



GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Institut für Numerische und Angewandte Mathematik

The blockwise coordinate descent method for integer programs

S. Jäger, A. Schöbel

Nr. 15

Preprint-Serie des
Instituts für Numerische und Angewandte Mathematik
Lotzestr. 16-18
D - 37083 Göttingen

The blockwise coordinate descent method for integer programs

Sven Jäger
Institut für Mathematik
Technische Universität Berlin
Germany

Anita Schöbel
Institut für Numerische und Angewandte Mathematik
Georg-August Universität Göttingen
Germany

June 27, 2018

Abstract

Blockwise coordinate descent methods have a long tradition in continuous optimization and are also frequently used in discrete optimization under various names. New interest in blockwise coordinate descent methods arises for improving sequential solutions for problems which consist of several planning stages.

In this paper we systematically formulate and analyze the blockwise coordinate descent method for integer programming problems. We discuss convergence of the method and properties of the resulting solutions. We extend the notion of Pareto optimality for blockwise coordinate descent to the case that the blocks do not form a partition and compare Pareto-optimal solutions to blockwise optimal and to global optimal solutions. Among others we derive a condition which ensures that the solution obtained by blockwise coordinate descent is Pareto optimal and we confirm convergence of the blockwise coordinate descent to a global optimum in matroid polytopes.

The results are interpreted in the context of multi-stage problems and illustrated for integrated planning in public transportation.

1 Introduction and Literature

Coordinate descent methods have a long history in optimization. In the continuous case the simplest form is the Gauß-Seidel method. In 1957, Hildreth [Hil57] analyzed this method for quadratic programs and proved that the sequence of objective values converges to the optimal objective value and that, if the constraint matrix has full rank, also the sequence of solutions converges to an optimal solution. Two years later, D’Esopo [DE59] generalized the procedure to convex optimization problems and proved that under rather strong conditions (feasible set is a box, all subproblems have a unique optimal solution) the sequence of objective values converges to the optimal objective value and every accumulation point of the sequence of generated solutions is an optimal solution. Blockwise coordinate descent was introduced by Warga [War63], who transferred the convergence result of D’Esopo. Subsequently, several weaker conditions that ensure convergence to an optimal solution appeared. For example, Grippo and Sciandrone [GS00] showed that if

the feasible region is the product of the closed, convex feasible regions of the subproblems, the objective function is continuously differentiable and pseudoconvex, and the set of feasible solutions whose objective value is at most the objective value of the initial solution is compact, then the sequence of solutions generated by the blockwise coordinate descent method has limit points and every limit point is a global minimizer. Furthermore, Jäger [Jäg16] proved that the compactness assumption can be replaced by the assumption that the feasible sets of the subproblems are either polyhedra or intersections of sublevel sets of strictly convex functions and that the sequence converges. For nonpseudoconvex objective functions Zangwill [Zan69] gave a condition that only guarantees coordinatewise optimality of the limit points because convergence to a globally optimal solution does not hold in this case. This result was generalized for the blockwise coordinate descent by Tseng [Tse01], who showed that if the feasible region is the product of the closed, convex feasible regions of the subproblems, the objective function is continuous, the set of feasible solutions whose objective value is at most the objective value of the initial solution is compact, and every subproblem has a unique optimal solution, then every limit point of the sequence generated by the blockwise coordinate descent algorithm is blockwise optimal. Similar algorithms are compass search and the Frank-Wolfe algorithm. The notion of blockwise optimality is related to the notions of an equilibrium point from equilibrium programming and of a Nash equilibrium from game theory, namely blockwise optimal solutions correspond to Nash equilibria of potential games, introduced in [MS96]. Coordinate descent applied to find them is known under the name of best-response paths and for example treated by Voorneveld [Voo00].

In discrete optimization there are many approaches in which a problem is solved by in turn fixing parts of the variables and solving the resulting subproblem. This is done in many publications under many different names. We list a few examples. In facility location, blockwise coordinate descent has become popular as Cooper’s method [Coo64]. Here a set of new facilities is randomly chosen as starting solution, then the assignment of the customers to the facilities is determined, and then this assignment is kept fixed and the facilities are re-optimized. Cooper’s method is still under research, see, e.g., [DBMS15, DS17]. A similar iterative method is used also in statistics for multi regression lines or for trimmed regression, see, e.g., [Rou87, Haw94]. Here a (set of) regression lines is randomly chosen and fixed, then the assignment and/or the outliers are determined and the lines are further optimized according to these fixed assignments. Using a large number of starting solutions provides good results. The same procedure is used for k -means clustering and there known as Lloyd’s algorithm [Llo82]. It has been shown to converge to the so-called centroidal Voronoi tessellation in [DEJ06]. Under the name *iterative variable fixing heuristics*, blockwise coordinate descent is furthermore popular in many applications in logistics, e.g., for solving a combined blending and distribution planning problem [Bil07] or in transportation planning when combining vehicle and crew scheduling [GSS05]. A general scheme on how to apply blockwise coordinate descent in problems of transportation planning is proposed in the eigenmodel in [Sch17].

2 Problem definition

We consider integer programming problems given as

$$(P) \quad \min\{f(x) : G(x) \leq 0, x \in \mathbb{Z}^n\}$$

for $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $G : \mathbb{R}^n \rightarrow \mathbb{R}^m$. We denote the feasible set of P by $\mathfrak{X} := \{x \in \mathbb{Z}^n : G(x) \leq 0\}$. Note that (without loss of generality) we require the functions f and G to be defined anywhere on \mathbb{R}^n .

Let $I \subseteq \{1, \dots, n\}$ be a group of variables (sometimes called block) and $I^C := \{1, \dots, n\} \setminus I$

its complement. We split the variables in a part $x_I \in \mathbb{R}^{|I|}$ and $x_{I^c} \in \mathbb{R}^{n-|I|}$ such that we may write $f(x) = f(x_I, x_{I^c})$ and $G(x) = G(x_I, x_{I^c})$. For any fixed $y \in \mathbb{R}^{n-|I|}$ we receive a subproblem

$$(P_I(y)) \quad \min\{f(x, y) : G(x, y) \leq 0, x \in \mathbb{Z}^{|I|}\}$$

in which we only determine values for the variables with indices in I . We often write *subproblem I* referring to the subproblem P_I determined by the variables in set I .

In the blockwise coordinate descent method we iteratively solve a sequence of such subproblems. To this end, let

$$I_1, I_2, \dots, I_p \in \text{Pot}(\{1, 2, \dots, n\})$$

be the indices of the variables of the respective p subproblems. We require that every variable appears in at least one of the subproblems, i.e., $\bigcup_{j=1, \dots, p} I_j = \{1, \dots, n\}$. We write $\mathfrak{J} := \{I_1, \dots, I_p\}$ for the set of subproblems to be considered. Note that \mathfrak{J} is a cover of $\{1, \dots, n\}$. Let $n_j := |I_j|$ be the number of variables of subproblem j .

In order to simplify notation, we use the index j instead of the subproblem I_j and formulate the subproblem P_{I_j} in which the variables from I_j^c are fixed and we only determine new values for the variables in I_j as

$$(P_j(x_{-j})) \quad \min\{f(x_{+j}, x_{-j}) : G(x_{+j}, x_{-j}) \leq 0, x_{+j} \in \mathbb{Z}^{n_j}\},$$

where

$$\begin{aligned} P_j &:= P_{I_j} \text{ is the subproblem with variables only in } I_j, \\ x_{+j} &:= x_{I_j} = (x_i : i \in I_j) \in \mathbb{R}^{n_j} \text{ is its set of variables, and} \\ x_{-j} &:= x_{I_j^c} = (x_i : i \notin I_j) \in \mathbb{R}^{n-n_j} \text{ are the remaining variables.} \end{aligned}$$

The blockwise coordinate descent method can then be formulated as follows.

Algorithm Blockwise Coordinate Descent (BCD)

Input: Problem P with objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and constraints $G : \mathbb{R}^n \rightarrow \mathbb{R}^m$, sequence of subproblems given by sets $I_1, \dots, I_p \in \text{Pot}(\{1, \dots, n\})$, initial solution $x^{(0)}$.

Set $x := x^{(0)}$, $k := 0$

Repeat

For $j := 1$ to p **do**

1. Solve $P_j(x_{-j}^{(k)})$, let x^* be an optimal solution thereof.
2. Set $x^{(k+1)} := (x^*, x_{-j}^{(k)})$, $k := k + 1$.

Until stopping criterion

We call a complete execution of the For-loop a *round* and the single execution of Steps 1 and 2 an *iteration*. When nothing else is mentioned, we use the following stopping criterion: We terminate if in one round all problems I_1, \dots, I_p have been executed without an improvement in the objective function.

3 Properties of the sequence

We denote by $x^{(k)}$ the solution at the end of the k -th iteration of Algorithm BCD.

3.1 Existence of optimal solutions in the subproblems

Since the solution $x^{(k)}$ of iteration k is a feasible solution to $P_j(x_{-j}^{(k)})$ for all $j = 1, \dots, p$, all subproblems occurring during the blockwise coordinate descent have a feasible solution if the starting solution $x^{(0)}$ is feasible for P . However, if \mathfrak{X} is an unbounded set it might happen that some of the problems $P_j(x)$ are unbounded and hence admit no optimal solution. We investigate this case for integer linear programs.

