   | 0  | <b>912.3</b>  | 2691.2        | 3600.0        | 10044 (2) | 17   |
| 5   | 20  | 100 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 30  | 90  | 10  | 3 | 3  | 3   | 3  | <b>0.2</b>    | 44.0          | 63.6          | 268       | 3    |
| 5   | 30  | 90  | 20  | 3 | 3  | 3   | 1  | <b>23.1</b>   | 642.0         | 3072.1        | 2347      | 6    |
| 5   | 30  | 90  | 30  | 3 | 3  | 3   | 0  | <b>234.7</b>  | 1844.5        | 3600.0        | 6001      | 9    |
| 5   | 30  | 90  | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 30  | 90  | 50  | 3 | 1  | 0   | 0  | <b>2605.1</b> | 3600.0        | 3600.0        | 14210 (1) | 12   |
| 5   | 30  | 120 | 10  | 3 | 3  | 3   | 2  | <b>1.3</b>    | 142.7         | 347.8         | 371       | 4    |
| 5   | 30  | 120 | 20  | 3 | 2  | 2   | 1  | <b>180.4</b>  | 1250.1        | 3553.2        | 2928 (2)  | 10   |
| 5   | 30  | 120 | 30  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 30  | 120 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 30  | 120 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 30  | 150 | 10  | 3 | 2  | 1   | 1  | <b>220.8</b>  | 1649.2        | 2902.3        | 4102 (2)  | 11   |
| 5   | 30  | 150 | 20  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 30  | 150 | 30  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 30  | 150 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 30  | 150 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 40  | 120 | 10  | 3 | 3  | 3   | 3  | <b>0.4</b>    | 68.5          | 127.2         | 172       | 4    |
| 5   | 40  | 120 | 20  | 3 | 3  | 3   | 1  | <b>31.5</b>   | 1049.6        | 2100.5        | 3173      | 7    |
| 5   | 40  | 120 | 30  | 3 | 3  | 2   | 0  | <b>322.5</b>  | 2041.4        | 3600.0        | 6172      | 12   |
| 5   | 40  | 120 | 40  | 3 | 1  | 0   | 0  | <b>2324.7</b> | 3600.0        | 3600.0        | 10758 (1) | 19   |
| 5   | 40  | 120 | 50  | 3 | 1  | 0   | 0  | <b>3176.7</b> | 3600.0        | 3600.0        | 19262 (1) | 12   |
| 5   | 40  | 160 | 10  | 3 | 3  | 3   | 2  | <b>2.7</b>    | 254.0         | 634.5         | 407       | 5    |
| 5   | 40  | 160 | 20  | 3 | 3  | 2   | 0  | <b>281.9</b>  | 1912.5        | 3600.0        | 4894      | 16   |
| 5   | 40  | 160 | 30  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 40  | 160 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 40  | 160 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 40  | 200 | 10  | 3 | 3  | 3   | 3  | <b>7.2</b>    | 506.0         | 1667.4        | 1061      | 9    |
| 5   | 40  | 200 | 20  | 3 | 1  | 1   | 0  | <b>1570.3</b> | 3208.6        | 3600.0        | 6216 (1)  | 17   |
| 5   | 40  | 200 | 30  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 40  | 200 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 40  | 200 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 50  | 150 | 10  | 3 | 3  | 3   | 3  | <b>2.2</b>    | 309.0         | 1043.3        | 631       | 5    |
| 5   | 50  | 150 | 20  | 3 | 3  | 0   | 0  | <b>723.5</b>  | 3600.0        | 3600.0        | 7581      | 12   |
| 5   | 50  | 150 | 30  | 3 | 1  | 0   | 0  | <b>3562.7</b> | 3600.0        | 3600.0        | 15273 (1) | 33   |
| 5   | 50  | 150 | 40  | 3 | 1  | 1   | 0  | <b>1953.3</b> | 3349.1        | 3600.0        | 8423 (1)  | 14   |
| 5   | 50  | 150 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 50  | 200 | 10  | 3 | 3  | 2   | 1  | <b>17.6</b>   | 659.0         | 1437.2        | 1389      | 7    |
| 5   | 50  | 200 | 20  | 3 | 1  | 0   | 0  | <b>2006.5</b> | 3600.0        | 3600.0        | 7284 (1)  | 21   |

