



A new mixed-integer programming formulation for the maximally diverse grouping problem with attribute values

Arne Schulz¹

Accepted: 4 April 2022 / Published online: 12 May 2022
© The Author(s) 2022

Abstract

The paper presents a new mixed-integer programming formulation for the maximally diverse grouping problem (MDGP) with attribute values. The MDGP is the problem of assigning items to groups such that all groups are as heterogeneous as possible. In the version with attribute values, the heterogeneity of groups is measured by the sum of pairwise absolute differences of the attribute values of the assigned items, i.e. by the Manhattan metric. The advantage of the version with attribute values is that the objective function can be reformulated such that it is linear instead of quadratic like in the standard MDGP formulation. We evaluate the new model formulation for the MDGP with attribute values in comparison with two different MDGP formulations from the literature. Our model formulation leads to substantially improved computation times and solves instances of realistic sizes (for example the assignment of students to seminars) with up to 70 items and three attributes, 50 items and five attributes, and 30 items and ten attributes to (near) optimality within half an hour.

Keywords Mixed-integer programming · Assignment · Grouping · Combinatorial optimization

1 Introduction

The maximally diverse grouping problem (MDGP) is the problem of assigning a set of items, $i, j \in I$, to groups, $g \in G$, such that each group gets the same or a similar number of items assigned and the sum of pairwise distances d_{ij} between all items assigned to the same group is maximized. Thus, for equal-sized groups the heterogeneity inside groups is maximized, which leads also to similar groups.

Although the MDGP is in its general version formulated with arbitrary distances d_{ij} , common applications like the assignment of students to seminars use attribute values av_i^a for all students and attributes. Examples are grades or other performance scores or zero-one attributes (e.g. for international or non-international). Given these attribute values, the

✉ Arne Schulz
arne.schulz@uni-hamburg.de

¹ Institute of Operations Management, Universität Hamburg, Moorweidenstraße 18, 20148 Hamburg, Germany

distances are computed as $d_{ij}^a = |av_i^a - av_j^a|$, i.e. by the Manhattan metric, for all attributes $a \in A$ and items/students $i, j \in I$.

In this paper, we consider the MDGP where $d_{ij}^a = |av_i^a - av_j^a|$ for attribute values $av_i^a \in \mathbb{Q}_{\geq 0}$ and use results of Schulz (2021b) to introduce a new mixed-integer programming formulation for the problem setting. We prove that our formulation leads to the same optimal objective values as the standard formulation. Our computational results show that this formulation outperforms the standard formulation and the model formulation by Papenberg and Klau (2021) for the MDGP by far. Moreover, we outline how the approach by Schulz (2021b) for the balanced MDGP, i.e. the problem of finding a best balanced solution among all optimal MDGP solutions, can be adapted if there is no MDGP solution fulfilling Assumption 1 (compare page 6) for every attribute or if not equal-sized groups are considered.

The paper is organised as follows: In Sect. 2, the relevant literature is reviewed. Afterwards, we give a formal problem description in Sect. 3. Our mixed-integer programming (MIP) formulation is introduced in Sect. 4. The MIP is compared with the standard formulation and the model by Papenberg and Klau (2021) in a computational study in Sect. 5. The paper closes with a conclusion (Sect. 6).

2 Literature review

The MDGP has been investigated in different settings in the literature. Weitz and Lakshminarayanan (1997) showed that it is mathematically equivalent to VLSI design (group highly connected modules onto the same circuit) and exam scheduling (assigning exam blocks to days). In a further paper, Weitz and Lakshminarayanan (1998) mentioned the assignment of students to project groups as a possible application.

Assignments of students are a common application of grouping problems. In an early work, Beheshtian-Ardekani and Mahmood (1986) assigned students to project groups. In more recent works, students were assigned to study groups (Krass and Ovchinnikov, 2010), work groups (Caserta and Vo, 2013), and multiple teaching groups (on the basis of preferences; Heitmann and Brüggemann (2014)). Johnes (2015) published a review on operations research in education and considered amongst others the assignment of students to courses. Dias and Borges (2017) applied the MDGP to assign students to teams. Students are mostly assigned according to their academic performance (overall average grade or grades in specific courses) to groups. Furthermore, students can be distributed to groups according to their gender to reach an equally distribution of male, female, and third gender students. Moreover, international students can be distributed equally over all groups. All these measures can be implemented as attribute values (e.g. attribute value 0 for male, 1 for female, and 2 for third gender, or 0 for non-international and 1 for international). Mingers and O'Brien (1995), Krass and Ovchinnikov (2006), Krass and Ovchinnikov (2010), and Caserta and Voß (2013) assign students to groups according to binary attributes.

Baker and Benn (2001) investigated in a case study how pupils in a school should be assigned to tutor groups such that the groups are as similar as possible. Criteria are the gender, the ability level, ethnic minority groups, feeder schools, and special educational needs of pupils. All of them can also be represented by attribute values. Rubin and Bai (2015) considered the problem of assigning individuals to teams to make the teams as similar as possible. Schulz (2021b) extended the MDGP (in the version with attribute values) by such a balancing component.

Homogeneity over different days and therefore heterogeneity inside days is also helpful to distribute workload evenly over days. This is, for example, important in surgery scheduling to avoid overtime (surgery durations can be represented by attribute values). Overtime minimization is a frequently investigated objective in surgery scheduling (compare the review by Cardoen et al. (2010)). Schulz (2021c) assigned surgeries to days such that the days are balanced according to the surgery durations. Schulz and Flidner (2021) analyzed the intra-day assignment of surgeries to starting times and rooms according to several balancing criteria. Papenberg and Klau (2021) used the fact that the MDGP aims at homogeneity over groups and heterogeneity inside groups in psychology to partition data sets into equivalent parts.

The MDGP is NP-hard (Feo and Khellaf, 1990) which may be the reason why only a few exact solution approaches for the MDGP were investigated. Gallego et al. (2013) presented a computational study with the standard formulations of both problem variants presented in Sect. 3.1 ((1)–(4) and (1)–(2), (4)–(5), respectively), where only instances with up to 12 items could be solved to optimality within 1800s (general d_{ij}). Papenberg and Klau (2021) introduced an exact MDGP formulation for the setting with equal-sized groups (compare Sect. 3.2) based on a work by Grötschel and Wakabayashi (1989). They could solve instances with 28 items in 950s and instances with 30 items in nearly 10000s to optimality.

Schulz (2021b) investigated the MDGP with attribute values. The author proved that the set of optimal solutions for the MDGP with attribute values equals for at most two attributes the set of feasible solutions of a special system of equations (for more than two attributes this does not hold in general). He searched for the best balanced solution amongst all optimal solutions of the MDGP with attribute values in the case if there is a solution fulfilling Assumption 1 (compare page 7) for every attribute. We explain the ideas of this paper further in Sect. 4, where we work with them. Schulz (2021a) generalised this research for the case in which no MDGP solution fulfilling Assumption 1 for every attribute exists. The paper considered also only equal-sized groups. Schulz (2021a) solved instances with up to 15 items and 5 attributes to optimality within 600s (version with equal sized-groups; (1–4)).

In contrast to exact approaches, the MDGP has been solved by a variety of heuristic solution approaches. Fan et al. (2011) applied a hybrid genetic algorithm to it. An artificial bee colony algorithm has been investigated for the MDGP by Rodriguez et al. (2013). Gallego et al. (2013) developed a tabu search algorithm with strategic oscillation. Tabu search in an iterated version was considered by Palubeckis et al. (2015). Moreover, Brimberg et al. (2015) applied a skewed general variable neighbourhood search algorithm to solve the MDGP, Lai and Hao (2016) iterated maxima search, and Singh and Sundar (2019) a hybrid genetic algorithm. Lai et al. (2020) implemented a neighbourhood decomposition based variable neighbourhood search and a tabu search algorithm to solve the MDGP. A recent review on metaheuristics applied to solve grouping problems can be found in Ramos-Figueroa et al. (2020).

Brimberg et al. (2017) solved the clique partitioning problem as an MDGP. A similar class of problems are dispersion problems. In contrast to the MDGP, only a single group of a given size is selected. Dispersion problems are considered for example in Fernández et al. (2013), Aringhieri et al. (2015), and Amirgaliyeva et al. (2017).

3 Problem description

In this paper, we consider a set of items $i, j \in I$, a set of groups $g \in G$, and a set of attributes $a \in A$ as given. Each item has an attribute value $av_i^a \in \mathbb{Q}_{\geq 0}$ for each attribute $a \in A$. Given

them, we compute the distances between each pair of items for each attribute according to the Manhattan metric as

$$d_{ij}^a = |av_i^a - av_j^a|.$$

Note that it is no restriction to consider non-negative attribute values. As d_{ij}^a measures only differences between them, we can add a constant $c^a = \max_{i \in I: 0 > av_i^a} \{|av_i^a|\}$ to all attribute values of attribute a without changing the d_{ij}^a values since

$$d_{ij}^a = |av_i^a - av_j^a| = |av_i^a + c^a - av_j^a - c^a| = |(av_i^a + c^a) - (av_j^a + c^a)|.$$

In the following subsections, we present the standard formulation of the MDGP (Sect. 3.1) and the formulation by Papenberg and Klau (2021) (Sect. 3.2).

3.1 Standard formulation

The standard integer programming formulation for the MDGP with equal-sized groups is (compare e.g. Gallego et al. (2013), Singh and Sundar (2019) or Schulz (2021a)):

$$\max \sum_{a \in A} \sum_{g \in G} \sum_{i \in I} \sum_{j \in I: j > i} d_{ij}^a x_{ig} x_{jg} \tag{1}$$

with the constraints

$$\sum_{g \in G} x_{ig} = 1 \quad \forall i \in I \tag{2}$$

$$\sum_{i \in I} x_{ig} = |I|/|G| \quad \forall g \in G \tag{3}$$

$$x_{ig} \in \{0, 1\} \quad \forall i \in I, g \in G \tag{4}$$

x_{ig} is a binary variable which equals to one if item i is assigned to group g and is zero otherwise. Objective function (1) maximizes the pairwise differences between each pair of items assigned to the same group according to all attributes. Constraints (2) ensure that each item is assigned to exactly one group while Constraints (3) take care that all groups are equal-sized. Constraints (4) are the binary constraints for the x variables.

Several authors (Fan et al., 2011; Gallego et al., 2013; Singh and Sundar, 2019; Lai et al., 2020) relax (3) to

$$l_g \leq \sum_{i \in I} x_{ig} \leq u_g \quad \forall g \in G, \tag{5}$$

where l_g is a lower bound and u_g an upper bound for the number of items in group g . In this paper, we consider as a start the more restricted case with (3) and relax it afterwards to (5).