Lemma 1. *Let P be an integer linear program with rational data, $j \in \{1, \dots, p\}$, and $x, y \in \mathfrak{X}$ such that $P_j(x_{-j})$ and $P_j(y_{-j})$ are feasible. Then $P_j(x_{-j})$ is unbounded if and only if $P_j(y_{-j})$ is unbounded.*

Proof. Suppose that $P_j(y_{-j})$ is unbounded. Then also its LP relaxation is unbounded. By the decomposition theorem for polyhedra (Minkowski's theorem, see, e.g., [NW88]), its feasible region can be written as $\text{conv}(\{v^{(1)}, \dots, v^{(g)}\}) + \text{conic}(\{r^{(1)}, \dots, r^{(h)}\})$ for some $v^{(1)}, \dots, v^{(g)}, r^{(1)}, \dots, r^{(h)} \in \mathbb{R}^{n_j}$, and $c_j^T r^{(k)} < 0$ for some $k \in \{1, \dots, h\}$. The vector $((r^{(k)})^T, 0_{-j}^T)^T$ lies in the recession cone (see, e.g., [Roc70]) of the LP relaxation of P , so $r^{(k)}$ lies in the recession cone of the LP relaxation of $P_j(x_{-j})$, i.e., of the convex hull of the feasible region of $P_j(x_{-j})$. As $r^{(k)}$ has a negative objective value, the latter is unbounded, so the problem $P_j(x_{-j})$ must be unbounded as well. \square

Note that the assumption of linearity is crucial in Lemma 1 since for integer non-linear programs, the lemma does not hold in general as the following example shows.

Example 2. *Consider the integer non-linear program*

$$(P) \quad \min\{x_1 + x_2 \mid x_1 \cdot x_2 \geq -1, x_1 \leq 0, x_1, x_2 \in \mathbb{Z}\}.$$

For $I = \{1\}$ and $x_2 = 1$ the subproblem $P_I(1)$ has the optimal solution $x_1 = -1$ but for $x_2 \leq 0$ the subproblem $P_I(0)$ is unbounded.

For integer linear programs with rational data, Lemma 1 tells us the following: If Algorithm BCD finds an optimal solution for every problem in the first round (i.e., when every subproblem $I \in \mathfrak{J}$ has been considered once), then it will find an optimal solution in every subproblem considered later. This can be tested by only looking at the first feasible solution $x^{(0)}$.

Corollary 3. *Let P be an integer linear program with rational data. If $x^{(0)}$ is feasible for P and $P_j(x_{-j}^{(0)})$ is bounded for all $j = 1, \dots, p$, then all subproblems $P_j(x_{-j})$ which appear when executing Algorithm BCD have an optimal solution.*

3.2 Objective values

If all subproblems have an optimal solution, then the sequence $(f(x^{(k)}))_{k \in \mathbb{N}}$ of objective values is monotonically decreasing since the solution $x^{(k)}$ of step k is a feasible solution to $P_j(x_{-j}^{(k)})$ for all $j = 1, \dots, p$. This implies convergence of the sequence if the optimization problem P is bounded. Together with Corollary 3 we hence get the following condition for convergence.

Corollary 4. *Let P be an integer linear program with rational data. If $x^{(0)}$ is feasible for P and $P_j(x_{-j}^{(0)})$ is bounded for all $j = 1, \dots, p$, then the sequence $(f(x^{(k)}))_{k \in \mathbb{N}}$ of objective values generated by Algorithm BCD is monotonically decreasing. Furthermore, if P is bounded, the sequence converges.*

Note that boundedness of all subproblems $P_j(x_{-j})$ and for all $x \in \mathfrak{X}$ does not imply boundedness for P , also not in the case of integer linear programming of Corollary 4, hence the assumption that P is bounded is in fact, necessary. We next investigate the

question if the sequence $(f(x^{(k)}))_{k \in \mathbb{N}}$ (if it converges) becomes eventually constant, i.e., if Algorithm BCD terminates. This need not hold in general even if P has an optimal solution, as the next example demonstrates.

Example 5. Consider the integer nonlinear program

$$\min \left\{ \frac{2x_1 + 2x_2 - 4}{(x_1 + x_2)^2} : -x_1 - x_2 \leq -1, x_1 - x_2 \leq 1, -x_1 + x_2 \leq 1, x_1, x_2 \in \mathbb{Z} \right\}.$$

An optimal solution to it is $(0, 1)$ with objective value -2 . However, for the initial solution $x^{(0)} = (3, 3)$ the sequence generated by Algorithm BCD for $I_1 = \{1\}$ and $I_2 = \{2\}$ is given by

$$x^{(k)} = \begin{cases} (k+3, k+2) & \text{if } k \text{ is odd} \\ (k+2, k+3) & \text{if } k \text{ is even} \end{cases} \quad \text{for } k = 1, 2, \dots$$

and the resulting sequence of objective values for $k \geq 1$ is $f(x^{(k)}) = \frac{6+4k}{(5+2k)^2} \rightarrow 0$ which converges to zero but never becomes constant.

The next lemma gives some cases in which the sequence $(f(x^{(k)}))_{k \in \mathbb{N}}$ becomes constant.

Lemma 6. Let P be given and let the sequence $(x^{(k)})_{k \in \mathbb{N}}$ exist (i.e., assume that Algorithm BCD is executable). The sequence of objective values $(f(x^{(k)}))_{k \in \mathbb{N}}$ generated by Algorithm BCD becomes constant if one of the following conditions hold.

- (i) The set \mathfrak{X} is bounded.
- (ii) The sublevel set $\{x \in \mathfrak{X} : f(x) \leq f(x^{(l)})\}$ is bounded for some $x^{(l)}$.
- (iii) The function f is a polynomial with rational coefficients and P is bounded.

Proof. We show that $(f(x^{(k)}))$ can only take finitely many different values. Since it is monotonically decreasing, it then must become constant. For (i) this is clear, since a bounded set of integers is finite. For (ii) we use that $(f(x^{(k)}))_{k \in \mathbb{N}}$ is decreasing, i.e., $f(x^{(k)}) \leq f(x^{(l)})$ for all $k \geq l$ and hence the sequence $(x^{(k)})$ remains for $k \geq l$ in the bounded set of feasible solutions with objective value at most $f(x^{(l)})$, which is a finite set. In case (iii) this holds because $f(x^{(k)})$ can only take values that are integer multiples of the product of the denominators of the coefficients of f , and again there are only finitely many such values between $f(x^{(0)})$ and a finite lower bound. \square

In particular, for integer linear programming, f is certainly a polynomial. We hence obtain the following condition for termination of Algorithm BCD.

Corollary 7. Let P be a bounded integer linear program with rational coefficients. Algorithm BCD terminates if $x^{(0)}$ is feasible for P and $P_j(x_{-j}^{(0)})$ is bounded for all $j = 1, \dots, p$.

Proof. If $x^{(0)}$ is feasible for P and $P_j(x_{-j}^{(0)})$ is bounded for all $j = 1, \dots, p$, then the sequence generated by Algorithm BCD exists due to Corollary 3. Hence the result follows from Lemma 6, (iii). \square

We remark that Lemma 6 and Corollary 7 rely on the integrality of the variables: both do not hold for continuous linear programming.

3.3 Sequence elements

We now investigate the localization of the sequence elements. For continuous strictly quasiconcave optimization problems, the following is well known: If there exists a minimum then all minimizers lie on the boundary $\partial\mathfrak{X}$ of the feasible set \mathfrak{X} . In this case, the sequence of points $(x^{(k)})_{k \in \mathbb{N}}$ (if it exists) is contained in the boundary of \mathfrak{X} . For continuous quasiconcave functions, if a minimum exists, a minimizer can be chosen as a boundary

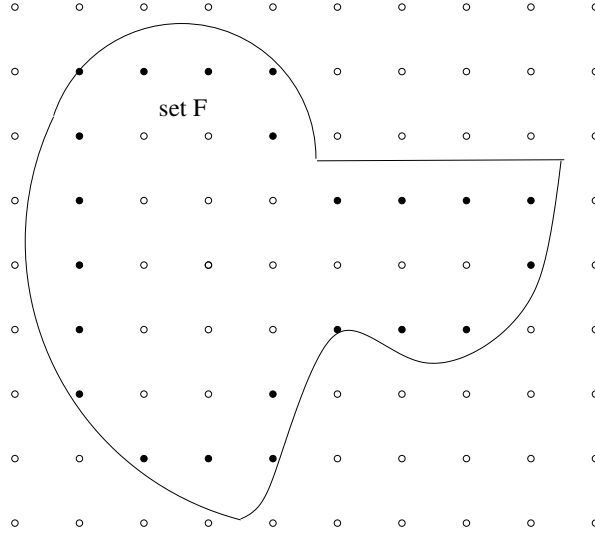


Figure 1: Let \mathfrak{X} be the integer points contained in F . The close-to-boundary points are the filled circles. They are contained in a strip along the boundary ∂F .

point, which is in fact common for most algorithms (e.g. for the Simplex algorithm in linear programming). If such an algorithm is chosen for solving the subproblems we again receive that $(x^{(k)})_{k \in \mathbb{N}} \subseteq \partial \mathfrak{X}$. In this section we transfer this localization result to integer problems.