| $q$ | $n$ | $m$ | $S$ | # | CR | DPA | TV | CR            | DPA           | TV            | $ Y_N $   | vec. |
|-----|-----|-----|-----|---|----|-----|----|---------------|---------------|---------------|-----------|------|
| 5   | 50  | 200 | 30  | 3 | 1  | 0   | 0  | <b>3341.2</b> | 3600.0        | 3600.0        | 15456 (1) | 21   |
| 5   | 50  | 200 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 50  | 200 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 50  | 250 | 10  | 3 | 3  | 2   | 0  | <b>103.5</b>  | 3010.2        | 3600.0        | 3107      | 14   |
| 5   | 50  | 250 | 20  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 50  | 250 | 30  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 50  | 250 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 50  | 250 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 75  | 225 | 10  | 3 | 3  | 3   | 1  | <b>8.9</b>    | 772.3         | 1330.5        | 1549      | 6    |
| 5   | 75  | 225 | 20  | 3 | 3  | 2   | 1  | <b>79.5</b>   | 1593.8        | 2091.1        | 5807      | 11   |
| 5   | 75  | 225 | 30  | 3 | 1  | 0   | 0  | <b>1463.4</b> | 3600.0        | 3600.0        | 5352 (1)  | 13   |
| 5   | 75  | 225 | 40  | 3 | 1  | 0   | 0  | <b>1941.2</b> | 3600.0        | 3600.0        | 8097 (1)  | 14   |
| 5   | 75  | 225 | 50  | 3 | 1  | 1   | 0  | <b>1314.1</b> | 3364.5        | 3600.0        | 4732 (1)  | 14   |
| 5   | 75  | 300 | 10  | 3 | 3  | 3   | 3  | <b>2.6</b>    | 267.8         | 535.9         | 527       | 6    |
| 5   | 75  | 300 | 20  | 3 | 3  | 0   | 0  | <b>613.3</b>  | 3600.0        | 3600.0        | 7721      | 19   |
| 5   | 75  | 300 | 30  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 75  | 300 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 75  | 300 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 75  | 375 | 10  | 3 | 2  | 2   | 0  | <b>132.3</b>  | 2360.6        | 3600.0        | 1464 (2)  | 12   |
| 5   | 75  | 375 | 20  | 3 | 1  | 0   | 0  | <b>2512.2</b> | 3600.0        | 3600.0        | 7926 (1)  | 42   |
| 5   | 75  | 375 | 30  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 75  | 375 | 40  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |
| 5   | 75  | 375 | 50  | 3 | 0  | 0   | 0  | <b>3600.0</b> | <b>3600.0</b> | <b>3600.0</b> | —         | —    |

The additional block with  $q = 3$  and  $S = 100$  uses the networks and the seeds of the instances with  $q = 3$  of our benchmark (45 instances). Table S5 summarizes it by number of nodes. DPA solves 38 instances, TV 18 and CR 14, and DPA is the fastest method for every number of nodes. The median of  $|Y_N|$  over the 38 instances with known  $|Y_N|$  is 21 608, and CR processes 120 partitions per nondominated vector, against 24 with  $q = 3$  and supply at most 50. These instances are used in the diagnosis of Section 10.3 of the paper.

Table S5: Additional block with  $q = 3$  and total supply  $S = 100$ , by number of nodes  $n$ , with a limit of 3600 seconds. Columns as in Table 1 of the paper.

| $n$ | $m$           | #  | Solved |     |    | Time (s) |               |        | Part./     |      |  |
|-----|---------------|----|--------|-----|----|----------|---------------|--------|------------|------|--|
|     |               |    | CR     | DPA | TV | CR       | DPA           | TV     | $ Y_N $    | vec. |  |
| 20  | 60, 80, 100   | 9  | 5      | 9   | 7  | 1465.9   | <b>426.4</b>  | 1587.9 | 22555      | 134  |  |
| 30  | 90, 120, 150  | 9  | 4      | 9   | 3  | 2047.3   | <b>557.7</b>  | 2034.6 | 18382      | 188  |  |
| 40  | 120, 160, 200 | 9  | 3      | 7   | 4  | 1372.4   | <b>865.8</b>  | 1456.9 | 13545 (7)  | 85   |  |
| 50  | 150, 200, 250 | 9  | 1      | 8   | 2  | 2962.3   | <b>1261.0</b> | 3074.0 | 31362 (8)  | 123  |  |
| 75  | 225, 300, 375 | 9  | 1      | 5   | 2  | 2096.0   | <b>1727.8</b> | 2409.6 | 40153 (5)  | 34   |  |
| All |               | 45 | 14     | 38  | 18 | 1912.3   | <b>852.0</b>  | 2034.6 | 21608 (38) | 120  |  |