Objective function (1) is quadratic. Thus, we have to linearize it to use an off-the-shelf solver for mixed-integer programming. Therefore, we introduce a new set of variables z_{ijg} , $i, j \in I$ with $i < j$ and $g \in G$, such that the variable z_{ijg} is one if items i and j are assigned to group g and zero otherwise. Then, we replace (1) by

$$\max \sum_{a \in A} \sum_{g \in G} \sum_{i \in I} \sum_{j \in I: i < j} d_{ij}^a z_{ijg} \tag{6}$$

and add the constraints

$$z_{ijg} \leq x_{ig} \quad \forall i, j \in I : i < j, g \in G \tag{7}$$

$$z_{ijg} \leq x_{jg} \quad \forall i, j \in I : i < j, g \in G \tag{8}$$

$$z_{ijg} \geq x_{ig} + x_{jg} - 1 \quad \forall i, j \in I : i < j, g \in G \tag{9}$$

$$z_{ijg} \in [0, 1] \quad \forall i, j \in I : i < j, g \in G \tag{10}$$

Objective function (6) replaces the product of variables in (1) by the new variable z_{ijg} . This variable has to be one if items i and j are assigned to group g and zero else. Since $0 \leq z_{ijg} \leq 1$ (10), Constraints (7) and (8) set $z_{ijg} = 0$ if item i or item j is not assigned to group g . Constraints (9) set $z_{ijg} = 1$ if items i and j are assigned to group g .

Model (1)–(4) contains symmetric solutions. Given a solution, swapping the group assignments of all items assigned to two different groups g_1 and g_2 with $\sum_{i \in I} x_{ig_1} = \sum_{i \in I} x_{ig_2}$ leads to a different solutions which is structurally identical because the two groups have the same size. In the case with equal-sized groups, this symmetry results in $|G|!$ structurally identical solutions. The following set of inequalities avoids symmetric solutions by sorting homogeneous groups in increasing order of the smallest index of their assigned items (Salem and Kieffer (2020)): $x_{11} = 1$ and

$$x_{ig} \leq \sum_{j=1}^{i-1} x_{j,g-1} \quad \forall i \in \{2, \dots, |I|\}, g \in \{2, \dots, |G|\} \tag{11}$$

In objective function (1) as well as in (6), d_{ij} is only counted if both items i and j are assigned to the same group. This means that at the moment when an item i is assigned to a group—for example in a branch-and-bound procedure—we are not (fully) aware of the consequences. If i is the first item assigned to the group, this has even no immediate influence on the objective value. If i is not the first but also not the last item assigned to the group, we know that certain d_{ij}^a values are realized but it is still possible that we have to add a large d_{ij}^a value later if the corresponding item j is added to the same group. This might lead to unprofitable decisions at early stages in the branch-and-bound tree. Thus, although objective function (6) is linearized, its quadratic character still influences the search process. We reduce this drawback by a reformulation of the model in Sect. 4.

3.2 Formulation by Papenberg and Klau (2021)

The model formulation by Papenberg and Klau (2021) is based on a work by Grötschel and Wakabayashi (1989). We adapt it here for the multi-attribute case. It uses binary variables \bar{x}_{ij} , $i, j \in I, i < j$, which are one if items i and j are assigned to the same group and zero else. The model formulations is:

$$\max \sum_{a \in A} \sum_{i \in I} \sum_{j \in I: j > i} d_{ij}^a \bar{x}_{ij} \tag{12}$$

with the constraints

$$-\bar{x}_{ij} + \bar{x}_{ik} + \bar{x}_{jk} \leq 1 \quad \forall i, j, k \in I : i < j < k \tag{13}$$

$$\bar{x}_{ij} - \bar{x}_{ik} + \bar{x}_{jk} \leq 1 \quad \forall i, j, k \in I : i < j < k \tag{14}$$

$$\bar{x}_{ij} + \bar{x}_{ik} - \bar{x}_{jk} \leq 1 \quad \forall i, j, k \in I : i < j < k \tag{15}$$

$$\sum_{j \in I: i < j} \bar{x}_{ij} + \sum_{j \in I: j < i} \bar{x}_{ji} = \frac{|I|}{|G|} - 1 \quad \forall i \in I \tag{16}$$

$$\bar{x}_{ij} \in \{0, 1\} \quad \forall i, j \in I : i < j \tag{17}$$

Constraints (13)–(15) are transitivity constraints which ensure that two items are in the same group if both of them are in the same group with a third item. Constraints (16) form equal-sized groups with $\frac{|I|}{|G|}$ items each. Given an item i , it has to be in the same group with $\frac{|I|}{|G|} - 1$ further items to ensure that the group size is $\frac{|I|}{|G|}$. Finally, Constraints (17) are the binary constraints. Note that this model formulation does not contain symmetric solutions, as variables do not have a group index. Note further that $|I| \cdot |G| < \frac{|I|(|I|-1)}{2}$ is equivalent to $2|G| < |I| - 1$. This means that the model formulation by Papenberg and Klau (2021) has more binary variables than the standard formulation if at least three items are assigned to the same group. Moreover, the model is restricted to equal-sized groups. Varying group sizes could be implemented by bounding the left side of (16) to both sides, but then all groups have the same lower and upper bound. If lower and upper bounds for group sizes vary, we would need a set of variables which indicate the assignment of items to groups.

4 New mixed-integer programming formulation

First, we investigate the case with equal-sized groups (Sect. 4.1), i.e. model (1)–(4) to introduce the ideas and get familiar with the notation. Afterwards, we generalize the findings and consider varying group sizes, i.e. model (1)–(2) and (4)–(5), in Sect. 4.2.

4.1 Equal-sized groups

As mentioned above a disadvantage of the problem formulation (1)–(4) is that the share of an item i in the objective function, for example expressed by

$$\sum_{a \in A} \sum_{g \in G: x_{ig}=1} \sum_{j \in I} \frac{d_{ij}^a \cdot x_{jg}}{2},$$

depends on the decision which of the other items are assigned to the same group.

To overcome this drawback, Schulz (2021b) introduced an assignment of the items to blocks $k \in K$ with $|K| = |I|/|G|$ ($|K|$ is the number of items per group) for each attribute according to

Assumption 1 *Let $|G|$ be the number of groups. Then, the $|G|$ items with the largest attribute values according to the considered attribute are assigned to the first block, the $|G|$ items with the next largest attribute values according to the considered attribute are assigned to the second block, and so on.*

Moreover, a binary parameter b_{ki}^a is introduced which is 1 if item i is according to attribute a assigned to block k and 0 else. Schulz (2021b) (Theorem 1 in that paper) proved that the set of optimal solutions for (1)–(4) equals the set of feasible solutions for (2), (4), and

$$\sum_{i \in I: b_{ki}^a=1} x_{ig} = 1 \quad \forall a \in A, g \in G, k \in K \tag{18}$$

if blocks are determined according to Assumption 1 and one of the following two criteria is fulfilled:

1. $|A| = 1$,

$ K $	2	3	4	5	6	$ K $	2	3	4	5	6
				4	5					4	5
		2	3	2	3			2	3	2	3
	1	0	1	0	1		1	0	1	0	1
	1	2	1	2	1		-1	-2	-1	-2	-1
			3	4	3				-3	-4	-3
					5						-5

Fig. 1 \bar{c}_k values (left) and c_k values (right)

2. $|A| > 1$ and the assignment according to Assumption 1 is unique (no two items with identical attribute values are assigned to different blocks regarding the corresponding attribute).

Schulz (2021b) proved also that the set of feasible solutions for (2), (4), and (18) might be empty if $|A| > 2$ (Theorem 3 in that paper).

With the help of the block notation (parameter b_{ki}^a), the optimal objective value for (1)–(4) can be calculated, if the set of feasible solutions for (2), (4), and (18) is not empty (a proof can be found in Schulz (2021b)), as

$$\sum_{a \in A} \sum_{g \in G} \sum_{i \in I} \sum_{j \in I: i < j} d_{ij}^a x_{ig} x_{jg} = \sum_{a \in A} \sum_{k \in K} \sum_{i \in I} b_{ki}^a c_k a v_i^a \tag{19}$$

with

$$c_k = \begin{cases} \bar{c}_k & \text{if } k \leq \lceil \frac{|K|}{2} \rceil \\ -\bar{c}_k & \text{else,} \end{cases} \tag{20}$$

where

$$\bar{c}_k = 2 \cdot \max \left(\frac{|K|}{2} - k, k - \frac{|K|}{2} - 1 \right) + 1 \tag{21}$$

if $|K|$ is even and

$$\bar{c}_k = 2 \cdot \left| k - \left\lceil \frac{|K|}{2} \right\rceil \right| \tag{22}$$

if $|K|$ is odd.

Figure 1 illustrates \bar{c}_k (left side of the figure) and c_k (right side of the figure) values for small numbers of blocks. The first row states the number of blocks. The numbers below are the \bar{c}_k and c_k values, respectively, in increasing order of the block number. If $|K| = 2$, (21) leads to $\bar{c}_1 = \bar{c}_2 = 1$. If $|K| = 3$, (22) leads to $\bar{c}_1 = 2$, $\bar{c}_2 = 0$, and $\bar{c}_3 = 2$. It can easily be seen that \bar{c}_k values are symmetric. The same is true for c_k values but with a different algebraic sign.

Let us consider an example for (19). Let $|K| = 4$, $|A| = 1$. Four items with attribute values $av_1^1 = 8$, $av_2^1 = 5$, $av_3^1 = 4$, and $av_4^1 = 1$ are assigned to the one considered group. Then,

$$\begin{aligned} & \sum_{a \in A} \sum_{g \in G} \sum_{i \in I} \sum_{j \in I: i < j} d_{ij}^a x_{ig} x_{jg} \\ &= \sum_{i \in I} \sum_{j \in I: i < j} |av_i^1 - av_j^1| \\ &= |8 - 5| + |8 - 4| + |8 - 1| + |5 - 4| + |5 - 1| + |4 - 1| \end{aligned}$$

$$\begin{aligned}
 &= (8 - 5) + (8 - 4) + (8 - 1) + (5 - 4) + (5 - 1) + (4 - 1) \\
 &= \underbrace{(8 - 5) + (5 - 1)}_{=(8-1)} + \underbrace{(8 - 4) + (4 - 1)}_{=(8-1)} + (8 - 1) + (5 - 4) \\
 &= 3 \cdot (8 - 1) + 1 \cdot (5 - 4) \\
 &= 3 \cdot 8 - 3 \cdot 1 + 1 \cdot 5 - 1 \cdot 4 \\
 &\stackrel{(21)}{=} \bar{c}_1 \cdot 8 + \bar{c}_2 \cdot 5 - \bar{c}_3 \cdot 4 - \bar{c}_4 \cdot 1 \\
 &\stackrel{(20)}{=} c_1 \cdot 8 + c_2 \cdot 5 + c_3 \cdot 4 + c_4 \cdot 1 \\
 &= \sum_{k \in K} \sum_{i \in I} b_{ki}^1 c_k a v_i^1 \\
 &= \sum_{a \in A} \sum_{k \in K} \sum_{i \in I} b_{ki}^a c_k a v_i^a \tag{23}
 \end{aligned}$$

with b_{ki}^1 according to Assumption 1, i.e. $b_{11}^1 = 1, b_{22}^1 = 1, b_{33}^1 = 1, b_{44}^1 = 1$, and $b_{ki}^1 = 0$ else. Equation (23) shows us why we should assign items to blocks according to Assumption 1. Each c_k value with $k \leq \lceil \frac{|K|}{2} \rceil$ is multiplied with exactly one d_{ij}^a such that each item i is considered exactly once ($d_{ii}^a = |a v_i^a - a v_i^a| = 0$). Because of our definition of d_{ij}^a and the fact that $c_1 > c_2 > \dots > c_{\lceil \frac{|K|}{2} \rceil}$, it is optimal to multiply the largest c_k value, i.e. c_1 , with the largest d_{ij}^a value, i.e. the difference of the largest and the smallest attribute value assigned to the group. This means, we assign the item with the largest attribute value to the first block and the item with the smallest attribute value to the last block. If we repeat this with all remaining items until all items are assigned, we get the assignment according to Assumption 1.