Definition 8. Let P be an integer optimization problem with feasible set $\mathfrak{X} \subseteq \mathbb{Z}^n$. We call $x \in \mathfrak{X}$ close-to-boundary if there is some $y \in \mathbb{Z}^n \setminus \mathfrak{X}$ with $\|x - y\|_1 = 1$.

Note that $\|x - y\|_1 = 1$ if and only if there exists $i \in \{1, \dots, n\}$ such that $y = x + e_i$ or $y = x - e_i$ ($e_i \in \mathbb{R}^n$ being the i -th unit vector). Let us first give some meaning to this definition for the case that $\mathfrak{X} = \{x \in \mathbb{Z}^n : x \in F\}$ is given as the set of integer points of some set F . For example, F might be the feasible set of the linear programming relaxation of an integer program.

Consider a close-to-boundary point $x \in \mathfrak{X}$ and a point $y \in \mathbb{Z}^n \setminus \mathfrak{X}$ with $\|x - y\|_1 = 1$, say, $y = x + e_i$ for some $i \in \{1, \dots, n\}$. Since $x \in F$ and $y \notin F$, there exists a point $z \in \partial F$ on the straight line between x and y which satisfies $\|z - x\|_2 = \|z - y\|_2 \leq \|y - x\|_1 = 1$. This means that the (Euclidean) distance between a close-to-boundary point x and the boundary ∂F satisfies

$$d(x, \partial F) = \inf_{z' \in \partial F} \|z' - x\|_2 \leq \|x - z\|_2 \leq 1,$$

i.e., all close-to-boundary points in \mathfrak{X} are located within a strip along the boundary of F with a diameter of at most one.

The next lemma shows that for strictly quasiconcave functions the sequence $(x^{(k)})_{k \in \mathbb{N}}$ generated by Algorithm BCD is contained in such a strip along the boundary of F .

Lemma 9. Let P be given with a strictly quasiconcave objective function f and let the sequence $(x^{(k)})_{k \in \mathbb{N}}$ exist. Then $x^{(k)}$ is a close-to-boundary point for all $k \geq 1$.

For an illustration of the lemma, see Figure 1.

Proof. Let $x = x^{(k)}$ be a solution generated by Algorithm BCD, let's say after optimizing the variables x_{+j} of subproblem $I_j \in \mathfrak{J}$. This means that $f(x) = f(x_{+j}, x_{-j}) \leq f(y, x_{-j})$

for all y with $(y, x_{-j}) \in \mathfrak{X}$. Now, choose $i \in I_j$ as one of the variables of subproblem I_j and consider the two points $x' := x - e_i$ and $x'' := x + e_i$. Since $i \in \mathfrak{J}_j$ we have that $x' = ((x - e_i)_{+j}, x_{-j})$ and $x'' = ((x + e_i)_{+j}, x_{-j})$. Furthermore, x is a convex combination of x' and x'' and f is strictly quasiconcave; hence we get

$$f(x) > \min\{f(x - e_i), f(x + e_i)\} = \min\{f((x - e_i)_{+j}, x_{-j}), f((x + e_i)_{+j}, x_{-j})\},$$

and due to the optimality of x we conclude that

$$f(y, x_{-j}) > \min\{f((x - e_i)_{+j}, x_{-j}), f((x + e_i)_{+j}, x_{-j})\} \text{ for all } y \text{ with } (y, x_{-j}) \in \mathfrak{X}.$$

Hence, either $x - e_i$ or $x + e_i$ has to be infeasible and x is close-to-boundary. \square

4 Properties of the resulting solutions

4.1 Notions of optimality

Definition 10. A feasible solution x^* of P is called (globally) optimal if for every $x' \in \mathfrak{X}$ we have $f(x^*) \leq f(x')$.

We certainly would wish to receive an optimal solution, and we will see that in special cases this is in fact the result of the Algorithm BCD. However, in general the algorithm converges to some suboptimal solution which can be characterized as follows.

Definition 11 (see, e.g., [Tse01]). A feasible solution x^* of P is called blockwise optimal if for every $j \in \{1, \dots, p\}$ and every $x'_{+j} \in \mathbb{Z}^{n_j}$ such that (x'_{+j}, x^*_{-j}) is feasible, it holds that $f(x^*) \leq f(x'_{+j}, x^*_{-j})$.

If Algorithm BCD terminates, i. e., if the objective value of the solution is not improved during a whole round, then the resulting solution is obviously blockwise optimal. In its game-theoretic interpretation, blockwise optimality is a stable situation since none of the subproblems can be improved if only its own variables may be changed. Here, we assume that all players have the same objective function. A more detailed description of the relation to game theory can for example be found in [KT14, Section 13.7]. The question arising is: what happens if also other variables can be changed? A state in which no subproblem can improve its objective value without worsening at least one other subproblem would be a Pareto solution. To use this concept, we first have to define objective functions for each of the subproblems. This is easy if the subproblems $\mathfrak{J} = \{I_1, \dots, I_p\}$ are a partition and the objective function f is *additively separable* with respect to this partition, i. e.,

$$f(x) = \sum_{j=1}^p f_j(x_{+j})$$

can be decomposed into a sum of functions f_j where each function f_j only depends on variables from the subproblems $I_j \in \mathfrak{J}$. However, \mathfrak{J} does not need to be a partition of the variables $\{1, \dots, n\}$ but may be just a cover. We now transfer the notion of separability to this situation.

To this end, we need the coarsest partition of $\{1, \dots, n\}$ that is conform to the given cover \mathfrak{J} .

Definition 12. For a given cover $\mathfrak{J} = \{I_1, \dots, I_p\}$ we define its induced partition.

$$\mathfrak{K} := \left\{ \left(\bigcap_{j \in J} I_j \right) \setminus \left(\bigcup_{j \notin J} I_j \right) : J \subseteq \{1, \dots, p\} \right\} \setminus \{\emptyset\}.$$

\mathfrak{K} contains for every subset $J \subseteq \{1, \dots, p\}$ the set of all variable indices that lie in exactly the sets I_j with $j \in J$ if this is non-empty. This is a partition of $\{1, \dots, n\}$ into sets K_1, \dots, K_q which satisfies

$$K_\ell \subseteq I_j \text{ or } K_\ell \subseteq I_j^C \text{ for all } \ell = 1, \dots, q, j = 1, \dots, p$$

and is the coarsest partition (i.e., the one with the smallest number of sets) under this condition.

The identification of the sets K_1, \dots, K_q is demonstrated in two examples.

Example 13. Let us consider five variables $\{1, 2, 3, 4, 5\}$ and three subproblems $\mathfrak{J} = \{I_1, I_2, I_3\}$ with $I_1 = \{1, 2, 3\}$, $I_2 = \{3, 4, 5\}$, and $I_3 = \{1, 2, 4, 5\}$. This results in

$$\mathfrak{K} = \{\{1, 2\}, \{3\}, \{4, 5\}\} \text{ and } q = 3$$

since for $J = \{1, 2\}$ we receive $(I_1 \cap I_2) \setminus I_3 = \{3\}$, $J = \{1, 3\}$ gives $(I_1 \cap I_3) \setminus I_2 = \{1, 2\}$, $J = \{2, 3\}$ gives $(I_2 \cap I_3) \setminus I_1 = \{4, 5\}$, and all other subsets J result in empty sets.

Example 14. Changing I_3 to $I'_3 = \{1, 5\}$ also changes the partition \mathfrak{K} , in this case we receive for $\mathfrak{J} = \{I_1, I_2, I'_3\}$ that

$$\mathfrak{K} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\} \text{ and } q = 5$$

with, e.g., $I = \{1\}$ leading to $I_1 \setminus (I_2 \cup I'_3) = \{2\}$.

In case that the sets $\mathfrak{J} = \{I_1, \dots, I_p\}$ already form a partition we receive $\mathfrak{K} = \mathfrak{J}$ and $q = p$. Hence, the next definition is a proper generalization of separability as it is known for partitions.

Definition 15. Let $\mathfrak{J} \subseteq \{1, \dots, n\}$ and \mathfrak{K} be its induced partition. A function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is called additively separable with respect to \mathfrak{J} if $f(x) = \sum_{\ell=1}^q g_\ell(x_{K_\ell})$ for some functions $g_\ell: \mathbb{R}^{|K_\ell|} \rightarrow \mathbb{R}$.

If the objective function f is additively separable, we can assign one objective function to each of the subproblems $I_j \in \mathfrak{J}$, namely, we determine which of the sets K_ℓ are contained in I_j and sum up their corresponding functions g_ℓ . This is formalized next.

Definition 16. Let \mathfrak{J} be given and let \mathfrak{K} be its induced partition. Let f be additively separable (with respect to \mathfrak{J}) with $f(x) = \sum_{\ell=1}^q g_\ell(x_{K_\ell})$, and let

$$f_j(x_{+j}) := \sum_{\ell \in \{1, \dots, q\}: K_\ell \subseteq I_j} g_\ell(x_{K_\ell}).$$

A feasible solution x^* to P is called

- weakly Pareto optimal if there does not exist a solution x' which is feasible for P such that $f_j(x'_{+j}) < f_j(x^*_{+j})$ for all $j \in \{1, \dots, p\}$.
- Pareto optimal if there does not exist a solution x' which is feasible for P and dominates x^* , i.e., such that $f_j(x'_{+j}) \leq f_j(x^*_{+j})$ for all $j \in \{1, \dots, p\}$ and such that $f_j(x'_{+j}) < f_j(x^*_{+j})$ for some $j \in \{1, \dots, p\}$.