## S5. Results on the instances of Könen, Löhken and Stiglmayr

Könen, Löhken and Stiglmayr [5] published a collection of MOIMCF instances [6], together with the nondominated sets computed with the defining

point algorithm. We use the classes with  $q \geq 3$  whose results are published, either in [5] (Tables 4.3, 4.4, 4.5 and 4.8) or in the thesis of Könen [4] (Tables 5.1, 5.5 and 5.6), except the two classes with 1000 nodes, which are left out to bound the computing time of the experiment: 22 classes, counting each supply factor of Table 4.3 of [5] as a class. In each class we take, as in our benchmark, three instances: the first ones in the numbering of the repository whose nondominated set is known with certainty, that is, for which both DPA and TV terminate within 3600 seconds and return a set of the same size as the one published with the instance. In class N04, where TV solves only three of the fifteen instances, this rule retains the easier ones. The sample has 66 instances, all of them unchanged in the repository on the date of access. Except for the classes N01 and N04 (Table 4.8 of [5]), derived from the instances of Raith and Ehrgott, with 9 and 18 sources and costs in  $[1, 100]$ , the classes of Table 4.3 of [5] with supply and capacities multiplied by  $\mu = 2, 3$  and 4, and the classes of Table 5.6 of [4], generated with a maximum capacity  $u_{\max}$  of 1, 5 or 100, all instances have two sources and two sinks, a total supply of 50, capacities in  $[0, 50]$  and costs in  $[1, 10]$ . Table S6 gives the results per class.

Table S6: Sample of the instances of Könen et al. with known nondominated set: table of [5] or [4] where the class appears,  $q$ ,  $n$ ,  $m$ , variant ( $\mu$ : factor applied to the supply and the capacities;  $u_{\max}$ : maximum capacity), number of instances, mean and maximum  $|Y_N|$ , shifted geometric mean of the CPU time in seconds with a shift of 1 second, and number of instances solved by CR within 3600 seconds, with unsolved instances at 3600 seconds. DPA and TV terminate on all of them.

| Table                                             | $q$ | $n$ | $m$ | Variant          | # | $ Y_N $ |       | Time (s) |       |        | CR<br>solved |
|---------------------------------------------------|-----|-----|-----|------------------|---|---------|-------|----------|-------|--------|--------------|
|                                                   |     |     |     |                  |   | mean    | max   | DPA      | TV    | CR     |              |
| <i>Classes of Könen, Löhken and Stiglmayr [5]</i> |     |     |     |                  |   |         |       |          |       |        |              |
| 4.3                                               | 3   | 50  | 100 | $\mu = 1$        | 3 | 214     | 613   | 1.3      | 1.8   | 0.3    | 3            |
| 4.3                                               | 3   | 50  | 100 | $\mu = 2$        | 3 | 1065    | 3132  | 3.1      | 3.2   | 1.8    | 3            |
| 4.3                                               | 3   | 50  | 100 | $\mu = 3$        | 3 | 2935    | 8709  | 5.8      | 6.4   | 5.7    | 3            |
| 4.3                                               | 3   | 50  | 100 | $\mu = 4$        | 3 | 5222    | 15537 | 7.4      | 9.9   | 10.9   | 3            |
| 4.4                                               | 3   | 20  | 40  |                  | 3 | 44      | 87    | 0.2      | 0.6   | 0.0    | 3            |
| 4.4                                               | 3   | 20  | 80  |                  | 3 | 4645    | 6952  | 41.7     | 58.6  | 105.9  | 3            |
| 4.4                                               | 3   | 100 | 200 |                  | 3 | 603     | 1678  | 5.3      | 6.5   | 1.7    | 3            |
| 4.4                                               | 3   | 100 | 400 |                  | 3 | 1479    | 3450  | 35.0     | 34.0  | 16.7   | 3            |
| 4.5                                               | 3   | 10  | 20  |                  | 3 | 612     | 1023  | 2.7      | 4.6   | 0.2    | 3            |
| 4.5                                               | 4   | 10  | 20  |                  | 3 | 1161    | 2743  | 5.6      | 15.6  | 0.7    | 3            |
| 4.5                                               | 5   | 10  | 20  |                  | 3 | 112     | 163   | 1.2      | 2.7   | 0.0    | 3            |
| 4.8                                               | 3   | 20  | 60  | N01              | 3 | 3257    | 5289  | 66.7     | 126.5 | 7.7    | 3            |
| 4.8                                               | 3   | 40  | 120 | N04              | 3 | 3792    | 4920  | 117.9    | 198.3 | 51.8   | 3            |
| <i>Classes of the thesis of Könen [4]</i>         |     |     |     |                  |   |         |       |          |       |        |              |
| 5.1                                               | 3   | 200 | 400 |                  | 3 | 1497    | 2873  | 40.6     | 42.9  | 61.0   | 3            |
| 5.1                                               | 3   | 200 | 800 |                  | 3 | 3404    | 6106  | 127.3    | 132.8 | 438.5  | 1            |
| 5.5                                               | 3   | 50  | 200 |                  | 3 | 1799    | 2986  | 34.4     | 62.7  | 23.3   | 3            |
| 5.5                                               | 3   | 50  | 300 |                  | 3 | 4917    | 6413  | 107.3    | 121.0 | 713.6  | 3            |
| 5.5                                               | 3   | 50  | 400 |                  | 3 | 2134    | 4944  | 32.4     | 36.3  | 350.1  | 1            |
| 5.5                                               | 3   | 50  | 600 |                  | 3 | 13724   | 17782 | 472.0    | 720.0 | 3600.0 | 0            |
| 5.6                                               | 3   | 50  | 200 | $u_{\max} = 1$   | 3 | 13      | 28    | 0.3      | 0.5   | 0.0    | 3            |
| 5.6                                               | 3   | 50  | 200 | $u_{\max} = 5$   | 3 | 319     | 470   | 7.6      | 11.5  | 1.3    | 3            |
| 5.6                                               | 3   | 50  | 200 | $u_{\max} = 100$ | 3 | 1795    | 2635  | 25.8     | 48.5  | 12.3   | 3            |