Note that the right side of (19) is independent of x_{ig} . Because of (2), item i is assigned to exactly one group such that

$$\sum_{a \in A} \sum_{k \in K} \sum_{i \in I} \sum_{g \in G} b_{ki}^a c_k a v_i^a x_{ig} = \sum_{a \in A} \sum_{k \in K} \sum_{i \in I} b_{ki}^a c_k a v_i^a.$$

In (19), the objective value is independent of the assignment of items to groups, which underlines that the set of optimal solutions for (1)–(4) equals the set of feasible solutions for (2), (4), and (18). However, we are only sure that (19) is valid if $|A| = 1$ or $|A| = 2$. For $|A| > 2$, in contrast, we are not sure that (19) is valid (Schulz (2021b), Theorems 2 and 3); there might be no feasible solution for (2), (4), and (18)).

The right side of (19) has in comparison to the left side the advantage that the share of item i in the objective value, i.e. $\sum_{a \in A} \sum_{k \in K} b_{ki}^a c_k a v_i^a$, is independent of the assignment of all other items. The idea of this paper is to replace b_{ki}^a by a variable, i.e. make the assignment of items to blocks to an endogenous decision. By this, we overcome the drawback that we do not know whether there is a solution fulfilling (19) if $|A| > 2$.

So, we replace b_{ki}^a (in combination with x_{ig}) by a new variable y_{kig}^a which is 1 if item i is assigned to group g and according to attribute a to block k and 0 else. This makes the assignment of items to blocks to an endogenous decision, which is only allowed if the assignment is feasible regarding to (2) and (4). This does not only ensure feasibility, the resulting model formulation is, moreover, equivalent to the standard MDGP formulation ((1)–(4); compare Theorem 1).

The new model formulation is

$$\max \sum_{a \in A} \sum_{i \in I} \sum_{g \in G} \sum_{k \in K} c_k a v_i^a y_{kig}^a \tag{24}$$

with the constraints

$$(2), (4)$$

$$\sum_{i \in I} y_{kig}^a = 1 \quad \forall a \in A, g \in G, k \in K \tag{25}$$

$$\sum_{k \in K} y_{kig}^a = x_{ig} \quad \forall a \in A, i \in I, g \in G \tag{26}$$

$$0 \leq y_{kig}^a \quad \forall a \in A, i \in I, g \in G, k \in K \tag{27}$$

Objective function (24) adds up the shares of all items for all attributes. Although the items are evaluated independently from each other (no d_{ij}^a values), they are only at the first glance independent because the assignment of items to blocks depends on the other items assigned to the same group. Constraints (25) ensure that each group and each block gets for each attribute exactly one item assigned according to which the share in the objective value is determined. Moreover, each item is assigned to exactly one block for each attribute (Constraints (26)). Therefore, $|K| = |I|/|G|$ items are assigned to each group. Constraints (26) set in combination with Constraints (27) also the range of the y_{kig}^a variables. Their value can be between zero and one if item i is assigned to group g . Otherwise they must be zero. The proof of Theorem 1 shows that there is an optimal solution in which all y variables are zero or one.

Theorem 1 proves that the model formulation (2), (4), and (24)–(27) is equivalent to the standard MDGP formulation with attribute values (1)–(4).

Theorem 1 *Let $|K| = |I|/|G|$. Then, the model formulation (2), (4), and (24)–(27) is equivalent to the formulation (1)–(4) in the sense that both model formulations lead to the same objective value for any feasible assignment of the x_{ig} variables.*

Proof Let any feasible solution of (1)–(4) be given. Due to (3) exactly $|K| = |I|/|G|$ items are assigned to each group. In the following, we decompose the instance into one instance per combination of attributes and groups, i.e. given $a \in A$ and $g \in G$. This reduced instance consists of all items assigned to the fixed group g , i.e. the items with $x_{ig} = 1$ in the solution of (1)–(4). Since we consider only a single attribute, we know from Schulz (2021b) that (19) holds with b_{ki}^a according to Assumption 1. Set $y_{kig}^a = b_{ki}^a$ with the fixed $a \in A$ and $g \in G$. Then, (19) implies

$$\sum_{i \in I} \sum_{j \in I: i < j} d_{ij}^a x_{ig} x_{jg} = \sum_{k \in K} \sum_{i \in I} c_k a v_i^a y_{kig}^a.$$

As b_{ki}^a is binary, $0 \leq y_{kig}^a$ for all $i \in I$ with $x_{ig} = 1$ and $k \in K$ (compare (27)). Since the number of blocks equals the number of items in our reduced instance, exactly one item is assigned to each block. Thus, $\sum_{i \in I: x_{ig}=1} y_{kig}^a = 1$ for all $k \in K$ (compare (25)) and $\sum_{k \in K} y_{kig}^a = x_{ig} = 1$ for all $i \in I$ with $x_{ig} = 1$ holds (compare (26)).

We can repeat the procedure for every pair (a, g) with $a \in A$ and $g \in G$. Together,

$$\sum_{a \in A} \sum_{g \in G} \sum_{i \in I} \sum_{j \in I: i < j} d_{ij}^a x_{ig} x_{jg} = \sum_{a \in A} \sum_{i \in I} \sum_{g \in G} \sum_{k \in K} c_k a v_i^a y_{kig}^a,$$

Table 1 Attribute values for the example

i	av_i^1	av_i^2	av_i^3
1	1	1	0.01
2	1	0	0
3	0	1	0
4	0	0	0.01

and (25), (26), and (27) are fulfilled. Moreover, (25) leads to

$$\sum_{k \in K} \sum_{i \in I} y_{kig}^a = |K|$$

$$\stackrel{(26)}{\implies} \sum_{i \in I} x_{ig} = |K| = |I|/|G|,$$

as $|K| = |I|/|G|$, for all $g \in G$ in every feasible solution of (2), (4), and (24)–(27). Thus, every assignment of x_{ig} variables which is feasible for (2), (4), and (24)–(27) is also feasible for (1)–(4). Together the theorem follows. \square

The idea of the proof is that it is still optimal to assign the items assigned to a group in decreasing order of their attribute values to the blocks (compare (23) and the following explanations). This means that, given fixed x_{ig} variables, (1)–(4) and (2), (4), and (24)–(27) are solved optimally if we assign for each attribute and each group g the items assigned to that group, i.e. with $x_{ig} = 1$, in decreasing order of their attribute values to the blocks (set y_{kig}^a accordingly). If (2), (4), and (18) has a feasible solution, $y_{kig}^a = b_{ki}^a \cdot x_{ig}$ holds in an optimal solution for all $i \in I, a \in A, k \in K$, and $g \in G$ (b_{ki}^a according to Assumption 1). If not, a pair of items i and j with $b_{ki}^a = b_{kj}^a = 1$ for an $a \in A$ and a $k \in K$ exists which are assigned to the same group ($x_{ig} = x_{jg} = 1$ for a $g \in G$). Thus, (18) is not fulfilled for all $a \in A, g \in G$, and $k \in K$. In other words, our model (2), (4), and (24)–(27) decides in the assignment of the y variables which block constraints (18) should be violated, if necessary, such that (1) is maximized. Thereby, we avoid the drawback of the block constraints that there might be no feasible solution but still benefit from the formulation on the right side of (19).

Let us consider an example. Let $|A| = 3, |I| = 4$, and $|G| = 2$. Let the attribute values be like in Table 1. If we assign the items according to Assumption 1 to blocks, items 1 and 2 are assigned to the first block according to the first attribute, i.e. $b_{11}^1 = b_{12}^1 = 1$ and $b_{23}^1 = b_{24}^1 = 1$. For the second attribute we get $b_{11}^2 = b_{13}^2 = 1$ and $b_{22}^2 = b_{24}^2 = 1$. For the third attribute items 1 and 4 are in the first block, i.e. $b_{11}^3 = b_{14}^3 = 1$ and $b_{22}^3 = b_{23}^3 = 1$. This means that item 1 is for the first attribute with item 2 in one block, for the second attribute with item 3 and for the third attribute with item 4. Thus, there is no feasible solution fulfilling (2), (4), and (18).

There are three possibilities to assign four items to two groups with two items each. They are

$$x_{11} = x_{21} = 1 : |1 - 1| + |1 - 0| + |0.01 - 0| = 1.01$$

$$x_{32} = x_{42} = 1 : |0 - 0| + |1 - 0| + |0 - 0.01| = 1.01,$$

$$x_{11} = x_{31} = 1 : |1 - 0| + |1 - 1| + |0.01 - 0| = 1.01$$

$$x_{22} = x_{42} = 1 : |1 - 0| + |0 - 0| + |0 - 0.01| = 1.01,$$

and

$$\begin{aligned} x_{11} = x_{41} &= 1 : |1 - 0| + |1 - 0| + |0.01 - 0.01| = 2 \\ x_{22} = x_{32} &= 1 : |1 - 0| + |0 - 1| + |0 - 0| = 2, \end{aligned} \tag{28}$$

where the right side equals the sum of pairwise differences over all attributes of the two items assigned to the group $(\sum_{a=1}^3 \sum_{i=1}^4 \sum_{j:j>i} d_{ij}^a x_{ig} x_{jg})$. Hence, the third solution has the largest objective value such that the model sets $x_{11} = x_{41} = x_{22} = x_{32} = 1$ (beside symmetry) which fulfills (2) and (4).