Let us demonstrate the construction of the objective functions for the subproblems by continuing Example 13.

Example 17. We continue Example 13, i.e., we have $n = 5$, $\mathfrak{J} = \{I_1, I_2, I_3\}$ with $I_1 = \{1, 2, 3\}$, $I_2 = \{3, 4, 5\}$, and $I_3 = \{1, 2, 4, 5\}$ and $\mathfrak{K} = \{\{1, 2\}, \{3\}, \{3, 4\}\}$. Let us consider a polynomial

$$\begin{aligned} f(x) &= x_2^2 + 2x_3^2 + x_5^2 + x_1x_2 - 3x_4x_5 + 3x_1 + x_4 + 2x_5 \\ &= \underbrace{3x_1 + x_2^2 + x_1x_2}_{=:g_1(x_1, x_2)} + \underbrace{2x_3^2}_{=:g_2(x_3)} + \underbrace{x_5^2 - 3x_4x_5 + x_4 + 2x_5}_{=:g_3(x_4, x_5)} \end{aligned}$$

The resulting functions f_1, f_2, f_3 for the subproblems I_1, I_2, I_3 are then given as

$$\begin{aligned} f_1(x_1, x_2, x_3) &= g_1(x_1, x_2) + g_2(x_3) = 3x_1 + x_2^2 + x_1x_2 + 2x_3^2 \\ f_2(x_3, x_4, x_5) &= g_2(x_3) + g_3(x_4, x_5) = 2x_3^2 + x_5^2 - 3x_4x_5 + x_4 + 2x_5 \\ f_3(x_1, x_2, x_4, x_5) &= g_1(x_1, x_2) + g_3(x_4, x_5) = 3x_1 + x_2^2 + x_1x_2 + x_5^2 - 3x_4x_5 + x_4 + 2x_5 \end{aligned}$$

Note that $\sum_{j=1}^3 f_j(x_{+j}) = 2f(x)$ since each of the variables x_i appears in exactly two of the sets I_1, I_2, I_3 .

4.2 Relations between the notions

In this section, for the case that the objective function is additively separable and every element of \mathfrak{R} is contained in exactly $\pi \in \mathbb{N}$ elements of \mathfrak{J} , the relations between the three notions optimality, blockwise optimality, and Pareto optimality are investigated.

Clearly, Pareto optimality implies weak Pareto optimality. However, in general there is no relation between Pareto optimality and blockwise optimality: Pareto optimality does not imply blockwise optimality, nor does blockwise optimality imply weak Pareto optimality.

Example 18. In the integer linear program

$$\begin{aligned} \text{(P)} \quad \min \quad & x_1 + x_2 \\ \text{s. t.} \quad & -x_1 + 2x_2 \leq 1 \\ & 2x_1 - x_2 \leq 1 \\ & x_1, x_2 \in \mathbb{Z} \end{aligned}$$

with the two subproblems given by $I_1 = \{1\}$ and $I_2 = \{2\}$ the solution $(1, 1)$ is blockwise optimal since in $P_1(1)$ the only feasible solution is $x_1 = 1$ and in $P_2(1)$ the only feasible solution is $x_2 = 1$. For discussing Pareto optimality we look at $f_1(x_1) = x_1$ and $f_2(x_2) = x_2$ as the two parts of the additively separable objective function and receive that $(1, 1)$ is not weakly Pareto optimal since $(0, 0)$ is a strict improvement for both subproblems.

Example 19. We consider the integer linear program

$$\begin{aligned} \text{(P)} \quad \min \quad & x_1 + x_2 + x_3 \\ \text{s. t.} \quad & x_1 + 2x_2 = 0 \\ & x_3 = 0 \\ & x_2 \leq 2 \\ & x_2 \geq 0 \\ & x_1, x_2, x_3 \in \mathbb{Z} \end{aligned}$$

with the groups $I_1 = \{1, 2\}$, $I_2 = \{2, 3\}$, and $I_3 = \{1, 3\}$, yielding the partition $\mathfrak{R} = \{\{1\}, \{2\}, \{3\}\}$, and hence the objective functions for the subproblems are $f_1(x_1, x_2) = x_1 + x_2$, $f_2(x_2, x_3) = x_2 + x_3$, and $f_3(x_1, x_3) = x_1 + x_3$. The solution $x^* = (0, 0, 0)$ is then Pareto optimal: To see this, we assume that x' is also feasible. If $x'_2 = 0$ then $x' = x^*$. If $x'_2 > 0$ then $f_2(x'_2, x'_3) > 0 = f_2(x_2^*, x_3^*)$, so x' does not dominate x^* , and x^* is Pareto optimal.

On the other hand, x^* is not blockwise optimal, since an optimal solution to

$$(P_1(0)) \quad \min\{x_1 + x_2 : x_1 + 2x_2 = 0, x_2 \leq 2, x_2 \geq 0\}$$

is given as $x'_1 = -4$, $x'_2 = 2$ with an objective function value of $f(x'_1, x'_2, 0) = -2 < 0 = f(0, 0, 0)$.

Nevertheless, we now give conditions under which Pareto optimality implies blockwise optimality and under which blockwise optimality implies weak Pareto optimality.

Theorem 20. *Let P be given with a partition $\{I_1, \dots, I_p\}$ of the variable indices and an additively separable objective function f . Then every Pareto optimal solution of P is blockwise optimal.*

Proof. Since $\{I_1, \dots, I_p\}$ is a partition we have $\mathfrak{K} = \mathfrak{J}$ and $f(x) = \sum_{j=1}^p f_j(x_{+j})$ with x_{+j} containing the variables from I_j , $j = 1, \dots, p$. Assume that x^* is not blockwise optimal, i.e., that there is a $j_0 \in \{1, \dots, p\}$ and an x'_{+j_0} such that $(x'_{+j_0}, x^*_{-j_0})$ is feasible and $f(x'_{+j_0}, x^*_{-j_0}) < f(x^*)$. Define $x' := (x'_{+j_0}, x^*_{-j_0})$. Then for all $j \in \{1, \dots, p\} \setminus \{j_0\}$ it holds that $x'_{+j} = x^*_{+j}$, whence $f_j(x'_{+j}) = f_j(x^*_{+j})$, and for j_0 we have

$$f_{j_0}(x'_{+j_0}) = f(x') - \sum_{j \in \{1, \dots, p\} \setminus \{j_0\}} f_j(x'_{+j}) < f(x^*) - \sum_{j \in \{1, \dots, p\} \setminus \{j_0\}} f_j(x^*_{+j}) = f_{j_0}(x^*_{+j_0}).$$

So x^* is not Pareto optimal. \square

The interpretation of the theorem for subproblems forming a partition is: if it is not possible to improve any of the subproblems without worsening at least another one, then also none of the subproblems can improve its solution by changing only its own variables. Intuitively, the reverse direction will usually not hold. This can be seen in Example 18 showing that blockwise optimality does not imply weakly Pareto optimality even for linear programs and subproblems forming a partition.

However, as the next theorem shows, there exist cases in which the solutions of integer programs with linear objective function found by Algorithm BCD are weakly Pareto optimal. We need the following condition.

Definition 21. *Let a set of subproblems $\mathfrak{J} = \{I_1, \dots, I_p\}$ with corresponding induced partition $\mathfrak{K} = \{K_1, \dots, K_q\}$ be given. We say that \mathfrak{J} has uniform structure if every set K_ℓ is included in the same number π of sets in \mathfrak{J} , i.e., $|\{j \in \{1, \dots, p\} \mid K_\ell \subseteq I_j\}| = \pi$ for every $\ell \in \{1, \dots, q\}$.*

We remark that \mathfrak{J} has uniform structure in Example 13 ($\pi = 2$), in Example 18 ($\pi = 1$), and in Example 19 (again, $\pi = 2$). Also in typical settings in which \mathfrak{J} is chosen as the set of all groups of variables with the same number of elements, i.e.,

$$\mathfrak{J} = \{J \subseteq \{1, \dots, n\} : |J| = k\}$$

for some given (usually small) number $k \in \mathbb{N}$, it has uniform structure with $\pi = k$. Using the uniform structure we now can give a sufficient condition under which each blockwise optimal solution is weakly Pareto optimal.

Theorem 22. *Let P be given with $\mathfrak{J} = \{I_1, \dots, I_p\}$ and induced partition $\mathfrak{K} = \{K_1, \dots, K_q\}$. Let \mathfrak{J} have uniform structure. If the objective function f of P is linear and the feasible set is given as*

$$\mathfrak{X} = \{x \in \mathbb{Z}^n : a^\top x \leq b, G_\ell(x_{K_\ell}) \leq 0 \text{ for } \ell \in \{1, \dots, q\}\}$$

with one coupling constraint $a^\top x \leq b$ for $a \in \mathbb{R}^n, b \in \mathbb{R}$ and for every $\ell \in \{1, \dots, q\}$ some $m_\ell \in \mathbb{N}$ constraints $G_\ell: \mathbb{R}^{|K_\ell|} \rightarrow \mathbb{R}^{m_\ell}$ only containing variables from the same set $K_\ell \in \mathfrak{K}$, then every blockwise optimal solution to P is weakly Pareto optimal.