CR solves 59 of the 66 instances and is faster than DPA on 48 of them and than TV on 49. The seven unsolved instances belong to the densest classes of the thesis, with  $n = 200$ ,  $m = 800$  (two instances) and  $n = 50$ ,  $m = 400$  (two) and  $m = 600$  (three); DPA solves each of them in less than 540 seconds. CR is also slower than DPA on the classes with  $n = 50$ ,  $m = 300$  and  $n = 200$ ,  $m = 400$ , and on the class of Table 4.3 with  $\mu = 4$ , whose fronts have several thousand vectors. As with  $q = 3$  and large supply in our benchmark, CR loses its advantage on dense networks with large nondominated sets. On the classes of Table 5.6, which differ only in the maximum capacity, the ratio between the times of DPA and CR decreases from 5.8 with  $u_{\max} = 5$  to 2.1 with  $u_{\max} = 100$ .

## S6. Ablation study

The ablation study uses 24 instances of our benchmark, 8 with each value of  $q$ , chosen mechanically: the method without arc fixing (variant `nofix` below) is run with a limit of 60 seconds on the 135 instances with total supply  $S \leq 30$  that are the first of the three instances of their combination of  $q$ ,  $n$ ,  $m$  and  $S$  (index  $k = 0$  in Section S3), and for each  $q$  the 8 solved instances with the largest running times are taken. Every variant is run with a limit of 300 seconds of CPU time. Each variant removes one component of the complete method of Section 7:

- nofix** no compatible arcs and no arc fixing (Section 7.5);
- noprobe** no probing along the returned circuits (Section 6.5);
- pairs1** weighted bounds with the vectors  $e_i + e_j$  and the all-ones vector only (Section 9);
- nocert** no box certificates: every live box is examined at every state;
- noweighted** no weighted bounds: only the ideal point  $m(P)$  in the pruning test and in the definition of the live boxes;
- simplearc** the opening arc is the arc of the returned cycle with the largest reduced first-objective cost, instead of the rule of Section 9;
- basicgen** the basic generator of Section 5, with the search region used only for pruning, and therefore without arc fixing;
- nobox** no search region: the basic algorithm of Section 6, with the pruning test of Proposition 4;
- s1** every box is examined from the representative  $x^1$ , instead of the representative closest to it (Section 7.3);
- noreps** the representatives other than the one used to branch are not offered to the incumbent set when they are computed.

All the variants that terminate return the same nondominated sets as the complete method. Table S7 reports, for each  $q$ , the number of solved instances, the shifted geometric mean of the CPU time with a shift of 1 second, with unsolved instances at 300 seconds, and the ratio between the geometric means of the number of processed partitions of the variant and of the complete method on the instances solved by both.