As we assign two items to each group, $c_1 = 1$ and $c_2 = -1$ (compare (20) and (21)). Thus, (24) is maximized if we set y_{kig}^a variables such that the item with the larger attribute value of the group is in block one and the item with the smaller attribute value of the group is in block two. Hence, the model sets $y_{111}^1 = y_{111}^2 = y_{111}^3 = y_{122}^1 = y_{222}^2 = y_{122}^3 = y_{232}^1 = y_{232}^2 = y_{241}^1 = y_{241}^2 = y_{241}^3 = 1$ and all remaining y_{kig}^a variables to zero. Then, (25)–(27) are fulfilled and (24) equals

$$\begin{aligned} &c_1 \cdot (av_1^1 + av_2^1 + av_1^2 + av_3^2 + av_1^3 + av_2^3) \\ &\quad + c_2 \cdot (av_3^1 + av_4^1 + av_2^2 + av_4^2 + av_3^3 + av_4^3) \\ &= 1 \cdot (1 + 1 + 1 + 1 + 0.01 + 0) - 1 \cdot (0 + 0 + 0 + 0 + 0 + 0.01) \\ &= 4.01 - 0.01 = 4, \end{aligned}$$

which equals the sum in (28). If we compare b_{ki}^a and y_{kig}^a , $y_{kig}^a \neq b_{ki}^a \cdot x_{ig}$ for items 2 and 4 and attribute 3. While item 4 is assigned to block 1 according to Assumption 1, the model assigns item 2 to block 1 (can also be item 3). Thus, the model selects the block constraints (18) which should be violated if necessary (here both block constraints of attribute 3).

4.2 Varying group sizes

We adapt the approach of the previous subsection in this subsection to investigate the relaxation of (3) to (5) and obtain our reformulation of (1)–(2) and (4)–(5).

We replace c_k and \bar{c}_k in the following by $c_{k\bar{k}}$ and $\bar{c}_{k\bar{k}}$, respectively, because individual groups may contain a different number of items. This means that $\bar{c}_{k\bar{k}} = \bar{c}_k$ for $|K| = \bar{k}$ in (21) and (22), respectively, and all $k \in K$. Correspondingly, $c_{k\bar{k}} = c_k$ for $|K| = \bar{k}$ in (20) and all $k \in K$. Note that $\bar{k} \leq \max_g u_g$.

Moreover, we introduce the binary variable $w_{g\bar{k}}$ which is one if group g has \bar{k} blocks, i.e. g has \bar{k} assigned items. Variable y_{kig}^a is replaced by variable $\bar{y}_{kig\bar{k}}^a$ which is defined continuously between zero and one but is one if item i is assigned to block k in group g and group g has \bar{k} assigned items. Both variables, w and \bar{y} , are for all $g \in G$ only defined for $l_g \leq \bar{k} \leq u_g$. Moreover, we fix $\bar{y}_{kig\bar{k}}^a = 0$ for $k > \bar{k}$.

By this, we get the following model formulation:

$$\max \sum_{a \in A} \sum_{i \in I} \sum_{g \in G} \sum_{l_g \leq \bar{k} \leq u_g} \sum_{k=1}^{\bar{k}} c_{k\bar{k}} a v_i^a \bar{y}_{kig\bar{k}}^a \tag{29}$$

$$(2), (4) \quad \sum_{i \in I} \sum_{l_g \leq \bar{k} \leq u_g} \bar{y}_{kig\bar{k}}^a \leq 1 \quad \forall a \in A, g \in G, k \in K : k \leq u_g \quad (30)$$

$$\sum_{l_g \leq \bar{k} \leq u_g} \sum_{k \in K : k \leq \bar{k}} \bar{y}_{kig\bar{k}}^a = x_{ig} \quad \forall a \in A, i \in I, g \in G \quad (31)$$

$$0 \leq y_{kig\bar{k}}^a \leq w_{g\bar{k}} \quad \forall a \in A, i \in I, g \in G, \bar{k} \in \{l_g, \dots, u_g\}, k \in K : k \leq \bar{k} \quad (32)$$

$$\sum_{l_g \leq \bar{k} \leq u_g} \bar{k} \cdot w_{g\bar{k}} = \sum_{i \in I} x_{ig} \quad \forall g \in G \quad (33)$$

$$\sum_{l_g \leq \bar{k} \leq u_g} w_{g\bar{k}} = 1 \quad \forall g \in G \quad (34)$$

$$w_{g\bar{k}} \in \{0, 1\} \quad \forall g \in G, l_g \leq \bar{k} \leq u_g \quad (35)$$

The model sets x_{ig} variables to assign items to groups. Thereby, it ensures that at least l_g and at most u_g items are assigned to group $g \in G$ (33). Because of (33)–(35), $w_{g\bar{k}}$ indicates the number of items in group g . Given $w_{g\bar{k}}$, (32) fixes $\bar{y}_{kig\bar{k}}^a$ for all but one \bar{k} (dependent on g) to zero. So, $\sum_{l_g \leq \bar{k} \leq u_g} (\cdot)$ includes only one non-zero addend in (29)–(31) such that they set $\bar{y}_{kig\bar{k}}^a$ in line with (24)–(26) and the argumentation in the proof of Theorem 1.—Note that we fix $\bar{y}_{kig\bar{k}}^a = 0$ for $k > \bar{k}$. Thus, (30) is an equality for $k \leq \bar{k}$ and the left side of (30) is zero for $k > \bar{k}$.—Hence, the largest attribute values within each group are multiplied with the largest $c_{k\bar{k}}$ values in (29). By this, we are able to prove the following theorem which is an analogon to Theorem 1.

Theorem 2 *The model formulation (2), (4), and (29)–(35) is equivalent to the formulation (1)–(2) and (4)–(5) in the sense that both model formulations lead to the same objective value for any feasible assignment of the x_{ig} variables.*

Proof Let any feasible solution of (1)–(2) and (4)–(5) be given. In the following, we decompose the instance into one instance per combination of attributes and groups, i.e. given $a \in A$ and $g \in G$. This reduced instance consists of all items assigned to the fixed group g , i.e. the items with $x_{ig} = 1$ in the solution of (1)–(2) and (4)–(5). Since x_{ig} variables are fixed, we can set $w_{g\bar{k}} = 1$ for $\bar{k} = \sum_{i \in I} x_{ig}$ and zero for all other \bar{k} s. Thus, (33)–(35) are fulfilled. Furthermore, (32) fixes $y_{kig\bar{k}}^a$ to zero for all but one \bar{k} and all $a \in A, i \in I, g \in G$, and $k \in K$. Thus, \bar{k} is fixed in (29)–(31). Given the fixed \bar{k} , it follows by the same argumentation as in the proof of Theorem 1 that both model formulations, (1)–(2) and (4)–(5) as well as (2), (4), and (29)–(35), have the same objective value for the given solution.

It remains to show that each feasible solution of (2), (4), and (29)–(35) is also feasible for (1)–(2) and (4)–(5). As $w_{g\bar{k}}$ is binary (35), $w_{g\bar{k}} = 1$ for exactly one $l_g \leq \bar{k} \leq u_g$ for all $g \in G$ due to (34). By this, (33) implies that $l_g \leq \sum_{i \in I} x_{ig} \leq u_g$ for all $g \in G$. Thus, (5) is fulfilled. Since (2) and (4) are part of both model formulations, a feasible solution of (2), (4), and (29)–(35) is also feasible for (1)–(2) and (4)–(5). \square

The model formulation (2), (4), and (29)–(35) is clearly a generalization of the setting with equal-sized groups ((2), (4), and (24)–(27)). However, we need a further set of binary variables $w_{g\bar{k}}$ for the number of items within each group. Thus, there is a further generalization of

(2), (4), and (24)–(27) but a special case of (2), (4), and (29)–(35) where we do not need $w_{g\bar{k}}$ variables. If we set $l_g = n_g = u_g$, i.e. fix the number of items assigned to each group (not necessarily with $n_g = n_{g'}$ for all $g, g' \in G$), $\sum_{l_g \leq \bar{k} \leq u_g} (\cdot)$ contains only one addend such that $w_{g,n_g} = 1$ due to (34). So $w_{g\bar{k}}$ is already determined such that the model can be reformulated to omit $w_{g\bar{k}}$ variables. Together with $w_{g\bar{k}}$, \bar{k} is known for each g . Thus, \bar{k} can be omitted in the definition of $y_{ki\bar{k}}^a$ (compare (32)).

5 Computational study

This section describes our computational study which is divided into two parts. In the first part, we consider equal-sized groups and compare the model formulations (1)–(4), called *standard*, (12)–(17), called *Papenberg and Klau*, and (2), (4), and (24)–(27), called *blocks*. In the second part, we consider the case with varying group sizes and the most general formulations of *standard* ((1)–(2) and (4)–(5)) and *blocks* ((2), (4), and (29)–(35)). The models were implemented in *GAMS* (version 32.0) and solved by *CPLEX* (version 12.10). The *standard* model was implemented in the linearized version, i.e. we used the formulations (2)–(4) and (6)–(10) and (2) and (4)–(10), respectively. The symmetry breaking constraints (11) were added to the *standard* model as well as to the *blocks* model when equal-sized groups are considered. The computational study was executed on a single AMD EPYC 7302 core with 2.99GHz. Section 5.1 describes the composition, Sect. 5.2 the results for the case with equal-sized groups, and Sect. 5.3 the results for the case with varying group sizes.

5.1 Composition

We tested the models with instances with 10, 20, 30, 40, 50, 60, 70, and 80 items. All of them were distributed into 2 and 5 groups. Starting with 20 items they were also distributed in 10 groups. 60, 70, and 80 items were further distributed into 15 groups and 80 items into 20 groups. For the case with equal-sized groups we used only those settings where the number of items divided by the number of groups is integer. All settings were tested in 30 runs each for 3, 5, and 10 attributes. Attribute values were determined according to a [0,1] uniform distribution. Considering [0,1] values is no restriction, as we are only interested in their absolute differences and each other interval can be normalized to [0,1] without changing the relation of two attribute values. Before we determined the range for the group sizes (l_g and u_g) for the second part of the computational study, we distributed the items randomly to the groups according to the following procedure to ensure that there is a feasible assignment: First, we draw $|G| - 1$ uniform integers between 1 and $|I| - |G|$. They were sorted in increasing order and we computed the differences between 0 and the first, the first and the second, and so on and added 1 to all of them. Moreover, we computed the difference between the sum of them and $|I|$. Thereby, we ensured that each difference is positive. In total, this leads to $|G|$ numbers which sum up to $|I|$. Finally, we set n_1 equal to the first of them, n_2 equal to the second, and so on. Afterwards, we determined l_g and u_g . We set $l_g = \max(1, n_g - \mathcal{U}\{0, \lceil n_g \cdot 0.05 \rceil\})$ and $u_g = n_g + \mathcal{U}\{0, \lceil n_g \cdot 0.05 \rceil\}$, i.e. the group size interval is determined uniformly and bounded by about 10% around n_g . We interrupted the search after 30 minutes (1800 seconds) if no solution was proved to be optimal like it was done by Gallego et al. (2013).