Proof. Since f is linear, it is additively separable, i.e., $f(x) = \sum_{\ell=1}^q g_\ell(x_{K_\ell})$ for some linear functions g_ℓ . Consequently, we have for every subproblem I_j and $(x_{+j}, 0_{-j})$ that

$$f(x_{+j}, 0_{-j}) = \sum_{\ell \in \{1, \dots, q\}: K_\ell \subseteq I_j} g_\ell(x_{K_\ell}) + \sum_{\ell \in \{1, \dots, q\}: K_\ell \subseteq I_j^c} \underbrace{g_\ell(0_{K_\ell})}_{=0} = f_j(x_{+j})$$

with $f_j(x_{+j})$ the objective for subproblem I_j from Definition 16.

Let $x^* \in \mathbb{R}^n$ be feasible, but not weakly Pareto optimal. So there is an $x' \in \mathbb{R}^n$ with $a^\top x' \leq b$, $G_\ell(x'_{K_\ell}) \leq 0$, $\ell = 1, \dots, q$, and $f_j(x'_{+j}) < f_j(x^*_{+j})$ for all $j \in \{1, \dots, p\}$. Let $y := x' - x^*$, and $\beta := \max\{0, a^\top y\}$. Since every variable is contained in exactly π blocks, we have that

$$y = \frac{1}{\pi} \cdot \sum_{j=1}^p (y_{+j}, 0_{-j}).$$

Hence,

$$\beta \geq a^\top y = \frac{1}{\pi} \sum_{j=1}^p a^\top (y_{+j}, 0_{-j}) \geq \frac{p}{\pi} \min_{j \in \{1, \dots, p\}} a_{+j}^\top y_{+j}.$$

Let j_0 be an index where the minimum is attained, so $\frac{p}{\pi} a^\top (y_{+j_0}, 0_{-j_0}) \leq \beta$. Since $\pi \leq p$ and $\beta \geq 0$, this implies that $a^\top (y_{+j_0}, 0_{-j_0}) \leq \beta$. Define

$$x'' := x^* + (y_{+j_0}, 0_{-j_0}) = (x'_{+j_0}, x^*_{-j_0}),$$

i.e., x'' coincides with x' for all variables $i \in I_{j_0}$ and it coincides with x^* for all variables $i \notin I_{j_0}$. We obtain that x'' is feasible:

- For every set K_ℓ we have either $K_\ell \subseteq I_{j_0}$ or $K_\ell \subseteq I_{j_0}^C$. In the former case $G_\ell(x'') = G_\ell(x') \leq 0$ and in the latter case $G_\ell(x'') = G_\ell(x^*) \leq 0$.
- $a^\top x'' = a^\top x^* + a^\top (y_{+j_0}, 0_{-j_0}) \leq a^\top x^* + \beta = \max\{a^\top x^*, a^\top x'\} \leq b$.

Furthermore, using that f is linear and the construction of x' , we receive

$$f(x'') = f(x^*) + f(y_{+j_0}, 0_{-j_0}) = f(x^*) + f_{j_0}(x'_{+j_0}) - f_{j_0}(x^*_{+j_0}) < f(x^*).$$

So x^* was not blockwise optimal. \square

We now apply Theorem 22 to integer linear programs and assume that \mathfrak{J} forms a partition, i.e., $\mathfrak{J} = \mathfrak{R}$.

Corollary 23. *Let P be an integer linear program with a partition $\mathfrak{J} = \{I_1, \dots, I_p\}$ of the variable indices with $n_j := |I_j|$, which may have*

- m_j constraints involving variables from the same set $I_j \in \mathfrak{J}$,
- but only one coupling constraint,

i.e.,

$$\begin{aligned} \min \quad & \sum_{j=1}^p c_{+j}^\top x_{+j} \\ \text{s.t} \quad & A_j x_{+j} \leq b_j, \quad j = 1, \dots, p, \\ & a^\top (x_{+1}, \dots, x_{+p}) \leq b \\ & x_{+j} \in \mathbb{R}^{n_j}, \quad j = 1, \dots, p \end{aligned}$$

with $c_{+j} \in \mathbb{R}^{n_j}$, $A_j \in \mathbb{R}^{m_j, n_j}$, $b_j \in \mathbb{R}^{m_j}$, and $a \in \mathbb{R}^n$, $b \in \mathbb{R}$.

Then Algorithm BCD terminates with a Pareto optimal solution for P .

Note that Algorithm BCD terminates with a global optimum in the case of Corollary 23 if no coupling constraint is present.

4.3 Blockwise coordinate descent in matroids

In this section we discuss the properties of Algorithm BCD in matroids. Clearly, to find an optimal solution of a matroid optimization problem, one would use the well-known Greedy approach. Nevertheless, it is of theoretical interest to understand the behaviour of Algorithm BCD in this case.

First, the matroid optimization problem can be formulated as follows. Given a set E with n elements and a set of independent sets \mathcal{I} on E , $\mathcal{I} \subseteq \text{Pot}(E)$. Let (E, \mathcal{I}) be a matroid. Furthermore, let weights c_e for all $e \in E$ be given. The matroid optimization problem then asks for an independent set $A \subseteq E$ with maximum weight, i.e.,

$$\max\{f(A) = \sum_{e \in A} w_e : A \in \mathcal{I}\}.$$

A matroid optimization problem can be written as integer program with variables x_e being one if and only if e is contained in the independent set:

$$(P) \quad \max\{\sum w_e x_e : \sum_{e \in E} x_e \leq r(A) \text{ for all } A \subseteq E\}$$

where $r : \mathcal{I} \rightarrow \mathbb{R}$ is the (submodular) rank function of the matroid. Each variable of P corresponds to an element $e \in E$. We hence can use the elements of E directly to define the subsets of variables in our optimization problems P_{I_j} .

For the minimum spanning tree problem, it is known (see, e.g., [DPV08]) that one can start from a spanning tree and by swapping edges (adding an edge and removing another one instead) one can reach any other spanning tree in the graph. When one starts from an arbitrary spanning tree and moves towards a minimum spanning tree, the elements of the sequence can easily be constructed such that they have decreasing costs. For blockwise coordinate descent, we assume that the sequence of subproblems is fixed, i.e., it cannot be chosen according to the actual tree. Still, the following result confirms that Algorithm BCD finds a global optimal solution in matroid optimization problems when all two-element sets of variables are contained in the set of subproblems.

Theorem 24. *Let P be a matroid optimization problem. Let all two-element subsets be contained in \mathfrak{J} , i.e.,*

$$\{\{e, e'\} : e, e' \in E \text{ and } e \neq e'\} \subseteq \mathfrak{J}.$$

Then Algorithm BCD terminates with a global optimal solution of P if the set $A^{(0)} = \{e \in E : x_e^{(0)} = 1\}$ obtained from the starting solution $x^{(0)}$ is an independent set.

Proof. Algorithm BCD terminates, since the set of feasible solutions $\{x \in \{0, 1\}^n : A(x) \in \mathcal{I}\}$ is finite. Let x^{BCD} be the solution with which Algorithm BCD stops. Assume that x^{BCD} is not optimal, i.e., there exists some solution x^* to the matroid optimization problem P with $f(x^*) < f(x^{\text{BCD}})$.

Define $A(x) := \{e \in E : x_e = 1\}$ for every solution x of P, and for every subset $A \subseteq E$ define its corresponding solution $x^A \in \{0, 1\}^n$ through $x_e^A = 1$ if and only if $e \in A$. Let $A^{\text{BCD}} := A(x^{\text{BCD}}) \subseteq E$. Let us choose an optimal solution x^* as similar as possible to x^{BCD} , i.e., such that for $A^* := A(x^*) \subseteq E$ we have that

$$|A^* \cap A^{\text{BCD}}|$$

is maximal. We now construct a set $I \in \mathfrak{J}$ for which x^{BCD} can be improved.