Table S7: Ablation study with a limit of 300 seconds.

| Variant    | $q = 3$ (8) |       |       | $q = 4$ (8) |       |       | $q = 5$ (8) |      |       |
|------------|-------------|-------|-------|-------------|-------|-------|-------------|------|-------|
|            | Sol.        | Time  | Part. | Sol.        | Time  | Part. | Sol.        | Time | Part. |
| complete   | 8           | 33.9  | 1.00  | 8           | 26.7  | 1.00  | 8           | 14.9 | 1.00  |
| nofix      | 8           | 39.2  | 1.37  | 8           | 33.6  | 1.70  | 8           | 18.8 | 2.09  |
| noprobe    | 8           | 42.2  | 1.29  | 8           | 27.2  | 1.04  | 8           | 15.0 | 1.04  |
| pairs1     | 6           | 108.5 | 3.17  | 8           | 32.7  | 1.70  | 8           | 15.2 | 1.43  |
| nocert     | 8           | 35.9  | 1.05  | 8           | 26.9  | 1.01  | 8           | 14.8 | 1.00  |
| noweighted | 0           | 300.0 | –     | 2           | 272.4 | 23.22 | 7           | 78.2 | 8.69  |
| simplearc  | 8           | 38.4  | 1.43  | 8           | 26.6  | 1.08  | 8           | 16.6 | 1.20  |
| basicgen   | 8           | 178.4 | 6.94  | 8           | 54.8  | 3.46  | 8           | 21.0 | 2.87  |
| nobox      | 0           | 300.0 | –     | 2           | 263.3 | 37.80 | 7           | 67.2 | 25.77 |
| s1         | 8           | 86.1  | 3.17  | 8           | 28.0  | 1.31  | 8           | 13.8 | 1.05  |
| noreps     | 8           | 37.0  | 1.18  | 8           | 27.8  | 1.13  | 8           | 15.5 | 1.15  |

The search region and the weighted bounds are the decisive components. Without the search region (nobox) or without the weighted bounds (noweighted), none of the instances with  $q = 3$ , 2 of those with  $q = 4$  and 7 of those with  $q = 5$  are solved within the limit, and on the solved instances the number of partitions grows by factors between 9 and 38. With the search region used only for pruning (basicgen), the number of partitions grows by a factor between 2.9 and 6.9; with the reduced family of weights (pairs1), by a factor between 1.4 and 3.2, and two instances with  $q = 3$  are not solved. Examining every box from  $x^1$  instead of from the closest representative (s1) multiplies the number of partitions by 3.2 with  $q = 3$ , and by 1.3 and 1.05 with  $q = 4$  and  $q = 5$ . Arc fixing (nofix) reduces the number of partitions by factors of 1.4, 1.7 and 2.1 with  $q = 3, 4$  and 5, and the running time by factors between 1.16 and 1.26. The rule for the opening arc of Section 9 reduces the number of partitions by a factor between 1.08 and 1.43 with respect to the arc of largest reduced first-objective cost, probing by a factor between 1.04 and 1.29, and offering the representatives to the incumbent set by a factor between 1.13 and 1.18. Except for arc fixing, whose effect grows with  $q$ , the effect of most components decreases with  $q$  on these instances, which, with  $q = 5$ , have a smaller supply and fewer partitions. The certificates (nocert) hardly change the partition: the geometric mean of the number of partitions changes by at most 5%, and on 22 of the 24 instances the number changes by less than 7%; on the other two, both with  $q = 3$ , it grows by 9% and 29%. A certificate records that a box contains no vector of a partition and of all its subpartitions; without it, the box is examined again in each subpartition, where the detection, being a conservative test, usually finds no cycle, but may occasionally return a cycle and open a different branch, which

explains these small differences. The detections avoided by the certificates are cheap, and the time they save is offset by their maintenance, so that their effect on the running time is negligible.

## References

- [1] D. Klingman, A. Napier, J. Stutz, NETGEN: A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems, *Management Science* 20(5) (1974) 814–821.
- [2] A. Makhorin, GNU Linear Programming Kit, version 4.57 (2015), <https://www.gnu.org/software/glpk/>.
- [3] A. Rumpf, PyNETGEN 1.0.0: a Python implementation of the NETGEN network generator (2022), <https://github.com/adam-rumpf/pynetgen>.
- [4] D. Könen, Multi-Objective Combinatorial Optimization: Supportedness, Representations, and Integer Network Flows, PhD thesis, University of Wuppertal, 2025.
- [5] D. Könen, L. Löhken, M. Stiglmayr, Representing the non-dominated set of multi-objective network problems by supported non-dominated points, arXiv:2607.11821 (2026).
- [6] D. Könen, L. Löhken, M. Stiglmayr, Supp\_Representations: test instances, [https://git.uni-wuppertal.de/loehken/supp\\_representations](https://git.uni-wuppertal.de/loehken/supp_representations), commit 166a4527. Accessed: 2026-10-04.