Table 2 Results for three attributes

		<i>blocks</i>						<i>standard</i>						<i>Papenberg and Klau</i>					
<i>I</i>	<i> G </i>	\varnothing_{obv}	\varnothing_{time} [s]	\varnothing_{gap} [%]	$\varnothing_{rel.}$	# <i>fea.</i>	# <i>opt.</i>	\varnothing_{obv}	\varnothing_{time} [s]	\varnothing_{gap} [%]	$\varnothing_{rel.}$	# <i>fea.</i>	# <i>opt.</i>	\varnothing_{obv}	\varnothing_{time} [s]	\varnothing_{gap} [%]	$\varnothing_{rel.}$	# <i>fea.</i>	# <i>opt.</i>
10	2	21.97	0.02	-	659.98	30	30	21.97	0.17	-	1213.03	30	30	21.97	0.05	-	674.07	30	30
10	5	6.77	0.03	-	204.28	30	30	6.77	0.43	-	758.23	30	30	6.77	0.02	-	203.06	30	30
15	5	18.73	0.06	-	563.48	30	30	18.73	182.58	-	2268.10	30	30	18.73	0.58	-	610.02	30	30
20	2	94.48	0.06	-	2835.97	30	30	94.48	16.24	-	5408.70	30	30	94.48	6.39	-	2993.25	30	30
20	5	35.43	0.13	-	1064.62	30	30	35.30	1800.00	91.5	4470.37	30	0	35.43	111.68	-	1207.32	30	30
20	10	14.22	0.23	-	428.64	30	30	14.22	362.89	-	3069.97	30	30	14.22	0.04	-	426.64	30	30
25	5	58.35	0.33	-	1752.25	30	30	58.01	1800.00	166.2	7605.59	30	0	58.24	1800.00	3.5	2054.07	30	0
30	2	215.51	0.12	-	6466.82	30	30	215.46	1800.00	7.3	12542.54	30	0	215.46	1799.21	1.3	6913.79	30	1
30	5	83.95	1.45	-	2520.29	30	30	83.14	1800.00	219.1	11184.52	30	0	83.41	1800.00	8.3	3014.44	30	0
30	10	39.16	2.52	-	1175.20	30	30	38.42	1800.00	480.4	9119.20	30	0	39.15	1800.00	3.1	1314.72	30	0
40	2	391.96	0.28	-	11760.53	30	30	391.73	1800.00	29.0	22952.53	30	0	391.42	1800.00	5.6	12669.75	30	0
40	5	151.86	12.48	-	4557.15	30	30	149.54	1800.00	278.3	20704.96	30	0	147.19	1800.00	14.8	5609.97	30	0
40	10	73.84	23.75	-	2215.32	30	30	71.76	1800.00	637.6	18224.76	30	0	73.40	1800.00	10.7	2643.04	30	0
50	2	607.96	0.54	-	18240.95	30	30	607.46	1800.00	47.1	35793.01	30	0	606.60	1800.00	7.4	19724.84	30	0
50	5	245.57	142.80	-	7368.60	30	30	242.07	1800.00	313.5	34229.67	30	0	233.84	1800.00	19.6	9266.47	30	0
50	10	117.90	137.86	-	3536.96	30	30	112.19	1800.00	736.1	30108.21	30	0	112.67	1800.00	19.3	4344.24	30	0
60	2	888.15	1.21	-	26646.71	30	30	887.16	1800.00	61.0	52409.99	30	0	875.08	1800.00	9.9	28927.14	30	0
60	5	353.16	1174.75	<0.1	10596.86	30	20	348.56	1800.00	334.5	49696.11	30	0	331.60	1800.00	22.6	13475.99	30	0
60	10	173.07	603.52	<0.1	5192.12	30	25	164.89	1800.00	775.4	45196.04	30	0	160.99	1800.00	26.3	6570.76	30	0
60	15	109.97	104.80	<0.1	3299.15	30	29	102.72	1800.00	1175.5	40270.63	30	0	104.79	1800.00	19.9	3945.35	30	0

Table 2 continued

		<i>blocks</i>										<i>standard</i>										<i>Papenberg and Klau</i>									
I	G	øbv	øtime [s]	øgap [%]	ørel.	#fea.	#opt.	øbv	øtime [s]	øgap [%]	ørel.	#fea.	#opt.	øbv	øtime [s]	øgap [%]	ørel.	#fea.	#opt.	øbv	øtime [s]	øgap [%]	ørel.	#fea.	#opt.						
70	2	1211.24	2.31	-	36339.60	30	30	1209.82	1800.00	68.6	71687.72	30	0	1191.58	1800.00	10.3	39526.32	30	0												
70	5	483.32	1751.74	<0.1	14503.63	30	1	474.51	1800.00	353.8	68664.04	30	0	-	1800.00	-	18559.81	0	0												
70	10	237.36	1298.25	<0.1	7121.06	30	14	227.23	1800.00	803.5	58326.17	30	0	213.30	1800.00	33.0	9183.57	30	0												
80	2	1580.26	6.25	-	47410.24	30	30	1578.44	1800.00	75.2	93675.70	30	0	1559.68	1800.00	10.4	51685.89	30	0												
80	5	622.41	1800.00	<0.1	18679.73	30	0	611.37	1800.00	365.8	88982.24	30	0	592.63	1800.00	26.7	24017.26	30	0												
80	10	311.23	1794.59	<0.1	9338.80	30	1	299.17	1800.00	827.7	66523.41	30	0	281.41	1800.00	34.8	12241.70	30	0												
80	20	149.84	187.53	-	4495.30	30	30	135.13	1800.00	1693.2	52294.99	30	0	141.01	1800.00	25.0	5409.30	30	0												

Table 3 Results for five attributes

		<i>blocks</i>						<i>standard</i>						<i>Papenberg and Klau</i>					
<i>I</i>	<i> G </i>	ϕ_{obv}	ϕ_{time} [s]	ϕ_{gap} [%]	ϕ_{rel}	#fea.	#opt.	ϕ_{obv}	ϕ_{time} [s]	ϕ_{gap} [%]	ϕ_{rel}	#fea.	#opt.	ϕ_{obv}	ϕ_{time} [s]	ϕ_{gap} [%]	ϕ_{rel}	#fea.	#opt.
10	2	36.19	0.03	-	1091.06	30	30	36.19	0.16	-	2029.61	30	30	36.19	0.03	-	1095.78	30	30
10	5	11.02	0.04	-	335.80	30	30	11.02	0.43	-	1266.06	30	30	11.02	0.02	-	330.75	30	30
15	5	30.20	0.14	-	919.55	30	30	30.20	217.54	-	3750.54	30	30	30.20	0.30	-	948.45	30	30
20	2	154.92	0.15	-	4658.38	30	30	154.92	17.98	-	8915.64	30	30	154.92	4.21	-	4814.06	30	30
20	5	59.96	0.88	-	1820.59	30	30	59.62	1800.00	102.2	7670.01	30	0	59.96	14.40	-	1964.01	30	30
20	10	23.35	0.46	-	715.97	30	30	23.35	300.28	-	5075.90	30	30	23.35	0.04	-	700.56	30	30
25	5	95.84	7.93	-	2899.82	30	30	94.87	1800.00	177.0	12614.21	30	0	95.70	1764.46	1.3	3219.36	30	4
30	2	354.50	0.48	-	10649.14	30	30	354.30	1800.00	8.9	20642.33	30	0	354.48	1682.09	0.6	11143.45	30	12
30	5	140.51	91.32	-	4245.17	30	30	138.49	1800.00	225.6	18853.90	30	0	139.30	1800.00	5.3	4811.09	30	0
30	10	63.20	54.01	-	1922.79	30	30	61.30	1800.00	502.5	14928.45	30	0	63.20	564.22	1.0	2050.62	30	29
40	2	642.92	3.07	-	19304.75	30	30	642.25	1800.00	32.6	37701.98	30	0	641.82	1800.00	4.0	20364.82	30	0
40	5	257.76	1635.59	0.2	7769.51	30	11	253.43	1800.00	283.1	35361.47	30	0	250.32	1800.00	10.8	9041.08	30	0
40	10	119.67	1691.32	0.4	3620.68	30	7	115.73	1800.00	653.0	29739.24	30	0	118.43	1800.00	6.3	4094.12	30	0
50	2	1019.52	13.64	-	30603.70	30	30	1018.21	1800.00	50.1	60016.17	30	0	1017.23	1800.00	5.5	32409.31	30	0
50	5	403.39	1800.00	0.3	12152.30	30	0	397.31	1800.00	317.2	56358.92	30	0	389.90	1800.00	13.5	14388.15	30	0
50	10	193.76	1800.00	0.9	5871.12	30	0	183.08	1800.00	749.7	49976.37	30	0	185.56	1800.00	13.9	6852.54	30	0
60	2	1478.92	73.04	-	44386.95	30	30	1476.65	1800.00	62.6	87352.88	30	0	1457.86	1800.00	7.5	47142.96	30	0
60	5	581.45	1800.00	0.4	17514.82	30	0	572.50	1800.00	338.9	82199.86	30	0	549.77	1800.00	17.3	20962.62	30	0
60	10	285.27	1800.00	0.8	8630.99	30	0	270.12	1800.00	794.2	75316.32	30	0	269.18	1800.00	18.9	10358.89	30	0
60	15	182.56	1800.00	1.0	5535.19	30	0	168.83	1800.00	1202.5	67395.72	30	0	174.58	1800.00	13.7	6390.26	30	0