Let e^* be an element of $A^* \setminus A^{\text{BCD}}$ with maximal weight. $A^{\text{BCD}} \cup \{e^*\}$ contains a cycle C with $e^* \in C$. Since $A^* \in \mathcal{I}$ we know that $C \setminus A^* \neq \emptyset$. Let $e' \in C \setminus A^*$. Removing e' leads to a new independent set

$$\tilde{A} := A^{\text{BCD}} \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}.$$

Suppose $w_{e^*} \leq w_{e'}$. Then $A^* \cup \{e'\}$ contains a cycle from which we can remove an element $\tilde{e} \in A^* \setminus A^{\text{BCD}}$ to obtain an independent set $A' := A^* \cup \{e'\} \setminus \{\tilde{e}\}$. Since e^* has been chosen as an element with maximal weight in $A^* \setminus A^{\text{BCD}}$ we obtain

$$w_{e'} \geq w_{e^*} \geq w_{\tilde{e}},$$

hence $f(A') = f(A^*) + w_{e'} - w_{\tilde{e}} \geq f(A^*)$ and $|A' \cap A^{\text{BCD}}| = |A^* \cap A^{\text{BCD}}| + 1$, a contradiction to the choice of A^* as optimal solution with maximal number of same elements as A^{BCD} . We conclude $w_{e^*} > w_{e'}$ and hence compute

$$f(x(\tilde{A})) = \sum_{e \in \tilde{A}} w_e = \sum_{e \in A^{\text{BCD}}} w_e - w_{e'} + w_{e^*} > \sum_{e \in A^{\text{BCD}}} w_e = f(x(A^{\text{BCD}})).$$

This finally shows that Algorithm BCD cannot have terminated with x^{BCD} since for $I := \{e', e^*\} \in \mathfrak{J}$ the algorithm would have improved x^{BCD} to $x(\tilde{A})$. \square

5 Relation to the continuous relaxation

In many applications, even the subproblems $P_I(x)$ are hard to solve (in particular in the application mentioned in the next section). Hence, one could try to find a solution by not solving the integer programming problems, but their relaxations in every step of Algorithm BCD. Here we investigate what we can conclude from a solution obtained by applying Algorithm BCD to the continuous relaxation of P instead of applying it to the problem P itself.

Let R denote the continuous relaxation of P. Clearly, applying Algorithm BCD to R instead of P does not yield an upper bound on its objective value. The following example of an integer linear program shows that it neither yields a lower bound, even if

- Algorithm BCD is started at the same feasible solution $x^{(0)}$ when solving R and P, and,
- it is applied to the relaxation of the ideal formulation of P, and
- it terminates with an integer point when applied to the continuous relaxation R.

Example 25. Consider the integer linear program

$$\begin{aligned} \text{(P)} \quad & \min \quad x_1 + x_2 \\ & \text{s. t.} \quad -2x_1 + x_2 \leq 0 \\ & \quad \quad 2x_1 - x_2 \leq 10 \\ & \quad \quad -x_1 - 2x_2 \leq 0 \\ & \quad \quad x_1, x_2 \in \mathbb{Z} \end{aligned}$$

This is the ideal formulation for the feasible set because every feasible point of its linear programming relaxation R can be written as a convex combination of feasible points of P. We apply Algorithm BCD to this problem as well as to its continuous relaxation R w.r.t. the subproblems $I_1 = \{1\}$ and $I_2 = \{2\}$. If $x^{(0)} = (9, 9)$, then for the integer program, we obtain $x_P^{(1)} = (5, 9)$, $x_P^{(2)} = (5, 0)$, and $x_P^{(3)} = (0, 0)$, where the algorithm terminates. This point has objective value 0. Applied to the linear relaxation, we obtain $x_R^{(1)} = (\frac{9}{2}, 9)$, $x_R^{(2)} = (\frac{9}{2}, -1)$, and $x_R^{(3)} = (2, -1)$, which has the objective value 1 being worse than the value obtained from applying Algorithm BCD to P. Both sequences are illustrated in Figure 2.

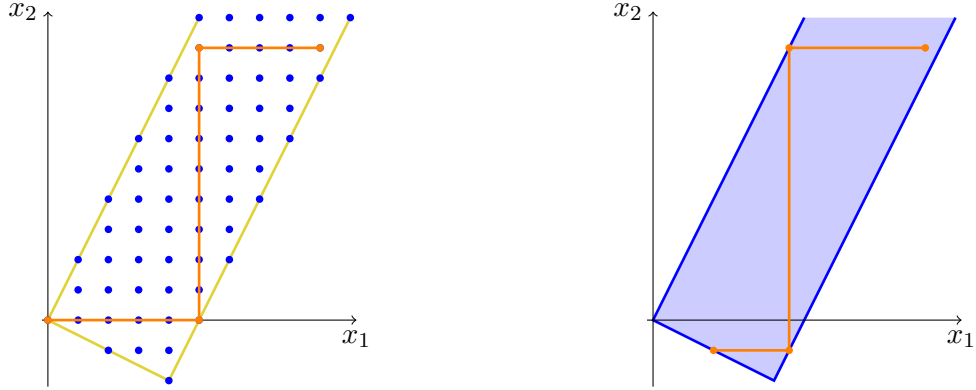


Figure 2: Illustration of Example 25, on the left Algorithm BCD applied to P, and on the right to its ideal formulation

Note that in the example, the solution obtained by Algorithm BCD applied to the continuous relaxation is still blockwise optimal for the original problem P. In case Algorithm BCD happens to find a solution which is feasible for P this holds in general, as the next lemma shows. Moreover, the lemma gives a condition under which also Pareto optimality for a solution to R is transferred to P.

Lemma 26. *Let $Y \subset \mathbb{R}^n$ be a set containing the feasible set X of P, and let*

$$(R) \quad \begin{aligned} \min \quad & f(x) \\ \text{s. t.} \quad & x \in Y. \end{aligned}$$

Let \mathfrak{J} be the subproblems to be considered.

1. Let x^* be a blockwise optimal solution to R which is feasible to P. Then x^* is (strictly) blockwise optimal to P.
2. Let f be additively separable with respect to \mathfrak{J} , and let x^* be a (weakly) Pareto optimal solution to R which is feasible to P. Then x^* is (weakly) Pareto optimal to P.

Proof.

1. If x^* were not blockwise optimal for P, there would be a $j \in \{1, \dots, p\}$ and an x'_{+j} with $(x'_{+j}, x^*_{-j}) \in X$ and $f(x'_{+j}, x^*_{-j}) < f(x^*)$. Then $(x'_{+j}, x^*_{-j}) \in Y$ so that x^* would not be blockwise optimal for R. For strict blockwise optimality, the proof is analogous.
2. If x^* were not weakly Pareto optimal for P, there would be an $x' \in X$ with $f_j(x'_{+j}) < f_j(x^*_{+j})$ for all $j \in \{1, \dots, p\}$. Since $x' \in Y$, the solution x^* would not be weakly Pareto optimal to P. The statement for Pareto optimality can be proven in the same way. \square

From Lemma 26 we obtain the following consequences for Algorithm BCD.

Corollary 27. *Let R be the continuous relaxation of P, and let x_R be the solution obtained by applying Algorithm BCD to R. We then have:*

- If x_R is feasible for P, it is blockwise optimal for P.
- Under the conditions of Theorem 22, and if x_R is feasible for P, then it is weakly Pareto optimal for P.

6 Application of Algorithm BCD for improving solutions to sequential processes

Many applications deal with multi-stage optimization problems in which the stages are optimized sequentially because it is computationally too complex to solve the integrated problem as a whole. Let us look at such a series of p optimization problems. The sequential process starts by solving the first problem (P'_1) , i.e., it determines the variables $x_{+1} \in \mathbb{Z}^{n_1}$ appearing in the first stage. The obtained solution is then fixed in the second stage in which the variables x_{+2} are determined. The process continues until in the last step, the values of all previous stages, i.e., $x_{+1}, \dots, x_{+(p-1)}$ are fixed and a solution x_{+p} to the last stage is determined. This gives the *sequential* solution $x^{\text{seq}} = (x_{+1}, \dots, x_{+p})$ to the multi-stage optimization problem.

$$\begin{aligned}
 (P'_1) \quad & \min\{f_1(x_{+1}) : G_1(x_{+1}) \leq 0, x_{+1} \in \mathbb{Z}^{n_1}\} \\
 (P'_2)(x_{+1}) \quad & \min\{f_2(x_{+1}, x_{+2}) : G_2(x_{+1}, x_{+2}) \leq 0, x_{+2} \in \mathbb{Z}^{n_2}\} \\
 (P'_3)(x_{+1}, x_{+2}) \quad & \min\{f_3(x_{+1}, x_{+2}, x_{+3}) : G_3(x_{+1}, x_{+2}, x_{+3}) \leq 0, x_{+3} \in \mathbb{Z}^{n_3}\} \\
 & \vdots \\
 (P'_p)(x_{+1}, \dots, x_{+(p-1)}) \quad & \min\{f_p(x_{+1}, \dots, x_{+p}) : G_p(x_{+1}, \dots, x_{+p}) \leq 0, x_{+p} \in \mathbb{Z}^{n_p}\}
 \end{aligned}$$

In many applications the resulting solution x^{seq} of the sequential process is then returned as solution to the problem. However, it is usually not optimal for the (*integrated*) multi-stage optimization problem

$$(MP) \quad \min\{f(x_{+1}, \dots, x_{+p}) : G(x_{+1}, \dots, x_{+p}) \leq 0, x \in \mathbb{Z}^n\} \quad (1)$$

where $n := \sum_{i=1}^p n_i$ is the number of all variables of all stages, $G : \mathbb{R}^n \rightarrow \mathbb{R}^M$ contains all constraints of the single stages, and f is the objective function given as a positive linear combination of the objective functions of the p planning steps, i.e.,

$$f(x_{+1}, \dots, x_{+p}) = \sum_{i=1}^p \alpha_i f_i(x_{+1}, \dots, x_{+i}). \quad (2)$$

In order to improve the sequential solution x^{seq} we propose to apply Algorithm BCD with subproblems I_1, \dots, I_p corresponding to the variables x_{+1}, \dots, x_{+p} of the stages $1, \dots, p$, i.e., $|I_i| = n_i$. The p subproblems arising during the execution of Algorithm BCD are

$$(P_i) \quad \min\{f_i(x_{+i}, x_{-i}) : G(x_{+i}, x_{-i}) \leq 0, x = (x_{+1}, x_{-i}) \in \mathbb{Z}^n\}, \quad i = 1, \dots, p$$

The problems (P_i) and (P'_i) have the same variables and may be similar to each other such that algorithms for solving the subproblems (P_i) may be easily derived from known algorithms used in the sequential planning process. In fact, P_p and P'_p coincide. The resulting algorithm is depicted in Figure 3, where the construction of the starting solution is shown in the first p steps, before Algorithm BCD iterates along a cycle to improve the starting solution.