Table 3 continued

		<i>blocks</i>										<i>standard</i>										<i>Papenberg and Klau</i>									
I	G	øbv	øtime [s]	øgap [%]	ørel.	#fea.	#opt.	øbv	øtime [s]	øgap [%]	ørel.	#fea.	#opt.	øbv	øtime [s]	øgap [%]	ørel.	#fea.	#opt.	øbv	øtime [s]	øgap [%]	ørel.	#fea.	#opt.						
70	2	2033.00	358.75	–	61009.91	30	30	2029.71	1800.00	69.8	120302.46	30	0	2004.71	1800.00	7.9	64998.03	30	0	–	1800.00	–	29263.73	0	0						
70	5	804.31	1800.00	0.3	24214.99	30	0	788.10	1800.00	356.0	114501.83	30	0	–	1800.00	–	29263.73	0	0	–	1800.00	–	29263.73	0	0						
70	10	392.39	1800.00	0.9	11878.43	30	0	373.89	1800.00	817.1	96195.44	30	0	354.67	1800.00	26.7	14545.30	30	0	–	1800.00	–	14545.30	30	0						
80	2	2622.76	1557.94	<0.1	78704.08	30	13	2618.49	1800.00	76.0	155482.51	30	0	2592.67	1800.00	7.9	83918.75	30	0	–	1800.00	–	83918.75	30	0						
80	5	1045.58	1800.00	0.3	31470.94	30	0	1025.68	1800.00	369.3	150006.38	30	0	998.00	1800.00	21.0	38177.23	30	0	–	1800.00	–	38177.23	30	0						
80	10	510.91	1800.00	0.9	15461.65	30	0	488.07	1800.00	841.3	109314.48	30	0	465.04	1800.00	27.7	19132.16	30	0	–	1800.00	–	19132.16	30	0						
80	20	242.87	1800.00	1.4	7384.97	30	0	218.47	1800.00	1726.7	84743.63	30	0	229.54	1800.00	18.6	8627.30	30	0	–	1800.00	–	8627.30	30	0						

Table 4 Results for ten attributes

		<i>blocks</i>										<i>standard</i>										<i>Papenberg and Klau</i>									
<i>I</i>	<i> G </i>	ϕ obv	ϕ time [s]	ϕ gap [%]	ϕ rel.	#fea.	#opt.	ϕ obv	ϕ time [s]	ϕ gap [%]	ϕ rel.	#fea.	#opt.	ϕ obv	ϕ time [s]	ϕ gap [%]	ϕ rel.	#fea.	#opt.	ϕ obv	ϕ time [s]	ϕ gap [%]	ϕ rel.	#fea.	#opt.						
10	2	69.81	0.05	-	2117.50	30	30	69.81	0.16	-	3930.61	30	30	69.81	0.03	-	2099.47	30	30												
10	5	20.72	0.08	-	648.82	30	30	20.72	0.43	-	2524.75	30	30	20.72	0.02	-	621.74	30	30												
15	5	59.00	0.83	-	1851.12	30	30	59.00	313.57	-	7557.56	30	30	59.00	0.23	-	1814.01	30	30												
20	2	308.27	0.49	-	9307.87	30	30	308.27	21.97	-	17820.85	30	30	308.27	2.53	-	9418.38	30	30												
20	5	114.83	16.61	-	3560.28	30	30	114.10	1800.00	112.9	15059.53	30	0	114.83	4.34	-	3621.97	30	30												
20	10	43.04	5.17	-	1390.25	30	30	43.04	410.69	-	10102.06	30	30	43.04	0.04	-	1291.45	30	30												
25	5	186.23	262.13	-	5728.66	30	30	183.79	1800.00	187.8	24944.79	30	0	186.23	389.53	-	5987.54	30	30												
30	2	721.83	5.48	-	21736.02	30	30	721.25	1800.00	11.5	42154.15	30	0	721.83	723.38	-	22268.24	30	30												
30	5	277.13	1734.75	0.6	8484.42	30	2	273.16	1800.00	233.9	37652.48	30	0	275.57	1800.00	2.6	9047.56	30	0												
30	10	122.55	1555.92	1.5	3860.29	30	12	118.69	1800.00	533.8	30207.75	30	0	122.66	32.97	-	3850.34	30	30												
40	2	1293.74	72.70	-	38907.60	30	30	1291.98	1800.00	35.3	76003.81	30	0	1291.69	1800.00	2.3	40142.41	30	0												
40	5	506.68	1800.00	1.2	15444.06	30	0	498.00	1800.00	291.0	70353.10	30	0	496.10	1800.00	6.8	16870.41	30	0												
40	10	234.95	1800.00	3.1	7329.97	30	0	226.66	1800.00	674.2	60318.76	30	0	234.55	1800.00	3.1	7690.31	30	0												
50	2	2043.11	1025.97	0.0	61398.72	30	28	2039.90	1800.00	52.1	120435.03	30	0	2037.86	1800.00	3.6	63592.60	30	0												
50	5	796.07	1800.00	1.2	24202.85	30	0	783.98	1800.00	322.2	112309.58	30	0	775.94	1800.00	8.8	26856.48	30	0												
50	10	375.44	1800.00	3.3	11672.50	30	0	356.27	1800.00	767.9	99365.05	30	0	364.95	1800.00	8.9	12669.69	30	0												
60	2	2949.70	1800.00	0.1	88611.40	30	0	2944.66	1800.00	64.1	174406.77	30	0	2908.84	1800.00	5.4	92117.62	30	0												
60	5	1153.05	1800.00	1.1	34985.69	30	0	1133.52	1800.00	344.2	164403.34	30	0	1104.82	1800.00	11.8	39244.31	30	0												
60	10	557.56	1800.00	3.0	17253.67	30	0	532.11	1800.00	808.1	150365.61	30	0	537.77	1800.00	12.0	19193.60	30	0												
60	15	352.78	1800.00	4.6	11095.39	30	0	328.79	1800.00	1241.3	135089.75	30	0	344.64	1800.00	8.4	11879.20	30	0												

Table 4 continued

		<i>standard</i>										<i>Papenberg and Klau</i>									
<i>blocks</i>		ϕ obv	ϕ time [s]	ϕ gap [%]	ϕ rel.	#fea.	#opt.	ϕ obv	ϕ time [s]	ϕ gap [%]	ϕ rel.	#fea.	#opt.	ϕ obv	ϕ time [s]	ϕ gap [%]	ϕ rel.	#fea.	#opt.		
70	2	4025.23	1800.00	0.1	120896.17	30	0	4017.58	1800.00	71.3	238504.07	30	0	3976.17	1800.00	5.4	125914.47	30	0		
70	5	1585.95	1800.00	1.0	48051.33	30	0	1552.94	1800.00	360.7	227587.04	30	0	—	1800.00	—	54359.19	0	0		
70	10	769.80	1800.00	2.6	23715.61	30	0	733.53	1800.00	834.6	192256.67	30	0	711.90	1800.00	18.6	26844.41	30	0		
80	2	5255.75	1800.00	0.1	157833.55	30	0	5245.50	1800.00	77.2	311960.81	30	0	5195.27	1800.00	5.7	164717.50	30	0		
80	5	2081.25	1800.00	0.8	62962.14	30	0	2038.44	1800.00	372.5	300017.59	30	0	2001.72	1800.00	14.9	71639.86	30	0		
80	10	1011.10	1800.00	2.3	31022.22	30	0	968.84	1800.00	850.1	223988.53	30	0	931.04	1800.00	20.4	35578.17	30	0		
80	20	471.45	1800.00	4.9	14841.90	30	0	426.81	1800.00	1777.4	171909.07	30	0	457.45	1800.00	11.4	16088.42	30	0		

5.2 Results (equal-sized groups)

Tables 2–4 present the results with equal-sized groups split according to the number of attributes. The tables are structured as follows: The first two columns indicate the parameter setting. The next six columns show the results for the *blocks* model (average objective value, average computation time in seconds, average gap in percent, average objective value of the relaxed model, number of feasible solutions found, and number of optimal solutions found). The next six columns present the same classifications numbers for the *standard* model, the last six for the model by *Papenberg and Klau*. Note that average objective values contain only instances where the corresponding model found a feasible solution. Average gaps contain only instances where the corresponding model found a feasible but no proven optimal solution.

Table 2 presents the results for three attributes. The model formulations *blocks* and *standard* found for all instances a feasible solution. The formulation by *Papenberg and Klau* found also for all settings beside the one with 70 items and 5 groups a feasible solution. However, for the setting with 70 items and 5 groups no feasible solution was found. Having a closer look on the problem setting, finding a feasible solution is easy because any assignment with exactly $|I|/|G|$ items in each group is feasible. The reason why we have not found a feasible solution for the model formulation by *Papenberg and Klau* seems to be that CPLEX used the whole 1800s in the root node such that no solution was detected (also for five and ten attributes).

blocks terminated for almost all instances up to 60 items with a proven optimal solution within 1800s and terminated also for larger instances often with a proven optimal solution before the time limit was reached. The two other formulations, however, found only optimal solutions for small instance sizes up to 20 (*standard*) and 30 items (*Papenberg and Klau*), respectively. Both models reached comparable results regarding the number of optimally solved instances.

Considering the solution gap for the instances for which we found feasible solutions but could not prove optimality, the model formulation by *Papenberg and Klau* clearly outperformed the *standard* formulation. For the *blocks* model the gap was almost zero for all instances which could not be solved to proven optimality. The required computation times confirm the results for the three model formulations.

In total, we can conclude that the *blocks* model leads for instances with three attributes to the best results in comparison with the *standard* model and the formulation by *Papenberg and Klau*. If we solve the *blocks* model but relax (4), i.e. with $0 \leq x_{ig} \leq 1$ for all $i \in I$ and $g \in G$ instead of (4), we find a reason for it. Column 6 shows the average objective values for the relaxed model of *blocks*, column 12 the corresponding average objective values for the *standard* model and column 18 for the formulation by *Papenberg and Klau* (relaxing (17)). Relaxing *blocks* leads to substantially better upper bounds for the objective value than relaxing *standard*. Relaxing the formulation by *Papenberg and Klau* leads to slightly worse upper bounds than the *blocks* formulation. However, the difference between the average optimal objective value and the average optimal objective value of the relaxed model is still large. Thus, we can assume that we have to fix a large number of x_{ig} variables to zero or one before we get tight upper bounds by relaxing the remaining binary constraints in a branch-and-bound procedure. This may also explain why larger instances could not be solved reliably to optimality.

Tables 3 and 4 show the results for five and ten attributes, respectively. For the *standard* model the results are similar for all three numbers of attributes. The performance of the *blocks* model decreases. The larger the number of attributes is the lower is the number of solutions where we proved optimality within 1800s. However, the gap is still comparably

small with under 5% on average. In contrast, the formulation by *Papenberg and Klau* leads to better results regarding the number of proven optimal solutions found and the gap the larger the number of attributes is. Interestingly, the model by *Papenberg and Klau* requires less computation time if the number of attributes increases. For both of the other models computation times increase if the number of attributes increases. Because of this the model by *Papenberg and Klau* is faster than *blocks* for some settings with up to 30 items—especially if the number of attributes increases.

5.3 Results (varying group sizes)

Tables 5–7 present the results regarding the setting with varying group sizes. Analogous to Sect. 5.2 the tables are split according to the number of attributes. The tables are structured as in Sect. 5.2. As the formulation by *Papenberg and Klau* works only for equal-sized groups, we consider only the *blocks* and the *standard* formulation in this subsection.

Table 5 presents the results for three attributes. Both model formulations found for almost all instances a feasible solution but the *blocks* model found clearly more proven optimal solutions. For up to 50 items even all instances could be solved to optimality while *standard* could only solve instances with two groups or 10 items to optimality. This results also in smaller computation times for the *blocks* model in comparison with the *standard* model. Moreover, the *blocks* formulation reached small gaps after 1800s for up to 70 items and gaps up to 8.1% on average for instances with up to 80 items.

Table 6 examines the performance of the models for five attributes. For five attributes *blocks* still managed to find feasible solutions for almost all instances but had trouble in proving optimality for larger numbers of groups (at least 5) in combination with larger numbers of items (at least 40). From 60 items on the model further struggled with instances with two groups. Nevertheless, there is again a transition where the model found less optimal solutions but reached small gaps (40 or 50 items and 5 or 10 groups, and 60, 70 or 80 items and 2 or 5 groups).

The trend confirms for ten attributes (Table 7). Although *blocks* still found in at least 75% of the instances a feasible solution in each setting (23 out of 30 for 80 items and 2 groups), it found only for small instances up to 20 items always a proven optimal solution. Accordingly, the gaps increased for larger instances. However, there is again a transition where the model did not manage to find proven optimal solutions for all instances but reached small gaps (25 items and 10 groups, 30 or 40 items and 5 groups, and 50 items and 2 groups). The development of the solution quality for *standard* is similar as in the setting with equal-sized groups although the model is able to prove some solutions to be optimal for larger instances with two groups. A reason could be that l_g and u_g and therefore the group size is determined randomly such that some instances might be particularly easy to solve if one of the groups has only a small number of items assigned while the other one has a large number of assigned items. In comparison with the setting with equal-sized groups, the *blocks* model had more difficulties with the setting with varying group sizes which results in less instances which could be solved to proven optimality within 1800s and larger gaps for those which could not be solved to proven optimality.

Table 5 Results for three attributes

I	G	<i>blocks</i>						<i>standard</i>					
		ø obv.	ø time [s]	ø gap [%]	ø cl.	#fea.	#opt.	ø obv.	ø time [s]	ø gap [%]	ø rel.	#fea.	#opt.
10	2	27.37	0.14	-	954.38	30	30	27.37	0.29	-	1352.65	30	30
10	5	11.13	0.30	-	494.51	30	30	11.13	7.38	-	1335.02	30	30
15	2	63.75	0.62	-	2115.03	30	30	63.75	0.89	-	3132.40	30	30
15	5	28.41	1.67	-	1137.44	30	30	28.39	1495.18	26.1	3125.46	30	11
20	2	119.82	1.17	-	3879.55	30	30	119.82	11.80	-	5702.18	30	30
20	5	55.20	3.51	-	1925.30	30	30	54.94	1800.00	67.8	5679.19	30	0
20	10	24.99	8.28	-	1085.54	30	30	24.53	1800.00	300.7	5707.16	30	0
25	2	191.41	2.61	-	6148.52	30	30	191.41	184.64	-	8736.65	30	30
25	5	90.41	6.84	-	3064.02	30	30	89.71	1800.00	105.4	9089.92	30	0
25	10	43.05	19.99	-	1677.86	30	30	41.70	1800.00	380.2	9114.45	30	0
30	2	318.57	4.96	-	10199.24	30	30	318.56	466.84	6.4	13023.59	30	23
30	5	143.08	14.70	-	4764.42	30	30	142.05	1800.00	104.3	13021.64	30	0
30	10	67.56	41.71	-	2521.86	30	30	64.55	1800.00	398.5	12892.26	30	0
40	2	533.45	15.67	-	17383.68	30	30	533.36	1259.03	16.2	23418.57	30	11
40	5	258.20	45.73	-	8416.88	30	30	253.89	1800.00	140.6	23460.00	30	0
40	10	127.03	112.09	-	4518.13	30	30	120.12	1800.00	457.0	23294.28	30	0
50	2	832.47	41.26	-	27068.53	30	30	832.03	1461.57	27.4	36826.71	30	7
50	5	452.63	93.76	-	14608.42	30	30	443.30	1800.00	135.0	36729.64	30	0
50	10	228.86	257.73	-	7736.84	30	30	215.48	1800.00	429.3	37232.32	30	0
60	2	1172.47	119.49	-	38325.02	30	30	1171.28	1590.92	35.1	53198.36	30	4
60	5	613.33	230.63	-	19753.63	30	30	597.48	1800.00	165.4	53017.18	30	0
60	10	326.38	500.90	<0.1	10750.66	30	29	303.70	1800.00	457.3	53394.14	30	0
60	15	204.50	913.06	2.4	7093.32	30	23	182.20	1800.00	816.5	47888.75	30	0
70	2	1807.25	112.20	-	57568.33	30	30	1804.49	1682.58	21.4	72203.82	30	3
70	5	796.09	392.29	-	25608.82	30	30	769.41	1800.00	192.6	72510.69	30	0

Table 5 continued

I	G	blocks						standard					
		ϕ obv.	ϕ time [s]	ϕ gap [%]	ϕ el.	#fea.	#opt.	ϕ obv.	ϕ time [s]	ϕ gap [%]	ϕ rel.	#fea.	#opt.
70	10	429.90	975.55	2.1	14128.28	30	25	399.86	1800.00	486.1	70388.98	30	0
70	15	289.62	1584.31	2.2	9864.70	30	9	253.42	1800.00	833.8	57208.58	30	0
80	2	2258.74	500.15	–	70567.62	26	26	2192.46	1677.70	35.9	94792.09	30	3
80	5	1065.04	681.67	4.0	34388.67	30	25	1025.94	1800.00	195.3	94845.75	30	0
80	10	612.65	1404.00	3.9	19954.06	30	14	559.52	1800.00	449.6	80811.59	30	0
80	15	370.98	1714.26	5.8	12546.61	30	4	320.83	1800.00	866.6	64830.48	30	0
80	20	275.27	1800.00	8.1	9546.22	30	0	218.92	1800.00	7485.2	62658.90	25	0

Table 6 Results for five attributes

I	G	<i>blocks</i>					<i>standard</i>						
		ø obv.	ø time [s]	ø gap [%]	ø rel.	#fea.	#opt.	ø obv.	ø time [s]	ø gap [%]	ø rel.	#fea.	#opt.
10	2	47.25	0.18	–	1570.19	30	30	47.25	0.31	–	2231.53	30	30
10	5	16.65	0.57	–	772.64	30	30	16.65	11.45	–	2251.01	30	30
15	2	109.10	1.05	–	3664.85	30	30	109.10	1.18	–	5250.76	30	30
15	5	47.43	4.30	–	1877.36	30	30	47.41	1442.59	31.6	5216.91	30	10
20	2	219.51	2.59	–	7007.82	30	30	219.51	9.37	–	9436.21	30	30
20	5	88.37	9.77	–	3122.87	30	30	87.85	1800.00	77.8	9459.22	30	0
20	10	39.53	28.81	–	1806.49	30	30	38.62	1800.00	323.8	9473.85	30	0
25	2	330.89	7.40	–	10499.55	30	30	330.89	185.15	–	14981.36	30	30
25	5	156.73	27.26	–	5397.12	30	30	155.50	1800.00	92.7	14842.85	30	0
25	10	71.17	78.93	–	2912.49	30	30	68.75	1800.00	382.4	14892.39	30	0
30	2	464.14	14.26	–	15072.78	30	30	464.06	816.64	10.5	21586.12	30	20
30	5	223.26	52.08	–	7394.35	30	30	221.19	1800.00	122.6	21811.88	30	0
30	10	108.35	316.04	–	4164.12	30	30	103.49	1800.00	443.1	22034.13	30	0
40	2	896.92	43.80	–	28647.96	30	30	896.72	1263.03	17.9	38962.75	30	11
40	5	447.95	351.97	0.1	14613.97	30	29	440.46	1800.00	133.7	38737.59	30	0
40	10	209.04	1266.82	0.3	7478.51	30	15	198.75	1800.00	465.1	39268.43	30	0
50	2	1352.41	75.45	–	42448.01	30	30	1351.53	1627.87	25.4	61302.05	30	3
50	5	682.37	593.05	0.5	21901.33	30	26	666.50	1800.00	165.9	61635.37	30	0
50	10	339.60	1703.60	1.5	11632.39	30	3	317.34	1800.00	495.2	61383.39	30	0
60	2	1993.35	281.04	<0.1	62600.07	29	28	1970.70	1599.86	32.9	87593.07	30	4
60	5	990.79	1066.41	2.3	31849.09	30	20	966.78	1800.00	176.5	88574.75	30	0

Table 6 continued

I	G	<i>blocks</i>						<i>standard</i>					
		ø obv.	ø time [s]	ø gap [%]	ø rel.	#fea.	#opt.	ø obv.	ø time [s]	ø gap [%]	ø rel.	#fea.	#opt.
60	10	491.47	1800.00	3.1	16343.83	30	0	459.41	1800.00	511.6	88443.04	30	0
60	15	315.71	1800.00	10.8	11217.79	30	0	282.47	1800.00	915.0	79950.22	30	0
70	2	2928.92	365.18	<0.1	93136.20	29	28	2903.32	1564.48	31.8	121175.62	30	5
70	5	1287.42	1327.38	1.9	41261.00	30	11	1243.97	1800.00	205.5	121308.77	30	0
70	10	718.51	1800.00	4.5	23508.69	30	0	669.39	1800.00	482.6	116375.11	30	0
70	15	447.70	1800.00	9.6	15181.32	30	0	396.60	1800.00	896.5	95677.90	30	0
80	2	3745.21	784.61	2.8	116948.82	26	21	3645.91	1565.23	39.5	157594.50	30	4
80	5	1730.18	1694.40	2.8	56637.03	30	4	1669.27	1800.00	199.2	156623.22	30	0
80	10	922.05	1800.00	6.9	30199.63	30	0	839.18	1800.00	518.6	131471.22	30	0
80	15	644.24	1800.00	12.2	21964.88	30	0	545.46	1800.00	854.8	109329.60	30	0
80	20	451.91	1800.00	16.4	15967.47	30	0	360.79	1800.00	7695.2	97557.97	25	0