In the following we summarize what the results derived in this paper mean for improving a sequential solution if all problems (P_i) are linear integer optimization problems. We apply Algorithm BCD for the subproblems $\mathfrak{J} = \{I_1, \dots, I_p\}$ which correspond to the different stages $1, \dots, p$ as described above. This is a special case since I_1, \dots, I_p form a partition, and the objective function (2) is a linear combination of the single objectives of the p planning stages and hence additively separable. Using this, Theorem 20 gives the following result.

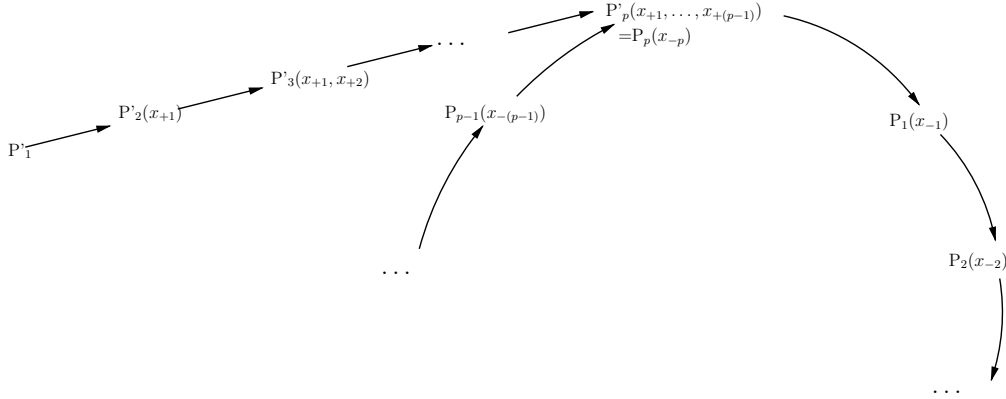


Figure 3: Construction of a starting solution and applying Algorithm BCD for a multi-stage optimization problem MP.

Corollary 28. *Let the multi-stage optimization problem (MP) be given. Then every Pareto solution to (MP) is blockwise optimal.*

We now summarize the results when we apply Algorithm BCD to integer linear multi-stage optimization problems.

Corollary 29. *Let the multi-stage optimization problem (MP) be a linear integer optimization problem with rational coefficients. Let $x^{(0)} = (x_{+1}^{(0)}, \dots, x_{+p}^{(0)})$ be the starting solution obtained by solving the problems $P'_1, P'_2(x_{+1}^{(0)}), \dots, P'_p(x_{+1}^{(0)}, \dots, x_{+(p-1)}^{(0)})$ sequentially. Then the following hold:*

1. *Algorithm BCD is executable if any of the following two conditions is satisfied:*
 - $P_i(x_i^{(0)})$ is a bounded optimization problem for all $i = 1, \dots, p$.
 - The p problems appearing in the first round of Algorithm BCD have optimal solutions.
2. *If Algorithm BCD is executable, then we furthermore have:*
 - (i) *If none of the variables is constant in the objective function f , then all sequence elements are close-to-boundary.*
 - (ii) *If (P) is bounded, then the sequence of objective function values converges and becomes constant, i.e., Algorithm BCD terminates.*
3. *Finally, if Algorithm BCD terminates with a solution x , then we have:*
 - (i) *x is blockwise optimal.*
 - (ii) *If G contains only one coupling constraint, x is Pareto optimal.*

Proof. The first statement follows from Lemma 1 and Corollary 3. Statement 2(i) follows from Lemma 9 since the (linear) objective function is strictly quasiconcave if it is not constant with respect to any of the variables. Statement 2(ii) follows from Corollary 7. Finally, Statement 3(i) is always true, and Statement 3(ii) follows from Theorem 22. \square

For the special case that P is a combinatorial optimization problems, boundedness of the problem and of the subproblems is guaranteed. We can hence strengthen the result of Corollary 29 as follows.

Corollary 30. *Let the multi-stage optimization problem (MP) be a combinatorial optimization problem with rational data. Let $x^{(0)} = (x_{+1}^{(0)}, \dots, x_{+p}^{(0)})$ be the starting solution*

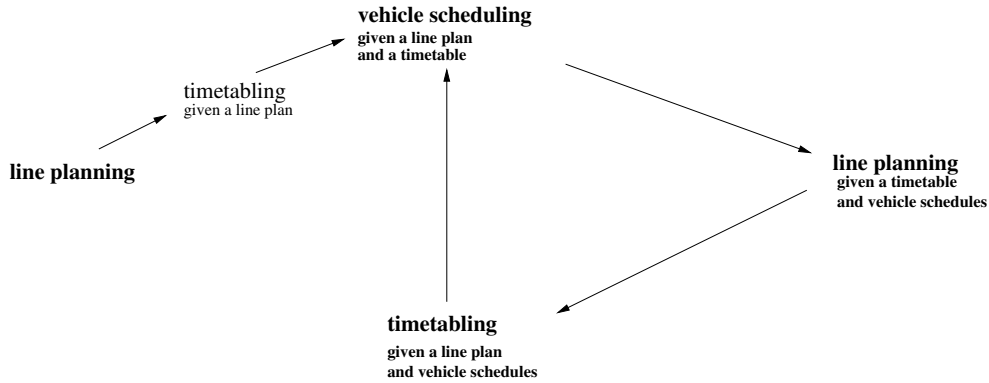


Figure 4: Construction of a starting solution and applying Algorithm BCD for the case of the three stages line planning, timetabling, and vehicle scheduling in public transportation.

obtained by solving the problems $P'_1, P'_2(x_{+1}^{(0)}), \dots, P'_p(x_{+1}^{(0)}, \dots, x_{+(p-1)}^{(0)})$ sequentially. Then Algorithm BCD is executable, stops after a finite number of iterations and the solution obtained is blockwise optimal.

We finally illustrate the application of Algorithm BCD using the multi-stage optimization problems arising in planning of public transportation. The planning process in public transportation can be split into the following planning stages: After the design of the public transportation network, the lines have to be planned. After that, the timetable can be designed, followed by vehicle scheduling, crew scheduling, and crew rostering. For all of these planning problems, models are known and advanced solution techniques are available. However, going through all these stages sequentially leads only to suboptimal solutions. This motivates the tremendous amount of recent research on integrated planning in which two or even more of the planning stages are treated simultaneously, see, e.g., [LM07, GH08, SGSK10, CM12, PLM⁺13, SE16, BBLV17, PSSS17, MCZT18, FvdHRL18]. In [Sch17], Algorithm BCD has been set up for the three stages line planning, timetabling, and vehicle scheduling, and [GSS16] presents experimental results showing that the solution quality can be improved by Algorithm BCD in this application. Figure 4 shows the process: First, a starting solution is found by the classic sequential approach in which first the line planning problem is solved, then a timetable is determined and finally the vehicle schedules are planned. This sequential solution is then treated as starting solution and improved by iterating through the circle depicted in Figure 4.

We finally give an interpretation of results for multi-stage problems when applied to MP arising in public transportation depicted in Figure 4. The notion of *blockwise optimal* of Definition 11 means that we cannot improve any of the planning stages without changing another one. E.g., we cannot improve the timetable without changing the vehicle schedules or the line plan. Looking at *Pareto optimality*, we may change all variables, and not only the variables of one of the stages. A solution is Pareto optimal if we cannot improve one of the planning stages without worsening another one. In general, blockwise optimal solutions need not be Pareto optimal. For example, we might be able to improve the timetable when making a few changes to the vehicle schedules and this may be done without increasing the costs of the vehicle schedule. According to Theorem 22 Pareto optimality only follows from blockwise optimality if there is only one coupling constraint, i.e., when the subproblems are almost independent of each other.

Finally, planning a public transportation system involves integer linear optimization problems with only a finite number of feasible solutions. For this case we can use Corollary 29

and Corollary 30 showing that Algorithm BCD is always executable and terminates after a finite number of iterations with a blockwise optimal solution.

7 Conclusion and further research

In this paper we presented a systematic analysis for blockwise coordinate descent methods applied for integer (linear or nonlinear) programs. We analyzed existence of the sequence elements, convergence of the algorithm, and discussed properties of the (local) minima found. We in particular discussed not only blockwise coordinate descent on a partition of the variables but also on a cover.