Table 7 Results for ten attributes

I	G	<i>blocks</i>					<i>standard</i>						
		ø obv.	ø time [s]	ø gap [%]	ø rel.	#fea.	#opt.	ø obv.	ø time [s]	ø gap [%]	ø rel.	#fea.	#opt.
10	2	92.45	0.50	—	3124.88	30	30	92.45	0.28	—	4447.93	30	30
10	5	33.87	2.70	—	1773.58	30	30	33.87	14.72	—	4541.32	30	30
15	2	233.90	2.89	—	7869.89	30	30	233.90	0.99	—	10535.80	30	30
15	5	95.41	14.39	—	3728.20	30	30	95.28	1363.85	36.1	10516.24	30	11
20	2	402.69	22.59	—	12982.61	30	30	402.69	15.94	—	18958.03	30	30
20	5	184.24	70.87	—	6834.57	30	30	183.51	1800.00	74.0	18867.01	30	0
20	10	71.23	165.74	—	3623.15	30	30	69.54	1800.00	384.2	18874.16	30	0
25	2	667.43	22.03	—	21212.19	30	30	667.43	304.81	—	29830.46	30	30
25	5	312.29	249.56	—	10777.35	30	30	310.38	1800.00	98.8	30016.66	30	0
25	10	139.23	925.79	1.6	6026.22	30	22	134.73	1800.00	394.3	30086.60	30	0
30	2	908.04	116.33	—	30125.89	30	30	907.85	1091.14	9.0	43424.08	30	14
30	5	459.04	837.80	0.6	15474.93	30	25	454.63	1800.00	124.5	43507.28	30	0
30	10	211.85	1603.26	2.6	8093.75	30	5	202.76	1800.00	452.6	43660.87	30	0
40	2	1737.56	292.09	5.4	56229.51	30	28	1739.10	1537.54	16.1	77253.16	30	6
40	5	914.20	1271.62	1.3	30123.34	30	13	902.48	1800.00	130.1	78049.22	30	0
40	10	433.05	1782.01	9.4	15404.98	30	1	412.83	1800.00	445.2	78180.08	30	0
50	2	2539.02	614.31	1.2	82492.29	29	26	2545.43	1589.95	36.3	122481.35	30	4
50	5	1383.33	1730.89	4.1	45610.33	30	3	1358.65	1800.00	160.8	122475.62	30	0
50	10	658.05	1800.00	13.8	23091.94	30	0	619.12	1800.00	507.2	122196.88	30	0
60	2	4014.60	1103.84	4.0	123611.71	25	17	3949.80	1683.79	32.8	177081.58	30	3
60	5	1814.40	1735.80	5.9	59607.58	30	2	1787.89	1800.00	199.6	176343.94	30	0
60	10	950.74	1800.00	12.5	32364.00	30	0	897.64	1800.00	527.7	177091.11	30	0

Table 7 continued

I	G	<i>blocks</i>						<i>standard</i>					
		ø obv.	ø time [s]	ø gap [%]	ø rel.	#fea.	#opt.	ø obv.	ø time [s]	ø gap [%]	ø rel.	#fea.	#opt.
60	15	624.85	1800.00	19.8	22660.76	30	0	575.03	1800.00	884.2	161687.85	30	0
70	2	5337.30	1152.24	5.7	176013.41	25	16	5409.12	1533.92	43.5	242136.32	30	5
70	5	2467.38	1800.00	6.0	143006.97	28	0	2459.64	1800.00	205.5	240095.04	30	0
70	10	1337.91	1800.00	11.9	45546.74	30	0	1269.25	1800.00	512.9	236808.61	30	0
70	15	846.11	1800.00	24.0	31726.13	30	0	799.43	1800.00	884.7	190436.62	30	0
80	2	7122.01	1253.69	4.0	231628.53	23	11	7107.17	1705.10	39.0	315359.44	30	2
80	5	3504.99	1800.00	9.0	117203.56	30	0	3511.73	1800.00	188.0	316142.62	30	0
80	10	1697.25	1800.00	13.4	58273.83	30	0	1654.35	1800.00	527.1	261393.30	30	0
80	15	1048.96	1800.00	26.3	40305.15	27	0	1042.43	1800.00	894.6	220816.06	30	0
80	20	781.70	1800.00	37.1	32278.40	30	0	739.62	1800.00	8559.1	199197.56	28	0

6 Conclusion

We introduced a new mixed-integer programming formulation for the MDGP with attribute values. As common applications like the assignment of students to groups (grades, gender, international or not) or surgery scheduling (surgery duration) use attribute values, this is an interesting special case of the MDGP. Nevertheless, using the Manhattan metric is a limitation of our study. Our MIP clearly outperforms comparable approaches for instances with attribute values. Moreover, it is able to solve instances of realistic size, for example in the assignment of students to seminars—e.g. 40 students assigned to two seminar groups or 20 students within a seminar assigned to working groups with two or four students each—in reasonable time to (near) optimality.

As our model formulation is able to solve larger instances to near optimality, the model can help to evaluate heuristic approaches for the general MDGP by testing them on the version with attribute values even for a larger number of attributes.

This paper shows that it is worth to have a closer look on the data structure of the MDGP. Thus, a direction for future research may also be to investigate how instances with binary or integer attributes or other special data structures and their combinations over several attributes, which occur in practical applications, can be tackled by made-to-measure solution approaches.

Funding Open Access funding enabled and organized by Projekt DEAL. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Availability of data and material The test instances were generated by the author in *GAMS*.

Declaration

Conflicts of interest/Competing interests There are no interests to declare.

Code availability The *GAMS* implementation is saved on a server of the University of Hamburg.

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

References

- Amirgaliyeva, Z., Mladenović, N., Todosijević, R., & Urošević, D. (2017). Solving the maximum min-sum dispersion by alternating formulations of two different problems. *European Journal of Operational Research*, 260(2), 444–459.
- Aringhieri, R., Cordone, R., & Grosso, A. (2015). Construction and improvement algorithms for dispersion problems. *European Journal of Operational Research*, 242(1), 21–33.
- Baker, B., & Benn, C. (2001). Assigning pupils to tutor groups in a comprehensive school. *Journal of the Operational Research Society*, 52(6), 623–629.

- Beheshtian-Ardekani, M., & Mahmood, M. A. (1986). Education development and validation of a tool for assigning students to groups for class projects. *Decision Sciences*, 17(1), 92–113.
- Brimberg, J., Mladenović, N., & Urošević, D. (2015). Solving the maximally diverse grouping problem by skewed general variable neighborhood search. *Information Sciences*, 295, 650–675.
- Brimberg, J., Janičijević, S., Mladenović, N., & Urošević, D. (2017). Solving the clique partitioning problem as a maximally diverse grouping problem. *Optimization Letters*, 11(6), 1123–1135.
- Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201(3), 921–932.
- Caserta, M., & Voß, S. (2013). Workgroups diversity maximization: A metaheuristic approach. In *International Workshop on Hybrid Metaheuristics*, Springer, pp. 118–129
- Dias, T. G., & Borges, J. (2017). A new algorithm to create balanced teams promoting more diversity. *European Journal of Engineering Education*, 42(6), 1365–1377.
- Fan, Z., Chen, Y., Ma, J., & Zeng, S. (2011). Erratum: A hybrid genetic algorithmic approach to the maximally diverse grouping problem. *Journal of the Operational Research Society*, 62(7), 1423–1430.
- Feo, T. A., & Khellaf, M. (1990). A class of bounded approximation algorithms for graph partitioning. *Networks*, 20(2), 181–195.
- Fernández, E., Kalcics, J., & Nickel, S. (2013). The maximum dispersion problem. *Omega*, 41(4), 721–730.
- Gallego, M., Laguna, M., Martí, R., & Duarte, A. (2013). Tabu search with strategic oscillation for the maximally diverse grouping problem. *Journal of the Operational Research Society*, 64(5), 724–734.
- Grötschel, M., & Wakabayashi, Y. (1989). A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45(1), 59–96.
- Heitmann, H., & Brüggemann, W. (2014). Preference-based assignment of university students to multiple teaching groups. *OR Spectrum*, 36(3), 607–629.
- Johnes, J. (2015). Operational research in education. *European Journal of Operational Research*, 243(3), 683–696.
- Krass, D., & Ovchinnikov, A. (2006). The university of toronto's rotman school of management uses management science to create mba study groups. *Interfaces*, 36(2), 126–137.
- Krass, D., & Ovchinnikov, A. (2010). Constrained group balancing: Why does it work. *European Journal of Operational Research*, 206(1), 144–154.
- Lai, X., & Hao, J. K. (2016). Iterated maxima search for the maximally diverse grouping problem. *European Journal of Operational Research*, 254(3), 780–800.
- Lai, X., Hao, J. K., Fu, Z. H., & Yue, D. (2020). Neighborhood decomposition based variable neighborhood search and tabu search for maximally diverse grouping. *European Journal of Operational Research*, 289, 1067.
- Mingers, J., & O'Brien, F. A. (1995). Creating student groups with similar characteristics: a heuristic approach. *Omega*, 23(3), 313–321.
- Palubeckis, G., Ostreika, A., & Rubliauskas, D. (2015). Maximally diverse grouping: an iterated tabu search approach. *Journal of the Operational Research Society*, 66(4), 579–592.
- Papenberg, M., & Klau, G. W. (2021). Using anticlustering to partition data sets into equivalent parts. *Psychological Methods*, 26(2), 161.
- Ramos-Figueroa, O., Quiroz-Castellanos, M., Mezura-Montes, E., & Schütze, O. (2020). Metaheuristics to solve grouping problems: A review and a case study. *Swarm and Evolutionary Computation*, 53, 100643.
- Rodriguez, F. J., Lozano, M., García-Martínez, C., & González-Barrera, J. D. (2013). An artificial bee colony algorithm for the maximally diverse grouping problem. *Information Sciences*, 230, 183–196.
- Rubin, P. A., & Bai, L. (2015). Forming competitively balanced teams. *IIE Transactions*, 47(6), 620–633.
- Salem, KH., & Kieffer, Y. (2020). An experimental study on symmetry breaking constraints impact for the one dimensional bin-packing problem. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, IEEE., pp 317–326
- Schulz, A. (2021a). The balanced maximally diverse grouping problem. In: *Arne Schulz: Selected topics on balanced assignment problems (dissertation)*
- Schulz, A. (2021b). The balanced maximally diverse grouping problem with block constraints. *European Journal of Operational Research*, 294(1), 42–53.
- Schulz, A. (2021c). Scheduling elective surgeries on a weekly level to minimize expected non-elective waiting time. In: *Arne Schulz: Selected topics on balanced assignment problems (dissertation)*
- Schulz, A., & Fliedner, M. (2021). Resource scheduling for unplanned high priority jobs. In *Arne Schulz: Selected topics on balanced assignment problems (dissertation)*
- Singh, K., & Sundar, S. (2019). A new hybrid genetic algorithm for the maximally diverse grouping problem. *International Journal of Machine Learning and Cybernetics*, 10(10), 2921–2940.
- Weitz, R., & Lakshminarayanan, S. (1997). An empirical comparison of heuristic and graph theoretic methods for creating maximally diverse groups, vlsi design, and exam scheduling. *Omega*, 25(4), 473–482.

Weitz, R., & Lakshminarayanan, S. (1998). An empirical comparison of heuristic methods for creating maximally diverse groups. *Journal of the Operational Research Society*, 49(6), 635–646.

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