For further research we plan to discuss how to construct a good set of subproblems \mathfrak{J} as input for Algorithm BCD. Clearly, nested sets $I_j \subseteq I_{j+1}$ can be avoided in an efficient sequence since the smaller one is always redundant. However, an overlap between the sets seem to be beneficial as first experiments [Spü18] show that using covers instead of partitions yields better results in less time. Using this as a starting point, different strategies on how to choose the cover \mathfrak{J} should be discussed, among them having all equal-size subsets of the variables (yieldings \mathfrak{J} with uniform structure), having larger or smaller sets and sets with big or small overlap to others.

Based on the observations on matroids and on the decomposition results, we furthermore aim at identifying problem classes in which the gap between the quality of the solution obtained by Algorithm BCD compared to an optimal solution can be bounded.

Finally, blockwise coordinate descent methods should be further refined and applied to public transportation, and location problems, both classes having practical applications and being suitable to study the performance of the respective approaches.

References

- [BBLV17] S. Burggraeve, S.H. Bull, R.M. Lusby, and P. Vansteenwegen. Integrating robust timetabling in line plan optimization for railway systems. *Transportation Research C*, 77:134–160, 2017.
- [Bil07] B. Bilgen. *An Iterative Fixing Variable Heuristic for Solving a Combined Blending and Distribution Planning Problem*, pages 231–238. Springer Berlin Heidelberg, 2007.
- [CM12] L. Cadarso and A. Marin. Integration of timetable planning and rolling stock in rapid transit networks. *Annals of Operations Research*, 199:113–135, 2012.
- [Coo64] L. Cooper. Heuristic methods for location-allocation problems. *SIAM Rev*, 6:37–53, 1964.
- [DBMS15] Z. Drezner, J. Brimberg, N. Mladenović, and S. Salhi. New heuristic algorithms for solving the planar p-median problem. *Computers and Operations Research*, 62:296–304, 2015.
- [DE59] D. A. D’Esopo. A convex programming procedure. *Naval Research Logistics Quarterly*, 6(1):33–42, March 1959.
- [DEJ06] Q. Du, M. Emelianenko, and L. Ju. Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. *SIAM Journal on Numerical Analysis*, 44:102–119, 2006.
- [DPV08] Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani. *Algorithms*. McGraw-Hill, 2008.

- [DS17] Z. Drezner and S. Salhi. Incorporating neighborhood reduction for the solution of the planar p-median problem. *Annals of Operations Research*, 258(2):639–654, 2017.
- [FvdHRL18] João Paiva Fonseca, Evelien van der Hurk, Roberto Roberti, and Allan Larsen. A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling. *Transportation Research Part B: Methodological*, 109:128–149, 2018.
- [GH08] V. Guihaire and J.-T. Hao. Transit network design and scheduling: A global review. *Transportation Research Part E*, 42:1251, 2008.
- [GS00] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Operations Research Letters*, 26(3):127–136, April 2000.
- [GSS05] V. Gintner, I. Steinzen, and L. Suhl. A variable fixing heuristic for the multiple-depot integrated vehicle and crew scheduling problem. In *Proceedings of 10th Meeting of Advanced OR and AI Methods in Transportation, and 16th Mini EURO Conference of the EURO Working Group on Transportation (EWGT)*, pages 547–552, 2005.
- [GSS16] P. Gattermann, A. Schiewe, and A. Schöbel. An iterative approach for integrated planning in public transportation. In *9th Triennial Symposium on Transportation Analysis*, 2016.
- [Haw94] Douglas M. Hawkins. The feasible solution algorithm for least trimmed squares regression. *Computational Statistics & Data Analysis*, 17(2):185–196, 1994.
- [Hil57] Clifford Hildreth. A quadratic programming procedure. *Naval Research Logistics Quarterly*, 4(1):79–85, March 1957.
- [Jäg16] Sven Jäger. Iterative algorithms for integrated optimization problems. Masterarbeit, Georg-August-Universität Göttingen, 5 2016.
- [KT14] J. Kleinberg and E. Tardos. *Algorithm Design: Pearson New International Edition*. Pearson, 2014.
- [Llo82] S.P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [LM07] C. Liebchen and R. Möhring. The modeling power of the periodic event scheduling problem: Railway timetables and beyond. In *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes in Computer Science*, pages 3–40. Springer, 2007.
- [MCZT18] L. Meng, F. Corman, X. Zhou, and T. Tang. Special issue on integrated optimization models and algorithms in rail planning and control (editorial). *Transportation Research C*, 88:87–90, 2018.
- [MS96] Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, May 1996.
- [NW88] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988.
- [PLM⁺13] H.L. Petersen, A. Larsen, O.B.G. Madsen, B. Petersen, and S. Ropke. The simultaneous vehicle scheduling and passenger service problem. *Transportation Science*, 47(4):603–616, 2013.
- [PSSS17] J. Pätzold, A. Schiewe, P. Schiewe, and A. Schöbel. Look-Ahead Approaches for Integrated Planning in Public Transportation. In Gianlorenzo

- D'Angelo and Twan Dollevoet, editors, *17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017)*, volume 59 of *OpenAccess Series in Informatics (OASICs)*, pages 1–16, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [Roc70] R. Tyrrell Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, Princeton, NJ, 1970.
- [Rou87] P.J. Rousseeuw. *Robust Regression and Outlier Detection*. John Wiley & Sons, 1987.
- [Sch17] A. Schöbel. An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation. *Transportation Research C*, 74:348–365, 2017.
- [SE16] V. Schmid and J.F. Ehmke. Integrated timetabling and vehicle scheduling with balanced departure times. *OR Spectrum*, 37(4):903–928, 2016.
- [SGSK10] I. Steinzen, V. Gintner, L. Suhl, and N. Kliwer. A time-space network approach for the integrated vehicle- and crew-scheduling problem with multiple depots. *Transportation Science*, 44(3):367–382, 2010.
- [Spü18] F. Spühler. Blockkoordinaten-Abstiegsverfahren für kombinatorische Probleme. Bachelor-Arbeit, Georg-August Univesität Göttingen, 2018.
- [Tse01] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, June 2001.
- [Voo00] Mark Voorneveld. Best-response potential games. *Economics Letters*, 66(3):289–295, 2000.
- [War63] J. Warga. Minimizing certain convex functions. *Journal of the Society for Industrial and Applied Mathematics*, 11(3):588–593, September 1963.
- [Zan69] Willard I. Zangwill. *Nonlinear programming: a unified approach*. Prentice-Hall international series in management. Prentice-Hall, Englewood Cliffs, NJ, 1969.

Institut für Numerische und Angewandte Mathematik
 Universität Göttingen
 Lotzestr. 16-18
 D - 37083 Göttingen

Telefon: 0551/394512
 Telefax: 0551/393944

Email: trapp@math.uni-goettingen.de URL: <http://www.num.math.uni-goettingen.de>

Verzeichnis der erschienenen Preprints 2018

Number	Authors	Title
2018 - 1	Brimberg, J.; Schöbel, A.	When closest is not always the best: The distributed p-median problem
2018 - 2	Drezner, T.; Drezner, Z.; Schöbel, A.	The Weber Obnoxious Facility Location Model: A Big Arc Small Arc Approach
2018 - 3	Schmidt, M., Schöbel, A., Thom, L.	Min-ordering and max-ordering scalarization methods for multi-objective robust optimization
2018 - 4	Krüger, C., Castellani, F., Geldermann, J., Schöbel, A.	Peat and Pots: An application of robust multi-objective optimization to a mixing problem in agriculture
2018 - 5	Krüger, C.	Peat and pots: Analysis of robust solutions for a biobjective problem in agriculture
2018 - 6	Schroeder, P.W., John, V., Lederer, P.L., Lehrenfeld, C., Lubbe, G., Schöberl, J.	On reference solutions and the sensitivity of the 2d Kelvin–Helmholtz instability problem
2018 - 7	Lehrenfeld, C., Olshanskii, M.A.	An Eulerian Finite Element Method for PDEs in Time-dependent Domains
2018 - 8	Schöbel, A., Schiewe, A., Albert, S., Pätzold, J., Schiewe, P., Schulz, J.	LINTIM : An integrated environment for mathematical public transport optimization. Documentation
2018 - 9	Lederer, P.L. Lehrenfeld, C. Schöberl, J.	HYBRID DISCONTINUOUS GALERKIN METHODS WITH RELAXED H(DIV)-CONFORMITY FOR INCOMPRESSIBLE FLOWS. PART II

Number	Authors	Title
2018 - 10	Lehrenfeld, C.; Rave, S.	Mass Conservative Reduced Order Modeling of a Free Boundary Osmotic Cell Swelling Problem
2018 - 11	M. Friedrich, M. Hartl, A. Schiewe, A. Schöbel	System Headways in Line Planning
2018 - 12	M. Lübbecke, C. Puchert, P. Schiewe, A. Schöbel	Integrating line planning, timetabling and vehicle scheduling
2018 - 13	M. Friedrich, M. Müller-Hannemann, R. Rückert, A. Schiewe, A. Schöbel	Robustness as a Third Dimension for Evaluating Public Transport Plans
2018 - 14	J. Pätzold, A. Schiewe, A. Schöbel	Cost-Minimal Public Transport Planning
2018 - 15	S. Jäger, A. Schöbel	The blockwise coordinate descent method for integer programs